« IIDRE: geolocation solution by UWB»

User guide

Review	Date	Description
1.0	12/11/2018	- Document creation.

IIDRE SAS
25 rue Marcel Issartier
33700 Mérignac
www.iidre.com

Table of contents

I. General description

The indoor geolocation solution provided by IIDRE uses ultra wideband (UWB) technology. This solution is based on exchanging signals between the mobile (also called tag) placed on the element to be tracked (vehicles, goods, persons) and the anchors preliminarily georeferenced in the tag progression area. The UWB signals exchanged (double-sided two-way ranging protocol for time of flight measurement) allow to determine the distances between the tag and each anchor. This then allows for a multilateration algorithm to compute the tag position onto a 2D plan from the measured distances and anchors coordinates.

The embedded intelligence in the tag allows for the IIDRE geolocation solution to ensure a good accuracy, even in dense environments, no matter the tag velocity.

The geolocation solution provided by IIDRE is easy to deploy. Furthermore, anchors can be added to the infrastructure during the runtime, thus giving a great scalability. These transceivers with range up to 150 meters, depending on the configuration and the environment, allows for wide coverage areas.

A dedicated application allows for the configuration of both UWB parameters and infrastructure. This application also displays tag motions on a map. Therefore, diagnostic information with characteristic features are available to determine the best configuration and troubleshoot the system.

II. Main characteristics

The main characteristics of the indoor geolocation solution provided by IIDRE are shown below:

- accuracy in the region of 10 cm,
- transceivers range up to 150 m in Line Of Sight (LOS),
- tag positions rate in the region of 30 Hz,
- wide area coverage,
- tracking of a fast moving tag (speed up to 30 km/h),
- adaptability to the surrounding environment (tunable UWB parameters),
- adaptability to the host system (data rate, coordinates format),
- Non-Line Of Sight (NLOS) measures management,
- embedded IMU (9-axis absolute orientation sensor BOSCH BNO055).

Thanks to those key features, the indoor geolocation solution provided by IIDRE could be used in several applications. For example; **RTLS** (automotive, robotics, UAV), **logistics** (goods tracking, data returns for maintenance) or **security** (people tracking, warnings in case of intrusion in a restricted area).

III. UWB parameters description

The DWM1000[1] module provided by Decawave and implemented in the IIDRE geolocation solution combines an integrated circuit (UWB transceiver compliant with IEEE 802.15.4-2011 standard), an UWB antenna and the associated circuitry. This module is entirely dedicated to RTLS applications and is fully configurable by users. The main tunable parameters are detailed in the following subsections.

A. Communication channel

The first parameter which can be tuned is the communication channel (CHAN). The DWM1000 module allows the user to choose among six channels presenting different bandwidths (from 500 to 900 MHz) and central frequencies (from 3.5 GHz to 6.5 GHz). The table 1 below shows the properties of each CHAN. By modifying the CHAN, users can adapt the solution to the surrounding electromagnetic environment. This is in order to avoid interferences with others transmitters in the same frequency range. Using a CHAN with a lower central frequency will increase the transceiver range, without reducing the performances in terms of ranging accuracy.

CHAN	Central frequency (MHz)	Bandwidth (MHz)
1	3,494.4	499.2
2	3,993.6	499.2
3	4,492.8	499.2
4	3,993.6	900
5	6,489.6	499.2
7	6,489.6	900

Table 1: communication channels parameters

It should be noted that tags and anchors must be configured on the same CHAN to ensure the smooth operation of the system. Devices are set by default to CHAN2.

B. Pulse repetition frequency

The user can also set the pulse repetition frequency (PRF). Two PRF values of 16 or 64 MHz are available. If two infrastructures are in use at the same time, they must share the same channel. Using different PRF values for each infrastructure allows them to cohabit independently. A PRF of 16 MHz marginally reduces the devices power consumption.

It should be noted that tags and anchors must be configured with the same PRF to ensure the smooth operation of the system. Devices are set by default to 64 MHz.

C. Preamble code

The preamble code (TRXcode) must be chosen depending on the two parameters CHAN and PRF. The combinations advocated by Decawave are summarized in the following table 2. When two devices are tuned to the same CHAN and PRF, the selection of two distinct TRXcode allows to separate communication channels.

It should be noted that tags and anchors must be configured with the same TRXcode to ensure the smooth operation of the system. Devices are set by default to TRXcode 9 in order to respect the coherence with the other default parameters.

1 « *DWM1000 Product Brief* » and « *DWM1000 Datasheet* » accessible in open download, in the *Design Center* tab, following this link http://www.decawave.com/.

CHAN	TRXcode (PRF = 16 MHz)	TRXcode (PRF = 64 MHz)
1	1-2	9-10-11-12
2	3-4	9-10-11-12
3	5-6	9-10-11-12
4	7-8	17-18-19-20
5	3-4	9-10-11-12
7	7-8	17-18-19-20

Table 2: TRXcode depending on both CHAN and PRF values

D. Baudrate

The baudrate (BR) can also be configured. From the three values (110 kbps, 850 kbps and 6.8 Mbps) initially available with the DWM1000 module, the IIDRE solution only implements 850 kbps and 6.8 Mbps. Modifying the BR value will influence the receiver sensitivity and the communications range. The table 3 below the presents data extracted from the technical documentation provided by Decawave.

Error rate	BR (kbps)	Receiver sensibility (dBm/ 500 MHz)
1 %	850	-101
	6,800	-93
10 %	850	-102
	6,800	-94

Table 3: error rate depending on both BR and receiver sensibility values (CHAN 5 and PRF of 16 MHz)

It should be noted that tags and anchors must be configured with the same BR to ensure the smooth operation of the system. Devices are set by default with a BR of 6.8 Mbps (PLEN and PAC are respectively set to 128 and 8). Although management of PLEN and PAC is automatic in the IIDRE geolocation solution, they can be set individually. It is advisable to respect the combinations recommended by Decawave which are summarized in the tables 4 and 5 below:

BR (kbps)	Recommended PLEN
850	256-512 or 1,024
6,800	64-128 or 256

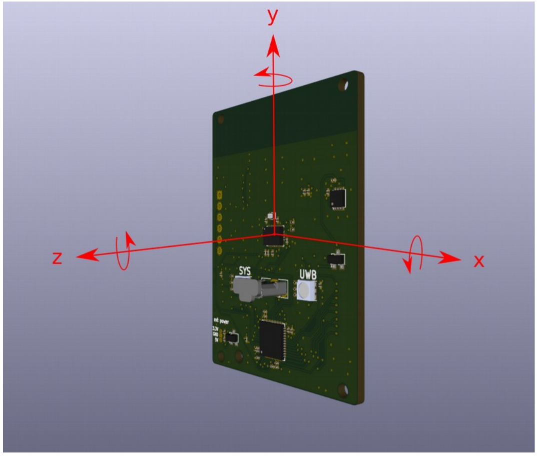
Table 4: recommended PLEN depending on BR

PLEN	Recommended PAC
64-128	8
256-512	16
1,024	32
1,536-2,048 or 4,096	64

Table 5: recommended PAC depending on PLEN

IV. Inertial Measurement Unit (IMU)

Devices integrate a 9-axis absolute orientation sensor BOSCH BNO055. The solution developed by IIDRE provides linear accelerations (m/s²), angular velocities (°/s), gravity vectors (m/s²) and heading (quaternion) according to 3 axis that are indicated on the figure below.



V. Supervision / configuration by AT commands

A set of commands based on AT commands principle allows for control and configuration of a device using a serial link. When a keyboard entry occurs, the user has three seconds to enter the following character. A command entry (not case-sensitive) always finishes by an action on the *enter* key. An error message appears in case of an unknown command entry, the system will then return to its normal operation.

Supervision commands allow the user to check the configuration of the elements in service and control the values of the UWB parameters. Commands are all based on the same model, i.e. **at+<parameter>?**.

Configuration commands allow the user to modify the current operation and the UWB parameters of a device. Commands are all based on the same model i.e. **at+<parameter>=<argument_1>, …,<argument_N>**. In case of a wrong entry (insufficient number of arguments or incoherent values), the system responds with an error message. Note that the modification of one of the UWB parameters is effective only after a reset on the device.

A. Device information

1. AT+ID

Read command

Gives the unique ID of the UWB chip on 8 hexadecimal characters and the device type, i.e. UNDEFINED, ANCHOR, MOBILE or GATEWAY.

```
AT+ID?
+ID:<DEVICE_UID>,<DEVICE_TYPE>
OK
```

2. AT+VER

Read command

Gives the current firmware version.

```
AT+VER?
+VER:<FIRMWARE_VERSION>
OK
```

B. Network configuration

1. AT+NODE

Infrastructure management by adding or removing anchors.

Read command

```
AT+NODE?
+NODE:<DEVICE_UID>,<DEVICE_TYPE>
OK
```

Write command

```
AT+NODE=<+ OR -><DEVICE_UID>,<DEVICE_TYPE>
OK
```

Parameters

- *DEVICE_UID:*
  - *+: add a node*
  - *-: remove a node*
- *DEVICE_TYPE:*
  - *0: UNDEFINED*
  - *1: ANCHOR*
  - *2: MOBILE*
  - *3: GATEWAY*

Examples

Add the node D4000E92 as an anchor into the infrastructure.

```
AT+NODE=+D4000E92,1
OK
```

Remove the node D4000E92 from the infrastructure. The second parameter is not needed in this case.

```
AT+NODE=-D4000E92
OK
```

Error

In case of the first character does not begin by + or -.

```
AT+NODE=D4000E92
+NODE=MUST BEGIN WITH '+' OR '-'
ERROR
```

2. AT+POS

Anchors coordinates management. In order to compute coherent positions, a tag must know the coordinates of every anchor.

Read command

Gives a list of the N elements that make up the infrastructure known by the device. Anchors ID are shown on 8 hexadecimal characters. Cartesian anchors coordinates are expressed in centimetres.

```
AT+POS?
+POS:<ANCHOR_1_UID>,<X_ANCHOR_1>,<Y_ANCHOR_1>,<Z_ANCHOR_1>
…
+POS:<ANCHOR_N_UID>,<X_ANCHOR_N>,<Y_ANCHOR_N>,<Z_ANCHOR_N>
OK
```

Write command

Allows to set the coordinates of an anchor known by the device.

```
AT+POS=<ANCHOR_UID>,<X_ANCHOR>,<Y_ANCHOR>,<Z_ANCHOR>
OK
```

Note that a simple error message appears in case of extra or missing arguments.

3. AT+NBTRY

Management of the number of attempts realized by the tag in order to initiate a communication process with an anchor.

Read command

Gives the number of attempts.

```
AT+NBTRY?
+NBTRY:<NUMBER_OF_ATTEMPTS>
OK
```

Write command

Allows to set the number of attempts. The value entered for this parameter must be a number strictly less than 256.

```
AT+NBTRY=<NUMBER_OF_ATTEMPTS>
OK
```

Error

In case of a wrong entry, the following error message appears.

```
+NBTRY:<NUMBER_OF_ATTEMPTS>
ERROR
```

C. Global UWB configuration

A set of AT commands allows for the configuration of the UWB parameters. Note that a new configuration becomes effective only after a reset on the device.

1. AT+CFG

Management of the overall UWB configuration of a device.

Read command

Gives the current UWB settings of the device[2].

```
AT+CFG?
+CFG:<CHAN>,<PRF>,<TRXCODE>,<BR>,<PLEN>,<PAC>,<TX_GAIN>
OK
```

Action command

Allows to restore the default UWB settings of the device.

```
AT+CFG
+CFG:2,64,9,6800,128,8,33
OK
```

Write command

Allows to set the overall UWB configuration of the device.

```
AT+CFG=<CHAN>,<PRF>,<TRXCODE>,<BR>,<PLEN>,<PAC>,<TX_GAIN>
OK
```

2. AT+CHAN

Management of the communication channel of a device.

Read command

Gives the current CHAN value.

```
AT+CHAN?
+CHAN:<CHANNEL>
OK
```

Write command

Allows the user to set the CHAN value. Possible values are provided in the table 1.

```
AT+CHAN=<CHANNEL>
OK
```

Errors

In case of a wrong entry, the following error message appears.

```
+CHAN:(1,2,3,4,5,7)
ERROR
```

Coherence of the settings are ensured by an automatic modification of the TRXcode if necessary (depending on the two parameters CHAN and PRF), the following error message appears in this case.

```
+CHAN: TRXCODE DEFAULT VALUE (<TRXCODE>) ACCORDING TO [CHANNEL,PRF]=[<CHAN>,<PRF>]
ERROR
```

2 The parameters are detailed in the Decawave datasheet « *DW1000 USER MANUAL* ». Furthermore, the application note APU001 « *CONFIGURING THE DW1000 FOR DATASHEET USE CASES* », recommends typical configurations depending on use cases.

3. AT+PRF

Management of the pulse repetition frequency of a device.

Read command

Gives the current PRF value.

```
AT+PRF?
+PRF:<PULSE_REPETITION_FREQUENCY>
OK
```

Write command

Allows the user to set the PRF value. Only the values 16 and 64 do not return errors.

```
AT+PRF=<PULSE_REPETITION_FREQUENCY>
OK
```

Errors

In case of a wrong entry, the following error message appears.

```
+PRF: (16-64)
ERROR
```

Coherence of the settings are ensured by an automatic modification of the TRXcode if necessary (depending on the couple of parameters CHAN and PRF), the following error message is then returned in this case.

```
+PRF: TRXCODE DEFAULT VALUE (<TRXCODE>) ACCORDING TO [CHANNEL,PRF]=[<CHAN>,<PRF>]
ERROR
```

4. AT+TRXCODE

Management of the preamble code of a device.

Read command

Gives the current TRXcode value.

```
AT+TRXCODE?
+TRXCODE:<PREAMBLE_CODE>
OK
```

Write command

Allows the user to set the TRXcode value. Possible values according to the two parameters CHAN and PRF are detailed in the table 2.

```
AT+TRXCODE=<PREAMBLE_CODE>
OK
```

Errors

In case of a wrong entry, the following error message appears.

```
+TRXCODE: (1,2,3,4,5,6,7,8,9,10,11,12,17,18,19,20)
ERROR
```

In case of an inconsistency with the current configuration, an error message indicates the compliant values.

```
+TRXCODE: MUST BE (<VALUES>) ACCORDING TO [CHANNEL,PRF]=[<CHAN>,<PRF>]
ERROR
```

5. AT+BR

Management of the baudrate of a device.

Read command

Gives the current BR value.

```
AT+BR?
+BR:<BAUDRATE>
OK
```

Write command

Allows the user to set the BR value. Only 850 and 6,800 do not return errors.

```
AT+BR=<BAUDRATE>
OK
```

Errors

In case of wrong entry, the following error message appears.

```
+BR: (850,6800)
ERROR
```

Coherence of the settings is ensured by an automatic modification of PLEN and PAC if necessary (see tables 4 and 5 for further information). The following error message is then returned in this case.

```
+BR: PLEN AND PAC DEFAULT VALUES (<PLEN>,<PAC>) ACCORDING TO BR (<BR>)
ERROR
```

6. AT+PWR

Management of the emisison gain of a device.

Read command

Gives the current emission gain value, expressed in dBi.

```
AT+PWR?
+PWR:<EMISSION_GAIN>
OK
```

Write command

Allows the user to set the device emission gain. The value entered for this parameter must be a number strictly less than or equal to 33.

```
AT+PWR=<TX_GAIN>
OK
```

Error

In case of a wrong entry, the following error message appears.

```
+PWR: MUST BE ≤ 33DBI
ERROR
```

D. Device settings

A set of AT commands allowing the configuration of the device settings.

1. AT+NBDIST

Management of the number of distances successively measured with an anchor before moving on to the next one.

Read command

Gives the number of distances.

```
AT+NBDIST?
+NBDIST:<NUMBER_OF_DISTANCES>
OK
```

Write command

Allows the user to set the number of distances. The value entered for this parameter must be a number strictly inferior or equal to 12.

```
AT+NBDIST=<NUMBER_OF_DISTANCES>
OK
```

Error

In case of a wrong entry, the following error message appears.

```
AT+NBDIST:MUST BE ≤ 12
ERROR
```

2. AT+FMT

Management of the format of the positions computed by the device. The user can choose between cartesian (CART), decimal degree (GPS_DD) or degrees and minutes (GPS_DDMM).

Read command

Gives the current tag positions format.

```
AT+FMT?
+FORMAT:<POSITIONS_FORMAT>
OK
```

Write command

Allows the user to set the tag positions format.

```
AT+FMT=<POSITIONS_FORMAT>
OK
```

Parameters

- *POSITIONS_FORMAT:*
  - *0: CART*
  - *1: GPS_DD*
  - *2: GPS_DDMM*

3. AT+SETACC

Management of the offsets applied on the data coming from the accelerometer. The offset values must be in m/s^2. Note that a fourth optional parameter allows to adjust the sensor position (in cm).

Read command

Gives the current offsets and radius.

```
AT+SETACC?
+SETACC:<OFFSET_X>,<OFFSET_Y>,<OFFSET_Z>,<OFFSET_R>
OK
```

Write command

Allows the user to set both the linear accelerations offsets and the radius.

```
AT+SETACC=<OFFSET_X>,<OFFSET_Y>,<OFFSET_Z>[,<OFFSET_R>]
OK
```

4. AT+SETGYRO

Management of the offsets applied on the data coming from the gyroscope. The offset values must be in °/s.

Read command

Gives the current offsets.

```
AT+SETGYRO?
+SETGYRO:<OFFSET_X>,<OFFSET_Y>,<OFFSET_Z>
OK
```

Write command

Allows the user to set the angular velocities offsets.

```
AT+SETGYRO=<OFFSET_X>,<OFFSET_Y>,<OFFSET_Z>
OK
```

E. Antenna delays calibration[3]

Manual and automatic management of the antenna delays of the elements that make up the infrastructure. The antenna delays are expressed in the internal Decawave unit. These values could be converted in the time domain by applying the following transformation: 1/(128*499.2e6).

1. AT+CALIB

Read command

Returns the current antenna delays values for each node of the network.

```
AT+CALIB?
+CALIB:<ANCHOR_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
…
+CALIB:<ANCHOR_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
OK
```

Write command

Allows the user manually set the antenna delays values of an anchor known by the tag.

```
AT+CALIB:<ANCHOR_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
OK
```

3 See Decawave application note APS014 and datasheet for further information.

Example

The read command described above is also available by replacing the keyword *calib*, with *caliba* or *calibm*.

```
AT+CALIBM?
+CALIB:<ANCHOR_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
…
+CALIB:<ANCHOR_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
OK
```

2. AT+CALIBA

Write command

Allows for an automatic calibration of every device that makes up the infrastructure from the exact tag position. Note that the anchors coordinates must have been set into the flash memory before use. The value entered for this command must be expressed in cm.

```
AT+CALIBA=<X_MOBILE>,<Y_MOBILE>,<Z_MOBILE>
OK
```

It is also possible to apply this automatic calibration process only on a part of the anchors (up to seven) by adding the corresponding UID at the end of this command.

```
AT+CALIBA=<X_MOBILE>,<Y_MOBILE>,<Z_MOBILE>,<ANCHOR_UID_1>,…,<ANCHOR_UID_N>
OK
```

3. AT+CALIBM

Write command

Allows for an automatic calibration of a single device from the distance between the anchor and the tag.

```
AT+CALIBM=<ANCHOR_UID>,<DISTANCE>
OK
```

F. Device output messages

Management of the device output messages. See chapter VI for further details.

1. AT+TRACE

Management of the messages sent on the serial link.

Read command

Gives the number and the list of the N registered traces.

```
AT+TRACE?
+TRACE:<NUMBER_OF_TRACES>,<TRACE_1>,…,<TRACE_N>
OK
```

Write command

Allows for the log management by addition or removal of messages.

```
AT+TRACE=<+ OR -><TRACE>
OK
```

Removal of all the messages sent on the serial link.

```
AT+TRACE=0
OK
```

Parameters

- *TRACE:*
  - *+ add a trace*
  - *- remove a trace*
  - *DIST: displays a report of a DS2WR process (filtered data)*
  - *DIST_DBG: displays a report of a DS2WR process (raw data)*
  - *MPOS: displays the timestamped mobile position*
  - *MACC: displays the timestamped mobile linear acceleration*
  - *MGYRO: displays the timestamped mobile angular velocity*
  - *MGVT: displays the timestamped mobile gravity vector*
  - *MQUAT: displays the timestamped mobile heading (quaternion)*
  - *DPOS: displays the timestamped deported mobile position*
  - *DIMU: displays the timestamped deported IMU data*

Examples

```
AT+TRACE=+MGYRO
OK
```

Add the frame *MGYRO* onto the serial link.

```
AT+TRACE=-MACC
OK
```

Remove the frame *MACC* from the serial link.

Error

If the first character does not begin with + or -, or if the entered field does not correspond to one of those described in the chapter VI, the following error message appears.

```
AT+TRACE=DPOS
+TRACE:PARAM MUST BEGIN BY '+' '-'
ERROR
```

2. AT+TIME

Management of the messages timestamp. The default timestamp source is based on the embedded system clock and is expressed in ms. Thus the user can synchronize the messages with its host system.

Read command

Gives the current offset.

```
AT+TIME?
+TIME:<OFFSET_TIME>
OK
```

Write command

Manual adjustment of the offset.

```
AT+TIME=<OFFSET_TIME>
OK
```

G. Miscellaneous

1. AT+RESET

Write command

Reset the device attached to the serial link.

```
AT+RESET=<RESET_TYPE>
OK
```

Parameters

- *RESET_TYPE:*
  - *0: reset device*
  - *1: reset device, firmware update*
  - *2: reset device, bootloader mode*

VI. Unsolicited responses

Description of the messages available at the device output.

A. +DIST

Message with the exchanges synthesis (filtered data) between a tag and one anchor. The list of the +DIST messages are send only if a position is computed by the system.

```
+DIST:<TMSTP>,<ANCHOR_UID>,<DIST>,<X>,<Y>,<Z>,<FP_PWRLVL>,<IDIFF>,<MC>
```

Parameters

- *TMSTP: message emission timestamp (ms)*
- *ANCHOR_UID: anchor ID on 8 hexadecimal characters*
- *DIST: distance between the tag and the corresponding anchor (cm)*
- *X,Y,Z: corresponding anchor coordinates (cm)*
- *FP_PWRLVL: first-path (*1000) power (dBm)*
- *IDIFF: LOS/NLOS communication indicator, without unit*
- *MC: LOS/NLOS communication indicator (*10000), without unit*

B. +DIST_DBG

Message with the exchanges synthesis (raw data) between the tag and one anchor.

```
+DIST_DBG:<TMSTP>,<ANCHOR_UID>,<DIST>,<X>,<Y>,<Z>,<FP_PWRLVL>,<IDIFF>,<MC>
```

Parameters

- *TMSTP: message emission timestamp (ms)*
- *ANCHOR_UID: anchor ID on 8 hexadecimal characters*
- *DIST: distance between the tag and the corresponding anchor (cm)*
- *X,Y,Z: corresponding anchor coordinates (cm)*
- *FP_PWRLVL: first-path (*1000) power (dBm)*
- *IDIFF: LOS/NLOS communication indicator, without unit*
- *MC: LOS/NLOS communication indicator (*10000), without unit*

C. +MPOS

Message with the tag position.

```
+MPOS:<TMSTP>,<X_MOBILE>,<Y_MOBILE>,<Z_MOBILE>
```

Parameters

- *TMSTP: message emission timestamp (ms)*
- *X_MOBILE: X-axis tag coordinate (cm)*
- *Y_MOBILE: Y-axis tag coordinate (cm)*
- *Z_MOBILE: Z-axis tag coordinate (cm)*

D. +MACC

Message with the tag linear accelerations.

```
+MACC:<TMSTP>,<ACC_DATA_X>,<ACC_DATA_Y>,<ACC_DATA_Z>
```

Parameter

- *TMSTP: data reading timestamp from the IMU (ms)*
- *ACC_DATA_X: X-axis linear acceleration (*100) (m/s^2)*
- *ACC_DATA_Y: Y-axis linear acceleration (*100) (m/s^2)*
- *ACC_DATA_Z: Z-axis linear acceleration (*100) (m/s^2)*

E. +MGYRO

Message with the tag angular velocities.

```
+MGYRO:<TMSTP>,<GYR_DATA_X>,<GYR_DATA_Y>,<GYR_DATA_Z>
```

Parameters

- *TMSTP: data reading timestamp from the IMU (ms)*
- *GYR_DATA_X: X-axis angular velocity (*16) (°/s)*
- *GYR_DATA_Y: Y-axis angular velocity (*16) (°/s)*
- *GYR_DATA_Z: Z-axis angular velocity (*16) (°/s)*

F. +MGVT

Message with the tag gravity vector.

```
+MGVT:<TMSTP>,<GRV_DATA_X>,<GRV_DATA_Y>,<GRV_DATA_Z>
```

Parameters

- *TMSTP: data reading timestamp from the IMU (ms)*
- *GRV_DATA_X: X-axis gravity (*100) (m/s^2)*
- *GRV_DATA_Y: Y-axis gravity (*100) (m/s^2)*
- *GRV_DATA_Z: Z-axis gravity (*100) (m/s^2)*

G. +MQUAT

Message with the tag heading (quaternion).

```
+MQUAT:<TMSTP>,<QUAT_DATA_W>,<QUAT_DATA_X>,<QUAT_DATA_Y>,<QUAT_DATA_Z>
```

Parameters

- *TMSTP: reading data timestamp from the IMU (ms)*
- *QUAT_DATA_W: W parameter of the quaternion (*214), without unit*
- *QUAT_DATA_X: X parameter of the quaternion (*214), without unit*
- *QUAT_DATA_Y: Y parameter of the quaternion (*214), without unit*
- *QUAT_DATA_Z: Z parameter of the quaternion (*214), without unit*

H. +DPOS

Message with the tag position. This message is accessible from the devices known as ANCHOR or GATEWAY.

```
+DPOS:<TMSTP>,<MOBILE_UID>,<X_MOBILE>,<Y_MOBILE>,<Z_MOBILE>,
  <ANCHOR_UID>,<X_ANCHOR>,<Y_ANCHOR>,<Z_ANCHOR>,
  <DIST>,<WEIGHT>,<RX_PWRLVL>
```

Parameters

- *TMSTP: message emission timestamp (ms)*
- *MOBILE_UID: tag ID on 8 hexadecimal characters*
- *X_MOBILE,Y_MOBILE,Z_MOBILE: tag coordinates (cm)*
- *ANCHOR_UID: anchor ID on 8 hexadecimal characters*
- *X_ANCHOR,Y_ANCHOR,Z_ANCHOR: anchor coordinates (cm)*
- *DIST: distance between <mobile_UID> and <anchor_UID> (cm)*
- *WEIGHT: LOS/NLOS communication indicator, without unit*
- *RX_PWRLVL: signal power reception (dBm)*

I. +DIMU

Message with tag inertial data. This message is accessible from the devices known as ANCHOR or GATEWAY.

```
+DIMU:<TMSTP>,<MOBILE_UID>,<ACC_DATA_X>,<ACC_DATA_Y>,<ACC_DATA_Z>,
   <GYR_DATA_X>,<GYR_DATA_Y>,<GYR_DATA_Z>,
   <GRV_DATA_X>,<GRV_DATA_Y>,<GRV_DATA_Z>
```

Parameters

- *TMPSTP: message emission timestamp (ms)*
- *MOBILE_UID: tag ID on 8 hexadecimal characters*
- *ACC_DATA_X: X-axis linear acceleration (*100) (m/s2)*
- *ACC_DATA_Y: Y-axis linear acceleration (*100) (m/s2)*
- *ACC_DATA_Z: Z-axis linear acceleration(*100) (m/s2)*
- *GYR_DATA_X: X-axis angular velocity (*16) (°/s)*
- *GYR_DATA_Y: Y-axis angular velocity (*16) (°/s)*
- *GYR_DATA_Z: Z-axis angular velocity (*16) (°/s)*
- *GRV_DATA_X: X-axis gravity (*100) (m/s^2)*
- *GRV_DATA_Y: Y-axis gravity (*100) (m/s^2)*
- *GRV_DATA_Z: Z-axis gravity (*100) (m/s^2)*