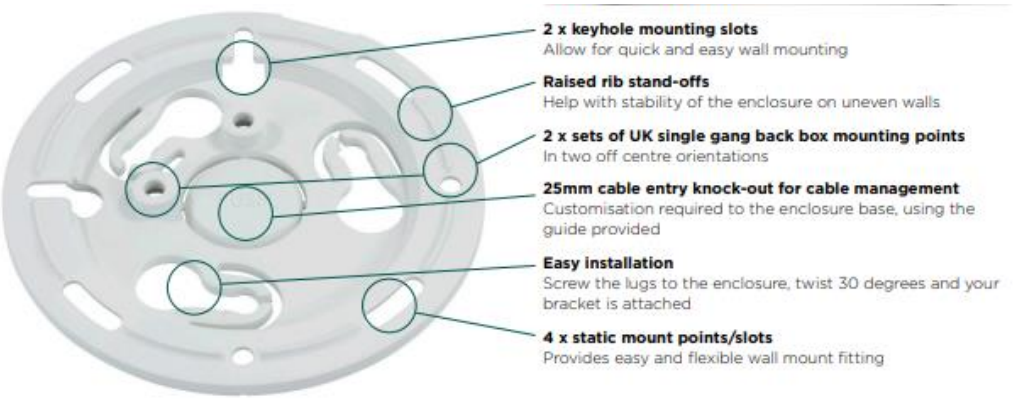
WISka Air LoRa Sensor

Item Specifications

Go-IoT Item Id:	WISka Air
Wireless Technology	LoRaWAN® 1.0.3
Wireless Security	LoRaWAN® End-to-End encryption (AES-CTR), Data Integrity Protection (AES-CMAC)
LoRaWAN Device Type	Class A/C (configurable) End-device. (Class B is optional)
Supported LoRaWAN® features	OTAA, ABP, ADR, Adaptive Channel Setup
Supported LoRaWAN® regions	EU433, CN470, IN865, EU868, AU915, US915, KR920, AS923 (Other Regions are optional)
Link Budget	131 dB (SF5) to 151 dB (SF12)
RF Transmit Power	14 dB / 22 dB (Region specific)
Antenna	Internal or External
Battery Type	3.6 V D Size Lithium Battery ER34615 19000mAh Li-$SOCL_2$ non-rechargeable 3.6 Volt
Battery Life	Up to 4 years (30mins reporting @ SF7)
Other PSUs	5VDC – 35VDC – 200mA , 9VAC – 25VAC – POE 802.11af
Sensors	
Temperature	Resolution: +-1 °C
Humidity	Resolution: 3 % RH
Air Quality VOC	Range: VOC Index 0 – 100
CO2	Range: 0 – 5000 ppm
Particulate Matter Sensor	Range: PM1.0, PM2.5, PM4, PM10
Battery Status	Range: 2.4VDC – 5.0VDC
Case Style Hex	146 x 130 x 45 mm (excluding optional wall mount bracket)
Case Color	ABS – White, Grey, Black

WISka Air Cases

Wall Mount Bracket



The WISka Air Case is made from 100% recyclable material with five or six vented panels and can be supplied in Black, Grey and White with vented wall measuring 100mm x 100mm x53.25mm. Supplied with a Universal Wall Mount bracket in the same colour that offers seamless and discrete mounting to retain a modern aesthetic look.

1 Installation

The WISka Air can be powered by the following: -

- 19Ahr Internal Lithium Non rechargeable Battery
- 6-30V DC Supply
- 6-24V AC Supply
- POE Cable

1.1 Connecting the 3.6V Internal Lithium Battery

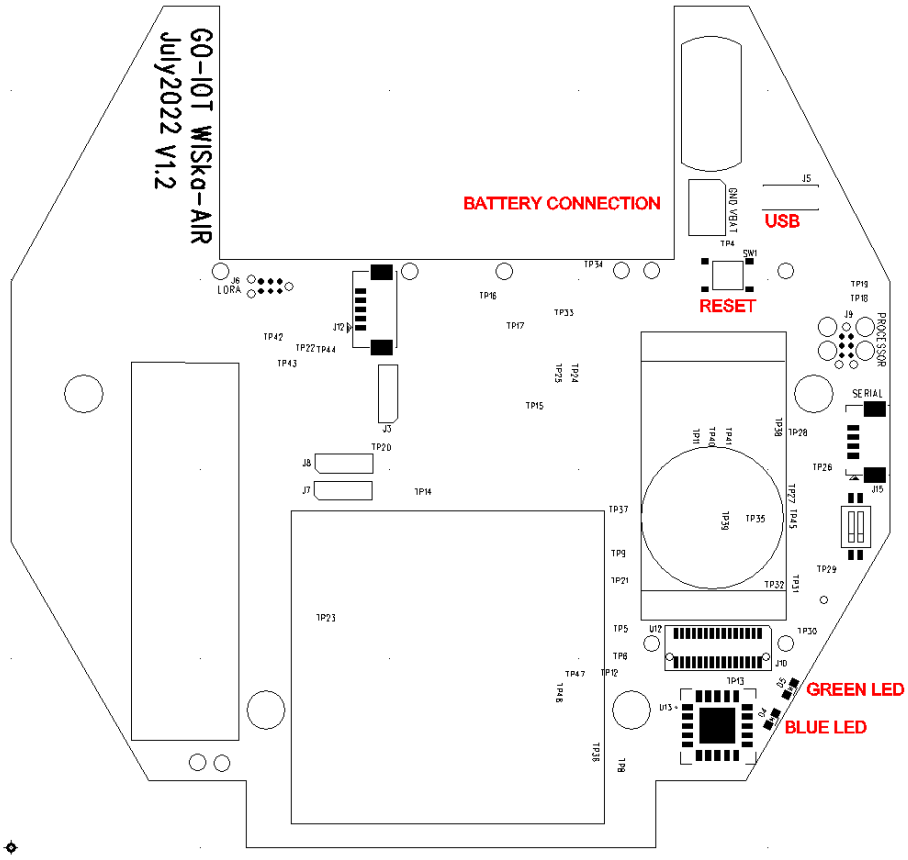
Ensure the LoRa Gateway and LoRa Packet Forwarders have been programmed and powered up

Connection the 2-way battery cable. The connector is polarized. Press Reset Switch for 1 second.

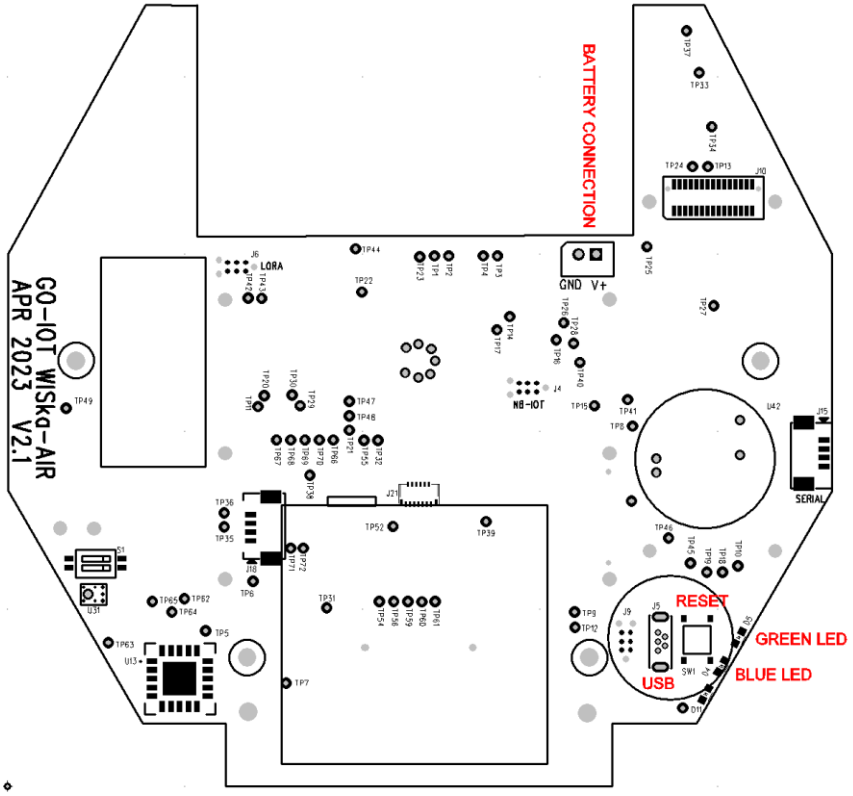
LED Operation

- WISka Air is working correctly (power and memory) - single Blue flash (5 seconds after reset)
- Connecting to LoRaWAN - single Blue flash (12 seconds after reset)
- Connected to LoRA Server – Green LED ON for 1 second

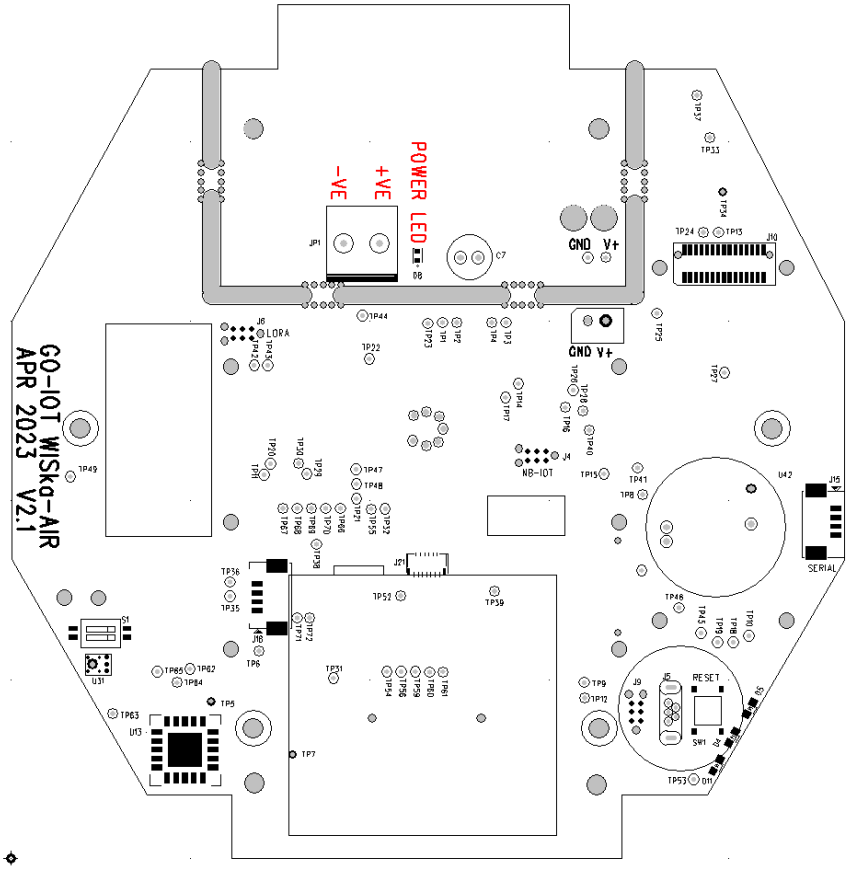
WISka Air Version 1.2



WISka Air Version 2.1



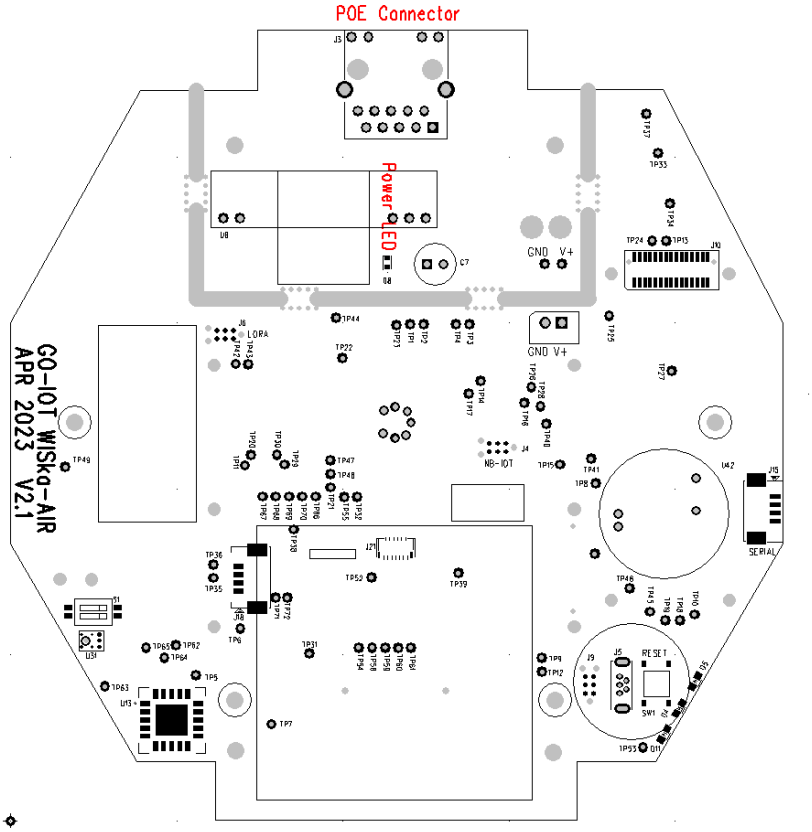
Connecting 6-30V DC Supply



Connecting 6-24V AC Supply

As above – There is no polarity on the Power Connections

Connecting POE Cable



1.2 Wall Mounting

Screw the mounting nuts to the rear of the WISka Air Backplate.

Screw the WISka Air mounting bracket to the wall using your preferred mounting holes.

See YouTube video - https://www.youtube.com/watch?v=57sSa11qq3g

LoRa Payload Decoder

This will be available via an attachment on the email

```
//   __        _____ ____  _  __    _
//   \ \      / /_ _/ ___|| |/ /   / \
//    \ \ /\ / / | |\___ \| ' /   / _ \
//     \ V  V /  | | ___) | . \  / ___ \
//      \_/\_/  |___|____/|_|\_\/_/   \_\
//
//
//  WISKA simple payload decoder.
//  www.go-iot.io
//  Version 1.5  December 2022
//  TYPE_MODEL & TYPE_SENSORS added
//  V1.4 PULSE 3 & PULSE 4 ADDED
//  V1.5 var TYPE_PULSE1_KW_ABS   = 0xAC;  //2bytes Absolute Running Total
//       var TYPE_PULSE1_KW       = 0xAD;  //4bytes Count since last uplink  (was 2Bytes)
//       var TYPE_PULSE2_KW_ABS   = 0xAE;  //2bytes Absolute Running Total
//       var TYPE_PULSE2_KW       = 0xAD;  //4bytes Count since last uplink  (was 2Bytes)
//
//  January 26th Jan 2023
//  V1.6 Pulse Inputs renamed and renumbered
//  var TYPE_PULSE1          = 0x8A;  RENAMED to  var TYPE_PULSE5          = 0x8A
//  var TYPE_PULSE1_ABS      = 0x8B;  RENAMED to  var TYPE_PULSE5_ABS      = 0x8B
//  var TYPE_PULSE2          = 0x96;  RENAMED to  var TYPE_PULSE6          = 0X96
//  var TYPE_PULSE2_ABS      = 0x97;  RENAMED to  var TYPE_PULSE6          = 0X97
//  var TYPE_PULSE1_KW       = 0xAB;  RENAMED to  var TYPE_PULSE1          = 0xAB
//  var TYPE_PULSE1_KW_ABS   = 0xAC;  RENAMED to  var TYPE_PULSE1_ABS      = 0xAC
//  var TYPE_PULSE2_KW       = 0xAD;  RENAMED to  var TYPE_PULSE2          = 0xAD
//  var TYPE_PULSE2_KW_ABS   = 0xAE;  RENAMED to  var TYPE_PULSE2_ABS      = 0xAE
//
// IN PROGRESS:
// ==============================================================================
// February 4th 2023
// Those items are already  removed
//  var TYPE_PULSE1_KW       = 0xAB;  RENAMED to  var TYPE_PULSE1          = 0xAB
//  var TYPE_PULSE1_KW_ABS   = 0xAC;  RENAMED to  var TYPE_PULSE1_ABS      = 0xAC
//  var TYPE_PULSE2_KW       = 0xAD;  RENAMED to  var TYPE_PULSE2          = 0xAD
//  var TYPE_PULSE2_KW_ABS   = 0xAE;  RENAMED to  var TYPE_PULSE2_ABS      = 0xAE

//  Reduced Decoder just for WISka AIR


    var TYPE_TEMP            = 0x81;  //temp 2 bytes -3276.8°C -->3276.7°C
    var TYPE_RH              = 0x82;  //Humidity 1 byte  0-100%
    var TYPE_CO2             = 0x86;  //Co2 2 bytes 0-65535 ppm
    var TYPE_VDD             = 0x87;  //Internal Battery Voltage 2byte 0-65535mV
    var TYPE_PRESSURE        = 0x94;  //4byte pressure data (hPa)
    var TYPE_CONC_PM0_5      = 0xB0;  //2bytes Number Concentration PM0.5 [#/cm³]
    var TYPE_CONC_PM1_0      = 0xB2;  //2bytes Number Concentration PM1.0 [#/cm³]
    var TYPE_MASS_CONC_PM1_0 = 0xB3;  //2bytes Mass Concentration PM1.0 [µg/m³]
    var TYPE_CONC_PM2_5      = 0xB4;  //2bytes Number Concentration PM2.5 [#/cm³]
    var TYPE_MASS_CONC_PM2_5 = 0xB5;  //2bytes Mass Concentration PM2.5 [µg/m³]
    var TYPE_CONC_PM4_0      = 0xB6;  //2bytes Number Concentration PM4.0 [#/cm³]
    var TYPE_MASS_CONC_PM4_0 = 0xB7;  //2bytes Mass Concentration PM4.0 [µg/m³]
    var TYPE_CONC_PM10       = 0xB8;  //2bytes Number Concentration PM10 [#/cm³]
    var TYPE_MASS_CONC_PM10  = 0xB9;  //2bytes Mass Concentration PM10 [µg/m³]
    var TYPE_CONC_PM5_0      = 0xBB;  //2bytes Number Concentration PM5.0 [#/cm³]
    var TYPE_MASS_CONC_PM5_0 = 0xBC;  //2bytes Mass Concentration PM5.0 [µg/m³]
    var TYPE_VOC_RAW         = 0xC0;  //2bytes (ppm)  RAW
    var TYPE_VOC_INDEX       = 0xC1;  //2bytes VOC Index
    var TYPE_VOC_INDEX_MIN   = 0xC2;  //2bytes VOC Index min
    var TYPE_VOC_INDEX_MAX   = 0xC3;  //2bytes VOC Index max
    var TYPE_MODEL           = 0xED;  //12bytes - Sensor Model and Firmware
    var TYPE_SENSORS         = 0xEE;  //20bytes - Installed Sensors
    var TYPE_DEBUG           = 0xEF;  //4bytes debug EE
```

```
function bin16dec(bin) {
    var num=bin&0xFFFF;
    if (0x8000 & num)
        num = - (0x010000 - num);
    return num;
}
function bin8dec(bin) {
    var num1=bin&0xFF;
    if (0x80 & num1)
        num1 = - (0x0100 - num1);
    return num1;
}
function hexToBytes(hex) {
    for (var bytes = [], c = 0; c < hex.length; c += 2)
        bytes.push(parseInt(hex.substr(c, 2), 16));
    return bytes;
}
var decodeWiskaPayload=function(data,port){
    var obj = new Object();
    for(i=0;i<data.length;i++){
        //console.log(data[i]);
        switch(data[i]){
            case TYPE_TEMP: //Temperature
                var temp=(data[i+1]<<8)|(data[i+2]);
                temp=bin16dec(temp);
                obj.temperature=temp/10;
                i+=2;
                break
            case TYPE_RH: //Humidity
                var rh=(data[i+1]);
                obj.humidity=rh;
                i+=1;
                break
            case TYPE_CO2: //CO2
                obj.co2=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_VDD: //Battery level
                obj.vdd=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_CONC_PM0_5: //Number Concentration PM0.5 [#/cm³]
                obj.concPM0_5=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_CONC_PM1_0: //Number Concentration PM1.0 [#/cm³]
                obj.concPM1_0=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_MASS_CONC_PM1_0: //Mass Concentration PM1.0 [µg/m³]
                obj.massConcPM1_0=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_CONC_PM2_5: //Number Concentration PM2.5 [#/cm³]
                obj.concPM2_5=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_MASS_CONC_PM2_5: //Mass Concentration PM2.5 [µg/m³]
                obj.massConcPM2_5=(data[i+1]<<8)|(data[i+2]);
                i+=2;
                break
            case TYPE_CONC_PM10: //Number Concentration PM10 [#/cm³]
                obj.concPM10=(data[i+1]<<8)|(data[i+2]);
                i+=2;
```

```
                    break
                case TYPE_MASS_CONC_PM10: //Mass Concentration PM10 [µg/m³]
                    obj.massConcPM10=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                    break
                case TYPE_CONC_PM5_0: //Number Concentration PM5.0 [#/cm³]
                    obj.concPM5_0=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                    break
                case TYPE_MASS_CONC_PM5_0: //Mass Concentration PM5.0 [µg/m³]
                    obj.massConcPM5_0=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                case TYPE_VOC_RAW: //VOC
                    obj.voc_raw=(data[i+1]<<8)|(data[i+2]);      //name changed
                    i+=2;
                    break
                case TYPE_VOC_INDEX: //VOC Index
                    obj.voc_index=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                    break
                case TYPE_VOC_INDEX_MIN: //VOC Index Min
                    obj.voc_index_min=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                    break
                case TYPE_VOC_INDEX_MAX: // VOC Index Max
                    obj.voc_index_max=(data[i+1]<<8)|(data[i+2]);
                    i+=2;
                    break
                case TYPE_MODEL:
                    obj.model = [];
                    for(var j = 0; j < 20; j++)  {
                        obj.model[j] = (data[j]);       // C
                    }
                    i += 20;
                    break
                case TYPE_SENSORS:
                    obj.sensors = [];
                    for(var j = 0; j < 20; j++)  {
                        obj.sensors[j] = (data[j]);     // New
                    }
                    i += 20;
                    break
                default: //somthing is wrong with data
                    i=data.length;
                    break
            }
        }
        return obj;
    }

exports.decode = function(bytes,port)
{
    return decodeWiskaPayload(bytes,port);
}

```

WISka Air downlink payload

Downlink payloads are sent on the configured LoRa port + 1.

If configured port is the default (5) then downlink settings should be sent on port 6.

Header byte (0xA8)	**Payload length**	**Settings data**	**Default**	**Settings data**
1 byte	**1 bytes**	n bytes		n bytes

0x41	Sensor Time Base	2 byte time (seconds)	900	4 Bytes HEX
0x42	Send Time	2 byte period	1	4 Bytes HEX
0x43	Temperature	2 byte period	1	4 Bytes HEX
0x47	CO2	2 byte period	1	4 Bytes HEX
0x48	VOC	2 byte period	1	4 Bytes HEX
0x4B	Battery	2 byte period	1	4 Bytes HEX
0x4F	Particle Matter	2 byte period	1	4 Bytes HEX

Example 1

The default uplink time is 900 seconds – 15 minutes

The LoRa Uplink Port = 05

Therefore, to change the uplink time to 1200 seconds – 20 minutes

Send a Downlink to Port 6

A803410480

Where A8 = Header Byte
03 = Number of bytes following in settings data
41 = Sensor Time Base
0480 = 1200 in hex

Example 2

The default setting for the uplink period for the Temperature Payload is (1)

X

Sensor Time Base (900 seconds) X Send Time Period (1) = 900seconds

Therefore, a Temperature Payload is uplinked every 900 seconds (15minutes)

If this was required to be changed from 1 to 2 (every 1800 seconds – 30minutes)

Send a Downlink to Port 6

A803430002

Where A8 = Header Byte
03 = Number of bytes following in settings data
43 = Temperature Uplink Period
0002 = 2