MOVEE USER MANUAL

Version 2.01

Technical contact **Julien Lefort / jlefort@nke.fr**

Sales contact : **Pierre Savary / psavary@nke.fr**

Sales Assistant contact : **Ghada Nicot / gnicot@nke.fr**

Table of contents

I. Introduction

1.1. Overview

Movee is a low power LoRa™ Motion sensor for indoor and outdoor industrial applications (asset management, predictive maintenance…), coupled with a temperature sensor for environmental conditions, and a push button for activation or debug frame transmit.

Movee implements several algorithms, which can synthesize the MPU (Motion Processing Unit) data into useful information, lowering the data transmitted on the LoRa™ Network:

- ALIVE = sends an alive frame (battery level + temperature information) periodically
- SHOCK = sends the acceleration information if the configured threshold has been crossed
- MOTION = sends the information (still/in motion) either periodically, or when the activity threshold has been crossed. It can also send the activity (motion/stillness) duration.
- TEMPERATURE = sends the ambient temperature either periodically or if a threshold min or max has been crossed
- TILT = sends the tilt information (pitch/roll) if the orientation threshold has been crossed
- ROTATION = sends the number of turns periodically

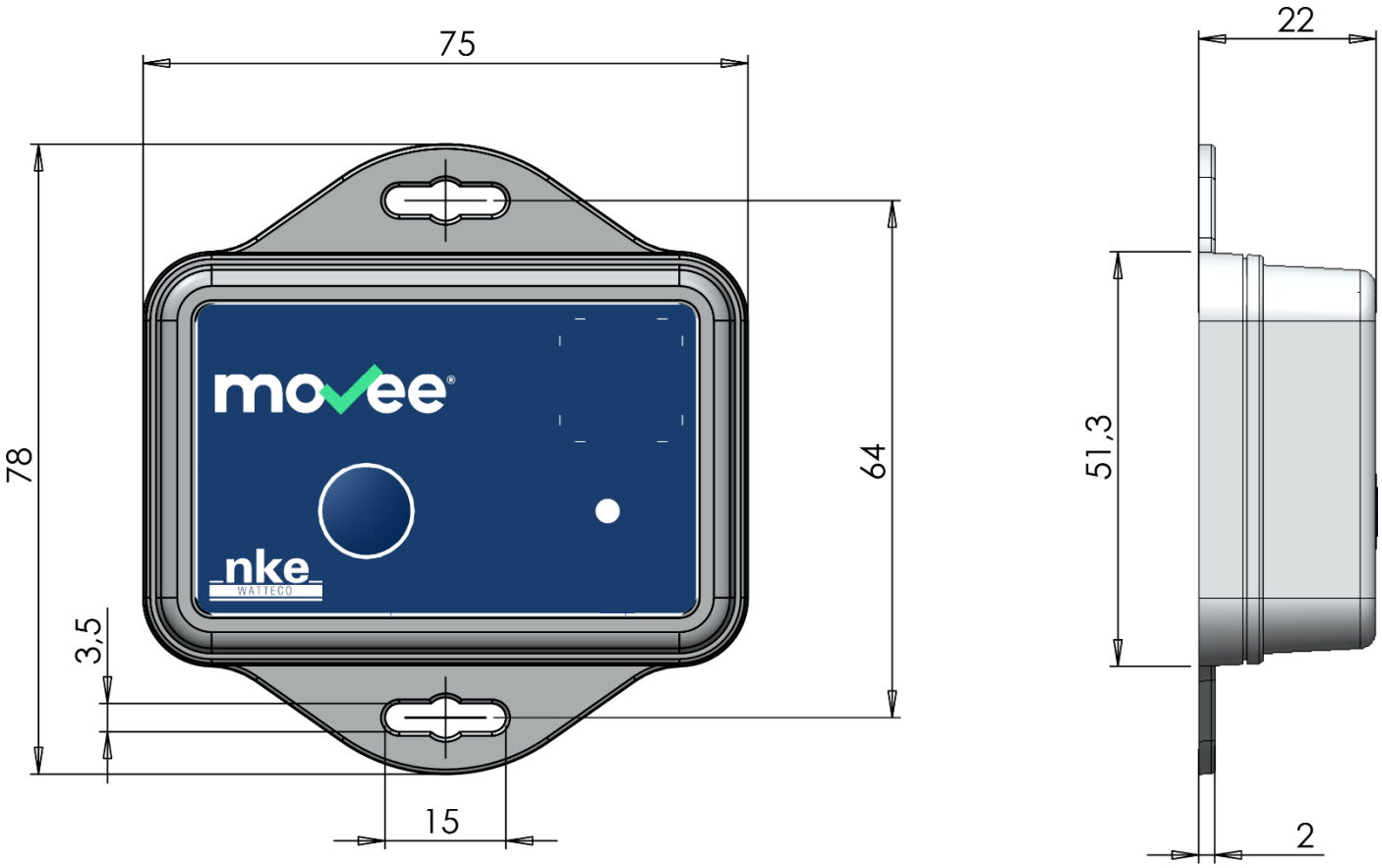
2 more algorithms are implemented, but not recommended as they are not supported anymore :

- ORIENT = sends the orientation information (pitch/roll/yaw) either periodically, or if the orientation threshold has been crossed / Needs always On gyroscope and drains the battery very fast
- VIBE = sends the detected vibration frequency for measures above a defined amplitude threshold – Available on SW version 1.6 and above

The expected battery life is 5 years (for 4 frames per day), and 4 years (for Shock + Motion detection algorithms)

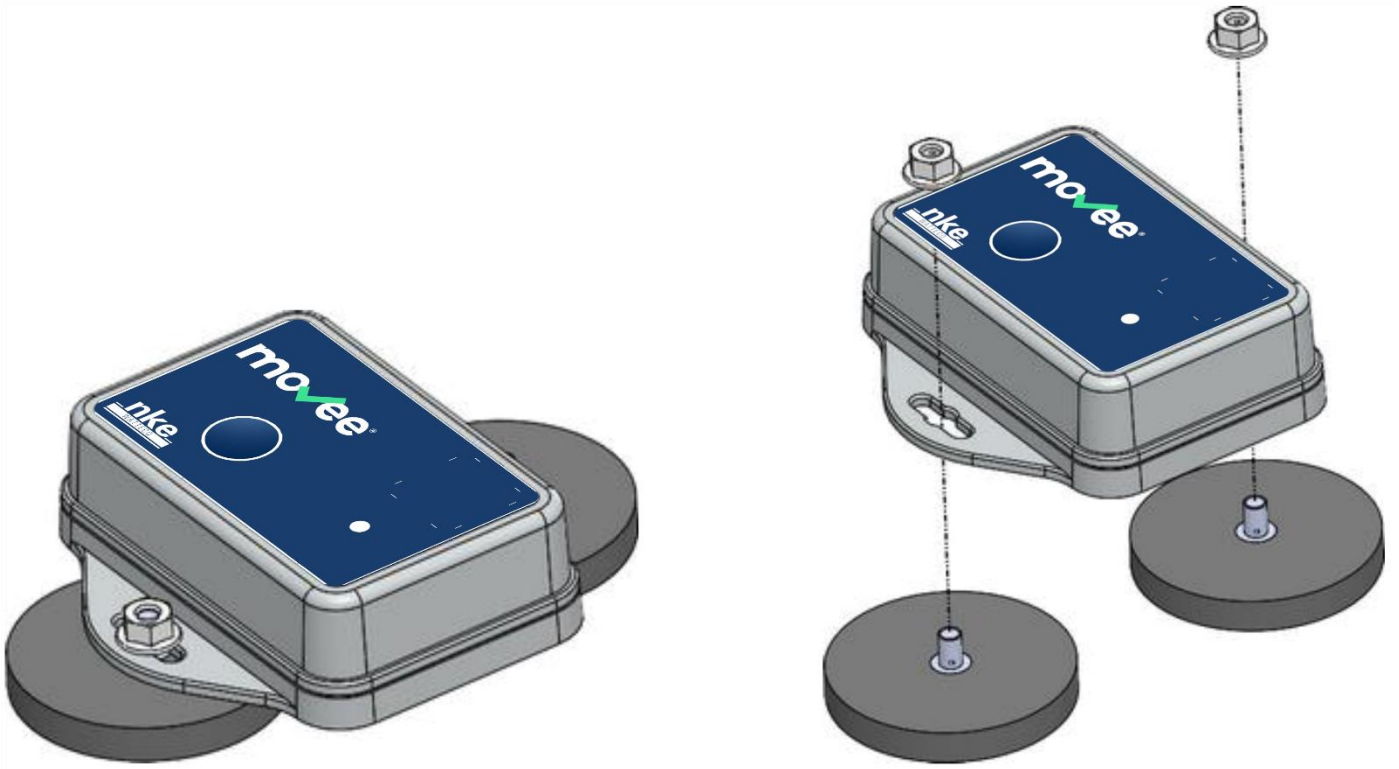
A user interface is available in order to set the parameters for the algorithms (algorithm activation, detection threshold, min, max value…), which can also be set using the LoRaWAN™ DownLink interface.

Movee also comes in an industrial form factor, with a ruggedized 75x78x22mm casing, IP68 compliant, with multiple fastening options.



1.2. Recommended fastening solutions

1.2.1. Magnets

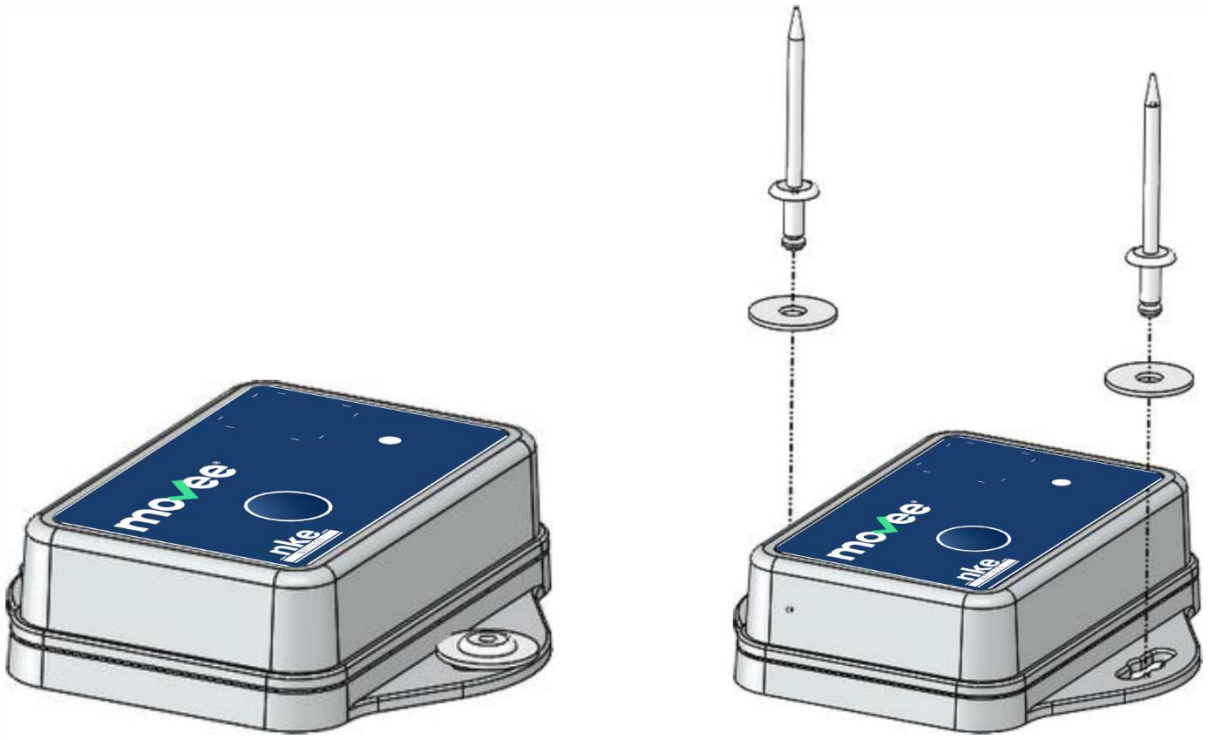


Manufacturer	P/N	Comments
BRUGGER	A43AG-KsM4x6	Magnet - for outdoor installation
BRUGGER	A22AG-KsM4x6	Magnet - for outdoor installation
ECLIPSE	E1053/NEO	Magnet - for indoor installation
ECLIPSE	E1054/NEO	Magnet - for indoor installation
BOSSARD	BN 11207 / 3061765	Nut

These magnets references have been successfully tested as compliant with ETSI 300-019 class 5.2 when mounted on Movee.

Class 5.2: all types of road vehicles used in areas with a well-developed road system, except tracked vehicles, motorcycles, scooters and other vehicles with low mass. The equipment can be mounted on surfaces which may be subjected to flying stones. The equipment may be mounted on passenger car instrument panels to which high frequency vibrations from the engine, or from other parts connected to the engine, may be transmitted. This class also applies to fork lift trucks and trains with soft suspension and shock reducing buffers

1.2.2. Rivet

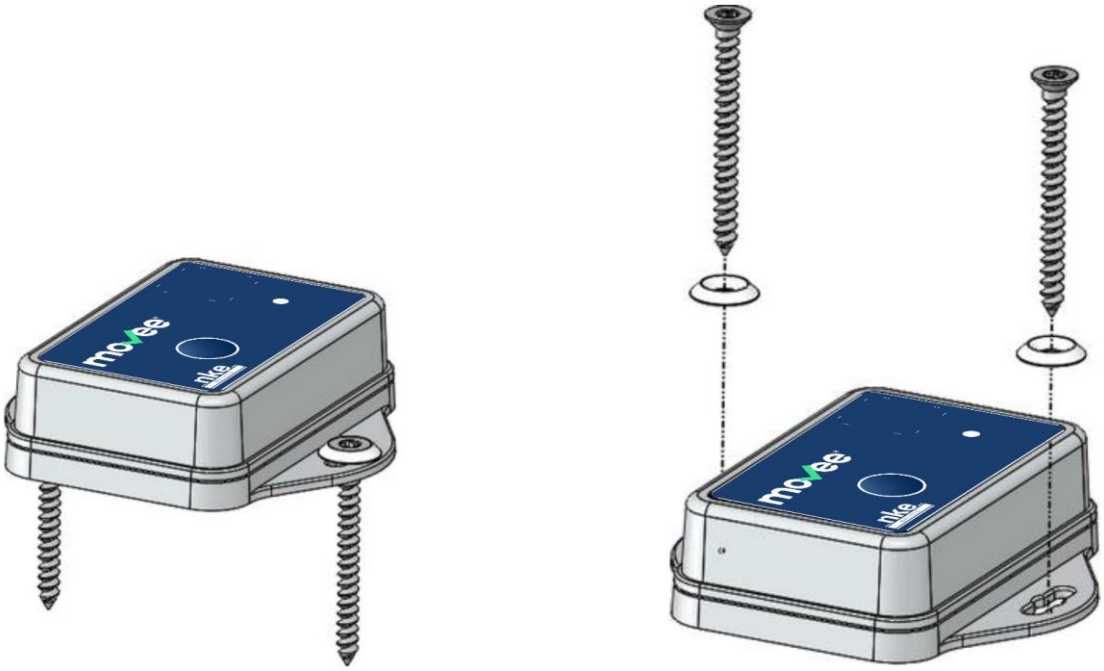


Rivet properties: Ø4mm, length 7/10.2/13.3mm

Drilling: Ø4.1mm / Ø4.2mm

Manufacturer	P/N	Comments
BOSSARD	BN 84545 / 8031215	Ring washer
BOSSARD	BN 1409 / 3206579	Rivet

1.2.3. Screw



Manufacturer	P/N	Comments
BENE INOX	211309	TF TORX 4x45mm wood screw
WURTH	0455000304	Cup washer

1.3. Key Features

Product Application

Smart Logistics

Smart Industry

Main Features

- LoRa motion sensor (accelerometer & gyroscope) with ambient temperature sensor and push button
- LoRaWAN 1.0 compatible with public LoRa networks
- 5 years battery life (4 frames per day)
- 4 years battery life (for Shock + Motion detection algorithms)
- IP68 case (75x22x50mm)
- Operating temperature: -30°C to +60°C
- Fastening: Screws, Rivet, hose clip, Glue, Adhesive…
- PC UI for programming and calibration

TECHNICAL SPECIFICATIONS

Performances

- Drop detection (+/-1g)
- 3-axis shock detection (+/-1mg up to +/-16g)
- 3-axis tilt and orientation (+/-1°)
- Temperature sensor (+/-1°C)

Hardware

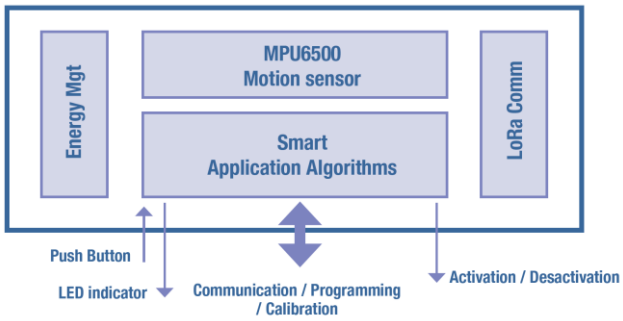
- 6-axis accelerometer and gyroscope
- Temperature sensor
- Silicon Labs Cortex-M3 microcontroller EFM32
- Flash: 256KB
- RAM: 32KB
- Semtech SX1272 LoRa™ transceiver
- 2000mAh battery (non rechargeable)
- 1x RGB LED
- 1x Push button

Protocol - Network

- LoRa™ SF6-SF12
- LoRa™ 1.0 Class A, Class C (compatible)
- LoRa™ sensitivity: -137 dBm

Casing

- CE marking
- Polyamide 6.6 (PA 66) material
- Indoor & outdoor use (IP68/60min.@1m)
- 2m drop resistant
- Dimensions: 75mmx50mmx22mm
- Fastening: Screws, Rivet, hose clip, Glue, Adhesive...
- Lexan front panel (customizable)
- Storage temperature: -25°C to +55°C



ADDITIONAL SERVICES

Software - Firmware

- Custom algorithm

Antenna design

- PCB antenna

Casing

- Custom Lexan
- Magnetic fastening

ORDERING INFORMATION

Designation	Part Number
Movee Sensor LORAWAN EU	50-80-002

II. Functional Overview

2.1. Manufacturing configuration

2.1.1. Device default configuration

When delivered, the device will be OFF. To switch the device ON, please see *§2.2.1Turn the device ON*.

The default LoRaTM configuration is OTAA (Over The Air Activation, with DevEUI, and AppEUI contained in the QR code of the product sticker, see *§2.1.2 QR code labels*). AppKey is transmitted along with DevEUI and AppEUI through Excel sheet via e-mail at the moment.

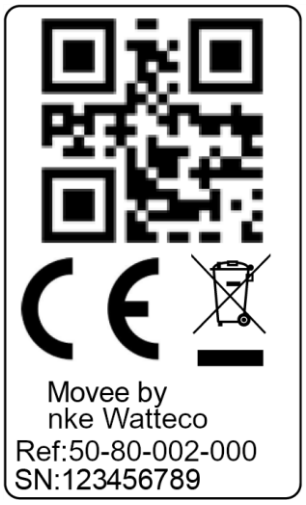
And the default algorithm configuration is:

- ALIVE: 1 alive frame sent every 2 hours (7200s)

2.1.2. QR code labels

2.1.2.1. Rear label

The label under the device contains the coordinate system used by the MPU, regulatory information and device configuration details coded in the QR code:



QR code content (77 digits for P/N < 3.6 ; 45 for P/N ≥ 3.6):

- 16 digits: AppEUI (e.g. 70B3D531C0001190)
- 16 digits: DevEUI (e.g. 70B3D531C0001242)
- 4 digits: manufacturing Year on 2 digits (e.g. 17 for 2017) and Week on 2 digits (e.g. 03 for W03)
- 9 digits: Product serial Number (e.g. 123456789)

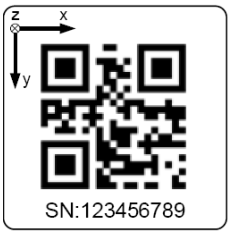
QR Code content example:

70B3D531C000119070B3D531C00012421703123456789

2.1.2.2. Front label

The label on the Lexan eases the identification of the device once installed. It contains the coordinate system used by the accelerometer (MPU), and device identification details in the QR code:

QR Code content (32 digits):



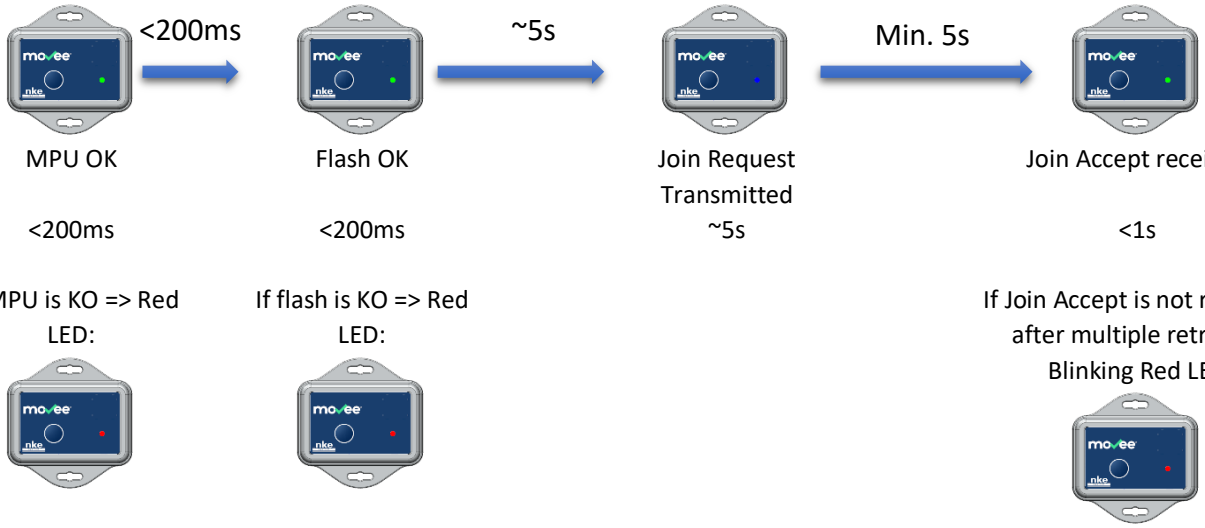
- 16 digits: AppEUI (e.g. 70B3D531C0001190)
- 16 digits: DevEUI (e.g. 70B3D531C0001242)

The serial Number is also written in plain text on the bottom of the label.

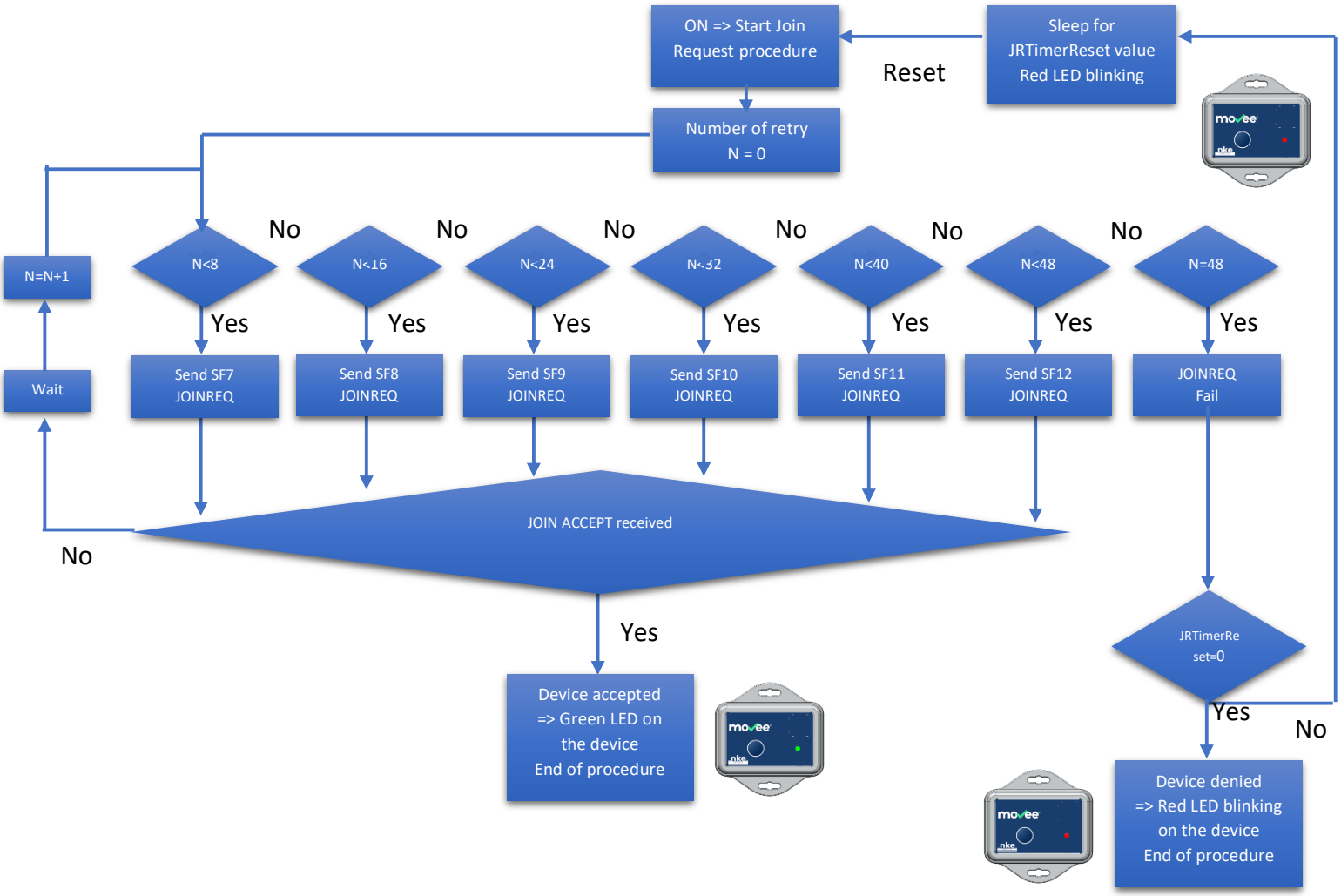
2.2. Device interaction

2.2.1. Turn the device ON

2.2.1.1. OTAA configuration (default)



The Join Accept reception might take some time, depending on the Network signal strength. The following figure describes the back off procedure when no Join Accept has been received:



2.2.1.2. OTAA first frame (only for devices with a P/N ≥ 3.6)

When the device has received the join accept from the Network Server, it will send a first "VERSION" frame. The VERSION frame is coded in ASCII, each value is separated by a ";", coded as 0x3B in hexadecimal (see VERSION payload). This frame contains:

Name	Battery level	Temperature	Product version	SW version	LoRa Stack version	Reset cause
Size	1 byte	1 byte	N bytes	N bytes	N bytes	N bytes
Coding	8 bits (unsigned)	8 bits (signed)	ASCII	ASCII	ASCII	ASCII
Coded value:	2,8V…3,6V	-127°C…+128°C	Text	Text	Text	Text

2.2.1.3. Reset cause codes

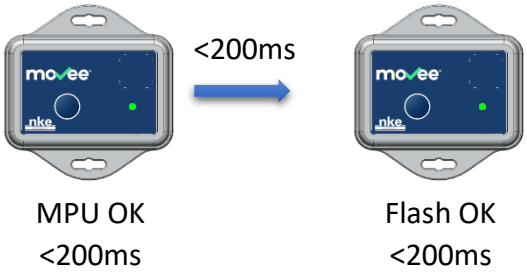
The following list gives the codes and the explanation of why the product has reset/restarted:

- POR = Power On Reset = Cold boot (product has been powered off with the button, then restarted) or power drop off (battery level has felt under the minimal value)
- EXTRST = External Reset = Forcing the RESETn pin low generates a reset of the Microcontroller = can only be triggered by a JTAG probe.
- BOD = Brown-Out Detection on Analog Power Domain (AVDD0/1) = The main supply voltage for the Microcontroller has dropped below the minimal value, causing a system reset
- BODREG = Brown-Out Detection on un/regulated Domain = The backup supply voltage for the Microcontroller has dropped below the minimal value, causing a system reset
- WDOGRST or EXTWDOG = Watchdog reset (not used on Movee)
- SYSRST = System Reset (requested by software, JTAG…)
- LOCKRST or LOCKSYS = Lockup Reset (result of the core being locked up because of an unrecoverable exception)
- UKN(0xTT) = unknown reset cause, TT is the error code, for more details, see *§4.3 EFM32 Reset cause table*

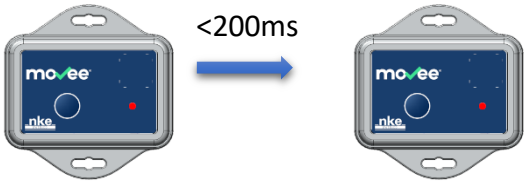
2.2.1.4. ABP configuration

ABP mode is not recommended, but supported by Movee.

When in ABP mode there is no join procedure, the device starts sending messages directly to the gateway. ABP mode is not supported by the Movee Configurator application, and is not recommended (please contact us if you need ABP mode).



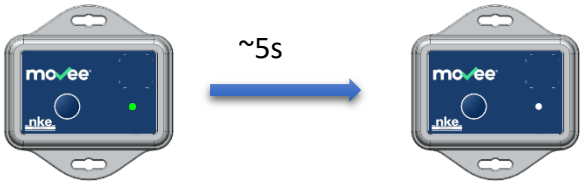
If MPU is KO => Red LED: If flash is KO => Red LED:



2.2.2. Send an ALIVE frame

To send an ALIVE frame (when the device is ON, and has joined a network), press the push button for more than 1s (and less than 7 seconds), the LED will stay green for 2s, and the LED will blink green when the frame is sent.

See *§3.4.4.1 ALIVE payload* for Alive frame content description.



Push button pressed

LoRa ALIVE frame sent

If duty cycle is not respected => Red Led blinking twice, frame ignored

2.2.3. LoRa™ Radio

2.2.3.1. LoRa™ channels configuration

2.2.3.1.1. Tx channels

Movee supports the 3 mandatory channels for LoRaWAN™ communication:

Channel name	Frequency	Data Rate
LC1	868.1MHz	DR0 to DR5
LC2	868.3MHz	DR0 to DR5
LC3	868.5MHz	DR0 to DR5

It also supports 7 optional channels:

Channel name	Frequency	Data Rate
LC4	867.1MHz	DR0 to DR5
LC5	867.3MHz	DR0 to DR5
LC6	867.5MHz	DR0 to DR5
LC7	867.7MHz	DR0 to DR5
LC8	867.9MHz	DR0 to DR5
LC9	868.8MHz	DR7 (FSK)
LC10	868.3MHz	DR6

Reminder for Data Rate configuration (based on LoRaWANTM 1.0.2 rev.B specification):

DataRate	Configuration	Indicative physical bit rate [bit/s]
0	LoRa: SF12 / 125 kHz	250
1	LoRa: SF11 / 125 kHz	440
2	LoRa: SF10 / 125 kHz	980
3	LoRa: SF9 / 125 kHz	1760
4	LoRa: SF8 / 125 kHz	3125
5	LoRa: SF7 / 125 kHz	5470
6	LoRa: SF7 / 250 kHz	11000
7	FSK: 50 kbps	50000
8..15	RFU	

2.2.3.1.2. Rx channels

For the RX1 receive window, the channel list is the same as for TX channel.

For the RX2 receive window, the channel configuration is fixed:

Channel name	Frequency	Data Rate
RX2	869.525MHz	DR0

2.2.3.2. EU frequency band Duty cycle limitation

The LoRaWAN™ protocol is based on the 868MHz frequency band for EU region (Europe), which requires that the devices using channels in this band can't use more than 1% of the bandwidth (max 36s on 1hour) on the 3 mandatory channels (LC1/LC2/LC3) plus optional channel LC10 and 0.1% of the bandwidth (max 3.6s on 1hour) on the optional channels (LC4 to LC9).

The embedded LoRaWAN™ stack will calculate the duty cycle conformity, for example on a 10 minutes time span, the "time on air" for the device shall not be more than 36s (calculated on one or more transmission). Providing that the allowed bandwidth has already been used (e.g. by the OTAA Join Request/Join Accept procedure just after startup), the device will not be able to send LoRaWAN™ frames as long as the duty cycle calculation at the LoRaWAN™ stack level will not release the transmit permission.

Once the transmit permission is granted by the LoRaWAN™ stack, transmissions will resume.

Information can then be lost due to the duty cycle limitation. For example, a shock is detected, a shock frame shall be sent, but the duty cycle has not been respected. As long as the transmission permission is not granted, other events (alive frame requested by the push button, tilt detection…) will not be recorded, nor sent.

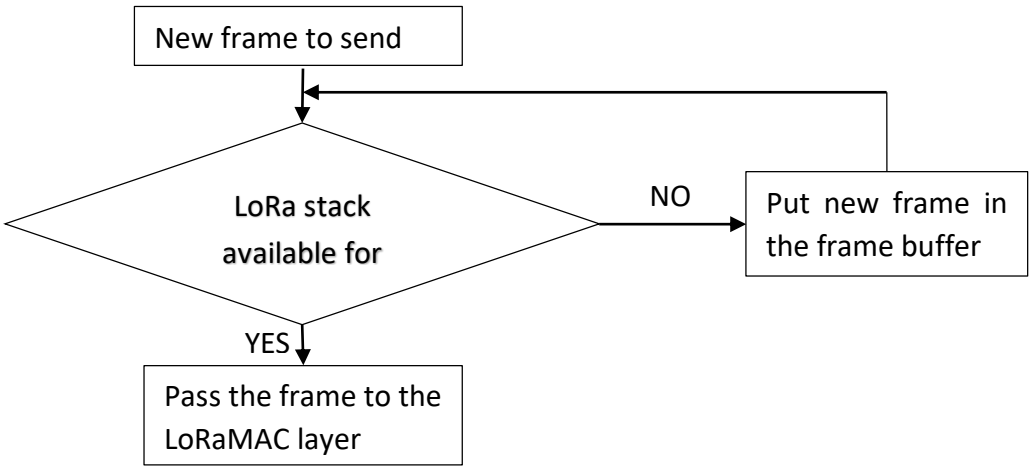
When a frame is dismissed because of the duty cycle limitation, a red LED blinks twice on the device:

For products with P/N≥3.9, an individual buffer for each algorithm has been added. So that if events occur while the duty cycle limitation has not been released by the LoRaWANTM stack, the new information for each algorithm will replace the information which has been buffered.

2.2.4. LoRa Frame buffer

In order to comply with the duty cycle regulation and not lose important frame, **a frame buffer has been implemented from P/N 3.9 and above**.

The LoRaMAC layer does not provide a buffer and can handle just one frame at a time. The frame buffer is used when a frame could not be sent (it happens when the LoRaMAC layer is busy = a frame is pending, or the previous sent frame calculated duty cycle is not over yet).



The frame buffer has 3 levels of priority depending on the type of event received by the running algorithms. Default configuration is set as follows:

- High priority

TILT
MOTION
ROTATION
ORIENT
VIBE

- Medium priority

TEMP
SHOCK

- Low priority

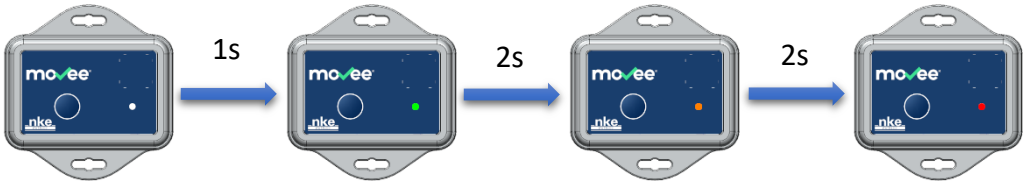
ALIVE

When the stack status returns as available, Movee will send the frame(s) tagged as high priority first, then frame(s) pending tagged as Medium priority (if any), and finally frame(s) tagged as Low priority.

When a new frame is to be sent while there already is a frame of the same algorithm in the frame buffer, the new frame will replace the frame in the frame buffer, which will be dumped.

2.2.1 Turn the device OFF

To turn the device OFF, press the push button for more than 5 seconds. The RGB LED will go from Green to Yellow to Red. Once the RGB LED is Red, release the push button, the device is OFF.



For product P/N≥3.9, it is possible to disable the power off action of the button, see the parameters list for "ALGO" in *§3.2.2 ALGO*

2.3. Tools

The device can be configured with a Windows compliant application called "MoveeConfigurator".

You can download the tool on our support platform:

http://support.nke-watteco.com/movee/

See *§ 3.6 Movee Configurator user interface* for detailed user interface manual

III. User Guide

3.1. Operating modes

3.1.1. NORMAL mode

Normal mode is the default mode entered by the device upon power-on / reset

3.1.2. SERVICE mode

Service mode is entered upon a downlink command reception to enter service mode.

In this mode, all the running algorithms are stopped, and a service frame is transmitted every minute (with eventual duty cycle limitation), in order to speed up the parameters reception through DownLink (DL) frames. The LED alerts that the product is in service mode by blinking yellow (1Hz).

It is advised to enter this mode to update the parameters of the algorithms, and to leave the service mode once the parameters have been updated.

3.2. Movee algorithms and parameters

3.2.1. MPU - Motion Processing Unit

3.2.1.1. MPU parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
MPU	maxRange (= MR)	Dynamic range – **Default value is ±8000 for P/N<3.9 and ±16000 for P/N≥3.9**		**±16000**		N/A	mG
				±8000			
				±4000			
				±2000			
	orientation	Selects the orientation of the installed product – **Parameter available for P/N≥3.9** 0: the device is laid flat (horizontal position)  1: the device is set upright (vertical position) 	0	0	1	N/A	Boolean

"Orientation" parameter is used by the SHOCK/TILT/MOTION/ORIENTATION/VIBE algorithms (rotation of the coordinate system), it will be ignored by the ALIVE/TEMPERATURE/ROTATION algorithms

3.2.2. ALGO

3.2.2.1. ALGO Parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
ALGO	alertPeriod	Sets the period between each alert when an algorithm is set to send multiple alerts (e.g. TILT with parameter alertMultiple=1) - **Parameter available for P/N≥3.9**	0	1	1	N/A	Boolean
	poweroff	Sets the button effect on the possibility to power off the device after a long press (5s) 0 = Poweroff inhibited 1 = Poweroff available - **Parameter available for P/N≥3.9**	0	1	1	N/A	Boolean

3.2.3. LORA

3.2.3.1. LoRa™ parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
LORA	JRERROR	Activates a timer allowing to restart a Join Request cycle after JRTimerReset value (in hours) when a device has completed a full Join Request Cycle without getting a Join Accept response If JRTimerReset is set to 0, this feature is deactivated **Parameter available for P/N≥3.9**	0	12	240	h	s
	DEFAULT_DR	Sets the preferred data rate for LoRa communication (otherwise set by the network and not dynamic) **Parameter available for P/N≥3.9**	DR0	DR0	DR5	N/A	N/A

3.2.4. ALIVE algorithm

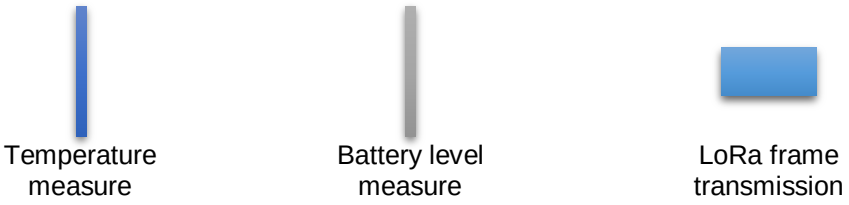
3.2.4.1. ALIVE parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
ALIVE	modeRefresh	Activates Periodical frame transmission	0	1	1	N/A	Boolean
	period	Period between 2 frames	60	3600	36E+05	ms	s
	nbSavedValue	Number of period between 2 LoRa frames, with temperature measure for each period	1	1	48	N/A	N/A

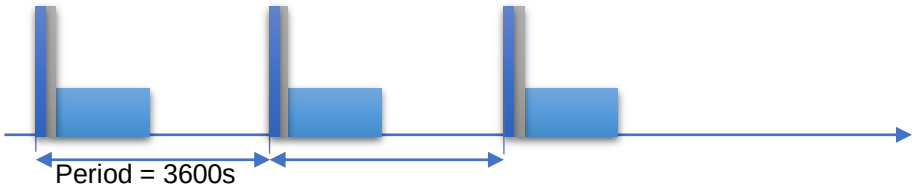
See *§3.4LoRa frames payload description* for payload decoding

For SW version < 1.8.8, period value is limited to 32400 s

3.2.4.2. ALIVE implementation



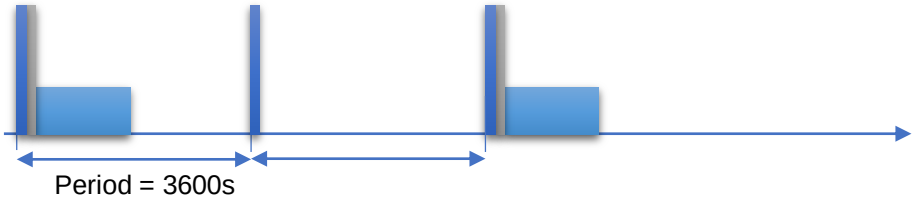
Example 1: modeRefresh = 1 / period = 3600 / nbSavedValue = 1



LoRa frame sent: **e61801aa**

- Payload header: **e618**
  - **0xE6**: battery level : 0xE6 = 230: $\frac{(3,6-2,8)}{255} * 230 + 2,8 = 3,52\ Volts$
  - **0x18**: Temperature = 24°C
- Payload data : **01**
  - **0x01**: Data type = ALIVE => no more payload data
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: modeRefresh = 1 / period = 3600 / nbSavedValue = 2



LoRa frame sent: **e6180218aa**

- Payload header: **e618**
  - **0xE6**: battery level : 0xE6 = 230: $\frac{(3,6-2,8)}{255} * 230 + 2,8 = 3,52\ Volts$
  - **0x18**: last temperature = 24°C
- Payload data : **0218**
  - **0x02**: Data type = TEMPERATURE => expected payload data = N bytes (signed)
    - **0x18**: Recorded temperature = 24°C
- Payload end of frame: **aa**

0xAA: End of frame

3.2.5. SHOCK algorithm

3.2.5.1. SHOCK parameters

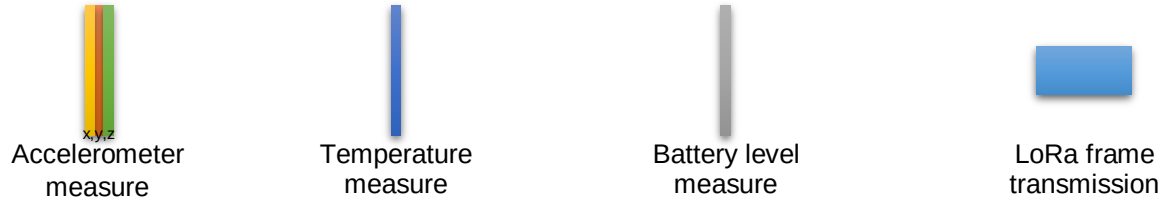
	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
SHOCK	gxSup	Positive threshold on X axis	25	5000	MR[1]	mG	G
	gxInf	Negative threshold on X axis	25	5000	MR		
	gySup	Positive threshold on Y axis	25	5000	MR		
	gyInf	Negative threshold on Y axis	25	5000	MR		
	gzSup	Positive threshold on Z axis	25	5000	MR		
	gzInf	Negative threshold on Z axis	25	5000	MR		
	freq	Sensor sampling rate Possible values = 0.24, 0.49, 0.98, 1.95, 3.91, 7.81, 15.63, 31.25, 62.50, 125, 250 and 500Hz	0.24	15.63	500	Hz	Hz
	inhibition	Inhibition time before a shock can be detected ***If set to 0, then inhibition is deactivated***.	50	150	65535	ms	s
	removeGravity	Activates relative acceleration measure mode (Gravity removed)	0	0	1	N/A	Boolean

See *§3.4 LoRa frames payload description* for payload decoding

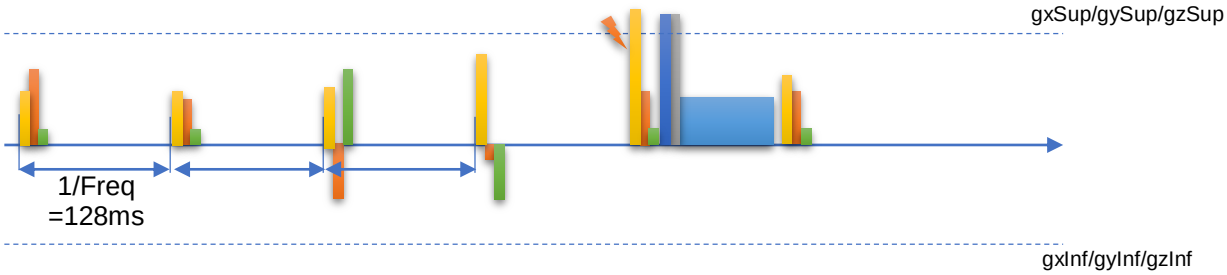
removeGravity parameter activates an algorithm, which removes the gravity 's vector projection from the gx, gy, gz values recorded by the MPU when an acceleration is detected. This algorithm returns less reliable values than normal operation mode (***removeGravity*** not activated).

[1] MR = Max Range, see ***§Erreur ! Source du renvoi introuvable. Erreur ! Source du renvoi introuvable.***

3.2.5.2. SHOCK implementation



Example 1: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; inhibition = 0, removeGravity = 0

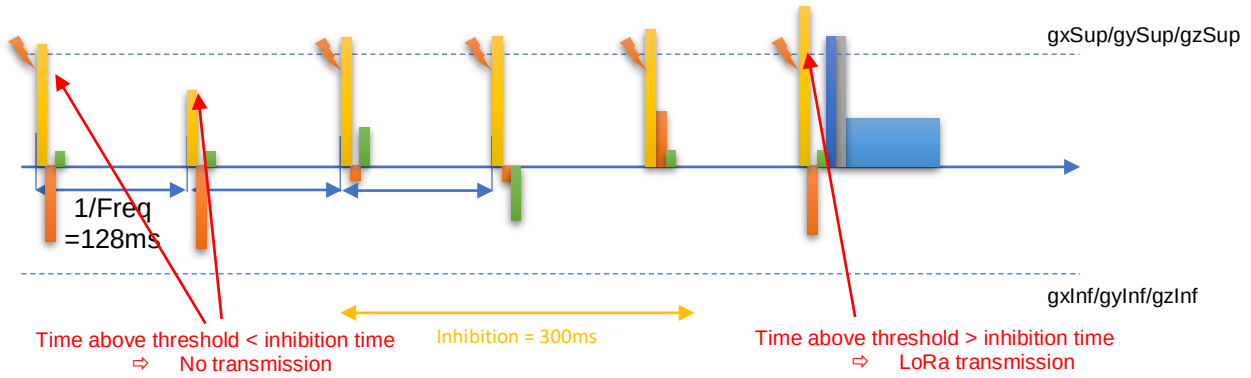


LoRa frame sent: **c11a040ad104540134aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **040ad104540134**
  - **0x04**: Data type = SHOCK => expected payload data = Gx (2 bytes signed), Gy (2 bytes signed), Gz (2 bytes signed)
    - **0x0AD104540134**
      - **0x0AD1**: Gx = 2769mG
      - **0x0454**: Gy = 1108mG
      - **0x0134**: Gz = 308mG
- Payload end of frame: **aa**

0xAA: End of frame

Example 2: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; inhibition = 300, removeGravity = 0



LoRa frame sent: **c11a040c9fff970134aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **040c9fff970134**
  - **0x04**: Data type = SHOCK => expected payload data = Gx (2 bytes signed), Gy (2 bytes signed), Gz (2 bytes signed)
    - **0x0C9FFF970134**
      - **0x0C9F**: Gx = 3231mG
      - **0xFF97**: Gy = - 1385mG
      - **0x0134**: Gz = 308mG
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.2.6. MOTION algorithm

3.2.6.1. MOTION parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
MOTION	gxSup	Positive threshold on X axis	25	300	MR[2]	mG	G
	gxInf	Negative threshold on X axis	25	300	MR		
	gySup	Positive threshold on Y axis	25	300	MR		
	gyInf	Negative threshold on Y axis	25	300	MR		
	gzSup	Positive threshold on Z axis	25	300	MR		
	gzInf	Negative threshold on Z axis	25	300	MR		
	freq	Sensor sampling rate Possible values = 0.24, 0.49, 0.98, 1.95, 3.91, 7.81, 15.63, 31.25, 62.50, 125, 250 and 500Hz	0.24	15.63	500	Hz	Hz
	timerA	Lapse of time with motion detection before a motion is detected/reported	500	2000	65535	ms	s
	timerB	Lapse of time without motion detection before stillness is detected/reported	500	3000	65535		
	sensitivity	Number of samples greater than the defined threshold, during timerA value, involving activity detection	0	10	$\frac{freq.timerA}{1000}$	N/A	Integer
	activity	Activates motion duration calculation	0	1	1	N/A	Boolean
	additionateActivity	If activated, the motion duration adds up, and is not reset at each LoRa transmission	0	1	1		
	periodicActivity	Activates a periodic summary of the activity measured on the period defined by "activityResumePeriod"	0	1	1		
	activityResumePeriod	Period of the activity summary transmission. Defined in hours.	1	1	360	h	s
	loraAtStartMvt	The LoRa frame is sent when motion is detected/reported	0	0	1	N/A	Boolean
	loraAtStopMvt	The LoRa frame is sent when stillness is detected/reported	0	0	1		

See *§3.4 LoRa frames payload description* for payload decoding

If ***activity*** parameter is set to 1, ***loraAtStartMvt*** parameter will be ignored.

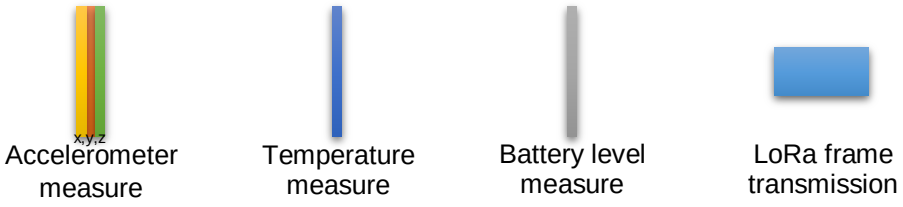
If **activity** parameter is set to 0, ***additionateActivity***, ***periodicActivity*** and ***activityResumePeriod*** will be ignored.

If ***periodicActivity*** is set to 1, ***loraAtStartMvt*** and ***loraAtStopMvt*** parameters will be ignored, and ***activityResumePeriod*** parameter will be used.

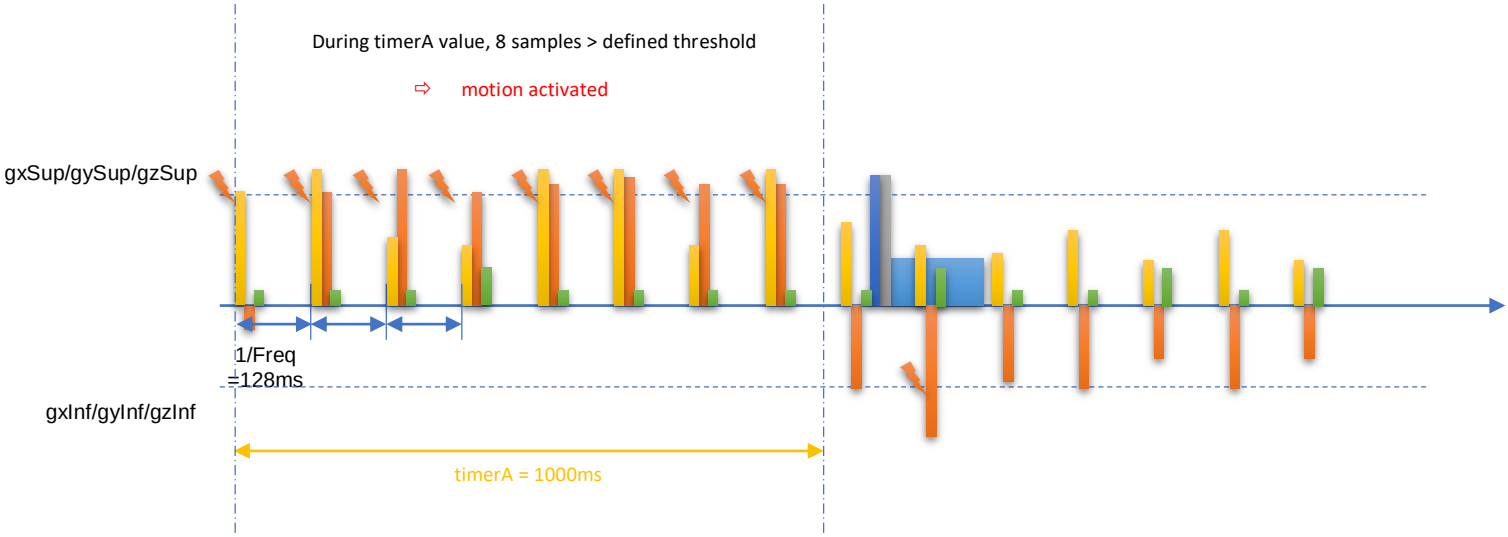
activityResumePeriod parameter will only be used when ***activity*** and ***periodicActivity*** are set to 1.

[2] MR = Max Range, see ***§Erreur ! Source du renvoi introuvable. Erreur ! Source du renvoi introuvable.***

3.2.6.2. MOTION implementation



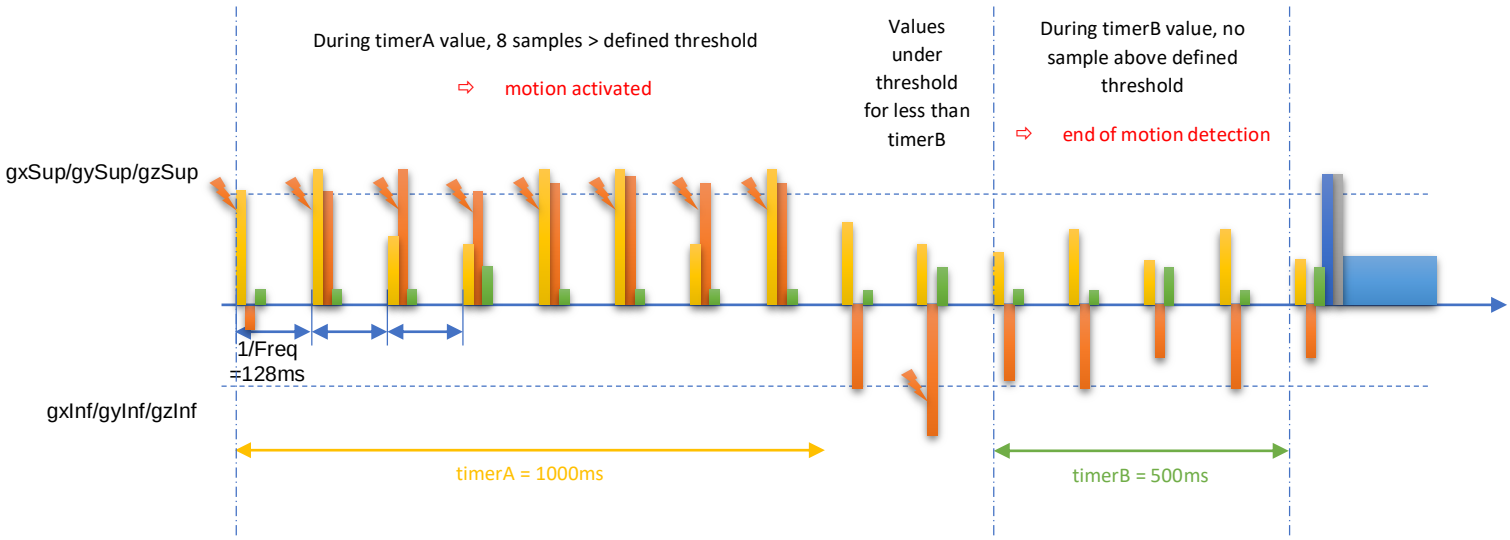
Example 1: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 1000; *timerB = 500*; sensitivity = 6; **activity = 0**; *additionateActivity = 0; periodicActivity = 0; activityResumePeriod = 1*; loraAtStartMvt = 1; loraAtStopMvt = 0 (*parameter ignored*)



LoRa frame sent: **c11a2000aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **2000**
  - **0x20**: Data type = MOTION => expected payload data = Motion/Stillness (1 byte)
    - **0x00**: Motion
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 1000; timerB = 500; sensitivity = 6; **activity = 0**; *additionateActivity = 0; periodicActivity = 0; activityResumePeriod = 1*; loraAtStartMvt = 0; loraAtStopMvt = 1 (*parameter ignored*)

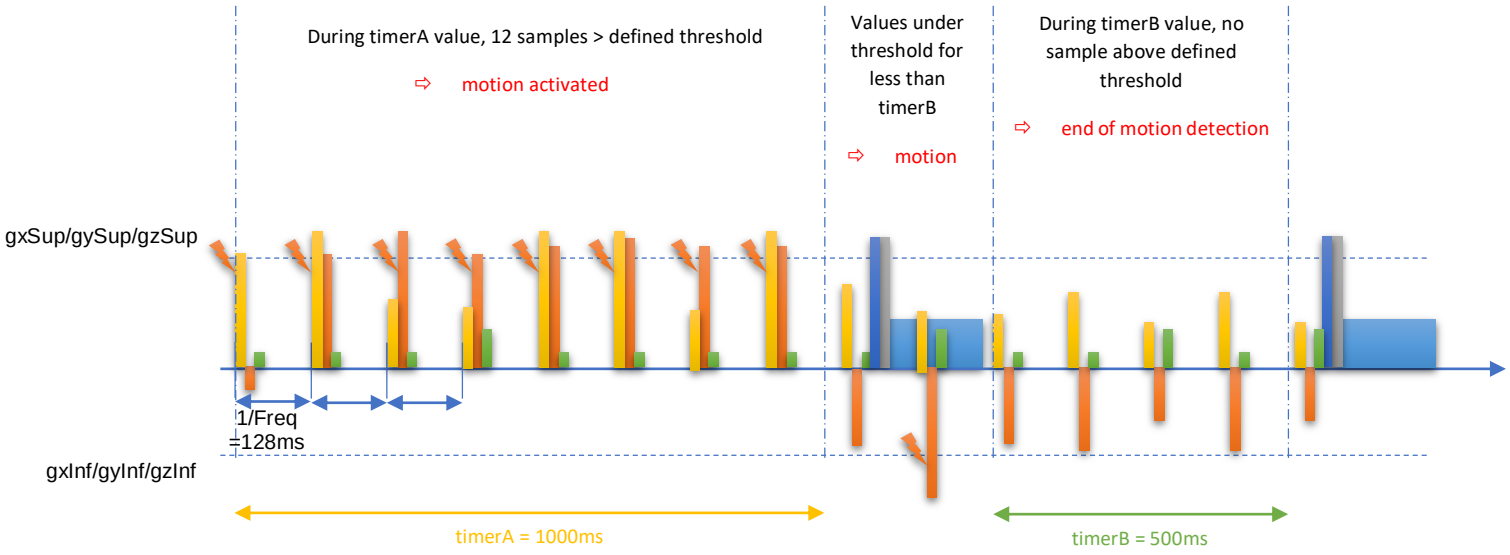


LoRa frame sent: **c11a2001aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **2001**
  - **0x20**: Data type = MOTION => expected payload data = Motion/Stillness (1 byte)
    - **0x01**: Stillness
- Payload end of frame: **aa**

0xAA: End of frame

Example 3: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 1000; timerB = 500; sensitivity = 6; **activity = 0**; *additionateActivity = 0; periodicActivity = 0; activityResumePeriod = 1*; loraAtStartMvt = 1; loraAtStopMvt = 1 (*parameter ignored*)



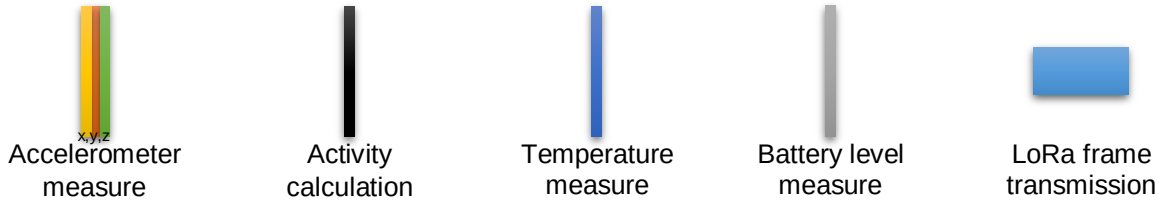
LoRa frame sent at start: **c11a2000aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **2000**
  - **0x20**: Data type = MOTION => expected payload data = Motion/Stillness (1 byte)
    - **0x00**: Motion
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame sent at stop (example given, real life duty cycle restrictions will not allow this transmission): **c11a2001aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **2001**
  - **0x20**: Data type = MOTION => expected payload data = Motion/Stillness (1 byte)
    - **0x01**: Stillness
- Payload end of frame: **aa**

0xAA: End of frame



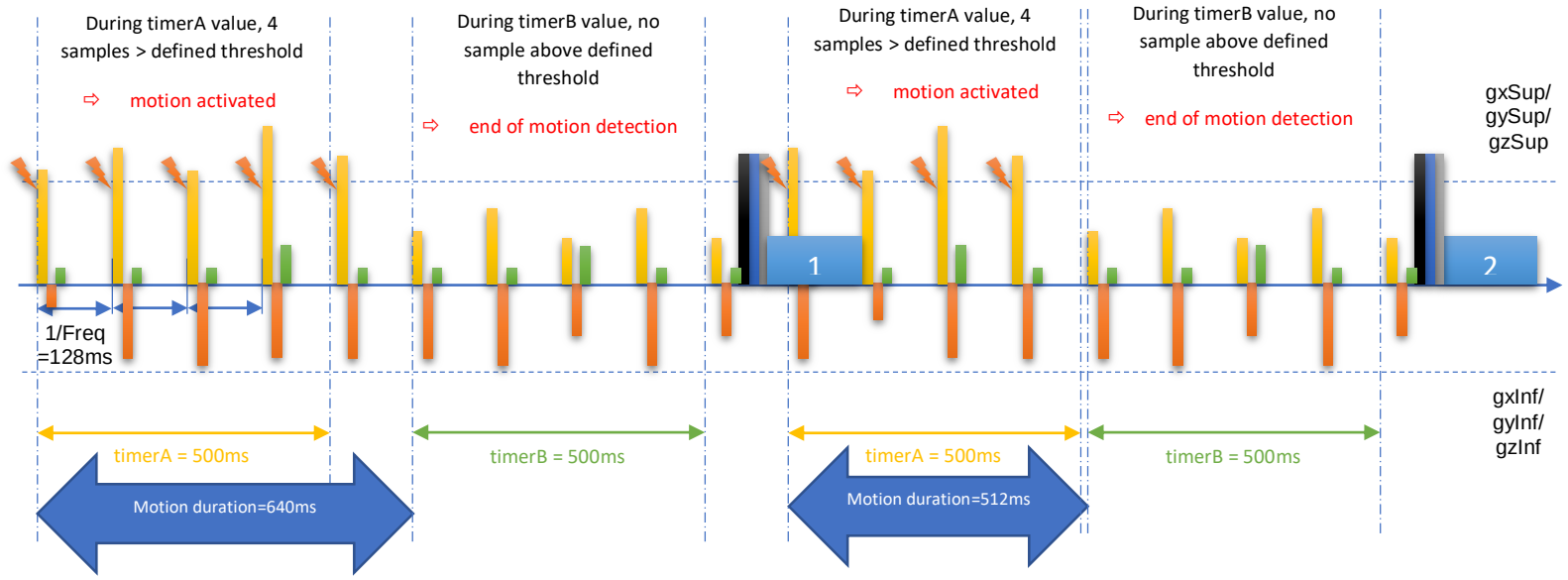
Example 4: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 500; timerB = 500; sensitivity = 3; **activity = 1**; **additionateActivity = 0**; *periodicActivity = 0; activityResumePeriod = 1; loraAtStartMvt = 1; loraAtStopMvt = 1* (*parameter ignored*)

LoRa frame sent: **c11a400000000300aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000000300**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00000300**: duration = 768ms
- Payload end of frame: **aa**

0xAA: End of frame

Example 5: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 500; timerB = 500; sensitivity = 3; **activity = 1**; **additionateActivity = 1**; periodicActivity = 0; *activityResumePeriod = 1; loraAtStartMvt = 1; loraAtStopMvt = 1* (*parameter ignored*)



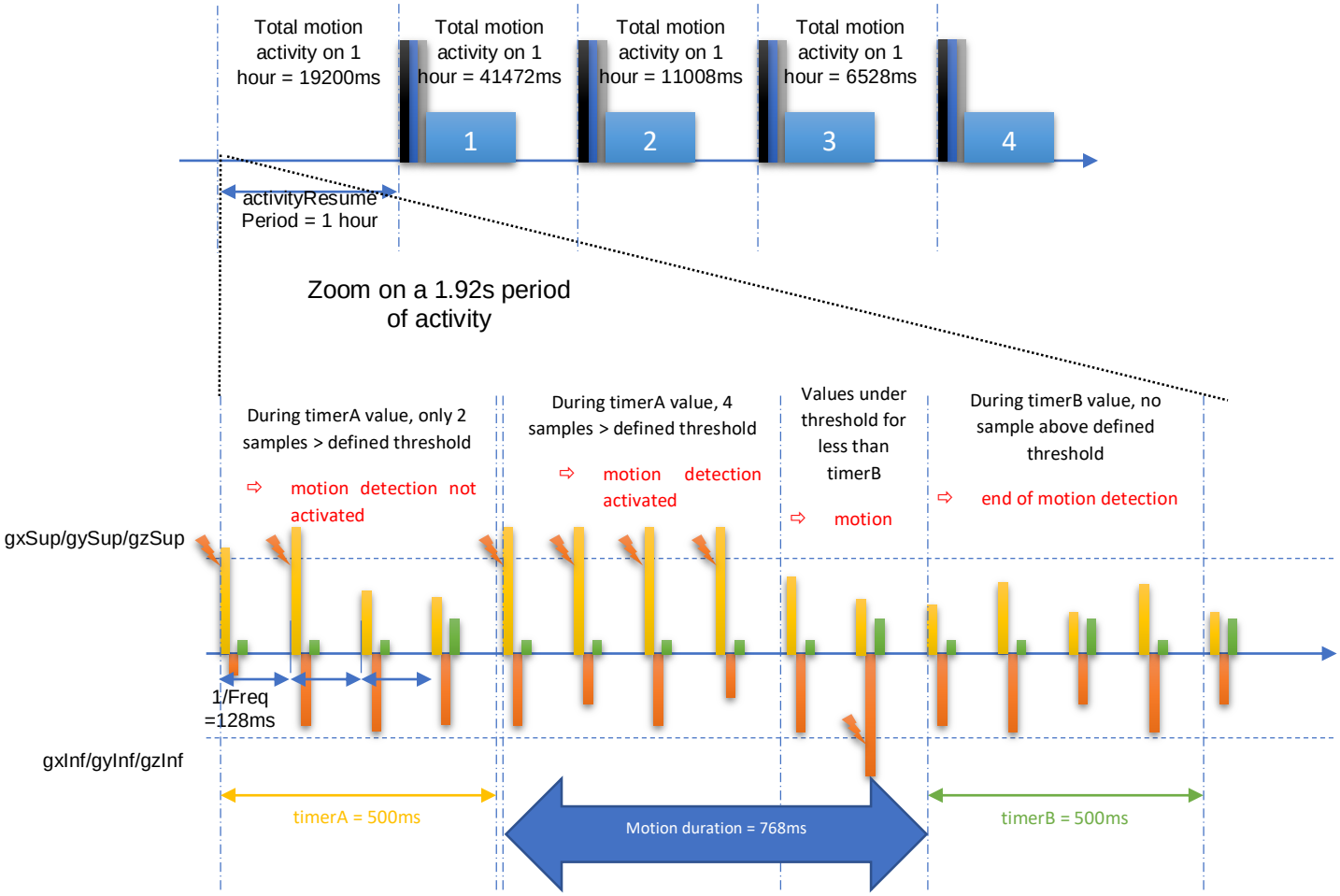
LoRa frame "1" sent: **c11a400000000280aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000000280**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00000280**: duration = 640ms
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "2" sent: **c11a400000000480aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000000480**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00000480**: duration = 1152ms (=640+512ms)
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 6: gxSup / gxInf / gySup / gyInf / gzSup / gzInf = 2000mG; freq = 7.81; timerA = 500; timerB = 500; sensitivity = 3; **activity = 1**; **additionateActivity = 1**; **periodicActivity = 1**; **activityResumePeriod = 1**; *loraAtStartMvt = 1; loraAtStopMvt = 1* (*parameter ignored*)



LoRa frame "1" sent: **c11a400000004b00aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000004b00**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00004B00**: duration = 19200ms
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "2" sent: **c11a400000000ed00aa**

- Payload header: **c11a**

  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000000ed00**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x0000ED00**: duration = 60672ms (=19200+41472ms)
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "3" sent: **c11a400000011800aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000011800**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00011800**: duration = 71680ms (=60672+11008ms)
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "4" sent: **c11a400000013180aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **400000013180**
  - **0x40**: Data type = ACTIVITY => expected payload data = Motion/Stillness (1 byte), duration (4 bytes unsigned)
    - **0x00**: Motion
      - **0x00013180**: duration = 78208ms (=71680+6528ms)
- Payload end of frame: **aa**
  - **0xAA**: End of frame

If **periodicActivity = 0**, LoRa payloads would be the activity on the corresponding activityResumePeriod time slot (e.g. LoRa frame 3 payload would have been **0x0000ED00**: duration = 60672ms)

3.2.7. TEMPERATURE algorithm

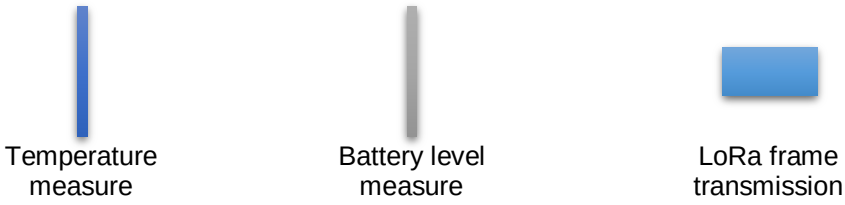
3.2.7.1. TEMPERATURE parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
TEMPERATURE	nbSavedValue	Number of values to measure and store between 2 frames	1	12	48	N/A	Integer
	modeThreshold	Activates Temperature threshold	0	1	1		Boolean
	max	Max temperature threshold	-32768	2500	32767	0.01 °C	°C
	min	Min temperature threshold	-32768	2000	32767		
	delta	Gap before a threshold (min or max) is reached. If the gap is reached, the period between 2 measures is set to fastPeriod	0	100	255		
	period	Period between 2 measures when the temperature is normal (gap not reached)	500	30000	2,16E+07	ms	S
	fastPeriod	Period between 2 measures when the temperature has reached the defined gap (min or max)	250	5000	65535		
	ultraFastPeriod	Period between 2 measures when the temperature has reached the threshold (min or max)	150	2000	65535		
	inhibition	Number of values above max threshold or under min threshold before an alert is sent	0	3	255	N/A	Integer

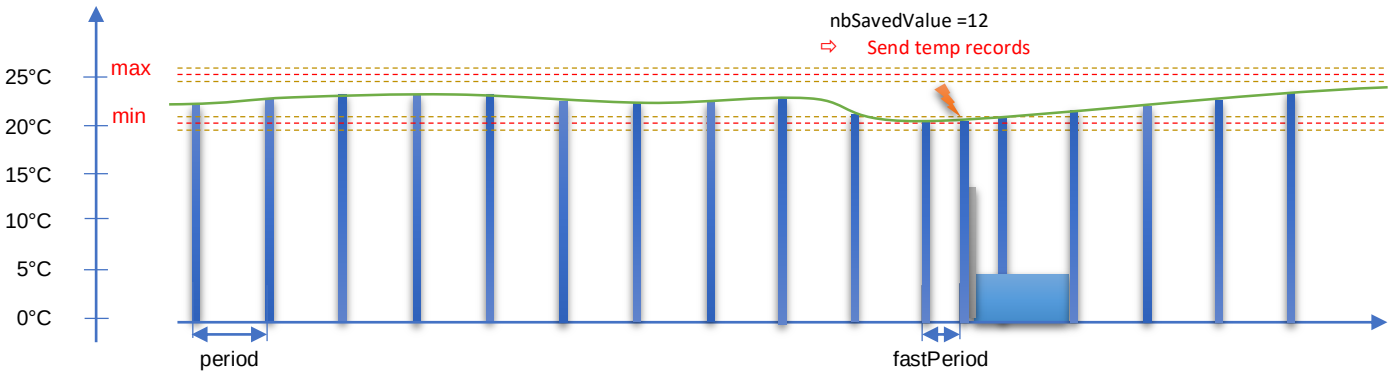
See *§3.4 LoRa frames payload description* for payload decoding

For SW version < 1.8.8, period value is limited to 32400 s

3.2.7.2. TEMPERATURE implementation



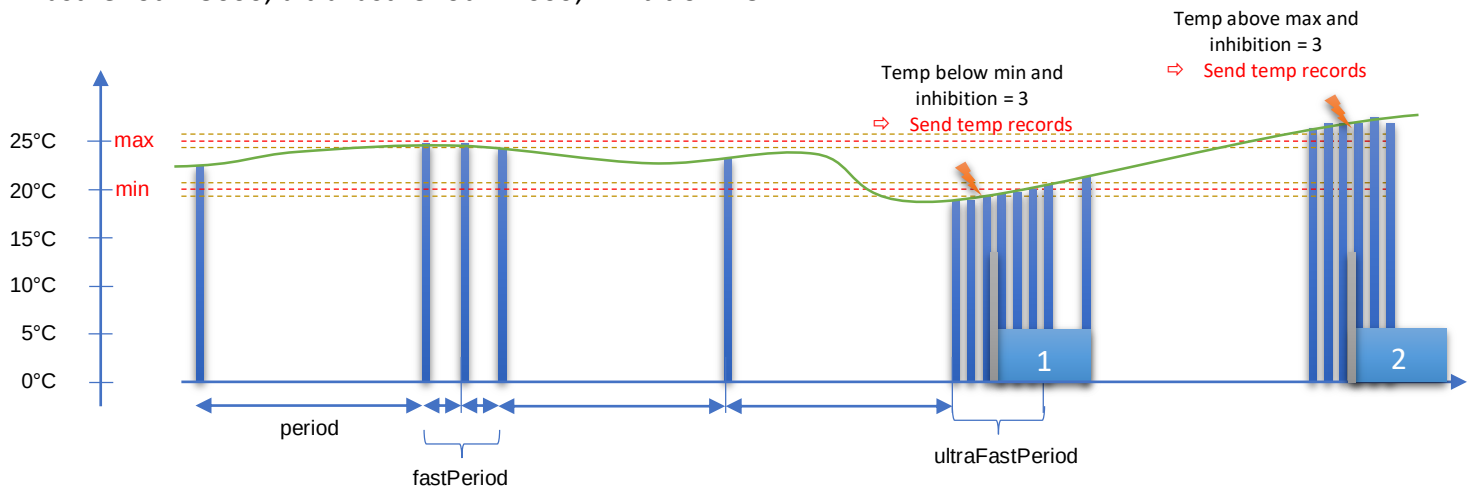
Example 1: nbSavedValue = 12; modeThreshold = 1; max = 2500; min = 2000; delta = 100; period = 30000; fastPeriod = 15000; ultraFastPeriod = 2000; inhibition = 3



LoRa frame sent: **c115021515161817171718181817 16aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **021515161817171718181817 16**
  - **0x02**: Data type = TEMPERATURE => expected payload data = N bytes (signed)
    - **0x15**: Recorded temperature 1 = 21°C
    - **0x15**: Recorded temperature 2 = 21°C
    - **0x16**: Recorded temperature 3 = 22°C
    - **0x18**: Recorded temperature 4 = 24°C
    - **0x17**: Recorded temperature 5 = 23°C
    - **0x17**: Recorded temperature 6 = 23°C
    - **0x17**: Recorded temperature 7 = 23°C
    - **0x18**: Recorded temperature 8 = 24°C
    - **0x18**: Recorded temperature 9 = 24°C
    - **0x18**: Recorded temperature 1 = 24°C
    - **0x17**: Recorded temperature 1 = 23°C
    - **0x16**: Recorded temperature 1 = 22°C
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: nbSavedValue = 12; modeThreshold = 1; max = 2500; min = 2000; delta = 100; period = 30000; fastPeriod = 5000; ultraFastPeriod = 2000; inhibition = 3



LoRa frame "1" sent: **c113021312121718181816aa**

- Payload header: **c113**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x13**: Temperature = 19°C
- Payload data : **021312121718181816**
  - **0x02**: Data type = TEMPERATURE => expected payload data = N bytes (signed)
    - **0x13**: Recorded temperature 1 = 19°C
    - **0x12**: Recorded temperature 2 = 18°C
    - **0x12**: Recorded temperature 3 = 18°C
    - **0x17**: Recorded temperature 4 = 23°C
    - **0x18**: Recorded temperature 5 = 24°C
    - **0x18**: Recorded temperature 6 = 24°C
    - **0x18**: Recorded temperature 7 = 24°C
    - **0x16**: Recorded temperature 8 = 22°C
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "2" sent: **c11b021b1b1a1615141413aa**

- Payload header: **c11b**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1B**: Temperature = 27°C
- Payload data : **021b1b1a1615141413**
  - **0x02**: Data type = TEMPERATURE => expected payload data = N bytes (signed)
    - **0x1B**: Recorded temperature 1 = 27°C
    - **0x1B**: Recorded temperature 2 = 27°C
    - **0x1A**: Recorded temperature 3 = 26°C
    - **0x16**: Recorded temperature 4 = 22°C
    - **0x15**: Recorded temperature 5 = 21°C
    - **0x14**: Recorded temperature 6 = 20°C

- **0x14**: Recorded temperature 7 = 20°C
- **0x13**: Recorded temperature 8 = 19°C

- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.2.8. TILT algorithm

3.2.8.1. TILT parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
TILT	signedMeasure	Sets the measure as absolute or signed 0 = absolute 1 = signed **Parameter available for P/N≥3.9**	0	0	1	N/A	Boolean
	alertMultiple	Sets the redundancy of the alert. If set, the device will send alert as long as the measured value is above the threshold (limited by the duty cycle). If not set, the device will send only one alert when threshold is passed - **Parameter available for P/N≥3.9**	0	0	1		
	zoneDetection	Activates the transmission of an alert when threshold is passed, and when the measured angle gets back under the threshold value - **Parameter available for P/N≥3.9**	0	0	1		
	modeRefresh	Activates periodical automatic tilt measurement	0	1	1		
	modeOnMove	Activates tilt measurement on motion detection	0	1	1		
	pitch	Sends a LoRa frame when this threshold is reached (threshold defined as absolute value)	1	45	90	0.1°	Degree
	roll	Sends a LoRa frame when this threshold is reached (threshold defined as absolute value)	1	90	180		
	threshold	Motion detection threshold	50	200	1020	mG	G
	inhibition	Number of values above threshold before tilt calculation	1	5	255	N/A	Integer
	period	Period between 2 measures	250	2000	2,16E+07	ms	s

See *§3.4 LoRa frames payload description* for payload decoding

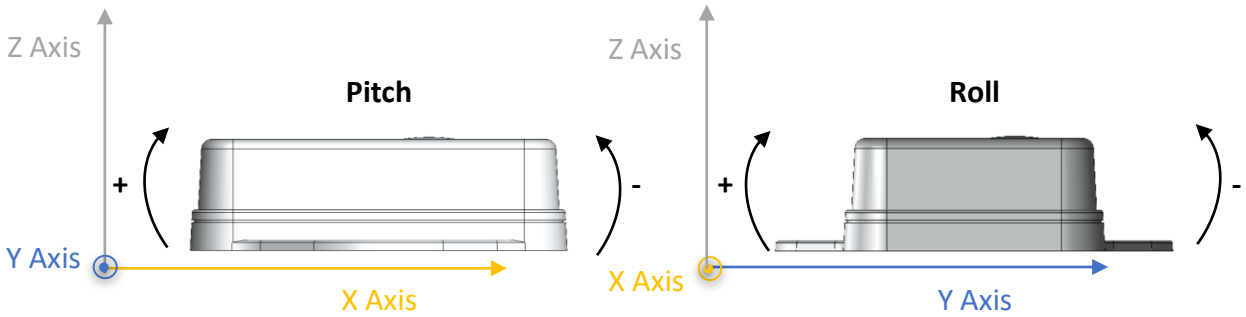
For SW version < 1.8.8, period value is limited to 32400 s

If ***modeOnMove*** is set to 0, ***threshold*** parameter is ignored.

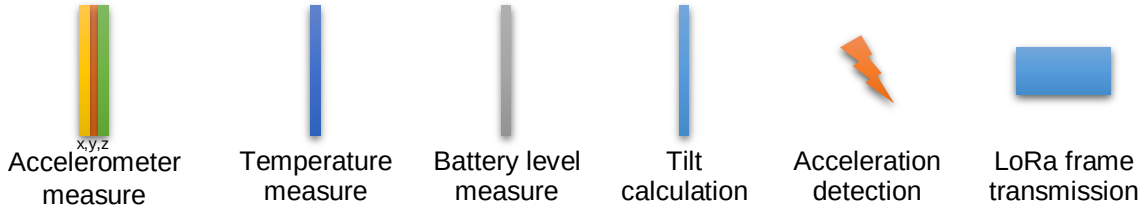
If ***modeRefresh*** is set to 0, ***period*** parameter is ignored.

modeOnMove and ***modeRefresh*** can be activated at the same time.

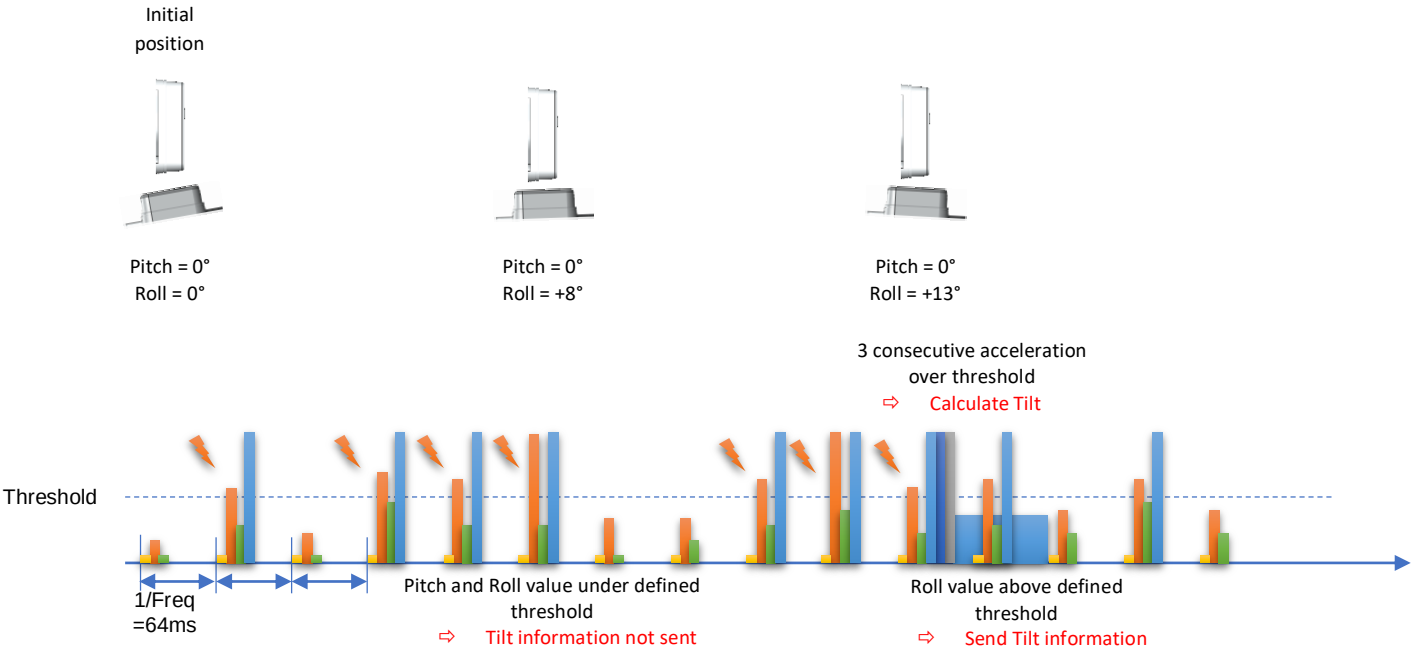
Motion sensor sampling rate is set to 15.63Hz.



3.2.8.2. TILT implementation



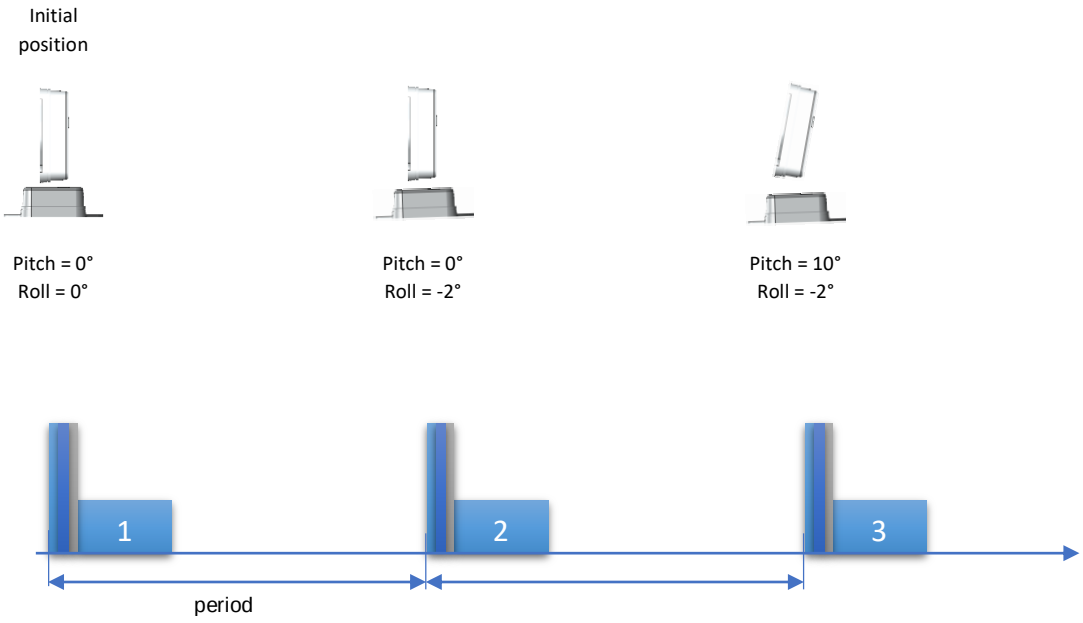
Example 1: modeRefresh = **0**; modeOnMove = **1**; pitch = +10; roll = +10; threshold = 200; inhibition = 3; *period = 10000* (*parameter ignored*)



LoRa frame sent: **c1150800000082aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **0800000082**
  - **0x08**: Data type = TILT => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed)
    - **0x0000**: Pitch angle value = 0 x 0.1° = 0°
    - **0x0082**: Roll angle value = 130 x 0.1° = +13°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: modeRefresh = **1**; modeOnMove = **0**; pitch = 10; roll = 10; *threshold = 200; inhibition = 3*; period = 10000 (*parameter ignored*)



LoRa frame "1" sent: **c11508000000000aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **0800000000**
  - **0x08**: Data type = TILT => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed)
    - **0x0000**: Pitch angle value = 0 x 0.1° = 0°
    - **0x0000**: Roll angle value = 0 x 0.1° = 0°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "2" sent: **c11508000ffecaa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **080000ffec**
  - **0x08**: Data type = TILT => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed)
    - **0x0000**: Pitch angle value = 0 x 0.1° = 0°
    - **0xFFEC**: Roll angle value = -20 x 0.1° = -2°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame “3” sent: **c115080064ffecaa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **080064ffec**
  - **0x08**: Data type = TILT => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed)
    - **0x0064**: Pitch angle value = 100 x 0.1° = +10°
    - **0xFFEC**: Roll angle value = -20 x 0.1° = -2°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.2.9. ROTATION algorithm

3.2.9.1. ROTATION parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
ROTATION	modeRefresh	Activates Rotation measurement (with defined period between 2 measures)	0	1	1	N/A	Boolean
	modeOnMove	Activates rotation measurement if motion is detected	0	1	1		Boolean
	lap	Number of turns before a LoRa frame is sent	1	5	32767		Integer
	resetLap	If activated, a press on the push button (more than 3s and less than 5s = yellow LED) will reset the turn counter - **Parameter available for P/N≥3.9**	0	1	1		Boolean
	threshold	Motion detection threshold	50	200	1020	mG	G
	period	Period between 2 measures (if modeRefresh = 1)	500	5000	65535	ms	S

See *§3.4 LoRa frames payload description* for payload decoding

For SW version < 1.8.8, period value is limited to 32400 s

If ***modeOnMove*** is set to 0, ***threshold*** parameter is ignored.

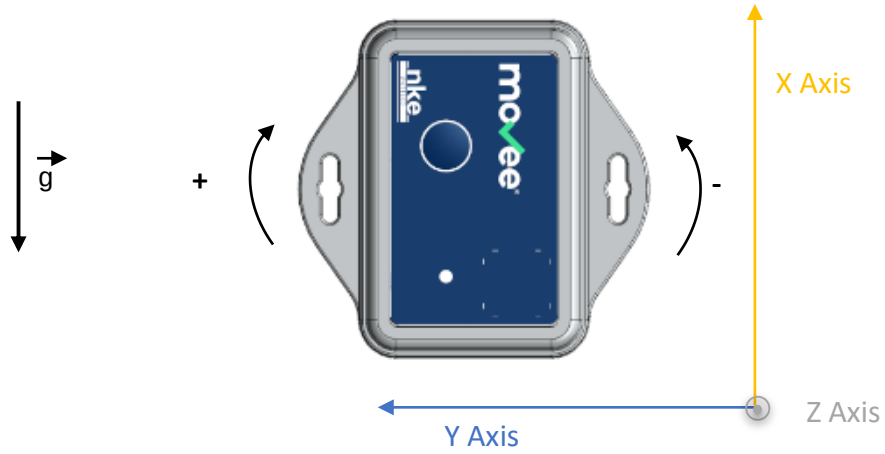
If ***modeRefresh*** is set to 0, ***period*** parameter is ignored.

In ***modeRefresh*** mode, maximum detectable speed is 30 turns/min (period = 500ms), else the number of data acquisition won't be sufficient to get consistent measures.

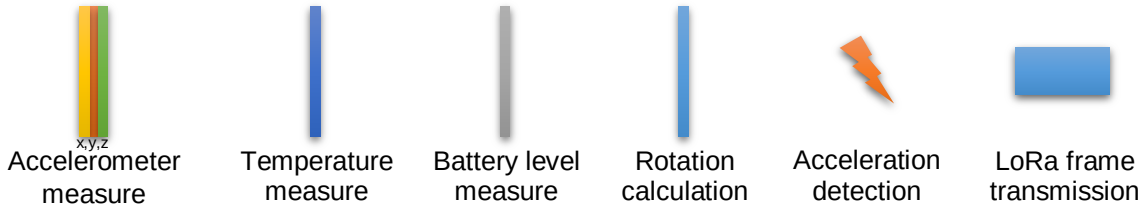
modeOnMove and ***modeRefresh*** can be activated at the same time.

In order to calculate the lap number on a motor axis, a system rotation is operated on the accelerometer value in order to set the gravity ($\vec{g}$) on -X axis, instead of -Z axis in Tilt and Orientation modes. Yaw is then placed on X axis instead of Z axis.

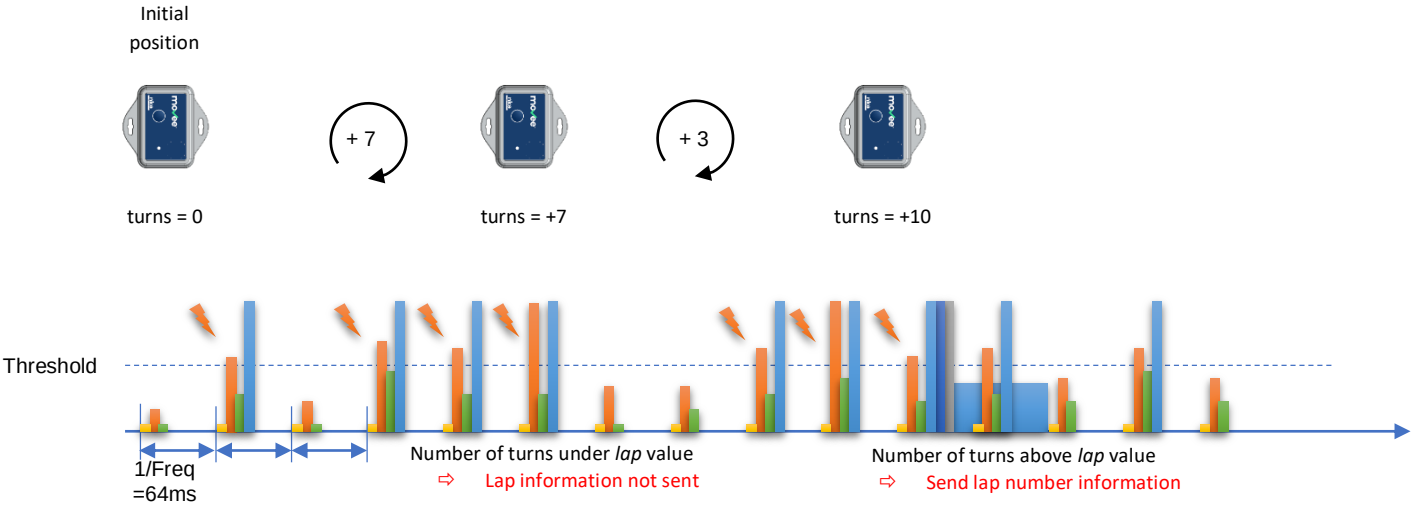
For slow turn count, ***threshold*** shall be set with a low value. Maximum detectable speed is 230 turns/min. Motion sensor sampling rate is set to 15.63Hz.



3.2.9.2. ROTATION implementation



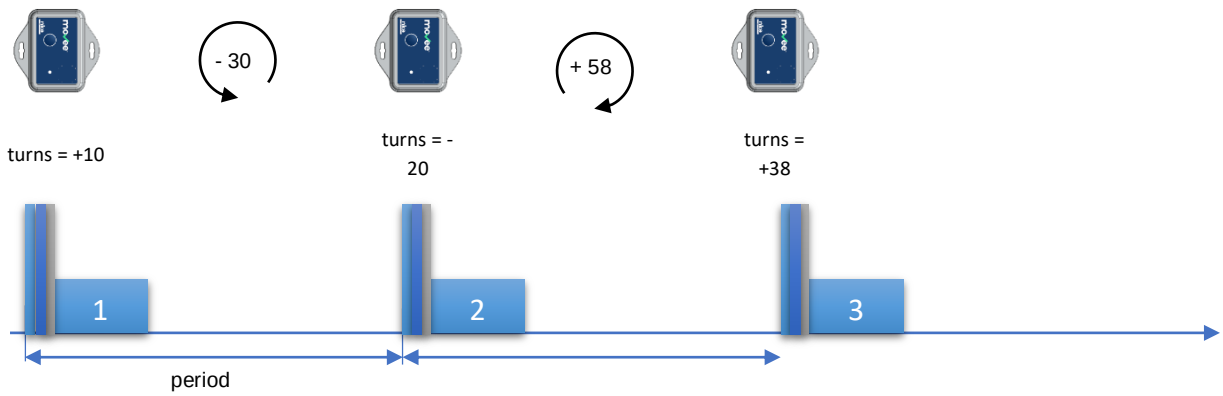
Example 1: modeRefresh = **0**; modeOnMove = **1**; lap = 10; resetLap = 1; threshold = 200; *period = 5000* (*parameter ignored*)



LoRa frame sent: **c11580000aaa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **80000a**
  - **0x80**: Data type = ROTATION => expected payload data = 2 bytes (signed) number of turns
    - **0x000A**: number of turns = +10
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: modeRefresh = **1**; modeOnMove = **0**; *lap = 10*; resetLap = 1; *threshold = 200*; period = 5000 (*parameter ignored*)



LoRa frame "1" sent: **c11580000aaa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **80000a**
  - **0x80**: Data type = ROTATION => expected payload data = 2 bytes (signed) number of turns
    - **0x000A**: number of turns = +10
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame "2" sent: **c11580ffe2aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **80ffe2**
  - **0x80**: Data type = ROTATION => expected payload data = 2 bytes (signed) number of turns
    - **0xFFE2**: number of turns = -20
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame “3” sent: **c115800026aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **800026**
  - **0x80**: Data type = ROTATION => expected payload data = 2 bytes (signed) number of turns
    - **0x0026**: number of turns = +38
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.2.10. ORIENT algorithm

3.2.10.1. ORIENT parameters

	Parameter name	Description	Value			Precision	Unit
			min	typ	max		
ORIENT	period	Minimum period between 2 measures	1	30	65535	ms	s
	modeRefresh	Activates orientation measurement (with defined period between 2 measures)	0	1	1	N/A	Boolean
	modeOnMove	Activates orientation measurement if motion is detected	0	1	1		
	threshold	Acceleration threshold for orientation measurement	0	800	2000	mG	G
	mesureLength	Orientation duration measurement (gyroscope activation period)	3	5	255	s	s
	pitch[3]	Pitch threshold for LoRa frame transmission	-180	90	180	1°	Degree (signed)
	roll[4]	Roll threshold for LoRa frame transmission	-90	-45	90		
	yaw[5]	Yaw threshold for LoRa frame transmission	-180	90	180		

See *§3.4 LoRa frames payload description* for payload decoding

For SW version < 1.8.8, period value is limited to 32400 s

If ***modeOnMove*** is set to 0, ***threshold*** parameter is ignored.

If ***modeRefresh*** is set to 0, ***period*** parameter is ignored.

In ***modeOnMove*** mode, the Yaw information is consistent with the real position as long as the threshold is set with a low value. Pitch and Roll values stay consistent with the real position of the product at any time.

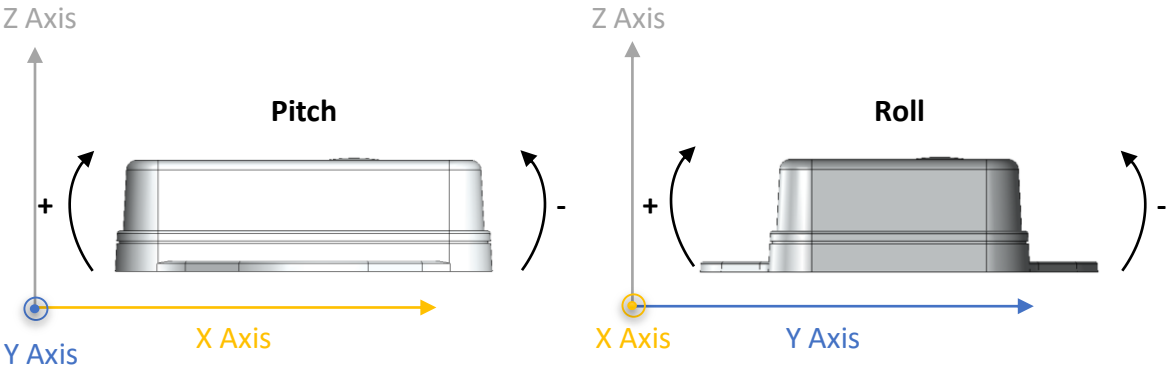
In ***modeRefresh*** mode, the Yaw information is reset at each orientation measure, and any Yaw rotation of the product is lost between 2 measures. Pitch and Roll values stay consistent with the real position of the product at any time.

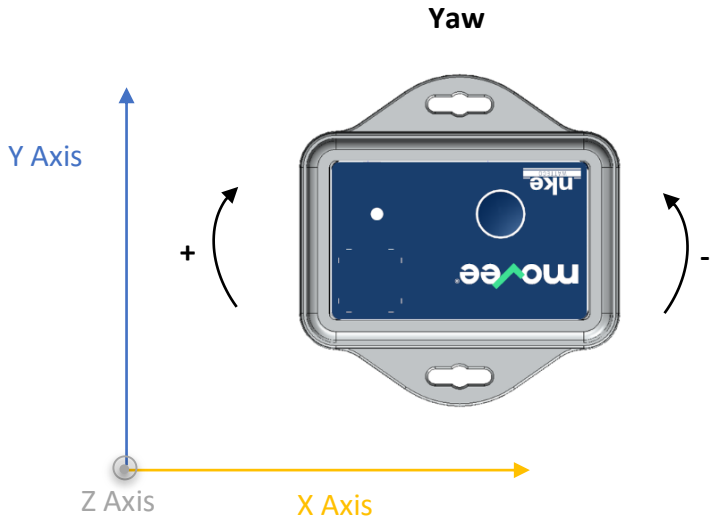
modeOnMove and ***modeRefresh*** can be activated at the same time.

Motion sensor sampling rate is set to 7.81Hz

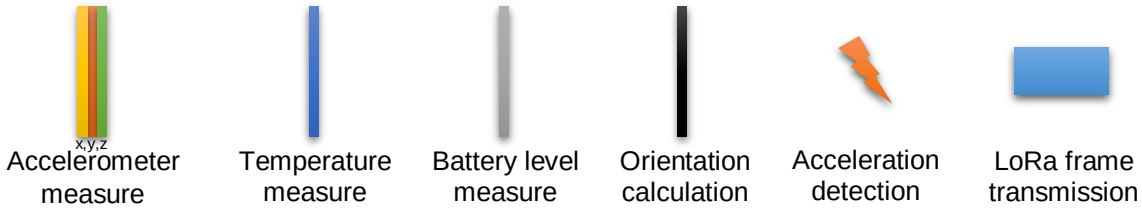
In the ***modeOnMove*** mode, the acceleration threshold is used to start the gyroscope activation and orientation measurement. The orientation measurement is then activated for a ***mesureLenght*** period. Orientation algorithm is more demanding on resources than Tilt algorithm, as a consequence the measure precision is of 1° (instead of 0.1° for Tilt algorithm)

[3] Threshold measured from initial position at device init.
[4] Threshold measured from initial position at device init.
[5] Yaw is measured upon measure trigger, it does not correspond to the device Yaw measured from initial position at device init as the device doesn't have a magnetic compass.

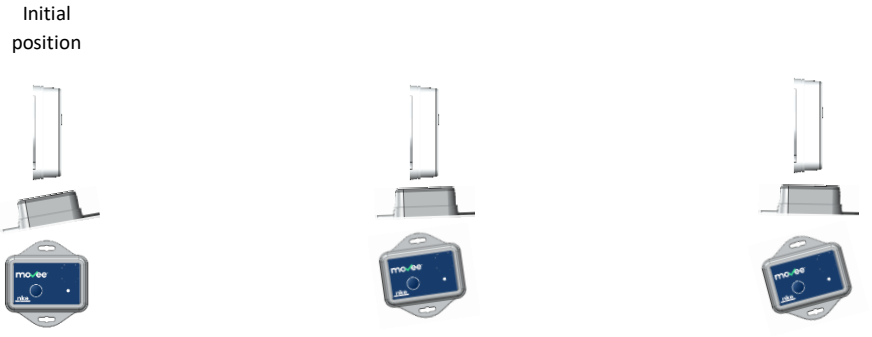

Z Axis
Pitch
+
-
Y Axis
X Axis
Z Axis
Roll
+
-
X Axis
Y Axis

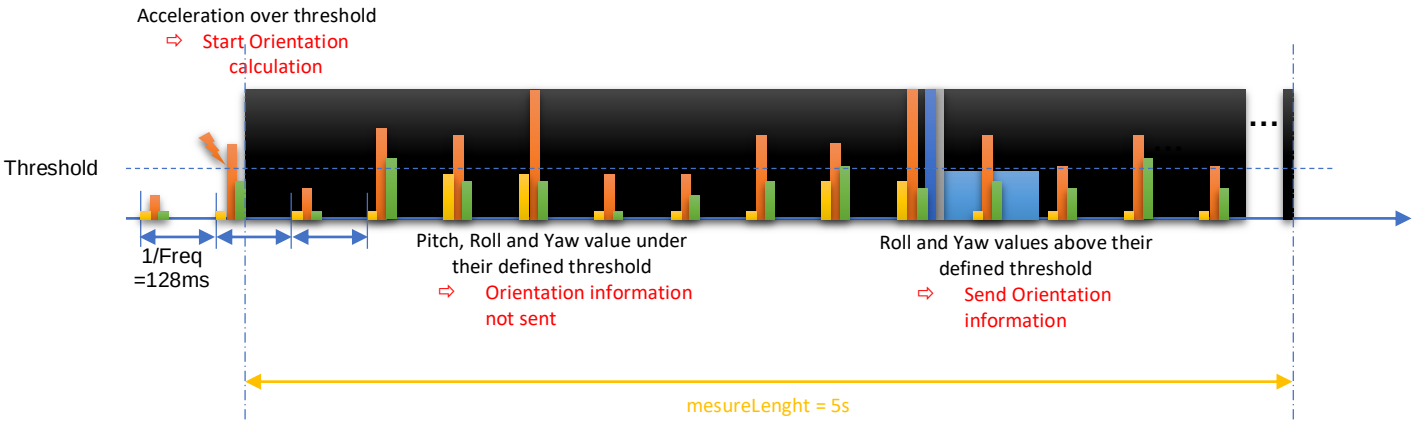

Yaw
Y Axis
+
-
Z Axis
X Axis

3.2.10.2. *ORIENT implementation*



Example 1: *period = 30*; modeRefresh = **0**; modeOnMove = **1**; threshold = 200; mesureLength = 5; pitch = +10; roll = +10; yaw = -14 (*parameter ignored*)

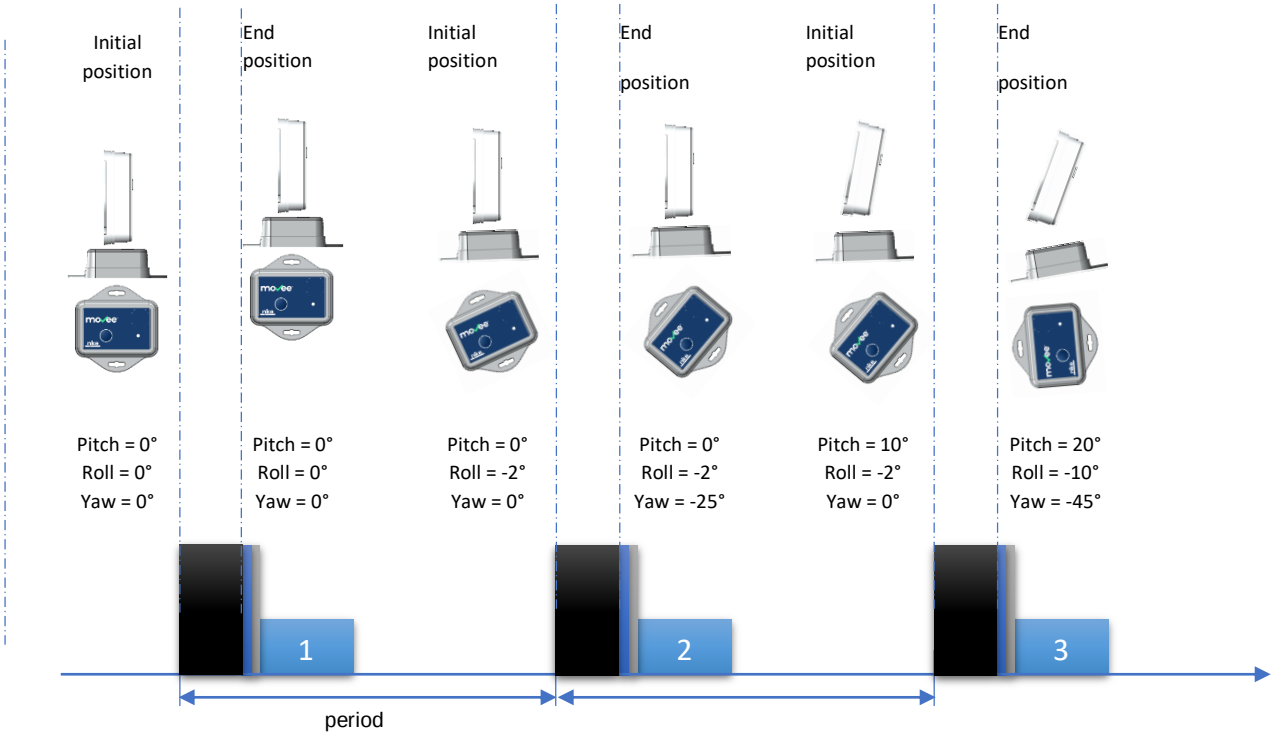




LoRa frame sent: **c115100000000bfff1aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **100000000bfff1**
  - **0x10**: Data type = ORIENTATION => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed) + 2 bytes Yaw (signed)
    - **0x0000**: Pitch angle value = 0 x 1° = 0°
    - **0x000B**: Roll angle value = 11 x 1° = +11°
    - **0xFFF1**: Yaw angle value = -15 x 1° = -15°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

Example 2: period = 30; modeRefresh = **0**; modeOnMove = **1**; *threshold = 200*; mesureLength = 5; *pitch = +10; roll = +10; yaw = -14* (*parameter ignored*)



LoRa frame "1" sent: **c115100000000000000aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **100000000000000**
  - **0x10**: Data type = ORIENTATION => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed) + 2 bytes Yaw (signed)
    - **0x0000**: Pitch angle value = 0 x 1° = 0°
    - **0x0000**: Roll angle value = 0 x 1° = 0°
    - **0x0000**: Yaw angle value = 0 x 1° = 0°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame “2” sent: **c115100000fffeffe7aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **100000fffeffe7**
  - **0x10**: Data type = ORIENTATION => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed) + 2 bytes Yaw (signed)
    - **0x0000**: Pitch angle value = 0 x 1° = 0°
    - **0xFFFE**: Roll angle value = -2 x 1° = -2°
    - **0xFFE7**: Yaw angle value = -25 x 1° = -25°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

LoRa frame “3” sent: **c115100014fff6ffd3aa**

- Payload header: **c115**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x15**: Temperature = 21°C
- Payload data : **100014fff6ffd3**
  - **0x10**: Data type = ORIENTATION => expected payload data = 2 bytes Pitch (signed) + 2 bytes Roll (signed) + 2 bytes Yaw (signed)
    - **0x0014**: Pitch angle value = 20 x 1° = +20°
    - **0xFFF6**: Roll angle value = -10 x 1° = -10°
    - **0xFFD3**: Yaw angle value = -45 x 1° = -45°
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.2.11. VIBE algorithm

3.2.11.1. VIBE parameters

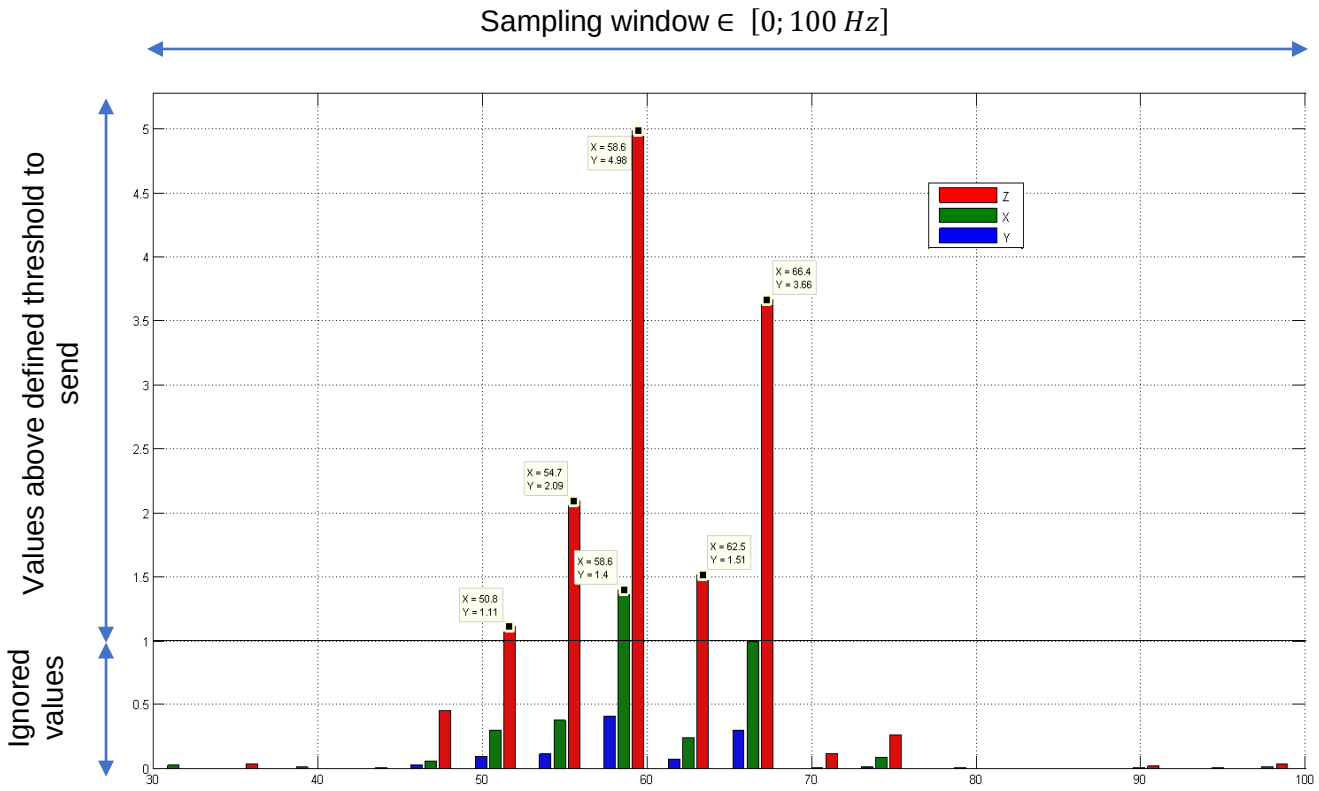
	Parameter	Description	Value			Precision	Unit
			min	typ	max		
VIBE	onX	Activates vibration measurement on X-Axis	0	1	1	N/A	Boolean
	onY	Activates vibration measurement on Y-Axis	0	1	1		
	onZ	Activates vibration measurement on Z-Axis	0	1	1		
	period	Minimum period between 2 measures	1	60	65535	ms	s
	amplitudeX	X axis vibration amplitude threshold	100	1000	65535	mG	G
	amplitudeY	Y axis vibration amplitude threshold	100	1000	65535	mG	G
	amplitudeZ	Z axis vibration amplitude threshold	100	1000	65535	mG	G
	freqMin	Minimum frequency (lower limit for the sampling window)	0	0	400	Hz	Hz
	freqMax	Maximum frequency (upper limit for the sampling window)	100	500	500	Hz	Hz

See *§3.4 LoRa frames payload description* for payload decoding

For SW version < 1.8.8, period value is limited to 32400 s

3.2.11.2. VIBE implementation

Example: onX = 1; onY = 1; onZ = 1; period = 60; amplitude = 1000; amplitude = 1000; amplitudeZ = 1000; freqMin = 0; freqMax = 100



LoRa frame sent: **c11a86020456 0d02082a0e00 0578 0f02 1374 0f02 05e6 1002 0e4c 11aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **86020456 0d02082a0e00 0578 0f02 1374 0f02 05e6 1002 0e4c 11**
  - **0x86**: Data type = VIBE => expected payload data = N x **Axis (1 byte)**, **Amplitude (2 bytes unsigned)**, **frequency index (1 byte unsigned)**
    - **0x02**: Z Axis
      - **0x0456**: amplitude = 1110mG
        - **0D**: index = 13 => frequency = 3,90625 x 13 = 50,8HZ
    - **02**: Z Axis
      - **082A**: amplitude = 2090mG
        - **0E**: index = 14 => frequency = 3,90625 x 14 = 54,7HZ
    - **00**: X Axis
      - **0578**: amplitude = 1400mG
        - **0F**: index = 15 => frequency = 3,90625 x 15 = 58,6HZ
    - **02**: Z Axis
      - **1374**: amplitude = 4980mG
        - **0F**: index = 15 => frequency = 3,90625 x 15 = 58,6HZ
    - **02**: Z Axis
      - **05E6**: amplitude = 1510mG
        - **10**: index = 16 => frequency = 3,90625 x 16 = 62,5HZ
    - **02**: Z Axis
      - **0E4C**: amplitude = 3660mG
        - **11**: index = 17 => frequency = 3,90625 x 17 = 66,4HZ
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.3. Algorithms compatibility

In order to guarantee that the device is fully functional, a limitation on the number of algorithm activated at the same time has been set.

The following table shows which algorithm with which other algorithm. The table must be read from left to right, and when a green arrow is encountered from top to bottom.

3.3.1. Algorithms Compatibility for P/N < 3.9

		VIBE	ALIVE		SHOCK	MOTION	TEMP		TILT		ROTATION		ORIENT	
	Mode	-	Auto	Manual	-	-	Continuous	Threshold	Refresh	Motion	Refresh	Motion	Refresh	Motion
VIBE	-	Yes	X	X	X	X	X	X	X	X	X	X	X	X
ALIVE	Auto	No	X	X	X	X	X	X	X	X	X	X	X	X
	Manual	Yes	Yes	X	X	X	X	X	X	X	X	X	X	X
SHOCK	-	No	Yes	Yes	X	X	X	X	X	X	X	X	X	X
MOTION	-	No	Yes	Yes	Yes	X	X	X	X	X	X	X	X	X
TEMP	Continuous	No	No	Yes	Yes	Yes	X	X	X	X	X	X	X	X
	Threshold	No	No	Yes	Yes	Yes	No	X	X	X	X	X	X	X
TILT	Refresh	No	No	Yes	Yes	Yes	No	No	X	X	X	X	X	X
	Motion	No	Yes	Yes	No	No	Yes	Yes	Yes	X	X	X	X	X
ROTATION	Refresh	No	No	Yes	Yes	Yes	No	No	No	No	X	X	X	X
	Motion	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	X	X	X
ORIENT	Refresh	No	No	Yes	Yes	No	No	No	No	Yes	No	Yes	X	X
	Motion	No	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	No	Yes	X

Examples:

- ALIVE algorithm can be activated at the same time as any other algorithm
- TILT algorithm in Motion detection mode can be activated at the same time as
  - ALIVE (auto or manual mode),
  - Or TEMP (continuous or Threshold mode),
  - Or TILT in Refresh mode,
  - Or ORIENTATION in Refresh mode,

    but NOT at the same time as
  - SHOCK,
  - MOTION
  - ROTATION (Refresh or Motion detection mode)
  - ORIENTATION in Motion Detection mode

3.3.2. Algorithms Compatibility for P/N ≥ 3.9

Compatibility between algorithms has been improved from P/N 3.9 and above

		VIBE	ALIVE		SHOCK	MOTION	TEMP		TILT		ROTATION		ORIENT	
	Mode	-	Auto	Manual	-	-	Continuous	Threshold	Refresh	Motion	Refresh	Motion	Refresh	Motion
VIBE	-	Yes	X	X	X	X	X	X	X	X	X	X	X	X
ALIVE	Auto	No	X	X	X	X	X	X	X	X	X	X	X	X
	Manual	Yes	Yes	X	X	X	X	X	X	X	X	X	X	X
SHOCK	-	No	Yes	Yes	X	X	X	X	X	X	X	X	X	X
MOTION	-	No	Yes	Yes	Yes	X	X	X	X	X	X	X	X	X
TEMP	Continuous	No	Yes	Yes	Yes	Yes	X	X	X	X	X	X	X	X
	Threshold	No	Yes	Yes	Yes	Yes	No	X	X	X	X	X	X	X
TILT	Refresh	No	Yes	Yes	Yes	Yes	No	No	X	X	X	X	X	X
	Motion	No	Yes	Yes	Yes	No	Yes	Yes	Yes	X	X	X	X	X
ROTATION	Refresh	No	Yes	Yes	Yes	Yes	No	No	No	No	X	X	X	X
	Motion	No	Yes	Yes	Yes	No	Yes	Yes	No	No	Yes	X	X	X
ORIENT	Refresh	No	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	X	X
	Motion	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	Yes	X

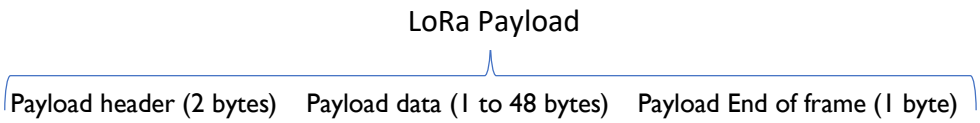
Examples:

- ALIVE algorithm can be activated at the same time as any other algorithm
- TILT algorithm in Motion detection can be activated at the same time as
  - ALIVE (auto or manual mode),
  - Or SHOCK,
  - Or TEMP (continuous or Threshold mode),
  - Or TILT in Refresh mode,
  - Or ORIENTATION in Refresh mode,
    but NOT at the same time as
  - MOTION
  - ROTATION (Refresh or Motion detection mode)
  - ORIENTATION in Motion Detection mode

3.4. LoRa frames payload description

The data are sent in the payload of LoRa frames. A specific payload layout has been defined in order to detect which algorithm is running, and allows to retrieve the corresponding data.

The payload will then change depending on which algorithm is running on the product. The payload structure is defined as:



3.4.1. Payload header

The payload header is always sent, and contains the battery level and the temperature measures:

Name	**Battery level**	**Temperature**
Size	1 byte	1 byte
Coding	8 bits (unsigned)	8 bits (signed)
Coded value:	2,8V…3,6V	-127°C…+128°C

The battery level is coded on the first byte (MSB) as an unsigned byte, the value is then between 0 and 255.
0 is for Vbat=2.8V, and 255 is for Vbat=3.6V.
Each step equals $\frac{(3,6-2,8)}{255} \cong 3,14\ mV$

The temperature is coded on the second byte (LSB) as a signed byte, the value is between -128 and +127. The value is in Celsius degrees.

3.4.2. Payload data

The payload data is made of an MSB "data type" byte (unsigned) describing the algorithm sending the data, followed by the corresponding data.

Data type		Data 1			Data 2			Data 3		
Name	**Value**	**Name**	**Coding**	**Unit**	**Name**	**Coding**	**Unit**	**Name**	**Coding**	**Unit**
ALIVE	0x01	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
TEMPERATURE[6]	0x02	Temp1	8 bits (signed)	°C	Temp2	8 bits (signed)	°C	Temp3	8 bits (signed)	N/A
ALIVE-BUTTON	0x03	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SHOCK	0x04	Gx	16 bits (signed)	mG	Gy	16 bits (signed)	mG	Gz	16 bits (signed)	N/A
TILT	0x08	Pitch	16 bits (signed)	0.1°	Roll	16 bits (signed)	0.1°	N/A	N/A	N/A
ORIENT	0x10	Pitch	16 bits (signed)	1°	Roll	16 bits (signed)	1°	Yaw	16 bits (signed)	1°
MOTION	0x20	Motion	0x00	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		Stillness	0x01	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ACTIVITY	0x40	Motion	0x00	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		Stillness	0x01	N/A	Activity duration	32 bits (unsigned)	ms	N/A	N/A	N/A
		Activity Summary	0x02	N/A	Activity duration	32 bits (unsigned)	ms	N/A	N/A	N/A
ROTATION	0x80	Number of turns	16 bits (signed)	turns	N/A	N/A	N/A	N/A	N/A	N/A
VIBRATION[7]	0x86	X axis	0x00	N/A	Vibration amplitude	16 bits (unsigned)	mG	Index[8] (*i*) of the vibration frequency	8 bits (unsigned)	N/A
		Y axis	0x01	N/A						
		Z axis	0x02	N/A						
SERVICE	0xFF	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

For VERSION frame, values are in ASCII, each value is separated by a ";", coded as 0x3B in hexadecimal

Data type		Data 1			Data 2			Data 3			Data 4		
Name	**Value**	**Name**	**Coding**	**Unit**	**Name**	**Coding**	**Unit**	**Name**	**Coding**	**Unit**	**Name**	**Coding**	**Unit**
VERSION	0xFE	Product version	ASCII	N/A	SW version	ASCII	N/A	LoRa stack version	ASCII	N/A	Reset cause	ASCII	N/A

[6] TEMPERATURE algorithm can send up to 50 successive values.
Temp1 temperature value is the latest temperature measurement before the transmission of the data.
TempN temperature values are the recorded values since the last transmission.

[7] For vibration frames, the number of data depends on the result of the measure. Maximum size for a vibration frame is 48 bytes (12 x data packets)

[8] Frequency is retrieved with the following formula: $f = 3{,}90625 * i$ Hz

3.4.3. Payload End of frame

The end of the payload is defined as a byte with a specific value: 0xAA

Size	1 byte
Coding	-
Value:	0xAA

3.4.4. Payload examples

3.4.4.1. ALIVE payload

Payload example: **e61801aa**

- Payload header: **e618**
  - **0xE6**: battery level : 0xE6 = 230: $\frac{(3,6-2,8)}{255} * 230 + 2,8 = 3,52\ Volts$
  - **0x18**: Temperature = 24°C
- Payload data : **01**
  - **0x01**: Data type = ALIVE => no more payload data
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.4.4.2. SHOCK algorithm payload

Payload example: **c11a040290fe700db0aa**

- Payload header: **c11a**
  - **0xC1**: battery level : 0xC1 = 193: $\frac{(3,6-2,8)}{255} * 193 + 2,8 = 3,4\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **040290fe700db0**
  - **0x04**: Data type = SHOCK => expected payload data = Gx (2 bytes unsigned), Gy (2 bytes unsigned), Gz (2 bytes unsigned)
    - **0x0290FE700DB0**
      - **0x0290**: Gx = 656mG
      - **0xFE70**: Gy = -400mG
      - **0x0DB0**: Gz = 3504mG
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.4.4.3. TEMPERATURE algorithm payload

Payload example: **be1a021a1919191919191919aa**

- Payload header: **be1a**
  - **0xBE**: battery level : 0xBE = 230: $\frac{(3,6-2,8)}{255} * 190 + 2,8 = 3,40\ Volts$
  - **0x1A**: Temperature = 26°C
- Payload data : **021a1919191919191919**
  - **0x02**: Data type = TEMPERATURE => expected payload data = N bytes (signed)
    - **0x1A1919191919191919**
      - **0x1A**: temp = 26°C (latest measure)
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
      - **0x19**: temp = 25°C
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.4.4.4. VERSION payload

The VERSION frame is the first frame sent after a Join Accept in OTAA or is sent after a VERSION Downlink request.

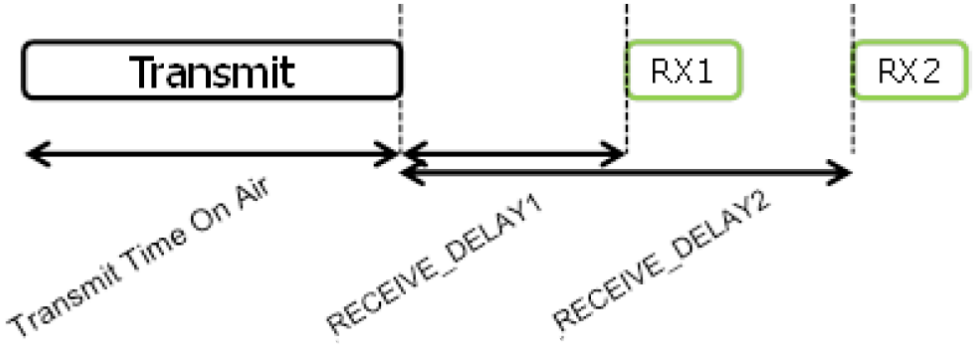
Payload example: **8a19fe76332e373b76312e382e373b76342e332e31623b7273743a504f52aa**

- Payload header: **8a19**
  - **0x8A**: battery level : 0x8A = 138: $\frac{(3,6-2,8)}{255} * 138 + 2,8 = 3,125\ Volts$
  - **0x19**: Temperature = 25°C
- Payload data : **fe76332e373b76312e382e373b76342e332e31623b7273743a504f52**
  - **0xFE**: Data type = VERSION => expected payload data = Product version (N bytes); SW version (N bytes); LoRa Stack version (N bytes); Reset cause(N bytes)
    - **0x76332E373B76312E382E373B76342E332E31623B7273743A504F52**
      - **0x76332E373B**: Product version = “v3.7;”
      - **0x76312E382E373B**: SW version = “v1.8.7;”
      - **0x76342E332E31623B**: LoRa stack version = “v4.3.1b”
      - **0x7273743A504F52**: Reset cause = “rst:POR”
- Payload end of frame: **aa**
  - **0xAA**: End of frame

3.5. Downlink

Only available on Product version 1.2 and above (SW version 1.7 and above)

LoRaTM protocol offers a bidirectional link between the sensor and the LoRaTM Core Network Server. The data transfer is always initiated by the sensor, sending an uplink to the LoRaTM Core Network Server. Once the sensor has sent the uplink, it will open 2 receive windows, which can be used by the LoRaTM Core Network Server to initiate a Downlink transmission.



Movee is able to get parameters updates Over The Air (OTA), through LoRa™ downlink frame transmission. The downlink frame transmission can not only change the parameters, but also the algorithm running on the product.
The downlink cannot be used to update the embedded firmware of the product.

Downlink frame transmission allows 2 actions:

- Change the parameters and running algorithm
- Execute a remote action on the product (e.g.: save parameters, restart the product…)

3.5.1. Ports

Commands are received/sent on port 1.

Parameters are received/sent on port 2.

3.5.2. Commands

It is possible to send up to 3 commands in a single downlink frame. Upon reception, the product will execute these commands one after another (starting by Command ID#1). The following table describes the downlink frame format to send commands to the product:

Name	Command ID #1	Command ID #2	Command ID #3	0xFF
Syze	1 byte	1 byte	1 byte	1 byte
	Mandatory	*Option*	*Option*	Mandatory

The following table gives the command list:

Command ID	**Name**	**Description**
00	Reserved	Reserved
01	Save	Save the updated parameters on the internal flash
02	Clean parameters	Erase all the parameters loaded in RAM. Each parameter is then set to 0
03	Reset parameters	Restaure default parameters
04	Reset board	Reset the product
05	Service mode	Set the product in service mode
06	Normal mode	Restaure normal mode of operation (= exit service mode)
07	Version	Send a frame with product version details
08	RFU	
09	RFU	
0A	Dump RAM param	Dump the parameters loaded in RAM on the serial output
0B	Dump FLASH param	Dump the parameters saved in flash on the serial output
0C	Enable debug	Enable detailed debug traces on debug output
0D	Disable debug	Disable detailed debug traces on debug output
0E	Enable LED	Enable LED activity with motion detection
0F	Disable LED	Disable LED activity with motion detection
10	ADR On	Enable ADR for LoRa communication
11	ADR Off	Disable ADR for LoRa communication
12	ACK On	Enable confirmed frames for LoRa communication
13	ACK off	Disable confirmed frames for LoRa communication

3.5.3. Parameters

It is possible to send up to 10 parameters in a single frame. Upon reception, the product will modify the parameters one by one. The following table describes the downlink frame format to change parameters on the product

Name	Parameter #1		Parameter #2		…	Parameter #10		EOF[9]
Content	ID	Valeur	ID	Valeur		ID	Valeur	0xFF
Size (bytes)	1	4	1	4	…	1	4	1
Status	Mandatory		*Optional*		…	*Optional*		Mandatory

Note: the size of the received payload is between 6 and 51 bytes

Most time relative parameters (period, timer) are to be defined in ms.

[9] EOF = End Of Frame

The following table gives the parameters list - **Most time relative parameters (period, timer) are to be defined in ms**:

ID (hexa)	Algorithm	Name
1	MPU	maxRange
2	Algo	choseAlgo[10]
3	Common	alertPeriod (only for P/N≥3.9)
4	Common	Poweroff (only for P/N≥3.9)
B	Alive	modeRefresh
C	Alive	Period (defined in ms)
D	Alive	nbSavedValue
13	Shock	gxSup
14	Shock	gxInf
15	Shock	gySup
16	Shock	gyInf
17	Shock	gzSup
18	Shock	gzInf
19	Shock	freq
1A	Shock	Inhibition (defined in ms)
1B	Shock	removeGravity
21	Motion	gxSup
22	Motion	gxInf
23	Motion	gySup
24	Motion	gyInf
25	Motion	gzSup
26	Motion	gzInf
27	Motion	Freq
28	Motion	timerA (defined in ms)
29	Motion	timerB (defined in ms)
2A	Motion	sensitivity
2B	Motion	activity
2C	Motion	additionateActivity
2D	Motion	periodicActivity
2E	Motion	activityResumePeriod (defined in hours)
2F	Motion	loraAtStartMvt
30	Motion	loraAtStopMvt
36	Temp	nbSavedValue
37	Temp	modeThreshold
38	Temp	max
39	Temp	min
3A	Temp	delta
3B	Temp	period (defined in ms)
3C	Temp	fastPeriod (defined in ms)
3D	Temp	ultraFastPeriod (defined in ms)
3E	Temp	inhibition
41	Tilt	signedMeasure (only for P/N≥3.9)
42	Tilt	alertMultiple (only for P/N≥3.9)
43	Tilt	zoneDetection (only for P/N≥3.9)
44	Tilt	modeRefresh
45	Tilt	modeOnMove
46	Tilt	pitch
47	Tilt	roll
48	Tilt	threshold
49	Tilt	inhibition
4A	Tilt	Period (defined in ms)
50	Rotation	modeRefresh
51	Rotation	modeOnMove
52	Rotation	lap
53	Rotation	resetLap
54	Rotation	threshold
55	Rotation	period (defined in ms)
5B	Orient	period (defined in ms)
5C	Orient	modeRefresh
5D	Orient	modeOnMove
5E	Orient	threshold
5F	Orient	mesureLenght (defined in seconds)
60	Orient	Pitch
61	Orient	Roll
62	Orient	Yaw
68	Vibe	onX
69	Vibe	onY
6A	Vibe	onZ
6B	Vibe	Period (defined in ms)
6C	Vibe	amplitudeX
6D	Vibe	amplitudeY
6E	Vibe	amplitudeZ
6F	Vibe	freqMin
70	Vibe	freqMax
71	LoRa	JRTimerReset (defined in hours / only for P/N≥3.7)
72	LoRa	DEFAULT_DR (only for P/N≥3.9)

Note : Parameters with a grey background need a reset to be taken into account

[10] See: *§ **Erreur ! Source du renvoi introuvable. Erreur ! Source du renvoi introuvable.***

3.5.4. Algorithm selection

The parameters allow the selection of the algorithm(s) running on the product, **please refer to *§3.3 Algorithms compatibility* for algorithm mutual compatibility**.

In order to modify the activated algorithm(s), it is necessary to code the *chosealgo* parameter value, which is coded on 9 bits:

Bit	8	7	6	5	4	3	2	1	0
Name	VIBE	ROTATION	Reserved	MOTION	ORIENT	TILT	SCHOCK	TEMP	ALIVE

3 examples on how to code the *chosealgo* parameter:

- ALIVE and SCHOCK activation: *chosealgo* = $1 * 2^0 + 1 * 2^2$ = 0x05
- MOTION and SHOCK activation: *chosealgo* = $1 * 2^2 + 1 * 2^5$ = 0x24
- VIBE activation : *chosealgo* = $1 * 2^8$ = 0x100

Please refer to *§3.3 Algorithms compatibility* for algorithm mutual compatibility.

3.5.5. Downlink examples

3.5.5.1. Enter Service mode

If you want to change some parameters, you can either send the parameters directly on port N°2, or use the service mode to send a long list of parameters.

Entering service mode will force the Movee to send uplink frames every ~2 minutes, in order to get the ability to send the needed Downlink messages as fast as possible.

Activate Service mode

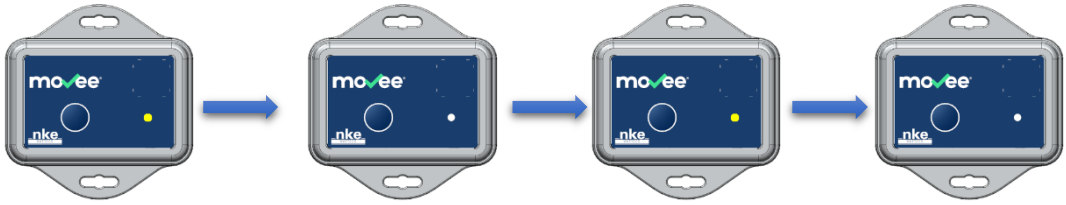
=> Service mode command ID = **0x05**
=> End of Downlink message = **0xFF**

In order to enter Service mode, send:

05FF on port #1 (commands are sent on port #1, parameters on port #2)

Once the downlink is received by the device, it will start blinking yellow.

Once the downlink is received by the device, it will start blinking yellow.



The device will keep blinking as long as it has not exited the Service mode.

Note: Before asking the network to send another downlink, make sure that the “enter service mode” downlink has been sent by the network. If you do not wait for the first downlink to be sent, the network will cancel the first downlink, and replace it with the new one.

3.5.5.2. Send parameters

3.5.5.2.1. Shock and Alive example

In this example, we will activate Shock and Alive modes, with the following configuration:

Alive:

- modeRefresh parameter: Refresh mode activated
- period parameter: period set to 10 minutes
- nbSavedValue parameter: One period between 2 LoRa frames, with temperature measure for each period.

MPU:

- maxRange parameter: set maximum range to 8000mG

Shock:

- gxSup, gxInf, gySup, gyInf, gzSup, gzInf parameters: set threshold to 1250mG on each axis
- freq parameter: not changed (use previously stored value)
- inhibition parameter: de-activate inhibition (set to 0)
- removeGravity

Note: If parameters are not updated, default or previously stored values will be used.

3.5.5.2.2. Alive parameters update frame

Activate Shock and Alive algorithms

=> choseAlgo ID = **0x02** / Parameter = **0x00000005** (see *§3.5.4 Algorithm selection*)

Set modeRefresh parameter to 1

=> modeRefresh ID = **0x0B** / Parameter = **0x00000001**

Set the period value to 10 minutes = 600 000 ms (**Reminder, period value has to be set in ms**)

=> period ID = **0x0C** / Parameter = **0x000927C0**

Set the number of period between 2 frames to 1

=> nbSavedValue ID = **0x0D** / Parameter = **0x00000001**
=> End of Downlink message = **0xFF**

In order to set this configuration, send (**Reminder: maximum 10 parameters in a single downlink frame**):

02000000050B00000001 0C000927C0 0D00000001FF on port #2 (commands are sent on port #1, parameters on port #2)

3.5.5.2.3. MPU & Shock parameters update frame

Set MPU range to 8000mG: Shock + Alive

=> maxRange ID = **0x01** / Parameter = **0x00001F40**

Set gxSup, gxInf, gySup, gyInf, gzSup, gzInf parameters to 1250mG

=> gxSup ID = **0x13** / Parameter = **0x00004E2**
=> gxInf ID = **0x14** / Parameter = **0x00004E2**
=> gySup ID = **0x15** / Parameter = **0x00004E2**
=> gyInf ID = **0x16** / Parameter = **0x00004E2**
=> gzSup ID = **0x17** / Parameter = **0x00004E2**
=> gzInf ID = **0x18** / Parameter = **0x00004E2**

Set the inhibition time to 0

=> inhibition ID = **0x1A** / Parameter = **0x00000000**

In order to set this configuration, send (**Reminder: maximum 10 parameters in a single downlink frame**):

0100001F4013000004E214000004E215000004E216000004E217000004E218000004E21A00000000FF on port #2 (commands are sent on port #1, parameters on port #2)

3.5.5.3. Exit service mode

Once the parameters are received by the device, they are stored in RAM. In order to make these new parameters persistent when the device is switched off or reset, it is necessary to send the command to write the new configuration in flash (Command ID = 0x01).

Save parameters and exit Service mode

=> Save command ID = **0x01**
=> Normal mode (= exit Service mode) command ID = **0x06**
=> End of Downlink message = **0xFF**

In order to save the new parameters in flash and exit Service mode, send:

0106FF on port #1 (commands are sent on port #1, parameters on port #2)

3.6. Movee Configurator user interface

3.6.1. PC User interface install

To install the user interface on your computer, please download and **execute as administrator** the "MoveeConfigurator_setup_vx.x.exe" available on nke's support website:

http://support.nke-watteco.com/movee

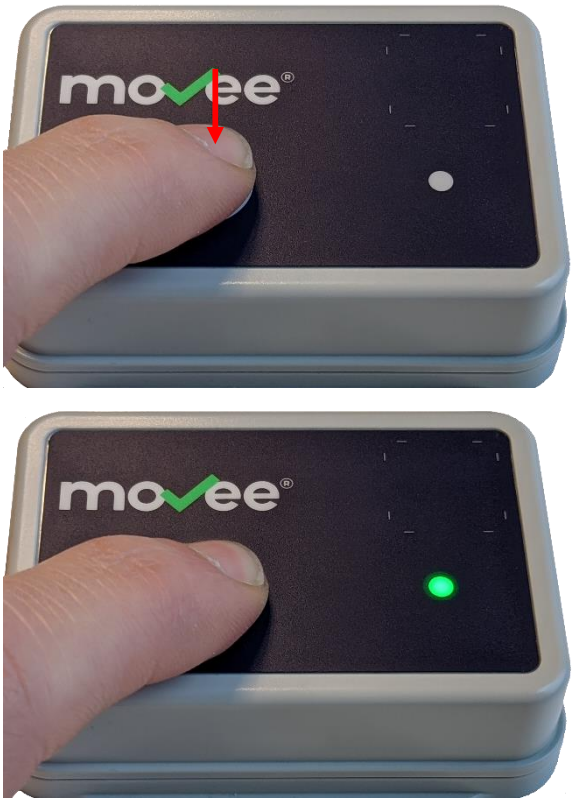
3.6.2. Setup

To access the internal debug interface, remove the 4 screws underneath the product:

Once opened, remove the battery and plug a micro-USB cable on the micro-USB connector:

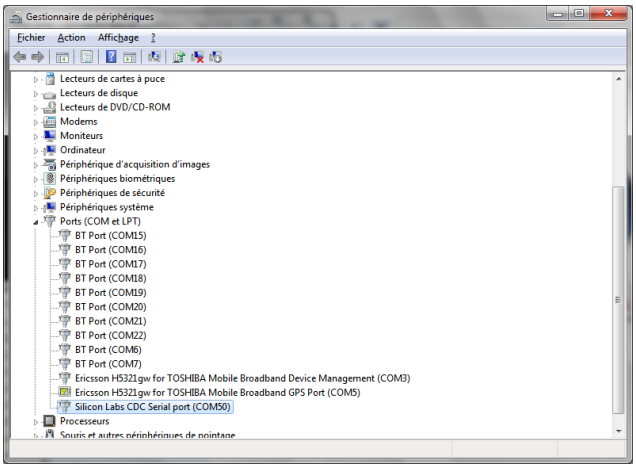
Plug the other side of the Micro-USB cable to a host computer, and put the battery back in place.

Once plugged on the host computer (where you should have installed the configuration “Movee configurator” PC user interface, with the Silicon Labs USB CDC/ACM driver) switch the device on (1s push on the button)



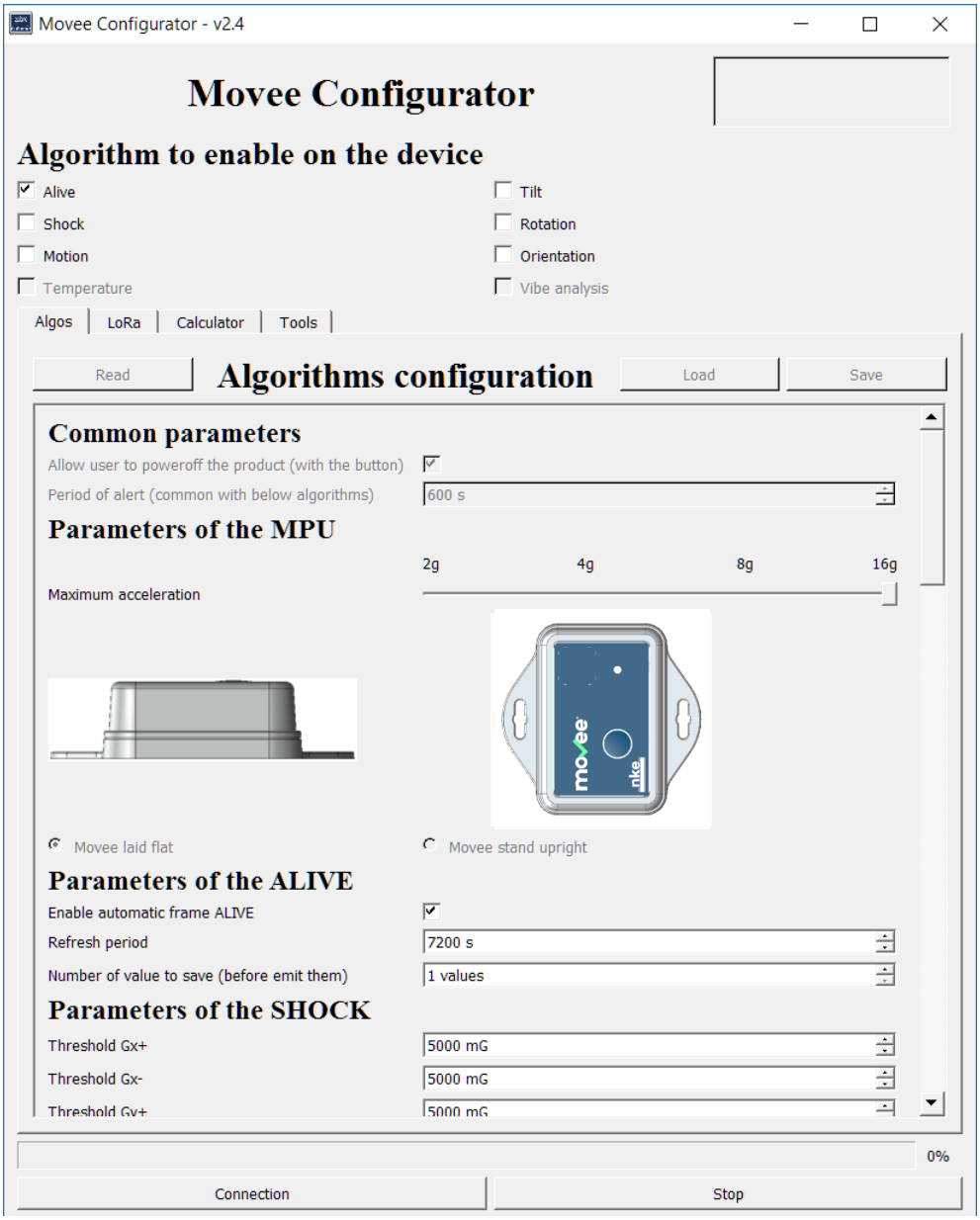
Once the LED has blinked green, release the button

You should then see the device in the COM ports of your computer:



Open the “Movee Configurator” tool

The UI is displayed on your screen:

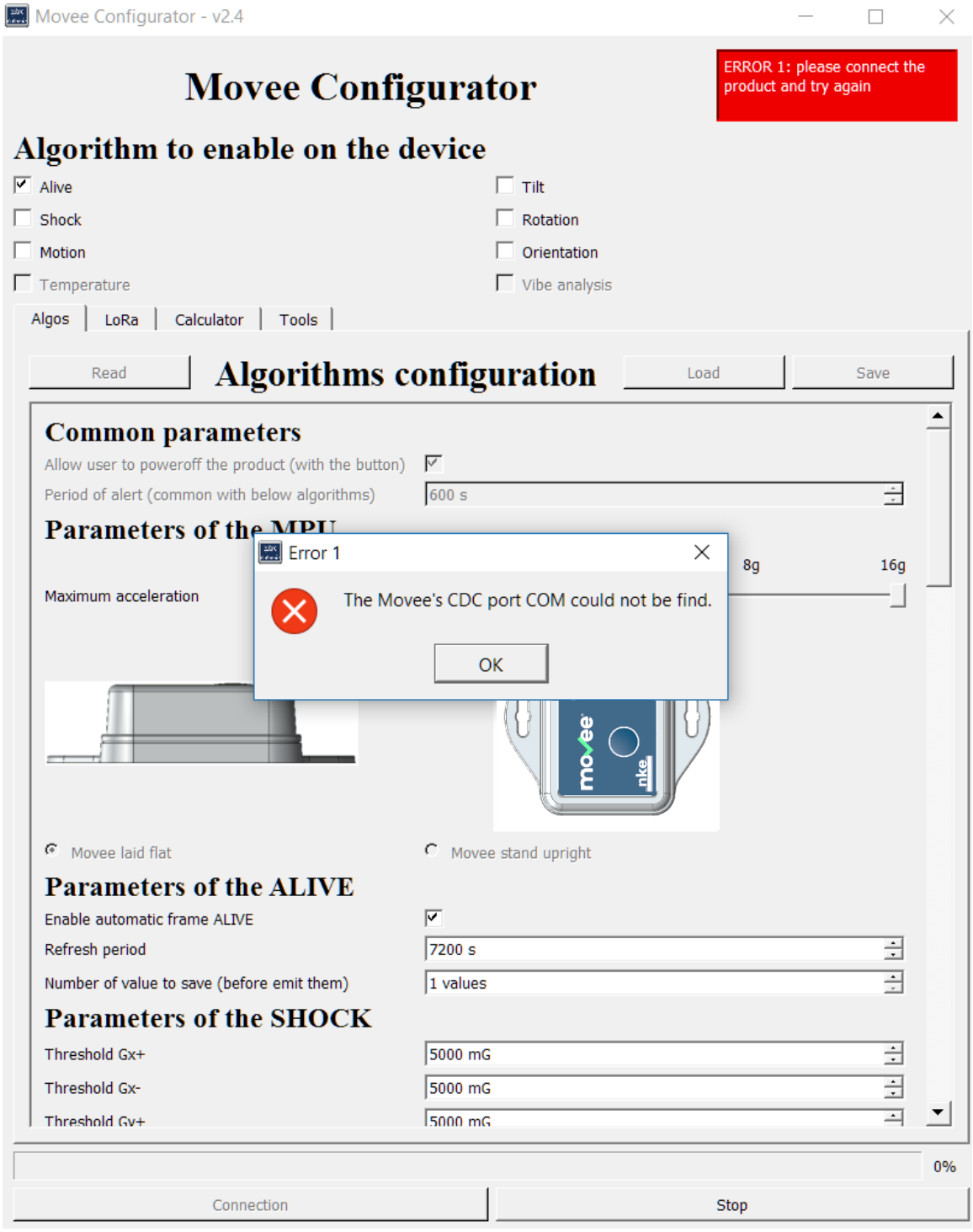


3.6.3. User interface

3.6.3.1. Connection

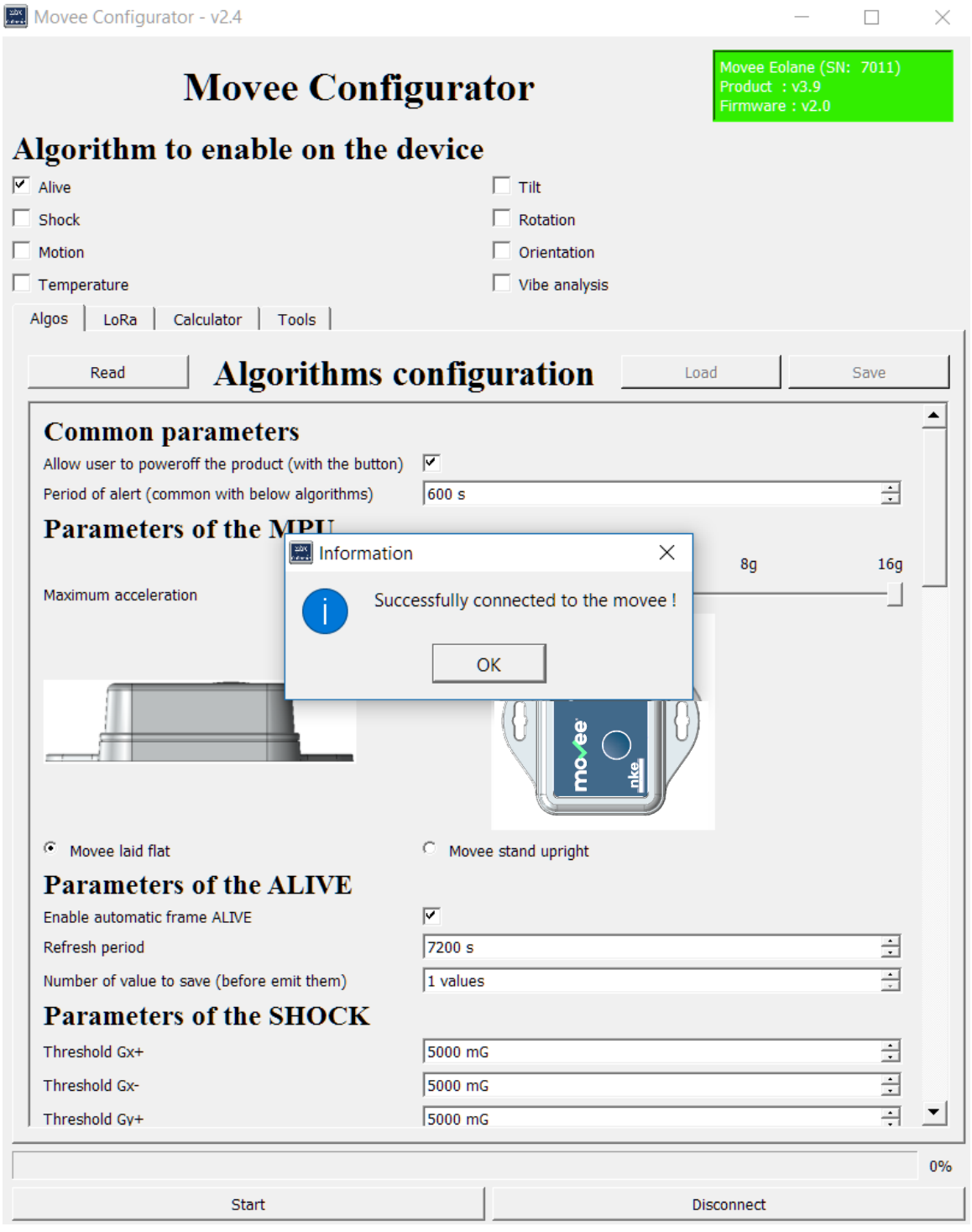
To connect your device to the interface (after following the steps described in *§3.6.2 Setup*), you have to click on “Connection”

If the device has not been setup properly, you will get the following message:

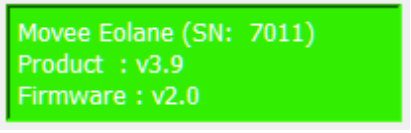


In this case, make sure you followed the steps described in *§3.6.2 Setup*

Else, your device will be connected, and you will get the following message:



At the top of the user interface, you will get the details on your device version:



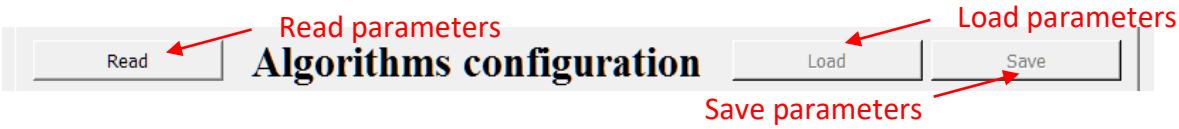
You can then click on “OK” and start configuring your device.

3.6.3.2. Algorithm selection

The upper section of the UI allows the selection of the algorithms. **Please refer to *§3.3 Algorithms compatibility* for algorithm mutual compatibility**.

3.6.3.3. Algorithms configuration

It is now possible to read the parameters set in a device, not yet to save a configuration or load a saved configuration, which is planned the next version of the Movee Configurator tool

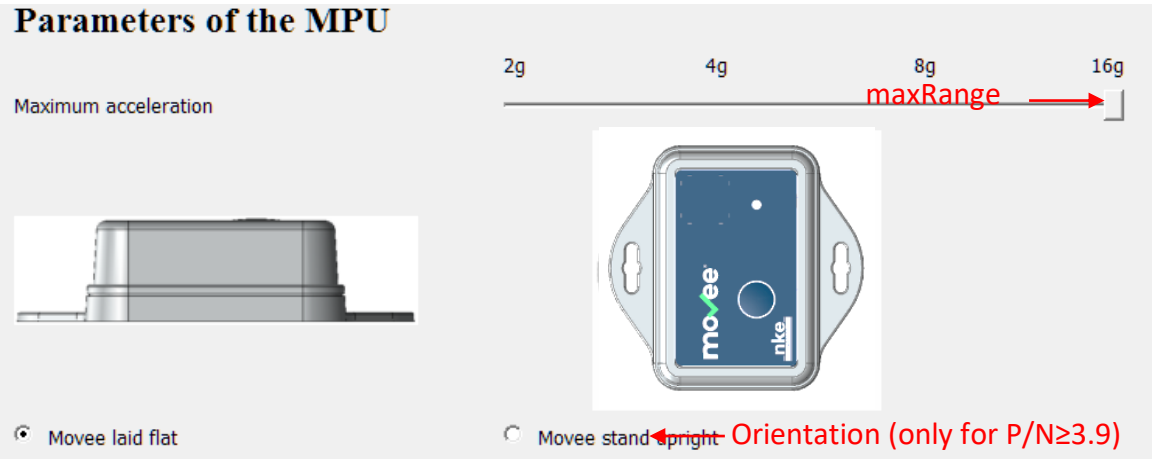


3.6.3.4. Algos tab

The Algos tab allows the setting of the selected algorithms parameters

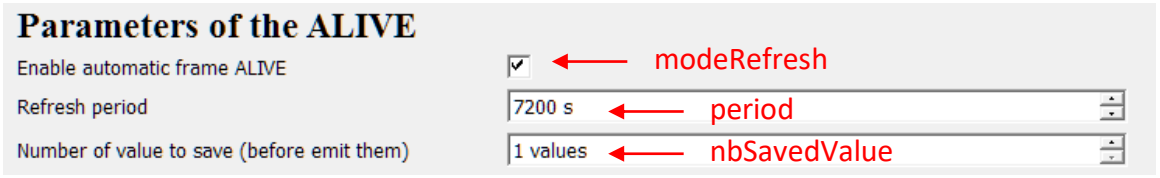
3.6.3.4.1. Common parameters (only for PN≥3.9)

3.6.3.4.2. MPU parameters

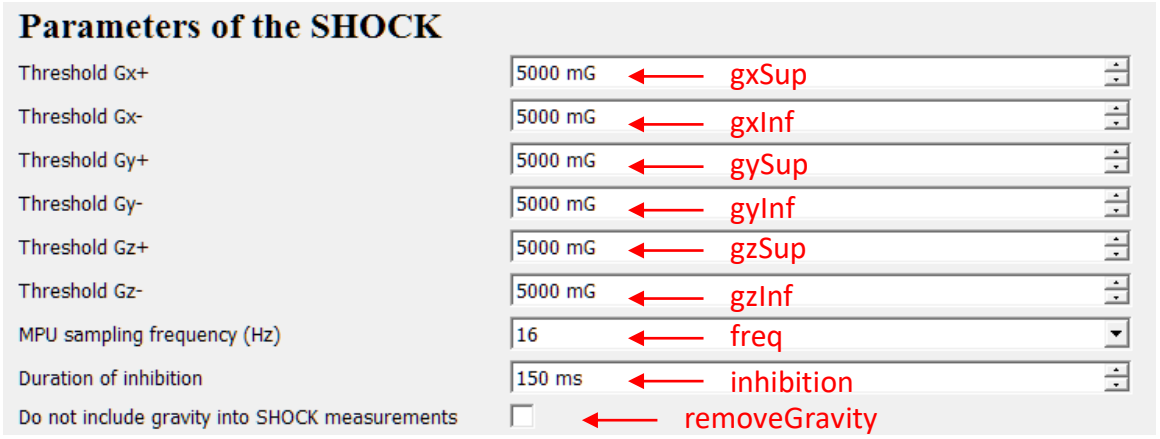


"Orientation" parameter is used by the SHOCK/TILT/MOTION/ORIENTATION/VIBE algorithms (rotation of the coordinate system), it will be ignored by the ALIVE/TEMPERATURE/ROTATION algorithms

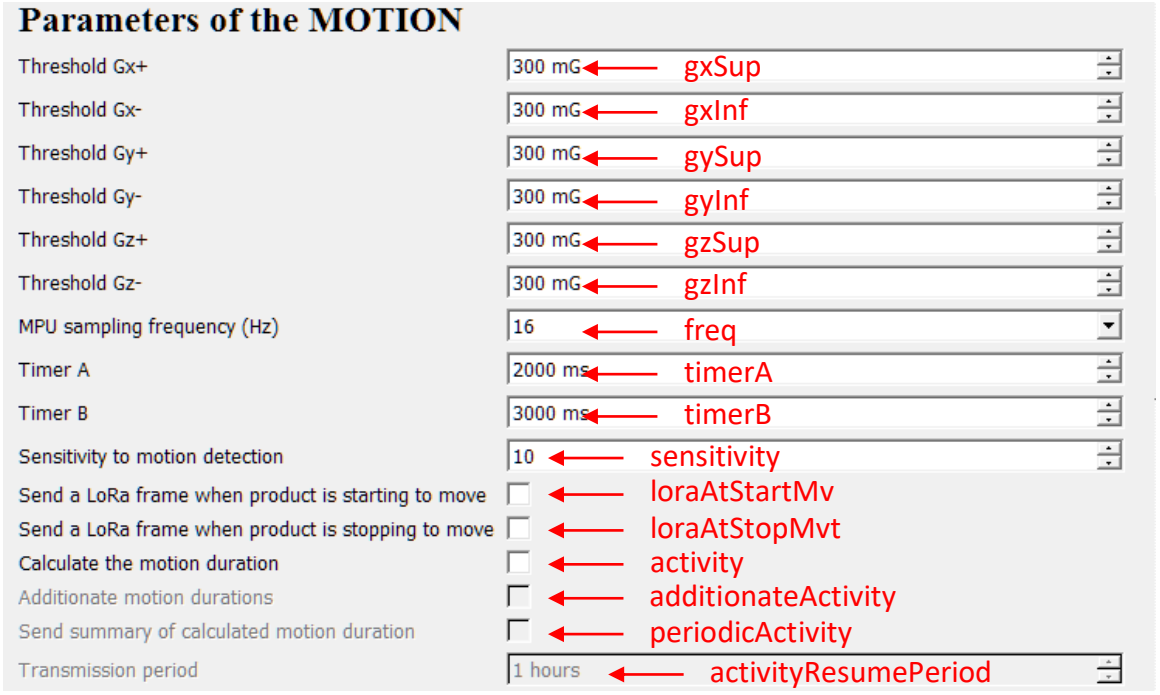
3.6.3.4.3. ALIVE parameters



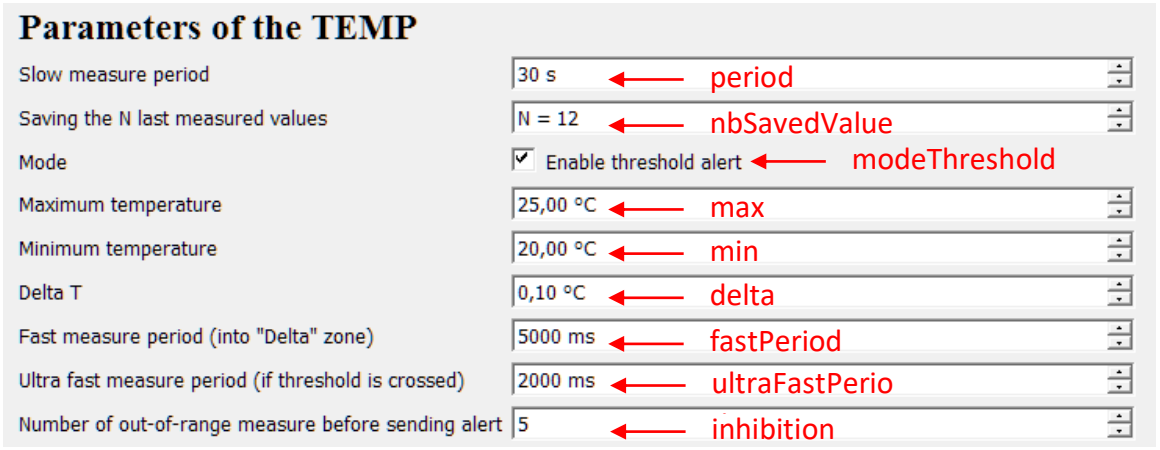
3.6.3.4.4. SHOCK parameters



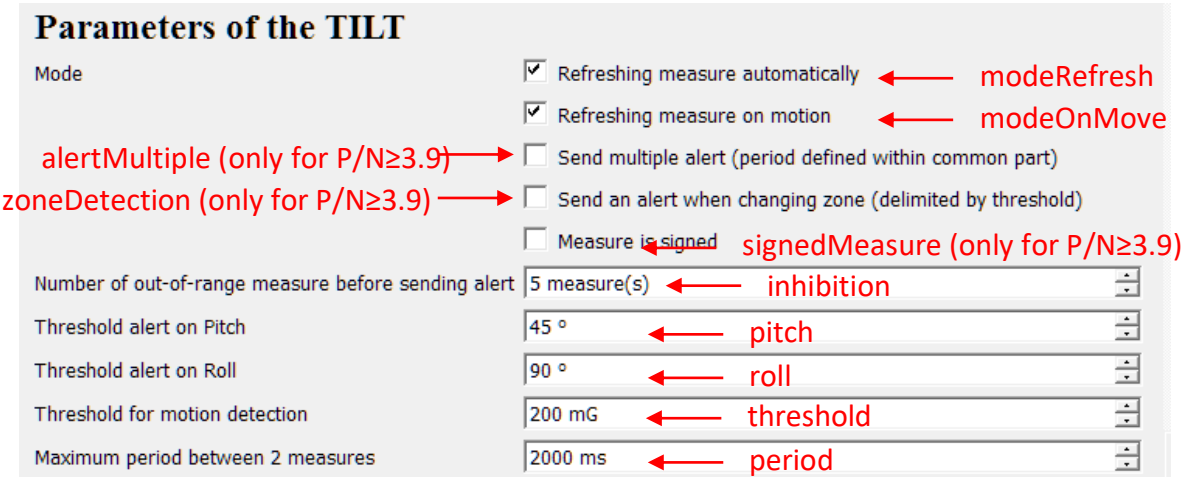
3.6.3.4.5. MOTION parameters



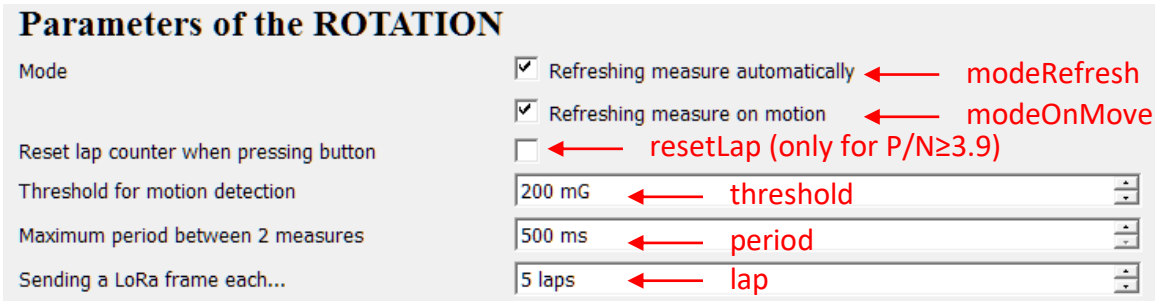
3.6.3.4.6. TEMPERATURE parameters



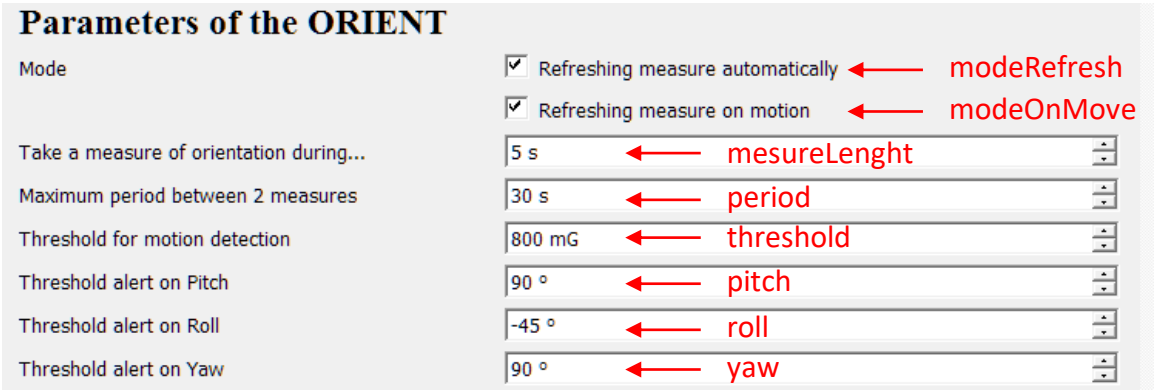
3.6.3.4.7. TILT parameters



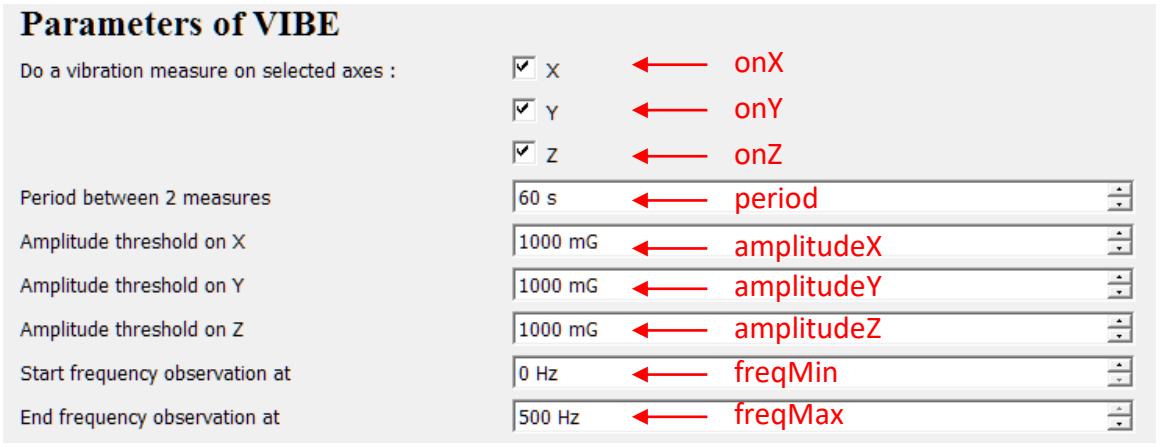
3.6.3.4.8. ROTATION parameters



3.6.3.4.9. ORIENTATION parameters

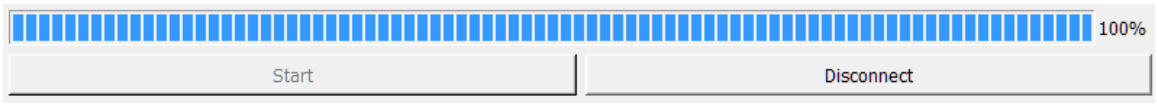


3.6.3.4.10. VIBRATION parameters



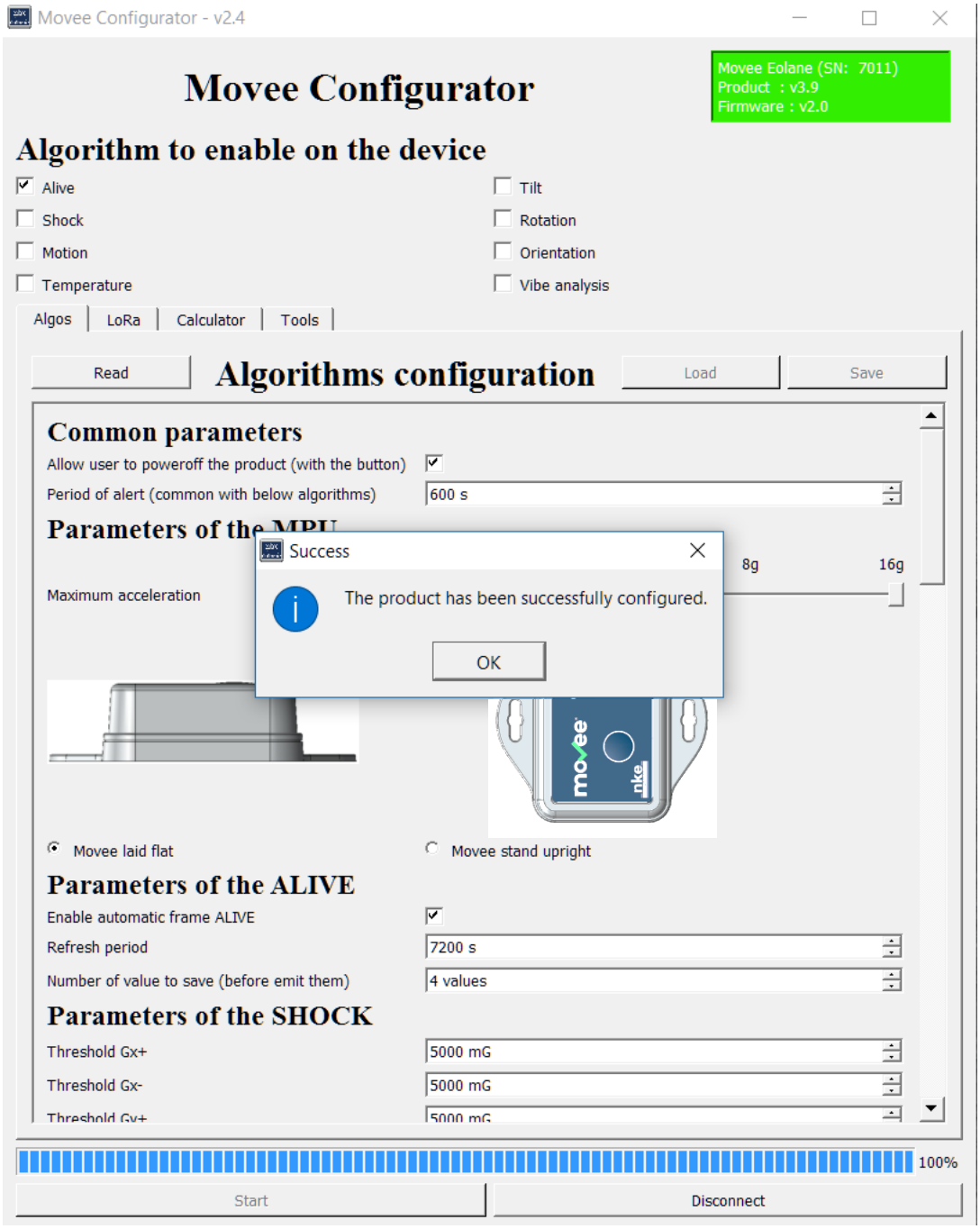
3.6.3.4.11. Status bar

After programming the device:



3.6.3.4.12. Configuration

Once you are satisfied with your configuration, click on “Start”, your device will get its configuration set with the new parameters:

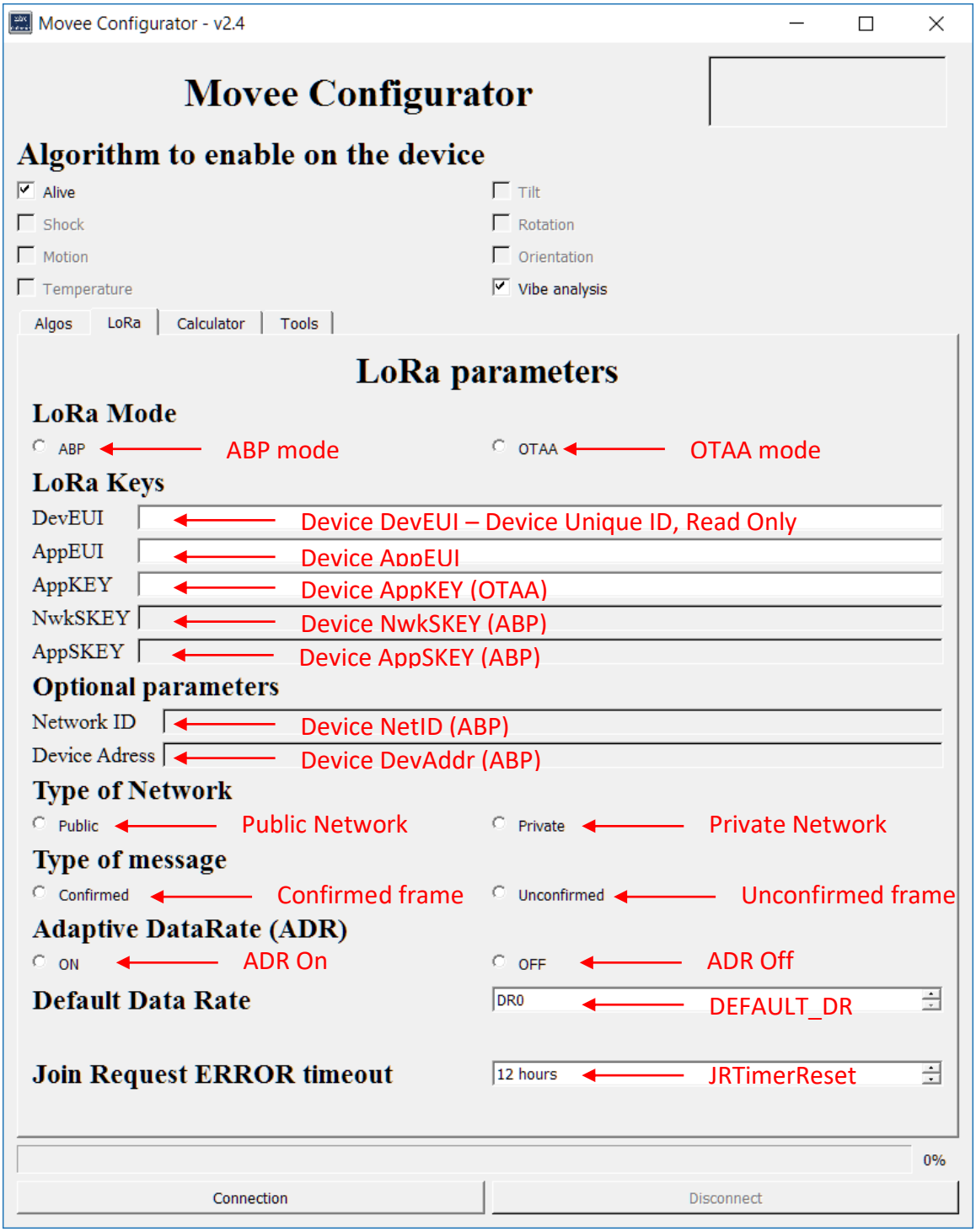


You can then click on “OK”, then “Disconnect” to disconnect your device.

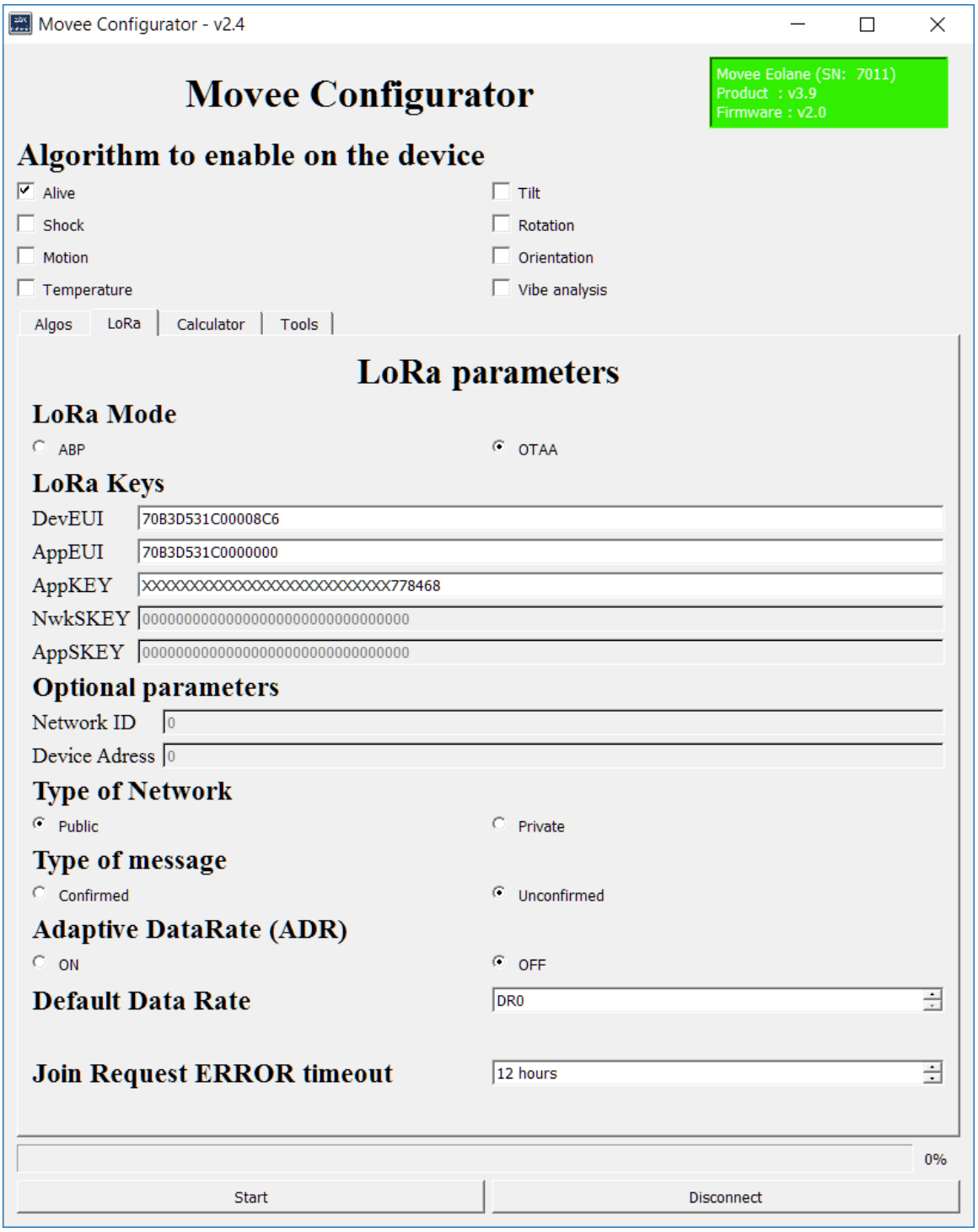
When clicking on “Disconnect”, your device will start the Join procedure on the LoRa network if set in OTAA mode (or start sending data if set in ABP mode), you can then remove the USB cable.

And close the device

3.6.3.5. LoRa settings tab

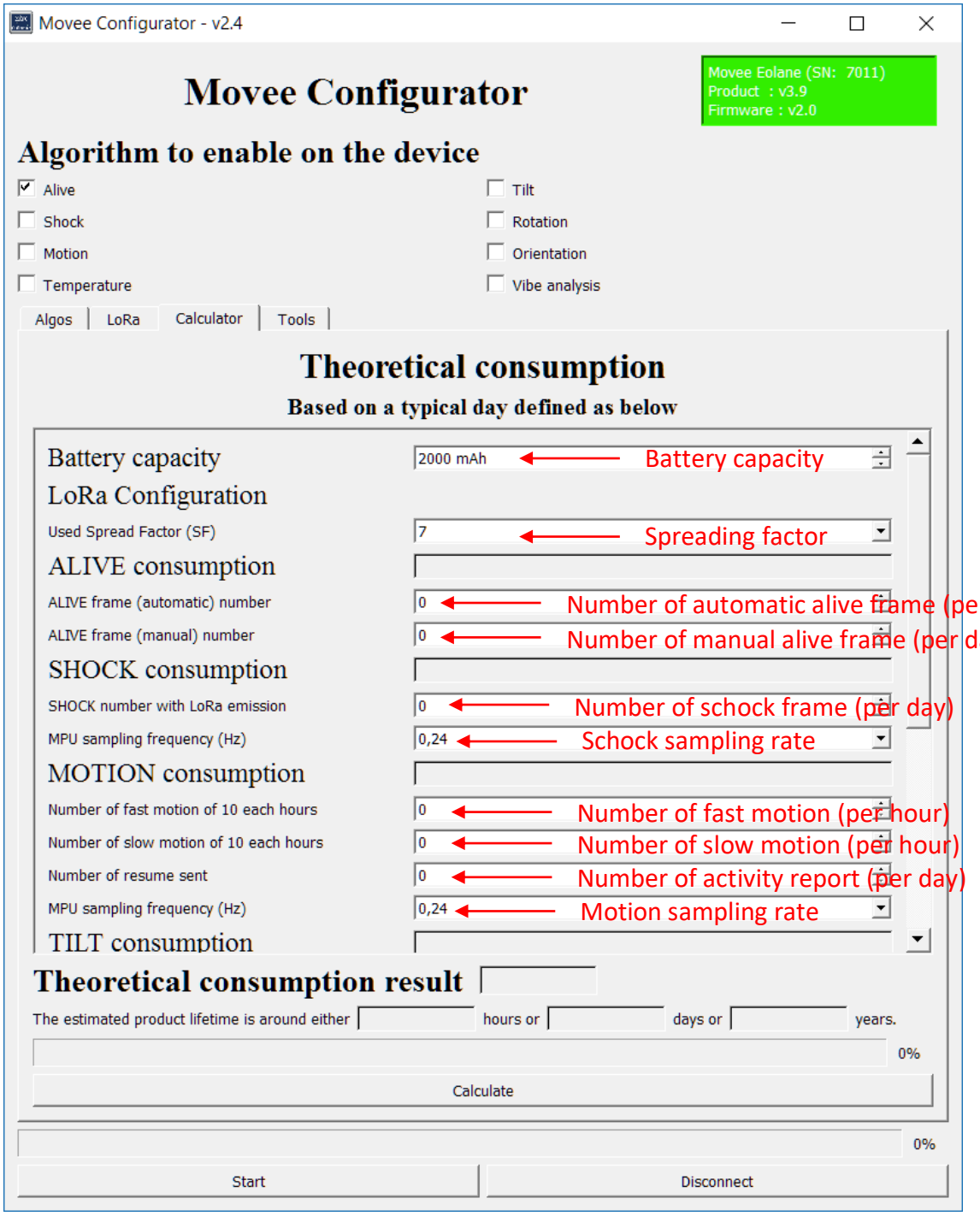


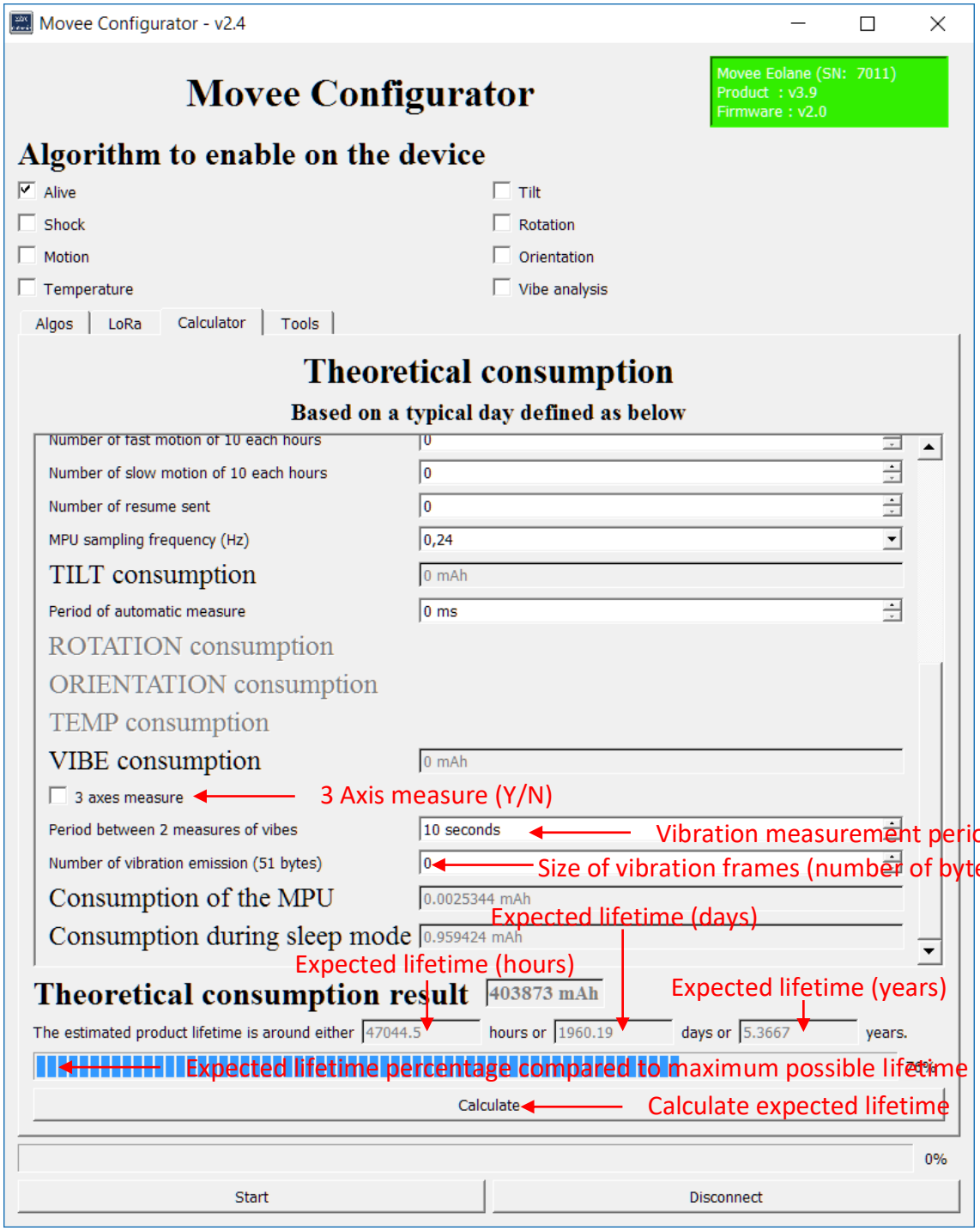
LoRa settings tab example:



The AppKEY insures the security level of your device, it is unique per device, and will be sent to you when you order a product, if not, please contact nke Watteco.

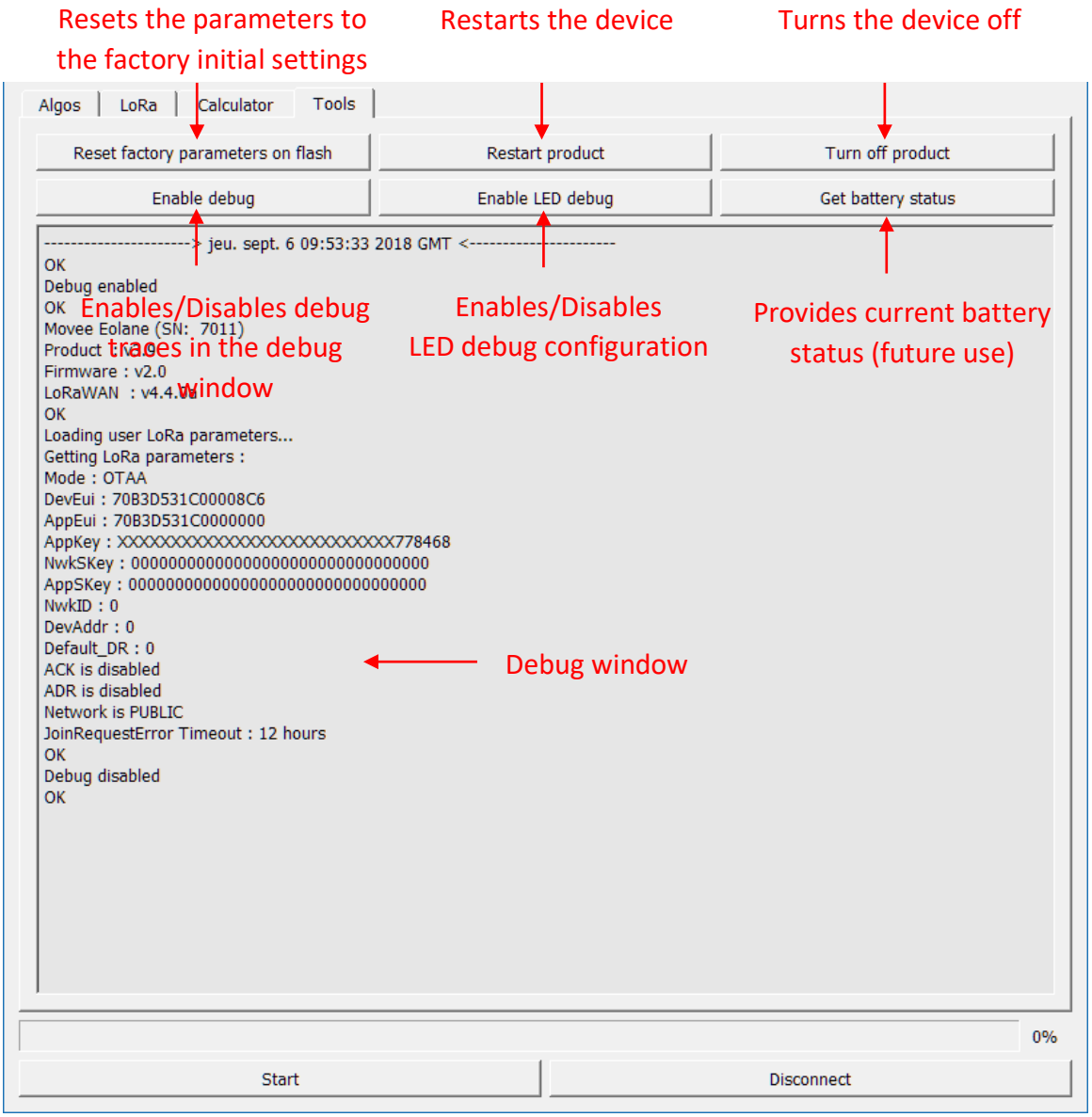
3.6.3.6. LoRa calculator tab




Movee Configurator - v2.4
Movee Configurator
Movee Eolane (SN: 7011)
Product : v3.9
Firmware : v2.0
Algorithm to enable on the device
Alive
Shock
Motion
Temperature
Tilt
Rotation
Orientation
Vibe analysis
Algos
LoRa
Calculator
Tools
Theoretical consumption
Based on a typical day defined as below
Number of fast motion of 10 each hours
Number of slow motion of 10 each hours
Number of resume sent
MPU sampling frequency (Hz)
TILT consumption
Period of automatic measure
ROTATION consumption
ORIENTATION consumption
TEMP consumption
VIBE consumption
3 axes measure
Period between 2 measures of vibes
Number of vibration emission (51 bytes)
Consumption of the MPU
Consumption during sleep mode
0,24
0 mAh
0 ms
10 seconds
0.0025344 mAh
0.959424 mAh
3 Axis measure (Y/N)
Vibration measurement period
Size of vibration frames (number of bytes)
Expected lifetime (days)
Expected lifetime (hours)
Expected lifetime (years)
Theoretical consumption result
403873 mAh
The estimated product lifetime is around either
47044.5
hours or
1960.19
days or
5.3667
years.
Expected lifetime percentage compared to maximum possible lifetime
Calculate
Calculate expected lifetime
0%
Start
Disconnect

3.6.3.7. Tools tab

The tools tab provides debug tools:



IV. Appendix

4.1. Revision history

Revision	Modifications
0.50	First edition
0.51	Typo corrections
0.9	Add Vibration algorithm details, LoRa frame examples, SW upgrade to v1.7 (downlink integration)
1.00	Add Tilt, Rotation and Orientation algorithm details
1.01	Correction on TILT, ROTATION and ORIENTATION frame examples (incorrect number of bytes in frame examples) Update operating temperature, and add storage temperature information
1.02	Correction on TILT frame examples (wrong scale for pitch and roll values)
1.03	Add Downlink commands
1.04	Correction on Shock data type (unsigned to signed) in §3.2.5
1.05	Add Downlink examples chapter Update Movee configurator user interface chapter for 1.7.e version
1.06	OTAA BackOff cycle revised with JRTimerReset parameter Add JRTimerReset parameter definition and DL parameter Update VERSION frame definition (formerly INFORMATION frame) Add OTAA first frame (VERSION) Update Movee configurator user interface chapter for 1.8 version
1.07	Update Payload data table in § 3.4.2
1.08	ABP mode update Typo correction (bookmarks)
1.09	X Axis correction for TILT algorithm Add information on casing holes center distance
2.00	Full update for FW 2.0 / P/N 3.9 compliance, and Movee configurator 2.4
2.01	2D and mechanicals information updated Add recommended fastening solutions chapter Operating and storage temperature update ALGO poweroff default value updated with correct value

4.2. P/N and Firmware history

P/N	FW version	LoRaWAN stack version	User Guide version
1.0	1.5	4.3.0	0.50 0.51
1.1	1.6	4.3.0	0.52
1.2	1.7	4.3.0	0.90 1.00 1.01 1.02
1.5	1.7.1_128k	4.3.1	1.03
2.4	1.7.1 1.7.1a	4.3.1	1.03 1.04
2.5	1.8	4.3.1	1.04
3.6	1.8.5	4.3.1	1.05
3.7	1.8.8	4.3.1	1.06 1.07 1.08
3.9	2.0	4.4.0	2.00 2.01

4.3. EFM32 Reset cause table

9.5.2 RMU_RSTCAUSE - Reset Cause Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access																	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name																	BUMODERST	BUBODREG	BUBODUNREG	BUBODBUVIN	BUBODVDDDREG	BODAVDD1	BODAVDD0	EM4WURST	EM4RST	SYSREQRST	LOCKUPRST	WDOGRST	EXTRST	BODREGRST	BODUNREGRST	PORST

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
15	BUMODERST	0	R	**Backup mode reset**
	Set if the system has been in Backup mode. Must be cleared by software. Please see Section 10.3.4 (p. 112) for details on how to interpret this bit.			
14	BUBODREG	0	R	**Backup Brown Out Detector Regulated Domain**
	Set if the Backup BOD sensing on regulated power triggers. Must be cleared by software. Please see Section 10.3.4.2 (p. 113) for details on how to interpret this bit.			
13	BUBODUNREG	0	R	**Backup Brown Out Detector Unregulated Domain**

Bit	Name	Reset	Access	Description
	Set if the Backup BOD sensing on unregulated power triggers. Must be cleared by software. Please see Section 10.3.4.2 (p. 113) for details on how to interpret this bit.			
12	BUBODBUVIN	0	R	**Backup Brown Out Detector, BU_VIN**
	Set if the Backup BOD sensing on BU_VIN triggers. Must be cleared by software. Please see Section 10.3.4.2 (p. 113) for details on how to interpret this bit.			
11	BUBODVDDDREG	0	R	**Backup Brown Out Detector, VDD_DREG**
	Set if the Backup BOD sensing on VDDD_REG triggers. Must be cleared by software. Please see Section 10.3.4.2 (p. 113) for details on how to interpret this bit.			
10	BODAVDD1	0	R	**AVDD1 Bod Reset**
	Set if analog power domain 1 brown out detector reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
9	BODAVDD0	0	R	**AVDD0 Bod Reset**
	Set if analog power domain 0 brown out detector reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
8	EM4WURST	0	R	**EM4 Wake-up Reset**
	Set if the system has been woken up from EM4 from a reset request from pin. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
7	EM4RST	0	R	**EM4 Reset**
	Set if the system has been in EM4. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
6	SYSREQRST	0	R	**System Request Reset**
	Set if a system request reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
5	LOCKUPRST	0	R	**LOCKUP Reset**
	Set if a LOCKUP reset has been requested. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
4	WDOGRST	0	R	**Watchdog Reset**
	Set if a watchdog reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
3	EXTRST	0	R	**External Pin Reset**
	Set if an external pin reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
2	BODREGRST	0	R	**Brown Out Detector Regulated Domain Reset**
	Set if a regulated domain brown out detector reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
1	BODUNREGRST	0	R	**Brown Out Detector Unregulated Domain Reset**
	Set if a unregulated domain brown out detector reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			
0	PORST	0	R	**Power On Reset**
	Set if a power on reset has been performed. Must be cleared by software. Please see Table 9.1 (p. 100) for details on how to interpret this bit.			

Register Value	Cause
0bXXXX XXXX XXXX XXX1	A Power-on Reset has been performed. X bits are don't care.
0bXXXX XXXX 0XXX XX10	A Brown-out has been detected on the unregulated power.
0bXXXX XXXX XXX0 0100	A Brown-out has been detected on the regulated power.
0bXXXX XXXX XXXX 1X00	An external reset has been applied.
0bXXXX XXXX XXX1 XX00	A watchdog reset has occurred.
0bXXXX X000 0010 0000	A lockup reset has occurred.
0bXXXX X000 01X0 0000	A system request reset has occurred.
0bXXXX X000 1XX0 0XX0	The system has woken up from EM4.
0bXXXX X001 1XX0 0XX0	The system has woken up from EM4 on an EM4 wakeup reset request from pin.
0bXXXX X01X XXX0 0000	A Brown-out has been detected on Analog Power Domain 0 (AVDD0).
0bXXXX X10X XXX0 0000	A Brown-out has been detected on Analog Power Domain 1 (AVDD1).
0bXXXX 1XXX XXXX 0XX0	A Brown-out has been detected by the Backup BOD on VDD_DREG.
0bXXX1 XXXX XXXX 0XX0	A Brown-out has been detected by the Backup BOD on BU_VIN.
0bXX1X XXXX XXXX 0XX0	A Brown-out has been detected by the Backup BOD on unregulated power
0bX1XX XXXX XXXX 0XX0	A Brown-out has been detected by the Backup BOD on regulated power.
0b1XXX XXXX XXXX XXX0	The system has been in Backup mode.