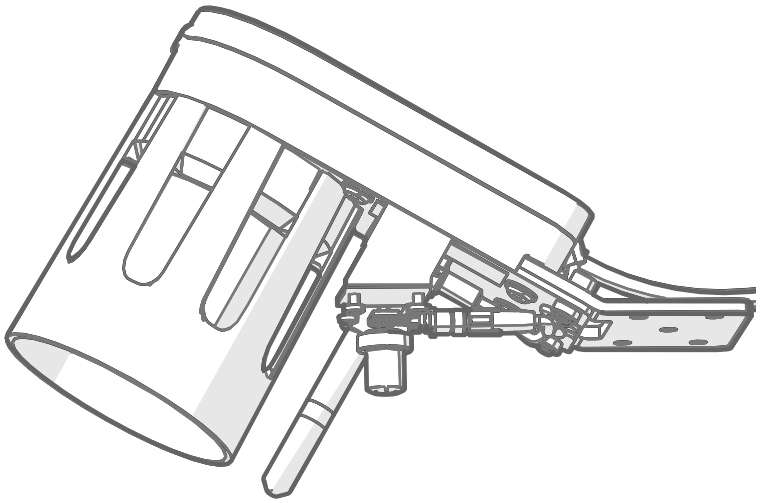
M212201EN-D

Interface Description

Vaisala Mobile Detector

MD30



VAISALA

PUBLISHED BY

Vaisala Oyj
Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland
P.O. Box 26, FI-00421 Helsinki, Finland
+358 9 8949 1

Visit our Internet pages at www.vaisala.com.

Table of contents

List of tables

1. About this document

1.1 Version information

This document provides a description of the data reporting interface of Vaisala Mobile Detector MD30.

Table 1 Document versions (English)

Document code	Date	Description
M212201EN-D	October 2020	For version D of the interface. • Added dual port capability (use of interface on multiple ports) • Added status information bits 15, 16, and 17 • Updated maximum reported range of layer thickness values • Clarified difference between dew point and frost point temperature
M212201EN-C	June 2020	For version C of the interface. • Updated surface state value tables • Clarified instructions about SET REFERENCES message • Added calibration overview • Added detailed MD30 data message description
M212201EN-B	August 2019	For version C of the interface.

1.2 Related manuals

Table 2 MD30 manuals

Document code	Name
M212339EN	*Vaisala Mobile Detector MD30 Product and Package Description Reference Guide*
M212169EN	*Vaisala Mobile Detector MD30 Setup Guide*
M212519EN	*Mounting MD30 on Towing Hook Installation Guide*
M212520EN	*Mounting MD30 on Trailer Hitch Installation Guide*
M212432EN	*Mounting Temperature Sensors Separately Installation Guide*
M212309EN	*Vaisala Mobile Detector MD30 Maintenance and Troubleshooting Reference Guide*
M212537EN	*Updating MD30 Software Quick Guide*
M212201EN	*Vaisala Mobile Detector MD30 Interface Description*
M212384EN	*Vaisala Mobile Detector MD30 Interface Client Technical Reference*
M212382EN	*Vaisala Mobile Detector MD30 Interface Client License Agreement*
M212310EN	*Vaisala Mobile Detector MD30 OSS Licenses*

For MD30 instructions, including documents and videos, see https://www.vaisala.com/en/support/md30.

1.3 Documentation conventions

WARNING! Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.

CAUTION! Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.

Note highlights important information on using the product.

Tip gives information for using the product more efficiently.

Lists tools needed to perform the task.

Indicates that you need to take some notes during the task.

1.4 Trademarks

Vaisala® is a registered trademark of Vaisala Oyj.

Python® is a trademark or registered trademark of the Python Software Foundation.

All other product or company names that may be mentioned in this publication are trade names, trademarks, or registered trademarks of their respective owners.

2. Vaisala Mobile Detector MD30 interface

2.1 Measurement system overview

The mobile sensor provides:

- Surface state information, including:
  - Grip
  - Surface state
  - Surface layer thicknesses
- Surface temperature
- Air temperature
- Dew point and frost point
- Relative humidity

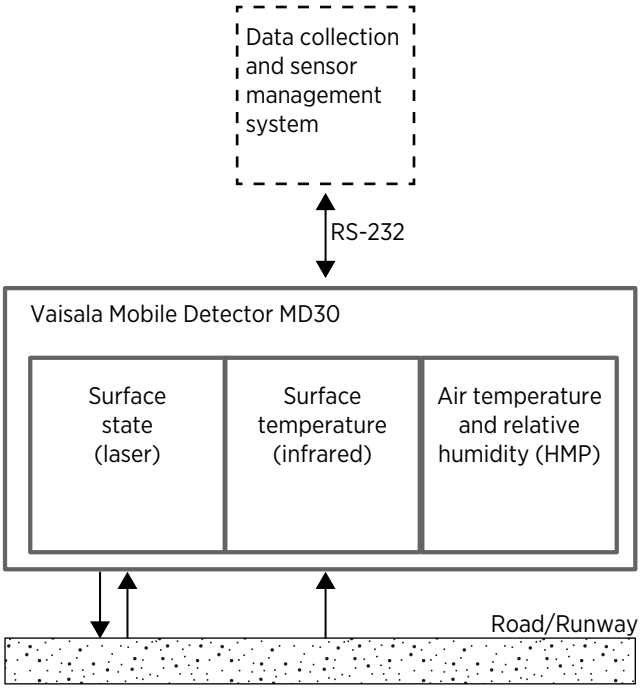


Figure 1 System overview

The data collection and sensor management system can be a spreader system in a winter maintenance vehicle (for example, snow plow truck), a road monitoring system in a vehicle (for example, patrol car), or any other data collection system.

The data collection system maps the mobile road sensor data to a geographic location.

The communication uses the RS-232 interface.

2.1.1 Measurement sensors in MD30

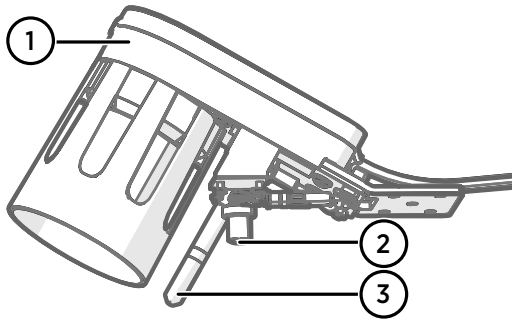


Figure 2 MD30 measurement sensors

1 Surface state sensor
2 Surface temperature sensor, MT10
3 Air temperature and humidity sensor, HMP113

MD30 contains 3 sensors:

- Surface state sensor measures the water, ice, and snow layer thicknesses and surface state with 3 lasers.
- Surface temperature sensor MT10 measures the surface temperature with infrared (IR) technology.
- Air temperature and humidity sensor, Vaisala HUMICAP® Humidity and Temperature Probe HMP113, measures the air temperature and relative humidity.

2.2 Interface overview

The mobile sensor unit interface applies a request-response pattern. Every message is acknowledged by the mobile sensor.

Table 3 Interface functions

Request	Message
Identify the installed unit.	`GET UNIT ID` `GET FULL PRODUCT INFO`
Report status and error information.	`GET UNIT STATUS` `SEND DATA`

Initiate road condition, temperature, and humidity data reporting continuously at defined interval.	`SEND DATA`
	`GET PARAMETER`
	`SET PARAMETER`

Stop continuous data reporting.	`SEND DATA`
	`GET PARAMETER`
	`SET PARAMETER`

Report road condition, temperature, and humidity data using polling.	`SEND DATA`

Request	Message
Set reference values.	SET REFERENCES SET ROAD COEFFICIENTS GET PARAMETER SET PARAMETER STOP REFERENCE SETTING
Set offset to road and air temperature.	GET PARAMETER SET PARAMETER
Configure the serial port speed.	GET PARAMETER SET PARAMETER
Configure units of reported quantities.	GET PARAMETER SET PARAMETER

More information

- GET UNIT ID message (page 27)
- GET FULL PRODUCT INFO message (page 30)
- GET UNIT STATUS message (page 34)
- SEND DATA message (page 36)
- SET REFERENCES message (page 41)
- SET ROAD COEFFICIENTS message (page 49)
- STOP REFERENCE SETTING message (page 47)
- GET PARAMETER message (page 52)
- SET PARAMETER message (page 55)
- RESTART UNIT message (page 58)

2.2.1 Communication settings

The communication uses RS-232 where sensor data is transmitted in binary format. The default settings are:

- 115200 bps
- 8 data bits
- 1 stop bit
- No parity
- No hardware control

2.2.2 Supported data formats

Data is transmitted in binary format. The following data types are supported.

Table 4 Supported data formats

Data type	Description
f32	Binary 32 IEEE 754 format floating point number

Data type	Description
U64/U32/24/16/8	64/32/24/16/8 bit unsigned integer
S32/24/16/8	32/24/16/8 bit signed integer
ASCII	ASCII characters are transmitted as binary values

The byte order is little endian. For example, 4-byte value 0x12345678 is transmitted in the following order: 0x78, 0x56, 0x34, and 0x12.

Bit numbering starts at zero (0) for the last significant bit (LSB).

2.2.3 Use of interface on multiple ports

There can be 2 data collection systems connected to 1 mobile road sensor:

- System that shows the relevant information on a display in the vehicle.
- System that collects the data to a database or cloud.

To use multiple ports, you must have a 7-pin version of the 8-m extension cable that supports dual port use (order code MD30DUALSET).

When using multiple ports, note the following:

- Both ports have the same default settings.
- Some of the parameters are device specific and some port specific.
- Reference setting (calibration):
  - Can be initiated from any port.
  - Cannot be started if there is reference setting ongoing, regardless of the port.
  - Can be stopped from any port, regardless of the port from which it was started.

More information

- Parameters (page 62)
- Surface state and temperature calibration (page 18)

3. Observations

3.1 Data status warnings

The following bits are set to warn the user that data quality cannot be fully guaranteed and data should be treated with suspicion. The data status warnings are reported in the `SEND DATA` message.

Table 5 Data status warnings

Bit	Description
0	Air temperature
1	Relative humidity
2	Dew point temperature
3	Frost point temperature
4	Surface temperature
5	Surface state according to Vaisala classification
6	Surface state according to the EN 15518 standard
7	Grip
8	Water layer thickness
9	Ice layer thickness
10	Snow layer thickness
11 ... 15	Reserved for future use

More information

- SEND DATA message (page 36)

3.2 Data status erroneous

The following bits are set if data contains errors. The erroneous data status is reported in the `SEND DATA` message.

Table 6 Data status erroneous

Bit	Description
0	Air temperature
1	Relative humidity
2	Dew point temperature
3	Frost point temperature

Bit	Description
4	Surface temperature
5	Surface state according to Vaisala classification
6	Surface state according to the EN 15518 standard
7	Grip
8	Water layer thickness
9	Ice layer thickness
10	Snow layer thickness
11 ... 15	Reserved for future use

More information

- SEND DATA message (page 36)

3.3 Status information

The status information is used to indicate the current status of the mobile sensor and to issue warnings. The status information is reported in the `GET UNIT STATUS`, `SEND DATA`, and `SET REFERENCES` messages.

Table 7 Status information

Bit [1)]	Message	Value	Description
0	Not ready to measure	`0` = Ready `1` = Not ready	Unit is starting up. When unit reaches full operational status, the flag is cleared. Measurement data may be invalid. If the condition persists, check error bits.
1	Reference setting ongoing	`0` = Not ongoing `1` = Ongoing	If reference setting does not start, check status information bits 10 ... 13 and error bits.
2	Laser temperature change in progress	`0` = Not ongoing `1` = Ongoing	Unit operational, but measurement data may be invalid. Wait for laser temperature change to finish. If the condition persists, check error bits.
3	Window contamination warning [2)]	`0` = OK `1` = Contaminated	Window is getting contaminated. Clean the window.
4	Window heating	`0` = OK `1` = Not working	If heating is not working, monitor window contamination.

Bit [1)]	Message	Value	Description
5	Low input voltage detected	`0` = Voltage OK `1` = Voltage low	Unit operational, but check input voltage.
6	High input voltage detected	`0` = Voltage OK `1` = Voltage high	
7	High internal temperature detected	`0` = Temperature OK `1` = Temperature high	First notification of unit getting too hot.
8	Temperature unit	`0` = °C `1` = °F	–
9	Layer thickness unit	`0` = mm `1` = inch	–
10	Reference setting interrupted due to laser temperature change	`0` = False `1` = True	Repeat reference setting when laser temperature change has finished.
11	Reference setting interrupted due to hardware error, check parameter `0x56`	`0` = False `1` = True	Check error bits and parameter `0x56`, which contains reason for error.
12	Reference setting values are not updated due to poor signal quality	`0` = False `1` = True	Excessive variation in road surface type. Find more representative road surface.
13	Reference setting was interrupted by client	`0` = False `1` = True	–
14	Signal levels low, uncertainty in surface layer thickness results	`0` = False `1` = True	Unit operational, but measurement data may be invalid. Verify installation and clean window.
15	Road surface type significantly different from road surface type used in reference setting	`0` = False `1` = True	Unit is operational, but measurement data can be invalid. If condition persists, perform reference setting.
16	Layer thickness result produced undefined result	`0` = False `1` = True	Verify installation. If condition persists, perform reference setting.
17	Measurement over range, layer thickness measurement range exceeded	`0` = False `1` = True	Maximum reported layer thickness 60 mm (2.36 in) exceeded. If this does not represent actual conditions, perform reference setting.
18 - 31	Reserved for future use	-	–

1) Bit is shown in MD30 interface response data message, not in RoadAI.
2) Reported by MD30 models that have the window contamination feature.

Reference setting cannot be started if status info bit `0`, `1`, or `2` is set.

Laser temperature change stops the ongoing reference setting.

More information

- GET UNIT STATUS message (page 34)
- SEND DATA message (page 36)
- SET REFERENCES message (page 41)
- Parameters (page 62)

3.4 Error bits

Error bits indicate error situations in the unit or in communication. The erroneous bits are reported in the `GET UNIT STATUS`, `SEND DATA`, and `SET REFERENCES` messages.

Error is indicated by setting bit to `1`.

Table 8 Error messages

Bit [1)]	Message	Probable cause	Actions
0	Surface temperature sensor error	Cables may be loose, damaged, or disconnected.	Check cables and connectors. If the problem persists, replace mobile sensor.
1	Air temperature error		
2	Relative humidity error		
3	Window contamination alarm [2)]	Mobile sensor window is heavily contaminated.	Clean the window.
4	Laser status error	-	Restart mobile sensor.
5	Laser heating error		
6	Excessive ambient light detected	Sunlight is reflected from road surface to mobile sensor.	Move vehicle or reposition mobile sensor.
7	Receiver error	–	Restart mobile sensor.
8	Signal level out of range, gain adjustment limit reached	–	Check installation height and angle of mobile sensor.
9	Received signals contain too much noise	–	Check that mobile sensor is firmly attached to vehicle.

Bit [1]	Message	Probable cause	Actions
10	Optical measurement data timeout	–	Restart mobile sensor.
11	Low input voltage	Incorrect operating voltage.	Check operating voltage.
12	High input voltage		
13	Flash failure status	-	Restart mobile sensor.
14	Internal temperature too high	Mobile sensor overheated.	Disconnect mobile sensor from power supply.
15	Reference status: `0` = OK `1` = Invalid or not set	–	Verify installation with reference plate and adapt mobile sensor to road surface types.
16	Factory calibration status: `0` = OK `1` = Not calibrated	–	Return mobile sensor to Vaisala.
17-31	Reserved for future use	–	–

1) *Bit is shown in MD30 interface response data message, not in RoadAI.*
2) *Reported by MD30 models that have the window contamination feature.*

The reference setting cannot be started or is interrupted if an error is detected and indicated by bits **3** ... **14**, or **16**.

More information

- GET UNIT STATUS message (page 34)
- SEND DATA message (page 36)
- SET REFERENCES message (page 41)

3.5 Surface states

The following table shows the surface state values. Surface states are reported in the `SEND DATA` message.

Table 9 Surface states

Value	Description
0	Error
1	Dry
2	Moist
3	Wet
4 - 5	–

Value	Description
6	Snowy
7	Icy
8	–
9	Slushy
10 - 12	–

More information

- SEND DATA message (page 36)

3.6 EN 15518 surface states

The following table shows the surface state values according to the EN 15518 standard. EN 15518 surface states are reported in the `SEND DATA` message.

Table 10 EN 15118 surface states

Value	Description
0	Error
1	Dry
2	Moist
3	Wet
4 - 9	–
10	Streaming water
11	Slippery [1)]

1) Icy and slushy are reported as slippery in EN 15518.

More information

- SEND DATA message (page 36)

3.7 Data types

The following data types and ranges are available. The data is reported in the `SEND DATA` message.

Table 11 Data types

Data type	Format	Description
Grip	f32	Grip value Range: 0.09 ... 0.82 Missing data: NaN
Water layer thickness [1)]	f32	Water layer thickness Range: 0 ... 60 mm (0 ... 2.36 in) Missing data: NaN
Ice layer thickness [1)]	f32	Ice layer thickness Range: 0 ... 60 mm (0 ... 2.36 in) Missing data: NaN
Snow layer thickness (water equivalent) [1)]	f32	Snow layer thickness (water equivalent) Range: 0 ... 60 mm (0 ... 2.36 in) Missing data: NaN
Surface temperature	f32	Measured road surface temperature Range: —40 ... +60 °C or —40 ... +140 °F Missing data: NaN
Air temperature	f32	Measured air temperature Range: —40 ... 60 °C or —40 ... +140 °F Missing data: NaN
Dew point temperature		Dew point temperature Range: —40 ... +60 °C or —40 ... +140 °F Missing data: NaN As dew does not form in freezing conditions, dew point value is not relevant when it is below 0 °C (+32 °F).
Frost point temperature	f32	Dew / frost point temperature (dew point reported above freezing conditions, frost point reported in freezing conditions) Range: —40 ... +60 °C or —40 ... +140 °F Missing data: NaN In freezing conditions, instead of dew point, the relevant value for understanding the risk of frost (ice) formation is frost point
Relative humidity	f32	Measured relative humidity Range: 0 ... 100 %RH Missing data: NaN

1) Measurement accuracy is affected when layer thickness reaches a specified limit. For the limits, see MD30 datasheet.

More information

- SEND DATA message (page 36)

4. Calibration overview

4.1 Surface state and temperature calibration

The data collection system must implement the calibration commands and procedures to enable surface state calibration, including plate reference and dry road reference.

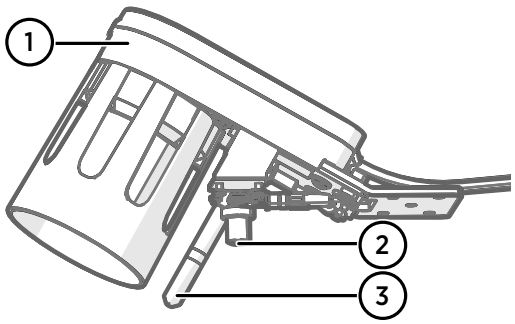
In addition, the interface must support surface temperature and air temperature calibration.

4.2 Surface state calibration

MD30 surface state calibration involves 2 separate adjustments, preceded by cleaning the surface state sensor. The steps are:

1. Cleaning
2. Plate reference
3. Dry road reference

This calibrates the surface state sensor.



1 Surface state sensor
2 Surface temperature sensor MT10
3 Air temperature and humidity sensor HMP113

MD30 calibration is a 3-step process. Make sure that you complete all the steps before starting to use MD30 for the first time.

When using multiple ports, note that reference setting:

- Can be initiated from any port.
- Cannot be started if there is reference setting ongoing, regardless of the port.
- Can be stopped from any port, regardless of the port from which it was started.

The following gives an overview of the calibration process and messaging. For detailed calibration steps and instructions, see *Vaisala Mobile Detector MD30 Setup Guide*.

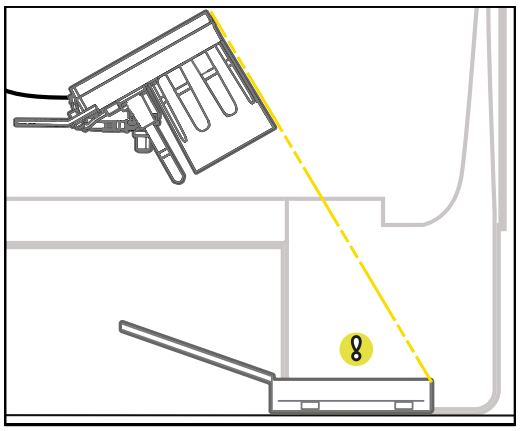
4.2.1 Plate reference overview

Why:

- Plate adjustment minimizes the effects caused by different installation heights and differences between individual mobile sensors. Repeat this procedure when changing the installation position.

How:

The reference setting is performed using the reference plate.



Which commands are needed to implement this?

Command: `SET REFERENCES`

- With parameter `0` (`PLATE`)
- Action takes typically 30 ... 60 seconds

What the command does?

- Changes reference value parameters `0x50`, `0x51`, and `0x52` (1 parameter for each laser).
- When started, changes status bit 1 (`reference setting ongoing`) to `1` = ongoing.
- When finished, changes back to `0` = not ongoing.

What is needed for command to succeed?

- Sensor should not report errors related to reference setting.

How do I know did the command succeed?

- If status of `reference setting ongoing` is `0` = not ongoing.
- There are no errors.

More information

- SET REFERENCES message (page 41)
- Parameters (page 62)
- Status information (page 12)
- Error bits (page 14)

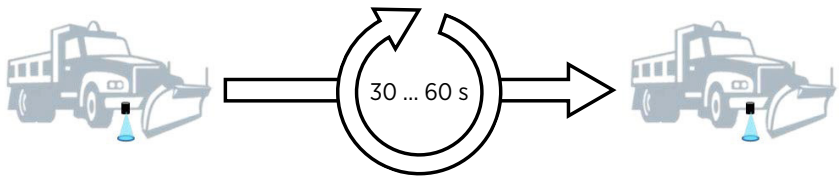
4.2.2 Dry road reference overview

Why:

- Sensor needs a reference of how local dry road looks like. The reference setting is performed while driving on the most common road surface type in the road network. This optimizes the sensor performance on the road type in question.

How:

- Option A: drive on local dry road with the most common surface type.
- Option B: use factory default dry road coefficients temporarily before local dry road becomes available.
- Option C: copy local dry road coefficients from another sensor.



Which commands are needed to implement this?

Command: SET REFERENCES

- With parameter `1` (**`ROAD`**)
- Action takes typically 30 ... 60 seconds

What the command does?

- Changes dry road coefficient parameters `0x53`, `0x54`, and `0x55` (1 parameter for each laser).
- When started, changes status bit 1 (`reference setting ongoing`) to `1` = ongoing.
- When finished, changes back to `0` = not ongoing.

What is needed for command to succeed?

- Sensor should not report errors related to reference setting.

How do I know did the command succeed?

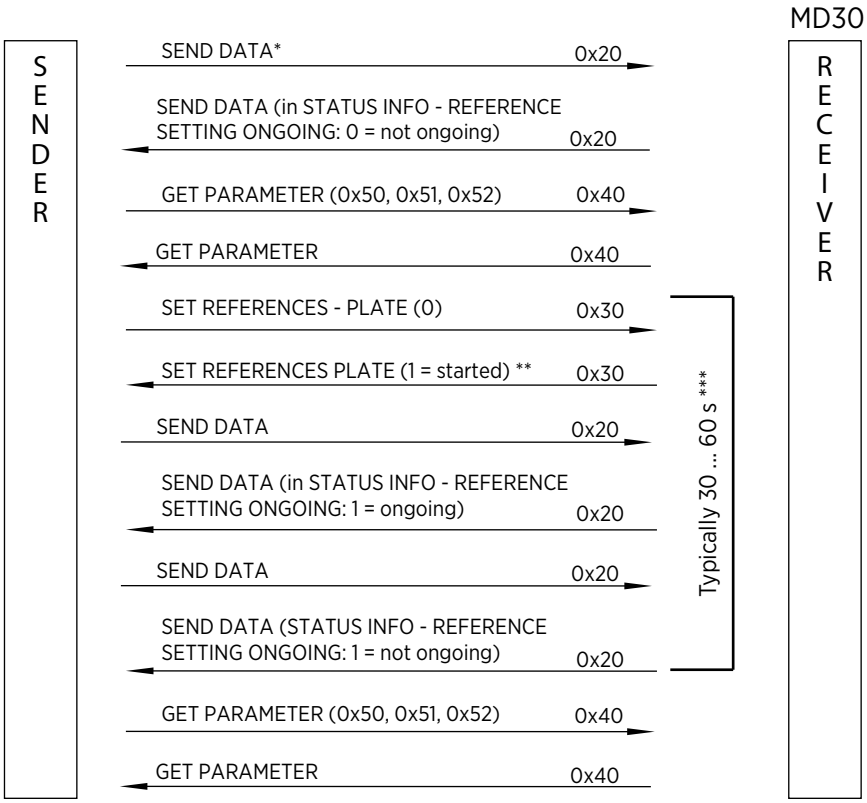
- If status of `reference setting ongoing` is `0` = not ongoing.
- There are no errors.

More information

- SET REFERENCES message (page 41)
- Parameters (page 62)
- Status information (page 12)
- Error bits (page 14)

4.2.3 Calibration messaging

ADJUSTMENT WITH REFERENCE PLATE

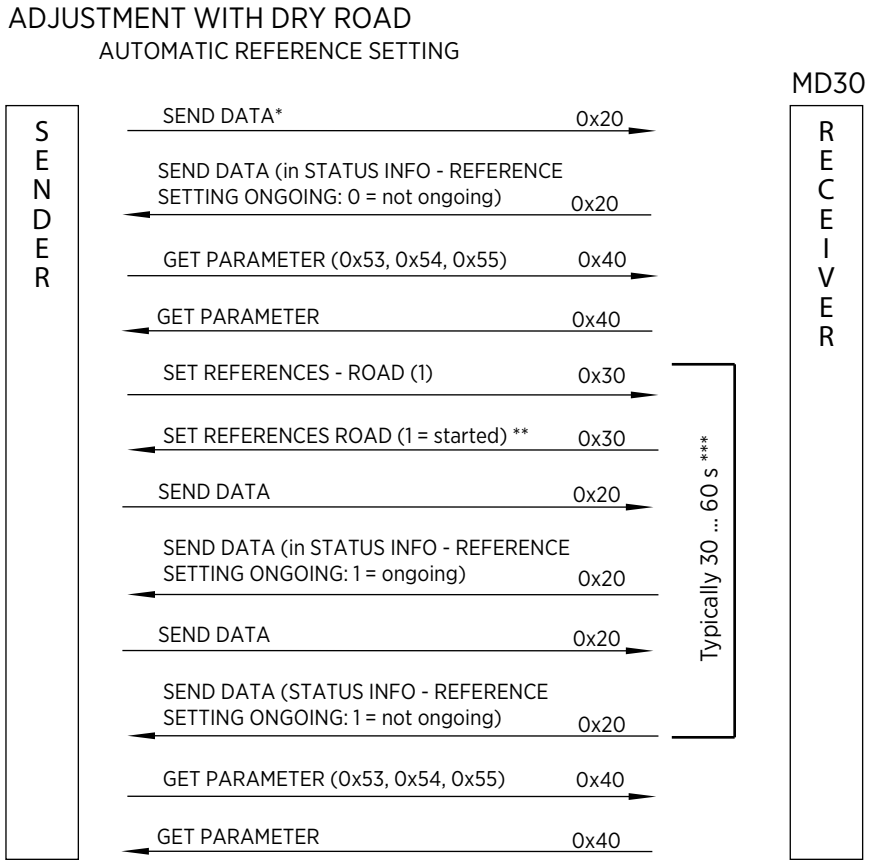


* Both GET UNIT STATUS and SEND DATA can be used to obtain status information.

** The response STARTED to SET REFERENCES request indicates that the data collection has started, not that references have been updated.

*** During reference setting, query status with GET UNIT STATUS or SEND DATA messages.

Figure 3 MD30 calibration messages: adjustment with reference plate



* Both GET UNIT STATUS and SEND DATA can be used to obtain status information.

** The response STARTED to SET REFERENCES request indicates that the data collection has started, not that references have been updated.

*** During reference setting, query status with GET UNIT STATUS or SEND DATA messages.

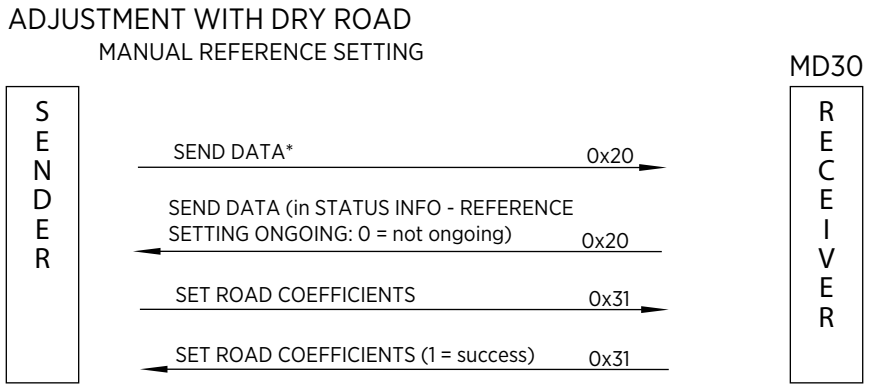
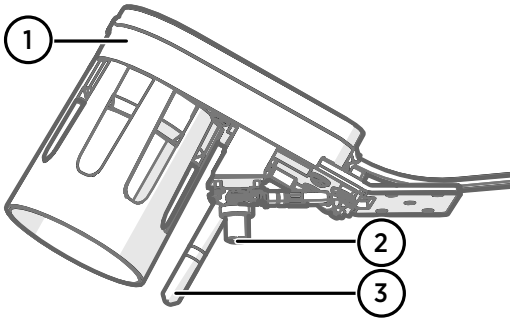


Figure 4 MD30 calibration messages: adjustment with dry road

4.3 Temperature calibration

MD30 temperature calibration is done with 1-point offset corrections for surface and air temperatures.

This calibrates the surface temperature sensor MT10 and air temperature and humidity sensor HMP113.



1 Surface state sensor
2 Surface temperature sensor MT10
3 Air temperature and humidity sensor HMP113

Surface temperature calibration	Air temperature calibration
• 1-point offset correction • Stored in parameter `0x40` • Can be changed with `SET PARAMETER` command • Can be any value (as 4-byte f32 floating point number)	• 1-point offset correction • Stored in parameter `0x41` • Can be changed with `SET PARAMETER` command • Can be any value (as 4-byte f32 floating point number)

Data ID	Read/Write	Description	Format	Value
0x40	R/W	Road surface temperature offset correction. Set and read using temperature unit selected with parameter `0x30`.	f32	0 = Default
0x41	R/W	Air temperature offset correction. Set and read using unit selected with parameter `0x30`.	f32	0 = Default

5. Data messages

5.1 Message format

The following tables list the data fields that are used in the request and response messages.

The byte sizes are calculated for binary format.

The messages, including the CRC error acknowledgment message, contain the listed fields. Data field is not always included.

Error messages do not carry any message data. The message ID and message number are copied from the request message and the error code is set according to the detected error. An exception is the CRC error acknowledgment message, where the message receiver ID, message ID, and message number are set to `0`.

Offset information indicates the position of the hexadecimal value in the message. The message start marker has offset 0.

Wait for the message to be acknowledged or for a communication timeout before sending subsequent messages.

Table 12 Request message fields

Field	Size in bytes	Description	Value
Message start	1	Message start marker, static value	`0xab`
Message sender ID	1	User	`0` = Default
Message receiver ID	1	MD30	`1` = Default
Message ID	1	Valid message ID	Message IDs (page 27)
Message number	1	Message number is copied from the request message to the response message and can be used for tracking which request message was acknowledged. If tracking is not needed, can be kept as zero (0).	`0` = Not used
Data length	2	Data length in bytes; depends on message type Data length can be zero (0).	0 ... 123
Data	0 ... 65535	Message data bytes	
CRC	2	CRC checksum CRC is calculated for the whole message, excluding the message start marker and the CRC bytes	CRC calculation (page 66)

Table 13 Response message fields

Field	Size in bytes	Description	Value
Message start	1	Message start marker, static value	`0xab`
Message sender ID	1	MD30	`1`= Default
Message receiver ID	1	User	`0` = Default
Message ID	1	Valid message ID	Message IDs (page 27)
Message number	1	Message number is copied from the request message. Exception is the **SEND DATA** response where the message number of the first message is copied from the request or set to zero in case on automatic sending, and after the first message, the message number is incremented in each response.	`0` = Not used
Data length	2	Data length in bytes; depends on message type The data length covers the interface version, error code, and response data. The data length is always at least 2 bytes because interface version and error code are always included.	2 ... 123
Interface version	1	Version of protocol in use by MD30 Hexadecimal value of the ASCII character, representing the revision	`A` = 0x41, `B` = 0x42, and so on
Error code	1	Error code according to detected error If no errors, set to `0`	Error codes (page 65)
Data	2 ... 65535	Message data bytes	
CRC	2	CRC checksum CRC is calculated for the whole message, excluding the message start marker and the CRC bytes	CRC calculation (page 66)

More information

- CRC error acknowledgment message (page 60)
- Error codes (page 65)
- Message and error handling (page 68)

5.2 Data message example

The transferred binary data is in hex format.

Request message

The format of the request data message is the following:

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <data> <CRC>
```

The following is an example of a single `SEND DATA` request that is sent to the mobile sensor in hex format.

```
0xab 0x00 0x01 0x20 0x0e 0x02 0x00 0x00 0x00 0x97 0x9e
```

The example request contains the following information:

- Unit ID: 1
- Client ID: 0
- Message ID: 0x20
- Message number: 14
- Length: 2
- Data interval: 0
- CRC: 0x9e97

Response message

The format of the response data message is the following:

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code> <data>
<CRC>
```

The following is an example of a single `SEND DATA` response that is sent from the mobile sensor in hex format.

```
0xab 0x01 0x00 0x20 0x0e 0x36 0x00 0x43 0x00 0xd7 0x08 0x00 0x00 0x00 0x00
0x8f 0xc2 0xbf 0x41 0x29 0x5c 0x45 0x42 0xfb 0x52 0x4b 0x41 0xfb 0x52 0x4b
0x41 0x08 0xd7 0x02 0x42 0x01 0x01 0x85 0xeb 0x51 0x3f 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x53 0xe8
```

The example response contains the following information:

- Unit ID: 1
- Client ID: 0
- Message ID: 0x20
- Message number: 14
- Interface version: C
- Length: 54
- Error code: 0
- Count: 2263
- Warning: 0

- Error: 0
- Air temperature: 23.9699
- RH: 49.3400
- Dew point: 12.7077
- Frost point: 12.7077
- Surface temperature: 32.7099
- Surface state: 1
- EN 15518: 1
- Grip: 0.8199
- Water: 0.00
- Ice: 0.00
- Snow: 0.00
- Status: 0
- Error: 0
- CRC: 0xe853

5.3 Message IDs

The message ID is used to identify the message type. It is part of the request and response messages.

Table 14 Message IDs

Message ID	Message
0x00	`CRC ERROR ACKNOWLEDGMENT`
0x10	`GET UNIT ID`
0x11	`GET FULL PRODUCT INFO`
0x12	`GET UNIT STATUS`
0x20	`SEND DATA`
0x30	`SET REFERENCES`
0x31	`SET ROAD COEFFICIENTS`
0x32	`STOP REFERENCE SETTING`
0x40	`GET PARAMETER`
0x41	`SET PARAMETER`
0x50	`RESTART UNIT`

5.4 GET UNIT ID message

`GET UNIT ID` is used for requesting the ID of the mobile sensor. The ID is the serial number.

Table 15 GET UNIT ID message

Message ID	0x10		
Data in request	None		
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Serial number	8
		Data length	8

5.4.1 GET UNIT ID request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
CRC	7 ... 8	2

Example

```
0xab 0x00 0x01 0x10 0x01 0x00 0x00 0xd6 0x88
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x10	0x10
Message number	0x01	1
Data length	0x00 0x00	0
CRC	0xd6 0x88	-

5.4.2 GET UNIT ID response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<serial number> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 … 6	2
Interface version	7	1
Error code	8	1
Serial number	9 … 16	8
CRC	17 … 18	2

Example

```
0xab 0x01 0x00 0x10 0x01 0x0a 0x00 0x44 0x00 0x52 0x32 0x37 0x33 0x30 0x30
0x31 0x31 0x10 0x02
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x10	0x10
Message number	0x01	1
Data length	0x0a 0x00	10
Interface version	0x44	D
Error code	0x00	0
Serial number	0x52 0x32 0x37 0x33 0x30 0x30 0x31 0x31	R2730011

Field	Bytes	Value
CRC	0x10 0x02	-

5.5 GET FULL PRODUCT INFO message

`GET FULL PRODUCT INFO` is used for requesting full product information.

Table 16 GET FULL PRODUCT INFO message

Message ID	0x11		
Data in request	None		
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Number of pairs	1
	10	Pair 1 key length	1
	11	Pair 1 key	max. 255 characters
	11 + pair 1 key length	Pair 1 value length	1
	11 + pair 1 key length + 1	Pair 1 value	max. 255 characters
	...	Next key value pair containing: • Key length • Key value • Value length • Value	...
		Data length	4 ... 65526

The message must contain:

- Product name
- Product serial number
- Software version
- Serial numbers of replaceable sensors

Also other information can be included.

The full product information is presented as key value pairs. Each key value pair contains:

- Key length presented with 1 byte
- Key value, a set of characters
- Value length presented with 1 byte
- Value, a set of characters

Table 17 Examples of key value pairs

Key	Value
Product name	MD30

Key	Value
Serial number	P1830002
Software version	0.2.0
MT10 ID	700572D61114B1C2
HMP serial number	P2130779

5.5.1 GET FULL PRODUCT INFO request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 … 6	2
CRC	7 … 8	2

Example

```
0xab 0x00 0x01 0x11 0x02 0x00 0x00 0x32 0xa7
```

Field	Bytes	Value
Message start	0xab	`0xab`
Message sender ID	0x00	`0`
Message receiver ID	0x01	`1`
Message ID	0x11	`0x11`
Message number	0x02	`2`
Data length	0x00 0x00	`0`
CRC	0x32 0xa7	-

5.5.2 GET FULL PRODUCT INFO response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <interface version number> <error code> <data length> <number
of pairs and key-value pairs and values> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
Data: number of pairs and key-value pairs and values	9 ... 119	111
CRC	120 ... 121	2

Example

For example, MD30 has 2 external sensors: HMP113 and MT10, so the full product information contains 5 key value pairs.

```
0xab 0x01 0x00 0x11 0x02 0x71 0x00 0x44 0x00 0x05 0x0c 0x50 0x72 0x6f 0x64
0x75 0x63 0x74 0x20 0x4e 0x61 0x6d 0x65 0x04 0x4d 0x44 0x33 0x30 0x0d 0x53
0x65 0x72 0x69 0x61 0x6c 0x20 0x4e 0x75 0x6d 0x62 0x65 0x72 0x08 0x52 0x32
0x37 0x33 0x30 0x30 0x31 0x31 0x0a 0x53 0x57 0x20 0x56 0x65 0x72 0x73 0x69
0x6f 0x6e 0x05 0x31 0x2e 0x31 0x2e 0x30 0x07 0x4d 0x54 0x31 0x30 0x20 0x49
0x44 0x10 0x37 0x43 0x30 0x45 0x32 0x36 0x31 0x41 0x36 0x34 0x41 0x34 0x42
0x31 0x43 0x32 0x11 0x48 0x4d 0x50 0x20 0x53 0x65 0x72 0x69 0x61 0x6c 0x20
0x4e 0x75 0x6d 0x62 0x65 0x72 0x08 0x50 0x34 0x30 0x33 0x30 0x30 0x32 0x32
0x40 0x07
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0

Field	Bytes	Value
Message ID	0x11	0x11
Message number	0x02	2
Data length	0x71 0x00	113
Interface version	0x44	D
Error code	0x00	0
Data: number of pairs and key-value pairs and values		
Number of pairs	0x05	5
Pair 1 key length	0x0c	12
Pair 1 key	0x50 0x72 0x6f 0x64 0x75 0x63 0x74 0x20 0x4e 0x61 0x6d 0x65	Product name
Pair 1 value length	0x04	4
Pair 1 value	0x4d 0x44 0x33 0x30	MD30
Pair 2 key length	0x0d	13
Pair 2 value	0x53 0x65 0x72 0x69 0x61 0x6c 0x20 0x4e 0x75 0x6d 0x62 0x65 0x72	Serial number
Pair 2 value length	0x08	8
Pair 2 value	0x52 0x32 0x37 0x33 0x30 0x30 0x31 0x31	R2730011
Pair 3 key length	0x0a	10
Pair 3 key	0x53 0x57 0x20 0x56 0x65 0x72 0x73 0x69 0x6f 0x6e	SW version
Pair 3 value length	0x05	5
Pair 3 value	0x31 0x2e 0x31 0x2e 0x30	1.1.0
Pair 4 key length	0x07	7
Pair 4 key	0x4d 0x54 0x31 0x30 0x20 0x49 0x44	MT10 ID
Pair 4 value length	0x10	16
Pair 4 value	0x37 0x43 0x30 0x45 0x32 0x36 0x31 0x41 0x36 0x34 0x41 0x34 0x42 0x31 0x43 0x32	7C0E261A64A4B1C2
Pair 5 key length	0x11	17
Pair 5 key	0x48 0x4d 0x50 0x20 0x53 0x65 0x72 0x69 0x61 0x6c 0x20 0x4e 0x75 0x6d 0x62 0x65 0x72	HMP serial number
Pair 5 value length	0x08	8
Pair 5 value	0x50 0x34 0x30 0x33 0x30 0x30 0x32 0x32	P4030022

Field	Bytes	Value
CRC	0x40 0x07	-

The example response contains the following information:

- Unit ID: 1
- Client ID: 0
- Message ID: 0x11
- Message number: 6
- Length: 113
- Interface version: C
- Error code: 0
- Product name: MD30
- Serial number: P1830002
- SW version: 0.9.0
- MT10 ID: 700572D61114B1C2
- HMP serial number: P2130779

5.6 GET UNIT STATUS message

`GET UNIT STATUS` is used for requesting status and error information from the mobile sensor.

Table 18 GET UNIT STATUS message

Message ID	0x12		
Data in request	None		
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Status info	4
	13	Error bits	4
		Data length	8

More information

- Status information (page 12)
- Error bits (page 14)

5.6.1 GET UNIT STATUS request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
CRC	7 ... 8	2

Example

```
0xab 0x00 0x01 0x12 0x10 0x00 0x00 0xed 0x11
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x12	0x12
Message number	0x10	16
Data length	0x00 0x00	0
CRC	0xed 0x11	-

5.6.2 GET UNIT STATUS response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<status info> <error bits> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1

Field	Offset	Size in bytes
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
Status info	9 ... 12	4
Error bits	13 ... 16	4
CRC	17 ... 18	2

Example

```
0xab 0x01 0x00 0x12 0x10 0x0a 0x00 0x44 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0xc2 0x26
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x12	0x12
Message number	0x10	16
Data length	0x0a 0x00	10
Interface version	0x44	D
Error code	0x00	0
Status info	0x00 0x00 0x00 0x00	0x00000000
Error bits	0x00 0x00 0x00 0x00	0x00000000
CRC	0xc2 0x26	-

5.7 SEND DATA message

SEND DATA is used for requesting the default data set from the mobile sensor unit. The message can be used for the following purposes:

- Report status and error information.
- Initiate road condition, temperature, and humidity data reporting continuously at defined interval.
- Stop continuous data reporting.
- Report road condition, temperature, and humidity data using polling.

Table 19 SEND DATA message

Message ID	0x20		
Data in request	**Offset**	**Data**	**Size in bytes**
	7	Data interval as u16	2
		Data length	2
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Data analyze count	2
	11	Data status warning	2
	13	Data status error	2
	15	Air temperature	4
	19	Relative humidity	4
	23	Dew point temperature	4
	27	Frost point temperature	4
	31	Surface temperature	4
	35	Surface state	1
	36	EN 15518 surface state	1
	37	Grip	4
	41	Water layer thickness	4
	45	Ice layer thickness	4
	49	Snow layer thickness	4
	53	Unit status info	4
	57	Unit error bits	4
		Data length	52

Data interval

- If the data interval is 0, only one set of requested data is sent back.
- If the data interval is 25 ... 5000 ms, the unit starts to send measurement data using the given interval. Data sending continues until data polling request with interval 0 is sent, or unit is restarted.

For a detailed description of the response message, see MD30 data message description (page 70).

More information

- Data status warnings (page 11)
- Data status erroneous (page 11)
- Status information (page 12)
- Error bits (page 14)
- Surface states (page 15)
- EN 15518 surface states (page 16)
- Data types (page 16)
- Automatic data sending (page 69)
- MD30 data message description (page 70)

5.7.1 SEND DATA request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <data interval> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Data interval	7 ... 8	2
CRC	9 ... 10	2

Example

```
0xab 0x00 0x01 0x20 0x11 0x02 0x00 0x00 0x00 0x34 0xff
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x20	0x20

Field	Bytes	Value
Message number	0x11	17
Data length	0x02 0x00	0
Data interval	0x00 0x00	0
CRC	0x34 0xff	-

5.7.2 SEND DATA response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<data analyze count> <data status warning> <data status error>
<air temperature> <relative humidity> <dew point temperature>
<frost point temperature> <surface temperature> <surface state>
<EN 15518 surface state> <grip> <water layer thickness> <ice layer thickness>
<snow layer thickness> <unit status info> <unit error bits> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
Data analyze count	9 ... 10	2
Data status warning	11 ... 12	2
Data status error	13 ... 14	2
Air temperature	15 ... 18	4
Relative humidity	19 ... 22	4
Dew point temperature	23 ... 26	4
Frost point temperature	27 ... 30	4
Surface temperature	31 ... 34	4
Surface state	35	1

Field	Offset	Size in bytes
EN 15518 surface state	36	1
Grip	37 ... 40	4
Water layer thickness	41 ... 44	4
Ice layer thickness	45 ... 48	4
Snow layer thickness	49 ... 52	4
Unit status info	53 ... 56	4
Unit error bits	57 ... 60	4
CRC	61 ... 62	2

Example

```
0xab 0x01 0x00 0x20 0x0a 0x36 0x00 0x44 0x00 0xfc 0xee 0x00 0x00 0x00 0x00
0x66 0x66 0xc4 0x41 0x5c 0x8f 0x51 0x42 0xad 0x83 0x57 0x41 0xad 0x83 0x57
0x41 0x3a 0x0a 0xbf 0x41 0x01 0x01 0x85 0xeb 0x51 0x3f 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x52 0xb3
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x20	0x20
Message number	0x0a	10
Data length	0x36 0x00	54
Interface version	0x44	D
Error code	0x00	0
Data analyze count	0xfc 0xee	61180
Data status warning	0x00 0x00	0
Data status error	0x00 0x00	0
Air temperature	0x66 0x66 0xc4 0x41	24.55
Relative humidity	0x5c 0x8f 0x51 0x42	52.39
Dew point temperature	0xad 0x83 0x57 0x41	13.47
Frost point temperature	0xad 0x83 0x57 0x41	13.47
Surface temperature	0x3a 0x0a 0xbf 0x41	23.88

Field	Bytes	Value
Surface state	0x01	1 = Dry
EN 15518 surface state	0x01	1 = Dry
Grip	0x85 0xeb 0x51 0x3f	0.82
Water layer thickness	0x00 0x00 0x00 0x00	0.00
Ice layer thickness	0x00 0x00 0x00 0x00	0.00
Snow layer thickness	0x00 0x00 0x00 0x00	0.00
Unit status info	0x00 0x00 0x00 0x00	0
Unit error bits	0x00 0x00 0x00 0x00	0
CRC	0x52 0xb3	-

The data analyze count starts from 0 and is incremented at the maximum measurement rate supported by the mobile road sensor. The counter goes to 0 after reaching the maximum value

When continuous sending is initiated, the message number of the request message is used in the first data message and then incremented by 1 every time a new message is generated and sent.

5.8 SET REFERENCES message

SET REFERENCES is used for updating reference values during surface state calibration for plate reference and dry road reference (dry road coefficient).

Table 20 SET REFERENCES message

Message ID	0x30		
Data in request	**Offset**	**Data**	**Size in bytes**
	7	Surface type: `0` = Plate `1` = Road	1
		Data length	1
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Operation started / not started: `0` = Not started `1` = Started	1
	10	Status info	4
	14	Error bits	4
		Data length	9

The ongoing reference setting does not affect the measurements, but the parameter update after data collection can interfere with the ongoing data transfer.

Reference setting is part of the following calibration steps:

- Plate reference
- Dry road reference

Reference setting consists of the following phases:

1. Initiation
2. Data collection
3. Reference setting update

The phases are the same for both plate reference and dry road reference setting.

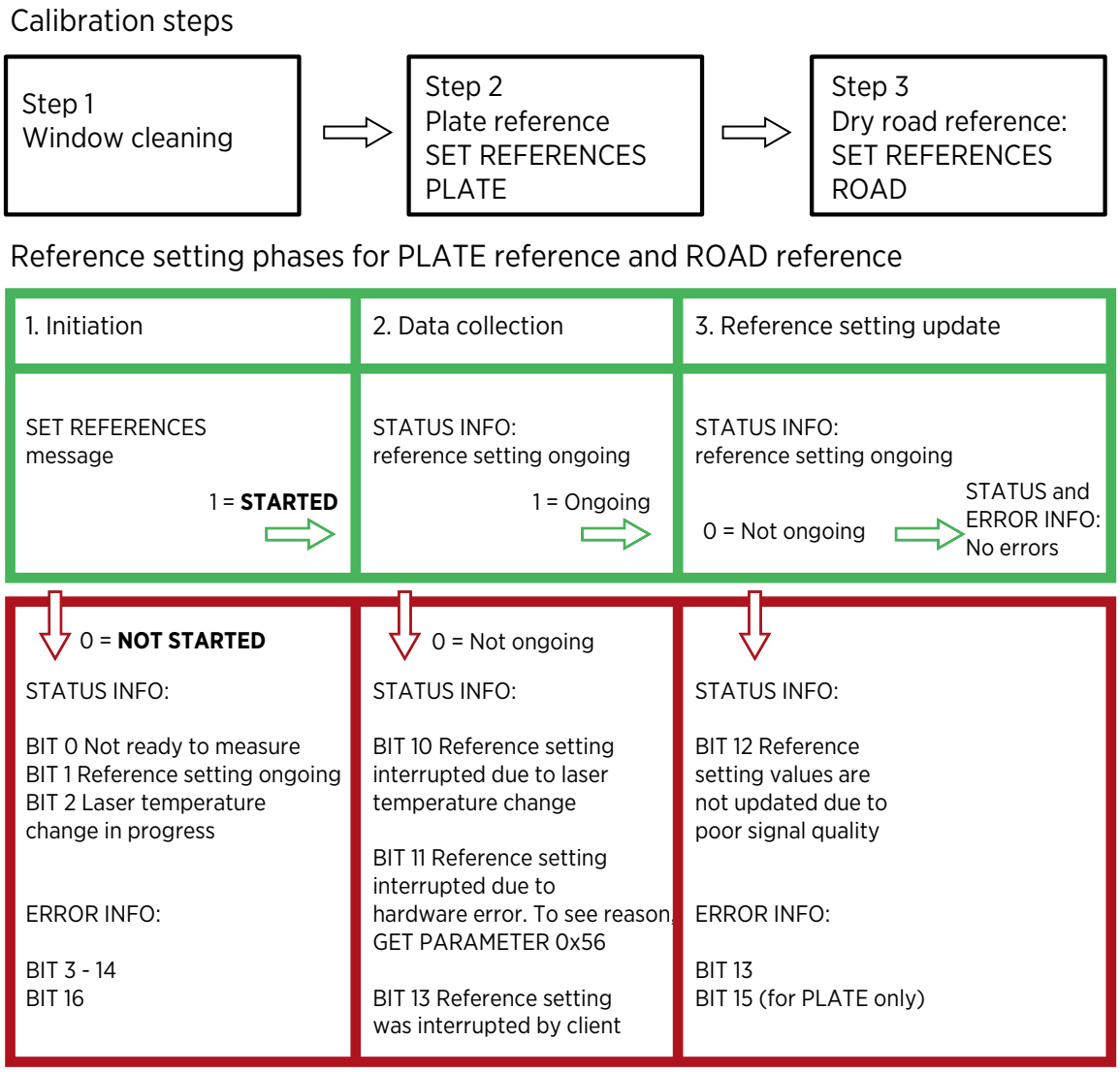


Figure 5 Reference setting as part of PLATE and ROAD reference calibration steps

The response `Started` to `SET REFERENCES` request indicates that the data collection has started, not that references have been updated.

During reference setting, check the progress with `GET UNIT STATUS` or `SEND DATA` message.

The updated reference setting values and corresponding parameters depend on the surface type.

Table 21 Reference setting parameters

Reference setting	Vehicle	Surface type	Parameters
Plate reference	At standstill	PLATE	0x50, 0x51, 0x52
Dry road reference	Moving	ROAD	0x53, 0x54, 0x55

Initiation

SET REFERENCES initiates data collection for the reference values setting.

When the mobile sensor receives the SET REFERENCES message, it clears the 3 status info bits that indicate a previous failed reference setting, and parameter 0x56 is reset.

If initiation fails, the error condition is indicated with the status info and error bits in the response message. The status and error information list the conditions that were used for checking whether reference setting can be started.

For a list of error conditions, see the previous image. For details about the error conditions, see Status information (page 12) and Error bits (page 14).

Data collection

Data collection takes a minimum of 25 seconds. If the road surface is not static during dry road reference, data collection can take longer.

Data collection is stopped if there are error conditions that prevent the start of reference setting. The reason is indicated in the status information, and stored in parameter 0x56. References are not updated.

Reference setting update

After collecting enough data, the sample quality is checked.

- If the quality criteria for the sample is met, new reference setting values are calculated and references updated.
- If the quality criteria is not met, references are not updated. Failure is indicated in the status info.

Example

After setting PLATE reference values using reference plate (calibration step 2), the reference values are set for dry road reference (calibration step 3):

1. Client sends the SET REFERENCES message with the parameter ROAD.
2. Initiation succeeds. The unit collects data for 25 seconds. The **reference setting ongoing** bit is set in status info bits.
3. The client can check from the status info bits if data collection is ongoing. Both GET UNIT STATUS response and SEND DATA contain status info bits.
4. All data is collected. Unit clears the **reference setting ongoing** bit. Dry road coefficients are updated and taken into use.

More information

- Status information (page 12)
- Error bits (page 14)
- Plate reference overview (page 19)
- Dry road reference overview (page 20)

5.8.1 SET REFERENCES request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <surface type> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Surface type	7	1
CRC	8 ... 9	2

Example

```
0xab 0x00 0x01 0x30 0x01 0x01 0x00 0x01 0x25 0xe9
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x30	0x10
Message number	0x01	1
Data length	0x01 0x00	1
Surface type	0x01	1 = Road
CRC	0x25 0xe9	-

5.8.2 SET REFERENCES response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<operation started / not started> <status info> <error bits> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
Operation started / not started	9	1
Status info	10 ... 13	4
Error bits	14 ... 17	4
CRC	18 ... 19	2

Example

```
0xab 0x01 0x00 0x30 0x01 0x0b 0x00 0x44 0x00 0x01 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0xbf 0x45
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x30	0x30
Message number	0x01	1
Data length	0x0b 0x00	11
Interface version	0x44	D
Error code	0x00	0

Field	Bytes	Value
Operation started / not started	0x01	1 = Started
Status info	0x00 0x00 0x00 0x00	0
Error bits	0x00 0x00 0x00 0x00	0
CRC	0xbf 0x45	-

5.9 STOP REFERENCE SETTING message

`STOP REFERENCE SETTING` is used for interrupting reference data collection.

The reference values and dry road coefficients are not changed when the reference setting is interrupted. Interrupted reference data collection is indicated in the status info. The flag is cleared when reference setting starts.

Table 22 STOP REFERENCE SETTING message

Message ID	0x32
Data in request	None
Data in response	None

5.9.1 STOP REFERENCE SETTING request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <CRC>
```

Field	Offset	Size in Bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
CRC	7 ... 8	2

Example

```
0xab 0x00 0x01 0x32 0x02 0x00 0x00 0xa0 0x0b
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x32	0x32
Message number	0x02	2
Data length	0x00 0x00	0
CRC	0xa0 0x0b	-

5.9.2 STOP REFERENCE SETTING response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code> <CRC>
```

Field	Offset	Size in Bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
CRC	9 ... 10	2

Example

```
0xab 0x01 0x00 0x32 0x02 0x02 0x00 0x44 0x00 0xc2 0xba
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x32	0x32
Message number	0x02	2
Data length	0x02 0x00	2
Interface version	0x44	D
Error code	0x00	0 = Stop reference setting acknowledged
CRC	0xc2 0xba	-

5.10 SET ROAD COEFFICIENTS message

`SET ROAD COEFFICIENTS` is used for updating all dry road coefficients for parameters `0x53`, `0x54`, and `0x55`, and taking new values in use immediately.

The parameter update can interfere with ongoing data transfer.

If the dry road coefficients are not known, the **SET REFERENCES** ROAD message must be used instead of the `SET ROAD COEFFICIENTS` message.

Table 23 SET ROAD COEFFICIENTS message

Message ID	0x31		
Data in request	**Offset**	**Data**	**Size in bytes**
	7	Dry road coefficient, laser 1	4
	11	Dry road coefficient, laser 2	4
	15	Dry road coefficient, laser 3	4
		Data length	12
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Operation success/fail: 0 = Fail 1 = Success	1
		Data length	1

The response times can be longer when parameter values are changed. The extra time is needed to write parameters into the permanent memory before sending the response.

5.10.1 SET ROAD COEFFICIENTS request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <dry road coefficient for laser 1>
<dry road coefficient for laser 2> <dry road coefficient for laser 3> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Dry road coefficient for laser 1	7 ... 10	4
Dry road coefficient for laser 2	11 ... 14	4
Dry road coefficient for laser 3	15 ... 18	4
CRC	19 ... 20	2

Example

```
0xab 0x00 0x01 0x31 0x0f 0x0c 0x00 0x9a 0x99 0xd1 0x40 0x85 0xeb 0xc9 0x40
0xb8 0x1e 0xc5 0x40 0xf4 0x49
```

Field	Bytes	Value
Message start	0xab	`0xab`
Message sender ID	0x00	`0`
Message receiver ID	0x01	`1`
Message ID	0x31	`0x31`
Message number	0x0f	`15`
Data length	0x0c 0x00	`12`

Field	Bytes	Value
Dry road coefficient for laser 1	0x9a 0x99 0xd1 0x40	6.55
Dry road coefficient for laser 2	0x85 0xeb 0xc9 0x40	6.31
Dry road coefficient for laser 3	0xb8 0x1e 0xc5 0x40	6.16
CRC	0xf4 0x49	-

5.10.2 SET ROAD COEFFICIENTS response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<operation success/fail> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
Operation success/fail	9	1
CRC	10 ... 11	2

Example

```
0xab 0x01 0x00 0x31 0x0f 0x03 0x00 0x44 0x00 0x01 0x20 0xe8
```

Field	Bytes	Value
Message start	0xab	`0xab`
Message sender ID	0x01	`1`
Message receiver ID	0x00	`0`
Message ID	0x31	`0x31`

Field	Bytes	Value
Message number	0x0f	15
Data length	0x03 0x00	3
Interface version	0x44	D
Error code	0x00	0
Operation success/fail	0x01	1 = Success
CRC	0x20 0xe8	-

5.11 GET PARAMETER message

GET PARAMETER returns the value of the given parameter. The data length in the response is determined by its type.

Table 24 GET PARAMETER message

Message ID	0x40		
Data in request	**Offset**	**Data**	**Size in bytes**
	7	Parameter ID	2
		Data length	2
Data in response	**Offset**	**Data**	**Size in bytes**
	9	Parameter ID	2
	11	Data value	1...4
		Data length	3...6

More information

- Parameters (page 62)

5.11.1 GET PARAMETER request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <parameter ID> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1

Field	Offset	Size in bytes
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Parameter ID	7 ... 8	2
CRC	9 ... 10	2

Example 1

The requested parameter is `0x13`, mobile sensor ID.

```
0xab 0x00 0x01 0x40 0x12 0x02 0x00 0x13 0x00 0xde 0x18
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x40	0x40
Message number	0x10	16
Data length	0x02 0x00	2
Parameter ID	0x13 0x00	0x13
CRC	0x5d 0x5c	-

Example 2

The requested parameter is `0x41`, air temperature offset correction.

```
0xab 0x00 0x01 0x40 0x11 0x02 0x00 0x41 0x00 0xd1 0x9e
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x40	0x40
Message number	0x11	17

Field	Bytes	Value
Data length	0x02 0x00	2
Parameter ID	0x41 0x00	0x41 = air temperature offset correction
CRC	0xd1 0x9e	-

5.11.2 GET PARAMETER response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code>
<parameter ID> <sensor ID> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 - 6	2
Interface version	7	1
Error code	8	1
Parameter ID	9 ... 10	2
Sensor ID	11	1
CRC	12 ... 13	2

Example 1

The response returns the mobile sensor ID that was requested with parameter `0x13`.

```
0xab 0x01 0x00 0x40 0x10 0x05 0x00 0x44 0x00 0x13 0x00 0x01 0xf0 0x85
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1

Field	Bytes	Value
Message receiver ID	0x00	0
Message ID	0x40	0x40
Message number	0x10	16
Data length	0x05 0x00	5
Interface version	0x44	D
Error code	0x00	0
Parameter ID	0x13 0x00	0x13 = Sensor ID
Sensor ID	0x01	1
CRC	0xf0 0x85	-

Example 2

The response returns the air temperature offset correction that was requested with parameter `0x41`.

```
0xab 0x01 0x00 0x40 0x11 0x08 0x00 0x44 0x00 0x41 0x00 0x00 0x00 0x00 0x00
0x79 0x1e
```

The example response contains the following information:

- Unit ID: `1`
- Client ID: `0`
- Message ID: `0x40`
- Message number: `17`
- Length: `8`
- Interface version: `D`
- Error code: `0`
- Parameter ID: `0x41`
- Value: `0.00000` (air temperature offset)

5.12 SET PARAMETER message

`SET PARAMETER` is used for setting a given parameter to a given value.

The message can be used for the following purposes:

- Stop continuous data reporting.
- Set reference values.
- Set offset to road and air temperature.
- Configure the serial port speed.
- Configure units of reported quantities.

Some of the parameters are read-only.
An attempt to set a value to a read-only parameter causes an `INVALID DATA` error.

Some of the parameters require a restart for the changes to take effect. The reported parameter value is reported immediately.

The response times can be longer when parameter values are changed. The extra time is needed to write parameters into the permanent memory before sending the response.

Table 25 SET PARAMETER message

Message ID	0x41		
Data in request	**Offset**	**Data**	**Size in bytes**
	7	Parameter ID	2
	9	Parameter value	1...4
		Data length	3...6
Data in response	None		

More information

- Parameters (page 62)
- Message and error handling (page 68)

5.12.1 SET PARAMETER request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <parameter ID> <parameter value> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1

Field	Offset	Size in bytes
Data length	5 … 6	2
Parameter ID	7 … 8	2
Parameter value	9 … 12	4
CRC	13 … 14	2

Example

The configured parameter is `0x41`, air temperature offset correction.

```
0xab 0x00 0x01 0x41 0x12 0x06 0x00 0x41 0x00 0x00 0x00 0x40 0x3f 0x7d 0xb1
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x41	0x41
Message number	0x12	18
Data length	0x06 0x00	6
Parameter ID	0x41 0x00	0x41 = air temperature offset
Parameter value	0x00 0x00 0x40 0x3f	0.75
CRC	0x7d 0xb1	-

5.12.2 SET PARAMETER response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1

Field	Offset	Size in bytes
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
CRC	9 ... 10	2

Example

```
0xab 0x01 0x00 0x41 0x12 0x02 0x00 0x44 0x00 0xe4 0x35
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x41	0x41
Message number	0x12	18
Data length	0x02 0x00	2
Interface version	0x44	D
Error code	0x00	0
CRC	0xe4 0x35	-

5.13 RESTART UNIT message

`RESTART UNIT` is used for initiating a software restart after an acknowledgment has been sent.

First the unit acknowledges the restart message, and restarts the software after that.

Table 26 RESTART UNIT message

Message ID	0x50
Data in request	None
Data in response	None

More information

- Parameters (page 62)

5.13.1 RESTART UNIT request

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <CRC>
```

Field	Offset	Size in bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
CRC	7 ... 8	2

Example

```
0xab 0x00 0x01 0x50 0x00 0x00 0x00 0x7a 0xd1
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x00	0
Message receiver ID	0x01	1
Message ID	0x50	0x50
Message number	0x00	0
Data length	0x00 0x00	0
CRC	0x7a 0xd1	-

5.13.2 RESTART UNIT response

Format

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code> <CRC>
```

Field	Offset	Size in Bytes
Message start	0	1
Message sender ID	1	1
Message receiver ID	2	1
Message ID	3	1
Message number	4	1
Data length	5 ... 6	2
Interface version	7	1
Error code	8	1
CRC	9 ... 10	2

Example

```
0xab 0x01 0x00 0x50 0x00 0x02 0x00 0x44 0x00 0x19 0x2a
```

Field	Bytes	Value
Message start	0xab	0xab
Message sender ID	0x01	1
Message receiver ID	0x00	0
Message ID	0x00	0
Message number	0x15	21
Data length	0x02 0x00	2
Interface version	0x44	D
Error code	0x00	0 = Restart acknowledged
CRC	0x19 0x2a	-

5.14 CRC error acknowledgment message

MD30 sends an CRC error acknowledgment message when it detects an error in the CRC checksum.

The CRC error acknowledgment message does not contain data. The message receiver ID, message ID, and message number are set to `0`.

To disable CRC error acknowledgments, use parameter `0x11`.

Example of request message with CRC error

```
0xab 0x00 0x01 0x10 0x00 0x00 0x00 0x00 0x00
```

The example request is a `GET UNIT ID` message with a checksum of `0`.

Format of CRC error acknowledgment message

```
<message start> <message sender ID> <message receiver ID> <message ID>
<message number> <data length> <interface version number> <error code> <CRC>
```

Message field	Offset	Size in bytes	Bytes	Value
Message start	0	1	0xab	0xab
Message sender ID	1	1	0x01	1
Message receiver ID	2	1	0x00	0
Message ID	3	1	0x00	0x00
Message number	6	1	0x00	0
Data length	4 - 5	2	0x02 0x00	2
Interface version	7	1	0x43	C
Error code	8	1	0x01	1 = CRC error
CRC	9 ... 10	2	0x3b 0xd3	-

Example

```
0xab 0x01 0x00 0x00 0x00 0x02 0x00 0x44 0x01 0xac 0x4a
```

The example response contains the following information:

- Unit ID: `1`
- Client ID: `0`
- Message ID: `0x00`
- Message number: `0`
- Length: `2`
- Interface version: `D`
- Error code: `1` (CRC error)

More information

- Error codes (page 65)

5.15 Parameters

The following table lists the parameter IDs and values.

The parameter value is sent in the request and response messages in the format given in the table.

The parameters are read and configured using the `GET PARAMETER` and `SET PARAMETER` messages.

Table 27 Parameters

Data ID	Read/ Write	Scope	Description	Format	Value
0x10	R/W	Port specific	Baud rate (bps). Defines the serial port speed. When writing to parameters, the parameter value is updated immediately, but the changes take effect after a restart. Communication works with old values until software is restarted. Baud rate change requires restart for the changes to take effect.	u8	`0` = 9600 `1` = 19200 `2` = 38400 `3` = 57600 `4` = 115200
0x11	R	Port specific	Send acknowledgment about CRC checksum error.	u8	`0` = No `1` = Yes
0x12	R	Port specific	Latest error code. Read-only, not stored to non-volatile memory.	u8	For possible values, see Error codes (page 65)
0x13	R/W	Device specific	Mobile sensor unit ID that is used in the header. When writing to parameters, the parameter value is updated immediately, but the changes take effect after a restart. Communication works with old values until software is restarted.	u8	`1` = Default Values `0xFE` and `0xFF` are not allowed.
0x14	R/W	Port specific	Receiver ID when automatic data sending is started. ID value is used as receiver ID only when automatic data sending is enabled.	u8	`0` = Default
0x20	R/W	Port specific	Data sending interval from MD30 to system (`SEND DATA`). Data interval in milliseconds (ms). See Automatic data sending (page 69).	u16	Range 25 ... 5000 `0` = Automatic sending off (default)

Data ID	Read/ Write	Scope	Description	Format	Value
0x21	R/W	Port specific	Continuous sending started automatically after start-up. See Automatic data sending (page 69).	u8	`0` = Not enabled (default) `1` = Enabled
0x30	R/W	Port specific	Temperature unit. Changing temperature unit updates temperature offset correction parameters `0x40` and `0x41`.	u8	`0` = °C (default) `1` = °F
0x31	R/W	Port specific	Layer thickness unit.	u8	`0` = mm (default) `1` = inch
0x40	R/W	Device specific	Road surface temperature offset correction. Set and read using temperature unit selected with parameter `0x30`.	f32	`0` = Default
0x41	R/W	Device specific	Air temperature offset correction. Set and read using unit selected with parameter `0x30`.	f32	`0` = Default
0x50	R/W	Device specific	Reference setting value of laser 1. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x51	R/W	Device specific	Reference setting value of laser 2. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x52	R/W	Device specific	Reference setting value of laser 3. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x53	R/W	Device specific	Dry road coefficient of laser 1. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x54	R/W	Device specific	Dry road coefficient of laser 2. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x55	R/W	Device specific	Dry road coefficient of laser 3. Value change requires restart for the changes to take effect.	f32	`1.0` = Default Value must be greater than 0
0x56	R	Device specific	Error condition causing reference setting to be interrupted. Read-only, not stored to non-volatile memory. Error code is reset when reference setting is started.	u32	

When the `SET PARAMETER` message updates reference related parameters `0x50` - `0x55`, changes take effect after the unit is restarted.

Receiver ID `0xFF` can be used in cases where the assigned mobile sensor ID is not known to the client. The assigned ID is in the response message.

More information

- GET PARAMETER message (page 52)
- SET PARAMETER message (page 55)
- RESTART UNIT message (page 58)

6. Flow control

6.1 Flow control

A generic receiver performs the following checks to the received data:

- Message start marker is located to find the beginning of the message.
- Data length is recorded and the recorded number of data bytes is received or timeout occurs.
- Checksum of the message is checked. Messages with a checksum error are ignored.
- Receiver ID must match the assigned unit ID.
- Message ID must be valid.
- Data length must comply with the message description.
- Data contents must comply with the message data description.

These rules apply to both requests and responses.

The client design should take it into account that if a mobile sensor is commanded to send continuously data (`SEND DATA`), the response to a command is not necessarily the first message received after issuing the command. If the mobile sensor is about to send the data message at the same time when client is sending its command, the client may receive the data message before the acknowledgement. For example:

1. Mobile road sensor is set to transmit `SEND DATA` response messages at 1 Hz.
2. Client sends the `GET STATUS` message.
3. Mobile road sensor starts sending `SEND DATA` at the same time with the client.
4. Client receives the `SEND DATA` message.
5. Client receives the `GET STATUS` response.

6.2 Error codes

The error codes are used in the response message.

Table 28 Error codes

Error code	Description	Explanation
0	No error	-
1	CRC checksum error	CRC checksum check fails.
2	Invalid message ID	Unit receives an unknown message ID.
3	Invalid length	Data length does not comply with the message type. For example, a data length of 0 in a message which always contains data triggers this error.
4	Invalid data	Data contained in the message is invalid. For example, an attempt to set an invalid port speed or to set value to a read-only parameter.

More information

- CRC error acknowledgment message (page 60)

6.3 CRC calculation

The cyclic redundancy check (CRC) that is used in the messages is CRC-16 CCITT-FALSE.

The following is a C code implementation.

```
u16 calculateCrc(u16   bufSize, u8 * buf)
{
u16 i;
u16 index;
u16 crc   = 0xFFFF;
for (i = 0; i <   bufSize; i++)
{
index = (crc   >> 8) ^ buf[i];
crc = (crc   << 8) ^ crcTable[index & 0xFF];
}
return crc;
}
#define   SIZE_CRC_TABLE  (256)
const u16   crcTable[SIZE_CRC_TABLE] =
{
  0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,  0x8108,
0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,  0x1231, 0x0210,
0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,  0x9339, 0x8318, 0xB37B,
0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,  0x2462, 0x3443, 0x0420, 0x1401,
0x64E6, 0x74C7, 0x44A4, 0x5485,  0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE,
0xF5CF, 0xC5AC, 0xD58D,  0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6,
0x5695, 0x46B4,  0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D,
0xC7BC,  0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,  0x5AF5,
0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,  0xDBFD, 0xCBDC,
0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,  0x6CA6, 0x7C87, 0x4CE4,
0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,  0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD,
0xAD2A, 0xBD0B, 0x8D68, 0x9D49,  0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13,
0x2E32, 0x1E51, 0x0E70,  0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A,
0x9F59, 0x8F78,  0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E,
0xE16F,  0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,  0x02B1,
0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,  0xB5EA, 0xA5CB,
0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,  0x34E2, 0x24C3, 0x14A0,
0x0481, 0x7466, 0x6447, 0x5424, 0x4405,  0xA7DB, 0xB7FA, 0x8799, 0x97B8,
0xE75F, 0xF77E, 0xC71D, 0xD73C,  0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657,
0x7676, 0x4615, 0x5634,  0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9,
0xB98A, 0xA9AB,  0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882,
0x28A3,  0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,  0xFD2E,
0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,  0x7C26, 0x6C07,
0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,  0xEF1F, 0xFF3E, 0xCF5D,
0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,  0x6E17, 0x7E36, 0x4E55, 0x5E74,
0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};
```

For example, the checksum of the following 10-byte sequence is 0xc241:

```
0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9
```

6.4 Message and error handling

The mobile sensor records the time when the message handling starts. The mobile sensor expects to receive a complete message within the given time, which is known as data discarding period. If a checksum error is detected, all received data is discarded until data discarding period has passed. If message reception is not ready by the end of data discarding period, the receiver is reset and the received data bytes are ignored. The unit is ready to receive the next message after the response message sending has started or the data discarding period has passed.

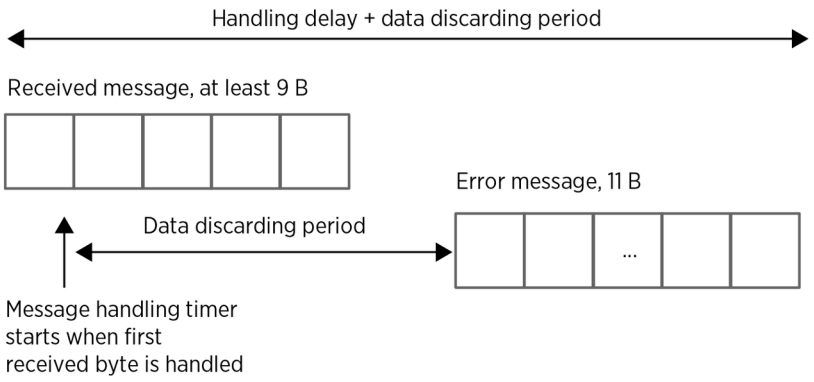


Figure 6 Error handling

Response time

In case a checksum error is acknowledged, the receiver waits the data discarding period of 20 ms before producing a response message. The data discarding period is included in the total response time. Disabling the checksum error acknowledgment does not affect the total time. Incomplete messages and time-outs have the same total time, but are not acknowledged.

The mobile sensor is expected to start data handling and to produce a response message in 500 ms after data reception has started. This is the total response time, including message handling delays and the data discarding period. This time does not include the time it takes to physically transmit and receive the message at a given data rate. Ongoing continuous data sending may be interrupted for the duration of the message handling.

Response times of up to 2.5 seconds can be expected from messages that update several parameters. The `SET ROAD COEFFICIENTS` message has the longest response time. Ongoing continuous data sending is interrupted for the duration of the message handling. Also the `SET REFERENCES` message updates several parameters. The response is sent within 500 ms, but the parameter update takes place after the data collection period. During the parameter update, continuous data sending frequency cannot be guaranteed. Assigned data sending frequency is assumed as soon as the parameter update is over.

Buffer overflow

Buffer overflow occurs if the mobile sensor receives a message that does not fit into the data buffer that is allocated for data reception. In case of buffer overflow, the receiver is reset and no acknowledgment is sent.

6.5 Automatic data sending

The mobile sensor can be configured to start sending data automatically after boot. Parameters `0x20` and `0x21` control the automatic data sending of the `SEND DATA` message.

Parameters are checked when the unit is started. If sending is enabled and the interval is not zero, the data sending starts.

The automatic data sending interval cannot be adjusted while the unit is sending data. Changes to the parameter values take effect when the unit is restarted.

`SEND DATA` can be used for stopping data sending or changing the interval. Note that the parameters controlling the automatic sending are not affected. If the parameter controlling the interval is not updated, automatic sending with the original interval resumes when the unit is restarted.

The receiver ID that is used in automatic sending is defined by parameter `0x14`. The parameter is common to all data messages.

More information

- SEND DATA message (page 36)

Appendix A. MD30 data message description

Here is a detailed description of the response message for SEND DATA message (page 36).

The following table includes:

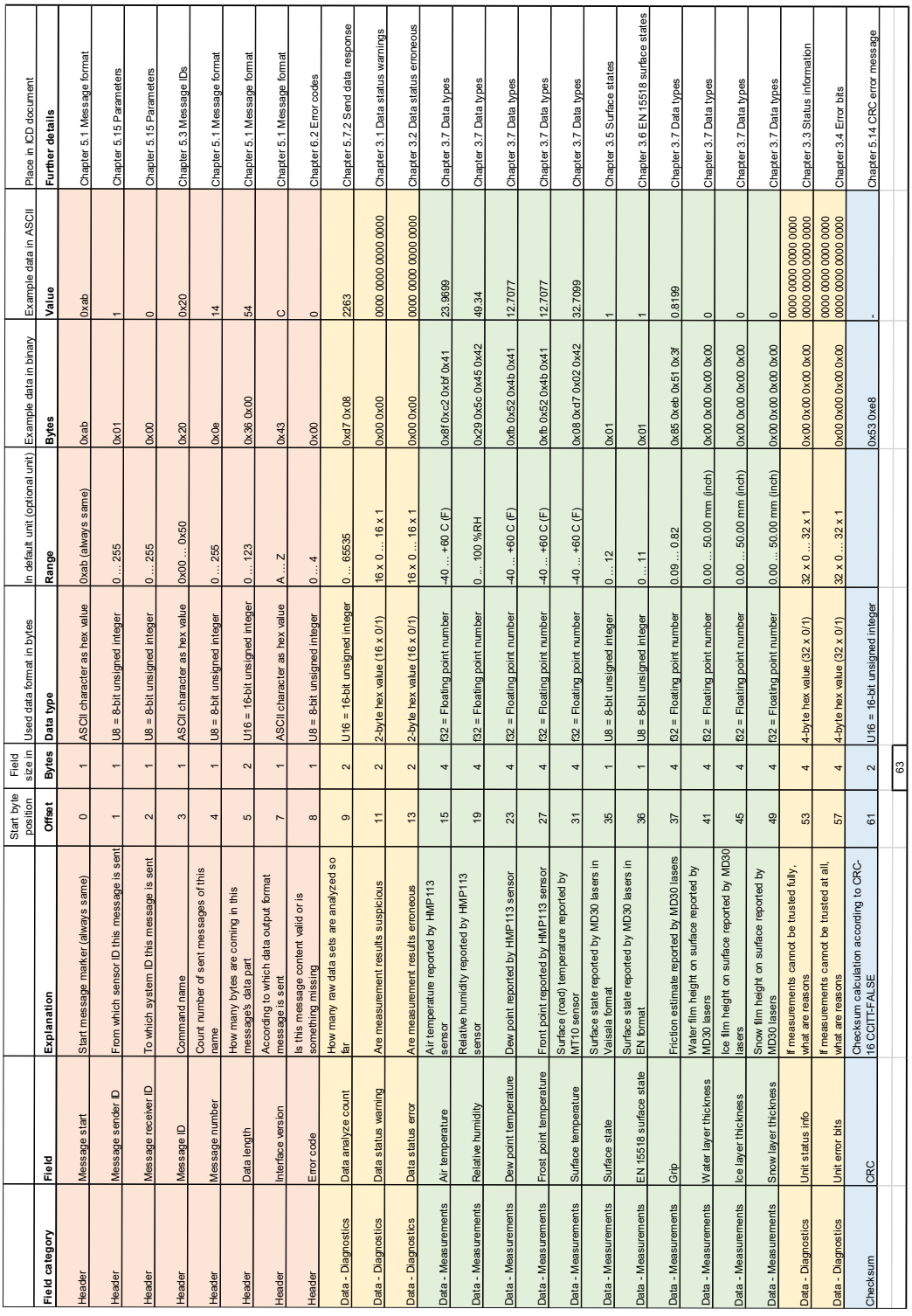
- Field: field name
- Explanation: short description of field contents
- Offset: start byte position
- Size in bytes: number of bytes in field
- Range: possible values for data, given in default unit (optional unit)
- Bytes: example data in binary
- Value: example data in ASCII

Field	Explanation	Offset	Size in bytes	Range	Bytes	Value	Details
Header							
Message start	Start message marker (always same)	0	1 [1)]	0xab (always same)	0xab	0xab	Message format (page 24)
Message sender ID	From which sensor ID this message is sent	1	1 [2)]	0 ... 255	0x01	1	Parameters (page 62)
Message receiver ID	To which system ID this message is sent	2	1 [2)]	0 ... 255	0x00	0	Parameters (page 62)
Message ID	Command name	3	1 [1)]	0x00 ... 0x50	0x20	0x20	Message IDs (page 27)
Message number	Count number of sent messages of this name	4	1 [2)]	0 ... 255	0x0e	14	Message format (page 24)
Data length	How many bytes are coming in this message's data part	5	2 [3)]	0 ... 123	0x36 0x00	54	Message format (page 24)
Interface version	According to which data output format message is sent	7	1 [1)]	A ... Z	0x43	C	Message format (page 24)

Field	Explanation	Offset	Size in bytes	Range	Bytes	Value	Details
Error code	Is this message content valid or is something missing	8	1 [2]	0 ... 4	0x00	0	Error codes (page 65)
Data - diagnostics							
Data analyze count	How many raw data sets are analyzed so far	9	2 [3]	0 ... 65535	0xd7 0x08	2263	SEND DATA response (page 39)
Data status warning	Are measurement results suspicious	11	2 [4]	16 x 0 ... 16 x 1	0x00 0x00	0000 0000 0000 0000	Data status warnings (page 11)
Data status error	Are measurement results erroneous	13	2 [4]	16 x 0 ... 16 x 1	0x00 0x00	0000 0000 0000 0000	Data status erroneous (page 11)
Data - measurements							
Air temperature	Air temperature reported by HMP113 sensor	15	4 [5]	−40 ... +60 C (F)	0x8f 0xc2 0xbf 0x41	23.9699	Data types (page 16)
Relative humidity	Relative humidity reported by HMP113 sensor	19	4 [5]	0 ... 100 %RH	0x29 0x5c 0x45 0x42	49.34	Data types (page 16)
Dew point temperature	Dew point reported by HMP113 sensor	23	4 [5]	−40 ... +60 C (F)	0xfb 0x52 0x4b 0x41	12.7077	Data types (page 16)
Frost point temperature	Front point reported by HMP113 sensor	27	4 [5]	−40 ... +60 C (F)	0xfb 0x52 0x4b 0x41	12.7077	Data types (page 16)
Surface temperature	Surface (road) temperature reported by MT10 sensor	31	4 [5]	−40 ... +60 C (F)	0x08 0xd7 0x02 0x42	32.7099	Data types (page 16)
Surface state	Surface state reported by MD30 lasers in Vaisala format	35	1 [2]	0 ... 12	0x01	1	Surface states (page 15)
EN 15518 surface state	Surface state reported by MD30 lasers in EN format	36	1 [2]	0 ... 11	0x01	1	EN 15518 surface states (page 16)

Field	Explanation	Offset	Size in bytes	Range	Bytes	Value	Details
Grip	Friction estimate reported by MD30 lasers	37	4 [5]	0.09 ... 0.82	0x85 0xeb 0x51 0x3f	0.8199	Data types (page 16)
Water layer thickness	Water film height on surface reported by MD30 lasers	41	4 [5]	0.00 ... 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Data types (page 16)
Ice layer thickness	Ice film height on surface reported by MD30 lasers	45	4 [5]	0.00 ... 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Data types (page 16)
Snow layer thickness	Snow film height on surface reported by MD30 lasers	49	4 [5]	0.00 ... 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Data types (page 16)
Data - diagnostics							
Unit status info	If measurements cannot be trusted fully, what are reasons	53	4 [6]	32 x 0 ... 32 x 1	0x00 0x00 0x00 0x00	0000 0000 0000 0000 0000 0000 0000 0000	Status information (page 12)
Unit error bits	If measurements cannot be trusted at all, what are reasons	57	4 [6]	32 x 0 ... 32 x 1	0x00 0x00 0x00 0x00	0000 0000 0000 0000 0000 0000 0000 0000	Error bits (page 14)
Checksum							
CRC	Checksum calculation according to CRC-16 CCITT-FALSE	61	2 [3]		0x53 0xe8	-	CRC error acknowledgment message (page 60)

1) *Used data format in bytes: ASCII character as hex value*
2) *Used data format in bytes: U8 = 8-bit unsigned integer*
3) *Used data format in bytes: U16 = 16-bit unsigned integer*
4) *Used data format in bytes: 2-byte hex value (16 x 0/1)*
5) *Used data format in bytes: f32 = Floating point number*
6) *Used data format in bytes: 4-byte hex value (32 x 0/1)*



			Start byte position	Field size in	Used data format in bytes		In default unit (optional unit)	Example data in binary	Example data in ASCII	Place in ICD document
Field category	**Field**	**Explanation**	**Offset**	**Bytes**	**Data type**		**Range**	**Bytes**	**Value**	**Further details**
Header	Message start	Start message marker (always same)	0	1	ASCII character as hex value		0xab (always same)	0xab	0xab	Chapter 5.1 Message format
Header	Message sender ID	From which sensor ID this message is sent	1	1	U8 = 8-bit unsigned integer		0 … 255	0x01	1	Chapter 5.15 Parameters
Header	Message receiver ID	To which system ID this message is sent	2	1	U8 = 8-bit unsigned integer		0 … 255	0x00	0	Chapter 5.15 Parameters
Header	Message ID	Command name	3	1	ASCII character as hex value		0x00 … 0x50	0x20	0x20	Chapter 5.3 Message IDs
Header	Message number	Count number of sent messages of this name	4	1	U8 = 8-bit unsigned integer		0 … 255	0x0e	14	Chapter 5.1 Message format
Header	Data length	How many bytes are coming in this message's data part	5	2	U16 = 16-bit unsigned integer		0 … 123	0x36 0x00	54	Chapter 5.1 Message format
Header	Interface version	According to which data output format message is sent	7	1	ASCII character as hex value		A … Z	0x43	C	Chapter 5.1 Message format
Header	Error code	Is this message content valid or is something missing	8	1	U8 = 8-bit unsigned integer		0 … 4	0x00	0	Chapter 6.2 Error codes
Data - Diagnostics	Data analyze count	How many raw data sets are analyzed so far	9	2	U16 = 16-bit unsigned integer		0 … 65535	0xd7 0x08	2263	Chapter 5.7.2 Send data response
Data - Diagnostics	Data status warning	Are measurement results suspicious	11	2	2-byte hex value (16 x 0/1)		16 x 0 … 16 x 1	0x00 0x00	0000 0000 0000 0000	Chapter 3.1 Data status warnings
Data - Diagnostics	Data status error	Are measurement results erroneous	13	2	2-byte hex value (16 x 0/1)		16 x 0 … 16 x 1	0x00 0x00	0000 0000 0000 0000	Chapter 3.2 Data status erroneous
Data - Measurements	Air temperature	Air temperature reported by HMP113 sensor	15	4	f32 = Floating point number		-40 … +60 C (F)	0x8f 0xc2 0xbf 0x41	23.9699	Chapter 3.7 Data types
Data - Measurements	Relative humidity	Relative humidity reported by HMP113 sensor	19	4	f32 = Floating point number		0 … 100 %RH	0x29 0x5c 0x45 0x42	49.34	Chapter 3.7 Data types
Data - Measurements	Dew point temperature	Dew point reported by HMP113 sensor	23	4	f32 = Floating point number		-40 … +60 C (F)	0xfb 0x52 0x4b 0x41	12.7077	Chapter 3.7 Data types
Data - Measurements	Frost point temperature	Front point reported by HMP113 sensor	27	4	f32 = Floating point number		-40 … +60 C (F)	0xfb 0x52 0x4b 0x41	12.7077	Chapter 3.7 Data types
Data - Measurements	Surface temperature	Surface (road) temperature reported by MT10 sensor	31	4	f32 = Floating point number		-40 … +60 C (F)	0x08 0xd7 0x02 0x42	32.7099	Chapter 3.7 Data types
Data - Measurements	Surface state	Surface state reported by MD30 lasers in Vaisala format	35	1	U8 = 8-bit unsigned integer		0 … 12	0x01	1	Chapter 3.5 Surface states
Data - Measurements	EN 15518 surface state	Surface state reported by MD30 lasers in EN format	36	1	U8 = 8-bit unsigned integer		0 … 11	0x01	1	Chapter 3.6 EN 15518 surface states
Data - Measurements	Grip	Friction estimate reported by MD30 lasers	37	4	f32 = Floating point number		0.09 … 0.82	0x85 0xeb 0x51 0x3f	0.8199	Chapter 3.7 Data types
Data - Measurements	Water layer thickness	Water film height on surface reported by MD30 lasers	41	4	f32 = Floating point number		0.00 … 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Chapter 3.7 Data types
Data - Measurements	Ice layer thickness	Ice film height on surface reported by MD30 lasers	45	4	f32 = Floating point number		0.00 … 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Chapter 3.7 Data types
Data - Measurements	Snow layer thickness	Snow film height on surface reported by MD30 lasers	49	4	f32 = Floating point number		0.00 … 50.00 mm (inch)	0x00 0x00 0x00 0x00	0	Chapter 3.7 Data types
Data - Diagnostics	Unit status info	If measurements cannot be trusted fully, what are reasons	53	4	4-byte hex value (32 x 0/1)		32 x 0 … 32 x 1	0x00 0x00 0x00 0x00	0000 0000 0000 0000	Chapter 3.3 Status information
Data - Diagnostics	Unit error bits	If measurements cannot be trusted at all, what are reasons	57	4	4-byte hex value (32 x 0/1)		32 x 0 … 32 x 1	0x00 0x00 0x00 0x00	0000 0000 0000 0000	Chapter 3.4 Error bits
Checksum	CRC	Checksum calculation according to CRC-16 CCITT-FALSE	61	2	U16 = 16-bit unsigned integer			0x53 0xe8	-	Chapter 5.14 CRC error message
				63						

Figure 7 MD30 data message description: response message for command SEND DATA

More information

- SEND DATA message (page 36)

Warranty and product returns

For standard warranty terms and conditions, see www.vaisala.com/warranty.

Please observe that any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. Please see the applicable supply contract or Conditions of Sale for details of the warranty for each product.

1. Read the warranty information.
2. Contact Vaisala technical support and request a Return Material Authorization (RMA) and shipping instructions.

Always request the RMA before returning any faulty material.

Provide the failure report as requested.

Technical support

Contact Vaisala technical support at helpdesk@vaisala.com. Provide at least the following supporting information as applicable:

- Product name, model, and serial number
- Software/Firmware version
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For more information, see www.vaisala.com/support.

Recycling

Recycle all applicable material.

Follow the statutory regulations for disposing of the product and packaging.

VAISALA

www.vaisala.com