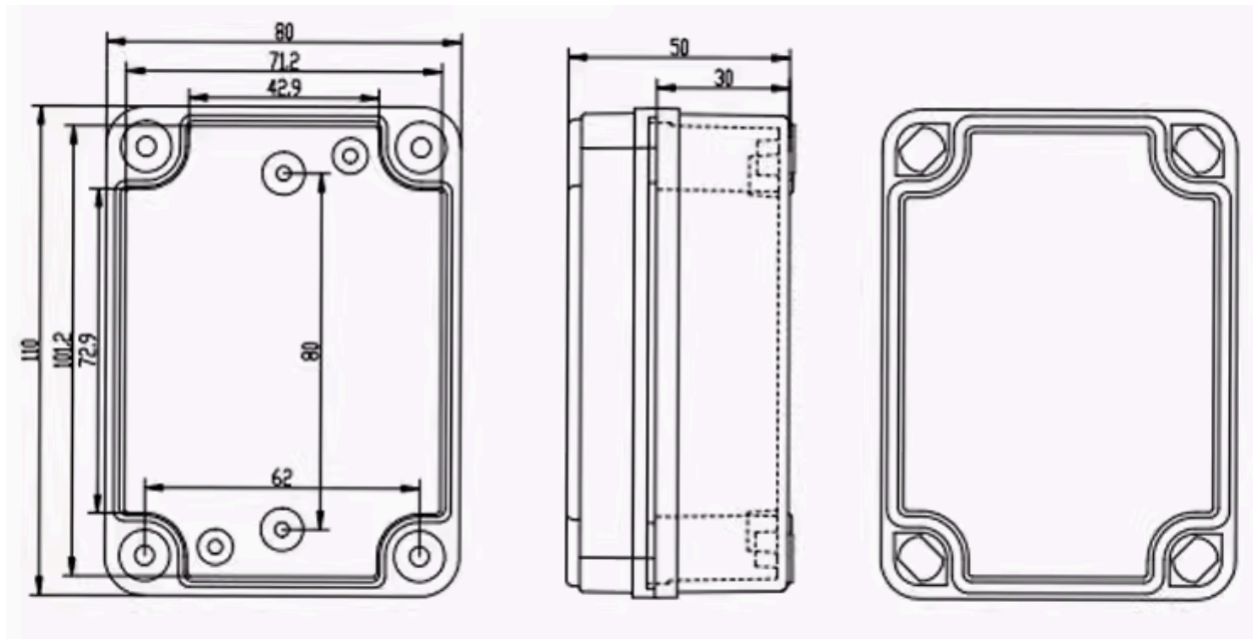




LoRaWAN Dry Contact Sensor v1.1

Housing Authority version (Customized)



Features

- Dry Contact Detection and Sampling Circuit
- Instant Handling on Dry Contact State Change
- Customizable Hold State Confirmation Period (Default 5s)
- Configurable Heartbeat Cycle
- Force Sampling by Magnet and On-board Force Test Button
- Design and Assembly in Hong Kong, local engineering support



INVISIBLE FUN STUDIO

Invisible Fun Studio Limited
608, Yuen Fat Industrial Building,
25 Wang Chiu Road, Kowloon Bay
Tel: 3565 6050 Fax: 3905 5659
Email: info@invisiblefun.com

Revision

Version	Date	Content	Editor
1.0	28 May 2024	Initial Release	Eric Fung
1.1	29 May 2024	Add Customized Version for Housing Authority	Eric Fung



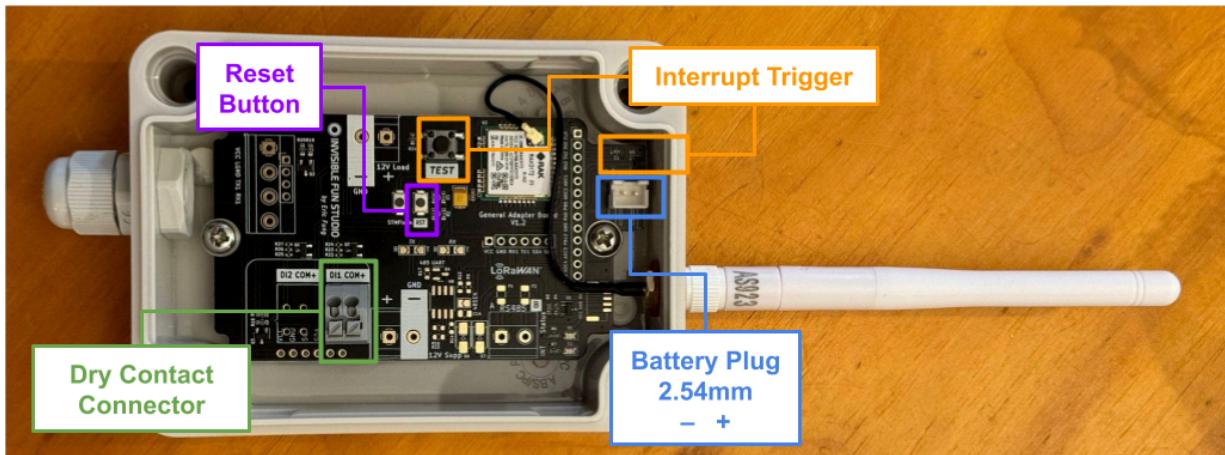
Specification

Reporting Data Type	
Battery Voltage	Unit: V
Dry Contact Logic Level	0 or 1, Active High
Uplink Type	0: Heartbeat or Force Sampling, 1: State Change
Force Sampling Trigger	
Magnet Trigger at Case Side	
Test Button On-Board	
LoRaWAN Interface	
Specification	1.0.3
Class	A, C (Default A)
Network Join Mode	OTAA, ABP (Default OTAA)
Configuration	LoRaWAN Downlink
Power Supply	
Rated Voltage	3.3 ~ 4.2V (Typ. 3.6V)
Battery Type	1* 8Ah 3.6V LifePo Battery
Battery Lifespan	2 years
Power Consumption	
Working Mode	Typical: 70mA Max 120mA @Tx
Sleep Mode	10uA
Routine Interval (Default)	1-hour Heartbeat
	5-Second Hold State Confirmation Period
Mechanical Features	
Operating Temperature	-20 ~ + 125°C
Relative Humidity	<= 85%RH
Dimension	110 *80 *50 mm (Reference only. Please refer to the delivery.)
Protection Level Rating	IP67

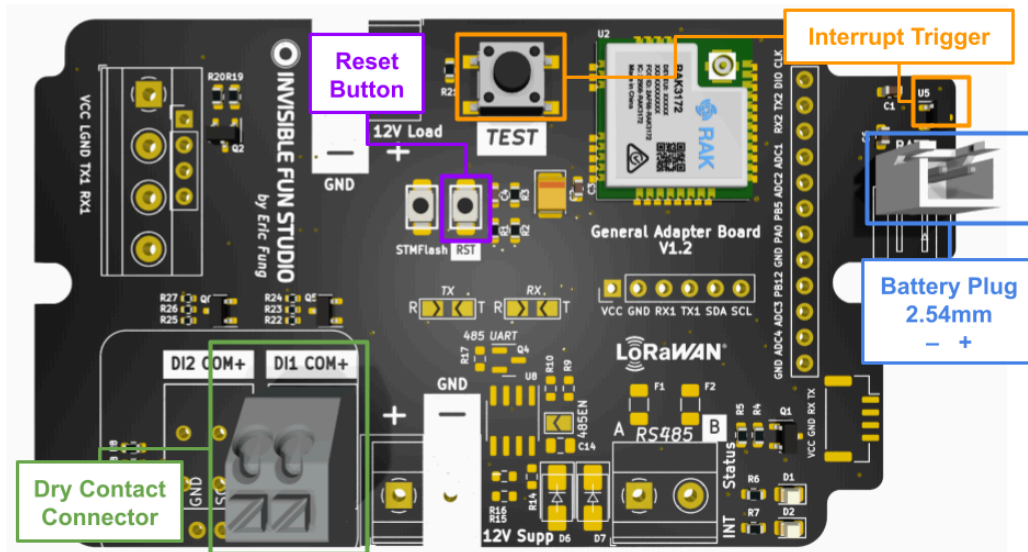


Hardware Interface

- Sensor Case



- PCB Main Features





Software Interface

- **LoRaWAN Specification**
 - a. Specification 1.0.3, Region AS923
 - b. Class A or C (Default Class A)
 - c. ABP or OTAA (Default OTAA)
- **Payload Inclusion**
 - a. Battery Voltage (Unit: V, 3d.p.)
 - b. Dry Contact Logic Level (Active High)
 - c. Uplink Type
(0: Heartbeat and Force Sampling, 1: State Change)

- Downlink List

Description	Downlink Format (HEX)	Value Range (HEX)	Default Value Dec (HEX)
Restart	1AFF	/	/
ADR On/Off	AD0XAD	0 or 1	0
Change Data Rate	ADA0XFF	2 to 5	2
Change Heartbeat Cycle (unit: second)	CDXXXXFF	0 to FFFF	1Hour: 3600 (0E10)
Change Heartbeat Cycle (unit: second) <u>Extended Range to support 1/2 years</u>	CDXXXXXXXXFF	0 to FFFFFFFF	1 Day: 86400 (015180)
Change Dry Contact Hold-State Confirmation Period to XX Second(s)	DCXXDC	0 to 255	5 (05)



- **Routine Behaviour**

- a. On dry contact channel state change, sensor hold n seconds (default 5) on tracking stable channel state. When the change is valid after the confirmation period, lorawan uplink will be sent.
- b. When Magnet Sensor is triggered at the side of case or On-board TEST button is pressed (**and released within 5 seconds**), green LED turns on and interrupt triggers. The sensor will force send an uplink for current channel state.
- c. If no valid state change appears within a cumulative heartbeat cycle (default 1 hour), the sensor will also force send an uplink.
- d. When Magnet Sensor is triggered at the side of case and holded **over 5 seconds**, the sensor will resend join request (if current Network Join Mode is OTAA).



LoRaWAN Payload Codec (compatible with Chirpstack v4)

```
// Decode uplink function.
//
// Input is an object with the following fields:
// - bytes = Byte array containing the uplink payload, e.g. [255, 230, 255, 0]
// - fPort = Uplink fPort.
// - variables = Object containing the configured device variables.
//
// Output must be an object with the following fields:
// - data = Object representing the decoded payload.
function decode_toHexString(bytes) {
  return bytes.map(function(byte) {
    return ("00" + (byte & 0xFF).toString(16)).slice(-2)
  }).join("")
}

function decodeUplink(input) {
  var tohex = decode_toHexString(input.bytes);
  var fPort = input.fPort;
  var return_json = {};
  if(fPort == 0) {return {data:return_json};}

  return_json["VCC"] = parseInt(tohex.slice(0,4), 16)/1000;
  if(fPort ==3)
    return_json["Uplink_Type"] = 1;
  else
    return_json["Uplink_Type"] = 0;
  return_json["Dry_Contact_Logic"] = parseInt(tohex.slice(5,6),16);
  return {data:return_json};
}

// Encode downlink function.
//
// Input is an object with the following fields:
// - data = Object representing the payload that must be encoded.
// - variables = Object containing the configured device variables.
//
// Output must be an object with the following fields:
// - bytes = Byte array containing the downlink payload.
function encodeDownlink(input) {
  return {
    bytes: [225, 230, 255, 0]
  };
}
```