
SENS'IT 3 USER MANUAL

Table of contents

Table of contents.....	1
SENS'IT 3 - 5 SENSORS, ONE BUTTON, SO MANY POSSIBILITIES	1
DEVICE ACTIVATION	1
USER INSTRUCTIONS	1
AVAILABLE MODES	2
TECHNICAL SPECIFICATIONS.....	2
SENS'IT FOR DEVELOPERS	3

SENS'IT 3 - 5 SENSORS, ONE BUTTON, SO MANY POSSIBILITIES

Sens'it is a configurable multi-sensor device: you can adapt it to many use cases, and build your own usage. The number of sensors it provides, along with its ability to communicate wherever there is Sigfox coverage, open it to a wide array of possibilities.

DEVICE ACTIVATION

It only takes a few minutes to get started with your Sens'it:

1. Visit www.sensit.io and click on "Start with my Sens'it".
2. Use the ID that is printed on the back of the Sens'it.
3. Make a long press (two seconds) on the button to change mode and send a message.
The main LED will turn light up in the color of the chosen mode (see below).
4. Wait until the LED blinks: this shows that data is being transmitted.
5. You can now access your sensit.io dashboard and see the transmitted data.

USER INSTRUCTIONS

The Sens'it is easy to configure:

1. Select the mode you want to use (see below).
2. Place the Sens'it on asset or in the location about which you want information and leave it.
...or press the Sens'it button twice to send data.
3. Keep track of your data on the online dashboard.
...or receive notifications when a threshold is passed.

The Sens'it button has a different action depending on how it is pressed:

- **Indicate the current mode:** one short press. The LED will blink in the current mode color.
- **Change mode:** one long press. The LED changes color depending on the active mode. You can keep pressing the button and it will keep switching mode until you stop. Once a mode is chosen, the Sens'it sends data.

- **Send data with the current measurement:** 2 short presses. This can be used to send an alert.
- **Reset the Sens'it:** 4 shorts presses (in Button mode only). The Sens'it returns to factory settings.

AVAILABLE MODES

The Sens'it can be configured in 6 different modes. Each mode has a specific color:

- **Green:** Temperature and Humidity mode. The Sens'it sends sensor data at a defined frequency.
- **Yellow:** Light mode. The Sens'it either sends light sensor data at a defined frequency.
- **Light blue:** Door opening detection. The Sens'it is affixed to a door, and sends data every time the door is being opened (through its magnetometer).
- **Dark blue:** Vibration detection. The Sens'it sends data every time the accelerometer has detected a movement.
- **Pink:** Magnet mode. The Sens'it sends data every time a magnet is put near the right side of the Sens'it.
- **White:** Button mode. The Sens'it sends data every time the button is pressed.

You can change mode by doing a long press on the button.

The sending frequency can be defined on the online app: every 10 minutes, every hour, every 6 hours, every 24 hours. You can also trigger an immediate data transmission by making two short presses on the button.

By default, only one mode can be selected at a time.

TECHNICAL SPECIFICATIONS

RADIO CHARACTERISTICS

	3.1	3.2	3.3	3.4
TX frequency	868.13 MHz	902.2 MHz	923.2 MHz	920.8 MHz
RX frequency	869.525 MHz	905.2 MHz	922.2 MHz	922.3 MHz
TX power	14 dBm	22 dBm	7 dBm	22 dBm
Sigfox Ready class	U0	U0	U1	U0

MECHANICAL CHARACTERISTICS

- Dimensions (W x H x D): 52 mm x 82 mm x 15 mm
- Weight: 50 g
- Solid particle protection: IP54

ENVIRONMENTAL CHARACTERISTICS

-
- Operating temperature: -10 °C to +55 °C
 - Relative humidity: 30 % to 90 %
 - Storage temperature: -20 °C to +75 °C

POWER SUPPLY CHARACTERISTICS

- Battery type: LiPo
- Technical characteristics: 250 mAh, 3.7 V
- Recharge: 5 V, via USB

SENS'IT FOR DEVELOPERS

Sens'it introduces everyone to the possibilities of Sigfox through a configurable end-to-end solution.

With the Sens'it SDK, developers can use their Sens'it as a devkit, build their own firmware, have access to the Sigfox Cloud... all that with Sigfox connectivity included!

Learn more about the Sens'it SDK on www.sensit.io!