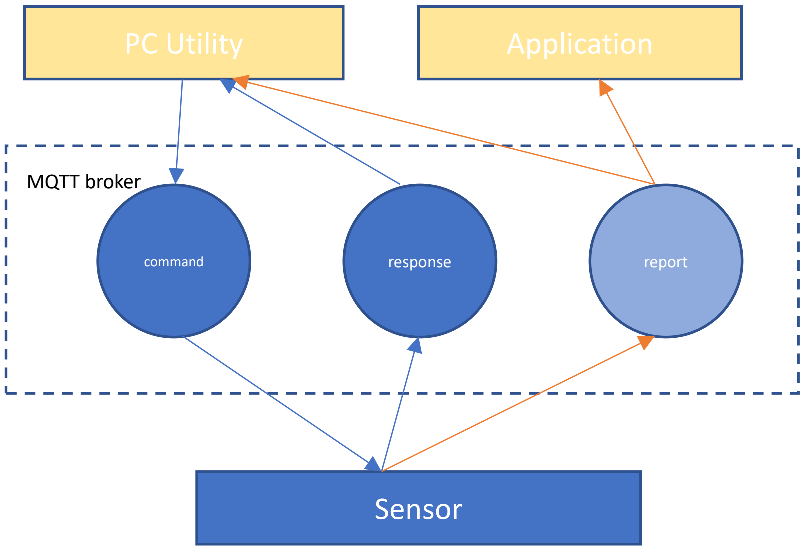
AIS Sensor Message Exchange Flow 1.4

Message Exchange Architecture



Revision History

Version	Date	Change Description
v1.4	2025-06-26	1. Added Report Type (2-Feature) 2. Command Type (0x03-Set Schedule Settings): Added setting for Feature Mode 3. Command Type (0x09-Check Online): Description revised 4. Command Type (0x01-Sensor Information): Added TCP Address and TCP Port info 5. Added battery voltage to level mapping table 6. Removed "Raw data + FFT / OA" recording mode

Version	Date	Change Description
v1.3	2025-03-06	1. Enhanced section 1.2 explanations and added Raw Data & FFT Data examples 2. Described Raw Data and FFT Data length 3. Added voltage value conversion formula 4. Indicated Big-Endian or Little-Endian format for values 5. Report Type (4-Hibernate/Wakeup): Added sensor sleep/wake info 6. Command Type (0x01-Sensor Information): Added MqttAddress & MqttPassword 7. Command Type (0x06-Set RTC): Added GMTOffset field
v1.2	2024-10-18	1. Added Ask Command in Report Type for offline command collection 2. Supplemented formula for temperature and raw data value conversion 3. Modified OA/FFT packet length 4. Added battery voltage value in response 5. Added new Command Types: 0x07 Set Sensor Sleep Now 0x08 Set Sensor Receive Command Mode 0x09 Check Online

Version	Date	Change Description
v1.1	2024-06-07	1. Changed FFT to single packet format 2. Added Report Types 7~10: 7: Raw data + FFT 8: Real Time Raw data + FFT 9: OA Only 10: Real Time OA Only 3. Get Sensor Information (Command ID = 0x01): Added MAC Address 4. Get Sensor Schedule Information (Command ID = 0x02): Added two new modes 5. Set Schedule Settings (Command ID = 0x03): Added two new modes

Message

1. Report

General Format

Type	Data Length(n)	Data
1 Byte	4 Bytes	n Bytes

- **Topic**
  Subscribe *<sensor id>/report*
- The report messages are unsolicited from sensor and contains different types of report. Report messages are of the format above. Here is the description and specification of each field below.
- **Type**
  Integer type to denote the report type of following data
  There are following possible report types:

0 – Raw data

1 – FFT

2 – Feature

3 – Battery

4 – Hibernate/Wakeup

5 – Real Time Raw Data

6 – Real Time FFT

71, 72 – Raw data + FFT

81, 82 – Real Time Raw data + FFT

9 – OA Only

10 – Real Time OA Only

11 – Ask Command

12~255 – Reserved

- **Data Length**

  Integer type to denote how many bytes are there in **Data** field.

- **Data**

  The data type and format of **Data** field is specific to each report type

1.2. Report Details

- **Raw Data(Type = 0)**
  - Raw data report is of the following format in **Data** field.
    - The integers behind the field name denotes bytes it takes.

Header (20B)								
Timestamp (8B)	***Control Flags*** (1B)	****Index*** (1B)	****Total*** (1B)	***Temp*** (2B)	***Real ODR*** (2B)	***Battery information*** (1B)	***Last ADC*** (2B)	***Average ADC*** (2B)
Acceleration Data (3k*2), k=28000*recording seconds								
x_1(2B)	y_1(2B)	z_1(2B)	x_2(2B)	y_2(2B)	z_2(2B)	…	…	…

  - The data length ***n*** should be ***3k*2+25(header size)***
    - ***k*** is the number of data points, currently set by AISSENS as 28000 * recording seconds.
  - **Timestamp** describes the UNIX time that the raw data captured. The timestamp can be treated as the unique ID for raw data report. If a raw data report is divided into multiple packets, they are of the same timestamp. Timestamp does not

include a time zone and in big-endian format.

- **Control flags**

  The purpose for each control flag is as below

7	6	5	4	3	2	1	0
reserved							record fail

  The 0th bit of **Control flags** field is record fail flag. It means the raw data is not record completed if it is set to **1**.

- **Index** value is 1, currently not in use.
- **Total** value is 1, currently not in use.
- **Temp** is short integer type and this field is current temperature of AISSENS in Celsius. The received data is 2 bytes in big-endian format, converted to a float, and then calculated using the following formula:

  temperature = {float value}/256.0+28

- **Real ODR** is a short integer ranges from 3000 to 30000 which can vary according to temperature, sensor or IC vendor.
- **Battery Information** field is of integer type and contains current battery level and corresponding percentage can be found in the table below.

Battery Level	**Battery Percentage**
0	0%~5%
1	5%~20%
2	20%~35%
3	35%~50%
4	50%~100%

- **ADC**

  There are two fields for ADC value. They are both of short integer type and stand for current battery voltage and average battery voltage respectively, and then calculated using the following formula:

  Battery Voltage = ({ADC}-1400)*0.001547+2.7

Battery Voltage	**Battery Level**
≥ 3.3V	High
3.3V ~ 3.15V	Medium
< 3.15V	Low

- ***x, y*** and ***z*** are raw data captured in order.
  AISSENS returns a data length of 6 bytes for each single-point measurement. These 6 bytes represent the measurement data for X, Y, and Z, with bytes 0 and 1 representing X, bytes 2 and 3 representing Y, and bytes 4 and 5 representing Z. The conversion method is as follows:
  X: (Byte1 << 8 | Byte0) x 0.0002441062
  Y: (Byte3 << 8 | Byte2) x 0.0002441062
  Z: (Byte5 << 8 | Byte4) x 0.0002441062
- **Example**
  The sensor will publish data to the report topic <sensor id>/report. Below is an example explanation of receiving a raw data entry.
  The hexadecimal representation shows the received raw data, and Value is the converted value.

Header (13B)		
Type	**Data Length(n)**	***Timestamp***
0x00	0x00 0x05 0x20 0x99 Value: 336025	0x00 0x00 0x00 0x00 0x67 0xc5 0x83 0x4b Value: 1740997451

Header (12B)							
Control Flags	****Index***	****Total***	***Temp***	***Real ODR***	***Battery information***	***Last ADC***	***Average ADC***
0x00	0x01	0x01	0xfd 0xed Value: -531 Temperature: 25.92	0x68 0x3d ODR: 26685	0x04	0x07 0x46 ADC: 1862 Voltage: 3.41	0x07 0x3c ADC: 1852 Voltage: 3.39

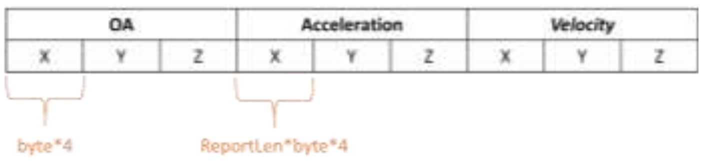
Acceleration Data (3k*2), k=28000*recording seconds						
x_1(2B)	y_1(2B)	z_1(2B)	x_2(2B)	y_2(2B)	z_2(2B)	
0x5b 0x00 Value: 91 Acc: 0.022	0x74 0xff Value: -140 Acc: -0.034	0xd8 0x10 Value: 4312 Acc: 1.052	0x79 0x00 Value: 121 Acc: 0.029	0x28 0xff Value: -216 Acc: -0.052	0x6b 0x10 Value: 4203 Acc: 1.025	...

- **FFT(Type = 1)**

  FFT report is of the following format in **Data** field.

Header (45B)										
Timestamp (8B)	***Status*** (1B)	***Battery information*** (1B)	***Average ADC*** (2B)	***Last ADC*** (2B)	***Temp*** (2B)	***OA*** (3*4B)	***Frequency Resolution*** (4B)	***FFT Length*** (4B)	***ReportLen*** (4B)	***Reserved*** (5B)
FFT Data										
Acceleration (4B) * ReportLen * 3 axis					***Velocity*** (4B) * ReportLen * 3 axis					

- The data length ***n*** should be 3*ReportLen*4*2+50(header size)
- **Timestamp** describes the UNIX time that the raw data captured. The timestamp can be treated as the unique ID for raw data report. If a raw data report is divided into multiple packets, they are of the same timestamp. Timestamp does not include a time zone and in big-endian format.
- **Status** is the FFT/OA calculation result.
- **Battery information** is same as Raw Data definition.
- **ADC** is same as Raw Data definition.
- **Temp** is same as Raw Data definition.
- **OA** *is a float value* according to ISO-10816-3**,** contains x, y, z axis data, in little-endian format.
- **Frequency Resolution** is dynamic according to the data length in the arguments, in little-endian format.
- **FFT Length** is the length for Velocity LINEAR spectrum and Acceleration LINEAR spectrum, in big-endian format.
- **ReportLen** is the length for single axis data volume, in big-endian format.



- **Acceleration** LINEAR spectrum (float value, little-endian)
  - Frequency resolution is DYNAMIC according to the data length in the augments
  - Bandwidth is fixed at SR/2

    - Represent amplitude in g,rms

- **Velocity** LINEAR spectrum (float value, little-endian)
    - Frequency resolution is DYNAMIC according to the data length in the augments
    - Bandwidth is fixed at SR/2
    - Represent amplitude in mm/s,rms
- **Example**

  The sensor will publish data to the report topic <sensor id>/report. Below is an example explanation of receiving a FFT data entry.

  The hexadecimal representation shows the received FFT data, and Value is the converted value.

Header (45B)							
Type 0x01	**Data Length(n)** 0x00 0x04 0x0c 0xb2 Value: 265394	***Timestamp*** 0x00 0x00 0x00 0x00 0x67 0xc0 0x3a 0x7f Value: 1740651135	***Status*** (1B*)* 0x00	***Battery information*** (1B) 0x04	***Average ADC*** (2B) 0x07 0x36 ADC: 1846 Voltage: 3.38	***Last ADC*** (2B) 0x07 0x12 ADC: 1810 Voltage: 3.33	***Temp*** (2B) 0xfd 0x65 Value: -667 Temperature: 25.39

Header (45B)						
X OA (4B) 0x1b 0x23 0x79 0x3d Value: 0.060	***Y OA*** (4B) 0x7f 0x82 0x59 0x3d Value: 0.053	***Y OA*** (4B) 0x8c 0xe2 0xa0 0x3d Value: 0.078	***Frequency Resolution*** (4B) 0x00 0xf0 0x0a 0x3f Value: 0.542	***FFT Length*** (4B) 0x00 0x00 0x60 0x00 Value: 24576	***ReportLen*** (4B) 0x00 0x00 0x2b 0x30 Value: 11056	***Reserved*** (5B)

FFT Data					
X Acceleration (4B)* ReportLen …	***Y Acceleration*** (4B)* ReportLen …	***Z Acceleration*** (4B)* ReportLen …	***X Velocity*** (4B)* ReportLen …	***Y Velocity*** (4B)* ReportLen …	***Z Velocity*** (4B)* ReportLen …

Feature(Type = 2)

- Feature report is of the following format in **Data** field.

Timestamp(8B)	***Feature Fields***(json)

- **Timestamp**
  describes the UNIX time that the raw data captured. The timestamp can be treated as the unique ID for raw data report. If a raw data report is divided into multiple packets, they are of the same timestamp. Timestamp does not include a time zone and in big-endian format.
- **Feature Fields**
  The **Feature Fields** The include the Temperature, BatVoltage, Mean, Standard Deviation, Skewness, Kurtosis, Median, Crest Factor, RMS, 0 to Peak, and Peak to Peak of three-axis acceleration.
- The JSON example is as follows:

```
{
    "Temperature": "27.2",
    "BatVoltage":3.34,
    "x_acc_rms": 102.7775,
    "x_acc_mean": 250.4765,
    "x_acc_std_dev": 102.7775,
    "x_acc_p2p": 907.2144,
    "x_acc_skewness": 0.010195,
    "x_acc_kurtosis": -0.020863,
    "x_acc_crest_factor": 4.463322,
    "x_acc_zero2peak": 453.6072,
    "x_acc_median": -1.531128,
    "y_acc_rms": 100.9656,
    "y_acc_mean": -487.9419,
    "y_acc_std_dev": 100.9656,
    "y_acc_p2p": 904.8207,
    "y_acc_skewness": -0.012661,
    "y_acc_kurtosis": -0.021248,
    "y_acc_crest_factor": 4.856464,
    "y_acc_zero2peak": 452.4103,
        "y_acc_median": -0.373993,
        "z_acc_rms": 113.8679,
        "z_acc_mean": 10179.79,
        "z_acc_std_dev": 113.8679,
        "z_acc_p2p": 935.9395,
        "z_acc_skewness": -0.003698,
        "z_acc_kurtosis": -0.031481,
        "z_acc_crest_factor": 4.399148,
        "z_acc_zero2peak": 467.9697,
        "z_acc_median": 0.638672
    }
```

Battery(Type = 3)

- Battery report is of the following format in **Data** field

Timestamp(8B)	*Battery information*(1B)	*Last ADC*(2B)	*Average ADC*(2B)

  - The integers behind the field name denotes bytes it takes.

- The data length should be 9 bytes
- **Timestamp**

  Timestamp describes the UNIX time in microsecond that the battery level captured
- **Battery Information**

  Battery information field is of **integer type** and contains current battery level and corresponding percentage can be found in the table below.

Battery Level	Battery Percentage
0	0%~5%
1	5%~20%
2	20%~35%
3	35%~50%
4	50%~100%

- **ADC**

  There are two fields for ADC value. They are both of **integer type** and stand for current battery voltage and average battery voltage respectively.

- **Hibernate/Wakeup(Type = 4)**
  - The Hibernate/Wakeup Report is used to notify online and offline status, along with additional information.
    Assume that Status is **Hibernate**, then follow the data format below.

Timestamp(8B)	***Status***(1B)	***Sensor Information***(json)

Assume that Status is **Wakeup**, then follow the data format below.

Timestamp (8B)	***Status*** (1B)	***Online Duration***(2B)	***Wi-Fi Online Duration***(2B)	***Transmission Duration***(2B)	***Battery Usage Time***(4B)

  - **Timestamp**
    Timestamp describes the UNIX time in microsecond that the hibernating status changed.
  - **Status**
    Status field is of one Byte **integer type.**
    - **0** – Manual Hibernated
    - **1** – Manual Wakeup
    - **2** – Schedule Hibernated
    - **3** – Schedule Wakeup
  - **Sensor Information**
    Sensor information in JSON string in ASCII code.
  - **Online Duration**
    This online duration in seconds.
  - **Wi-Fi Online Duration**
    This Wi-Fi online duration in seconds.
  - **Transmission Duration**
    Transmission duration in seconds.
  - **Battery Usage Time**
    Cumulative battery usage time in seconds.
- **Real Time Raw Data Report(Type = 5)**
  - Real time raw data report comes up only when real time raw data recording command**(0x05)** fired with **mode set to 0. The data format for this report type is exactly**

same as raw data report(report type = 0).

- **Real Time FFT Mode Report(Type = 6)**
  - Real time FFT report comes up only when real time FFT recording command**(0x06)** fired with **mode set to 1. The data format for this report type is exactly same as FFT report(report type = 1).**
- **RAW Data with FFT Report(Type = 71, 72)**
  - RAW Data with FFT report comes up only when raw data with FFT recording command**(0x071) (0x072)** fired with **mode set to 2. The data format for this report type is exactly same as raw data report(report type = 0) and FFT report(report type = 1). The packet will be sent twice, the first packet is raw data(0x071), the second packet is FFT(0x072).**
- **Real Time RAW Data with FFT Report (Type = 81, 82)**
  - Real time RAW Data with FFT report comes up only when raw data with FFT recording command**(0x081) (0x082)** fired with **mode set to 2. The data format for this report type is exactly same as raw data report(report type = 0) and FFT report(report type = 1). The packet will be sent twice, the first packet is raw data, the second packet is FFT.**
- **OA Only Report(Type = 9)**
  - ***OA Only report is of the following format in Data field.***

Timestamp (8B)	*Status* (1B)	*Battery information* (1B)	*Average ADC* (2B)	*Last ADC* (2B)	*Temp* (2B)	*OA* (4B*3)	*Reserved* (17B)

  - *The data length n should be 50.*
  - The data format for this report type is exactly same as FFT report(report type = 1), but without data after OA.
- **Real Time OA Only Mode Report(Type = 10)**
  - Real time OA Only report comes up only when OA Only recording command(0x10) fired with mode set to 3. The data format for this report type is exactly same as OA Only report(report type = 9).
- **Ask Command(Type = 11)**

- Ask Command report is schedule receive command mode send message to check any command need to execute.

Reserved(Type = 12~255)
TBD

2. Command/Response format

2.1. General Format

Command Format

Serial Number	Command ID	Data Length	Parameters
2 Bytes	1 Byte	4 Bytes	n Bytes

- **Serial Number**
  Integer type to denote the order of commands issued. This number is increasing by 1 after fire a command.
- **Command ID**
  Integer type to identify which type of command to be executed. Since there is only one byte for **Command ID**, there are 256 types command at most(0x00-0xFF). There are possible commands list as following:

Command ID	Description
0x00	Get API version
0x01	Get Sensor Information
0x02	Get Sensor Schedule Information
0x03	Set Schedule Settings
0x04	Start/Stop Scheduled Reporting
0x05	Real Time Recording
0x06	Set RTC
0x07	Set Sensor Sleep Now
0x08	Set Sensor Receive Command Mode
0x09	Check Online

- **Data Length**
  Integer type to denote how many bytes are there in the following **Parameters** field
- **Parameters**
  The data type and format of **Parameters** field is specific to each type of command.

Response Format

Serial Number	Command ID	Status Code	Data Length	Response Data
2 Bytes	1 Byte	1 Byte	4 Bytes	n Bytes

- **Serial Number**
  Integer type to denote the order of commands issued. This number can be used to trace back to exact which command this response belongs to.
- **Command ID**
  Integer type to identify which type of command was executed
- **Status Code**
  Integer type to identify the status of command

Status Code	Description
0x00	Success
0x01	Unknown command ID

- **Data Length**
  Integer type to denote how many bytes are there in the following **Response Data** field
- **Response Data**
  The data type and format of **Response Data** field is specific to each type of command.

Command and Response Topics

- **Command**
  Publish command to topic ***<sensor id>/command***
- **Response**
  Subscribe topic ***<sensor id>/response*** for command response.

2.2. Command Details

Get API Version(Command ID = 0x00)

- **Parameters**
  This command takes no parameter

- **Response**

  If failed, the status code is set and Data Length is set to 0.
  If succeeded, the response contains success status code(0x00) and with response data is API version string in **ASCII code**. The **Data Length** field is the version string length.
  For example, if the version string is "1.0", the **Data Length** is set to 3 and **Response Data** is set as below

0x31 '1'	**0x2E** '.'	**0x30** '0'

- **Example**

  If user is wondering the API version, the first create the binary payload as follow and publish to topic
  <sensor id>/command

Serial Number	**Command ID**	**Data Length**	**Parameters**
0x00 0x23	**0x00**	**0x00 0x00 0x00 0x00**	**--**

After sensor receives the command from command topic ***<sensor id>/command***, the command is then executed and the response is sent back to response topic ***<sensor id>/response***. The response contains version string 1.0 as follows

Serial Number	**Command ID**	**Status Code**	**Data Length**	**Response Data**
0x00 0x23	**0x00**	**0x00**	**0x00 0x00 0x00 0x03**	**0x31 0x2E 0x30**

- **Get Sensor Information(Command ID = 0x01)**
  - **Parameter**

    This command takes no parameter
  - **Response**

    If failed, the status code is set and Data Length is set to 0. If succeeded, the response contains success status code(0x00) and with response data is sensor information in JSON string in **ASCII code**. The **Data Length** field is set to the length of Sensor Information. Here are detailed explanation for the sensor information.
    - **FirmwareVersion**

      String type The version of firmware runs in AISSENS
    - **Brand**

      String type Sensor brand name
    - **Model**

      String type AISSENS model name
    - **Bandwidth**

      String type or Float type
    - **SamplingRate**

      String type or Float type Current sampling rate settings
    - **GValue**

      String type
    - **SsidPrim**

      String type Wi-Fi SSID connected to
    - **LocalIp**

      String type Wi-Fi IP address
    - **SingalStrength**

      Integer type Wi-Fi signal strength level. The higher this value is the better signal it has.
    - **BatteryLevel** is 1 Byte Integer which contains current battery level and corresponding percentage can be found in the table below.

Battery Level	Battery Percentage
0	0%~5%
1	5%~20%
2	20%~35%

3	35%~50%
4	50%~100%

- **MACAddress**
  String type MAC Address
- ***Temperature***
  Float type and this field is current temperature of AISSENS in Celsius.
- ***EnSchRecCMD***
  Integer type endable schedule receive command mode.
- ***BatVoltage***
  Float type Battery voltage.

Battery Voltage	Battery Level
≥ 3.3V	High
3.3V ~ 3.15V	Medium
< 3.15V	Low

- ***MqttAddress***
  String type MQTT Broker IP connected by AISSENS.
- ***MqttPassword***
  String type MQTT Broker password connected by AISSENS.
- ***TcpAddress***
  String type TCP Server IP connected by AISSENS.
- ***TcpPort***
  Integer type TCP Server port connected by AISSENS.

Example

The following message is published to topic ***<sensor id>/command*** to query sensor information.

Serial Number	Command ID	Data Length	Parameters
0x00 0x23	**0x01**	**0x00 0x00 0x00 0x00**	**--**

After sensor receives the command from command topic ***<sensor id>/command*,** the command is then executed and the response is sent back to response topic ***<sensor id>/response***. The response contains sensor information JSON string as follows

Serial Number	**Command ID**	**Status Code**	**Data Length**	**Response Data**
0x00 0x23	**0x01**	**0x00**	**0x00 0x00 0x00 0xF2**	**0x7B 0x22 0x46 …**

```
{
    "FirmwareVersion":"TW-AISSENS_100AW6K-0.00.11-T3-
user",
    "Brand":"ASUS",
    "Model":"AISSENS100AW",
    "Bandwidth":"6KHz",
    "SamplingRate":"26.7KHz",
    "GValue":"8g",
    "SsidPrim":"oppo",
    "LocalIp":"192.168.138.165",
    "SignalStrength":4,
    "BatteryLevel": 3,
    "MACAddress": "39:C7:F7:C7:51:10",
    "Temperature": "27.2",
    "EnSchRecCMD": 0,
    "BatVoltage":3.34,
    "MqttAddress": "192.168.138.77",
    "MqttPassword":"123456",
    "TcpAddress": "192.168.138.77",
    "TcpPort":1235
}
```

- **Get Sensor Schedule Information(Command ID = 0x02)**
  - **Parameter**
    This command takes no parameter
  - **Response**
    If failed, the status code is set and Data Length is set to 0.
    If succeeded, the response contains success status code(0x00) and with response data with the following format:

Start timestamp	End timestamp	Weekly Schedule	Duration	Interval	Mode	Status
8 Bytes	8 Bytes	1 Byte	2 Bytes	4 Bytes	1 Byte	1 Byte

Start and end timestamp are both 8 Bytes UNIX timestamp to define the start and end time of schedule in microsecond. If **Start/End timestamp** is set to 0, the schedule has no start/end time.
Weekly schedule stands for enable status of each day.

7	6	5	4	3	2	1	0
0	Sun	Sat	Fri	Thu	Wed	Tue	Mon

For example, if Monday and Thursday are enabled. The cycle field will be 0x09
Duration is a 2 Bytes Integer defines how long the sampling is in seconds.
Interval is 4 Bytes Integer defines length in seconds between two schedules
Mode is 1 Byte Integer defines target report type

- **0 – Raw data**
- **1 – FFT/OA**
- **3 – OA Only**
- **4 – Feature**

Note: Option 2 (Raw data + FFT / OA) has been removed in this version.

Status is 1 Byte defines schedule enable status. It means schedule enabled if this field is non-zero and disabled otherwise.

Set Schedule Settings (Command ID = 0x03)

Start timestamp	End timestamp	Weekly Schedule	Duration	Interval	Mode
8 Bytes	8 Bytes	1 Byte	2 Bytes	2 Bytes	1 Byte

- **Parameter**

  The command contains the following parameters.
  Start and end timestamp are both 8 Bytes UNIX timestamp in microseconds to define the period of schedule.
  Weekly schedule stands for enable status of each day.

7	6	5	4	3	2	1	0
0	Sun	Sat	Fri	Thu	Wed	Tue	Mon

  Duration is a 2 Bytes Integer describes how long to record in seconds.
  Interval is 4 Bytes Integer defines length in seconds between two schedules
  Mode is 1 Byte Integer defines target report type

  - **0 – Raw data**
  - **1 – FFT / OA**
  - **3 – OA Only**
  - **4 – Feature**

  Note: Option 2 (Raw data + FFT / OA) has been removed in this version.

- **Response**

  If failed, the status code is set and Data Length is set to 0.
  If succeeded, the response contains success status code(0x00)

Start/Stop Scheduled Report (Command ID = 0x04)

- **Parameter**

  This command takes one Byte as its parameter. If the value of this Byte is **0**, then raw data report schedule will be turned **OFF**. On the contrary, if it is **non-zero**, the raw data report will be turn **ON**.

- **Response**

  If failed, the status code is set and Data Length is set to 0.
  If succeeded, the response contains success status

code(0x00) and with no response data provided.

- **Real Time Recording (Command ID = 0x05)**
  - This command provides a one-shot real time recording according to parameters provided. After this command is received, a response will be sent to inform requester that the command is received and the recording is going to be performed. **The actual recorded data will be published through report topic with report type 5 and 6.**
  - **Parameter**

    This command has two parameters as follow:

Duration	**Mode**
2 Bytes	1 Byte

    Duration is a 2 Bytes integer describes how long to record in seconds.

    Mode is 1 Byte Integer defines target report type

    - **0 – Raw data**
    - **1 – FFT / OA**
  - **Response**

    If failed, the status code is set and Data Length is set to 0.

    If succeeded, the response contains success status code(0x00) and with no response data provided.

    The actual recorded report will be published through report topic with report type 5 and 6.

- **Set RTC (Command ID = 0x06)**
  - **Parameter**

    This command has one parameters as follow:

Timestamp	**GMTOffset**
8 Bytes	4 Bytes

    Timestamp is a long integer timestamp used to set to RTC.

    GMTOffset is a integer the time difference in seconds between a location's local time and GMT (Greenwich Mean Time).
  - **Response**

    If failed, the status code is set and Data Length is set to 0.

If succeeded, the response contains success status code(0x00) and with no response data provided.

- **Set Sensor Sleep Now (Command ID = 0x07)**
  - **Parameter**

    This command takes no parameter.

- **Set Sensor Receive Command Mode (Command ID = 0x08)**
  - **Parameter**

    This command takes one Byte as its parameter. If the value of this Byte is 0, then sensor receive command mode will be turned OFF. On the contrary, if it is non-zero, the sensor receive command mode will be turn ON.
  - **Response**

    If failed, the status code is set and Data Length is set to 0.
    If succeeded, the response contains success status code(0x00) and with no response data provided.

- **Check Online (Command ID = 0x09)**
  - **Parameter**

    This command takes no parameter.
  - **Response**

    If the AIS Sensor is disconnected from the MQTT broker, a reply will not be received.
    If the AIS Sensor is online, the response will include a success status code (0x00) and will not provide any response data.

Appendix

1. Floating number

According to different sensor IC, the float type can have different representation in binary format. There are two possible candidates:

- **ADI**

  The valid bits are 12 least significant bits ADI floating number. The nominal intercept is 1885 LSB at 25°C and the nominal slope is **−9.05**

LSB/°C

- **ST**
  The valid bits are 16 least significant bits ST floating number. It is represented as a number of 16bits in two's complement format with sensitivity of 256LSB/°C. The output zero level corresponds to 25°C.