

NIS-TR Temperature Sensor (RTD)

General information

Overview

LoRa™ WAN 1.0.2 EU Class A & C

Activation by Personalization
(ABP)

Over-The-Air Activation
(OTAA)

SIGFOX™ ready

Fully certified on major LoraWan &
Sigfox networks

Easy to use and deploy

3.6V / 2200mAh Primary battery
Li-SOCI2 embedded

Magnetic switch and LED to ease the installation

RoHS conform / Pre-Certified ETSI EN 300-220 / RED compliance under certification

RF Performance

- TX Power 14 dBm (25 mW)
- RX Sensitivity -137 dBm
- Approx. 10 km range
- Integrated PCB antenna

Environment

- Operating temperature -25°C to +60°C
- Storage temperature -40°C to 85°C
- Rel. humidity 0% to 95% (not condensing)

Options

- External antenna
- SDK for your SW development purposes
- 10-30V external power supply
- Compatible with the NEMEUS PicoGateway for private networks

Installation

By using the internal antenna, the position of the sensor is critical.

Install your product as high as possible, and prefer unobstructed area (concrete and metal are the worst materials).

If you cannot fulfill these conditions, it is still possible to use an optional external antenna.



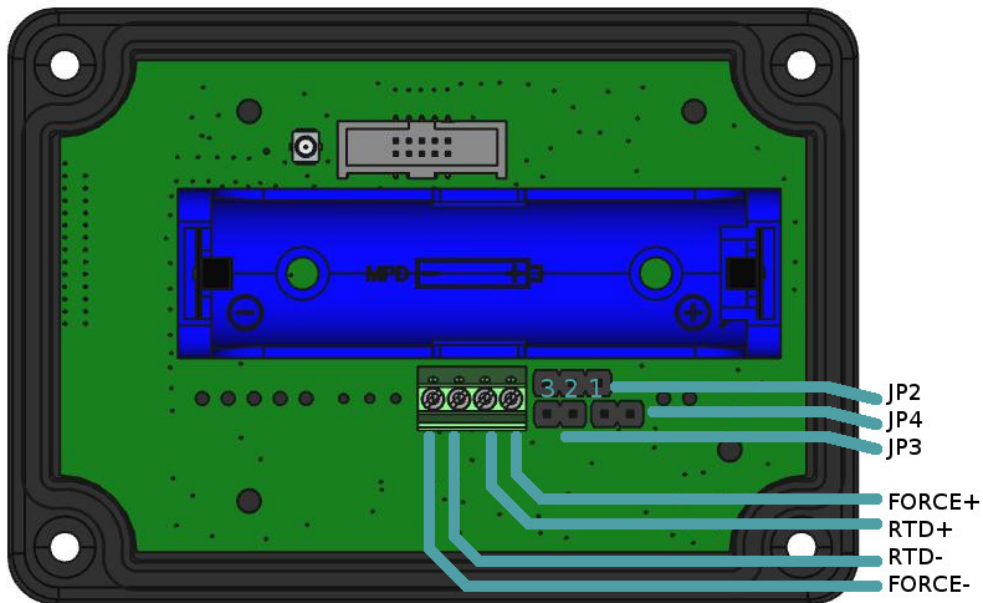
Mechanics

- Dimensions 60 x 95 x 85mm (109mm including fixing brackets)
- 96mm spacing between brackets
- IP 65
- Solid ABS & PC material

RTD Temperature sensor properties

- Handles 100Ω to 1kΩ (at 0°C) Platinum RTDs (PT100 to PT1000)
- 2-wires, 3-wires or 4-wires connection
- 15-Bit ADC Resolution; Nominal Temperature Resolution 0.03125°C (Varies Due to RTD Nonlinearity)
- Total Accuracy Over All Operating Conditions: 0.5°C (0.05% of Full Scale) max
- ±45V Input Protection
- Fault Detection (Open RTD Element, RTD Shorted to Out-of-Range Voltage, or Short Across RTD Element)
- Direct fifth-order linearization for best accuracy

Wiring



Probe settings

number of wires	jumpers configuration	Probe connection
4 wires	JP2 jumper on 1-2	FORCE+ IN+ IN- FORCE-
	JP3 NC	
	JP4 NC	
3 wires	JP2 jumper on 2-3	FORCE+ IN+ IN-
	JP3 NC	
	JP4 jumper ON	

2 wires	JP2 jumper on 1-2	IN+ IN-
	JP3 jumper ON	
	JP4 jumper ON	

Magnetic Switch Protocol description

Generic Magnetic Switch Protocol description

Nemeus sensors embed a magnetic switch and a green LED to implement a simple man to machine interface.

The LED blinks every second while the magnet is present.

If the magnet is present during 20 seconds, it triggers an hardware reset.

The 20 seconds range has been cut in 10 timeslots of 2 seconds, the actions performed for each timeslot (TSX) are the following:

- 0s <= TS1 < 2s: read current state, number of led blinks indicates the following states:
 - 1 blink: standby mode
 - 2 blinks: joint with LoRaWAN network only (Sigfox is disabled)
 - 3 blinks: Sigfox enabled only (no LoRaWAN)
 - 4 blinks: joint with LoRaWAN network and Sigfox is enabled
 - 6 blinks: joining LoRaWAN network (no Sigfox)
 - 8 blinks: joining LoRaWAN network and Sigfox is enabled
 - 9 blinks: measures are ongoing but transmission is forbidden
- 2s <= TS2 < 4s: switch to normal mode (if not already in normal mode) or force one measure and one Tx (if already in normal mode)
- 4s <= TS3 < 6s: switch to standby mode, it is the default mode when the device is stored and not yet activated
- 6s <= TS4 < 8s: perform a warm reset, the device resets and goes back to its previous mode
- 8s <= TS5 < 10s: forbid transmission, the device still performs measures but never transmit them
- 10s <= TS6 < 12s: switch to test mode, the device suspends the measures and forbid transmission, then it switches to continuous Rx on a specific channel and wait for test mode commands, if nothing is received after 60 seconds then it goes back to its previous mode. During test mode the led blinks and the magnetic switch is disabled.
- 12s <= TS7 < 14s: no action
- 14s <= TS8 < 16s: no action
- 16s <= TS9 < 18s: no action
- 18s <= TS10 < 20s: no action
- 20s <= TSHw: hardware reset

When the magnet is removed, the LED blinks fast X times, X identifies the timeslot (1 to 10).

Standby mode

It is the default mode set in factory, the device does nothing and consumes few μA (consumption depends on the sensor type).

Normal mode

The first time the device switches to normal mode, it performs the join procedure with the LoRaWAN network.

Then it performs a first measure and transmits it to the network whatever the number of measures required to trigger a transmission.

Finally it performs measures according to the configured period and transmits them according to configured number of measures required to trigger a transmission.

Forbid transmission mode

In this mode the device performs measures according to the configured period but never transmits them to the LoRaWAN network. This mode is mainly used in R&D phase to tune the measure algorithm, it shouldn't be used when the device is deployed.

Test mode

This mode is used to modify the device configuration over the air using a Nemeus tool, this tool must be in the vicinity of the device. A security key is used to authenticate the Nemeus tool and the device.

Warm reset

The warm reset restarts the device and put it in its previous state. For example, if the device was joining the network, it restarts its join procedure. If it was already joint and performing periodic measures then it transmits the first measure and restarts the periodic measures.

Radio frames description

Generic Application Protocol description

Nemeus uplink protocol

Nemeus uplink protocol is used to send sensor data to application server. The payload contains the following fields:

- **<mask>**: At least 1 byte containing a bit field identifying the fields included in the frame. The meaning of the mask is specific to the couple device/port. Thus, the application server knows the mask meaning according to the data profile linked to the couple device/port. The most significant bit of the **<mask>** is used to extend the mask on the next byte: 1 means not extended (mask is on 7 bits) and 0 means extended (mask is on 14 bits). When extended, the following byte can also be extended according to its most significant bit. When the **<mask>** field is on several bytes, it is encoded in little endian (LSB first). For example, a **<mask>** coded on 2 bytes like "00100001 11001100" will be on 14 bits: "10011000100001". Each bit indicates the presence/absence of a field. When a field is present, it appears in the frame according to the bit position (beginning with the least significant bit).
 - **<field1>**: first data field identified by the data profile linked to the couple device/port.
 - ...
 - **<fieldN>**: last data field identified by the data profile linked to the couple device/port.
-

Nemeus downlink protocol

Nemeus downlink protocol is used to exchange commands between application servers (located in network infra or in configuration PC) and sensor embedded applications.

Application port is used to address an application (application ports are bound with LoRaWAN ports). Thus, the protocol consists in a binary payload exchanged on a port. The payload contains the following fields:

- <cmd>: 1 byte identifying an applicative command/response.
- <mask>: At least 1 byte containing a bit field identifying the fields included in the frame. See uplink protocol above for description.
- <field1>: first data field identified by the data profile linked to the couple device/port.
- ...
- <fieldN>: last data field identified by the data profile linked to the couple device/port.

Most significant bit of <cmd> field specifies if response is expected by the server or not (commands of type “set” doesn’t require any answer, especially when radio interface is used).

<cmd> values are defined for each application port (e.g. <cmd> value 0x01 has a different meaning on port 1 than on port 2).

A response reuse the same <cmd> as the received command, except the most significant bit which is forced to 0 (because a response doesn’t expect any response). The response <mask> indicates which fields of the command have been actually processed.

When a parameter type size is more than one byte, it is encoded in little endian (LSB first).

Downlink command may be received either from network application server or from PC configuration tool through AT command.

The generic application AT command is used:

AT+GA=DIND,<port>,<hexpayload>.

Uplink data Frame format

The uplink payload contains multiple fields: <mask>: bit field on 1 byte indicating the presence of other fields. Bit 7 is always to 1 because the mask is never extended on next byte.

bit 0 : indicates the presence of <nb_meas> field.

bit 1 : indicates the presence of <rt_d_temp> field.

bit 2 : indicates the presence of <voltage> field.

bit 3 : indicates the presence of <internal_temp> field.

bit 4 : is reserved.

bit 5 : indicates the presence of <cause> field.

When bit 0 is not set, it means that other fields are present only once (nb_meas = 1).

if (mask.bit0 == 1) :

```
<nb_meas>: 1 byte containing the number of measurements which follows.
```

if (mask.bit1 == 1) :

```
<rt_d_temp>: 2*<nb_meas> bytes containing the temperature values in 0.1 °C. Each temperature value is a signed integer on 16 bits in big endian [-32768..+32767].
```

if (mask.bit2 == 1) :

`<voltage>`: $2 * \text{nb_meas}$ bytes containing the voltage values measured by the sensor in millivolts. Each voltage value is an unsigned integer on 16 bits in big endian.

if (mask.bit3 == 1)

`<internal_temp>`: $1 * \text{nb_meas}$ bytes containing the internal temperature values (using MCU internal sensor). Each temperature value is a signed integer on 8 bits in big endian in °C [-128..+127].

if (mask.bit5 == 1) :

`<cause>`: 1 byte containing the cause of the uplink frame. It is a bit field:

- Bit 0 indicates a periodic measure
- Bit 1 indicates the high threshold exceeding (temperature > high threshold)
- Bit 2 indicates the high hysteresis exceeding (temperature < high threshold - high hysteresis)
- Bit 3 indicates the low threshold exceeding (temperature < low threshold)
- Bit 4 indicates the low hysteresis exceeding (temperature > low threshold + low hysteresis)
- Bit 5 indicates the measure has been forced manually (with magnetic switch)

Usually `<cause>` field is absent when threshold detections are disabled (periodic measures only)

Thus the uplink frame format is `<mask><nb_meas><rtd_temp1>...<rtd_tempN><voltage1>...<voltageN><internal_temp1>...<internal_tempN><cause>`

Default `<mask>` value is 0x83 which means that the default frame is `<mask><nb_meas><rtd_temp1>...<rtd_tempN>`

Some examples (in hexadecimal):

"820119": the frame indicates one temperature measurement of +28.1 °C.
 "8302FFCB000C": the frame indicates two temperature measurements of -5.3 °c and +1.2 °C.

Downlink Frame format

The sensor configuration contains the following fields:

`<up_frame_mask>`: the mask identifying the different fields present in uplink frames (possible values are described in "Uplink data Frame format" section)

`<nb_meas_for_tx>`: the number of measures required to trigger a transmission (possible values are [1..8])

`<meas_period>`: the period of the measures in `<perdioid_unit>` seconds (possible values are [600..65535])

`<period_unit>`: the unit of `<meas_period>` in seconds, the resulting period is `<meas_perios> * <period_unit>` seconds

`<mode>`: bit field enabling/disabling the measures (threshold detections and periodic measures)

`<subperiod>`: subdivision of period for threshold detection, actual measurement period is $(\text{nb_meas_for_tx} * \text{period_unit}) / \text{subperiod}$ seconds

`<high_threshold>`: value of high threshold

`<high_hysteresis>`: value of high hysteresis

`<low_threshold>`: value of low threshold

```

<low_hysteresis>: value of low hysteresis
<sensor_type>: identifies the type of sensor (PT100/PT1000 and
2/3/4 wires)

```

It is possible to change the sensor configuration using the Nemeus downlink protocol. The downlink frame must be sent on the MS008 LoRaWAN port (10).

The downlink frame has the following format:

<cmd>: 1 byte containing WRITE_CFG_CMD=0x02

<mask>: bit field on 1 byte indicating the presence of other fields. Most significant bit (bit7) is always to 1 because the mask is never extended on next byte.

Bit0 indicates the presence of <up_frame_mask> field

Bit1 indicates the presence of <nb_meas_for_tx> field

Bit2 indicates the presence of <meas_period> and <period_unit> fields

Bit3 indicates the presence of <mode> and <subperiod> fields

Bit4 indicates the presence of <high_threshold> and <high_hysteresis> fields

Bit5 indicates the presence of <low_threshold> and <low_hysteresis> fields

Bit6 indicates the presence of <sensor_type> field

If (mask.bit0 == 1):

```

<up_frame_mask>: 1 byte containing the uplink frame mask

```

If (mask.bit1 == 1):

```

<nb_meas_for_tx>: 1 byte containing the number of measurements required to trigger a transmission

```

If (mask.bit2 == 1):

```

<meas_period>: 2 bytes in little endian (LSB first) containing the period of measurements in <period_unit> seconds
<period_unit>: 1 byte containing the unit of <meas_period> in seconds (default value is 1 second)

```

If (mask.bit3 == 1):

```

<mode>: 1 byte encoded as follows:
  bit0: enable(1)/disable(0) high threshold detection
  bit1: enable(1)/disable(0) high hysteresis detection (when leaving high threshold state)
  bit2: enable(1)/disable(0) low threshold detection
  bit3: enable(1)/disable(0) low hysteresis detection (when leaving low threshold state)
  bit4: enable(1)/disable(0) periodic measurements
  bit5 to bit7: reserved
<subperiod>: 2 bytes in little endian containing the period subdivision

```

If (mask.bit4 == 1):

```

<high_threshold>: signed integer on 2 bytes in little endian containing the high threshold in 0.1 °C [-2000..+8000]
<high_hysteresis>: unsigned integer on 1 byte containing the hysteresis when leaving high threshold state in 0.1 °C [0..255]

```

If (mask.bit5 == 1):

```

<low_threshold>: signed integer on 2 bytes in little endian containing the low threshold in 0.1 °C [-2000..+8000]
<low_hysteresis>: unsigned integer on 1 byte containing the hysteresis when leaving low threshold state in 0.1 °C [0..255]

```

If (mask.bit6 == 1):

```
<sensor_type>: unsigned integer on 1 byte containing the sensor type [11..16]
  11(0x0B): PT100 2 wires
  12(0x0C): PT100 3 wires
  13(0x0D): PT100 4 wires
  14(0x0E): PT1000 2 wires
  15(0x0F): PT1000 3 wires
  16(0x10): PT1000 4 wires
```

Thus the downlink frame format is
 <cmd><mask><up_frame_mask><nb_meas_for_tx><meas_period><period_unit><mode><subperiod><high_threshold><high_hysteresis><low_hysteresis><sensor_type>.

Some examples in hexadecimal:

```
"028182": set the <up_frame_mask> to 0x82 => the next
uplink frames will include the <rtd_temp> field only
"028E04605401100100": set <nb_meas_for_tx> to 4,
<meas_period> to 21600*<period_unit> seconds,
<period_unit> to 1 second and enable periodic measurements only
with subperiod to 1 => one measurement every 6 hours and one
transmission every 4 measures (every day). Each uplink frame will
include 4 measurements
```

Power consumption

Power consumption calculator based on the number of measurements and transmissions.

START_WIDGET97419f636e8a0555-0END_WIDGET

Article Sources and Contributors

NIS-TR Temperature Sensor (RTD) *Source:* <http://wiki.nemeus.fr/index.php?oldid=218> *Contributors:* Arl, Elc, Lcs

Image Sources, Licenses and Contributors

File:1 Exp PE RTD sensor.png *Source:* http://wiki.nemeus.fr/index.php?title=File:1_Exp_PE_RTD_sensor.png *License:* unknown *Contributors:* Lcs

File:MS008 config3.jpeg *Source:* http://wiki.nemeus.fr/index.php?title=File:MS008_config3.jpeg *License:* unknown *Contributors:* Arl