« IIDRE: geolocation solution by UWB »

User guide

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 signals exchanged between mobiles (also called tags) placed on the elements to be tracked (vehicles, goods, persons) and a set of anchors preliminarily georeferenced in the tags progression area. The UWB signals exchanged (double-sided two-way ranging protocol for time of flight measurement) make possible to determine the distances between the tags and each anchor. This then allows for a multilateration algorithm to compute the tags positions onto a 2D plan from both measured distances and anchors coordinates.

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

The geolocation solution provided by IIDRE is easy to deploy. Furthermore, anchors can be added to the infrastructure during runtime, thus giving a great scalability. Wide areas may be covered thanks to transceivers achieving a high range of up to 150 meters, depending on the configuration and the surrounding environment.

A dedicated application allows for the configuration of both UWB parameters and infrastructure. This application also displays tags motions on a map. Therefore, diagnostic information with characteristic features are available to determine the best configuration and troubleshoot the system (histograms of the distances relative to each anchors, number of distances measured, RF quality, ...). Note that another application tailored for asset tracking have also been developed to help the user to process the data monitored thanks to devoted metrics, such as: the covered distance, the average speed, an analysis of the stops (number, duration and location) and a heatmap generation.

II. Main characteristics

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

- accuracy in the region of 10-30 cm,
- transceivers range up to 150 m in Line Of Sight (LOS),
- tag positions rate up to 50 Hz in the case of a minimalist number of anchors,
- 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, power supply),
- 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** (assets tracking, data returns for maintenance), **traceability** or **security** (people tracking, warning in case of safety perimeters crossings, warning in case of intrusions in restricted areas).

III. UWB parameters

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. The main tunable parameters are detailed in the following subsections.

1. 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 (tags and anchors must be configured on the same CHAN to ensure the smooth operation of the system). Devices are set by default on CHAN2.

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

2. 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 cannot be set on different channels. Using different PRF values for each infrastructure allows them to cohabit independently. A PRF of 16 MHz marginally reduces the devices power consumption (tags and anchors must be configured on the same PRF to ensure the smooth operation of the system). Devices are set by default to 64 MHz.

3. Preamble code

The preamble code (TRXcode) must be chosen depending on the two parameters CHAN and PRF described above. 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 (tags and anchors must be configured on the same TRXcode to ensure the smooth operation of the system). Devices are set by default on TRXcode 9 in order to respect the coherence with the other default parameters.

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

[1] *«DWM1000 Product Brief»* and *«DWM1000 Datasheet»* accessible in open download (www.decawave.com).

4. Baudrate

The baudrate (BR) can also be configured. Modifying the BR value will influence the receiver sensitivity and then communications range. The table 3 below presents the 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 sensitivity (CHAN=5 / PRF=16 MHz)

It should be noted that tags and anchors must be configured on the same BR to ensure the smooth operation of the system. Devices are set by default with a BR of 850 kbps (PLEN and PAC are respectively set to 512 and 16). Although management of PLEN and PAC are automatic in the IIDRE geolocation solution, they can be set individually. It is advisable to respect the combinations recommended by Decawave and summarized in the tables 4 and 5 below:

BR (kbps)	Recommended PLEN
110	1,024-2,048-4,096
850	256-512-1,024
6,800	64-128-256

Table 4: recommended PLEN depending on BR

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

Table 5: recommended PAC depending on PLEN

IV. Inertial measurement unit (IMU)

1. IMU presentation

Devices may integrate a 9-axis absolute orientation sensor BOSCH BNO055 [2]. Therefore, in this case, the solution developed by IIDRE provides linear accelerations (m/s^2), angular velocities (°/s), gravity vectors (m/s^2) and heading, expressed in Euler angles (°) or in terms of a quaternion, according to 3 axis such as indicated on the figure below.

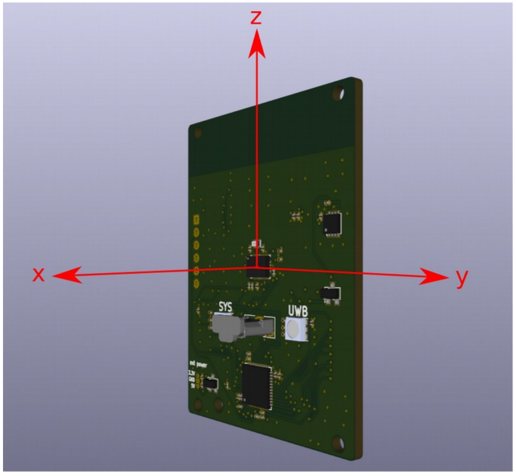

Figure 1: IMU orientation.

2. IMU calibration

If the device integrates an IMU, a calibration may be requested when the device is powered for the first time or also after a reset performed on the flash memory. Note that the user can skip this process by entering the appropriate command (described in the following part of this document). This calibration consists in a short sequence of movements applied on the device. It is possible to follow the progress of the calibration process thanks to the messages displayed on the serial line of the device. The user is kindly invited to refer to the datasheet of the IMU for further details (mode nine degrees of freedom with fast magnetic calibration off). Note also that the manufacturer of the component proposes a video tutorial [3] specifically dedicated to *"How to calibrate the absolute orientation sensor BNO055"*.

a. Accelerometer

In order to calibrate the accelerometer, the user has to rotate in 45° increments at least across on axis. For example, it is possible to start with the device face down on a table, then to apply the first 45° rotation clockwise across the Z-axis and continue until the accelerometer is fully calibrated.

b. Gyroscope

The gyroscope achieves calibration when the device keeps stationary a few seconds.

c. Magnetometer

The magnetometer achieves calibration when the user applies some rotations describing a figure such as an 8 in space with the device.

[2] « *BNO055 Intelligent 9-axis absolute orientation sensor* » accessible in open download (www.bosch-sensortec.com).
[3] « *How to calibrate the absolute orientation sensor BNO055* » video tutorial visible here www.youtube.com.

V. Configuration / supervision 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.

1. Device information

A. 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, GATEWAY or MESHDIST. Following is a brief description of each device type:

- UNDEFINED: default device type.
- ANCHOR: device type for fixed elements of the infrastructure during RTLS runtime.
- MOBILE: device type for the tags during RTLS runtime.
- GATEWAY: device type for data centralization. In this case, the device does not take part to the ranging process, it simply acts as a listener for deported messages.
- MESHDIST: device type for distances measurement. Each device configured in this mode successively determines the range from the other devices also known as MESHDIST and reports the corresponding distances in a dedicated message.

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

B. AT+VER

Read command

Gives information about the firmware, i.e. the current version and the date and time of the building. This command also gives information on the bootloader version and lists the various compilation options activated. Following is a brief description of the most important ones:

- USB_CDC: the device communicates via the Universal Serial Bus.
- UART: the device communicates via the Universal Asynchronous Receiver Transmitter.
- LP_DW1000: the device runs with the low-power mode activated.
- I2C: the device communicates with the embedded IMU (if equipped).
- WLS: the tag computes its positions by means of a Weighted Least Square (WLS) algorithm instead of a LS algorithm. The weight depends on the radio quality measured during the ranging process.

```
AT+VER?
+VER:<FW_VERSION>,<BOOTLOADER_VERSION>,<DATE>,<TIME>,<OPTION_1>,...,<OPTION_N>
OK
```

2. Network configuration

A. AT+NODE

Device type management. In order to work properly, the role of each device involved must be configured.

Read command

Gives a list of the N devices known by the device. Each device UID is shown on 8 hexadecimal characters while each device type is given with a corresponding number, i.e. 0, 1, 2, 3 or 4 respectively standing for UNDEFINED, ANCHOR, MOBILE, GATEWAY and MESHDIST.

```
AT+NODE?
+NODE:<DEVICE_1_UID>,<DEVICE_1_TYPE>
...
+NODE:<DEVICE_N_UID>,<DEVICE_N_TYPE>
OK
```

Write command

Allows the user to assign the desired type to a device and to set the infrastructure. The symbols + and − are respectively used to add or remove a node. The parameter DEVICE_UID corresponds to the unique ID of the device to add or remove while the parameter DEVICE_TYPE must be entered with its corresponding number as indicated above.

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

A shortcut is available to set the device type without having to enter its UID.

```
AT+NODE=+me,<DEVICE_TYPE>
OK
```

A shortcut is available to remove all nodes known by the device instantly.

```
AT+NODE=0
OK
```

Example 1: manage a RTLS infrastructure

Here is a quick example with the configuration steps for an RTLS application involving 2 tags (0x10000001 and 0x10000002) in an infrastructure made up of 4 anchors (0x1000000A, 0x1000000B, 0x1000000C and 0x1000000D) and 1 gateway (0x10005000) for data monitoring. The first step consists in the configuration of the tags. To that end, the following commands must be entered on both devices. Note that it is imperative to enter these commands in the same order on each tag. The first MOBILE of the list will act as a *master*, it will manage the sharing of a virtual *talking stick* allowing for the synchronization of the tags during the ranging process. This implies that the *master* must always be powered for the proper functionning of the solution.

```
AT+NODE=+10000001,2
AT+NODE=+10000002,2
AT+NODE=+1000000A,1
AT+NODE=+1000000B,1
AT+NODE=+1000000C,1
AT+NODE=+1000000D,1
AT+NODE=+10005000,3
```

Then the second step consists in the configuration of the anchors. The user is free to let the default device type (UNDEFINED) on each device or to enter the right type (ANCHOR) as indicated below with the first anchor of the infrastructure.

```
AT+NODE=+1000000A,1
OK
```

The final step consists in the configuration of the gateway. Again the user is free to let the default device type (UNDEFINED) or to set the right type (GATEWAY) as shown below.

```
AT+NODE=+10005000,3
OK
```

Example 2: remove a tag from the infrastructure

Using the previous example, if the user wants to put in service only one tag (0x10000001), it is possible to remove the second tag from the list. If a single tag is used, the role of *master* becomes obsolete, the solution works properly without the *talking stick*.

```
AT+NODE=-10000002,2
OK
```

Note that the parameter DEVICE_TYPE is not necessary in the case of a node removed from the infrastructure. The following command would have the same effect.

```
AT+NODE=-10000002
OK
```

Example 3: manage a infrastructure in MESHDIST mode

Here is a quick example for the configuration of 4 devices acting as MESHDIST (0x10000001, 0x10000002, 0x10000003 and 0x10000004). In this case, each device measures successively the distance from each other device in the infrastructure. The first device of the list will act as a *master*, it will manage the sharing of a virtual *talking stick* allowing for the synchronization of the other devices during the ranging process. This implies that the *master* must always be powered for the proper functionning of the solution. Note that the following commands must be entered on each device in service.

```
AT+NODE=+10000001,4
AT+NODE=+10000002,4
AT+NODE=+10000003,4
AT+NODE=+10000004,4
```

Error

In case of the first character is not + or −, the following error message appears.

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

B. AT+POS

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

Read command

Gives a list of the N anchors that make up the infrastructure known by the device. Anchors UIDs 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
```

Example 1: enter the anchors coordinates in the example *manage a RTLS infrastructure*

Using the previous example, if the coordinates of the 4 anchors are respectively set to (0,0,100), (500,0,100), (500,500,100) and (0,500,100), the following commands must be entered on the 2 tags.

```
AT+POS=1000000A,0,0,100
AT+POS=1000000B,500,0,100
AT+POS=1000000C,500,500,100
AT+POS=1000000D,0,500,100
```

3. 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.

A. AT+CFG

Management of the overall UWB configuration of a device.

Read command

Gives the current UWB settings of the device[4].

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

Action command

Allows the user to restore the default UWB settings of the device.

```
AT+CFG
+CFG:2,64,9,850,512,16,33
OK
```

Write command

Allows the user to set the overall UWB configuration of the device.

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

B. 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 [CHAN,PRF]=[<CHAN>,<PRF>]
ERROR
```

[4] 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.

C. 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 an error message.

```
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 [CHAN,PRF]=[<CHAN>,<PRF>]
ERROR
```

D. 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 [CHAN,PRF]=[<CHAN>,<PRF>]
ERROR
```

E. 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 the values 110, 850 and 6,800 do not return an error message.

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

Errors

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

```
+BR: (110,850,6800)
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
```

F. AT+PWR

Management of the emission 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.

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

Errors

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

```
+PWR: MUST BE > 0 AND <= 33 DBI
ERROR
```

4. Device settings

A set of AT commands allows for the configuration of additional device settings.

A. AT+IMU

Inertial measurement unit management.

Read command

Gives the current offsets and status of each component of the embedded IMU (accelerometer, gyroscope and magnetometer). The calibration statutes CLB_STAT_ACC, CLB_STAT_GYR, CLB_STAT_MAG and CLB_STAT_SYS are given with a corresponding number, i.e. 0, 1, 2, 3 or 4 respectively standing for not calibrated, calibration in progress (1, 2), calibrated and calibrated with stored data. The status of the embedded IMU set by the user and indicated with the field IMU_STAT is given with different keywords, i.e. RESET, DISABLE, CALIB and FORCE.

- RESET: the embedded IMU is disabled and the current offsets have been set to 0.
- DISABLE: the embedded IMU is disabled.
- CALIB: the embedded IMU is enabled and the current offsets have been determined by a calbibration process (see part *IMU calibration* of this document for further information).
- FORCE: the embedded IMU is enabled and the current offsets have been manually set by the user.

```
AT+IMU?
+IMU:<ACC_OFFSET_R>,<ACC_OFFSET_X>,<ACC_OFFSET_Y>,<ACC_OFFSET_Z>
    ,<GYR_OFFSET_X>,<GYR_OFFSET_Y>,<GYR_OFFSET_Z>
    ,<MAG_OFFSET_R>,<MAG_OFFSET_X>,<MAG_OFFSET_Y>,<MAG_OFFSET_Z>
    ,<CLB_STAT_ACC>,<CLB_STAT_GYR>,<CLB_STAT_MAG>,<CLB_STAT_SYS>
    ,<IMU_STAT>
OK
```

Write command

Allows the user to manage the embedded IMU status among RESET, DISABLE, CALIB and FORCE. In case of the value is FORCE, the user can enter the offsets values. The user is free to left blank the parameter between two commas, in this case the corresponding value remains unchanged.

```
AT+IMU=<IMU_STAT>
      [,<ACC_OFFSET_R>,<ACC_OFFSET_X>,<ACC_OFFSET_Y>,<ACC_OFFSET_Z>
       ,<GYR_OFFSET_X>,<GYR_OFFSET_Y>,<GYR_OFFSET_Z>
       ,<MAG_OFFSET_R>,<MAG_OFFSET_X>,<MAG_OFFSET_Y>,<MAG_OFFSET_Z>]
OK
```

B. 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:<POSITION_FORMAT>
OK
```

Write command

Allows the user to set the tag positions format, each POSITION_FORMAT is given with a corresponding number, i.e. 0, 1 or 2 respectively standing for CART, GPS_DD and GPS_DDMM.

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

5. Antenna delays calibration

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).

A. AT+CALIB

Read command

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

```
AT+CALIB?
+CALIB:<ANCHOR_1_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
...
+CALIB:<ANCHOR_N_UID>,<ANT_DELAY_RX>,<ANT_DELAY_TX>
OK
```

Write command

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

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

B. 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 centimetres.

```
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_1_UID>,...,<ANCHOR_N_UID>
OK
```

C. 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
```

6. Device output messages

Management of the device output messages. See chapter VI for further details on each message type.

A. 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 X-Y-(Z) coordinates and velocities
  - POS_INFO: displays a report of the timestamps of the rangings used to compute the current position
  - MESH: displays the timestamped distances of each device configured as MESHDIST
  - VDD: displays the timestamped power voltage
  - 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)
  - MHRP: displays the timestamped mobile heading (Euler angles)
  - DPOS: displays the timestamped deported mobile position
  - DIMU: displays the timestamped deported IMU data
  - DVDD: displays the timestamped deported power voltage

Example 1: monitoring on a tag during RTLS

Using the previous example (*manage a complete RTLS infrastructure*), if the user wants to monitor what happens on the *master* tag (UID: 10000001), it is possible to keep an eye on the distances measured from each anchors (+DIST), the positions computed from both measurements and anchors' coordinates (+MPOS), the heading (+MQUAT) and some inertial data (+MACC). In this case, the following commands must be entered on the tag.

```
AT+TRACE=+DIST
AT+TRACE=+MPOS
AT+TRACE=+MQUAT
AT+TRACE=+MACC
```

Example 2: deported monitoring on a gateway during RTLS

Using the example described above, if the user preferes to monitor what happens on a gateway (UID: 10005000) rather than on a tag, it is possible to keep an eye on the distances and position (+DPOS) coming from one or several tags. In this case, the following command must be entered on the gateway and also on each tag being monitored.

```
AT+TRACE=+DPOS
```

Example 3: monitoring on a device during MESHDIST mode

Using the previous example (*manage a complete infrastructure in MESHDIST mode*), if the user wants to monitor the activity on the network, it is necessary to enter the following command on each device involved.

```
AT+TRACE=+MESH
```

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: MUST BEGIN WITH '+' OR '-'
ERROR
```

B. 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
```

7. Miscellaneous

A. 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
  - . 3: reset device, restore default settings for UWB configuration and active traces

B. AT+RCMD

Send commands to remote devices.

Write command

Allows the user to send the commands detailed previously to distant devices. The command could either be specifically targeted with the RECEIVER_UID or broadcasted by using the value 0xFFFFFFFF. This command is then completed with the desired command by following the specific rules given in the previous subsections, removing the characters *at*. Note that, in order to avoid communications problems it is highly recommended to enter this command on the gateway while the *master* tag is turned off. Note also that the system uses the UWB link to send the command, that's why the transmitter and the receiver(s) must be tuned with the same configuration.

```
AT+RCMD=<RECEIVER_UID>,<+CMD=<PARAMETER>>
OK
```

The user is informed by the connected device about the command sent. The fields REQUEST_IDX and RECEIVER_UID respectively identify the current operation and receiver while COMMAND remind the entered parameters.

```
+RCMD_SEND=<REQUEST_IDX>,<RECEIVER_UID>,<COMMAND>
OK
```

The user is also informed by the receiver about the RESPONSE to the command sent. This response is sent 3 times, with random intervals between 10 and 12,000 ms.

```
+RCMD_RECV=<REQUEST_IDX>,<RECEIVER_UID>,<RESPONSE>
OK
```

VI. Unsolicited responses

Description of the messages available at the device output.

1. +DIST

Message with the exchanges synthesis (filtered data) between a tag and one anchor. These distances are the ones used by the multilateration algorithm.

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

Parameters

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

2. +DIST_DBG

Message with the exchanges synthesis (raw data) between the tag and one anchor. If an anchor is not responding, the TMSTP field will be 999999, indicating a time-out.

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

Parameters

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

3. +MPOS

Message with the tag position.

```
+MPOS:<TMSTP>
     ,<X_COORD>,<Y_COORD>,<Z_COORD>
     ,<X_VELOCITY>,<Y_VELOCITY>,<Z_VELOCITY>
```

Parameters

- TMSTP: message emission timestamp in ms
- X_COORD: X-axis tag coordinate in cm
- Y_COORD: Y-axis tag coordinate in cm
- Z_COORD: Z-axis tag coordinate in cm
- X_VELOCITY: X-axis tag velocity in m/s
- Y_VELOCITY: Y-axis tag velocity in m/s
- Z_VELOCITY: Z-axis tag velocity in m/s

4. +POS_INFO

Message with the timestamps differences between the ranging processes done in the current session.

```
+POS_INFO:<NB_ANCHORS>
         ,<ANCHOR_1_UID>,<ANCHOR_1_DT>
         ,...,
         ,<ANCHOR_N_UID>,<ANCHOR_N_DT>
```

Parameters

- NB_ANCHORS: message emission timestamp in ms
- ANCHOR_1_UID: anchor UID on 8 hexadecimal characters
- ANCHOR_1_DT: timestamp difference between the ranging processes
- ANCHOR_N_UID: anchor UID on 8 hexadecimal characters
- ANCHOR_N_DT: timestamp difference between the ranging processes

5. +MESH

Message with the MESHDIST synthesis.

```
+MESH:<TMSTP>,<UID_MASTER>,<NB_DIST>
     ,<UID_DEVICE_1>,<DIST_DEVICE_1>
     ,...
     ,<UID_DEVICE_N>,<DIST_DEVICE_N>
```

Parameters

- TMSTP: message emission timestamp in ms
- UID_MASTER: reference device UID on 8 hexadecimal characters
- NB_DIST: number of measured distances during the MESHDIST process
- UID_DEVICE_1: first device UID on 8 hexadecimal characters
- DIST_DEVICE_1: measured distance between UID_MASTER and UID_DEVICE_1 in cm
- UID_DEVICE_N: last device UID on 8 hexadecimal characters
- DIST_DEVICE_N: measured distance between UID_MASTER and UID_DEVICE_N in cm

6. +VDD

Message with the tag power voltage. This message enables the monitoring of the power supply measured at the output of the voltage converter.

```
+DVDD:<TMSTP>,<CURRENT_VOLTAGE>
```

Parameters

- TMSTP: message emission timestamp in ms
- CURRENT_VOLTAGE: current power supply value in mV

7. +MACC

Message with the tag linear accelerations.

Parameters

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

8. +MGYRO

Message with the tag angular velocities.

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

Parameters

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

9. +MGVT

Message with the tag gravity vector.

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

Parameters

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

10. +MQUAT

Message with the tag orientation (quaternion).

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

Parameters

- TMSTP: timestamp from the IMU in ms
- QUAT_DATA_W: W parameter of the quaternion ($*2^{14}$), without unit
- QUAT_DATA_X: X parameter of the quaternion ($*2^{14}$), without unit
- QUAT_DATA_Y: Y parameter of the quaternion ($*2^{14}$), without unit
- QUAT_DATA_Z: Z parameter of the quaternion ($*2^{14}$), without unit

11. +MHRP

Message with the tag orientation (Euler angles).

```
+MHRP:<TMSTP>,<EUL_YAW>,<EUL_PITCH>,<EUL_ROLL>
```

Parameters

- TMSTP: timestamp from the IMU in ms
- EUL_YAW: yaw of the tag ($*16$) in °
- EUL_PITCH: pitch of the tag ($*16$) in °
- EUL_ROLL: roll of the tag ($*16$) in °

12. +MBARO

Message with the tag pressure and temperature.

```
+MBARO:<TMSTP>,<PRESSURE>,<TEMPERATURE>
```

Parameters

- TMSTP: message emission timestamp in ms
- PRESSURE: pressure measured ($*10000$) in hPa
- TEMPERATURE: temperature measured ($*100$) in $°C$

13. +DPOS

Message with the deported 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>,<LOS_INDICATOR>,<RX_PWRLVL>
```

Parameters

- TMSTP: message emission timestamp in ms
- MOBILE_UID: tag ID on 8 hexadecimal characters
- X_MOBILE,Y_MOBILE,Z_MOBILE: tag coordinates in cm
- ANCHOR_UID: anchor ID on 8 hexadecimal characters
- X_ANCHOR,Y_ANCHOR,Z_ANCHOR: anchor coordinates in cm
- DIST: distance between <mobile_UID> and <anchor_UID> in cm
- LOS_indicator: LOS_indicator/1000 = LOS probability
- RX_PWRLVL: signal power reception in dBm

14. +DIMU

Message with deported tag inertial data. This message is accessible from the devices known as ANCHOR or GATEWAY and appears periodically (100 ms).

```
+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

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

15. +DBARO

Message with deported pressure and temperature.

```
+DBARO:<TMSTP>,<REMOTE_UID>,<PRESSURE>,<TEMPERATURE>
```

Parameters

- TMSTP: message emission timestamp in ms
- REMOTE_UID: remote ID on 8 hexadecimal characters
- PRESSURE: pressure measured ($*10000$) in hPa
- TEMPERATURE: temperature measured ($*100$) in $°C$

16. +DVDD

Message with the deported tag power voltage. This message enables the monitoring of the power supply measured at the output of the voltage converter and appears periodically (1000 ms).

```
+DVDD:<TMSTP>,<MOBILE_UID>,<CURRENT_VOLTAGE>
```

Parameters

- TMSTP: message emission timestamp in ms
- MOBILE_UID: tag ID on 8 hexadecimal characters
- CURRENT_VOLTAGE: current power supply value in mV