Zensys® ASCII interface

1. Message format

1.1. Message start: **“>”**

1.2. Message End: **“<cr>” or ”<cr><lf>”**

1.3. Input string is limited to 80 characters

2. Commands used

All commands are 1 or 2 characters. Below is a list of commands used with the serial interface:

ID	Description
"N"	Command starts association. Specifies the list of nodes IDs separated by **“,”**
"AP”	Command used when a node’s list has more than 80 characters
"GS"	Command stores the current group, represented by association list. Parameter specifies the group number
“GR”	Command recalls previously stored group. Parameter specifies the group number
"ON"	Command turns ON single light or group of lights defined by current association
"OF"	Command turns OFF single light or group of lights defined by current association
"L"	Command sets level for single light or group of lights defined by current association. Parameter (0-99) specifies the level according to Zensys® protocol
"S"	Command calls scene for single light or group of lights defined by current association. Parameters specify the following information 1: 1-255 specifies scene number, 2: specifies the fade rate (255 is default)
"PS"	Command programs scene for single light or group of lights defined by current association. Parameters specify the following information 1: 1-255 specifies scene number, 2: specifies the fade rate (255 is default)
"DI"	Command decreases light for the single light or group of lights defined by current association
"BR"	Command increases light for the single light or group of lights defined by current association
"ST"	Command stops the light from changing level
“UP”	Command starts to update the loop after implementing one of the commands. This is necessary to keep all controllers in the Leviton’s Z-Wave® system synced to show correct display
"SE"	Command sends any Z-Wave message. The first parameter is the command class, the second is the command and the rest of the command is the parameters of the appropriate command class. Refer to the Zensys command class specification for the information on the command classes available for Z-Wave system
"DE"	Command resets the ASCII interface to default. If button on the unit is held while the command is sent the unit will reset to the factory default state
"IN"	Command switches module from the add/remove mode to add or remove serial interface to the Z-Wave network
"AB"	Command Aborts current transmission

"SU"	Command enables node to be a SUC/SIS is parameter ==1 or disable if parameter==0
“NN”	Command asks for node name
“NL”	Command asks node location
“FI”	Command finds node ID with assigned properties
“RO”	Command assigns and delete Routes for the routing slaves

Command may include more than 2 characters if so only the first two characters are used by the serial device but you can use all letters in ASCII string. For example, “OFF” and “OF” will be interpreted the same way. The same goes for “ST” and “STOP”.

2.1. Command Descriptions

2.1.1. (N)Start Association

Nxxx, xxx, xxx…xxx, where xxx are decimal numbers representing nodes ID on a Z-wave network. To remove any association use **“N”** command without parameters (use **“,”** if necessary to separate from the next command).

>N1, 10, 23 **-** Creates group for including 3 nodes with IDs of 1, 10 and 23
>N, **-** Creates empty group, no nodes are included to this group.

Question mark **“?”** can be used before “N” to request (BASIC GET) levels from the node

>? N1, 10, 23 **-** Request node levels from nodes 1, 10 and 23

2.1.2 (AP) Append to Association

APxxx, xxx, xxx…xxx **-** Adds nodes to the previously existing association list
? APxxx, xxx **-** Will request node levels from group

2.1.3 (GS) Group Store

GS command **-** Stores in nonvolatile memory the current association list
> N1, 10,23GS1 **-** Store group of 3 nodes 1, 10, 23 as a group 1.

2.1.4 (GR) Group Recall

GR command – recalls from nonvolatile memory a specified group and sets it as the current association list.
>GR1ON – Turn lights on in group 1
>?GR1 - Requests level for nodes in group 1.

2.1.5 [(ON) (OF) (L) (S) (PS) (DI) (BR) (ST)] Light Control Commands

Light control commands apply to the nodes in the current association list. If current association list is empty a broadcast message will be sent. If association list includes just one node then a single cast message will be used. If there are multiple nodes on the association list, multicast message will be used.

All commands are using BASIC CLASS command. The abbreviations are self explanatory.

Examples:

Commands bellow apply to the nodes 1, 10 and 23

```
>N1, 10,23ON- Turn nodes ON
>N1, 10,23OFF- Turn nodes OFF
>N1, 10,23L50- Set nodes to 50% output
>N1, 10, 23S2, 255 - Calls scene 2 with default fade rate for nodes
>N1, 10, 23PS2, 255   Programs current levels as a scene 2 with default fade rate for nodes
```

>N, ON - Broadcast ON command (all ON). Broadcast commands don’t have an acknowledgment so it is not guaranteed that all nodes will receive this message.
Light level, current scene parameters and specified scene number parameters can be asked from nodes supporting appropriate classes.

>?Nxxx - Will request the node level/state. The response comes in form <NxxxLyyy.
>Nxxx?Syyy - Will request the scene information from the node. If yyy=0 current scene information will be returned.
The information is returned as <NxxxS###, lll, fff, where ### scene number, lll light level for the scene and fff fade rate programmed for the scene.

2.1.6. (UP) Update command

After any command, changing the level or state of the load, the sending node is responsible for updating information to all of the other controllers controlling the same nodes. So to keep the system up-to-date an update command should be appended to all “action” commands.

```
>N1,10,13ON
>UP
```

We would recommend a delay of a few seconds between the end of command implementation and starting of the update.

2.1.7. (SE)Send Command

To provide control to nodes different from lighting control nodes the send command can be used. This command sends over RF any Z-wave command. The nodes message will be send as before defined by any of the association of commands:
N, AP, GR.

N5SE69,2 - Will request thermostat fan state from thermostat node 5.

2.1.8. [(NN) (NL)] Node name and location commands

NN (NL) provides access to programming/recalling node name (node location name).

```
>NxxxNNdyyy…yyy
>NxxxNLdyyy… yyy
```

where xxx- node ID
yyy...yyy ASCII string up to 16 characters
where d specifies character set: 0=Standard ASCII,
1= Extended ASCII and 2=Unicode UTFIG
We recommend using d=1

To request the previously programmed name use >?NN or >?NL The name returns in format:
<Nxxx”yyyyyyy

2.1.9. (FI) Find Node ID With Requested Basic, Generic and Specific Properties.

This command allows finding network node ID for the Z-wave node with certain basic, generic and specific classes. It maybe used during installation procedure as well as to check if node is in the routing table.

```
>[?]Fibbb,ggg[,sss[,iii]]
```

where

bbb – basic class number
ggg – generic class number
sss - specific class number
iii – instance number for the device (between 1 and 232).

If any of the class numbers (bbb,ggg,sss) is 0, the command will search for any device. The instance number (1-232) will reference on the instance in the routing table for certain devices. Basic class and generic class are mandatory fields, specific and instance fields are not. If specific fields are missing it assumed 0. If instance class is missing it assumed 1 (the first instance).

Zensys specifies only 4 valid basic classes:

Controller	-1
Static controller	-2
Slave	-3
Routing Slave	-4

For generic and specific classes refer to the Zensys device class specification. For example switch device will have ggg=16, dimmer ggg=17, thermostat ggg=8.

After searching, the device found will be added to the current group. You can store the current group after or send any message using this group.

If '?' is set before the command interface, it will report back as Node ID:
<Fxxx - where xxx is a node ID for the device.

If no devices is found to satisfy the conditions the error message will be returned as:
<E010

2.1.10. (RO) Assign Routes for the Routing Slaves

To provide routing slaves with valid routes (when nodes can't reach each other directly) "RO" command is used.
Routing slave can have up to 5 nodes assign return routes.
Since every time routes are added to the existing list we recommend deleting all routes first.
>ROxxx,0 or >ROxxx

Above command will delete all routes for the routing slave with xxx node ID.
>ROxxx,yyy

Above command will setup up to 4 valid routes for node xxx to node yyy.

2.2 Serial Controller Replays

Serial controller replays every command sent over serial port. It also replays after finishing every RF transmission and processes adding/removing itself to the Z-wave network. When serial interface receives messages it addresses them to forward them to the serial output, except when the messages are processed by the interface.

2.2.1. Receive Command Replays

After receiving command over serial port, the serial interface checks and processes. If the command was correct and modules have enough recourse currently to implement, the interface responds with : <E000. – No error message.

If an error has been detected or some of the resources are unavailable the node responds with <Exxx. , xxx is the error number. Errors are listed bellow:

0 -No error
1 - Wrong start of the string symbol
2 - Input buffer is over flowed.
3 - Can't start RF transmission. All buffers are taken.
4 - Can't start RF transmission because previous transmission did not finished.
5 – Wrong, unrecognized command
6 - Attempt to send the new buffer over RS232 before previous one processed.
7 - Send Message doesn't have data fields specified.
8 - Can't stop SUC mode. Node is SUC.
9 - EEPROM is busy, can't store group information
10 - No devices with specified properties been found.

The new input string can be accepted only after response <Exxx was received as a result of processing the previous one. Attempt to send the new one before previous attempt has been processed, results in an Error#6 generated.

2.2.2 Replays after implementing RF transmission commands

After implementing the transmission request, the serial interface will replay with the string: <Xyyy, where yyy is the error code.

yyy=0, if transmission was successful or yyy=2, if an error was detected during transmission

After receiving <Xyyy it will replay the next command and the required transmission can be sent.

2.2.3 Add/Remove Node Replays

While going through the stages of add/remove to the Z-wave network the interface will replay with strings:

<!xxx, where xxx=16 if Learning mode has started and xxx=32 if this mode is unsuccessful and xxx=0 if successful.

2.2.4 Received Message Strings

If BASIC REPORT, MULTILEVEL_SWITCH or BINARY_SWITCH reports are received the module transfers it to the serial port as

<NxxxLyyy where xxx is a node ID and yyy is a light level between 1and 100.

0 indicates off and 255 indicates On.

The scene information reported back as

<NxxxS###,lll,fff where ### is the scene number, lll light level, fff fade rate.

Scene number 0 indicates that the node is not at any scene level.

Node name information displays as:
<Nxxx”yyy...yyy for the node name and <Nxxx’yyy…yyy for node location name.

All other general messages are displayed in general format:
<Nxxx:yyy,yyy…
where xxx is a node ID and yyy decimal command data.
The first byte represents command class, the second byte represents commands and the rest of the bytes represent the command data. Refer to the Zensys Command class specification for information regarding the data structure for the different commands.

2.2.5 Found Node ID Reply

If a node with specified properties has been found in the routing table and ‘?’ is in front of Request, the node ID returns the following message:

<Fxxx   where xxx is the node ID for the node that has been found.

3.0 UART Settings

9600-8-N-1

APPENDIX

A. Multilevel Sensor

Below is an example of getting information from a Multilevel Sensor (i.e. a temperature sensor used by thermostats). Assumed that sensor node ID is 4.

```
>N4SE49,4                      <- Send multilevel sensor GET
<E000                          <- RS interface processed the input
<X000                          <- Appropriate message has been send correctly
<N004:049,005,001,009,075      <- REPORT from the sensor:
```

049 – Command class
005 – Command report
001 – The value send is temperature
009 – The value represented by 1 byte and in deg. F
075 – Temperature 75F

B. Control Thermostat Fan

Assume that thermostat node is 4.

```
>N4SE68,1,1                         <- Set thermostat fan mode 1: Will turn the manual fan
                                      operation on.
<E000                                <- RS interface processed the input
<X000                                <- Appropriate message has been send correctly
>SE68,2                              <- Request thermostat fan mode (current group is still
                                     just node 4, you don't need to repeat N4 command.
<E000                                <- RS interface processed the input
<X000                                <- Appropriate message has been send correctly
<N004:068,003,001                    <- Thermostat report indicate mode 1
>SE68,1,0                            <- Set thermostat fan mode 0: Will turn the manual fan
                                     operation off .
<E000                                <- RS interface processed the input
<X000                                <- Appropriate message has been send correctly
>SE68,2                              <- Request thermostat fan mode
<E000                                <- RS interface processed the input
<X000                                <- Appropriate message has been send correctly
<N004:068,003,000                    <- Thermostat report indicate mode 1
```

C. Set Thermostat Set Point

Assume again thermostat is node 4.

```
>N4SE67,1,2,9,80              <- Set thermostat cooling set point to 80F
<E000
<X000
>N4SE67,2,2                     <- Request thermostat set point information
<E000
<X000
<N004:067,003,002,009,080    <- Get back thermostat report with set point 80F.
```

D. Dimmer control

Assume dimmer node is node 8.

```
>?N8                  <- Request dimmer level
<E000                 <- Request processed
<X000                 <- Request has been sent
<N008L100             <- Received dimmer’s response, current level 100%

>L50UP                <- Set light level at 50% and request update
<E000                  <- Request processed
<X000                  <- Request has been sent
<N008L050              <- Node 8 sends back it current level as result of update. Nothing will be
                          sent back if update wasn’t requested.
```

Turn dimmer OFF sequence.

```
>N8OFF,UP
<E000
<X000
<N008L000
```

E. Updates With Few Nodes

```
>?N8,4                          <- Request state of 2 nodes (8 –dimmer, 4- thermostat)
<E000                           <- Request processed
<N008L050                       <- Response from dimmer
<N004:064,003,003               <- Response from thermostat
<X000                           <- Requested RF transmission (GET) is finished
<N008L050                       <- Response from thermostat
>N8,4UP                         <- Instead of requesting response use update
<E000                           <- Request processed
<N004:064,003,003               <- Response from thermostat
<N008L050                       <- Response from dimmer
```

When ‘?’ has been used as BASIC GET it has been sent as a multicast. When responses have been sent for the first time , the broadcast portion of the multicast message has been received. The response shows up before the full multicast message has finished. Then the dimmer responds to the single cast part of the message one more time.

When using update the system regulates sending BASIC GET messages to the devices in order to avoid unnecessary collisions. This is the preferred way instead of sending direct BASIC GET as in the first case.

F. <u>If node ID is unknown the find node ID command can be used</u>.

```
>N,                                    <- Empty current group
>FI0,8 SE67,1,2,9,80                   <- Set the first thermostat cooling set point to 80F
<E000
<X000
>GS100                                  <- Store the thermostat as a group 100 (to recall this node
                                      instead of searching every time)
```

Find out what thermostat node ID:

```
>?FI0,8
<F004
<E000
```

Interface reported that the thermostat has node ID=4.

G. <u>Setup Association Tables for Node</u>

To setup an association table for the node use association class.
SE133,1, Group#, node1,node2…

Where Group# is 1 for the dimmers/switches (since they support just one group) and 1 through 4 for 4 button controllers based on what button you are programming.

node1, node2 are nodes ADDED to the node list for the dimmer or controller’s button

To remove node use SE133,4, Group#, node1, node2…

For example:

```
>N2SE133,1,1,7
<E000
<X000
```

The above will add node 7 to the association table for the dimmer