Wireless Accelerometer
and
Surface Temperature Sensor

R718EC
User Manual

Table of Contents

1. Introduction 2

2. Appearance 3

3. Features 3

4. Set up Instruction 4

5. Data Report 5

5.1 ActiveThreshold and InActiveThreshold 6

5.2 Calibration 6

5.3 The X,Y,Z axis Direction of R718EC 7

5.4 Examples of ReportDataCmd 7

5.5 Examples of Data Configuration 10

5.6 Examples of Set/GetSensorAlarmThresholdCmd 12

5.7 Example of GlobalCalibrateCmd 13

5.8 Example for MinTime/MaxTime logic 14

6. Applications 15

7. Installation 16

8. Information about Battery Passivation 18

8.1 To determine whether a battery requires activation 18

8.2 How to activate the battery 18

9. Important Maintenance Instruction 19

1. Introduction

The R718EC is a LoRaWAN ClassA device with three-axis acceleration, temperature and is compatible with the LoRaWAN protocol. When the device moves or vibrates over the threshold value, it immediately reports the temperature, acceleration, and velocity of the X, Y, and Z axes.

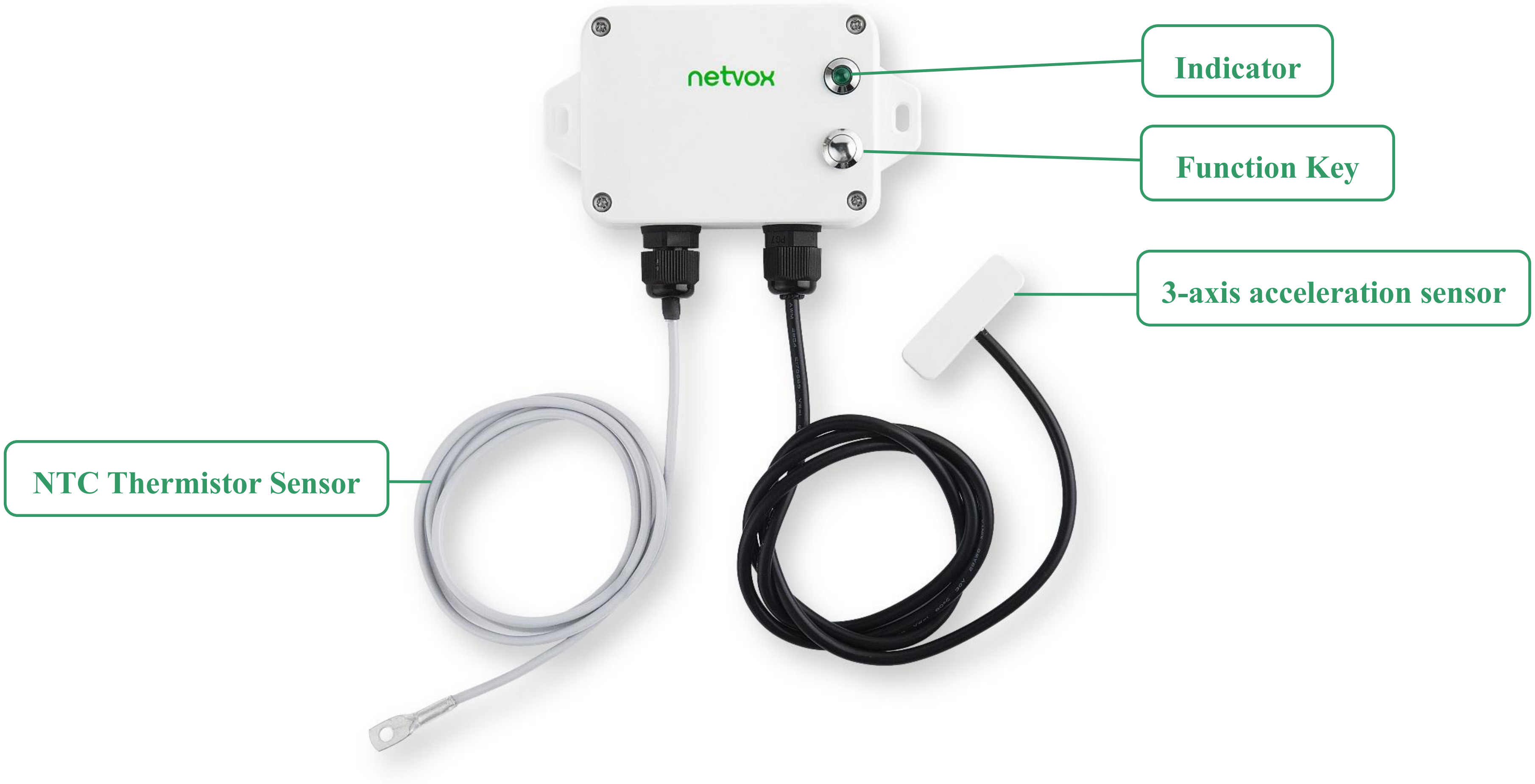
LoRa Wireless Technology:

LoRa is a wireless communication technology dedicated to long-distance and low power consumption. Compared with other communication methods, the LoRa spread spectrum modulation method greatly increases to expand the communication distance. Widely used in long-distance, low-data wireless communications. For example, automatic meter reading, building automation equipment, wireless security systems, and industrial monitoring. Main features include small size, low power consumption, transmission distance, anti-interference ability, and so on.

LoRaWAN:

LoRaWAN uses LoRa technology to define end-to-end standard specifications to ensure interoperability between devices and gateways from different manufacturers.

2. Appearance



3. Features

- SX1276 wireless communication module.
- 2 sections ER14505 3.6V Lithium AA size battery
- Detect the acceleration and velocity of the X, Y, and Z axes.
- Magnet base
- Protection level IP65/IP67 (optional)
- Compatible with LoRaWAN^{TM} Class A
- Frequency hopping spread spectrum technology
- Available third-party platforms: Actility / ThingPark, TTN, MyDevices/Cayenne
- Low power consumption and long battery life

Note: Please visit http://www.netvox.com.tw/electric/electric_calc.html for detailed information of battery life calculation.

1. Actual range may vary depending on the environment.
2. Battery life is determined by sensor reporting frequency and other variables.

4. Set up Instruction

On/Off

Power on	Insert batteries. (Users may need a screwdriver to open the battery cover.)
Turn on	Press and hold the function key for 3 seconds until the green indicator flashes once.
Turn off (Reset to factory setting)	Press and hold the function key for 5 seconds, and the green indicator flashes 20 times.
Power off	Remove Batteries.
Note	1. Remove and insert the battery; the device is off by default when the batteries are removed and inserted. 2. On/off interval should be at least 10 seconds long to avoid the interference of capacitor inductance and other energy storage components. 3. Five seconds after powering on, the device will be in engineering test mode.

Network Joining

Never joined the network	Turn on the device to search the network. The green indicator stays on for 5 seconds: success The green indicator remains off: fail
Had joined the network	Turn on the device to search the previous network. The green indicator stays on for 5 seconds: success The green indicator remains off: fail

Function Key

Press and hold for 5 seconds	Restore to factory setting / Turn off The green indicator flashes 20 times: success The green indicator remains off: fail
Press once	The device is in the network: green indicator flashes once and sends a report The device is not in the network: the green indicator remains off

Sleeping Mode

The device is on and in the network	Sleeping period: Min Interval. When the reportchange exceeds the setting value or the state changes: send a data report according to Min Interval.

Low Voltage Warning

Low Voltage	3.2V

5. Data Report

The device would send a version packet after powering on.

After detection, the device will send 3 uplink packets: (1) 3-axis acceleration, (2) 3-axis velocity and temperature, and (3) 3-axis angle and threshold alarm. If the threshold of acceleration, velocity, or temperature is configured, the 4th uplink packet will be reported as the data exceeds the threshold*[1]. The interval between each data packet should be 10 seconds.

The device sends data in the default configuration before any configuration is done.

Default setting:

MaxTime: 0x0E10 (3600s)

MinTime: 0x0E10 (3600s)

SensorAlarmThreshold : Disable

BatteryChange: 0x01 (0.1v)

AccelerationChange: 0x0003

AngleChange: 0x07D0 (10°, 0.005°)

ActiveThreshold = 0x0003

InActiveThreshold = 0x0002

Three-axis acceleration and velocity:

The device would calibrate*[2] after powering on. During the one-minute calibration, the data sent could be neglected. In addition to periodic reports, the device will be Active*[3] and report acceleration, velocity, and angle of 3 axes as any detected data exceeds the threshold. When the device is in the "Active" status, the detection stops. Only when the 3-axis acceleration is lower than the thresholds for 5 seconds (cannot be configured), the device will be in the "Inactive" status and start the next detection.

*[1] Please refer to 5.6 Examples of Set/GetSensorAlarmThresholdCmd.
*[2] Please refer to 5.2 Calibration.
*[3] To know whether the device is active or not, user may refer to the last byte of the data packet of reporttype 0x01.

Note:

(1) The device report interval will be programmed based on the default firmware which may vary.

(2) The interval between two reports must be the minimum time.

Please refer Netvox LoRaWAN Application Command document and Netvox Lora Command Resolver http://cmddoc.netvoxcloud.com/cmddoc to resolve uplink data.

Data report configuration and sending period are as following:

Min Interval (Unit:second)	Max Interval (Unit:second)	Reportable Change	Current Change≥ Reportable Change	Current Change＜ Reportable Change
Any number between 1~65535	Any number between 1~65535	Can not be 0.	Report per Min Interval	Report per Max Interval

5.1 ActiveThreshold and InActiveThreshold

Formula	**Active Threshold (or InActiveThreshold) = Critical value ÷ 9.8 ÷ 0.0625** * The gravitational acceleration at standard atmospheric pressure is 9.8 m/s^2 * The scale factor of the threshold is 62.5 mg
Active Threshold	Active Threshold can be changed by ConfigureCmd Active Threshold range is **0x0003-0x00FF** (default is 0x0003);
InActive Threshold	InActive Threshold can be changed by ConfigureCmd InActive Threshold range is **0x0002-0x00FF** (default is 0x0002)
Example	If the critical value is $10m/s^2$, the Active Threshold (or InActive Threshold) would be 10/9.8/0.0625=16.32 Active Threshold (or InActiveThreshold) would be 16 (round 16.32 to the nearest whole number) Note: When configuration, the Active Threshold should be greater than the InActive Threshold.

5.2 Calibration

The accelerometer is a mechanical structure that contains components that can move freely. These moving parts are very sensitive to mechanical stress, far beyond solid-state electronics. The 0g offset is an important accelerometer indicator because it defines the baseline used to measure acceleration.

Users would need to install and turn on R718EC first. 1 minute after joining the network, the R718EC would automatically deviate from the calibration. The data reported before deviation calibration could be neglected.

If users would like to adjust the device's position, it shall be disconnected from the power supply for 1 minute and turn it on to automatically deviate from the calibration.

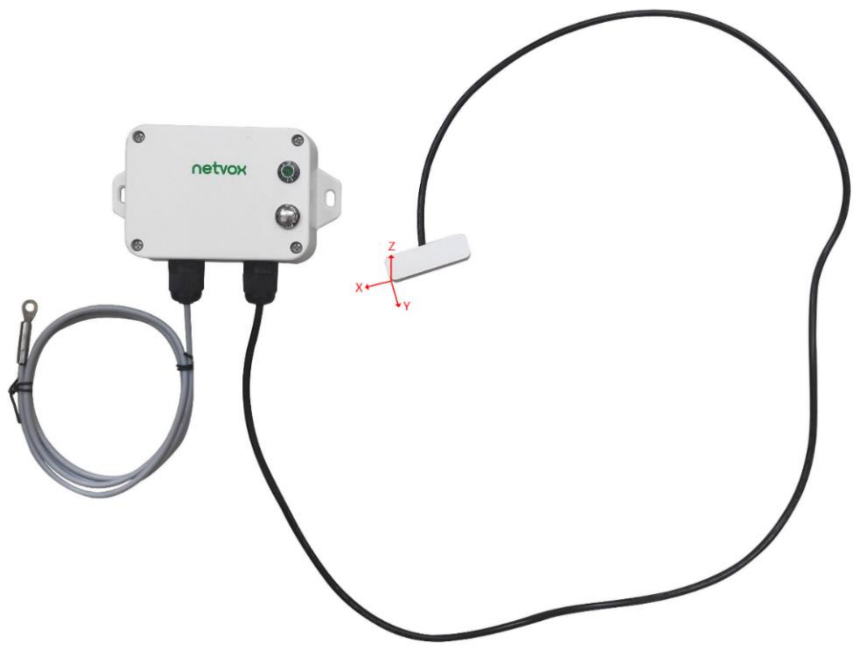
After deviating from calibration, the reported three-axis acceleration value will be within $1m/s^2$, which means the device remains still. (If the value exceeds $1m/s^2$, users need to repeat the above instructions until the value is within $1m/s^2$.)

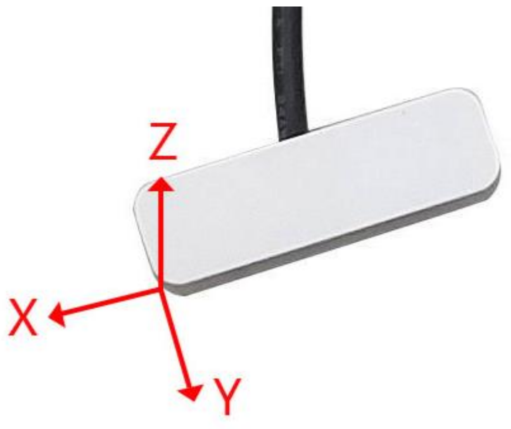
To have the correct reported value, the position of the sensor shall be fixed after deviating from calibration.

Note: (1) The sensor should be placed horizontally and still during auto-calibration.

(2) After calibration, the angles between the X, Y, and Z axes and the horizontal surface should be 0°, 0°, and 90°.

5.3 The X,Y,Z axis Direction of R718EC





5.4 Examples of ReportDataCmd

FPort：0x06

Bytes	1	1	1	Var(Fix=8 Bytes)
	Version	DeviceType	ReportType	NetvoxPayLoadData

Version– 1 byte –0x01——the Version of NetvoxLoRaWAN Application Command Version

DeviceType– 1 byte – Device Type of Device

The devicetype is listed in Netvox LoRaWAN Application Devicetype doc

ReportType – 1 byte –the presentation of the NetvoxPayLoadData，according the devicetype

NetvoxPayLoadData– Fixed bytes (Fixed =8bytes)

Tips

1. **Battery Voltage:**

   The voltage value is bit 0 ~ bit 6, bit 7=0 is normal voltage, and bit 7=1 is low voltage.

   Battery=0xA0, binary=1010 0000, if bit 7= 1, it means low voltage.

   The actual voltage is 0010 0000 = 0x20 = 32, 32*0.1v =3.2v

2. **Version Packet:**

   When Report Type=0x00 is the version packet, such as 011C00140220230823 0000, the firmware version is 2023.08.23.

3. **Data Packet:**

   a. When Report Type=0x01 is data packet. If the device data exceeds 11 bytes or there are shared data packets, the Report Type will have different values.

   b. R718EC value uses big-endian computing.

   c. Because of the length limitation of R718EC instruction. Therefore, R718E sends out 2 bytes and adds 0 to the data to form 4 bytes of float32.

Device	Device Type	Report Type	NetvoxPayLoadData				
R718EC	0x1C	0x00	SoftwareVersion (1Byte) Eg.0x0A—V1.0	HardwareVersion (1Byte)	DateCode (4Bytes,eg0x20170503)	Reserved (2Bytes,fixed 0x00)	
		0x01	Battery (1Byte, unit:0.1V)	AccelerationX (Float16_2Bytes, m/s^2)	AccelerationY (Float16_2Bytes, m/s^2)	AccelerationZ (Float16_2Bytes, m/s^2)	Status (1Byte, 0x01_Active 0x00_Inactive)
		0x02	VelocityX (Float16_2Bytes, mm/s)	VelocityY (Float16_2Bytes, mm/s)	VelocityZ (Float16_2Bytes, mm/s)	Temperature (Signed 2Bytes, unit:0.1°C)	
		0x03	Battery (1Byte, unit:0.1V)	AngleX (2Byte, Signed Value , Unit:0.005°)	AngleY (2Byte, Signed Value , Unit:0.005°)	AngleZ(2Byte, Signed Value , Unit:0.005°)	ThresholdAlarm (1Byte, Bit0_LowAngleXAlarm, Bit1_HighAngleXAlarm, Bit2_LowAngleYAlarm, Bit3_HighAngleYAlarm, Bit4_LowAngleZAlarm, Bit5_HighAngleAlarm, Bit6-7:Reserved)
		0x04	Battery (1Byte, unit:0.1V)	ThresholdAlarmAcceleration (1Byte, Bit0_LowAccelerationXAlarm, Bit1_HighAccelerationXAlarm, Bit2_LowAccelerationYAlarm, Bit3_HighAccelerationYAlarm, Bit4_LowAccelerationZAlarm, Bit5_HighAccelerationZAlarm, Bit6-7:Reserved)	ThresholdAlarmVelocity (1Byte, Bit0_LowVelocityXAlarm, Bit1_HighVelocityXAlarm, Bit2_LowVelocityYAlarm, Bit3_HighVelocityYAlarm, Bit4_LowVelocityZAlarm, Bit5_HighVelocityZAlarm, Bit6-7:Reserved)	ThresholdAlarmTemperature (1Byte, Bit0_LowTemperatureAlarm, Bit1_HighTemperatureAlar, Bit2-7:Reserved)	Reserved (5Bytes, fixed 0x00)

Example of uplink:

# Packet 1: 011C012488408640883F00

1st byte (01): Version

2nd byte (1C): DeviceType 0x1C — R718EC

3rd byte (01): ReportType

4th byte (24): Battery—3.6v , 24 Hex=36 Dec 36*0.1v=3.6v

5th 6th byte (8840): Acceleration X, float32(40880000) = 4.25 m/s^2

7th 8th byte (8640): Acceleration Y, float32(40860000) = 4.187 m/s^2

9th 10th byte (883F): Acceleration Z, float32(3F880000) = 1.0625 m/s^2

11th byte (00): Status — Inactive

# Packet 2: 011C02A43FA13FA43E011C

1st byte (01): Version

2nd byte (1C): DeviceType 0x1C — R718EC

3rd byte (02): ReportType

4th 5th byte (A43F): VelocityX, float32(3FA40000) = 1.28125 mm/s

6th 7th byte (A13F): VelocityY, float32(3FA10000) = 1.2578125 mm/s

8th 9th byte (A43E): VelocityZ, float32(3EA40000) = 0.3203125 mm/s

10th 11th byte (011C): Temperature—28.4°C, 011C(HEX)=284(DEC),284*0.1°C =28.4°C

# Packet 3: 011C0324F154F10630F500

1st byte (01): Version

2nd byte (1C): DeviceType 0x1C — R718EC

3rd byte (03): ReportType

4th byte (24): Battery—3.6v , 24 Hex=36 Dec 36*0.1v=3.6v

5th 6th byte (F154): AngleX, -18.78°, F154 (Hex) = -3756 (Dec), -3756* 0.005° = -18.78°

7th 8th byte (F106): AngleY, -19.17°, F106 (Hex) = -3834 (Dec), -3834* 0.005° = -19.17°

9th 10th byte (30F5): AngleZ, 62.665°, 30F5 (Hex) = 12533 (Dec), 12533* 0.005° = 62.665°

11th byte (00): ThresholdAlarm, 0x00 = 0000 0000 (Bin) — No alarm

# Packet 4: 011C042401010100000000

1st byte (01): Version

2nd byte (1C): DeviceType 0x1C — R718EC

3rd byte (04): ReportType

4th byte (24): Battery—3.6v , 24 Hex=36 Dec 36*0.1v=3.6v

5th byte (01): ThresholdAlarmAcceleration — 0x01 = 0000 0001 (Bin) //Bit 0 = 1 (alarm)

Bit0: LowAccelerationXAlarm

6th byte (01): ThresholdAlarmVelocity — 0x01 = 0000 0001 (Bin) //Bit 0 = 1 (alarm)

Bit0: LowVelocityXAlarm

7th byte (01): ThresholdAlarmTemperature — 0x01 = 0000 0001 (Bin) //Bit 0 = 1 (alarm)

Bit0: LowTemperatureAlarm

8th 11th byte (00000000): Reserved

5.5 Examples of Data Configuration

FPort：0x07

Bytes	1	1	Var(Fix =9 Bytes)
	CmdID	DeviceType	NetvoxPayLoadData

CmdID– 1 byte

DeviceType– 1 byte – Device Type of Device

NetvoxPayLoadData– var bytes (Max=9bytes)

<table>
<tr><th>Description</th><th>Device</th><th>Cmd ID</th><th>Device Type</th><th colspan="60">NetvoxPayLoadData</th></tr>
<tr><td>Config ReportReq</td><td rowspan="12">R718EC</td><td>0x01</td><td rowspan="12">0x1C</td><td colspan="12">MinTime (2bytes Unit:s)</td><td colspan="12">MaxTime (2bytes Unit:s)</td><td colspan="12">BatteryChange (1byte Unit:0.1v)</td><td colspan="12">AccelerationChange (2byte Unit:m/s^2)</td><td colspan="12">AngleChange (2byte Unit:0.005°)</td></tr>
<tr><td>Config ReportRsp</td><td>0x81</td><td colspan="30">Status (0x00_success)</td><td colspan="30">Reserved (8Bytes,Fixed 0x00)</td></tr>
<tr><td>ReadConfig ReportReq</td><td>0x02</td><td colspan="60">Reserved (9Bytes,Fixed 0x00)</td></tr>
<tr><td>ReadConfig ReportRsp</td><td>0x82</td><td colspan="12">MinTime (2bytes Unit:s)</td><td colspan="12">MaxTime (2bytes Unit:s)</td><td colspan="12">BatteryChange (1byte Unit:0.1v)</td><td colspan="12">AccelerationChange (2byte Unit:m/s^2)</td><td colspan="12">AngleChange (2byte Unit:0.005°)</td></tr>
<tr><td>SetActive ThresholdReq</td><td>0x03</td><td colspan="20">ActiveThreshold (2Bytes)</td><td colspan="20">InActiveThreshold (2Bytes)</td><td colspan="20">Reserved (5Bytes,Fixed 0x00)</td></tr>
<tr><td>SetActive ThresholdRsp</td><td>0x83</td><td colspan="30">Status (0x00_success)</td><td colspan="30">Reserved (8Bytes,Fixed 0x00)</td></tr>
<tr><td>GetActive ThresholdReq</td><td>0x04</td><td colspan="60">Reserved (9Bytes,Fixed 0x00)</td></tr>
<tr><td>GetActive ThresholdRsp</td><td>0x84</td><td colspan="20">ActiveThreshold (2Bytes)</td><td colspan="20">InActiveThreshold (2Bytes)</td><td colspan="20">Reserved (5Bytes,Fixed 0x00)</td></tr>
<tr><td>SetFilterVelocity ThresholdReq (REMAIN Lastconfig when resettofac)</td><td>0x05</td><td colspan="15">FilterVelocityThresholdX (1Byte,Unit:0.1mm/s)</td><td colspan="15">FilterVelocityThresholdY (1Byte,Unit:0.1mm/s)</td><td colspan="15">FilterVelocityThresholdZ(1 Byte,Unit:0.1mm/s)</td><td colspan="15">Reserved (6Bytes,Fixed 0x00)</td></tr>
<tr><td>SetFilterVelocity ThresholdReq</td><td>0x85</td><td colspan="30">Status(0x00_success)</td><td colspan="30">Reserved (8Bytes,Fixed 0x00)</td></tr>
<tr><td>GetFilterVelocit yThresholdReq</td><td>0x06</td><td colspan="60">Reserved (9Bytes,Fixed 0x00)</td></tr>
<tr><td>GetFilterVelocit yThresholdRsp</td><td>0x86</td><td colspan="15">FilterVelocityThresholdX (1Byte,Unit:0.1mm/s)</td><td colspan="15">FilterVelocityThresholdY (1Byte,Unit:0.1mm/s)</td><td colspan="15">FilterVelocityThresholdZ (1Byte,Unit:0.1mm/s)</td><td colspan="15">Reserved (6Bytes,Fixed 0x00)</td></tr>
</table>

(1) Configure device parameters MinTime = 1min, MaxTime = 1min, BatteryChange = 0.1v, Acceleratedvelocitychange = 1m/s^2, AngleChange = 1°

Downlink: 011C003C003C01000100C8   003C (Hex) = 60(Dec)

00C8 (Hex) = 200 (Dec), 200* 0.005° = 1°

Device returns:811C000000000000000000 (configuration is successful)

811C010000000000000000 (configuration failed)

(2) Read device parameters

Downlink: 021C000000000000000000

Device returns: 821C003C003C01000100C8 (current device parameters)

(3) Assuming that the Active Threshold is set to 10m/s^2, the value to be set is 10/9.8/0.0625=16.32, and the last value obtained is an integer and is configured as 16.

Assuming that the InActive Threshold is set to 8m/s^2, the value to be set is 8/9.8/0.0625=13.06, and the last value obtained is an integer and is configured as 13.

Configure device parameters ActiveThreshold=16, InActiveThreshold=13

Downlink: 031C0010000D0000000000

Device returns:831C000000000000000000 (configuration is successful)

831C010000000000000000 (configuration failed)

Read device parameters

Downlink: 041C000000000000000000

Device returns: 841C0010000D0000000000 (device current parameter)

(4) Configure device FilterVelocityThresholdX = FilterVelocityThresholdY =FilterVelocityThresholdZ =10mm/s

Downlink: 051C646464000000000000

Device returns: 851C000000000000000000 (configuration is successful)

851C000000000000000000 (configuration failed)

(5) Read device FilterVelocityThreshold

Downlink: 061C000000000000000000

Device returns: 861C646464000000000000

Note: To filter the abnormal results of velocity detection, the FilterVelocityThreshold is set to make range of velocity reasonable. If the threshold is 6.5mm/s by default, the 3-axis velocity would be compared with it every time after detection. The 3-axis velocity would only be sent when all of them are below 6.5mm/s. A new detection would start as any velocity value is higher than the threshold. If any of the 3-axis velocity values is still higher than the threshold, the data would eventually be reported at the third time of detection.

5.6 Examples of Set/GetSensorAlarmThresholdCmd

Function: The device detects every Min Interval and reports as the data is higher or lower than the threshold.

(Note: The last setting data would be saved if the device is reset to factory setting.)

Fport:0x10

CmdDescriptor	CmdID (1Byte)	Payload(10Bytes)			
SetSensorAlarmThresholdReq	0x01	Channel (1Byte, 0x00_Channel1, 0x01_Channel2, 0x02_Channel3, etc)	SensorType (1Byte, 0x00_Disable ALL SensorthresholdSet 0x0C_Acceleration 0x0D_Velocity, 0x0E_Angle,)	SensorHighThreshold (4Bytes, Unit:same as reportdata in fport6, 0Xffffffff_DISALBLErHighThreshold)	SensorLowThreshold (4Bytes,Unit:same as reportdata in fport6, 0Xffffffff_DISALBLErHigh Threshold)
SetSensorAlarmThresholdRsp	0x81	Status (0x00_success)	Reserved (9Bytes,Fixed 0x00)		
GetSensorAlarmThresholdReq	0x02	Channel (1Byte, 0x00_Channel1, 0x01_Chanel2, 0x02_Channel3, etc)	SensorType (1Byte, Same as the SetSensorAlarmThresholdReq's SensorType)	Reserved (8Bytes, Fixed 0x00)	
GetSensorAlarmThresholdRsp	0x82	Channel (1Byte, 0x00_Channel1, 0x01_Channel2, 0x02_Channel3, etc)	SensorType (1Byte, Same as the SetSensorAlarmThresholdReq's SensorType)	SensorHighThreshold (4Bytes, Unit:same as reportdata in fport6, 0Xffffffff_DISALBLErHighThreshold)	SensorLowThreshold (4Bytes, Unit:same as reportdata in fport6, 0Xffffffff_DISALBLErHigh Threshold)

(1) Configure device parameters SensorHighThreshold = 20m/s^2 (0x00000014), SensorLowThreshold = 10m/s^2 (0x0000000A)

Downlink: 01000C000000140000000A

Device returns: 810000000000000000000000

(2) GetSensorAlarmThresholdReq

Downlink: 02000C0000000000000000

Device returns: 82000C000000140000000A

Clear all sensor alarm thresholds (set SensorType = 0)

Downlink: 0100000000000000000000

Device returns: 8100000000000000000000

Note:

1. Channel 1: X axis;
   Channel 2: Y axis;
   Channel 3: Z axis
2. The device would report only when the status changes.
   For example, when the data is higher than the HighThreshold, the data would be reported. However, if the data still exceeds the HighThreshold in the next detection, the device would not report since its status remains the same.

5.7 Example of GlobalCalibrateCmd

FPort：0x0E

Description	CmdID	Sensor Type	PayLoad(Fix =9 Bytes)				
SetGlobalCalibrateReq	0x01	See below	Channel (1Byte) 0_Channel1 1_Channel2, etc	Multiplier (2bytes, Unsigned)	Divisor (2bytes, Unsigned)	DeltValue (2bytes, Signed)	Reserved (2Bytes, Fixed 0x00)
SetGlobalCalibrateRsp	0x81		Channel (1Byte) 0_Channel1 1_Channel2, etc		Status(1Byte,0x00_success)		Reserved (7Bytes, Fixed 0x00)
GetGlobalCalibrateReq	0x02		Channel (1Byte) 0_Channel1 1_Channel2, etc				Reserved (8Bytes, Fixed 0x00)
GetGlobalCalibrateRsp	0x82		Channel (1Byte) 0_Channel1 1_Channel2, etc	Multiplier (2bytes, Unsigned)	Divisor (2bytes, Unsigned)	DeltValue (2bytes, Signed)	Reserved (2Bytes, Fixed 0x00)
ClearGlobalCalibrateReq	0x03	Reserved 10Bytes,Fixed 0x00)					
ClearGlobalCalibrateRsp	0x83	Status(1Byte,0x00_success)			Reserved (9Bytes,Fixed 0x00)		

Note: The last setting value would be saved if the device is reset back to factory setting after calibration.

SensorType：0x37_AngleSensor

1. SetGlobalCalibrateReq (increase X-axis angle by 20°)

   Multiplier =0x0004, Divisor = 0x0000, DeltValue = 0x03E8

   Downlink: 0137000004000003E80000

   Device returns:8137000000000000000000

2. GetGlobalCalibrateReq

   Downlink: 0237000000000000000000

   Device returns:8237000004000003E80000

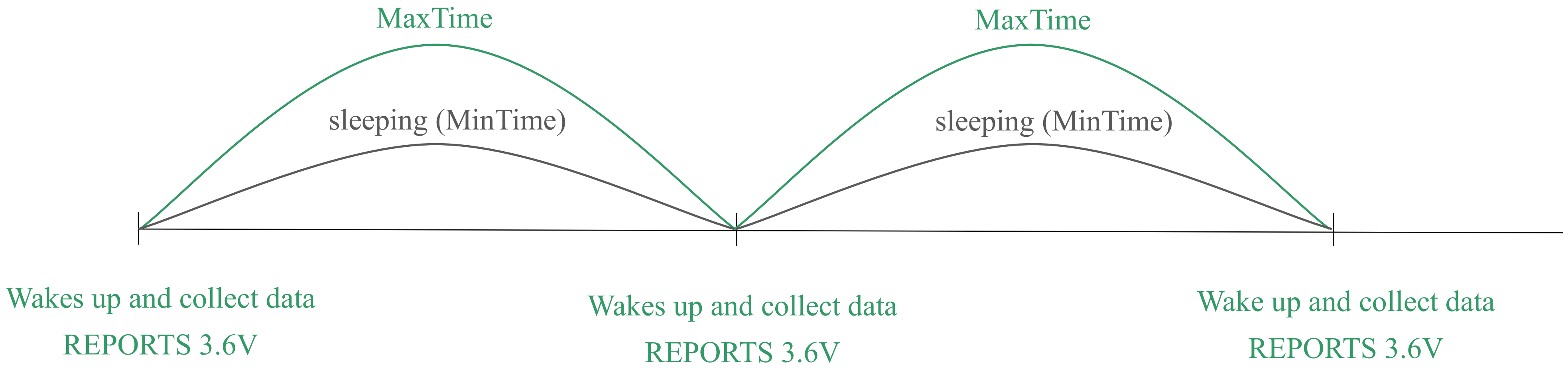
3. ClearGlobalCalibrateReq

   Downlink: 0300000000000000000000

   Device returns:8300000000000000000000

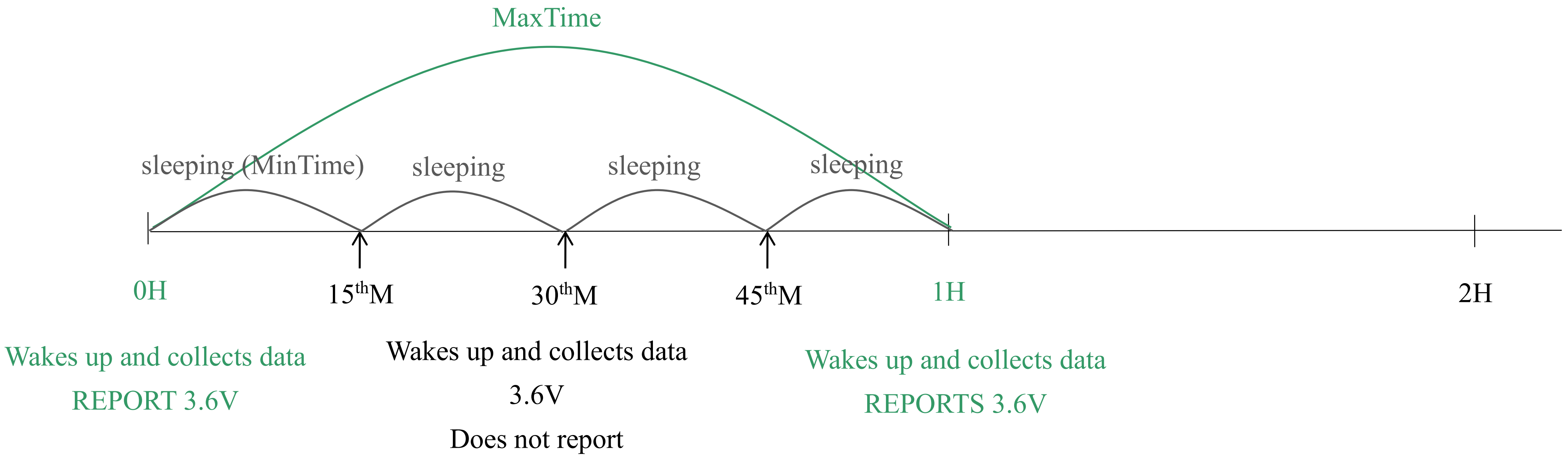
5.8 Example for MinTime/MaxTime logic

Example#1 based on MinTime = 1 Hour, MaxTime= 1 Hour, Reportable Change i.e. BatteryVoltageChange=0.1V

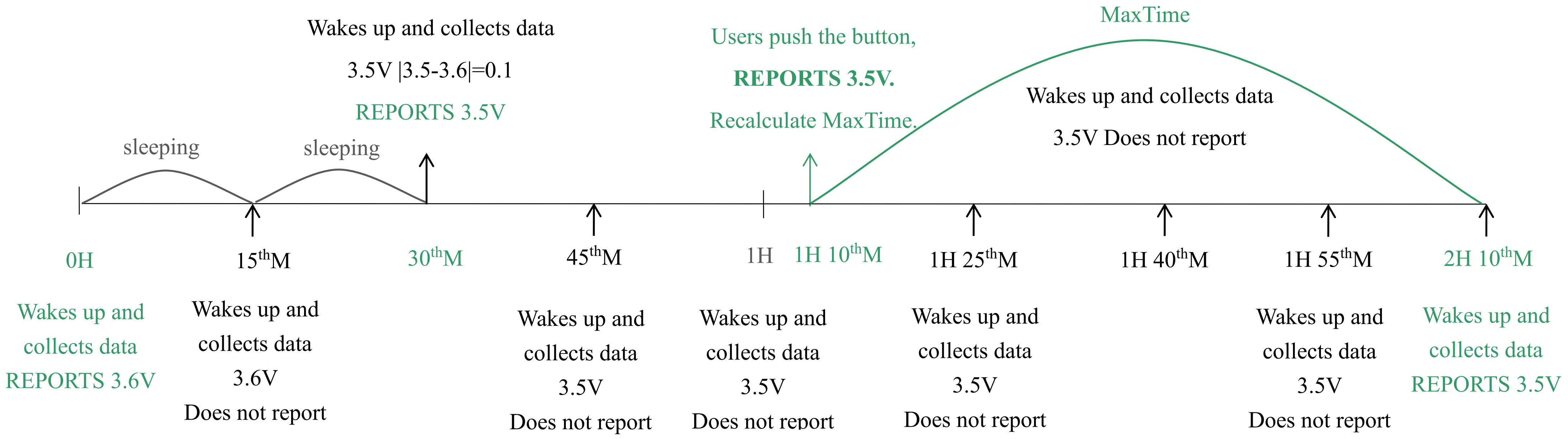


Note: MaxTime=MinTime. Data will only be reported according to MaxTime (MinTime) duration regardless BatteryVoltageChange value.

Example#2 based on MinTime = 15 Minutes, MaxTime= 1 Hour, Reportable Change i.e. BatteryVoltageChange= 0.1V.



Example#3 based on MinTime = 15 Minutes, MaxTime= 1 Hour, Reportable Change i.e. BatteryVoltageChange= 0.1V.


Wakes up and collects data
3.5V |3.5-3.6|=0.1
REPORTS 3.5V
Users push the button,
REPORTS 3.5V.
Recalculate MaxTime.
MaxTime
Wakes up and collects data
3.5V Does not report
sleeping
sleeping
0H
15thM
30thM
45thM
1H
1H 10thM
1H 25thM
1H 40thM
1H 55thM
2H 10thM
Wakes up and collects data REPORTS 3.6V
Wakes up and collects data 3.6V Does not report
Wakes up and collects data 3.5V Does not report
Wakes up and collects data 3.5V Does not report
Wakes up and collects data 3.5V Does not report
Wakes up and collects data 3.5V Does not report
Wakes up and collects data REPORTS 3.5V

Notes:

1) The device only wakes up and performs data sampling according to MinTime Interval. When it is sleeping, it does not collect data.
2) The data collected is compared with the last data reported. If the data variation is greater than the ReportableChange value, the device reports according to the MinTime interval. If the data variation is not greater than the last data reported, the device reports according to the MaxTime interval.
3) We do not recommend setting the MinTime Interval value too low. If the MinTime Interval is too low, the device wakes up frequently and the battery will be drained soon.
4) Whenever the device sends a report, no matter resulting from data variation, button pushed, or MaxTime interval, another cycle of MinTime/MaxTime calculation is started.

6. Applications

In the case of detecting if the generator is working normally, it is recommended to install R718EC horizontal while the generator is power-off and in static status. After installing and fixing R718EC, please turn on the device. After the device is joined, one minute later, R718EC would perform the calibration of the device (the device cannot be moved after the calibration. If it needs to be moved, the device needs to be turned off/powered off for 1 minute, and then the calibration would be performed again). R718EC would need some time to gather the data of the three-axis accelerometer & the temperature of the generator while it is working normally. The data is a reference for the settings of ActiveThreshold & InActiveThreshold, it is also for checking if the generator is working abnormally.

If the collected Z Axis Accelerometer data is stable at 100m/s², the error is ±2m/s², the ActiveThreshold can be set to 110m/s², and the InActiveThreshold is 104m/s².

Note: (1) Please do not disassemble the device unless it is required to replace the batteries.
(2) Do not touch the waterproof gasket, LED indicator light, or function keys when replacing the batteries. Please use a suitable screwdriver to tighten the screws (if using an electric screwdriver, it is recommended to set the torque as 4kgf) to ensure the device is impermeable.

7. Installation

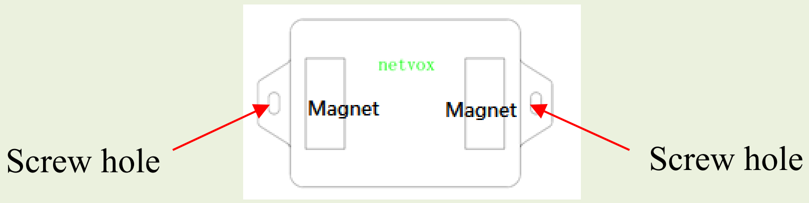
1. The R718EC main body has a built-in magnet (as shown in the figure below). During installation, the main body can be attached to the surface of the object with iron. In order to make the installation more secure, if possible, please use screws (to be purchased) to fix the device to the motor surface.

2. Before installing the three-axis accelerator, tear off the 3M adhesive on the back and paste it on the motor plane.

3. When installing NTC, lock it on the motor with appropriate screws. The contact surface shall be cleaned and coated with thermal conductive adhesive before locking.

Notes:

Do not install the device in a metal shielding box or in an environment with other electrical device around, so as to avoid affecting the wireless transmission signal of the device.



R718EC is applicable to the following scenarios:

- Industrial device
- Mechanical device

And other occasions where it is necessary to detect whether the motor operates normally.

4. Installation precautions:

During installation, the device shall be installed when the device is powered off and the motor is stationary. It is recommended to install it horizontally. Power on the device after fixing. The offset calibration of the device shall be carried out one minute after the network is added (the device cannot be moved after the offset calibration. If it needs to be moved, the device needs to be powered off for 1 minute and then re offset calibrated). The device needs a period of time to collect the three-axis acceleration and temperature of the motor under normal operation, so as to make reference for the setting of static threshold and motion threshold and whether the motor works abnormally.

Assume that the collected Z-axis acceleration is stable at $100m/s^2$, the error is ± $2m/s^2$, the activity threshold can be set to $110m/s^2$, static threshold is $104m/s^2$. The specific configuration shall be based on the actual situation.

For the configuration of active threshold and static threshold, please refer to the command document.

5. When the device detects that the three-axis acceleration exceeds the set activity threshold, it immediately sends the currently detected value. After sending the three-axis acceleration and speed, the next detection can be carried out only after the triaxial acceleration of the device is lower than the set static threshold and lasts for more than 5 seconds (not modifiable).

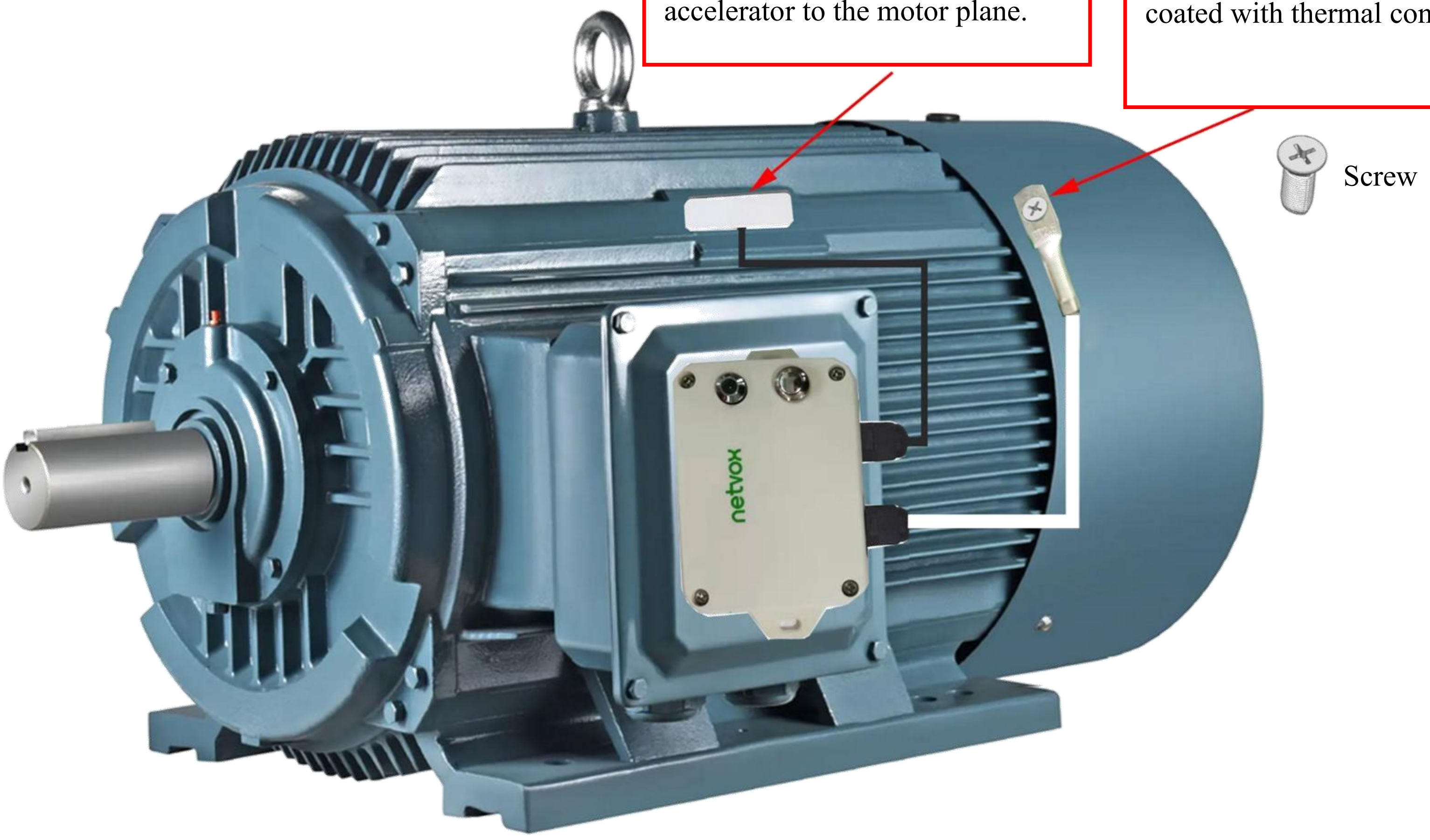
Notes:

- When the three-axis acceleration of the device is lower than the set static threshold and the duration is less than 5 seconds, if vibration continues to occur (the three-axis acceleration is higher than the set static threshold), it will be postponed for 5 seconds. Until the three-axis acceleration is lower than the static threshold and lasts for more than 5 seconds.

- The device will send two data packets. One is the three-axis acceleration. After 10 seconds, it will send the three-axis speed and temperature.

Three-axis Accelerometer Remove 3M adhesive from the back and stick the three-axis accelerator to the motor plane.

NTC shall be locked on the motor with appropriate screws. Before locking, the contact surface shall be wiped clean and coated with thermal conductive adhesive.



8. Information about Battery Passivation

Many of Netvox devices are powered by 3.6V ER14505 Li-SOCl2 (lithium-thionyl chloride) batteries that offer many advantages including low self-discharge rate and high energy density. However, primary lithium batteries like Li-SOCl2 batteries will form a passivation layer as a reaction between the lithium anode and thionyl chloride if they are in storage for a long time or if the storage temperature is too high. This lithium chloride layer prevents rapid self-discharge caused by continuous reaction between lithium and thionyl chloride, but battery passivation may also lead to voltage delay when the batteries are put into operation, and our devices may not work correctly in this situation.

As a result, please make sure to source batteries from reliable vendors, and it is suggested that if the storage period is more than one month from the date of battery production, all the batteries should be activated. If encountering the situation of battery passivation, users can activate the battery to eliminate the battery hysteresis.

ER14505 Battery Passivation

8.1 To determine whether a battery requires activation

Connect a new ER14505 battery to a resistor in parallel, and check the voltage of the circuit.

If the voltage is below 3.3V, it means the battery requires activation.

8.2 How to activate the battery

a. Connect a battery to a resistor in parallel

b. Keep the connection for 5~8 minutes

c. The voltage of the circuit should be ≧3.3, indicating successful activation.

Brand	Load Resistance	Activation Time	Activation Current
NHTONE	165 Ω	5 minutes	20mA
RAMWAY	67 Ω	8 minutes	50mA
EVE	67 Ω	8 minutes	50mA
SAFT	67 Ω	8 minutes	50mA

Note:

If you buy batteries from other than the above four manufacturers, then the battery activation time, activation current, and required load resistance shall be mainly subject to the announcement of each manufacturer.

9. Important Maintenance Instruction

Kindly pay attention to the following in order to achieve the best maintenance of the product:

- Do not put the device near or submerge into water. Minerals in rain, moisture, and other liquids could cause corrosion of electronic components. Please dry the device, if it gets wet.
- Do not use or store the device in dusty or dirty environments to prevent damage to parts and electronic components.
- Do not store the device in extremely hot places. This may shorten the lifespan of electronic components, damage batteries, and deform plastic parts.
- Do not store the device in extremely cold places. Moisture may damage circuit boards as the temperatures rise.
- Do not throw or cause other unnecessary shocks to the device. This may damage internal circuits and delicate components.
- Do not clean the device with strong chemicals or detergents.
- Do not apply the device with paint. This may block detachable parts and cause malfunction.
- Do not dispose of batteries in fire in case they explode.

The instructions are applied to your device, battery, and accessories. If any device is not working properly or has been damaged, please send it to the nearest authorized service provider for service.