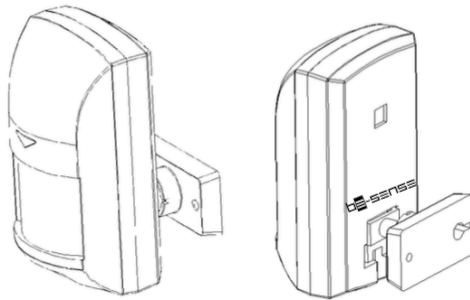




Zwave PIR Sensor



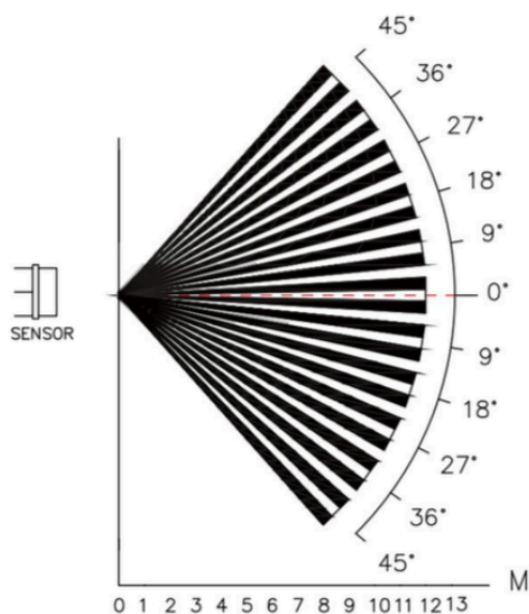
I. Introduction

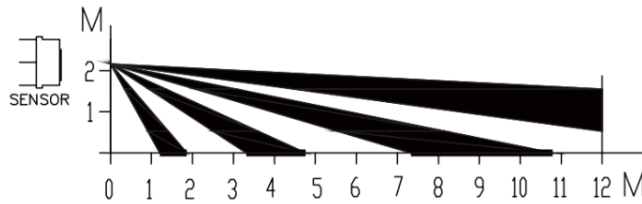
X30_ZWAVE wireless PIR sensor adopts latest detection & induction technologies including Energy Accumulation Management, Dynamic Random Time Division Technology etc. And it combines multi-zones comprehensive induction calculation and uses advanced MCU digital processing technology. Besides, it has the advantages of stable & reliable detection, low false alarm and pet immune.

- Special Fresnel lens with patented technology, built-in far infrared wave with above 95% transmittance rate
- Adopt original imported Heimann sensor from Germany, with high sensitivity & low false alarm
- Real anti-visible light technology, no false alarm under 6500LUX strong light interference With sealed induction cavity design, prevent false alarm caused by thermal current interference
- Configurable detection sensitivity, suitable for variety of detection environments With fuzzy logic algorithm & fuzzy recognition technology, pet-immune Ultra-low power consumption, 2 years of battery working life

Name	Parameter
Model No.	IX30_ZWAVE
Communication Protocol	ZWAVE_868MHz /908MHz
Indoor Transmit Distance	≥30m
Operating Voltage	DC3V (2PCS LR6 1.5V alkaline battery)
Operating Current	Static current≤16uA; transmit current≤35mA
Detection Distance	10m @ 25℃
Detection Area	See Figure 3
Alarm Indicator	LED status indicator
Output Signal Type	Alarm report, tamper report, battery level status, heartbeat report
Working Humidity & Temperature	-10℃~50℃; ≤95%RH no condensation
Infrared Area	11+8+6+5 (typical)
Max Coverage	10m*10m (32*32 feet)/90
Installation Height	2.2~2.7m (87~106 inch)
Anti-visible Light (indoor)	>6500LUX
Dimension	110*62*47mm (L*W*H)

Detection Area





II. Network Inclusion/ Exclusion

The sensor must be added to the Zwave network prior to use. To include the sensor in a network both the sensor and the Network controller or HUB must be in inclusion mode at the same time.

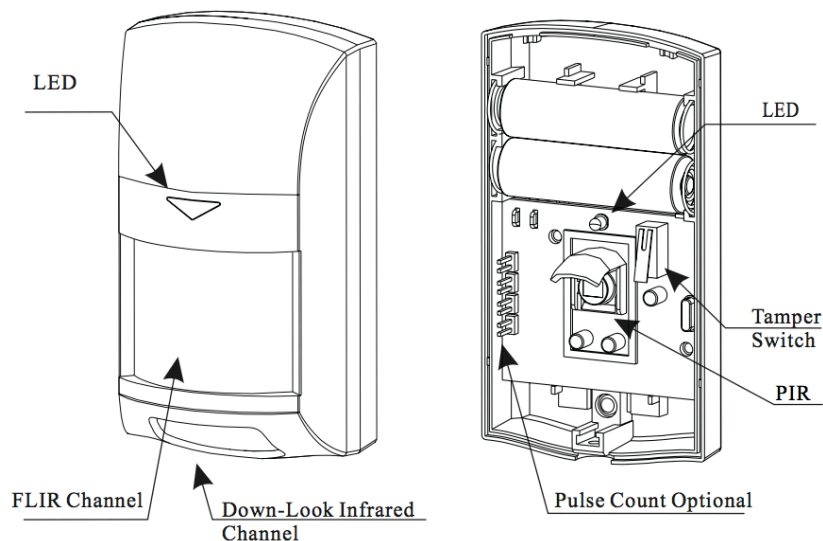
Add : start by placing the controller in inclusion mode. Activate the inclusion mode at the sensor pressing the tamper switch 3 times, then the door sensor will stay in enrollment state.

Wait about 15-30 seconds while the sensor and controller finished the inclusion process.

Remove: enter the Exclusion Mode on the controller, and press the tamper switch 3 times, then the door sensor will be removed after 15-30 seconds.

III. Restore Factory Settings

Press the tamper switch for 6 times to restore factory settings.



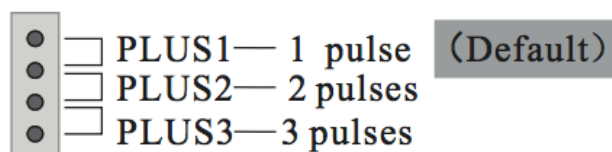
IX 30 has the following 3 types of pulse optional and 3 operates mode:

Pulse 1: The detector gives an alarm when it checks one pulse.

Pulse 2: The detector gives an alarm when it checks two pulses.

Pulse 3: The detector gives an alarm when it checks three pulses.

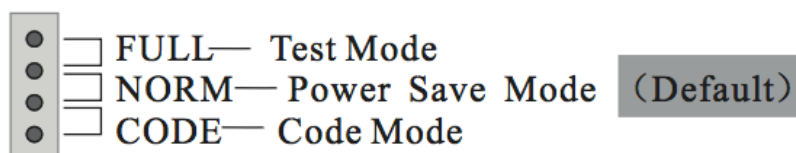
The pulse count is higher and the sensitivity is lower, but can reduce false alarm.



Test Mode (High Battery consumption) : Alarm will be triggered each 5s.

Power-save Mode (Recommended): Alarm will be triggered again with above 3min interval from its first alarm. After triggered, it will send alarm elimination signal to the relevant device in at least 3 minutes.

Code Mode: without any function





Here are some instructions that should help you get our Motion Sensor device handler to work in SmartThings.

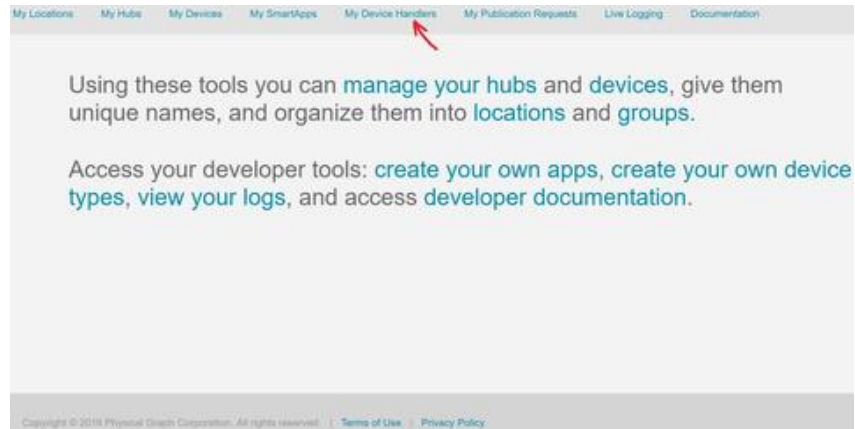
Why update the handler?

- Customized to avoid false alarms
- Generate Tamper alerts
- Battery indicator
- Colors and Background improved

How to ADD a device Handler for the BeSense Motion Sensor?

1. Log in here with the same username and password you use for your SmartThings app: <https://graph.api.smartthings.com/login/auth>

2. Click on My Device Handlers in the top menu



3. In the top right corner, click on **Create new device handler** and choose **From code** from the top tab menu. You'll be presented with a blank area where you can copy the code from this link and paste it in (it's always best to use the raw version of the code)

https://besense-iot/smarthings/besense_handler.txt



My Locations My Hubs My Devices My SmartApps My Device Handlers My Publication Requests Live Logging Documentation

Create New Device Handler

Every device requires a **Device Handler** to be recognized by the SmartThings platform. Select one of the options below to create a Device Handler for your device. Then, to test your device, publish the Device Handler to the SmartThings hub. After successful testing, you may [submit](#) your Device Handler and device for publication and certification, where someone from our certification team will get in touch with you to complete the pairing.

From Form **From Code** From Template From ZigBee Device Fingerprint

Name: Device Handler Name
Name of this device handler. By convention capitalized with words separated by spaces, e.g. My First Device Handler

Namespace: Namespace
Used to uniquely identify device handlers. We suggest using your GitHub username.

Author: LorenzoHD
Full name of the original author of this device handler

☒ Include Apache2.0 license in source

Capabilities:

- ☐ Acceleration Sensor
- ☐ Actuator
- ☐ Alarm
- ☐ Battery

4. Next, scroll to the bottom and click **Create**. Now click Save, then click **Publish > for me**.

My Locations My Hubs My Devices My SmartApps My Device Handlers My Publication Requests Live Logging Documentation

Create New Device Handler

Every device requires a **Device Handler** to be recognized by the SmartThings platform. Select one of the options below to create a Device Handler for your device. Then, to test your device, publish the Device Handler to the SmartThings hub. After successful testing, you may [submit](#) your Device Handler and device for publication and certification, where someone from our certification team will get in touch with you to complete the pairing.

From Form From Code **From Template** From ZigBee Device Fingerprint

```

1  commands(request) + ["delay 20000", zwave.wakeUpV1.wakeUpMoreInformation().format()]
2  }
3
4  private setConfigured() {
5      updateDataValue("configured", "true")
6      return {}
7  }
8
9  private isConfigured() {
10     getDataValue("configured") == "true"
11 }
12
13 private command(physicalgraph.zwave.Command cmd) {
14     if (state.sec) {
15         zwave.securityV1.securityMessageEncapsulation().encapsulate(cmd).format()
16     } else {
17         cmd.format()
18     }
19 }
20
21 private commands(commands, delay=1000) {
22     delayBetween(commands.collect{ command(it) }, delay)
23 }
24
25

```

Create Cancel



You should now see this device type in your list of Device Handlers

How To assign a custom **DEVICE HANDLER** to my zwave device

Simply exclude the device if you had included it prior to installing the new device handler. Then add the device to your SmartThings hub by clicking on Add thing > + Connect New Device in your app. Press and release the tamper switch three times on the sensor quickly.

Remember to keep the device within 3 feet away from your hub during inclusion — the device should be automatically discovered as the BeSense Motion sensor. Then, leave the sensor by the hub for a few hours so it can fully configure.

And that should be it! I hope this was helpful in simplifying the process behind custom device handlers. Feel free to contact us if you have any question.

BeSense Z-Wave

Command Class Specification

When the door sensor is opened or recovered, it will send “Binary Sensor Report” and “Notification Report” commands to the device under Lifeline group.

When door sensor is opened:

Sensor Binary Report, Value = 0xFF, Type = 0x0C

Notification Report, Notification Type = 0x07, Event = 0x08 When door sensor is recovered:

Sensor Binary Report, Value = 0x00, Type = 0x0C Notification Report, Notification Type = 0x06, Event = 0x16

When tamper switch is triggered or recovered, the door sensor will send “Sensor Binary Report” and “Notification Report” command to the device under Lifeline group.

Tamper Triggered:

Sensor Binary Report, Value = 0xFF, Type = 0x08

Notification Report, Notification Type = 0x07, Event = 0x00 Tamper recover (press tamper switch for 0.5s):

Sensor Binary Report, Value = 0x00, Type = 0x08 Notification Report, Notification Type = 0x07, Event = 0x00

Association Group2

If there is any device under Association Group2, the door sensor will send “BASIC SET” command to control those devices when the door sensor is triggered. For example: when the door sensor is triggered, it sends adjustable parameter “BASIC SET” command to a lamp under Group2, you can adjust the lamp’s luminance through the parameters of this command; if the set light-up time out (see the Configuration Description), the sensor will send “BASIC SET” command to turn-off the lamp.

When sensor is triggered:

[Command Class Basic, Basic Set, Value = 0xFF (default 0xFF, configurable, see the Configuration Description)] When light-up time out:

[Command Class Basic, Basic Set, Value = 0x00]

Configuration Description

a) “Basic Set” configuration

If there is any device under Association Group2, the door sensor will send “Basic Set = value”

command to control that device when the door sensor is opened. “Value” configuration rule is as below:

Function	Parameter	Byte	Range	Default
Basic Set Level	1	1	1-100 or 0xFF	0xFF

b) Turn Off Light Time Configuration

If there is any device under Association Group2, the door sensor will send “Basic Set = value” command to Group2, and send “Basic Set = 0x00” command to turn-off light after “t” seconds, Set value = “t”, means to send Basic Set command after “t” seconds.

Function	Parameter	Byte	Range	Default
Turn Off Light Time	2	1	1-120	20

c) PIR Sensor Alarm Elimination Time Configuration

Min set time is 5s. If the configuration is 1, that means it will eliminate alarm after 1*5s (5 seconds), if t, will eliminate alarm after t*10s.

Function	Parameter	Byte	Range	Default
Alarm Elimination Time	3	1	1-120	20

Z-Wave Supportive Commands

Generic Deice Type = GENERIC_TYPE_SENSOR_BINARY

Specific Device Type = SPECIFIC_TYPE_ROUTING_SENSOR_BINARY

Support Command Class =

COMMAND_CLASS_ZWAVEPLUS_INFO_V2

COMMAND_CLASS_ASSOCIATION_V2

COMMAND_CLASS_WAKE_UP_V2

COMMAND_CLASS_BATTERY

COMMAND_CLASS_ZWAVEPLUS_INFO_V2

COMMAND_CLASS_ASSOCIATION_GRP_INFO

COMMAND_CLASS_NOTIFICATION_V4

COMMAND_CLASS_SENSOR_BINARY_V2

COMMAND_CLASS_MANUFACTURER_SPECIFIC_V2

COMMAND_CLASS_VERSION_V2

COMMAND_CLASS_POWERLEVEL

COMMAND_CLASS_DEVICE_RESET_LOCALLY

Commands to Control Other Devices: COMMAND_CLASS_BASIC