Zip

IP Paging Solutions

3rd Party Control API

ATTEROTECH

Zip4 Control Overview

The Zip 4 paging station utilizes UDP (Universal Datagram Protocol) for station control and monitoring.
- All control commands should be sent to the IP address of the Zip4 on port 49494.
- Any response is returned to the IP and port that the initial request was sent from.

Note: Although this document refers to the ZIP4, these commands also apply to the Zip4-3G.

Write Command Format

Write commands consists of a one byte command word and a one byte length field followed by the data to be written. The length field includes not just the length of the data but also includes the command and length fields too, therefore the minimum length of a write command is 2.

`<Command><Length><Data>`

Field Name	Description
<Command>	Single byte - The command that will be given to the device (See command table for details)
<Length>	Single byte – The length of the message including the command and length fields
<Data>	Specific data to be written based on command being sent

Write commands are not acknowledged by the Zip4 so there will be no response.

Example SET_STATUS "write" type command (values indicated are the byte values in the actual UDP message payload):
`Send: 01 06 00 00 00 00`   Command = 0x001, Length = 0x06, Data = 0x00 0x00 0x00 0x00

Read Command Format

Read commands consist of a one byte command and a one byte length field. The length includes the command and length fields, therefore the minimum length of a command is 2.

`<Command><Length>`

Field Name	Description
<Command>	Single byte - The command that will be given to the device (See command table for details)
<Length>	Single byte – The length of the message including the command and length fields

Message responses are sent back to the source IP and port of the message that requested the data. They follow the same format as the requests using the same command value. Length value and data are dependent on the specific command.

Field Name	Description
<Command>	Single byte - The command that will be given to the device (See command table for details)
<Length>	Single byte – The length of the message including the command and length fields
<Data>	Optional data bytes required based on command being sent

Note: The following example uses data intended to indicate the format of the message. The device will return different values..

Example GET_VERSION "read" type command (values indicated are the byte values in the actual UDP message payload):

`Send: 08 02          Command = 0x08, Length = 0x02`

`Receive: 08 0a 01 02 03 04 05 06 07 08`
Command = 0x08, Length = 0x0a, App major = 0x0102, App minor = 0x0304, Bootloader major = 0x0506, .Bootloader minor = 0x0708

Status Indicators

The LED status indicators of the device are controlled by sending a status message. The status itself is a single 32-bit bit word where each bit indicates the state of the all the indicators along with a couple of status parameters on the Zip4. Each bit of the status word represents a specific control or function. The table below show status word definitions.

Zone 1 Green LED	0x00000001	**Zone 1 Red LED**	0x00000100	**Remote Reset**	0x00010000
Zone 2 Green LED	0x00000002	**Zone 2 Red LED**	0x00000200	**Remote WD**	0x00020000
Zone 3 Green LED	0x00000004	**Zone 3 Red LED**	0x00000400	**Mic LED**	0x00400000
Zone 4 Green LED	0x00000008	**Zone 4 Red LED**	0x00000800		
Ready Green LED	0x00000010	**Ready Red LED**	0x00001000		
Busy Green LED	0x00000020	**Busy Red LED**	0x00002000		

The "Remote Reset" and "Remote WD" are special cases. Writing a "1" to the "Remote Reset" bit will cause the ZIP4 to reboot.

The status word can be read from the device using the GET_STATUS command or written to the device using the SET_STATUS command. Using the SET_STATUS command configures the state of the Zip4's various indicators based on the settings with the included status word. The formats for both the GET_STATUAS and SET_STATUS commands are shown below.

Command Type	Command Byte	Length Byte	Data
SET_STATUS	0x01	0x06	Sets status of device such as the state of the indicator LED's. Data included is 32 bit status word (4 bytes) as defined above. The bytes are sent in network byte order.
GET_STATUS	0x02	0x02	This has no additional input data. Response of 6 bytes (command, length and 32 bit status word). The bytes of the 32-bit word will be sent in network byte order.

*Note: The following example uses data intended to indicate the format of the message and is not necessarily valid or usable.

Example SET_STATUS message (values indicated are the byte values in the actual UDP message payload):

`Send: 01 06 12 34 56 78`   Command = 0x01, Length = 0x06, Status = 0x12345678

`Receive:` N/A - No received message for write command

Example GET_STATUS message (values indicated are the byte values in the actual UDP message payload):

`Send: 02 02`   Command = 0x02, Length = 0x02

`Receive: 02 06 12 34 56 78`   Command = 0x08, Length = 0x0A, Status = 0x12345678

Event Messages

Each time the user presses or releases a button on the Zip4, it will send an asynchronous event message to the control system. This message indicates the device involved, which button has changed state and the new state of the button. These asynchronous messages are sent to the destination IP address and port of the control server.

By default, the default port is 6326 and the default IP is a broadcast address based on the devices current IP address and subnet mask. For example, a station IP address of 10.10.0.1 and a subnet of 255.255.255.0 yields a default event destination address of 10.10.0.255. The event message destination IP address and port can re-configured to a more appropriate value for the system. Below are the command definitions for reading and writing the asynchronous message destination information.

Command Type	Command Byte	Length Byte	Data
SET_ASYNC_RESPONSE	0x03	0x08	Sets the IP and port address where asynchronous responses will be sent. The data sent is first IP address as a 32-bit integer and then the port as a 16-bit integer. Each number is sent in network byte order.
GET_ASYNC_RESPONSE	0x04	0x02	No additional input data. Reads back the asynchronous messages settings. Response of 8 bytes (command, length and 32-bit integer for the IP address and a 16-bit integer for the port address). The IP and port addresses are where asynchronous responses will be sent. Each value is sent in network byte order.

Example SET_AYSNC_RESPONSE message to set up event messages to sent to IP address 169.254.1.1 and port 49494 (values indicated are the byte values in the actual UDP message payload):

`Send: 03 08 a9 fe 01 01 c1 56`   Command = 0x03, Length = 0x08, Data = 0xa9 0xfe 0x01 0x01 0xc1 0x56

`Receive:` N/A - No received message for write command

Example GET_AYSNC_RESPONSE message (values indicated are the byte values in the actual UDP message payload):

`Send: 04 02`   Command = 0x04, Length = 0x02

`Receive: 04 08 a9 fe 01 01 c1 56`   Command = 0x08, Length = 0x0A, IP Address = 169.254.1.1, Port = 49494

Event Message Formats

Each event message consists of the following format.

<IP address><Button ID><State>

Field Name	Description
<IP Address>	Four bytes - The IP address of the source device in network byte order
<Button ID>	Two Bytes – ID of button that has changed state. Bytes are in network order. Button ID's are defined as follows: Zone 1 – 0x0001 Zone 2 – 0x0002 Zone 3 – 0x0003 Zone 4 – 0x0004 PTT Key – 0x0005
<State>	Two Bytes – New state of button. Bytes are in network order. States are defined as follows Released – 0x0000 Pressed – 0x0001

Example received event message (this will be sent to the IP address and port configured):

```
Receive: A9 FE EF E9 00 03 00 01     IP address = 169.254.239.233, Button = 0x003, State = 0x0001 (Zone button 3 active)
```

When an event message is sent asynchronously by the device, the device sets the "Remote WD" bit in the status word and it activates an internal watchdog. If the control system does not acknowledge the receipt of the event message within 8 seconds, the Zip4 enters an error state where all the indicator LED's flash red in unison for a short period to warn the user of a communication issue.

To prevent this happening, the control system ***MUST*** acknowledge the event message within the 8 second time limit by sending a SET_STATUS message with the "Remote WD" bit set back to 0. As well as acknowledging the event message, the status returned in the sent will update the device status bits.

System Parameters (Read Only)

The following commands retrieve system parameters that are read only. There is no write commands to set these values.

Command Type	Command Byte	Length Byte	Data
GET_VERSION	0x08	0x02	No additional input data. Returns 10 bytes (command, length, app major version, app minor version, boot loader major version, bootloader minor version). Each of the four version numbers are two bytes in network byte order.
GET_VENDOR_ID	0x09	0x02	No additional input data. Returns 10 bytes (command, length and 8 bytes of Vendor ID data).
GET_PRODUCT_ID	0x0a	0x02	No additional input data. Returns 18 bytes (command, length, and 16 bytes of Product ID data).
GET_MAC_ADDRESS	0x0b	0x02	No additional input data. Returns 8 bytes (command, length, 6 bytes of Mac address)
GET_TX_CHANNEL_LABEL	0x0c	0x02	No additional input data. Returns 34 bytes (command, length and 32 bytes of channel label data). The channel label data will be null padded if the name is not 32 bytes.
GET_DEVICE_NAME	0x0d	0x02	No additional input data. Returns 34 bytes (command, length, and 32 bytes of ASCII device name data). The device name data will be padded with nulls if the names is not 32 bytes

Example GET_DEVICE_NAME message (values indicated are the byte values in the actual UDP message payload):

```
Send: 0d 02                    Command = 0x02, Length = 0x02
```

```
Receive: 0d 22 5a 69 70 34 2d 39 30 65 39 65 65 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```
Command = 0x0d, Length = 0x22, Device Name= "Zip4-90e9ee"

Device Configuration Commands

The following commands deal with configurable aspects of the device itself such indicator LED and backlight brightness. These parameters can be read and written. The Mic LED commands refer to the LED signals located on the side-mounted terminal block connector of the Zip4. For the Zip4-3G, this output drives the LED halo around the PTT button.

Command Type	Command Byte	Length Byte	Data
SET_BACKLIGHT_INTENSITY	0x0e	0x04	Data contains two bytes. The first byte is set to 1 or 0, which turns the backlight LED on or off. The second byte sets the backlight intensity from 0 to 255 with 0 being no output and 255 being maximum intensity.
GET_BACKLIGHT_INTENSITY	0x0f	0x02	No additional input data. Returns 4 bytes (command, length, and 2 bytes of backlight data). The data format matches that of the SET_BACKLIGHT_INTENSITY command.
SET_RED_GRN_INTENSITY	0x10	0x04	Sets intensity of red and green indicator LED's. Data is two bytes of data. The first byte sets the intensity of the red LEDs while the second byte sets the intensity of the green LEDs. Both bytes take a value of 0 to 255, with 0 being no output and 255 being maximum intensity.
GET_RED_GRN_INTENSITY	0x11	0x02	No additional input data. Response of 4 bytes (command, length, one byte of Red LED intensity, and one byte of Green LED intensity). The data format matches that of the SET_RED_GRN_INTENSITY command.
SET_MIC_LED_INTENSITY	0x12	0x05	Data contains a 3 bytes, bytes 1 and 2 are reserved. Byte 3 values range from of 0 to 255, with 0 being no output and 255 being maximum intensity.
GET_MIC_LED_INTENSITY	0x13	0x02	This has no additional input data. Response of 3 bytes (command, length, 1 byte of Mic LED intensity). The data format matches that of the SET_MIC_LED_INTENSITY command.

Example SET_MIC_INTENSITY message (values indicated are the byte values in the actual UDP message payload):
```
Send: 12 05 00 00 80              Command = 0x02, Length = 0x02, Intensity = 128
```

```
Receive:  N/A - No received message for write command
```

Example GET_MIC_INTENSITY message (values indicated are the byte values in the actual UDP message payload):
```
Send: 13 02                  Command = 0x02, Length = 0x02
```

```
Receive: 13 05 40 40 80        Command = 0x08, Length = 0x0A, Ignore bytes 1 & 2..Intensity = 128
```

Device Security Commands

The following commands describe the station security parameters Zip4 and Zip4-3G.

Command Type	Command Byte	Length Byte	Data
SET_SECURITY_TIMEOUT	0x15	0x03	Set the time in minutes before the Zip4 transitions between "Active" and "Dormant" modes. Default = 5 minutes Range = 1 to 30 minutes.
GET_SECURITY_TIMEOUT	0x16	0x02	Returns a byte with the time in minutes for the security timeout. No additional input data. Response is 3 bytes (command, length, timeout). The data format matches that of the SET_SECURITY_TIMEOUT command.
SET_SECURITY_TIMELIMIT	0x17	0x03	Set the amount of time in seconds that a user has to enter the three digit security code. If the amount of time exceeds this limit, the unlock code procedure is restarted. Default = 10 seconds. Range = 5 to 20 seconds
GET_SECURITY_TIMELIMIT	0x18	0x02	Returns the time in seconds that a user has to enter the security code. Response of 3 bytes (command, length, time limit).
SET_SECURITY_LOCKOUT_MODE	0x19	0x03	Set the current security locking mode. Values can be one of: 0x00 - OPEN - No Security. 0x01 - ACTIVE – Security active and unit unlocked and usable. Timeouts apply. 0x02 - DORMANT – Security active and unit locked. Requires code to activate.
GET_SECURITY_CODE	0x1a	0x02	Gets the current three digit security code. Response of 5 bytes (command, length, 3 byte code). Each code byte has a value 0x01 through 0x04. The default security code is 311.
SET_SECURITY_CODE	0x1b	0x05	Sets the current three digit security code. Response of 3 bytes (command, length, 3 bytes of security code). The data format matches that of the GET_SECURITY_CODE command.

Example SET_SECURITY_CODE message (values indicated are the byte values in the actual UDP message payload):

`Send: 1b 05 03 02 01`   Command = 0x02, Length = 0x02, Security Code = "321"

`Receive:` N/A - No received message for write command

Example GET_MIC_INTENSITY message (values indicated are the byte values in the actual UDP message payload):

`Send: 1a 02`   Command = 0x02, Length = 0x02

`Receive: 1a 05 03 02 01`   Command = 0x08, Length = 0x0A, Security Code = "311"