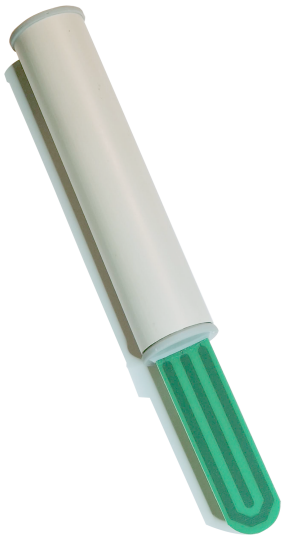
FEATURES

- **Universdal LoRaWAN module. Supported frequencies "AS923", "AU915", "EU868", "KR920", "IN865", "US915", "RU864".** Switch in the Android App.
- **SINGLE-1** solid or separate single/double on the cable **soil moisture and temperature** sensor
- **MULTI-6 Soil Moisture Multi sensor 6 part moisture/3 temp. sensors**
- **LEAF-1 moisture/temperature** sensor(solid, or separate on the cable)
- Combination of soil and leaf moisture sensors
- LoRaWAN v 1.0.3 class A device, may be activated with OTAA or ABP.
- Setup via BLE Bluetooth module(**sold separately**) and Android APP or COM TTL module AT commands.
- Dust and waterproof IP68
- Ultra-low 1 uA sleep current, operates from 2xAA type batteries(not included)

SINGLE-1



LEAF-1

DEVICE ELECTRICAL PROPERTIES

	Min/Sleep	Typical	Max
Supply voltage (VCC), V	2.0	3.0	3.6
Working current（VCC=3.6V）, mA	Sleep 1uA	20	150
Operating Temperature Range(not applicable to batteries), Celsius	-20	25	70

The device sleep mode consumes 1uA or 0.001 mA. Active mode consumption is 12-24 mA depending on sensor configuration and 150 mA in transmitting mode.
The required battery is **two 1.5 AA-type batteries**(not included)**.**
For device setup, it is required to attach an additional Bluetooth BLE module, which is operated by an Android application.
The device uses the RAK3172 module. Certificates are available at the location:
https://downloads.rakwireless.com/LoRa/RAK3172/Certification/

PHYSICAL PROPERTIES

SINGLE-1. The device's overall dimensions with the single(built-in) sensor, mm 230*35*35, sensor dimension 80*25
MULTI-6 The device's overall dimensions with the MULT-6 6 depth sensor, mm 730*35*35
Wired sensors cable length(for the sensor with cable variation) 1.4m; 2.9m; 4.9m

SOIL SENSOR MEASUREMENT PROPERTIES

Soil Sensor	**Resolution**	**Range /avg Tolerance**
Dielectric permittivity (ε) (Temperature corrected)	0.1ε	1 (air) to 80 (water) /5%
Volumetric water content - VWC calculation from Dielectric permittivity ε. VWC = 0.002974 * pow(ε,2) + 0.07424 * ε -1.295;		
Temperature (°C)	0.1°C	-20 to 70°C/3%

Degree of water saturation in the soil	0.1%	0 – 100% /8%

DATA OUTPUT FORMAT

The format that you are usually getting from LoRaWAN is a hex string, which you have to convert to bytes. Data output payload is variable length. To properly parse data you have to decode bits in 1st byte::

1. 1-byte setup bits: 0 - disabled
2. 1-byte setup bits: 32 - enabled multi sensory mode
3. 1-byte Battery voltage 254 - max

--Soil sensor output:

4. 2-byte 16-bit signed integer (divide by 100) Dielectric permittivity (ε) (Temperature corrected) resolution: 0.1ε (avg.Tolerance 5%) and range 1 (air) to 80 (water)
5. 2-byte 16-bit signed integer (divide by 10) Electrical Conductivity (mS/m) resolution: 0.01 mS/m (avg.Tolerance 20%)
6. 2-byte 16-bit signed integer (divide by 100) Temperature (°C) resolution: 0.1°C and range: -20 to 60°C (avg.Tolerance 3%)
7. 2-byte 16-bit signed integer Volumetric Water Content (%) resolution:1% and range: 0 – 100% VWC Note: VWC is calculated from dielectric permittivity by Topp equation (Topp et al, 1980): $\theta = 4.3 \cdot 10{-}6\varepsilon 3 - 0.00055\varepsilon 2 + 0.0292\varepsilon - 0.053$)

--Multisensor output:

1. 1-byte version [1]
2. **6x** 2-byte 16-bit signed integer (divide by 100) Dielectric permittivity (ε) (Temperature corrected) resolution: 0.1ε (avg.Tolerance 5%) and range 1 (air) to 80 (water)
3. **3x** 2-byte 16-bit signed integer (divide by 100) Temperature (°C) resolution: 0.1°C and range: -20 to 60°C (avg.Tolerance 3%)

The Things Network compatible payload decoder

```
var bytesToFloat32 = function(/*byte[]*/byteArray) {
    var buf = new ArrayBuffer(4);
    var view = new DataView(buf);

    // set bytes
    byteArray.forEach(function (b, i) {
        view.setUint8(i, b);
    });
    return view.getFloat32(0);
};
```

```
var bytesToInt = function(/*byte[]*/byteArray, dev) {
    var value = 0;
    for ( var i = 0; i < byteArray.length; i++) {
        value = (value * 256) + byteArray[i];
    }
    return value/dev;
};
var bytesToSignedInt = function(bytes, dev) {
  var sign = bytes[0] & (1 << 7);
  var x = ((bytes[0] & 0xFF) << 8) | (bytes[1] & 0xFF);
  if (sign) {
  x = 0xFFFF0000 | x;
  }
  return x/dev;
};
function decodeUplink(input) {
    var bytes = input.bytes;
    var decoded = {};
    var pos = 1;
    decoded.valv=((bytes[0] >> 7) & 1);
    decoded.leak=((bytes[0] >> 6) & 1);
    decoded.bat = bytes[pos++];
    if(((bytes[0] >> 0) & 1)===1){ //SOIL
        decoded.e25=bytesToInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        decoded.ec=bytesToInt(bytes.slice(pos,pos+2),10);
        pos = pos+2;

decoded.temp=bytesToSignedInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        decoded.vwc=bytesToInt(bytes.slice(pos,pos+2),1);
        pos = pos+2;
    }
    if(((bytes[0] >> 1) & 1)===1){ //BME

decoded.airTemp=bytesToSignedInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        decoded.airHum=bytesToInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        var airPressuse =
bytesToInt(bytes.slice(pos,pos+2),1)+50000;
        if(airPressuse!==65536){
        decoded.airPres=airPressuse;
        }
        pos = pos+2;
    }
    if(((bytes[0] >> 2) & 1)===1){ //OPT
        decoded.lux=bytesToInt(bytes.slice(pos,pos+4),100);
        pos = pos+4;
    }
    if(((bytes[0] >> 4) & 1)===1){ //PULSE
        decoded.pulse=bytesToInt(bytes.slice(pos,pos+4),1);
        pos = pos+4;
```

```
    }
    if(((bytes[0] >> 3) & 1)===1){ //SOIL
        decoded.e25_1=bytesToInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        decoded.ec_1=bytesToInt(bytes.slice(pos,pos+2),10);
        pos = pos+2;

decoded.temp_1=bytesToSignedInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        decoded.vwc_1=bytesToInt(bytes.slice(pos,pos+2),1);
        pos = pos+2;
    }
    if(((bytes[0] >> 5) & 1)===1){ //PRESSURE
        decoded.press=bytesToInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
    }
    var set1 = bytes[pos++];
    if(((set1 >> 1) & 1)===1){ //LEAF
        decoded.leafHum=bytesToInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;

decoded.leafTemp=bytesToSignedInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
    }
    if(((set1 >> 2) & 1)===1){ //ADC
        decoded.adc=bytesToFloat32(bytes.slice(pos,pos+4));
        pos = pos+4;
    }
    if(((set1 >> 3) & 1)===1){ //WIND
        decoded.windDir = bytes[pos++];

decoded.windSpeed=bytesToFloat32(bytes.slice(pos,pos+4));
        pos = pos+4;
    }
    if(((set1 >> 4) & 1)===1){ //SCALE
        decoded.scale=bytesToInt(bytes.slice(pos,pos+4),10);
        pos = pos+4;
    }
    if(((set1 >> 5) & 1)===1){ //BIT1_SOIL_MULTI
        var mult_v=bytes[pos++];//ver
        for(var c=0;c < 6;c++){
        var dp = bytesToInt(bytes.slice(pos,pos+2),100);
        decoded["dp"+(6-c)] = dp;
        pos = pos+2;
        decoded["wvc"+(6-c)] = Math.round((4.3e-6 *
Math.pow(dp,3) - 5.5e-4*Math.pow(dp,2) + 2.92e-2*dp -
5.3e-2)*10000)/100;
        }
        for(var i = 0;i < 3;i++){
        decoded["temp"+(3-i)] =
bytesToSignedInt(bytes.slice(pos,pos+2),100);
        pos = pos+2;
        }
```

```
    }
  return {
    data: decoded,
    warnings: [],
    errors: []
  };
}
```

COMMAND LINE

LoRaWAN configuration may be done via serial TTL adapter connected to the board pin headers or using plugin Bluetooth BLE module connection to android application. The serial communication speed is 9600.

COMMAND LINE DEVICE CONFIGURATION COMMANDS

AT? to list all available functions
AT+<CMD>? : Help on <CMD>
AT+<CMD> : Run <CMD>
AT+<CMD>=<value> : Set the value
AT+<CMD>=? : Get the value
ATZ Trig a MCU reset
AT+VL=<Level><CR>. Set the Verbose Level=[0:Off .. 3:High]
AT+APPEUI=<XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the App Eui
AT+NWKKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>: Get or Set the Network Key
AT+APPKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>: Get or Set the Application Key
AT+NWKSKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>: Get or Set the Network Session Key
AT+APPSKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>: Get or Set the Application Session Key
AT+DADDR=<XXXXXXXX><CR>. Get or Set the Device address
AT+DEUI=<XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the Device EUI
AT+NWKID=<NwkID><CR>. Get or Set the Network ID=[0..127]
AT+JOIN=<Mode><CR>. Joinnetwork with Mode=[0:ABP, 1:OTAA]
AT+LINKC. Piggyback a Link Check Request to the next uplink
AT+BAT Get the battery Level in mV
AT+int=<XXXX> set sleep time seconds
AT+stime=<XXXX> set sensor on time millis
AT+air=2 calibrate soil sensor in air <sensor id 0,1,2-multi>
AT+water=2 calibrate soil sensor in water <sensor id 0,1,2-multi>

AT+rescal=2 reset calibrate soil sensor <sensor id 0,1,2-multi>
AT+set=0:32 enter settings bytes 0 0
AT+jtype=<X> joint type 0 - none 1-ABP 2-OTAA 3-Local Gateway
AT+csv read settings
AT+lora read lora settings
AT+defaults reset default settings
AT+confirmed uplink confirmed=1 ; unconfirmed=0

DOWNLINK MESSAGE

Device supports LoRaWAN downlink messages on port 1. Downlink message contains 1st command byte followed by value:

1. 0x01 + device sleep time (data sending interval) adjustment. The message should contain 5 bytes, 1st byte is 0x01, and the following 4 bytes are unsigned 32-bit integer time in seconds to sleep. For example, sending HEX 010000000A will set the sleep interval to 10 seconds, but 0100000E10 will set the sleep interval to 3600 secs = 1 hour.
2. 0x020020 - setup
3. 0x0A02 - calibrate AIR
4. 0x0B02 - calibrate WATER

SETUP

The sensors are already factory calibrated in air and water, but in case needed they may be recalibrated using the USB terminal interface as described for the device-specific commands.
Hold a dry sensor in the air and issue this command **air**, response OK.
Put the sensor into water and issue command **water**, response OK.
For the sensors with EC measuring function. For **ec** <uS/m> calibration put the sensor int to liquid and set the right sensor reading value in uS/cm.

We recommend an Android mobile application for sensor setup and data reading/storage

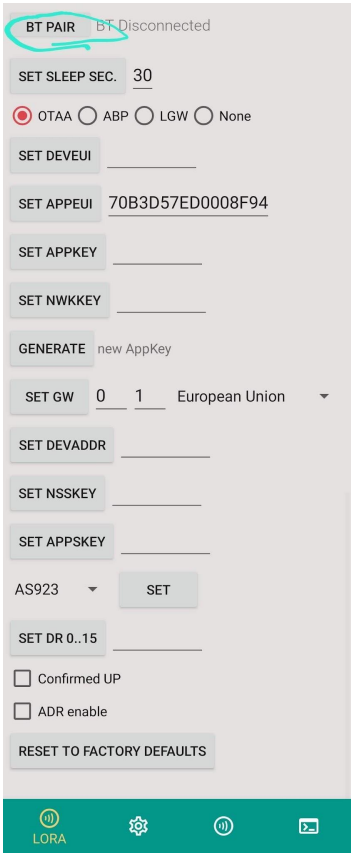
SOIL SENSOR CALIBRATION ON ANDROID APP

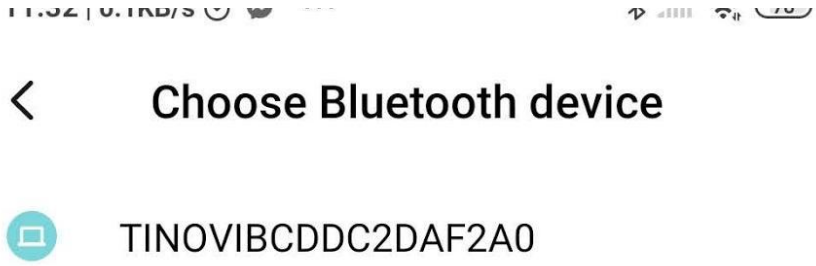
[Tinovi LW Configurator - Apps on Google Play]

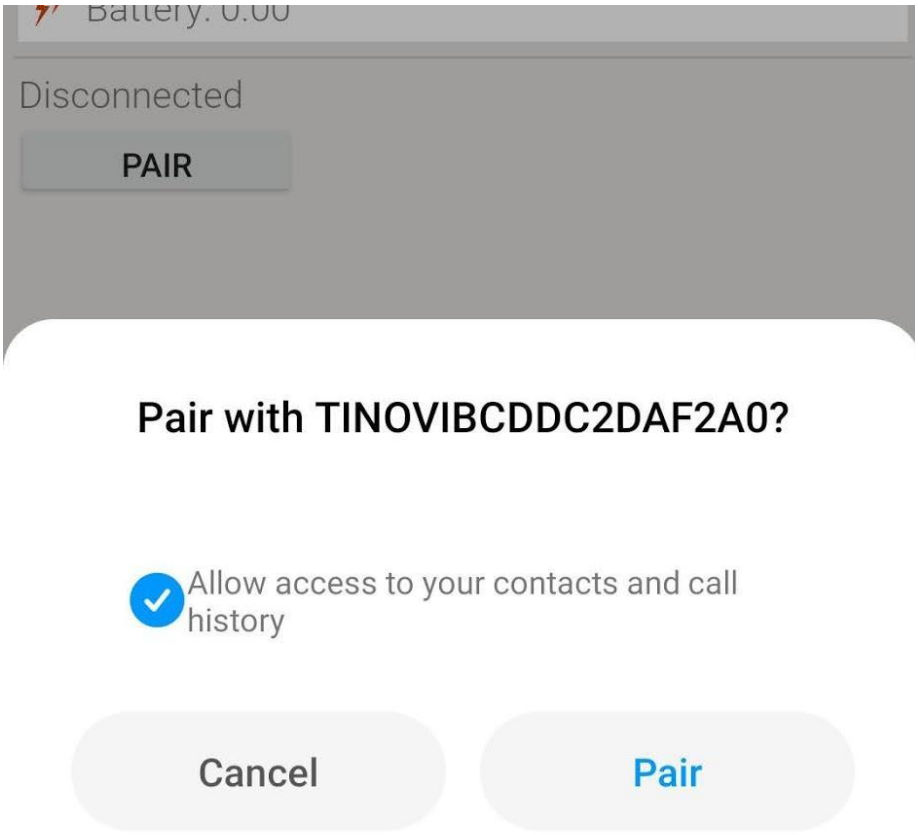
Android application lets you configure and calibrate soil moisture sensors for your device.

1. Your phone should support Bluetooth..
2. Turn on Bluetooth on your phone
3. Connect device setup - BLE Bluetooth programming module.

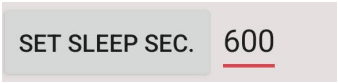
4. Ensure there are batteries plugged into your device. The batteries shall have a voltage of 1.5V to ensure the operation of the BLE Bluetooth module. We recommend using new batteries to calibrate sensors.
5. Restart the device by pressing the restart button on the device's main board or removing and replacing the battery.
6. Open the Android App and press the BT PAIR button and choose TNXXXXXXXXXXXXX device (pair with the BLE Bluetooth module which is attached to your device (if adding for the first time)).

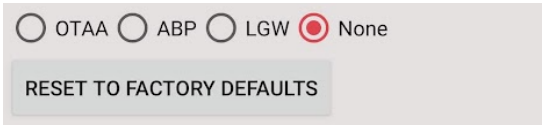

BT PAIR
BT Disconnected
SET SLEEP SEC.
30
OTAA
ABP
LGW
None
SET DEVEUI
SET APPEUI
70B3D57ED0008F94
SET APPKEY
SET NWKKEY
GENERATE
new AppKey
SET GW
0
1
European Union
SET DEVADDR
SET NSSKEY
SET APPSKEY
AS923
SET
SET DR 0..15
Confirmed UP
ADR enable
RESET TO FACTORY DEFAULTS
LORA


Choose Bluetooth device
TINOVIBCDDC2DAF2A0



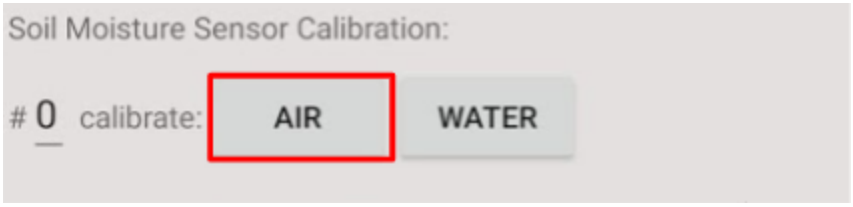
---

7. Wait up to 5-10 seconds while the application shows Connected TNXXXXXXXXXX on the main page, if no connection, repeat restart device and pair operation. The device may be restarted by removing and replacing the battery or clicking the button on the PCB board's left bottom corner when the battery looks upwards. Please don't use discharged batteries with the BLE Bluetooth module attached to the device.
8. If you have an interrupting the BLE Bluetooth connection, please set the Sleep time longer (So that you have time to make any necessary changes to the device settings, such as 600 seconds). 

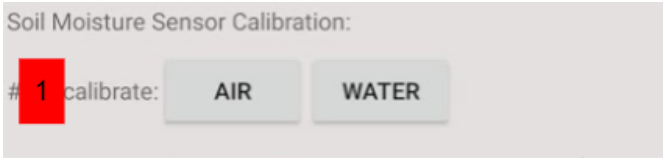
Alternatively, switch the device to "None" mode immediately after you connect to Bluetooth. 

CALIBRATION

1. Connect device setup - BLE Bluetooth programming module with the Android App. (see this manual above)
2. Go to the App SETTINGS tab.
3. Hold the dry sensor in the air – click the button AIR



4. Go to the MAIN tab and click READ to see values after calibration
5. Submerge the sensor(just the sensing part, not the whole device) in the water or soil with water, and click the button WATER
6. Go to the MAIN tab and click READ to see values after calibration
7. To calibrate the second soil moisture sensor, type number 1 in the red mark field picture below and repeat points 2 to 6



8. Go to the MAIN tab and click READ to see values after calibration