HALO Smart Sensor BASIC API Developer’s Guide v2.2

Forward

This document describes the group of facilities of the Halo Smart Sensor known collectively as the BASIC API, or Application Programming Interface. This discussion is intended for use by programmers or integrators who are interested in integrating one or more HALO Smart Sensors (HALOs) with 3rd party (non-IPVideo) software components or systems.

In general, the HALO API is intended to transfer information efficiently from HALO over a conventional Ethernet network to the external program. To accomplish this goal, the API is divided into three sections: Event Driven Socket Connection, Heartbeat Socket Connection, and Event Data URL.

API Design

The API is designed using industry standard formats such as TCP/IP. HTTP, HTTPS, and JSON. The design does not require any special or proprietary techniques or libraries to be used in the development of the external program or application.

The API is flexible and can be configured and programmed to deliver exactly the data required and in the most efficient manner. The details of the operation of each of the above sections are covered in the following sections of this guide.

Event Driven Socket Connection

This facility is used to deliver specific Event data to the external program when a triggered Event value crosses a set threshold. This delivery is made to a TCP/IP socket programmed in the external application and is generally timely. This method delivers only Event data with minimum latency and light loading. This method typically provides alerts and alarm to an external program. It is designed to be flexible and easily implemented for a variety of interfaces.

Heartbeat Socket Connection

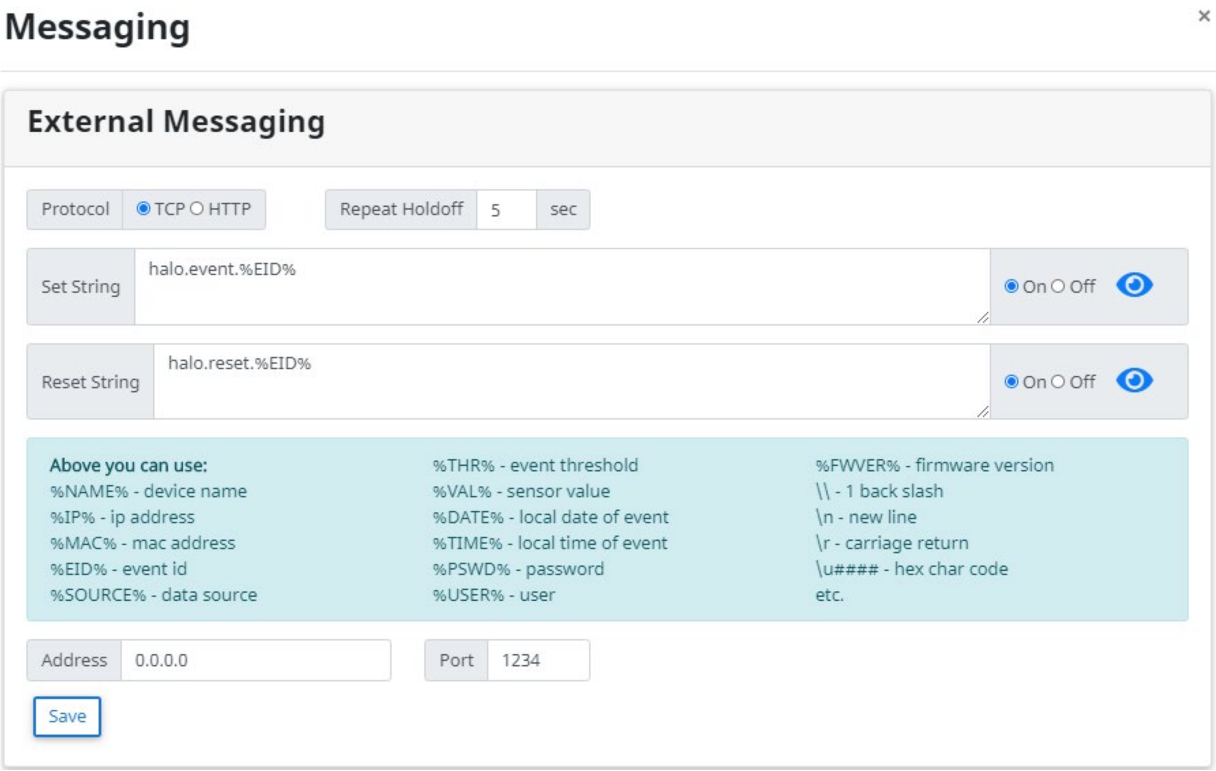
Heartbeat messages can be configured as desired but are generally short and intended to provide proof of live/availability of the Halo. They are delivered on a repeating basis with a configurable interval to TCP/IP socket programmed in the external application.

Event Data URL

This facility is only available under an NDA and should only be used when the external program requires access to any and all the Event values, thresholds, and state flags. This data is generally retrieved on demand by the external program but not at a very high frequency. This method generally incurs some latency when a modest polling rate is used. Typical polling rates range from once per minute to once per 5 seconds with an absolute maximum rate of once per second. This method can also be used to retrieve additional supporting data when an Event (alert) is received.

Event Driven Socket Connections

A section of the HALO internal web site Messaging page provides for configuration of a single 3rd party connection where various values can be sent to a remote TCP or UDP socket. Place holders (tokens) are used to insert live values into the transmitted text. Although labeled “External Messaging,” this channel can be used for almost any purpose requiring real time Event triggers, actively delivered by HALO. This arrangement is quite flexible because selections on the “Actions” page (shown on the following page) determine which HALO Events transmit through this channel.



Messaging

External Messaging

Protocol ● TCP ○ HTTP

Repeat Holdoff 5 sec

Set String: halo.event.%EID% — ● On ○ Off

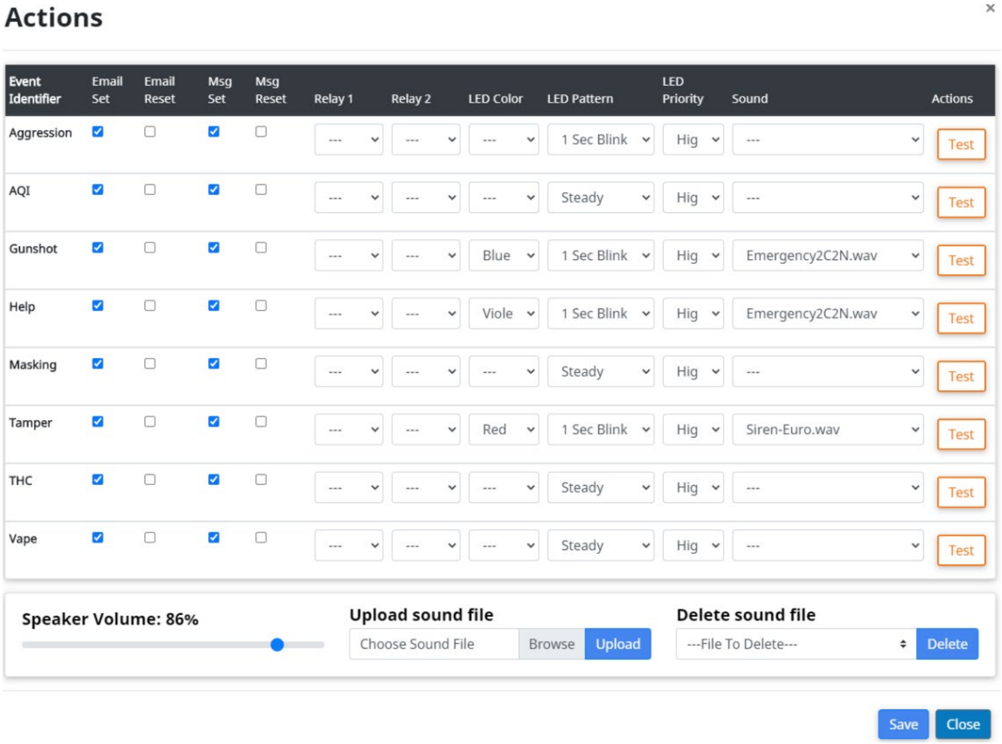
Reset String: halo.reset.%EID% — ● On ○ Off

Above you can use:

%NAME% - device name	%THR% - event threshold	%FWVER% - firmware version
%IP% - ip address	%VAL% - sensor value	\\ - 1 back slash
%MAC% - mac address	%DATE% - local date of event	\n - new line
%EID% - event id	%TIME% - local time of event	\r - carriage return
%SOURCE% - data source	%PSWD% - password	\u#### - hex char code
	%USER% - user	etc.

Address 0.0.0.0

Port 1234



Actions

Event Identifier	Email Set	Email Reset	Msg Set	Msg Reset	Relay 1	Relay 2	LED Color	LED Pattern	LED Priority	Sound	Actions
Aggression	☑	☐	☑	☐	---	---	---	1 Sec Blink	Hig	---	Test
AQI	☑	☐	☑	☐	---	---	---	Steady	Hig	---	Test
Gunshot	☑	☐	☑	☐	---	---	Blue	1 Sec Blink	Hig	Emergency2C2N.wav	Test
Help	☑	☐	☑	☐	---	---	Viole	1 Sec Blink	Hig	Emergency2C2N.wav	Test
Masking	☑	☐	☑	☐	---	---	---	Steady	Hig	---	Test
Tamper	☑	☐	☑	☐	---	---	Red	1 Sec Blink	Hig	Siren-Euro.wav	Test
THC	☑	☐	☑	☐	---	---	---	Steady	Hig	---	Test
Vape	☑	☐	☑	☐	---	---	---	Steady	Hig	---	Test

Speaker Volume: 86%

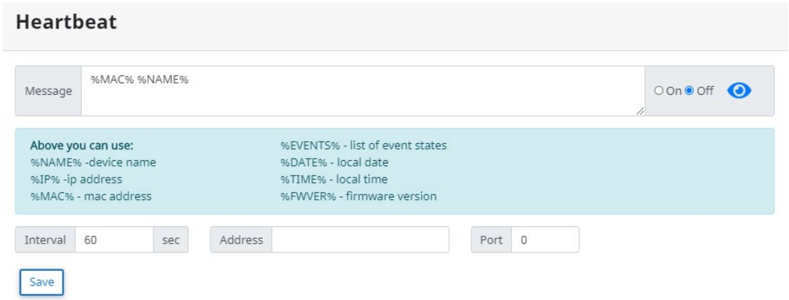
Upload sound file: Choose Sound File

Delete sound file: ---File To Delete---

Checkboxes in the "Msg Set" and "Msg Reset" columns determine which Events trigger sending. More on the setup of Events and Actions is available in the HALO Administrator's Guide.

Heartbeat Driven Socket Connections

The Heartbeat transmissions function in a similar way to the above except that there is no interaction with the Actions page. Instead, the Heartbeat transmission occurs on a regular basis as configured.



Heartbeat

Message: %MAC% %NAME%

○ On ● Off

Above you can use:	
%NAME% -device name	%EVENTS% - list of event states
%IP% -ip address	%DATE% - local date
%MAC% - mac address	%TIME% - local time
	%FWVER% - firmware version

Interval: 60 sec

Address:

Port: 0

While the primary purpose of the Heartbeat is to provide *proof of life* of a HALO Smart Sensor to a remote application, this message can also be used to transmit any or all of several values including the current Events state list enabling (for example) synchronization of a remote dashboard display or other user interface.

Delivery of JSON Event Messages

Some developers prefer to receive Event data formatted as industry standard self-labeled JSON rather than plain ASCII text as the former is more reliably and more easily parsed.

On the HALO web page “Messaging” tab, you can supply JSON messages in “External Messaging” settings “Set String” and “Reset String” and in the “Heartbeat” “Message.”

Examples:

“External Messaging” Settings Set String:

```
{ "device":"%NAME%", "event":"%EID%", "alarm":"yes" }
```

This will send a single TCP or UDP JSON message to the specified server reporting the friendly device name, event name and that it just started.

“External Messaging” Settings Reset String:

```
{ "device":"%NAME%", "event":"%EID%", "alarm":"no" }
```

This will send a single TCP or UDP JSON message to the specified server reporting the friendly device name, event name and that the condition has now stopped.

“Heartbeat” Message:

```
{ "device":"%NAME%", "alive":"%DATE% %TIME%" }
```

This will periodically send a TCP or UDP JSON message to the specified server reporting that the HALO is alive at the indicated time.