AT-Jenie
Reference Manual

JN-RM-2038
Revision 1.7
27 August 2009

Contents

About this Manual

This manual provides key reference information for developers using the Jenie interface to produce wireless network applications for the Jennic JN513x wireless microcontroller. The manual details the AT-Jenie serial command set which provides an easy-to-use interface for controlling a wireless network.

Tip: You should use this Reference Manual in conjunction with the *AT-Jenie User Guide (JN-UG-3043)*, which provides both relevant concept information and practical guidance on using AT-Jenie to develop wireless network applications.

Tip: Jenie is alternatively available in the form of an Application Programming Interface (API) containing C functions. The Jenie API is intended for programmers with a good knowledge of C and is described in separate documentation - the *Jenie API User Guide (JN-UG-3042)* and *Jenie API Reference Manual (JN-RM-2035)*.

Organisation

This manual consists of 4 chapters and 4 appendices, as follows:

- Chapter 1 introduces AT-Jenie, details the AT-Jenie command format and provides a command list with cross-references to the command descriptions in the remainder of the manual.
- Chapter 2 describes the AT-Jenie network control commands.
- Chapter 3 describes the AT-Jenie hardware control commands.
- Chapter 4 details the macros and function to add custom commands to the AT-Jenie command parser.
- The appendices provide ancillary information needed to use AT-Jenie: responses, hardware events and data transmission flags. Mappings are also provided between the AT-Jenie commands and C functions of the Jenie API.

Conventions

Files, folders, functions and parameter types are represented in **bold** type.

Function parameters are represented in *italics* type.

Code fragments are represented in the `Courier New` typeface.

This is a **Tip**. It indicates useful or practical information.

This is a **Note**. It highlights important additional information.

*This is a **Caution**. It warns of situations that may result in equipment malfunction or damage.*

Acronyms and Abbreviations

API	Application Programming Interface
JenNet	Jennic Network
MAC	Media Access Control
PAN	Personal Area Network
UART	Universal Asynchronous Receiver Transmitter

Related Documents

[1] AT-Jenie User Guide (JN-UG-3043)

[2] JenNet Stack User Guide (JN-UG-3041)

[3] IEEE 802.15.4 Wireless Networks User Guide (JN-UG-3024)

Feedback Address

If you wish to comment on this manual, or any other Jennic user documentation, please provide your feedback by writing to us (quoting the manual reference number and version) at the following postal address or e-mail address:

Applications
Jennic Ltd
Furnival Street
Sheffield S1 4QT
United Kingdom

doc@jennic.com

1. AT-Jenie Overview

Jennic's proprietary Jenie software provides an easy-to-use interface for developing wireless network applications for the Jennic JN513x wireless microcontrollers. AT-Jenie is a simplified interface comprising serial commands that can be sent to a Jennic JN513x wireless microcontroller from an application or from a command interface (such as HyperTerminal).

AT-Jenie offers functionality for controlling the network and the Jennic hardware within the network nodes. This functionality is outlined in Section 1.1.

The format of an AT-Jenie command is detailed in Section 1.2.

An alphabetic look-up table of the AT-Jenie commands is provided in Section 1.3, which provides cross-references to the command descriptions in the rest of the manual.

Note: For a complete introduction to AT-Jenie, refer to the *AT-Jenie User Guide (JN-UG-3043)*.

1.1 AT-Jenie Functionality

The AT-Jenie functionality are divided into network control and hardware control (of both JN513x on-chip peripherals and carrier board resources). These two areas are outlined in the sub-sections below.

1.1.1 Network Control

The functionality for controlling the network is divided into three main areas:

- **Management tasks:**
  - Configure and initialise network
  - Start a device as a Co-ordinator, Router or End Device
  - Determine whether a Router or Co-ordinator is accepting join requests
  - Advertise local node services and seek remote node services
  - Establish bindings between local and remote node services
  - Handle stack management events
- **Data transfer tasks:**
  - Send data to a remote node or broadcast data to all Router nodes
  - Send data to a bound service on a remote node
  - Handle stack data events
- **System tasks:**
  - Configure and start sleep mode
  - Configure, start and stop the radio transmitter
  - Obtain the version number of a component on the node
  - Handle hardware events

The AT-Jenie network control commands are described in Chapter 2.

1.1.2 Hardware Control

Jenie includes software for the control of hardware resources on the JN513x wireless microcontroller and carrier boards (such as those supplied in JN513x evaluation kits). These resources includes:

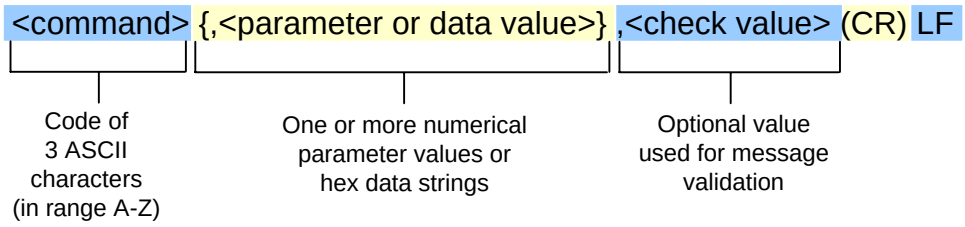
- **JN513x Integrated Peripherals:**
  - ADC
  - DACs
  - Comparators
  - Digital I/O (DIOs)
  - Timers
  - Wake timers
- **Board Resources:**
  - LEDs
  - Switches
  - Sensors
  - LCD screen

The AT-Jenie hardware control commands are described in Chapter 3.

1.2 AT-Jenie Command Format

This section gives the format of an AT-Jenie command. The available commands are listed in Section 1.3, and detailed in Chapter 2 and Chapter 3.

The general format of an AT-Jenie command is illustrated below (the format of a response is similar):



The above fields are described in more detail in the table below.

Field	Content	Representation
<command>	Code representing the AT-Jenie command. If the code is not recognised by target node, the command is rejected and an error returned.	Three ASCII characters - alphabetic, upper case (in range A-Z).
<parameter value>	A parameter value. Depending on the command, there may be several parameters and therefore a sequence of values. Each value must be preceded by a comma.	ASCII characters representing a number in either decimal or hexadecimal. The number system is selected in the command parser configuration. • If decimal is selected (default), hex values can still be used but must be preceded by the 'x' character (not required if hex has been selected as the number system). • If hex is selected, decimal values can still be used but must be preceded by the 'd' character (not required if decimal has been selected as the number system).
<data value>	A string representing the payload data of the command. Each string must be preceded by a comma.	A hexadecimal value expressed as an even number of hex digits (not preceded by 'x'). If the data string is enclosed in quotes ("), all ASCII characters except the quote mark itself are accepted - in received data, each character is decoded as two hex digits.
<check value>	Check value obtained by an XOR operation on all the ASCII characters (including commas and spaces) from the preceding fields of the command string. The value is recalculated by the command parser. If the two results differ, the command has been corrupted and is rejected. The check value must be preceded by a comma, but this is not included in the calculation.	This value is optional.
<CR>	Carriage Return character. This is ignored by the parser when it appears immediately before the Line Feed character.	0x0D
LF	Line Feed character. This indicates the end of the command.	0x0A

Table 1: Fields of AT-Jenie Command

The following are examples of AT-Jenie commands (check value not used):

Example 1

The following command sets the power level of the radio transmitter on a standard module to -12 dBm, assuming decimal mode:

Command: `RDP,-12,0 <LF>`

Return: `OK <LF>`

Example 2

The following command binds Service 1 on the local node and Service 7 on the node with hex network address 0x0123456789ABCDEF.

If working in decimal mode:

Command: `BND,1,x0123456789ABCDEF,7 <LF>`

Response: `OK <LF>`

If working in hex mode, the “x” can be omitted in the hex address:

Command: `BND,1,0123456789ABCDEF,7 <LF>`

Response: `OK <LF>`

Example 3

The following command sends a message to a bound service on another node. The local service has ID 23 and the data is of length 5 bytes, consisting of bytes 0x12, 0x34, 0x56, 0x78 and 0x9A. The message is to be sent with an acknowledgement request.

If working in decimal mode and the send is successful:

Command: `SDS,23,123456789A,5,1 <LF>`

Response: `OK <LF>`

Deferred Response: `PKS <LF>`

If working in hex mode and the send is unsuccessful:

Command: `SDS,17,123456789A,5,1 <LF>`

Response: `OK <LF>`

Deferred Response: `PKF <LF>`

1.3 AT-Jenie Command List

The table below provides an alphabetic list of the AT-Jenie commands. For each command, the table indicates its category (this corresponds to the heading of the section where the command description can be found) as well as the table that contains the command description.

Command	Category/Section Heading	Table/Page
BGF	Switches	Table 17 on page 40
BGH	Sensors	Table 16 on page 39
BGL	Sensors	Table 16 on page 39
BGR	Switches	Table 17 on page 40
BGT	Sensors	Table 16 on page 39
BGV	Sensors	Table 16 on page 39
BLF	LEDs	Table 18 on page 41
BLO	LEDs	Table 18 on page 41
BND	Service Discovery and Binding	Table 5 on page 21
BTX	LCD Screen (Controller Board Only)	Table 19 on page 41
CCF	Command Parser Configuration	Table 3 on page 18
CCS	Command Parser Configuration	Table 3 on page 18
CFG	Node Configuration and Start-up	Table 4 on page 19
CFP	Node Configuration and Start-up	Table 4 on page 19
GAS	Node Configuration and Start-up	Table 4 on page 19
GTV	Miscellaneous	Table 10 on page 25
INI	Node Configuration and Start-up	Table 4 on page 19
KEY	Node Configuration and Start-up	Table 4 on page 19
LVE	Miscellaneous	Table 10 on page 25
OAD	Miscellaneous	Table 10 on page 25
PAC	Analogue Resources (ADC and DACs)	Table 11 on page 29
PAD	Analogue Resources (ADC and DACs)	Table 11 on page 29
PAE	Analogue Resources (ADC and DACs)	Table 11 on page 29
PAO	Analogue Resources (ADC and DACs)	Table 11 on page 29
PAR	Analogue Resources (ADC and DACs)	Table 11 on page 29
PAS	Analogue Resources (ADC and DACs)	Table 11 on page 29
PCD	Comparators	Table 12 on page 31

Table 2: List of AT-Jenie Commands

Command	Category/Section Heading	Table/Page
PCE	Comparators	Table 12 on page 31
PCI	Comparators	Table 12 on page 31
PCS	Comparators	Table 12 on page 31
PCW	Comparators	Table 12 on page 31
PDD	Digital I/Os	Table 13 on page 32
PDE	Digital I/Os	Table 13 on page 32
PDO	Digital I/Os	Table 13 on page 32
PDP	Digital I/Os	Table 13 on page 32
PDR	Digital I/Os	Table 13 on page 32
PDS	Digital I/Os	Table 13 on page 32
POL	Data Transmission	Table 6 on page 22
PTC	Timers	Table 14 on page 34
PTD	Timers	Table 14 on page 34
PTE	Timers	Table 14 on page 34
PTF	Timers	Table 14 on page 34
PTG	Timers	Table 14 on page 34
PTR	Timers	Table 14 on page 34
PTX	Timers	Table 14 on page 34
PWC	Wake Timers	Table 15 on page 36
PWE	Wake Timers	Table 15 on page 36
PWF	Wake Timers	Table 15 on page 36
PWG	Wake Timers	Table 15 on page 36
PWS	Wake Timers	Table 15 on page 36
PWX	Wake Timers	Table 15 on page 36
RDP	Miscellaneous	Table 10 on page 25
REG	Service Discovery and Binding	Table 5 on page 21
RQS	Service Discovery and Binding	Table 5 on page 21
RST	Miscellaneous	Table 10 on page 25
SAS	Node Configuration and Start-up	Table 4 on page 19
SCN	Miscellaneous	Table 10 on page 25
SDS	Data Transmission	Table 6 on page 22
SLP	Sleep Mode (End Devices Only)	Table 9 on page 24

Table 2: List of AT-Jenie Commands

Command	Category/Section Heading	Table/Page
SND	Data Transmission	Table 6 on page 22
SSP	Sleep Mode (End Devices Only)	Table 9 on page 24
STR	Node Configuration and Start-up	Table 4 on page 19
TCL	Tunnelling	Table 8 on page 23
TCM	Tunnelling	Table 8 on page 23
TCN	Tunnelling	Table 8 on page 23
TOP	Tunnelling	Table 8 on page 23
UBN	Service Discovery and Binding	Table 5 on page 21

Table 2: List of AT-Jenie Commands

2. Network Control Commands

This chapter details the AT-Jenie network control commands. These commands are largely concerned with network initialisation and control tasks, and are divided into the following categories:

- Command Parser Configuration
- Node Configuration and Start-up
- Service Discovery and Binding
- Data Transmission
- Tunnelling
- Sleep Mode
- Miscellaneous

A separate section is presented for each of the above categories. The commands for a category are listed in a table detailing the command codes, descriptions, parameters and responses.

Note that:

- Some command parameters are specific to certain node types. These are indicated as follows: C - Co-ordinator, R - Router, E - End Device.
- All command parameters must be assigned values, even if a particular parameter is not applicable to the host node type (can usually be set to 0).
- The parameters for each command are listed in the order they must appear in the command string.
- Command parameters must be set according to the <parameter value> format described in Section 1.2, apart from the payloads of the **SND** and **SDS** commands, which must be set according to the <data value> format.
- The response codes are shown in the final column of the table. This is usually OK or ERR or some variation, but OK may be later followed by a “deferred response” (indicated after an arrow in the table). All responses are detailed in Appendix A.

Note: For a description of the AT-Jenie command format, refer to Section 1.2.

2.1 Command Parser Configuration

The table below lists and describes the AT-Jenie commands for configuring the command parser.

Command	Description	Parameters	Responses
CCF	Configures AT-Jenie command parser. Sets: • Data rate for serial link between UARTs • Number system for numerical values • Check value enable/disable for verification • Command echo enable/disable • Coding method for command string Note that it is not possible to implement data rates precisely - for example, the default rate of 115200 bps is implemented as 115942 bps. Also note that the OK/ERR response is sent back before changes are made.	**Data rate** Values 1-16000000 bps (default: 115200) **Number system** 0: Decimal (default) 1: Hexadecimal **Check value enable** 0: Off (default) 1: On **Command echo (to screen)** 0: Off 1: On (default) **Coding method for commands** 0: ASCII (default) 1: Binary (not yet supported)	OK ERR
CCS	Stores current AT-Jenie command parser settings as new default or restores default settings (saved or factory)	**Save/reset setting** 0: Restore saved default settings 1: Restore factory default settings 2: Save current settings to Flash	OK ERR

Table 3: Command Parser Configuration Commands

2.2 Node Configuration and Start-up

The table below lists and describes the AT-Jenie commands for configuring and starting nodes.

Key: C - Co-ordinator, R - Router, E - End Device

Command	Description	Parameters	Responses
CFG	Configures certain network parameters on node. Sets: • Channels to be included by Router or End Device in the scan for auto-channel selection. Only applicable to Co-ordinator if auto-channel selection enabled using **INI** • Maximum permissible number of child nodes • Maximum number of End Device children (nodes that are capable of sleeping). The remaining child slots are reserved exclusively for Routers, although any number of children can be Routers • Number of failed communications before parent or child considered to be lost • Timeout period for communication (excluding data polling) from an End Device child. If no message is received from the End Device within this period, the child is assumed lost.	**Channels for auto-channel selection** 32-bit bitmap in which bits 11 to 26 specify channels for scan: bit 11 set ⇒ include Channel 11, etc (default: all channels) If no scan enabled, set to 0x07FFF800 **Max. number of children** (C,R) 0-16 (default: 10) **Max. End Device children** (C,R) 0-8 (default: 8) **Max. failures before orphaning** 1-255 (default: 5) Must be set to 2 or higher on a parent (R or C) with Router children **End Device activity timeout** (C,R) 32-bit value, in 100-ms periods (default: 0 - timeout disabled)	OK ERR
CFP	Configures certain network parameters on node. Sets: • Period for auto-pings generated by a Router (to its parent Router or Co-ordinator) • Number of sleep cycles between auto-pings generated by an End Device (to its parent Router or Co-ordinator) • Amount of time following a failed scan that an End Device waits (sleeps) before starting another scan • Time between consecutive polls when an End Device auto-polls its parent for data. A zero value disables auto-polling. • Maximum number of hops in a broadcast (note that broadcast messages are not delivered to End Devices)	**Router ping period** (R) 0: Pings disabled (default) 1-6553 seconds **Sleep cycles between pings** (E) 0: Pings disabled 1-255 (default: 1) **Scan sleep** (E) 200-4294967275 (default: 10000), in milliseconds Values less than 1000 are not recommended for a large network **Poll period** (E) 32-bit value, in 100-ms periods Set to 0 to disable auto-polling **Max. number of hops for broadcast** 0-255	OK ERR

Table 4: Node Configuration and Start-up Commands

INI	Configures and initialises the device. Sets: • PAN ID of the network to be created (C) • Radio channel to adopt for network or auto-channel select (configured using **CFG**) • Network Application ID of the network to be created (C) or of the network to be found (R,E) • The option to automatically restore application and network context data from external non-volatile memory (previously saved with **SCN**), following power loss • The routing option of the node (Co-ordinator or Router). Always disable for End Device	**PAN ID** (C) 16-bit value **2400-MHz radio channel** (C) 0: Auto-channel selection (default) 11-26: Specific channel **Network Application ID** 32-bit value **Restore Context** 0: Disable 1: Enable **Routing** 0: Disable 1: Enable	OK ERR
STR	Starts the device as the specified node type. In the case of the Co-ordinator, this command starts the network.	**Node type** 0: Co-ordinator 1: Router 2: End Device	OK -> NTU ERR
SAS	Configures the ability of a node to allow other nodes to join the network through it. 'Permit joining' is enabled by default at start-up.	**Permit joining** (C,R) 0: Disable 1: Enable	OK ERR
GAS	Gets the current setting of "permit joining".	None	OKA ERR
KEY	Enables security and sets a key value for encrypting/decrypting data during communications between the local node and the specified remote node - that is, the local node will encode the data with the specified key and the remote will decode the data with the same key. Note that this function must therefore also be called on the remote node to set the same key value. In the current release, the specified security key is used for all node-to-node communications in the network (the specified address is ignored).	**Security key** 128-bit value **Address of remote node** 64-bit IEEE/MAC address (ignored in current release)	OK ERR

Table 4: Node Configuration and Start-up Commands

2.3 Service Discovery and Binding

The table below lists and describes the AT-Jenie commands for implementing service discovery and binding. Note that Service 32 is reserved for tunnelling.

Command	Description	Parameters	Responses
REG	Registers list of services of node so that they can be found by other nodes (through **RQS**).	**Services** 32-bit value: bit 0 set for Service 1, bit 1 set for Service 2, etc	OK -> RSR ERR
RQS	Sends list of requested services to other nodes. You must specify whether a remote node must reply if it has any of the requested services or all of the requested services. Replies are received as **SRR** responses - if no nodes have the required services, no **SRR** responses are received.	**Services** 32-bit value: bit 0 set for Service 1, bit 1 set for Service 2, etc. **Type of match** 0: Any of specified services 1: All of specified services	OK -> SRR ERR
BND	Creates a binding between a local service and a service on a remote node to simplify future communications between the services. Can be used multiple times to bind a local source service to several destination services. However, in AT-Jenie v1.4 or lower, you are advised not to bind a source service to more than four destination services.	**Local service** Service ID, value in range 1-32 **Address of remote node** 64-bit IEEE/MAC address **Remote service** Service ID, value in range 1-32	OK ERR
UBN	Unbinds two services previously bound using the command **BND**.	**Local service** Service ID, value in range 1-32 **Address of remote node** 64-bit IEEE/MAC address **Remote service** Service ID, value in range 1-32	OK ERR

Table 5: Service Discovery and Binding Commands

2.4 Data Transmission

The table below lists and describes the AT-Jenie commands for sending data.

Command	Description	Parameters	Responses
SND	Sends message to the specified target node. The flags allow the following features to be enabled/disabled for the send: • Silent send • Broadcast (also set target address to 0) • Security • Acknowledgement (**ACK**) These features are described in Appendix C. The maximum amount of message payload data depends on the type of transmission and whether security has been enabled (using the **KEY** command), as detailed in Table 7 below.	**Target node address** 64-bit IEEE/MAC address (set to 0 for a broadcast, when enabled using flag, or to send to Co-ordinator) **Payload** Payload data string (for format, see Table 1 on page 12) **Payload length** Number of bytes in payload (for maxima, see Table 7 below) **Flags** Value in range 0-15, as detailed in Appendix C.	OK -> PKS OK -> PKF ERR
SDS	Sends message from local service to the bound service(s). The flags allow the following features to be enabled/disabled for the send: • Silent send • Broadcast (ignored for **SDS**) • Security • Acknowledgement (**SAK**) These features are described in Appendix C. The maximum amount of message payload data depends on whether security has been enabled (using the **KEY** command), as detailed in Table 7 below.	**Local service** Service ID, value in range 1-32 **Payload** Payload data string (for format, see Table 1 on page 12) **Payload length** Number of bytes in payload (for maxima, see Table 7 below) **Flags** Value in range 0-15, as detailed in Appendix C.	OK -> PKS OK -> PKF ERR
POL	Polls the parent node for any pending data (issued by End Devices only)	None	OK -> PLC -> DAT DTS TNE ERR

Table 6: Data Transmission Commands

Command	Type of Transmission	Max. Data for Security Disabled	Max. Data for Security Enabled
SND	Broadcast to all nodes	89 bytes	68 bytes
	Unicast to Co-ordinator	90 bytes	69 bytes
	Unicast to any other node	82 bytes	61 bytes
SDS	Unicast to any node	74 bytes	53 bytes

Table 7: Maximum Payload Sizes

2.5 Tunnelling

The table below lists and describes the AT-Jenie commands for implementing tunnelling. Note that Service 32 is reserved for the tunnelling service.

Command	Description	Parameters	Responses
TCN	Connects a tunnel from the specified local service to the tunnelling service (Service 32) on the specified remote node. Note that these services do not need to be bound.	**Local tunelling service** Service ID, value in range 1-32 **Address of remote node** 64-bit IEEE/MAC address **Remote tunelling service** Service ID, must always be 32	OK -> PKS OK -> PKF ERR BSY
TCM	Sends an AT-Jenie command string through a tunnel to a remote node, where the tunnel has been previously set up using **TCN**. Note that it is not possible to tunnel a command with an ASCII string (between quotes) as its payload (e.g. **SND**). In such a case, the payload must be represented in hex form.	**AT-Jenie command string** ASCII command string delimited at both ends by quotes (“). Note that escaped characters must be preceded with \	OK -> TNR ERR
TOP	Opens a communication channel in a tunnel set up using **TCN**. The **TOP** command must be executed on the remote node, and sent to the node as a tunnelled command using **TCM**.	None	OK -> PKS OK -> PKF ERR BSY
TCL	Closes a communication channel in a tunnel previously opened using **TOP**. The **TCL** command must be executed on the remote node, and sent to the node as a tunnelled command using **TCM**.	None	OK -> PKS OK -> PKF ERR BSY

Table 8: Tunnelling Commands

2.6 Sleep Mode (End Devices Only)

The table below lists and describes the AT-Jenie commands for implementing sleep mode on End Devices.

Key: C - Co-ordinator, R - Router, E - End Device

Command	Description	Parameters	Responses
SSP	Sets the sleep period of one sleep cycle.	**Sleep period** (E) Duration, in milliseconds	OK ERR
SLP	Puts the node into sleep mode (as soon as all current tasks have been completed). Sleep can be entered with or without timers running, and with or without on-chip memory held. Alternatively, deep sleep can be entered (timers not running, memory not held). Note that 'doze mode' of the JN513x device is not supported by AT-Jenie.	**Timers and on-chip memory state** (E) 0: Timers running, memory held 1: Timers running, memory not held 2: Timers not running, memory held 3: Timers not running, memory not held 4: Deep sleep	OK ERR

Table 9: Sleep Mode Commands

2.7 Miscellaneous

The table below lists and describes the remaining AT-Jenie commands, not described in the previous sections.

Command	Description	Parameters	Responses
RDP	Sets the power level of the radio transmitter and enables a high-power module (if one is to be used). Issue this command after **STR**. Note that: • You must specify your module type (standard or high-power) • You must always issue this command when using a high-power module • For standard modules, the permissible range is -30 to 0 dBm • For high-power modules, the permissible range is -12 to +18 dBm • The values are applied in steps of 6 dBm • You can also use this command to switch the transmitter on or off Also note that 'boost mode' of the JN513x device is not supported by AT-Jenie.	**Radio power level** Value in range -30 to +18 dBm Value < -30: Switch transmitter off Value > +18: Switch transmitter on (Do not include the '+' symbol in your setting, as this will cause an error) **Module Type** 0: Standard module 1: High-power module * * Note that in AT-Jenie networks, high-power modules cannot be used on End Devices that sleep.	OK ERR
GTV	Gets the version number of the specified component. The version number is returned as an 8-digit hex value in the **OKV** response. The interpretation of this version number depends on the component - refer to Table 21 on page 67.	**Component** 0: Jenie interface 1: JenNet software 2: IEEE 802.15.4 software 3: JN5139 chip	OKV ERR
LVE	Removes node from network.	None	OK ERR
RST	Resets node.	None	OK ERR
OAD	Invalidates Flash memory and resets node.	None	OK ERR
SCN	Saves or deletes application and network context data in external non-volatile memory (e.g. Flash), as follows: • Saves context data to non-volatile memory • Deletes previously saved context data and disables save/restore context feature The save/restore context feature must have been previously enabled in **INI**.	**Save/Delete Context** 0: Delete context and disable feature 1: Save context	OK ERR

Table 10: Miscellaneous Commands

3. Hardware Control Commands

This chapter details the AT-Jenie hardware control commands. These commands are presented in two categories, as follows:

- **Peripheral Commands:** These are commands used to control the on-chip peripherals of the JN513x wireless microcontroller - see Section 3.1.
- **Board Commands:** These are commands used to control resources on the Jennic carrier boards for JN513x-based modules - see Section 3.2.

The commands are listed in tables detailing the command codes, descriptions, parameters and responses.

Note that:

- All command parameters must be assigned values, even if a particular parameter is not applicable to the host node type (can usually be set to 0).
- The parameters for each command are listed in the order they must appear in the command string.
- The response codes are shown in the final column of the table. This is usually OK or ERR or some variation, but OK may be later followed by a “deferred response” (indicated after an arrow in the table). All responses are detailed in Appendix A.

Note: For a description of the AT-Jenie command format, refer to Section 1.2.

3.1 Peripheral Commands

This section describes the peripheral commands of the AT-Jenie command set. These commands are used to interface with the on-chip peripherals of the Jennic JN513x wireless microcontroller.

These peripherals include:

- Analogue resources (ADC and DACs)
- Comparators
- Digital I/O (DIOs)
- Timers
- Wake timers

A separate sub-section is presented below for each of the above peripheral categories.

Before using the AT-Jenie peripheral commands, you are advised to consult the following documentation for information on the JN513x integrated peripherals:

- Refer to the hardware peripherals chapter of the *AT-Jenie User Guide (JN-UG-3043).*
- If you require more details, refer to the *JN513x Data Sheet (JN-DS-JN513x)*.

In addition, it is worth noting that the AT-Jenie peripheral commands provide functionality also covered by the Jennic Integrated Peripherals API, described in the *Integrated Peripherals API Reference Manual (JN-RM-2001)*.

Note: The peripheral commands are executed on the JN513x microcontroller which contains the peripherals to be controlled. However, these commands can be used in conjunction with tunnelling to remotely control JN513x resources (that is, to control the JN513x peripherals on one node from another node). Tunnelling is described in the *AT-Jenie User Guide (JN-UG-3043)*.

3.1.1 Analogue Resources (ADC and DACs)

The table below lists and describes the AT-Jenie commands for interacting with the JN513x ADC and DACs (DAC 1 and DAC 2).

Command	Description	Parameters	Responses
PAC	Configures common parameters shared by the analogue peripherals (ADC and DACs). Note that: • The regulator minimises digital noise and is sourced from the analogue supply pin VDD1. • If enabled, an interrupt is generated after each individual conversion. • The sampling period is dependent on the clock period specified. • For the ADC, the input signal is integrated over *3 x Sampling Interval*. • For the ADC and DACs, the total conversion period (for a single value) is given by *(3 x sampling interval) + (14 x clock period)*.	**Regulator control** 0: Off (no power) 1: On **Interrupts** 0: Disable 1: Enable **Sampling interval** 0: 2 x clock period (1/2 x frequency) 1: 4 x clock period (1/4 x frequency) 2: 6 x clock period (1/6 x frequency) 3: 8 x clock period (1/8 x frequency) **Clock frequency** 0: 2 MHz 1: 1 MHz 2: 500 kHz (recommended for ADC) 3: 250 kHz (recommended for DACs) **Reference voltage V_{ref}** 0: Internal 1: External (from the pin VREF)	OK ERR

Table 11: Analogue Resource Commands

PAE	Configures and enables the specified analogue peripheral (ADC or DAC). Note that: • The source of V_{ref} is defined using **PAC**. • For the ADC, the internal voltage monitor measures the voltage on the analogue supply pin VDD1. • For a DAC, the 'output hold' retains the last analogue voltage on the output pin (DAC1 or DAC2) for some time after the DAC has been disabled. • For a DAC, the first value to be converted is specified through this command. Subsequent values must be specified through **PAO**. • Once enabled using this command, the analogue peripheral can be disabled using **PAD**. • Only one DAC can be used at any one time, since the two DACs share resources. If both DACs are to be used, they can be multiplexed.	**Component** 0: DAC 1 1: DAC 2 2: ADC **Analogue voltage range** 0: 0-V_{ref} 1: 0-2V_{ref} **Conversion mode (ADC only)** 0: Single shot mode 1: Continuous mode **Source (ADC only)** 0: Pin ADC1 1: Pin ADC2 2: Pin ADC3 3: Pin ADC4 4: On-chip temperature sensor 5: Internal voltage monitor **Output hold (DAC only)** 0: Off 1: On **Initial input value (DAC only)** 16-bit value (only lower 11 bits used)	OK ERR
PAD	Disables the specified analogue peripheral (DAC or ADC).	**Component** 0: DAC 1 1: DAC 2 2: ADC	OK ERR
PAS	Starts sampling on the ADC analogue input.	None	OK ERR
PAR	Reads the latest digital output from the ADC. The read value is returned in the **OKP** response.	None	OKP ERR
PAO	Used to specify digital input value to be converted by the specified DAC. Note that the first value to be converted is specified through **PAE** and subsequent values through this command.	**Component** 0: DAC 1 1: DAC 2 **New input value** 16-bit value (only lower 11 bits used)	OK ERR

Table 11: Analogue Resource Commands

3.1.2 Comparators

The table below lists and describes the AT-Jenie commands for interacting with the two JN513x comparators (Comparator 1 and Comparator 2).

Command	Description	Parameters	Responses
PCE	Configures and enables the specified comparator. Note that: • The hysteresis voltage selected should be greater than the noise level in the input signal (on the COMP1P or COMP2P pin). • Once enabled using this command, the comparator can be disabled using **PCD**.	**Component** 0: Comparator 1 1: Comparator 2 **Hysteresis voltage** 0: 0 mV 1: ±5 mV 2: ±10 mV 3: ±20 mV **Reference signal** 0: COMP1M or COMP2M pin 1: DAC1 or DAC2 output 2: V_{ref}	OK ERR
PCD	Disables the specified comparator.	**Component** 0: Comparator 1 1: Comparator 2	OK ERR
PCI	Configures and enables interrupts from the specified comparator. If enabled, an interrupt is generated on one of the following conditions (which must be configured): • The input signal rises above the reference signal (plus hysteresis level, if non-zero) • The input signal falls below the reference signal (minus hysteresis level, if non-zero)	**Component** 0: Comparator 1 1: Comparator 2 **Interrupt** 0: Disable 1: Enable **Rising/falling transition** 0: Rising 1: Falling	OK ERR
PCS	Requests the status of the specified comparator. The status is returned in the **OKP** response as: • 0 if input signal is lower than reference signal • Non-zero value if input signal is higher than reference signal	**Component** 0: Comparator 1 1: Comparator 2	OKP ERR
PCW	Requests the status of the wake-up interrupt of the specified comparator. The status is returned in the **OKP** response as: • 0 if wake-up interrupt has not occurred • Non-zero value if wake-up interrupt has occurred	**Component** 0: Comparator 1 1: Comparator 2	OKP ERR

Table 12: Comparator Commands

3.1.3 Digital I/Os

The table below lists and describes the AT-Jenie commands for interacting with the JN513x Digital I/O lines (DIO0-DIO20).

Command	Description	Parameters	Responses
PDD	Defines which DIO pins (DIO0-DIO20) are inputs and which are outputs. Note that the command has no effect for a pin being used by an enabled on-chip peripheral. The input and output pins are specified in separate 32-bit bitmaps, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused). The bit settings for the same DIO pin must not conflict in the two bitmaps (for a conflict, the default is input).	**Inputs** 32-bit bitmap: bit 0 set means DIO0 input, bit 1 set means DIO1 input, etc. **Outputs** 32-bit bitmap: bit 0 set means DIO0 output, bit 1 set means DIO1 output, etc.	OK ERR
PDO	Sets the output status (on or off) of the DIO pins (DIO0-DIO20). Note that the command has no effect for a pin not configured as an output or being used by an enabled on-chip peripheral. The pins that are on and off are specified in separate 32-bit bitmaps, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused). The bit settings for the same DIO pin must not conflict in the two bitmaps (for a conflict, the default is off).	**On** 32-bit bitmap: bit 0 set means DIO0 on, bit 1 set means DIO1 on, etc. **Off** 32-bit bitmap: bit 0 set means DIO0 off, bit 1 set means DIO1 off, etc.	OK ERR
PDP	Sets the pull-ups (on or off) of the DIO pins (DIO0-DIO20). The pull-ups that are on and off are specified in separate 32-bit bitmaps, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused). The bit settings for the same DIO pin must not conflict in the two bitmaps.	**On** 32-bit bitmap: bit 0 set means DIO0 pull-up on, bit 1 set means DIO1 pull-up on, etc. **Off** 32-bit bitmap: bit 0 set means DIO0 pull-up off, bit 1 set means DIO1 pull-up off, etc.	OK ERR
PDR	Requests the status of the DIO input pins (DIO0-DIO20). The status of the input pins is returned in the **OKP** response. The status is represented as a 24-bit bitmap, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 23 are unused and set to 0). Only the bits corresponding to configured inputs are valid.	None	OKP ERR

Table 13: DIO Commands

PDE	Configures and enables/disables wake signals on DIO pins (DIO0-DIO20). Note that the command has no effect for a pin not configured as an input or being used by an enabled on-chip peripheral. The command allows the wake-up signals on the DIO pins to be individually enabled and disabled. Enable and disable are implemented using separate 32-bit bitmaps, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused). The bit settings for the same DIO pin must not conflict in the two bitmaps (for a conflict, the default is disable). An unset (0) bit means leave unchanged. The command also allows configuration of the change of state that will trigger the wake-up signal on each DIO pin (enabled for wake-up) - that is, a low-to-high (rising) transition or high-to-low (falling) transition. The rising and falling cases are configured using separate 32-bit bitmaps, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused). The bit settings for the same DIO pin must not conflict in the two bitmaps (for a conflict, the default is 'rising'). An unset (0) bit means leave unchanged.	**Enable** 32-bit bitmap: bit 0 set means DIO0 wake enable on, bit 1 set means DIO1 wake enable on, etc. **Disable** 32-bit bitmap: bit 0 set means DIO0 wake enable off, bit 1 set means DIO1 wake enable off, etc. **Rising (low to high)** 32-bit bitmap: bit 0 set means DIO0 triggers wake on rising edge, bit 1 set means DIO1 triggers wake on rising edge, etc. **Falling (high to low)** 32-bit bitmap: bit 0 set means DIO0 triggers wake on falling edge, bit 1 set means DIO1 triggers wake on falling edge, etc.	OK ERR
PDS	Requests the wake status of the DIO pins (DIO0-DIO20). The command also clears the wake status of all pins. The status is returned in the **OKP** response as a 32-bit bitmap, where each of bits 0 to 20 represents the corresponding DIO pin (bits 21 to 31 are unused and set to 0). For each DIO pin: • 1 indicates wake signal has been triggered • 0 indicates wake signal has not been triggered Only the bits corresponding to pins on which the wake signal is enabled are valid.	None	OKP ERR

Table 13: DIO Commands

3.1.4 Timers

The table below lists and describes the AT-Jenie commands for interacting with the two JN513x timers (Timer 0 and Timer 1).

Command	Description	Parameters	Responses
PTE	Configures and enables specified timer. Note that: • The clock divisor index (*p*) determines the power of two (2^p) used to divide down the source clock for the timer. • The source clock can be internal (16-MHz system clock) or external (from pin DIO8 for Timer 0 or DIO11 for Timer 1), and can be optionally inverted. • The timer signal can be optionally output on a DIO pin (DIO10 for Timer 0, DIO13 for Timer 1). • DIO pins for timer use (DIO8-10 for Timer 0, DIO11-13 for Timer 1) must be explicitly enabled. • Interrupts can be generated by the timer on each low-to-high transition (output rising) and/or at the end of the timed duration. • Once enabled using this command, the timer can be disabled using **PTD**.	**Component** 0: Timer 0 1: Timer 1 **Clock divisor index** (power of 2) Index in range 0-16 **Interrupt conditions** 0: Output rising (off), timer end (off) 1: Output rising (on), timer end (off) 2: Output rising (off), timer end (on) 3: Output rising (on), timer end (on) **External output** 0: Off 1: On **DIO pins used for timer** 0: Off 1: On **Clock select/polarity** 0: Internal/Normal 1: External/Normal 2: Internal/Inverted 3: External/Inverted	OK ERR
PTD	Disables specified timer.	**Component** 0: Timer 0 1: Timer 1	OK ERR

Table 14: Timer Commands

PTG	Starts the specified timer in the specified mode with the specified pulse cycle. Note that: • During one pulse cycle, the timer signal starts off low and then goes high. • The low period is determined by the 'time to rise'. • The complete pulse period (low and high) is determined by the 'time to fall'. • 'Single shot' mode produces one pulse cycle and stops, while "Repeat" mode produces a train of pulses until the timer is stopped using **PTX**. • 'Delta-Sigma' modes allow the timer to be used as a low-rate DAC that uses the technique of Pulse Width Modulation (PMW). • If Delta-Sigma mode is used, an RC circuit must be inserted between the output pin (DIO10 for Timer 0, DIO13 for Timer 1) and Ground.	**Component** 0: Timer 0 1: Timer 1 **Mode** 0: Single shot 1: Repeat 2: Delta-sigma RTZ 3: Delta-sigma NRZ **Time to rise** 16-bit value: Number of clock cycles from start to first low-to-high transition **Time to fall** 16-bit value: Number of clock cycles from start to first high-to-low transition	OK ERR
PTC	Starts specified timer in 'capture' mode. In this mode, an input signal is monitored and measurements made which allow the input pulse width to be determined. Note that: • The input signal must be provided on pin DIO9 for Timer 0 or DIO12 for Timer 1. • The captured measurements are the number of clock cycles to the last low-to-high transition and to the last high-to-low transition of the input signal.	**Component** 0: Timer 0 1: Timer 1	OK ERR
PTR	Requests 'capture' results from specified timer. The results are returned in the **OKP** response as a 32-bit value in which: • The upper 16 bits (bits 31-16) represent the number of clock cycles up to the last low-to-high transition. • The lower 16 bits (bits 15-0) represent the number of clock cycles up to the last high-to-low transition. The width of the last pulse can be calculated from the difference of these results (provided that the results were requested during a low period).	**Component** 0: Timer 0 1: Timer 1	OKP ERR
PTX	Stops specified timer.	**Component** 0: Timer 0 1: Timer 1	OK ERR
PTF	Requests interrupt status of specified timer. The command also clears the interrupt status of the timer. The interrupt status is returned in the **OKP** response as: • 1: Timer has fired and generated an interrupt • 0: Timer has not fired (so no interrupt generated)	**Component** 0: Timer 0 1: Timer 1	OKP ERR

Table 14: Timer Commands

3.1.5 Wake Timers

The table below lists and describes the AT-Jenie commands for interacting with the two JN513x wake timers (Wake Timer 0 and Wake Timer 1).

Command	Description	Parameters	Responses
PWE	Configures and enables the specified wake timer. This commands allows the wake timer interrupt (which is generated when the timer fires) to be enabled/disabled.	**Component** 0: Wake Timer 0 1: Wake Timer 1 **Interrupt** 0: Disable 1: Enable	OK ERR
PWG	Starts the specified wake timer with the specified count value. The wake timer will count down from this value, which is set according to the desired timer duration. Note that the 32-kHz internal clock, which drives the wake timer, may be running up to 30% fast or slow. For accurate timings, you are advised to first calibrate the clock using **PWC** and adjust the specified count value accordingly.	**Component** 0: Wake Timer 0 1: Wake Timer 1 **Count value** 32-bit value: Number of 32-kHz clock cycles (so 32 represents 1 millisecond)	OK ERR
PWX	Stops the specified wake timer. Note that no interrupt is generated.	**Component** 0: Wake Timer 0 1: Wake Timer 1	OK ERR
PWS	Requests which wake timers are active. Note that a wake timer remains active after its countdown has completed. The result is returned in the **OKP** response as a 2-bit bitmap where: • Bit 0 (LSB) represents Wake Timer 0 • Bit 1 (MSB) represents Wake Timer 1 A bit is set to '1' if the corresponding wake timer is active and to '0' if it is not active.	None	OKP ERR

Table 15: Wake Timer Commands

PWC	Requests a calibration of the 32-kHz internal clock (on which the wake timers run) against the more accurate 16-MHz system clock. Note that Wake Timer 0 is used in this calibration and must first be disabled using **PWX**, if necessary. The result, *n*, is returned in an **OKP** response and is interpreted as follows: • *n* = 10000 ⇒ 32-kHz clock running accurately • *n* > 10000 ⇒ 32-kHz clock running slow • *n* < 10000 ⇒ 32-kHz clock running fast Then, if the required timer duration is *T* seconds, the count value *N* that must be specified through **PWG** is given by *N* = (10000/*n*) x 32000 x *T*.	None	OKP ERR
PWF	Requests which wake timers have fired. The command also clears the timers that have fired. The result is returned in the **OKP** response as a 2-bit bitmap where: • Bit 0 (LSB) represents Wake Timer 0 • Bit 1 (MSB) represents Wake Timer 1 A bit is set to '1' if the corresponding wake timer has fired and to '0' if it has not fired.	None	OKP ERR

Table 15: Wake Timer Commands

3.2 Board Commands

This section describes the board commands of the AT-Jenie command set. These commands can be used to interact with resources on the Jennic carrier boards for JN513x-based modules (such as the boards provided in JN513x evaluation/starter kits).

These resources include:

- Sensors (for temperature, humidity, light level and battery level)
- Switches
- LEDs
- LCD screen (controller board only)

A separate sub-section is presented below for each of the above resource categories.

If you require more information on board resources before using the AT-Jenie board commands, refer to the following manuals:

- For information on the controller board in the JN513x evaluation kits, refer to the *Controller Board Reference Manual (JN-RM-2007).*
- For information on the sensor boards in the JN513x evaluation kits, refer to the *Sensor Board Reference Manual (JN-RM-2008).*
- For information on the sensor boards in the JN5139-EK020 AT-Jenie Starter Kit, refer to the *DR1080 Starter Kit Board Reference Manual (JN-RM-2037).*

Note: The sensor boards in the JN5139-EK020 AT-Jenie Starter Kit are different from those in the JN513x evaluation kits, but have similar on-board resources.

In addition, it is worth noting that the AT-Jenie board commands provide functionality also covered by the Jennic Board API, described in the *Board API Reference Manual (JN-RM-2003)*

Note: The board commands are executed on the Jennic board which contains the resources to be controlled. However, these commands can be used in conjunction with tunnelling to remotely control board resources (that is, to control the resources on one node from another node). Tunnelling is described in the *AT-Jenie User Guide (JN-UG-3043)*.

3.2.1 Sensors

The table below lists and describes the AT-Jenie commands for interacting with the on-board sensors (battery level, temperature, light level, humidity).

Command	Description	Parameters	Responses
BGV	Obtains the current voltage of the on-board batteries. The result is returned in an **OKP** response as value in millivolts.	None	OKP
BGT	Obtains the current measurement of the on-board temperature sensor. The result is returned in an **OKP** response as a value in degrees Celsius.	None	OKP
BGL	Obtains the current measurement of the on-board ambient light level sensor. The result is returned in an **OKP** response as a value in the range 0 to 4015. Sensible use of this result may require manual calibration.	None	OKP
BGH	Obtains the current measurement of the on-board humidity sensor. The result is returned in an **OKP** response as a percentage value (in the range 0-100%).	None	OKP

Table 16: Sensor Commands

3.2.2 Switches

The table below lists and describes the AT-Jenie commands for interacting with the on-board switches (SW1 and SW2, and additionally SW3 and SW4 on controller boards).

Command	Description	Parameters	Responses
BGR	Obtains the status (on or off) of all the switches on a sensor board (reduced function device). Note that sensor boards have two switches: SW1 and SW2. The result is returned in an **OKP** response as a bit-map in which each bit represents a switch: • Bit 0 represents switch SW1 • Bit 1 represents switch SW2	None	OKP
BGF	Obtains the status (on or off) of all the switches on the controller board (full function device). Note that the controller board has four switches: SW1, SW2, SW3 and SW4. The result is returned in an **OKP** response as a bit-map in which each bit represents a switch: • Bit 0 represents switch SW1 • Bit 1 represents switch SW2 • Bit 2 represents switch SW3 • Bit 3 represents switch SW4	None	OKP

Table 17: Switch Commands

3.2.3 LEDs

The table below lists and describes the AT-Jenie commands for interacting with the on-board LEDs (D1 and D2, and additionally D3 and D4 on controller boards).

Command	Description	Parameters	Responses
BLO	Illuminates the specified on-board LED. Note that sensor boards have two LEDs (D1 and D2) and controller boards have two additional LEDs (D3 and D4).	**LED Number** 0: LED1 (D1) 1: LED2 (D2) 2: LED3 (D3 - controller board only) 3: LED4 (D4 - controller board only)	OK
BLF	Extinguishes the specified on-board LED. Note that sensor boards have two LEDs (D1 and D2) and controller boards have two additional LEDs (D3 and D4).	**LED Number** 0: LED1 (D1) 1: LED2 (D2) 2: LED3 (D3 - controller board only) 3: LED4 (D4 - controller board only)	OK

Table 18: LED Commands

3.2.4 LCD Screen (Controller Board Only)

The table below describes the AT-Jenie command for interacting with the LCD screen on a controller board.

Command	Description	Parameters	Responses
BTX	Displays the specified text string in the specified position on the LCD screen (controller boards only). The text is left-justified and starts at the row and column specified. No attempt is made to prevent the text from spilling past the end of the current row and, if this occurs, it will wrap around to the next row.	**Text string** ASCII sequence **Row** 0-7: Row on which text will start **Column** 0-127: Column which text will start	OK

Table 19: LCD Command

4. Extending AT-Jenie

This chapter describes the facility to extend the AT-Jenie command set by adding custom commands. This involves defining commands, along with the corresponding C functions that they invoke, and registering the new commands with the AT-Jenie command parser.

The code to add new commands, together with their corresponding API functions, must be included in the AT-Jenie application **ATJenie_App.c**, supplied as part of the Jennic SDK. This application runs on the JN513x wireless microcontroller, receives AT-Jenie commands from the user application and passes them to the parser. If a command is recognised, the parser then invokes the corresponding API function to perform the required task. This is illustrated in Figure 1 below.

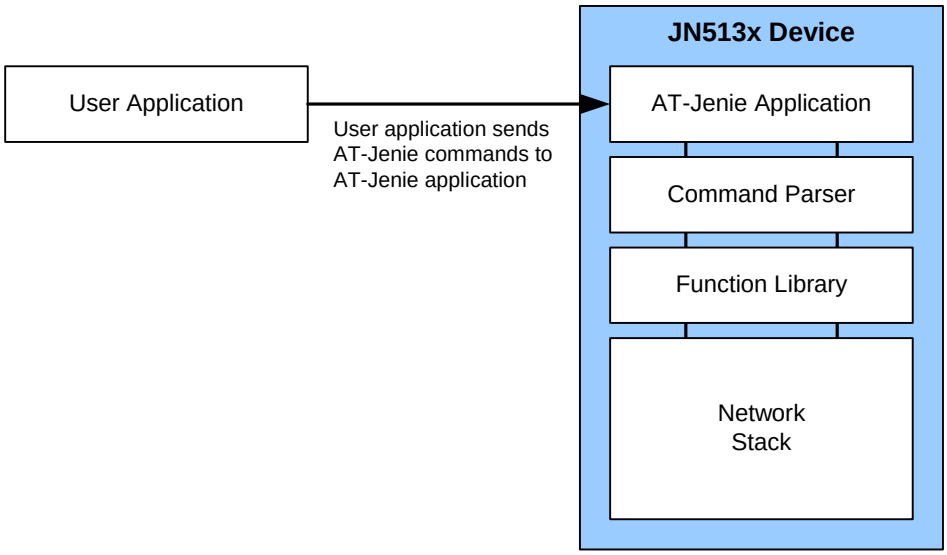


Figure 1: AT-Jenie Application

Note: Once the application **ATJenie_App.c** has been modified, it must be re-built and the resulting binary file must be downloaded to the JN513x device - refer to the *AT-Jenie User Guide (JN-UG-3043)*.

Resources are provided in the Jennic SDK to help you define your own commands:

- Macros that allow you to define your custom commands and associated parameters - they are described in Section 4.1.
- A function which allows you to register your new commands with the AT-Jenie command parser - this is described in Section 4.2.

Guidance on the use of these resources to introduce custom commands is provided in the *AT-Jenie User Guide (JN-UG-3043)*.

The macros/function provided for extending the AT-Jenie command set are listed in the table below, which also provides cross-references to the command descriptions in the remainder of this chapter.

4.1 Defining New Commands

This section details the macros provided in the Jennic SDK that allow you to define custom commands in the Jennic-supplied file **ATJenie_App.c**.

- Section 4.1.1 details the macros used to define commands.
- Section 4.1.2 details the macros used to define the command parameters.

In the code, the parameters must be defined before the commands, as illustrated in the code fragment below, which defines two commands, LIN and LWR, for initialising and controlling an LED respectively.

```
/* Define parameters of commands */
ATJ_DECLARE_CMD_PARAMS(LWR) = {ATJ_CMD_PARAM(E_ATJ_INPUT,
                               E_ATJ_PARAM, 4, ATJ_PARAM_VAL_NONE()),
};

/* Define new commands */
PRIVATE tsATJCommandSet asLCDCommands = {
    ATJ_BEGIN_COMMAND_SET
        ATJ_COMMAND_NP(LIN, vLcdInit,  E_ATJ_OK, NULL),
        ATJ_COMMAND   (LWR, vLcdWrite, E_ATJ_OK, NULL),
    ATJ_END_COMMAND_SET
};
```

New commands can be defined and registered with the AT-Jenie command parser as a command sub-set, comprising commands that may have a common theme (e.g. LED control). This command set is defined in the code using an array of the type **tsATJCommandSet**, where the list of command definitions is delimited by ATJ_BEGIN_COMMAND_SET and ATJ_END_COMMAND_SET (see above example).

Tip: Before defining your commands using the macros detailed in this section, you are advised to consult the corresponding section of the *AT-Jenie User Guide (JN-UG-3043)*.

4.1.1 Macros for Defining Commands

The parser stores its command set in a data structure, where each command has its own element which is itself a structure. The data structure for a command contains the following information:

- 3-character mnemonic code for command
- Availability of command (enabled or disabled)
- Number of parameters for the command
- Set of parameter descriptors (one for each parameter)
- Name of C function which corresponds to the command
- Type of response generated by the command
- Optional function to evaluate the result of command as success or failure

The macros to define commands are listed below, along with their page references:

ATJ_BEGIN_COMMAND_SET

ATJ_BEGIN_COMMAND_SET

Description

This macro is used to start a set of command definitions.

A list of command definitions started with this macro must be terminated with the macro **ATJ_END_COMMAND_SET**.

Parameters

None

ATJ_END_COMMAND_SET

ATJ_END_COMMAND_SET

Description

This macro is used to terminate a set of command definitions.

A list of command definitions terminated with this macro must be started with the macro **ATJ_BEGIN_COMMAND_SET**.

Parameters

None

ATJ_COMMAND

ATJ_COMMAND(*c***,** *api***,** *r***,** *v***)**

Description

This macro is used to define a command which has parameters. The command parameters are defined separately from the command definition, using the macros detailed in Section 4.1.2.

Note that a command with no parameters is defined using the macro **ATJ_COMMAND_NP**.

Parameters

c	3-character mnemonic code for the command (not in quotes)
api	Name of the C function that the command must invoke (this function must also be defined)
r	Response type generated by the command - can be set to one of the following (also see Appendix A.1): E_ATJ_OK (OK) E_ATJ_OKA (OKA) E_ATJ_OKP (OKP) E_ATJ_OKV (OKV) E_ATJ_OKO (OKO)
v	Name of a user-defined validation function that checks the C function return code to determine success or failure (if no such validation function exists, this parameter should be set to NULL). The prototype of the validation function is: **bool_t (*** *pfSuccess* **)(uint64** *rv*, **uint8 *** *au8ParamBuffer* **);**

ATJ_COMMAND_NP

ATJ_COMMAND_NP(*c***,** *api***,** *r***,** *v***)**

Description

This macro is used to define a command which has no parameters.

Note that a command with parameters is defined using the macro **ATJ_COMMAND**.

Parameters

c	3-character mnemonic code for the command (not in quotes)
api	Name of the C function that the command must invoke (this function must also be defined)
r	Response type generated by the command - can be set to one of the following (see Appendix A.1): E_ATJ_OK (OK) E_ATJ_OKA (OKA) E_ATJ_OKP (OKP) E_ATJ_OKV (OKV) E_ATJ_OKO (OKO)
v	Name of a user-defined validation function that checks the C function return code to determine success or failure (if no such validation function exists, this parameter should be set to NULL). The prototype of the validation function is: **bool_t (***_pfSuccess_**)(uint64** _rv_**, uint8 ***_au8ParamBuffer_**);**

4.1.2 Macros for Defining Parameters

The parameter descriptors each contain the following information about the corresponding parameter:

- Direction of parameter (input or output)
- Whether the parameter is a natural type or a structure
- Size of parameter, in bytes
- Optional validator to check the validity of an input parameter value - this validator must be defined as either:
  - a range of integer values within which the input value must lie, or
  - a function which implements a set of validation rules

The macros to define parameters are listed below, along with their page references:

ATJ_DECLARE_CMD_PARAMS

ATJ_DECLARE_CMD_PARAMS(*c*)

Description

This macro is used to declare the parameters of the specified command (which is defined using the macro **ATJ_COMMAND**). The macro is used as follows:

```
ATJ_DECLARE_CMD_PARAMS(c)={Parameter Descriptor1,
                           Parameter Descriptor2,
                           :
                           :
                           Parameter DescriptorN};
```

where the parameter descriptors are each declared using the macro **ATJ_CMD_PARAM**.

Parameters

c	3-character mnemonic code for the command (not in quotes)

ATJ_CMD_PARAM

ATJ_CMD_PARAM(*d***, *st*, *s*, *v*)**

Description

This macro is used to declare a parameter descriptor (in conjunction with the macro **ATJ_DECLARE_CMD_PARAMS**) for a command defined using the macro **ATJ_COMMAND**.

Parameters

d	Direction of parameter (input or output): E_ATJ_INPUT - input parameter E_ATJ_OUTPUT - output parameter
st	Specifies whether the parameter is a built-in type or a user-defined data structure: E_ATJ_PARAM - built-in type E_ATJ_STRUCT(n) - n^{th} data structure
s	Size of the parameter, in bytes
v	Macro used to specify validator for an input parameter (to check that the input value is acceptable): ATJ_PARAM_VAL_RANGE - see macro description ATJ_PARAM_VAL_FUNC - see macro description ATJ_PARAM_VAL_NONE - see macro description

ATJ_PARAM_VAL_RANGE

ATJ_PARAM_VAL_RANGE(*vid*)

Description

This macro is used in the macro **ATJ_CMD_PARAM** to specify the validator to check that an input parameter value is within a certain integer range.

The integer range associated with the validator is specified elsewhere in the code using the macros **ATJ_DECLARE_VALIDATOR** and **ATJ_VALIDATOR_RANGE**.

Parameters

vid	Identifier of validator

ATJ_PARAM_VAL_FUNC

ATJ_PARAM_VAL_FUNC(*vid*)

Description

This macro is used in the macro **ATJ_CMD_PARAM** to specify the validator to check that an input parameter value is acceptable, where this validation is performed by a user-defined function.

The validation function is specified elsewhere in the code using the macros **ATJ_DECLARE_VALIDATOR** and **ATJ_VALIDATOR_FUNC**.

Parameters

vid	Identifier of validator

ATJ_PARAM_VAL_NONE

ATJ_PARAM_VAL_NONE()

Description

This macro is used in the macro **ATJ_CMD_PARAM** to specify the that no validator will be used to check the value of the parameter being declared - that is, the parameter value will not be validated.

Parameters

None

ATJ_DECLARE_VALIDATOR

ATJ_DECLARE_VALIDATOR(*vid*)

Description

This macro is used to declare the specified validator (used to check if an input parameter value is acceptable, e.g. within a certain range).

The macro can be used in conjunction with the macro **ATJ_VALIDATOR_RANGE** (which defines the integer range within which an acceptable value lies), as follows

```
ATJ_DECLARE_VALIDATOR(vid)=ATJ_VALIDATOR_RANGE(a,b);
```

or in conjunction with the macro **ATJ_VALIDATOR_FUNC** (which specifies the user-defined function to be used for validation), as follows

```
ATJ_DECLARE_VALIDATOR(vid)=ATJ_VALIDATOR_FUNC(f);
```

Parameters

vid	Identifier of validator to be declared

ATJ_VALIDATOR_RANGE

ATJ_VALIDATOR_RANGE(*a*, *b*)

Description

This macro is used to specify the integer range for a validator that checks whether an input parameter value lies within a certain range of values.

The macro is used in conjunction with the macro **ATJ_DECLARE_VALIDATOR**, as follows

```
ATJ_DECLARE_VALIDATOR(vid)=ATJ_VALIDATOR_RANGE(a,b);
```

where *vid* is the identifier of the validator for which the range is being defined.

Parameters

a	Lowest acceptable integer value for range test
b	Highest acceptable integer value for range test

ATJ_VALIDATOR_FUNC

ATJ_VALIDATOR_FUNC(*f*)

Description

This macro is used to specify the user-defined function to be used by a validator to check whether an input parameter value is acceptable. The use of a validation function allows a set of validation rules to be applied to the input value.

The macro is used in conjunction with the macro **ATJ_DECLARE_VALIDATOR**, as follows

```
ATJ_DECLARE_VALIDATOR(vid)=ATJ_VALIDATOR_FUNC(f);
```

where *vid* is the identifier of the validator for which the function is being specified.

The validation function must be defined in the code or in a separate header file.

Parameters

f	Name of user-defined function to be used for validation

4.2 Registering New Commands

This section describes the functions used to register a new command set with the AT-Jenie command parser.

Note: Before attempting to add a command set to the AT-Jenie parser, the commands must be defined using the macros described in Section 4.1.

The Jenie functions to add a custom command set and enable/disable an individual command are listed below, along with their page references:

vATJ_ParserAddCommands

```
void vATJ_ParserAddCommands(
                        tsATCommandSet *psCommandSet);
```

Description

This function adds the specified command set to the AT-Jenie command parser. The command set must have been defined using the macros described in Section 4.1.

Parameters

*psCommandSet	Pointer to structure which contains new command set

Returns

None

Example

To add the LCD command set from the example code fragment in Section 4.1, the required function call would be:

```
vATJ_ParserAddCommands(&asLCDCommands);
```

vATJ_ParserSetCommandEnable

```
void vATJ_ParserSetCommandEnable(
                    tsATJChannelDescriptor *psChannel,
                    char *CmdsStr,
                    bool_t bEnabled);
```

Description

This function can be used to enable or disable a custom command that has been added to the AT-Jenie command parser. The command must have been defined as part of a command set using the macros described in Section 4.1 and added to the parser using the function **vATJ_ParserAddCommands()**.

Parameters

***psChannel**	Pointer to the currently opened command parser channel. This is pre-defined as `UartChannel` and corresponds to the UART port used by the parser.
***CmdsStr**	Pointer to character string for the command
bEnabled	Set to one of: TRUE - enable command FALSE - disable command

Returns

None

Example

To enable the new command "NHL" (node highlight) on the parser channel 'UartChannel', the required function call would be:

```
vATJ_ParserSetCommandEnable(&UartChannel, "NHL", TRUE);
```

vATJ_TunnelSetCommandEnable

```
void vATJ_TunnelSetCommandEnable(char *CmdsStr,
                                 bool_t bEnabled);
```

Description

This function can be used to enable or disable a custom command for tunnelling - that is, so that the command can be tunnelled and executed on a remote node.

The command must have been defined as part of a command set using the macros described in Section 4.1, and registered using the functions **vATJ_ParserAddCommands()** and **vATJ_ParserSetCommandEnable()**.

Parameters

***CmdsStr**	Pointer to character string for the command
bEnabled	Set to one of: TRUE - enable tunnelling for command FALSE - disable tunnelling for command

Returns

None

Example

To enable the new command "NHL" (node highlight) for tunnelling, the required function call would be:

```
vATJ_TunnelSetCommandEnable("NHL", TRUE);
```

Appendices

The appendices contain all the ancillary information that you need in order to use the AT-Jenie commands. This information includes details of responses, hardware events (from the JN513x chip) and data transmission flags. In addition, mappings are provided between the AT-Jenie commands and Jenie API functions.

A. Responses and Events

This appendix describes the responses to AT-Jenie commands. The commands and their corresponding responses are listed in Chapter 3.

Two types of response are presented below:

- Appendix A.1 lists and describes the responses which are returned immediately after the command has been issued, simply to indicate whether or not the command was successfully issued. These responses are only produced if response generation was enabled when the AT-Jenie command parser was configured using the **CCF** command.
- Appendix A.2 lists and describes the "deferred responses" that are received some time after the command has been issued, once the command has been implemented. In the case of sending a message to a remote node, a deferred response only represents a true reply from the remote node if the message was sent (using the command **SND** or **SDS**) with acknowledgements enabled - if acknowledgements are disabled, the deferred response is generated locally.

Note: Appendix A.2 also includes network events that may occur at any time.

A.1 Immediate Responses

The immediate responses are listed and described in the table below.

Response	Description	Parameters
Success		
OK	Standard response for success	None
OKA	OK with associate parameter	**State** 0: Disabled 1: Enabled
OKP	OK with peripheral parameter	**Value** For details of this value, see the individual command descriptions in Chapter 2 and Chapter 3.
OKV	OK with version parameter	**Version** 8-digit hex number (see Table 21 below)
OKO	OK with multiple output parameters	Comma separated list of values
Failure		
ERR	Standard response for error	None
Busy		
BSY	An opened tunnel is in use	None

Table 20: AT-Jenie Responses

Component	Bits	Description
AT-Jenie interface	31-0	AT-Jenie version number
JenNet software	31-16	Network stack protocol (JenNet) revision
	15-0	Network stack software revision
IEEE 802.15.4 software	31-24	Non-zero value identifying special or custom build
	23-16	Really major revision
	15-8	Minor (patch) revision
	7-0	Major revision (only changes with new ROM version)
JN513x chip	31-28	Revision number: 0x0 for R0, 0x1 for R1, etc
	27-22	Metal mask version ID
	21-12	Jennic part number: 0x000 for JN5121, 0x002 for JN5139
	11-0	Manufacturer's identification

Table 21: OKV Version Numbers

A.2 Deferred Responses and Network Events

The deferred responses are listed and described in the table below, which also includes network events that can occur at any time.

Response	Description	Parameters
Network Formation		
NTU	Network has started or been joined successfully. This response is received on the joining node.	**Parent address** 64-bit IEEE/MAC address of joining node's parent **Node address** 64-bit IEEE/MAC address of joining node **Depth in network** Depth of node in network (0 for Co-ordinator) **PAN ID** Assigned PAN ID (32-bit) **Channel** Assigned radio channel (11-26)
CHJ	A node has joined this Co-ordinator or Router. This response is received on the parent node.	**Address of joining node** 64-bit IEEE/MAC address
CHL	A child node has left this Co-ordinator or Router.	**Address of lost child node** 64-bit IEEE/MAC address
CHR	A child node has been rejected by this Co-ordinator or Router	**Address of rejected child node** 64-bit IEEE/MAC address
RST	A stack reset has occurred	None
Service Discovery and Binding		
RSR	Register Service Response - received in response to **REG** command once the specified list of services has been registered	None
SRR	Service Request Response - received in response to **RQS** command. Sent by remote node to indicate which of requested services it supports.	**Address of remote node** 64-bit IEEE/MAC address **Services** 32-bit value: bit 0 set for Service 1, bit 1 set for Service 2, etc.
Sending and Receiving Data		
PKS	Data packet successfully sent - received in response to **SND** or **SDS** command.	None
PKF	Data packet not successfully sent - received in response to **SND** or **SDS** command.	None

Table 22: AT-Jenie Deferred Responses and Network Events

DAT	Indicates that a message sent from another node (using the **SND** command) has been received	**Address of source node** 64-bit IEEE/MAC address **Flags** Unused (reserved) **Payload length** Number of bytes in payload **Payload** Payload data string (for format, see Table 1 on page 12)
DTS	Indicates that a message sent from another node (using the **SDS** command) to a particular service available on this node has been received	**Address of source node** 64-bit IEEE/MAC address **Source service** Service ID, value in range 1-32 **Destination service** Service ID, value in range 1-32 **Flags** Unused (reserved) **Payload length** Number of bytes in payload **Payload** Payload data string (for format, see Table 1 on page 12)
ACK	Indicates that an end-to-end acknowledgement has been received following data transmission to a remote node using the command **SND** (acknowledgements must be enabled in **SND**).	None
SAK	Indicates that an end-to-end acknowledgement has been received following data transmission to a service on a remote node using the command **SDS** (acknowledgements must be enabled in **SDS**).	None
PLC	Indicates that polling of the parent for data has completed and gives the outcome.	**Poll status** 0: Poll complete, OK 1: Unused (reserved) 2: Poll complete, timeout
Tunnelling		
TNR	Indicates that a tunnelled command or response has been received from another node	**Payload** ASCII string containing response delimited at both ends by quotes (“)

Table 22: AT-Jenie Deferred Responses and Network Events

B. Hardware Events

This appendix describes the hardware events that can be received from the on-chip peripherals of the JN513x wireless microcontroller. The response which contains the hardware event is detailed in the table below.

Response	Description	Parameters
PEV	Event (usually interrupt) from peripheral on JN513x wireless microcontroller	**Source Peripheral** 0: Analogue peripheral (ADC or DAC) 1: Comparator 2: DIO 3: Wake timer 4: Timer **Event bitmap** 32-bit bitmap in which each bit has a meaning dependent on the event source (see tables below) **Analogue value** Value from ADC (no values from other sources)

Table 23: Hardware Events from JN513x Integrated Peripherals

The bit representations in the 32-bit event bitmap depend on the peripheral that is the source of the event. Bit descriptions for each possible source peripheral are provided in the tables below.

Analogue Peripherals (ADCs and DAC)

Bit	Description
0	Asserted to indicate capture complete or new sample ready
1-31	Not used

Comparators

Bit	Description
0-28	Not used
29	Asserted to indicate comparator event (transition)
30-31	Not used

Digital I/Os (DIOs)

Bit	Description
0-20	Asserted to indicate event from corresponding DIO (bit 0 for DIO0, bit 1 for DIO1, etc)
21-31	Not used

Wake Timers

Bit	Description
0-25	Not used
26	Asserted to indicate event from Wake Timer 0
27	Asserted to indicate event from Wake Timer 1
28-31	Not used

Timers

Bit	Description
0	Asserted when timer has completed a period (on high-to-low transition). Can only be asserted if interrupts have been enabled for ‘timer end’ using the **PTE** command.
1	Asserted when the timer finishes a low period (on low-to-high transition). Can only be asserted if interrupts have been enabled for ‘output rising’ using the **PTE** command.
2-31	Not used

C. Data Transmission Flags

The data transmission commands **SND** and **SDS** (described in Section 2.4) use a set of flags to enable/disable the following features, in any combination:

- **Silent:** Message to be sent with deferred responses PKS and PKF disabled
- **Broadcast:** Message to be sent to all Router nodes in the network
- **Security:** Message content to be encrypted using security key (set with **KEY**)
- **Acknowledgement:** Confirmation of receipt required from target node

These flags are specified by a single value, as detailed in the table below.

Flags Value	Features			
	Silent	Broadcast[1]	Security[2]	Acknowledgement[3]
0	No	No	No	No
1	No	No	No	Yes
2	No	No	Yes	No
3	No	No	Yes	Yes
4	No	Yes	No	No
5	No	Yes	No	Yes
6	No	Yes	Yes	No
7	No	Yes	Yes	Yes
8	Yes	No	No	No
9	Yes	No	No	Yes
10	Yes	No	Yes	No
11	Yes	No	Yes	Yes
12	Yes	Yes	No	No
13	Yes	Yes	No	Yes
14	Yes	Yes	Yes	No
15	Yes	Yes	Yes	Yes

Table 24: Flags Settings for Send Commands

[1] To enable a broadcast, you must also set the target address to 0 in the **SND** command

[2] Security flag is ignored in this AT-Jenie release - to enable security, use the **KEY** command

[3] Acknowledgement will be received as a deferred response - an **ACK** for **SND**, a **SAK** for **SDS**

D. AT-Jenie to Jenie API Mappings

This appendix lists the AT-Jenie serial commands along with their corresponding C functions from the Jenie API (Application Programming Interface).

Note: In order to use AT-Jenie, no knowledge of the Jenie API functions is required. However, if you would like further information on these functions, refer to the *Jenie API Reference Manual (JN-RM-2035)*, available from the Support area of the Jennic web site.

Note: Some AT-Jenie commands are related to functions of the Board API, described in the *Board API Reference Manual (JN-RM-2003)*.

AT-Jenie Command	Jenie API Function
BGF	None (related to Board API functions)
BGH	None (related to Board API functions)
BGL	None (related to Board API functions)
BGR	None (related to Board API functions)
BGT	None (related to Board API functions)
BGV	None (related to Board API functions)
BLF	None (related to Board API functions)
BLO	None (related to Board API functions)
BND	**eJenie_BindService()**
BTX	None (related to Board API functions)
CCF	None (AT-Jenie only)
CCS	None (AT-Jenie only)
CFG	None (parameters set as Jenie global variables)
CFP	None (parameters set as Jenie global variables)
GAS	**bJenie_GetPermitJoin()**
GTV	**u32Jenie_GetVersion()**
INI	**vJenie_CbConfigureNetwork()** callback
KEY	**eJenie_SetSecurityKey()**
LVE	**eJenie_Leave()**

Table 25: AT-Jenie Command to Jenie API Function Mappings

AT-Jenie Command	Jenie API Function
OAD	None (AT-Jenie only)
PAC	**vJPI_AnalogueConfigure()**
PAD	**vJPI_AnalogueDisable()**
PAE	**vJPI_AnalogueEnable()**
PAO	**vJPI_AnalogueDacOutput()**
PAR	**u16JPI_AnalogueAdcRead()**
PAS	**vJPI_AnalogueAdcStartSample()**
PCD	**vJPI_ComparatorDisable()**
PCE	**vJPI_ComparatorEnable()**
PCI	**vJPI_ComparatorIntEnable()**
PCS	**bJPI_ComparatorStatus()**
PCW	**bJPI_ComparatorWakeStatus()**
PDD	**vJPI_DioSetDirection()**
PDE	**vJPI_DioWake()**
PDO	**vJPI_DioSetOutput()**
PDP	**vJPI_DioSetPullup()**
PDR	**u32JPI_DioReadInput()**
PDS	**u32JPI_DioWakeStatus()**
POL	**eJenie_PollParent()**
PTC	**vJPI_TimerStartCapture()**
PTD	**vJPI_TimerDisable()**
PTE	**vJPI_TimerEnable()**
PTF	**u8JPI_TimerFired()**
PTG	**vJPI_TimerStart()**
PTR	**vJPI_TimerReadCapture()**
PTX	**vJPI_TimerStop()**
PWC	**u32JPI_WakeTimerCalibrate()**
PWE	**vJPI_WakeTimerEnable()**
PWF	**u8JPI_WakeTimerFiredStatus()**
PWG	**vJPI_WakeTimerStart()**
PWS	**u8JPI_WakeTimerStatus()**
PWX	**vJPI_WakeTimerStop()**

Table 25: AT-Jenie Command to Jenie API Function Mappings

AT-Jenie Command	Jenie API Function
RDP	eJenie_RadioPower()
REG	e Jenie_RegisterServices()
RQS	eJenie_RequestServices()
RST	vJPI_SwReset()
SAS	eJenie_SetPermitJoin()
SCN	eJPDM_SaveContext()
SDS	eJenie_SendDataToBoundService()
SLP	eJenie_Sleep()
SND	eJenie_SendData()
SSP	eJenie_SetSleepPeriod()
STR	eJenie_Start()
TCL	None (AT-Jenie only)
TCM	None (AT-Jenie only)
TCN	None (AT-Jenie only)
TOP	None (AT-Jenie only)
UBN	eJenie_UnBindService()

Table 25: AT-Jenie Command to Jenie API Function Mappings

Revision History

Version	Date	Comments
1.0	28-Nov-2007	First release
1.1	20-Feb-2008	References to ZigBee removed
1.2	06-Mar-2008	Updated for Jenie v1.2
1.3	01-Apr-2008	Minor updates
1.4	10-July-2008	Updated for Jenie v1.3. Commands CFG, CFP, INI and SLP updated. Response CHR added
1.5	25-Sep-2008	Maximum data payload sizes added for commands SND and SDS
1.6	04-Dec-2008	Updated for Jenie v1.4: • Updated CFP command • Added functions vATJ_ParserSetCommandEnable() and vATJ_TunnelSetCommandEnable() for custom commands • Added mappings between AT-Jenie commands and Jenie API functions
1.7	27-Aug-2009	Various minor updates made

Important Notice

Jennic reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and services at any time, and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders, and should verify that such information is current and complete. All products are sold subject to Jennic's terms and conditions of sale, supplied at the time of order acknowledgment. Information relating to device applications, and the like, is intended as suggestion only and may be superseded by updates. It is the customer's responsibility to ensure that their application meets their own specifications. Jennic makes no representation and gives no warranty relating to advice, support or customer product design.

Jennic assumes no responsibility or liability for the use of any of its products, conveys no license or title under any patent, copyright or mask work rights to these products, and makes no representations or warranties that these products are free from patent, copyright or mask work infringement, unless otherwise specified.

Jennic products are not intended for use in life support systems/appliances or any systems where product malfunction can reasonably be expected to result in personal injury, death, severe property damage or environmental damage. Jennic customers using or selling Jennic products for use in such applications do so at their own risk and agree to fully indemnify Jennic for any damages resulting from such use.

Jennic Ltd
Furnival Street
Sheffield
S1 4QT
United Kingdom

Tel: +44 (0)114 281 2655
Fax: +44 (0)114 281 2951
E-mail: info@jennic.com

For the contact details of your local Jennic office or distributor, refer to the Jennic web site: