

# MQTT Protocol of the BATmode System

## Messages and Commands



**bat bioacoustictechnology GmbH**

**Revision: 1.2.0**

---

## 1 Inhalt

1	Inhalt .....	1
2	Revision History .....	3
3	Document Purpose and Scope.....	3
4	Conventions .....	3
5	Commands .....	4
5.1	Run Microphone Test (mictest_run).....	4
5.2	Set Microphone Test settings (mictest_set) .....	4
5.3	Reboot the BATmode (reboot).....	5
5.4	Start the WIFI Hotspot (wifi_start).....	5
5.5	Stop the WIFI Hotspot (wifi_stop) .....	5
5.6	Run backup to drive D:/ (backup_run).....	6
5.7	Set backup settings (backup_set).....	6
5.8	Set BATmode ID (id_set).....	6
5.9	Load BMU RECORDER Settings (monitoring_loadbmu).....	7
5.10	Set Location of BATmode (location_set) .....	7
5.11	Kill the RECORDER software (recorder_kill).....	7
5.12	Reset the Cellular Modem (modem_reset) .....	8
5.13	Reset the UltraSoundGates (usg_reset) .....	8
5.14	Set MQTT settings (mqtt_set).....	8
5.15	Start the acoustic monitoring (monitoring_start) .....	8
5.16	Stop the Acoustic Monitoring (monitoring_stop) .....	9
5.17	Change settings of the acoustic monitoring (monitoring_set).....	9
5.18	Change the Calibration Values of one Microphone Channel (calibration_set).....	10
5.19	Get the log file of the meteorological sensors (meteorolog_get) .....	11
5.20	Get the remote log file (remotelog_get).....	11
5.21	Get the log file of recording times (rectimelog_get).....	11
5.22	Get the log file of microphone tests (mictestlog_get).....	12

6	Status Messages.....	13
6.1	Ping .....	13
6.2	ID of the BATmode .....	13
6.3	Disc space of the Main Drive C:/ and the Backup Drive D:/ .....	13
6.4	Backup Information .....	14
6.5	Time and geographic Location related Information.....	14
6.6	Battery and Solar Power Supply related Information .....	14
6.7	State of WIFI Hotspot .....	15
6.8	State of Cellular Data Connection .....	15
6.9	Current Meteorological Data.....	15
6.10	Microphone Test Settings.....	16
6.11	Results of last Microphone Test .....	16
6.12	Current State of Monitoring .....	17
6.13	Numbers of recorded acoustic Files .....	17
6.14	Current Microphone Channel Calibrations.....	18
6.15	Trigger Channel 1.....	19
6.16	Trigger Channel 2.....	20
6.17	Trigger Channel 3.....	20
6.18	Trigger Channel 4.....	20
6.19	File Info of a new recorded WAV file on microphone channel 1 .....	20
6.20	File Info of a new recorded WAV file on microphone channel 2 .....	21
6.21	File Info of a new recorded WAV file on microphone channel 3 .....	21
6.22	File Info of a new recorded WAV file on microphone channel 4 .....	22
6.23	Transfer of a new recorded WAV file on microphone channel 1.....	22
6.24	Transfer of a new recorded WAV file on microphone channel 2.....	23
6.25	Transfer of a new recorded WAV file on microphone channel 3.....	23
6.26	Transfer of a new recorded WAV file on microphone channel 4.....	23
6.27	New recorded Meteorological Data .....	24
7	Last Will Message .....	25
8	Post-Process AI Call Detector.....	26
8.1	Results of the AI Call Detector.....	26

## 2 Revision History

Revision	Date	Change
1.0.0	16.12.2022	Initial Release
1.1.0	10.05.2023	Add AI Call Detector Post Process
1.2.0	29.06.2026	Extend <code>calibration_set</code> command as well as monitoring/calibration message for Cable Length and Cable Type

## 3 Document Purpose and Scope

This document describes the MQTT messages and commands used to control a BATmode system and to get status information, logs and WAV files.

It is primarily intended to guide the reader through the general structure of the commands itself as well as through the individual structures of the various information messages received from the BATmode.

bat bioacoustictechnology GmbH will provide a Qt example implementation for a MQTT client to display status information and send commands.

## 4 Conventions

Any reference to data items, data structures, functions or MQTT topics are set in `Courier New` font. The same goes for any data type (e.g., `bool`).

The term [MAC] represents the MAC address of the respective BATmode device, e.g. 11:22:33:44:AA:BB.

Time formats uses the following abbreviations:

- yyyy: Year, e.g., 2022
- MM: Month, e.g., 01 for January
- dd: Day in Month, e.g., 31
- hh: Hours, e.g., 15 for 3 o'clock PM
- mm: Minutes, e.g., 05
- ss: Seconds, e.g., 05

## 5 Commands

If the `<MQTT remote>` option is enabled in BATcontrol, the BATmode device listens to commands on the topic `batmode/[MAC]/request`

If using MQTTv3.1.1 or MQTTv3.1 protocol, data or command execution status are returned on the topic `batmode/[MAC]/response`. If using MQTTv5, a response topic as well as correlation data (integer counter) can be sent along with the command message. In this case, the BATmode sends its command response to the respective response topic. The correlation data will be forwarded unchanged. This allows to implement a command-response-structure and send response data only to the requesting client. Furthermore, the response can be uniquely assigned to the respective command.

### 5.1 Run Microphone Test (`mictest_run`)

The `mictest_run` command is used to run a microphone test. This command will only be executed when the acoustic monitoring is running.

#### 5.1.1 Parameters

The `mictest_run` command does not take any parameters.

#### 5.1.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.2 Set Microphone Test settings (`mictest_set`)

The `mictest_set` command allows to change the settings of the microphone test. This command will only be executed when the acoustic monitoring is stopped.

#### 5.2.1 Parameters

The `mictest_set` command takes the following parameters:

- `time`: A string in the format `hh:mm` representing the daily microphone test execution time. (optional)
- `enable`: A digit (0/1) for enabling or disabling microphone tests (optional)
- `atstart`: A digit (0/1) to enable or disable a microphone test when the monitoring is started (optional)

At least the parameter `time` or the other both parameters `enable` and `atstart` are required.

#### 5.2.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

Example

Topic: `batmode/[MAC]/request`

Message: `mictest_set enable=1 atstart=0 time=12:00`

## 5.3 Reboot the BATmode (`reboot`)

The `reboot` command is used to reboot the BATmode device.

### 5.3.1 Parameters

The `reboot` command does not take any parameters.

### 5.3.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.4 Start the WIFI Hotspot (`wifi_start`)

The `wifi_start` command is used to set the SSID and password of the WIFI Hotspot and start it.

### 5.4.1 Parameters

The `wifi_start` command takes the following parameters:

- `ssid`: A string representing the SSID of the WIFI Hotspot (at least 4 characters, no special characters allowed) (optional)
- `password`: A string representing the password for the WIFI Hotspot (at least 8 characters, no special characters allowed) (optional)

If the parameters `ssid` or `password` are not specified, the WIFI Hotspot will start with its last SSID or password setting, respectively.

### 5.4.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.4.3 Example

Topic: `batmode/[MAC]/request`

Message: `wifi_start ssid=BATmode password=12345678`

## 5.5 Stop the WIFI Hotspot (`wifi_stop`)

The `wifi_stop` command is used to deactivate the WIFI Hotspot.

### 5.5.1 Parameters

The `wifi_stop` command does not take any parameter.

### 5.5.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.6 Run backup to drive D:/ (backup\_run)

The `backup_run` command is used to trigger a backup of the monitoring directory to the backup drive D:/

### 5.6.1 Parameters

The `backup_run` command does not take any parameter.

### 5.6.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.7 Set backup settings (backup\_set)

The `backup_set` command allows to change the settings of the backup.

### 5.7.1 Parameters

The `backup_set` command takes the following parameters:

- `time`: A string in the format `hh:mm` representing the daily backup execution time. (optional)
- `enable`: A digit (0/1) for enabling or disabling daily backups (optional)

At least one parameter is required.

### 5.7.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.7.3 Example

Topic: `batmode/[MAC]/request`

Message: `backup_set enable=1 time=12:00`

## 5.8 Set BATmode ID (id\_set)

The `id_set` command allows to set the ID of the BATmode. This command will only be executed when the acoustic monitoring is stopped.

### 5.8.1 Parameters

The `id_set` command takes the following parameter:

- `id`: A string representing the ID for the BATmode system. (No special characters allowed) (required)

### 5.8.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.8.3 Example

Topic: `batmode/[MAC]/request`

Message: `id_set id=BATmode`

## 5.9 Load BMU RECORDER Settings (`monitoring_loadbmu`)

The `monitoring_loadbmu` command is used to load default BMU settings of the RECORDER software. This command will only be executed when the acoustic monitoring is stopped.

### 5.9.1 Parameters

The `monitoring_loadbmu` command does not take any parameter.

### 5.9.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.10 Set Location of BATmode (`location_set`)

The `location_set` command allows to set the geographic location of the BATmode system to calculate sunrise and sunset times. The location is defined as latitude and longitude, coordinates in the geographic coordinate system. This command will only be executed when the acoustic monitoring is stopped.

### 5.10.1 Parameters

The `location_set` command takes the following parameters:

- `lat`: A double representing the latitude of the BATmode location (-90.0 ...90.0) (required)
- `long`: A double representing the longitude of the BATmode location (-180.0 ...180.0) (required)

### 5.10.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.10.3 Example

Topic: `batmode/[MAC]/request`

Message: `location_set lat=51.6 long=12.4`

## 5.11 Kill the RECORDER software (`recorder_kill`)

The `recorder_kill` command is used to forcefully terminate the RECORDER software. This command will only be executed when the acoustic monitoring is running.

### 5.11.1 Parameters

The `recorder_kill` command does not take any parameter.

### 5.11.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.12 Reset the Cellular Modem (`modem_reset`)

The `modem_reset` command is used to trigger a hardware reset of the cellular module.

### 5.12.1 Parameters

The `modem_reset` command does not take any parameter.

### 5.12.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.13 Reset the UltraSoundGates (`usg_reset`)

The `usg_reset` command is used to trigger a short interruption of the power supply of the UltraSoundGate USB ports. This results in a hardware reset of the connected UltraSoundGates.

### 5.13.1 Parameters

The `usg_reset` command does not take any parameter.

### 5.13.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.14 Set MQTT settings (`mqtt_set`)

The `mqtt_set` command allows to change the MQTT settings.

### 5.14.1 Parameters

The `mqtt_set` command takes the following parameter:

- `sendWavs`: A digit (0/1) for enabling or disabling the sending of incoming WAV files via MQTT (required)

### 5.14.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.14.3 Example

Topic: `batmode/[MAC]/request`

Message: `mqtt_set sendWavs=1`

## 5.15 Start the acoustic monitoring (`monitoring_start`)

The `monitoring_start` command is used to start the acoustic recording.

### 5.15.1 Parameters

The `monitoring_start` command takes one of the following combination of parameters:

- No parameter: The acoustic recording is started ungated without any temporal restrictions.
- `relPeriod`: The acoustic recording is started relative to sunset and sunrise. The offset from the sunset and sunrise is specified in minutes. Negative numbers represent minutes before sunset or

sunrise, respectively. Positive numbers represent minutes after sunset or sunrise, respectively. Both numbers are separated by "/" (e.g., `relPeriod=-30/60`).

- `period1`: The acoustic recording is started in the absolute period of time specified by a string with the following format (hh:mm-hh:mm) e.g., `period1=16:00-09:00`.
- `period1` and `period2`: The acoustic recording is started in the 2 absolute periods of time specified each by a string with the following format (hh:mm-hh:mm) e.g., `period1=16:00-09:00`  
`period2= 11:00-12:00`.

### 5.15.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.15.3 Examples

Topic: `batmode/[MAC]/request`

Message:

`monitoring_start` (Starts the monitoring with permanent acoustic recording.)

`monitoring_start relPeriod=-30/60` (Starts the monitoring with recording periods relative to sunset and sunrise. In this case the acoustic recording will start each day 30 minutes before sunset and will end 60 minutes after sunrise.)

`monitoring_start period1=16:00-10:00` (Starts the monitoring with absolute recording periods regarding to time. In this case from 16 o'clock in the afternoon to 10 o'clock in the morning.)

`monitoring_start period1=16:00-23:00 period2=04:00-09:00` (Starts the monitoring with absolute recording periods regarding to time. In this case from 16 o'clock in the afternoon to 23 o'clock in the night as well as from 4 o'clock to 10 o'clock in the morning.)

## 5.16 Stop the Acoustic Monitoring (`monitoring_stop`)

The `monitoring_stop` command is used to stop the acoustic recording.

### 5.16.1 Parameters

The `monitoring_stop` command does not take any parameter.

### 5.16.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

## 5.17 Change settings of the acoustic monitoring (`monitoring_set`)

The `monitoring_set` command is used to change the settings of acoustic recording. This command will only be executed when the acoustic monitoring is stopped.

### 5.17.1 Parameters

The `monitoring_set` command takes one of the following combination of parameters:

- `dir`: A string representing the absolute path of the new monitoring directory. (separator is "/")

- `ch` and `name`: Changing the name of a microphone channel. `ch` is a digit representing the microphone channel (1/2/3/4). `name` is a string representing the new channel name.

### 5.17.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.17.3 Examples

Topic: `batmode/[MAC]/request`

Message:

```
monitoring_set dir=C:/BATmode
```

```
monitoring_set ch=1 name=chl
```

## 5.18 Change the Calibration Values of one Microphone Channel (`calibration_set`)

The `calibration_set` command is used to change the calibration value of a microphone channel. This command will only be executed when the acoustic monitoring is stopped.

### 5.18.1 Parameters

The `calibration_set` command takes the following parameters:

- `ch`: A digit representing the microphone channel (1/2/3/4) (required)
- `triggerLevel`: A double representing the new Trigger Level (0.0...200.0) (required)
- `referenceSpl`: A double representing the new SPL of Reference Signal (0.0...200.0) (required)
- `referenceFs`: A double representing the new Full-Scale Level of Reference Signal (-100.0...0.0) (required)
- `integratedRef`: 0 if the microphone does not include a reference speaker. 1 if the microphone includes a reference speaker. (required)
- `cableLength`: An int representing the new Microphone Cable Length in m (0...200) (optional, default is 0) (requires BATcontrol 4.2.0 and higher)
- `cableType`: 0 if the Microphone Cable is a standard Cable. 1 if the Microphone Cable is an offshore Cable. (optional, default is 0) (requires BATcontrol 4.2.0 and higher)

### 5.18.2 Returned data

The only returned data is the command execution status with topic `batmode/[MAC]/response`.

### 5.18.3 Example

Topic: `batmode/[MAC]/request`

Message:

```
calibration_set ch=1 triggerLevel=37.0 referenceSpl=90.0 referenceFs=-20.0
integratedRef=1 cableLength=2 cableType=0
```

## 5.19 Get the log file of the meteorological sensors (`meteorolog_get`)

The `meteorolog_get` command is used to request the file “conditionlog.txt” from the BATmode device.

### 5.19.1 Parameters

The `meteorolog_get` command does not take any parameter.

### 5.19.2 Returned data

The command returns the file “conditionlog.txt” as message either to the topic `batmode/[MAC]/responseFile` or to a specified response topic.

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: File as `byte array`.

## 5.20 Get the remote log file (`remotelog_get`)

The `remotelog_get` command is used to request the file “remotelog.txt” from the BATmode device.

### 5.20.1 Parameters

The `remotelog_get` command does not take any parameter.

### 5.20.2 Returned data

The command returns the file “remotelog.txt” as message either to the topic `batmode/[MAC]/responseFile` or to a specified response topic.

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: File as `byte array`.

## 5.21 Get the log file of recording times (`rectimelog_get`)

The `rectimelog_get` command is used to request the file “rectimelog.txt” from the BATmode device.

### 5.21.1 Parameters

The `rectimelog_get` command does not take any parameter.

### 5.21.2 Returned data

The command returns the file “rectimelog.txt” as message either to the topic `batmode/[MAC]/responseFile` or to a specified response topic.

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: File as `byte array`.

## 5.22 Get the log file of microphone tests (`mictestlog_get`)

The `mictestlog_get` command is used to request the file “reference.txt” from the BATmode device.

### 5.22.1 Parameters

The `mictestlog_get` command does not take any parameter.

### 5.22.2 Returned data

The command returns the file “reference.txt” as message either to the topic `batmode/[MAC]/responseFile` or to a specified response topic.

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: File as `byte array`.

## 6 Status Messages

If the `<MQTT>` option is enabled in BATcontrol, the BATmode sends periodic and event-driven status messages.

### 6.1 Ping

To allow the detection of connection losses from the BATmode to the host, the BATmode device periodically sends ping messages.

#### 6.1.1 Send time

One-minute interval.

#### 6.1.2 Topic

batmode/[MAC]/ping

#### 6.1.3 Format

The message is the current UTC time of the BATmode as `string` in the format [yyyy-MM-dd hh:mm:ss].

### 6.2 ID of the BATmode

#### 6.2.1 Send time

After the BATmode is connected to the host as well as at an ID changed event.

#### 6.2.2 Topic

batmode/[MAC]/id

#### 6.2.3 Format

The message is the BATmode ID as `string`.

### 6.3 Disc space of the Main Drive C:/ and the Backup Drive D:/

#### 6.3.1 Send time

After the BATmode is connected to the host and in a ten-minute interval.

#### 6.3.2 Topic

batmode/[MAC]/discspace

#### 6.3.3 Format

The message contains the total as well as the free disc spaces available on drives C:/ and D:/ in a JSON encoded format:

- freeC: double [GB]
- totalC: double [GB]
- freeD: double [GB]
- totalD: double [GB]

## 6.4 Backup Information

### 6.4.1 Send time

After the BATmode is connected to the host and at a backup setting changed event.

### 6.4.2 Topic

batmode/[MAC]/backup

### 6.4.3 Format

The message contains the following information in a JSON encoded format:

- `state: string` (enabled/disabled/error)
- `time: string [hh:mm]` – Execution time of the daily backup.
- `lastdate: string [yyyy-MM-dd hh:mm:ss]` – Last date the backup was successfully executed. Empty if no backup was executed since BATcontrol was started.
- `error: string` – Last error string, empty if no error is active.

## 6.5 Time and geographic Location related Information

### 6.5.1 Send time

After the BATmode is connected to the host and at a time settings changed event.

### 6.5.2 Topic

batmode/[MAC]/timeinfo

### 6.5.3 Format

The message contains the following information in a JSON encoded format:

- `rise: string [hh:mm]` – Current time of sunrise.
- `set: string [hh:mm]` – Current time of sunset.
- `long: double` – Longitude coordinate of the geographic location of the BATmode.
- `lat: double` – Latitude coordinate of the geographic location of the BATmode.
- `daylightsaving: bool` – True if daylightsaving is enabled in Windows system time.
- `timezone: string` – Timezone of Windows system time.

## 6.6 Battery and Solar Power Supply related Information

### 6.6.1 Send time

After the BATmode is connected to the host and in a ten-minute interval.

### 6.6.2 Topic

batmode/[MAC]/battery

### 6.6.3 Format

The message contains the following information in a JSON encoded format:

- `v: double [V]` – Current battery voltage.
- `ppv: double [W]` – Current output power of photovoltaic panels.
- `mpt: double [W]` – Maximum output power photovoltaic panels today.

- `yt`: double [Wh] – Today's solar yield.
- `mpy`: double [W] – Maximum output power photovoltaic panels yesterday.
- `yy`: double [Wh] –Yesterdays solar yield.

## 6.7 State of WIFI Hotspot

### 6.7.1 Send time

After the BATmode is connected to the host and at a WIFI Hotspot changed event.

### 6.7.2 Topic

`batmode/[MAC]/connectivity/wifi_state`

### 6.7.3 Format

The message contains the following information in a JSON encoded format:

- `state`: string (on/off/error)
- `ssid`: string – SSID of the WIFI Hotspot
- `pw`: string – Password of the WIFI Hotspot
- `error`: string – Last error string, empty if no error is active.

## 6.8 State of Cellular Data Connection

### 6.8.1 Send time

After the BATmode is connected to the host and at a cellular data connection changed event.

### 6.8.2 Topic

`batmode/[MAC]/connectivity/cellular_state`

### 6.8.3 Format

The message contains the following information in a JSON encoded format:

- `state`: string (connected/disconnected/connecting/disconnecting/error)
- `provider`: string – Name of the cellular provider of the inserted SIM card.
- `dataclass`: string – Connection type, e.g., LTE or UMTS.
- `signal`: int (0-100) [%] – Signal strength.
- `error`: string – Last error string, empty if no error is active.

## 6.9 Current Meteorological Data

### 6.9.1 Send time

After the BATmode is connected to the host and in a one-minute interval.

### 6.9.2 Topic

`batmode/[MAC]/meteo`

### 6.9.3 Format

The message contains the following information in a JSON encoded format:

- `mode: int (-1/0/1/2/3)` – Current selected meteorological data input. (-1: BATmode 2S – No meteorological sensors available. 0: Analog temperature and analog infrared precipitation sensor. 1: Analog temperature and serial connected laser precipitation sensor from Thies Clima. 2: Temperature and precipitation data from serial connected laser precipitation sensor from Thies Clima. 3: Analog temperature sensor and serial connected present weather sensor for precipitation and WMO code from Biral.)
- `temp: double [°C]` – Current temperature.
- `prec: double [mm/min]` – Current precipitation.
- `wmo: string` – Current WMO code. Only valid for mode 3. XX depicts invalid data.

## 6.10 Microphone Test Settings

### 6.10.1 Send time

After the BATmode is connected to the host and when microphone test settings are changed.

### 6.10.2 Topic

`batmode/[MAC]/monitoring/mictest`

### 6.10.3 Format

The message contains the settings of the microphone test in a JSON encoded format:

- `state: string (enabled/disabled)`
- `time: string [hh:mm]` – Execution time of the daily microphone test.
- `atstart: bool` – True if a microphone test is executed on monitoring start.

## 6.11 Results of last Microphone Test

### 6.11.1 Send time

After the BATmode is connected to the host and when new microphone test results are available.

### 6.11.2 Topic

`batmode/[MAC]/monitoring/mictestresult`

### 6.11.3 Format

The message contains the results of the last microphone test in a JSON encoded format:

- `date: string [yyyy-MM-dd hh:mm:ss]` – Date when the results were generated.
- `ch1:`
  - `abs: double (-1 if not available)` – Absolute loudness of the testsignal in dBFS of microphone 1.
  - `rel: double (-1 if not available)` – Deviation from the calibration values of microphone 1 in dB. Recorded data is valid for deviations between +/-6dB.
- `ch2:`
  - `abs: double (-1 if not available)` – Absolute loudness of the testsignal in dBFS of microphone 2.
  - `rel: double (-1 if not available)` – Deviation from the calibration values of microphone 2 in dB. Recorded data is valid for deviations between +/-6dB.

- ch3:
  - abs: double (-1 if not available) – Absolute loudness of the testsignal in dBFS of microphone 3.
  - rel: double (-1 if not available) – Deviation from the calibration values of microphone 3 in dB. Recorded data is valid for deviations between +/-6dB.
- ch4:
  - abs: double (-1 if not available) – Absolute loudness of the testsignal in dBFS of microphone 4.
  - rel: double (-1 if not available) – Deviation from the calibration values of microphone 4 in dB. Recorded data is valid for deviations between +/-6dB.

## 6.12 Current State of Monitoring

### 6.12.1 Send time

After the BATmode is connected to the host and at a monitoring state or settings changed event.

### 6.12.2 Topic

batmode/[MAC]/monitoring/state

### 6.12.3 Format

The message contains the state and settings of the monitoring in a JSON encoded format:

- state: string (on/off/error)
- abs1: string [hh:mm-hh:mm] – absolute Recording Period 1
- abs2: string [hh:mm-hh:mm] – absolute Recording Period 2
- rel: string [mm/mm] – relative Recording Period relative to sunset and sunrise, e.g., -30/60 depicts a relative Recording Period from 30 minutes before sunset till 60 minutes after sunset.
- usgs: int (0/1/2/3/4/100) – Number of connected UltraSoundGates. 100 depicts a connected UltraSoundGate 416Hnbm with four microphone channels.
- dir: string - Monitoring directory with "/" as separator.
- name1: string – Name of microphone channel 1.
- name2: string – Name of microphone channel 2.
- name3: string – Name of microphone channel 3.
- name4: string – Name of microphone channel 4.
- error: string – Last error string, empty if no error is active.

## 6.13 Numbers of recorded acoustic Files

### 6.13.1 Send time

After the BATmode is connected to the host as well as at a file recorded event.

### 6.13.2 Topic

batmode/[MAC]/monitoring/filenumbers

### 6.13.3 Format

The message contains the numbers of recorded files with respect to the current available microphone channels (depending on the connected UltraSoundGates) in a JSON encoded format:

- `date: string [yyyy-MM-dd hh:mm:ss]`
- `ch1:`
  - `abs: int (-1 if channel not available)` – Number of all recorded WAV files of microphone channel 1.
  - `24h: int (-1 if channel not available)` – Number of recorded WAV files of microphone channel 1 in the last 24h.
- `ch2:`
  - `abs: int (-1 if channel not available)` – Number of all recorded WAV files of microphone channel 2.
  - `24h: int (-1 if channel not available)` – Number of recorded WAV files of microphone channel 2 in the last 24h.
- `ch3:`
  - `abs: int (-1 if channel not available)` – Number of all recorded WAV files of microphone channel 3.
  - `24h: int (-1 if channel not available)` – Number of recorded WAV files of microphone channel 3 in the last 24h.
- `ch4:`
  - `abs: int (-1 if channel not available)` – Number of all recorded WAV files of microphone channel 3.
  - `24h: int (-1 if channel not available)` – Number of recorded WAV files of microphone channel 4 in the last 24h.

## 6.14 Current Microphone Channel Calibrations

### 6.14.1 Send time

After the BATmode is connected to the host as well as at a calibration changed event.

### 6.14.2 Topic

`batmode/[MAC]/monitoring/calibration`

### 6.14.3 Format

The message contains the following information in a JSON encoded format:

- `ch1:`
  - `integratedRef: bool` – True if microphone at Channel 1 has an integrated testsignal speaker.
  - `referenceFs: double [dBFS]` – Full Scale Level of Reference Signal Channel 1.
  - `referenceSpl: double [dB SPL]` – SPL of Reference Signal Channel 1.
  - `triggerLevel: double [dB SPL]` – Level of Trigger Threshold Channel 1.
  - `cableLength: int [m]` – Length of Microphone Cable of Channel 1 (available for BATcontrol 4.2.0 and higher).
  - `cableType: bool` – True for offshore Microphone Cable is, False for standard Microphone Cable (available for BATcontrol 4.2.0 and higher).
- `ch2:`
  - `integratedRef: bool` – True if microphone at channel 2 has an integrated testsignal speaker.
  - `referenceFs: double [dBFS]` – Full Scale Level of Reference Signal Channel 2.

- `referenceSpl`: double [dB SPL] – SPL of Reference Signal Channel 2.
- `triggerLevel`: double [dB SPL] – Level of Trigger Threshold Channel 2.
- `cableLength`: int [m] – Length of Microphone Cable of Channel 2 (available for BATcontrol 4.2.0 and higher).
- `cableType`: bool – True for offshore Microphone Cable is, False for standard Microphone Cable (available for BATcontrol 4.2.0 and higher).
- 
- `ch3`:
  - `integratedRef`: bool – True if microphone at channel 3 has an integrated test signal speaker.
  - `referenceFs`: double [dBFS] – Full Scale Level of Reference Signal Channel 3.
  - `referenceSpl`: double [dB SPL] – SPL of Reference Signal Channel 3.
  - `triggerLevel`: double [dB SPL] – Level of Trigger Threshold Channel 3.
  - `cableLength`: int [m] – Length of Microphone Cable of Channel 3 (available for BATcontrol 4.2.0 and higher).
  - `cableType`: bool – True for offshore Microphone Cable is, False for standard Microphone Cable (available for BATcontrol 4.2.0 and higher).
  -
- `ch4`:
  - `integratedRef`: bool – True if microphone at channel 4 has an integrated test signal speaker.
  - `referenceFs`: double [dBFS] – Full Scale Level of Reference Signal Channel 4.
  - `referenceSpl`: double [dB SPL] – SPL of Reference Signal Channel 4.
  - `triggerLevel`: double [dB SPL] – Level of Trigger Threshold Channel 4.
  - `cableLength`: int [m] – Length of Microphone Cable of Channel 4 (available for BATcontrol 4.2.0 and higher).
  - `cableType`: bool – True for offshore Microphone Cable is, False for standard Microphone Cable (available for BATcontrol 4.2.0 and higher).

## 6.15 Trigger Channel 1

### 6.15.1 Send time

The trigger message is sent at a trigger start (begin of a WAV file writing) and trigger end (when the file was successfully written to the internal drive) event of microphone channel 1.

### 6.15.2 Topic

`batmode/[MAC]/monitoring/trigging/ch1`

### 6.15.3 Format

The message contains a `bool` that is `true` if the BATmode currently writes a file to the internal drive and `false` if there is currently no trigger event that causes a file writing at microphone channel 1.

## 6.16 Trigger Channel 2

### 6.16.1 Send time

The trigger message is sent at a trigger start (begin of a WAV file writing) and trigger end (when the file was successfully written to the internal drive) event of microphone channel 2.

### 6.16.2 Topic

batmode/[MAC]/monitoring/triggering/ch2

### 6.16.3 Format

The message contains a `bool` that is `true` if the BATmode currently writes a file to the internal drive and `false` if there is currently no trigger event that causes a file writing at microphone channel 2.

## 6.17 Trigger Channel 3

### 6.17.1 Send time

The trigger message is sent at a trigger start (begin of a WAV file writing) and trigger end (when the file was successfully written to the internal drive) event of microphone channel 3.

### 6.17.2 Topic

batmode/[MAC]/monitoring/triggering/ch3

### 6.17.3 Format

The message contains a `bool` that is `true` if the BATmode currently writes a file to the internal drive and `false` if there is currently no trigger event that causes a file writing at microphone channel 3.

## 6.18 Trigger Channel 4

### 6.18.1 Send time

The trigger message is sent at a trigger start (begin of a WAV file writing) and trigger end (when the file was successfully written to the internal drive) event of microphone channel 4.

### 6.18.2 Topic

batmode/[MAC]/monitoring/triggering/ch4

### 6.18.3 Format

The message contains a `bool` that is `true` if the BATmode currently writes a file to the internal drive and `false` if there is currently no trigger event that causes a file writing at microphone channel 4.

## 6.19 File Info of a new recorded WAV file on microphone channel 1

### 6.19.1 Send time

The message is sent each time a new WAV file was recorded on channel 1.

### 6.19.2 Topic

batmode/[MAC]/monitoring/new/fileinfo/ch1

### 6.19.3 Format

The message contains information on the new recorded WAV file in a JSON encoded format:

- filename: string – Filename of the recorded WAV file.
- channel: int (1/2/3/4) – Microphone channel
- date: string [yyyy-MM-dd hh:mm:ss] – Recording data and time.
- samplerate: int [1/s] – Samplerate
- bits: int – Bits per Sample
- length: double [s] – Temporal length of the recorded WAV file.
- species: string ["No Call" or comma separated abbreviations of detected species] – Result of the post-processing AI Call Detector. Further information see chapter 8.

## 6.20 File Info of a new recorded WAV file on microphone channel 2

### 6.20.1 Send time

The message is sent each time a new WAV file was recorded on channel 2.

### 6.20.2 Topic

batmode/[MAC]/monitoring/new/fileinfo/ch2

### 6.20.3 Format

The message contains information on the new recorded WAV file in a JSON encoded format:

- filename: string – Filename of the recorded WAV file.
- channel: int (1/2/3/4) – Microphone channel
- date: string [yyyy-MM-dd hh:mm:ss] – Recording data and time.
- samplerate: int [1/s] - Samplerate
- bits: int – Bits per Sample
- length: double [s] – Temporal length of the recorded WAV file.
- species: string ["No Call" or comma separated abbreviations of detected species] – Result of the post-processing AI Call Detector. Further information see chapter 8.

## 6.21 File Info of a new recorded WAV file on microphone channel 3

### 6.21.1 Send time

The message is sent each time a new WAV file was recorded on channel 3.

### 6.21.2 Topic

batmode/[MAC]/monitoring/new/fileinfo/ch3

### 6.21.3 Format

The message contains information on the new recorded WAV file in a JSON encoded format:

- filename: string – Filename of the recorded WAV file.

- `channel: int (1/2/3/4)` – Microphone channel
- `date: string [yyyy-MM-dd hh:mm:ss]` – Recording data and time.
- `samplerate: int [1/s]` - Samplerate
- `bits: int` – Bits per Sample
- `length: double [s]` – Temporal length of the recorded WAV file.
- `species: string ["No Call" or comma separated abbreviations of detected species]` – Result of the post-processing AI Call Detector. Further information see chapter 8.

## 6.22 File Info of a new recorded WAV file on microphone channel 4

### 6.22.1 Send time

The message is sent each time a new WAV file was recorded on channel 4.

### 6.22.2 Topic

`batmode/[MAC]/monitoring/new/fileinfo/ch4`

### 6.22.3 Format

The message contains information on the new recorded WAV file in a JSON encoded format:

- `filename: string` – Filename of the recorded WAV file.
- `channel: int (1/2/3/4)` – Microphone channel
- `date: string [yyyy-MM-dd hh:mm:ss]` – Recording data and time.
- `samplerate: int [1/s]` - Samplerate
- `bits: int` – Bits per Sample
- `length: double [s]` – Temporal length of the recorded WAV file.
- `species: string ["No Call" or comma separated abbreviations of detected species]` – Result of the post-processing AI Call Detector. Further information see chapter 8.

## 6.23 Transfer of a new recorded WAV file on microphone channel 1

### 6.23.1 Send time

The file is sent each time a new WAV file was recorded on channel 1.

### 6.23.2 Topic

`batmode/[MAC]/monitoring/new/file/ch1`

### 6.23.3 Format

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: WAV file as `byte array`.

## 6.24 Transfer of a new recorded WAV file on microphone channel 2

### 6.24.1 Send time

The file is sent each time a new WAV file was recorded on channel 2.

### 6.24.2 Topic

batmode/[MAC]/monitoring/new/file/ch2

### 6.24.3 Format

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: WAV file as `byte array`.

## 6.25 Transfer of a new recorded WAV file on microphone channel 3

### 6.25.1 Send time

The file is sent each time a new WAV file was recorded on channel 3.

### 6.25.2 Topic

batmode/[MAC]/monitoring/new/file/ch3

### 6.25.3 Format

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: WAV file as `byte array`.

## 6.26 Transfer of a new recorded WAV file on microphone channel 4

### 6.26.1 Send time

The file is sent each time a new WAV file was recorded on channel 4.

### 6.26.2 Topic

batmode/[MAC]/monitoring/new/file/ch4

### 6.26.3 Format

The message has the following format:

- Bytes [0...3]: Length of filename as `int`. Byte order is little Endian.
- Bytes [4...length+3]: Filename as `byte array`.
- Bytes [length+4...end]: WAV file as `byte array`.

## 6.27 New recorded Meteorological Data

### 6.27.1 Send time

The message is sent each time meteorological data is save to the file conditionlog.txt.

### 6.27.2 Topic

batmode/[MAC]/monitoring/new/meteoro

### 6.27.3 Format

The message contains the following information in a JSON encoded format:

- `date: string [yyyy-MM-dd hh:mm:ss]` – Timestamp of the recorded data. The time stamp always represents the beginning of a ten-minute averaging interval.
- `temp: double [°C]` - Mean temperature in the ten-minute interval.
- `prec: double [mm/min]` - Mean precipitation in the ten-minute interval.
- `wmo: string` – WMO code at the end of the ten-minute interval. Only valid for mode 3. XX depicts invalid data (see Chapter 6.9).

## 7 Last Will Message

The Last Will message is used to notify other clients about an ungracefully disconnected BATmode from the host. If the BATmode disconnects gracefully with a disconnect command, the host will not send the Last Will message.

### 7.1.1 Topic

batmode/[MAC]/disconnected

### 7.1.2 Format

The message contains the following `string`: BATmode disconnected

## 8 Post-Process AI Call Detector

As conventional bat call detectors based on measured parameters like start-frequency, curvature or duration often lacks in accuracy, a bat-friendly real-time wind turbine curtailment cannot be implemented with these methods. The AI Call Detector integrated in BATcontrol provides a much better accuracy and can be retrained on turbine specific data for an even higher accuracy. This allows to provide reliable data for a real-time curtailment via MQTT messages (see chapters 6.19, 6.20, 6.21 and 6.22). Further information on the AI Call Detector can be found in the manuals of the BATmode 2S/2S+ system or BATscreen PRO.

To send the results of the AI Call Detector via MQTT in real-time, a so call Real-Time License is mandatory. Please contact bat bioacousticstechnology GmbH for prices and quotation.

### 8.1 Results of the AI Call Detector

The result of the AI Call Detector is either a “No Call” for no detected bat calls in the analysed file or abbreviations of the detected bat species. All supported species and the respective abbreviations are depicted in Table 1: Species of AI Call Detector. If several different species were detected in a file, the three most probable species are returned as a comma separated string.

If no Real-Time License was detected, the species item in the Fileinfo messages (see chapters 6.19, 6.20, 6.21 and 6.22) is always “License Failure”. A deactivated AI Call Detector post process is depicted with “AI Call Detector Deactivated”.

**TABLE 1: SPECIES OF AI CALL DETECTOR**

Abbreviation	Possible Species
<b>Nlei</b>	<i>Nyctalus leisleri</i>
<b>Nnoc</b>	<i>Nyctalus noctula</i>
<b>Enil</b>	<i>Eptesicus nilssonii</i>
<b>Eser</b>	<i>Eptesicus serotinus</i>
<b>Vmur</b>	<i>Vespertilio murinus</i>
<b>Pnat</b>	<i>Pipistrellus nathusii</i>
<b>Ppip</b>	<i>Pipistrellus pipistrellus</i>
<b>Ppyg</b>	<i>Pipistrellus pygmaeus</i>
<b>Pkuh</b>	<i>Pipistrellus kuhlii</i>
<b>Myotis</b>	<i>Myotis bechsteinii</i> , <i>Myotis nattereri</i> , <i>Myotis daubentonii</i> , <i>Myotis mystacinus</i> , <i>Myotis myotis</i>
<b>Plecotus</b>	<i>Plecotus auratus</i> , <i>Plecotus austriacus</i>
<b>Bbar</b>	<i>Barbastella barbastellus</i>
<b>Hsav</b>	<i>Hypsugo savii</i>
<b>Tten</b>	<i>Tadarida teniotis</i>