HeatBooster

Protocol specifications for firmware v2.7.0

Doc v2402

by SDR Engineering

1 Functional description

The HeatBooster module is able to communicate via MQTT and HTTP over WiFi to enable a range of smart home functions. Those smart home functions are accessible only through this MQTT and HTTP interface. For normal operation of the module, however, these smart home functions are not required and all required operating parameters can be configured using the built-in web interface.

On boot or reset, the startup procedure of the module is as follows, in chronological order:

- Check whether MODE button is pressed, if so:
  - Enter configuration mode, LED starts blinking slow (1x per 2 seconds).
  - WiFi access point is activated.
  - Web interface for configuration becomes available through http://192.168.4.1 when a client is connected to the WiFi access point.
  - Note: module configuration can be done as per the provided user manual.
  - No further functionality will be enabled while in configuration mode.
- If in Smart Home mode:
  - Connect to WiFi.
  - Connect to MQTT.
  - If WiFi and MQTT connect successfully, the status LED will be on continuously in dimmed mode.
- If in online mode:
  - Connect to WiFi.
- If in stand-alone mode:
  - Status LED will be on continuously in dimmed mode.
- Control algorithm is now activated.
- If in Smart Home mode: send MQTT update every 10 seconds .

The MQTT update and HTTP status request will have a range of messages that provide a complete overview of the current status. Additionally, the module will respond to MQTT and HTTP messages that are submitted to a suitable topic/URL. In the next section an overview of those items.

2 MQTT API

The HeatBooster module reports and responds under the topic “MODULE_NAME\”, where MODULE_NAME can be configured through the web interface in configuration mode.

1.1 Published messages

Examples:

Messages with topic “MODULE_NAME\ip” will report the IP-address of the module.
Messages with topic “MODULE_NAME\fan-controlmode” will report the fan speed control mode.

Overview:

Topic	Description
ip	WiFi IP-address of module
ssid	SSID of WiFi access point
rssi	Signal strength of WiFi connection in dB (should be > -80 dB for reliable operation, e.g. -75 dB is OK)
firmware	Firmware version
runtime	Runtime of module in seconds
interval	Publishing interval of MQTT messages
reconnects	Reconnection attempts (e.g. due to loss of connection)
control-state	Current operating mode; 0: idle, 1: overrun, 2: manual, 3: heat, 4: defrost, 5: startup support, 6: cool, 7: slave, 8: sensor error.
fan-controlmode	Fan speed control mode 0=automatic mode 1=manual mode
fan-boostmode	Boost mode enable 0=disabled 1=enabled
fan-enabled	Fan power enabled 0=fans are not powered 1=fans are powered
fan-speed	Current fan speed in %
ambientcontrol-enable	Ambient temperature control enabled 0=disabled 1=enabled
ambientcontrol-temp	Current target ambient temperature in °C, if ambient control is enabled
temp-inlet	Inlet water temperature in °C
temp-inlet-rate	Inlet water temperature rate of change in °C/min
temp-outlet	Outlet water temperature in °C
temp-delta-io	Temperature difference inlet-outlet in °C
temp-ambient	Ambient air temperature in °C

1.2 Subscribed topics

Example:

Sending a message with value 60 to topic “MODULE_NAME\fan-speed-ref” will set the fan speed to 60%. The module will only apply this value if you have set fan-controlmode-ref=1 to select manual mode.

Overview:

Topic	Description
fan-boostmode-ref	Enable boost mode 0 = boost mode disabled 1 = boost mode enabled
fan-controlmode-ref	Change control mode of fan speed 0 = automatic mode 1 = manual mode
fan-speed-ref	Set the fan speed in % (only in manual mode)
ambientcontrol-enable-ref	Enable ambient temperature control 0 = disabled 1 = enabled
ambientcontrol-temp-ref	Temperature setpoint in °C
temp-ambient-ext	External ambient temperature input in °C (overrides internal temperature reading)
temp-inlet-ext	External inlet water temperature input in °C (overrides internal temperature reading)
temp-outlet-ext	External outlet water temperature input in °C (overrides internal temperature reading)

Note: the external temperature sensor inputs require an update at least every 30 seconds, otherwise the override will be canceled and the internal temperature sensor will be used again.

Common:

The following commands are received by all HeatBooster modules in the network. The message should be published to root (without including “MODULE_NAME\”).

heatbooster/start-ota-update	Initiates an OTA update of all HeatBooster modules registered to the MQTT server

2 HTTP API

The HeatBooster can be monitored and controlled by sending HTTP GET requests. This API is also available when the MQTT mode is disabled.

2.1 Requesting state

The current state can be requested through http://ip_address/getStatus, where the response is a JSON string containing the full state of the HeatBooster.

If you configured a password, the request needs to be made as follows:
http://admin:PASSWORD@ipaddress/getStatus

2.2 Setting parameters

A few parameters can be controlled through the HTTP API, by submitting a request to http://ipaddress/setStatus. The following GET-parameters can be used:

Parameter	Description
FAN_CONTROLMODE	Fan speed control mode 0=automatic mode 1=manual mode
FAN_BOOSTMODE	Boost mode enable 0=disabled 1=enabled
FAN_SPEED	Current fan speed in %, note that you need to set CONTROLMODE=1 as well when setting the fan speed manually.
AMBIENTCONTROL_ENABLE	Ambient temperature control enabled 0=disabled 1=enabled
AMBIENTCONTROL_TEMP	Current target ambient temperature in °C, if ambient control is enabled

Examples

To set the fan speed to 80%, make the following HTTP request:
http://ipaddress/setStatus?FAN_SPEED=80&FAN_CONTROLMODE=1

To disable manual mode again the request is as follows:
http://ipaddress/setStatus?FAN_CONTROLMODE=0

If you configured a password, the request needs to be made as follows:
http://admin:PASSWORD@ipaddress/setStatus?FAN_SPEED=80&FAN_CONTROLMODE=1