MACNMAN



Wi-Fi Temperature & Humidity Datalogger

MS-WBSD-TH-X1

USER GUIDE

Hardware V1.0.0 | Firmware V1.0.0 | Maya App V3.3.4+20 | Guide V1.0 | October 2026

Contents

Chapter 1. Preface **3**
Copyright Statement 3
Safety Instruction 3
Revision History 3

Chapter 2. Product Introduction **4**
Overview 4
Features 4
Product Limitations 4

Chapter 3. Hardware Introduction **5**
Packing List 5
Hardware Overview 5
Dimensions (mm) 6
LED Indicator 6

Chapter 4. Quick Start **7**

Chapter 5. Operation Guide **8**
The Device Screen 8
Wi-Fi Settings 9
MQTT Settings 10
General Setting 12
Data Export (Reminisce) 12
Time Synchronization 14
Device Data 14
Network Test (Ping) 15
Threshold Setting (Trigger) 16
Sampling Setting 17
System Information 18
Updates 18

Chapter 6. Installation **20**
Installation Location 20
Wall Mounting 21
Pole Mounting 21

Chapter 7. Battery Information **22**

Chapter 8. Uplink and Downlink **24**
Overview 24
Uplink Data 25
Downlink Command 26

Chapter 9. Services **34**

Chapter 10. Coming Soon **35**

Chapter 1. Preface

Copyright Statement

This guide may not be reproduced in any form or by any means to create any derivative such as translation, transformation or adaptation without the prior written permission of Macnman Technologies Pvt Ltd (hereinafter referred to as Macnman).

Macnman reserves the right to change this guide and the specifications without prior notice. The latest specifications and user documentation for all Macnman products are available on our official website www.macnman.com.

Safety Instruction

These instructions are intended to ensure that users can use the product correctly and avoid danger or loss of property. Macnman will not be responsible for any loss or damage resulting from not following the instructions of this guide.

⚠ **CAUTION:**

Injury or equipment damage may be caused if any of these cautions are neglected.

- The device is not intended to be used as a reference instrument. Macnman will not be responsible for any damage which may result from inaccurate readings.
- The device must not be disassembled or modified in any way. The battery cannot be replaced by the user.
- Do not place the device close to naked flames or other heat sources.
- Do not use the device where the temperature is below or above its operating range.
- The device contains a 3.6 V Li-SOCl2 battery. Never recharge, short-circuit, crush, open or heat the device or its battery.
- The device must never be subjected to strong shocks or impacts.
- The device stores your Wi-Fi password and MQTT login. Keep them private and do not put them in shared documents or screenshots.

Revision History

Release Date	Version	Revision Content
October 1, 2026	V1.0	First release, for hardware V1.0.0, firmware V1.0.0 and Maya app V3.3.4+20

Chapter 2. Product Introduction

Overview

The MacSync MS-WBSD-TH-X1 is a compact, battery-powered Wi-Fi datalogger that measures air temperature and relative humidity with a Sensirion SHT40 sensor and publishes the readings over your existing Wi-Fi network straight to your MQTT broker - no gateway is needed. It saves every reading in its own memory before sending it, and runs for years on one 19 Ah lithium cell.

The device is set up with the Macnman Maya app, which connects to it over Bluetooth Low Energy. Through the app, users connect the device to Wi-Fi and to the MQTT broker, set the reporting interval, alarm thresholds and batch sampling, test the connection on the spot, view live readings, export the stored readings and update the firmware. The heartbeat, triggers and sampling can also be changed remotely with MQTT commands, and the firmware can be updated over the internet.

The MS-WBSD-TH-X1 is widely used for cold storage and warehousing, smart buildings, server rooms, greenhouses, industrial sites and environmental monitoring.

Features

- Sensirion SHT40 digital temperature and humidity sensor, factory calibrated
- Dual-band Wi-Fi 6 (2.4 GHz and 5 GHz): works with Wi-Fi 4, 5 and 6 routers secured with WPA2-Personal or WPA3-Personal - no gateway needed
- Standard MQTT 3.1.1 with QoS 1, over plain TCP (no TLS): JSON messages to any standard MQTT broker or private cloud
- Every reading is saved on the device before it is sent - up to 1,550,400 readings - and can be exported to a phone with Reminisce
- Built-in 3.6 V, 19 Ah Li-SOCl2 battery: more than 10 years at the default 1-hour reporting interval with a good Wi-Fi signal (Chapter 7)
- Rugged IP65 enclosure for indoor and outdoor use, -30 °C to +80 °C
- Two independent threshold triggers; each sends one ALARM when the value leaves its range and one CLEAR when it returns
- Batch sampling: 2 to 12 readings sent together in one message
- Quick configuration with the Macnman Maya app over Bluetooth, and remote configuration with MQTT commands
- Safe firmware updates over Bluetooth or over the internet: the new firmware is used only after it has arrived complete and been checked

Product Limitations

The device works from -30 °C to +80 °C, but two of its functions have a limit at low temperature:

Function	Works down to	Below this temperature
Bluetooth (Maya app)	-10 °C	The Maya app cannot find or connect to the device: no settings, Ping, Device data, Reminisce export or firmware update over Bluetooth. The device keeps measuring and sending over Wi-Fi.
Wi-Fi	-30 °C	The device cannot join the network or send its readings.

Note:

For cold rooms and freezers, finish the setup with the Maya app and check Ping before the device gets colder than -10 °C. After that, change settings by MQTT command (Chapter 8), or bring the device to a warmer place.

Chapter 3. Hardware Introduction

Packing List

Item	Quantity
MS-WBSD-TH-X1 datalogger, battery fitted	1
Mounting clip	1
Mounting screws	1 set
User manual	1
Warranty card	1
IP68 cable extender *	Optional

Note:
If any of the above items is missing or damaged, please contact your sales representative.
Items marked * are optional accessories and may have to be purchased separately.

Hardware Overview

The MS-WBSD-TH-X1 has two parts:

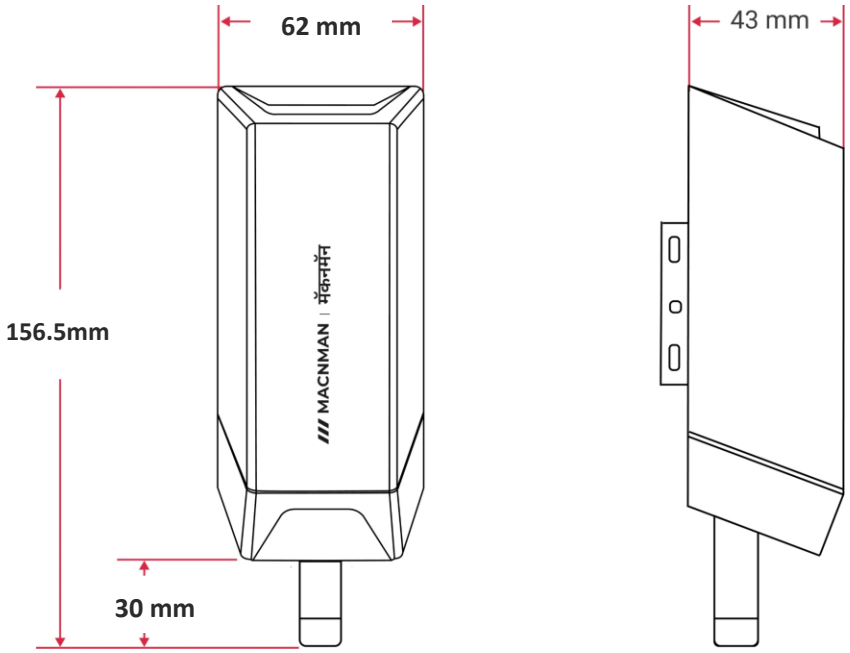
- **Enclosure** - IP65, glass-filled nylon, matte white. It holds the battery, the Wi-Fi radio and the Bluetooth module with their internal antennas, and the LED indicator.
- **Sensor probe** - 30 mm long, at the bottom of the enclosure; holds the Sensirion SHT40 temperature and humidity sensor.

The device has no user-accessible connector: everything is set up from the Maya app over Bluetooth.



Front Back

Dimensions (mm)



Dimension	Value
Height	156.5 mm overall: 126.5 mm enclosure + 30 mm sensor probe
Width x Depth	62 mm x 43 mm
Weight	About 175 g (with battery)

LED Indicator

The LED indicator lights blue, driven by the Bluetooth module, or red, driven by the main processor. In normal operation, with no phone connected, it stays off to save battery - even while the device measures and sends.

Function	LED Indicator	Meaning
Power on / restart	Red: one long and one short blink	The device has just started (power-on, reboot, firmware update or factory reset) and no phone is connected
App connected	Blue: on	The Maya app is connected over Bluetooth
App disconnected	Blue: off	No Bluetooth connection
Command from the app	Blue, then red for a moment	Command received from the app and carried out; quick commands give only a short red flash
Device busy	Red: on, while the app is connected	Carrying out a command, a Wi-Fi task (scan, connection test, Ping, uplink) or a Reminisce export - wait until it goes out
Firmware transfer	Blue: blinking	A firmware update from the app (Push firmware) is running

Note:
The device has no power button: it runs all the time. To restart it, tap Reboot in the Maya app (Chapter 5).

Chapter 4. Quick Start

This chapter describes the steps to quickly put the device into operation. For more advanced settings, please refer to Chapter 5, Operation Guide.

1. Install the **Macnman Maya** app on your smartphone and switch on Bluetooth.
2. Open the Maya app. The **Searching** screen lists the devices nearby, each with its Bluetooth name - **MS-WBSD-TH-X1** from the factory, or the name you have given it - and its Bluetooth address (Serial No), e.g. D6:0E:E5:9B:F0:50. Tap your device to connect. When the card at the top of the device screen shows **Connected** with a green dot, the app is connected.

If several units are nearby, tap the QR icon at the top right and scan the label on the unit in your hand.



Note: Only Supports With the Android Smartphones.

3. Under **WiFi & Servers**, tap **WiFi**. Tap your network under **Available Networks**, enter its password and tap **Okay**. Within about 30 seconds the network appears under **Current Network**, marked **CONNECTED** (Chapter 5).
4. Under **WiFi & Servers**, tap **MQTT**. Enter the broker's **Host** (name or IP address only, without mqtt:// or the port), the **Port** (the broker's plain MQTT port, usually 1883) and, if the broker needs a login, the **User Name** and **Password**. Tap **Set**: when the device has reached the broker, **Status** shows **Online**.
5. Keep the other settings at their defaults, or change them as required: the reporting interval (**Heartbeat**), triggers and sampling (Chapter 5).
6. In your MQTT client, subscribe to the device's topic root followed by /# - for example Macsync-WBS-TH-X1/d60ee59bf050/# - to see everything the device sends (Chapter 8). The device does not retain its messages, so subscribe before you test the connection.
7. Stand at the mounting position and tap **Ping** under **Network** (Chapter 5). A green tick means the broker has received the reading; it also appears on the heartbeat topic in your MQTT client. Aim for a green RSSI dot (Good or better, -65 dBm or stronger); a yellow dot usually works, but an uplink may need a retry. Then close the app connection.

Note:

The device stays awake while the app is connected and uses far more power than while it sleeps. Disconnect the app as soon as the settings are done.

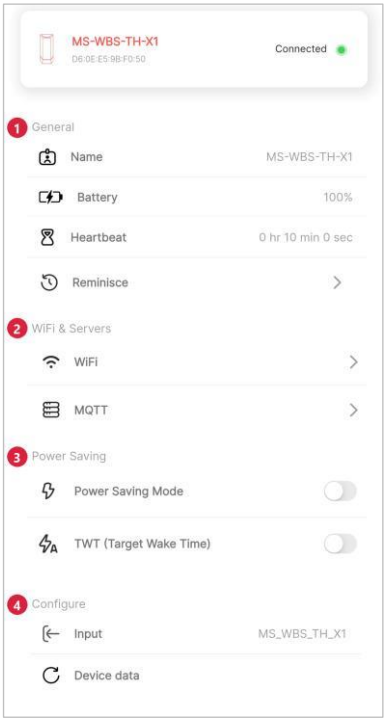
Nothing is sent until both Wi-Fi and an MQTT broker are set: a new or factory-reset device stays quiet.

The device uses plain MQTT without TLS: a broker that accepts only TLS connections (usually port 8883) cannot be used. Wi-Fi networks that need a sign-in page (guest or hotel Wi-Fi) or an enterprise login (WPA2-Enterprise) are not supported either (Chapter 5).

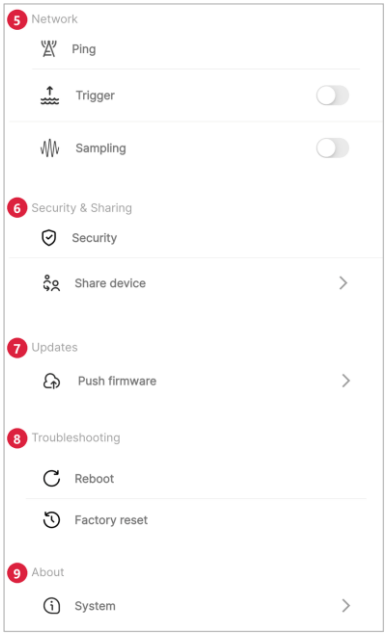
Chapter 5. Operation Guide

The Device Screen

The device is set up from the Macnman Maya app. After connecting, the device screen shows the settings in groups. The numbers in the pictures match the rows of the table.



Top of the screen

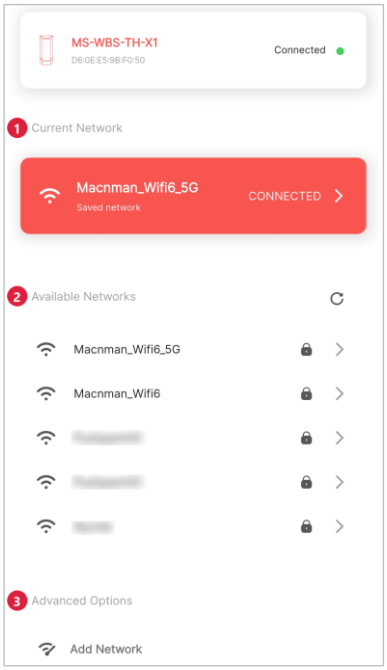


Scrolled down

No.	Group	Settings	Description
1	General	Name, Battery, Heartbeat, Reminisce	Device name, battery level, reporting interval and the export of stored readings
2	WiFi & Servers	WiFi, MQTT	Connection to your Wi-Fi network and to your MQTT broker
3	Power Saving	Power Saving Mode, TWT (Target Wake Time)	Still under implementation - not covered in this guide. The device already saves power by sleeping with its radio off between uplinks.
4	Configure	Input, Device data	Input shows the built-in sensor, MS_WBSD_TH_X1. Device data shows the live readings.
5	Network	Ping, Trigger, Sampling	Connection test, threshold alarms and batch sampling
6	Security & Sharing	Security, Share device	Still under implementation - not covered in this guide
7	Updates	Push firmware	Firmware update over Bluetooth
8	Troubleshooting	Reboot, Factory reset	Restart, or return to factory settings
9	About	System	Serial No and the hardware and software versions

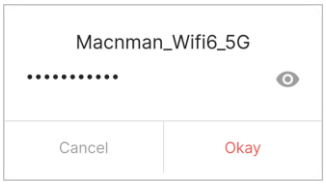
Wi-Fi Settings

The device sends its readings over your Wi-Fi network. It works with Wi-Fi 4, 5 and 6 routers on 2.4 GHz and 5 GHz, secured with WPA2-Personal or WPA3-Personal. Set up Wi-Fi first: the MQTT settings are tested over this connection.

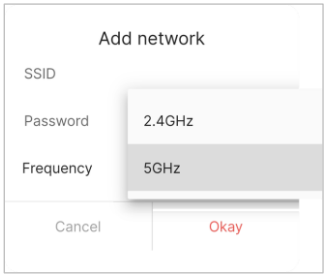


1. Under **WiFi & Servers**, tap **WiFi**.
2. Find your network under **Available Networks** (2). A lock means it needs a password. If your network is missing, tap the refresh icon to scan again - a scan takes about 10 seconds.
3. Tap your network, enter its password and tap **Okay**. Tap the eye icon to check what you typed.
4. Wait while the device tries the network - up to 30 seconds. When it works, the network appears under **Current Network** (1) marked **CONNECTED**, and it is saved on the device.

Use **Add Network** under **Advanced Options** (3) when your network is not in the list - for example a hidden network that does not broadcast its name. Enter the **SSID** exactly as set on the router (upper and lower case matter), the **Password** and the **Frequency** your network uses, **2.4GHz** or **5GHz**, and tap **Okay**. The device then looks for the network on that band only.

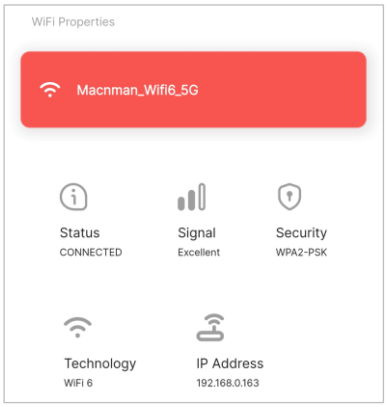


Password



Add network

Tap the network under **Current Network** to see the details of the device's last connection:



- **Status** - CONNECTED if the last connection worked, DISCONNECTED if it failed.
- **Signal** - how strong the Wi-Fi signal was at the device, e.g. Excellent.
- **Security** - the network's security, e.g. WPA2-PSK (WPA2-Personal) or WPA3-SAE (WPA3-Personal).
- **Technology** - the Wi-Fi generation of the link, e.g. WiFi 4, WiFi 5 or WiFi 6.
- **IP Address** - the address your router gave the device.

The device switches its radio off between uplinks, so these details come from its last connection, not a live reading. They are updated at every uplink.

If a connection fails, the app shows a message. A failed attempt never costs you a working network: the device saves a new network only after it has connected to it.

Message	Meaning	What to Do
Invalid WiFi details - SSID cannot be empty	The network name was left empty. Nothing is changed.	Enter the SSID.
WiFi connection failed - check SSID and password (the network is marked AUTH FAILED)	The device could not join within about 30 seconds: usually a wrong password, or a mistyped SSID, the wrong frequency, or a router that is off or out of range. The previous network stays in use.	Check the password (eye icon), the SSID and the frequency, move the device closer to the router, and try again.
DISCONNECTED under Current Network	The last connection failed - for example the router was off or out of reach at the last uplink. The network stays saved.	Check the router. The device tries again at every heartbeat; tap Ping to test at once.

Note:

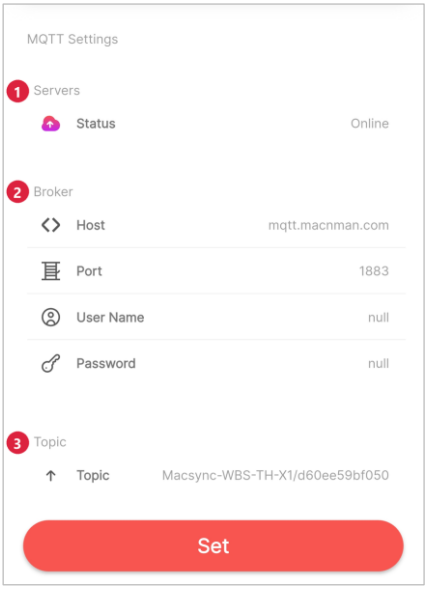
One network is saved. Connecting to a new one replaces it - but only after the new one has worked. The network is kept through sleep, reboots and firmware updates; a factory reset clears it.

The device joins the network only when it sends - at a heartbeat, an alarm, a sampling batch or a Ping - and remembers your router's channel to connect faster. A scan uses about as much battery as four heartbeats.

Not supported: networks that need a sign-in page (guest or hotel Wi-Fi) or an enterprise login (WPA2-Enterprise), and an SSID or password that contains a comma or ends with a space.

MQTT Settings

The device publishes its readings to an MQTT broker - your own server or a hosted one - and collects commands from it. It uses MQTT 3.1.1 over TCP and sends every message with QoS 1, so the broker confirms each one.



1. Under **WiFi & Servers**, tap **MQTT**.
2. Under **Broker** (2), tap **Host** and enter the broker's name or IP address only, without mqtt:// and without the port, e.g. broker.example.com or 192.168.1.100.
3. Check **Port**: 1883 unless your broker uses another port for plain MQTT.
4. If the broker needs a login, enter the **User Name** and **Password**. If not, leave them empty - the app shows null.
5. Leave **Topic** (3) as it is, unless your server expects another one.
6. Tap **Set**. The device joins your Wi-Fi and connects to the broker to test the settings - up to about 30 seconds. If it works, the settings are saved and **Status** (1) shows **Online**.

Parameter	Description
Host	Broker name or IP address, up to 63 characters. Factory setting: not set.
Port	1 to 65535. Factory setting: 1883.
User Name	Up to 32 characters; empty if the broker needs no login. Factory setting: empty (shown as null).
Password	Up to 64 characters. Factory setting: empty (shown as null).
Topic	Base topic, up to 128 characters. The device adds its ID - the Serial No in lowercase without colons - to make the topic root, written <root> in this guide, e.g. Macsync-WBS-TH-X1/d60ee59bf050. Factory setting: Macsync-WBS-TH-X1.
Status	Online: the device reached the broker at the last Set or the last uplink. Offline: no broker is set yet, or the last attempt failed. This is the result of the last connection, not a live check: tap Ping to test now.

To change the base topic, tap **Topic**, type your own - e.g. factory1/coldroom - and tap **Okay**, then **Set**. The device always adds its ID once; an empty field keeps the current base.

The device saves new broker settings only after it has connected with them, so a failed attempt never replaces settings that work.

Message	Meaning	What to Do
Invalid broker details - host and port required	Host is empty, or Port is not a number from 1 to 65535. Nothing is changed.	Fill in Host and Port.
MQTT connection failed - check broker details	The device could not connect within about 30 seconds: a wrong Host, Port, User Name or Password; a broker that is down, blocked by a firewall or accepts TLS only; or no working Wi-Fi. The previous settings stay in use, and the screen shows them again.	Check each setting and the Wi-Fi. Make sure the broker accepts plain MQTT on this port.
Offline after it was Online	The last uplink could not reach the broker. Readings are still saved on the device.	Check the broker and the network; tap Ping to test. Readings missed meanwhile can be exported with Reminisce.

Note:

Plain MQTT only: the device connects without TLS encryption, so use the broker's plain MQTT port - usually 1883. A broker that accepts only TLS connections (usually port 8883) cannot be used.

One broker is saved, kept through sleep, reboots and firmware updates. A factory reset clears it and sets the topic back to Macsync-WBS-TH-X1.

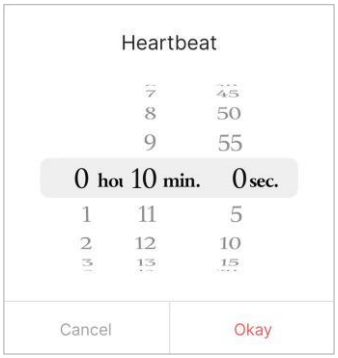
If the network or the broker is down, readings are still saved on the device but not sent later. After 3 failed uplinks in a row the device waits longer between wakes - its normal interval times the number of failures, at most 1 hour - and is back to normal after the next successful uplink or Set, or as soon as a phone connects.

No commas in Host, User Name, Password or Topic, and no space at the end. The broker requirements (sessions, client ID) are in Chapter 8, Overview, under QoS, Retain and Sessions.

General Setting

General settings include the basic parameters of the device.

- Under **General**, tap **Name** to rename the device, or **Heartbeat** to change the reporting interval; turn the hours, minutes and seconds wheels and tap **Okay**.



Parameter	Description
Name	Bluetooth name used by the Maya app, e.g. cold_storage or Warehouse_B: up to 20 characters. Default MS-WBSD-TH-X1. After saving, the Bluetooth module restarts and the app disconnects; reconnect under the new name. A factory reset does not change the name. The name does not change the MQTT topic or any message: on your server the device is always identified by its Serial No.
Battery	Estimated remaining battery, in percent (Chapter 7). Information only.
Heartbeat (Reporting Interval)	How often the device wakes up, measures, saves the reading and sends it to your MQTT broker. Range: 1 minute to 24 hours. Default: 60 minutes. Takes effect at once: the next reading is sent one full interval after you tap Okay.
Reminisce	Export of the readings stored on the device to your phone (Data Export below).

At every heartbeat:

- The device reads the sensor and saves the reading in its memory.
- It joins your Wi-Fi network and publishes the reading on <root>/heartbeat (Chapter 8), usually within 4 to 5 seconds.
- It collects any MQTT commands waiting at the broker and replies to them.
- It switches the radio off and sleeps until the next heartbeat.

Note:

Outside the range, the app shows "Minimum uplink interval is 1 minute" or "Maximum uplink interval is 24 hours", and the device keeps its previous heartbeat.

The next heartbeat is timed from the end of the previous uplink, so readings are spaced by the interval plus the few seconds an uplink takes - for example 10:00:00, 10:10:05, 10:20:10.

A shorter heartbeat gives more detail but uses the battery faster: every uplink switches Wi-Fi on for a few seconds. At 1 minute the battery lasts about 4.6 months; at 1 hour more than 10 years (Chapter 7).

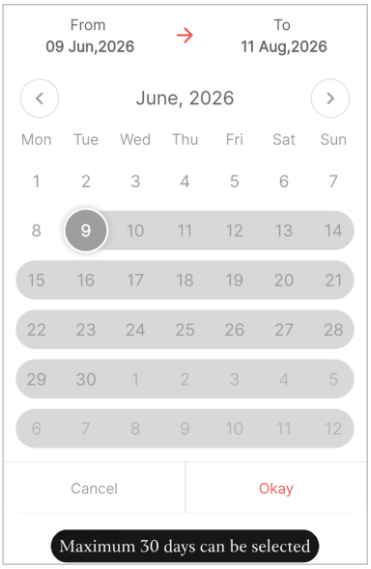
While Sampling is on, the heartbeat is paused. The heartbeat can also be changed remotely with an MQTT command (Chapter 8).

Data Export (Reminisce)

The device saves every reading in its own memory before it sends it - up to 1,550,400 readings. **Reminisce** copies the readings stored between two dates to your phone over Bluetooth. Use it to recover readings taken while Wi-Fi or the broker was down, or for offline analysis and record-keeping.

1. Under **General**, tap **Reminisce**.

2. Choose the **From** and **To** dates and tap **Okay**. One export covers 2 to 30 days; for a longer range the app shows **Maximum 30 days can be selected**.
3. The app shows **Downloading historical data...** while the device sends the readings. Keep the phone close to the device and stay on this screen.
4. When the download-complete message appears, the file is on your phone in **Download > Macnman**, named after the dates, e.g. **21 Aug 2026 - 16 Sep 2026.txt**.



Choosing the dates



Downloading

The file is plain text that opens in Excel or any spreadsheet program. The first line names the columns; every line after it is one reading, oldest first:

```
Timestamp,Temperature(C),Humidity(%)
1789157472,31.87,49.99
1789157631,31.89,49.19
1789157935,31.82,49.92
```

Column	Description
Timestamp	When the reading was taken, as Unix time in UTC - seconds since 1 January 1970
Temperature(C)	Degrees Celsius, two decimals; below zero with a minus sign, e.g. -0.25
Humidity(%)	Relative humidity in %, two decimals

Date and time in Excel. Split the file into columns (Data > Text to Columns > Delimited > Comma). Enter =A2/86400 + DATE(1970,1,1) + 5.5/24 in cell D2, format it as dd/mm/yyyy hh:mm:ss and fill it down. The 5.5/24 converts UTC to India time (IST); use your own offset elsewhere, or 0 to keep UTC.

What is stored:

Reading	Stored
Every heartbeat reading	Yes - saved before it is sent, so it is kept even if the uplink fails
Every sample while Sampling is on	Yes - each with its own time. Temperature and humidity are both saved, whichever parameter Sampling sends.
A reading sent with Ping	Yes
Trigger checks	No - they only compare the value with the limits
Device data (the live reading in the app)	No
A failed sensor read	No - it shows as a gap in the timestamps
Readings taken before the device's clock is set	No (Time Synchronization below)
A second reading in the same second	No - skipped to avoid duplicate rows, e.g. after two quick Pings

Note:

More than 30 days? Export in parts of up to 30 days each; every part is saved as its own file.

Export time: about 190 readings a second. Exporting 30 days takes about 4 min at a 1-minute heartbeat, 25 s at 10 minutes and 4 s at 60 minutes.

Only stored readings are exported - nothing is filled in. If the Bluetooth connection drops, the file is incomplete: export the same dates again.

Stored readings are kept through sleep, reboots and firmware updates; a factory reset clears them. When the memory is full, the oldest readings are overwritten, 510 at a time - at a 1-minute heartbeat after about 3 years; from 10 minutes up, the memory outlasts the device.

Each saved reading takes 8 bytes: 4 for the time, 2 for the temperature and 2 for the humidity.

Time Synchronization

The device sets its clock from the internet, the first time your Wi-Fi network gives it internet access. The time of each reading is sent in the heartbeat message, and the time of each batch in the sampling message, as Unix time (UTC) and saved with every stored reading.

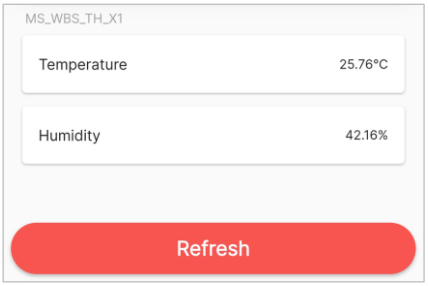
Note:

Until the clock is set, the ts field of a heartbeat or a sampling message is 0 and readings are not stored in the memory. A new device therefore stores readings only after it has first reached the internet.

Device Data

Device data shows the temperature and humidity the sensor is measuring now - use it to check the sensor during installation, or to compare it with a reference instrument.

1. Under **Configure**, tap **Device data**.
2. The app shows the **Temperature** in °C and the **Humidity** in %, with two decimals.
3. Tap **Refresh** for a new reading at any time; the new values appear in under a second.

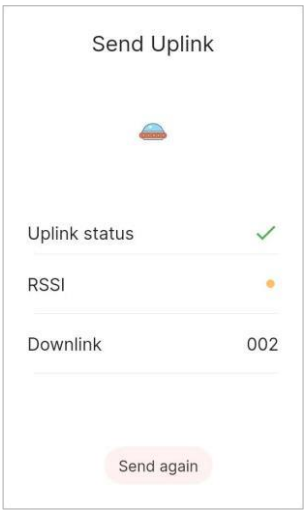


Nothing is sent to the broker, nothing is saved in the memory and the reporting schedule is not affected. **Error** instead of a value means the sensor could not be read: tap Refresh again, and contact Macnman support if it continues.

Network Test (Ping)

Ping tests the whole path from the device to your server: the device takes a reading, joins your Wi-Fi network, sends the reading to your MQTT broker and waits for the broker to confirm it - exactly like a heartbeat, but straight away.

1. Stand at the mounting position. Under **Network**, tap **Ping**.
2. The **Send Uplink** window shows the result, normally within 5 to 10 seconds. If the network or the broker cannot be reached, it can take up to about a minute before the failure shows.
3. Tap **Send again** to repeat, for example at another spot.



Result	Description
Uplink status	Green tick: the broker confirmed that it received the reading, so the whole path works - device, Wi-Fi, internet, broker. A failure means the reading did not reach the broker.
RSSI	The Wi-Fi signal strength at the device during this uplink, as a coloured dot (table below). Closer to 0 dBm is stronger.
Downlink	How many MQTT commands the device has received, e.g. 002. A Ping also collects commands waiting at the broker, so the number can go up after a Ping. It starts again from 0 after a factory reset.

The RSSI dot follows the signal bands below. The same words - Excellent, Good, Fair, Weak - appear as **Signal** in the Wi-Fi connection details.

RSSI Dot	Signal	RSSI (dBm)	What to Expect
Green	Excellent	-30 to -65	Strong signal; uplinks are reliable.
Yellow	Fair	-66 to -75	Usually works, but an uplink may need a retry. Move closer to the router if you can.
Red	Weak	-76 to -90	Uplinks may fail. Move the device or the router, or add an access point.

If the Ping fails:

What You See	Likely Cause	What to Do
Uplink fails on a new device	Wi-Fi or the MQTT broker is not set yet. The device sends nothing until both are set.	Set up Wi-Fi and MQTT (this chapter).
Uplink fails after a working setup	The device could not join Wi-Fi (router off, out of range, password changed) or could not reach the broker (broker down, network blocked).	Check the Wi-Fi connection details and the MQTT Status, fix the cause and tap Send again.
Uplink succeeds, red RSSI dot	The reading got through, but the signal is weak, so later uplinks may fail.	Move the device or the router, or add an access point; then Ping again.
The result takes a long time	The device is retrying: up to 2 Wi-Fi and 2 broker attempts of 15 seconds each.	Wait for the result. A long wait usually ends in a failure - check the network.

Note:

Each Ping is a real transmission and uses about as much battery as a heartbeat. The heartbeat schedule does not move.

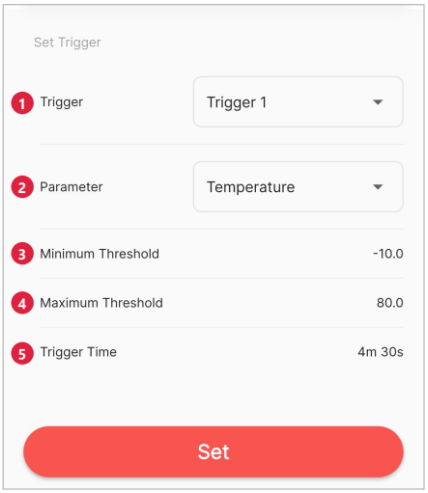
The Ping reading is published on <root>/heartbeat marked "wake":"ping" and saved in the memory. Ping also updates the Wi-Fi connection details and the MQTT Status.

Ping is also the quickest way to make the device collect a waiting MQTT command (Chapter 8).

Threshold Setting (Trigger)

A trigger watches temperature or humidity and sends an alarm as soon as the value leaves the range you set. The device has two independent triggers, each with its own parameter, limits and check time.

1. Tap **Trigger**.
2. Choose **Trigger 1** or **Trigger 2** (1) and the **Parameter** (2): Temperature or Humidity.
3. Enter the **Minimum Threshold** (3) and **Maximum Threshold** (4) - the normal range - and the **Trigger Time** (5), the check interval.
4. Tap **Set**. The trigger is saved and switched on; Sampling is switched off.



Parameter	Description
Trigger	Trigger 1 or Trigger 2. Factory setting: both off; Trigger 1 on Temperature, Trigger 2 on Humidity.
Parameter	Temperature or Humidity. The thresholds are then filled in with the full range of that parameter.
Minimum / Maximum Threshold	Temperature -30.00 to 80.00 °C, humidity 0.00 to 100.00 %RH, two decimals. The minimum must not be above the maximum. Factory setting: the full range.
Trigger Time	How often the value is checked: 10 seconds to 24 hours. Factory setting: 30 seconds.

How a trigger works:

- At every Trigger Time the device reads the sensor with the radio off. If the value is inside the range, nothing is sent; a value exactly equal to the minimum or maximum counts as inside.
- If the value is outside, the device reads it twice more . Only if all three readings are outside the value it sends an **ALARM** on <root>/trigger (Chapter 8).
- While the value stays outside, nothing more is sent. When it comes back inside, the device sends one **CLEAR**.
- If an alarm cannot be sent because the network is down, the device tries again at the next check. Changing or re-saving a trigger starts it afresh: if the value is still outside, a new ALARM is sent.
- The heartbeat carries on while triggers are on. Trigger checks are not saved in the memory. The **Trigger** switch under Network turns Trigger 1 on with its saved settings (and Sampling off), or both triggers off; switch Trigger 2 on with Set Trigger.

Note:

Short Trigger Times wake the device often and shorten the battery life. With a 1-hour heartbeat and a good signal, the battery lasts more than 10 years with no trigger, about 5.4 years with one trigger checked every 5 minutes and about 11 months at the default 30 seconds. Choose the longest Trigger Time that fits how fast your conditions really change.

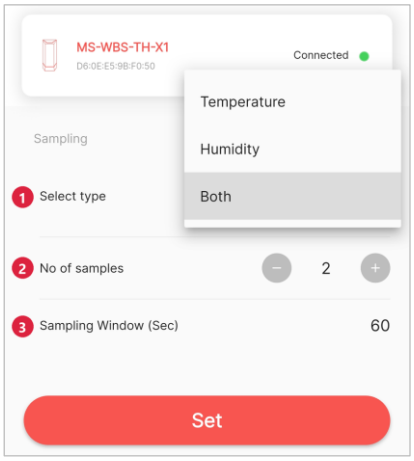
If a setting is out of range, the app shows a message and the trigger keeps its previous settings.

Switching Sampling on switches both triggers off. They do not come back on by themselves when Sampling is switched off.

Sampling Setting

Sampling takes several readings with the radio off and sends them together in one message - detailed data at low power.

1. Tap **Sampling**.
2. **Select type** (1): Temperature, Humidity or Both.
3. **No of samples** (2): 2 to 12. **Sampling Window (Sec)** (3): tap it and choose the time between two readings, 1 to 59 minutes; the row then shows it in seconds (60 = 1 minute).
4. Tap **Set**. Sampling is switched on; both triggers are switched off and the heartbeat pauses.



Parameter	Description
Select type	Temperature, Humidity or Both: what goes into the message. Factory setting: Both.
No of samples	2 to 12 readings per message. Factory setting: 2.
Sampling Window	Time between two readings: 1 to 59 minutes (60 to 3,540 seconds). Factory setting: 60 seconds. One message is sent every No of samples x Sampling Window - e.g. 6 x 5 min = every 30 minutes.

Note:

The batch is published on <root>/sampling, oldest reading first (Chapter 8). A batch that cannot be sent is not sent again later; its samples stay in the memory and can be exported with Reminisce.

The Sampling switch under Network turns sampling on or off. When sampling is switched off, the heartbeat resumes, usually with one heartbeat soon after; the triggers stay off until switched on again.

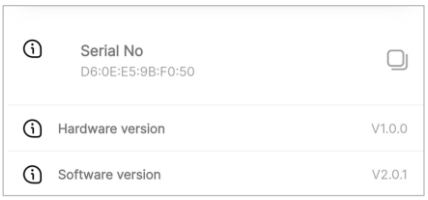
While sampling is on, the device connects only when it sends a batch, so MQTT commands wait until then - up to 11 h 48 min at the maximum settings. Changing the settings starts a new batch.

If a setting is out of range, the app shows a message and sampling keeps its previous settings.

Sampling saves battery: 12 samples 5 minutes apart, sent once an hour, last about 5.7 years - three times as long as a 5-minute heartbeat.

System Information

Under **About**, tap **System** to see the device's identity:



- **Serial No** - the device's unique Bluetooth address, e.g. D6:0E:E5:9B:F0:50. In lowercase without colons it is the device ID in the MQTT topics. Tap the copy icon to copy it.
- **Hardware version** - the version of the device's electronics, e.g. V1.0.0.
- **Software version** - the firmware version, e.g. V1.0.0.

The same versions are sent in the boot message after every restart (Chapter 8).

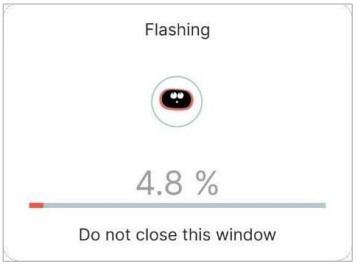
Updates

New firmware can be installed from the Maya app over Bluetooth, or remotely with an MQTT command (Chapter 8). Either way, the device keeps running its current firmware until the new one has arrived complete and been checked, so an interrupted update never leaves it unusable.

1. Connect to the device and note its Software version under **About > System**.
2. Under **Updates**, tap **Push firmware**. Open the folder for this product - **Wi-Fi > Sensors > MS-WBSD-TH-X1** - and select the firmware file; its badge shows the version you are about to install. The phone needs an internet connection.
3. Slide to confirm. Keep the phone next to the device with the app open until the transfer finishes - it takes a few minutes.
4. The device checks the firmware, restarts with it and the connection ends. Connect again and check the Software version.



Selecting the firmware file



Transfer in progress

Note:

During the transfer the device does not send heartbeats or check triggers.

If anything goes wrong - the connection drops, the phone moves out of range, the power fails or the file is damaged - the device keeps its current firmware and carries on as before. Start the update again.

Settings, stored readings and the battery level are kept. After the restart, the device sends a boot message with the new firmware version at its first uplink.

Reboot

Under **Troubleshooting**, tap **Reboot**. No slide to confirm is needed: the device restarts as soon as the app disconnects, at most 30 seconds later.

All settings, the stored readings, the battery level and the downlink count are kept. After the restart the device takes a reading and sends it straight away, together with a boot message; the uplink message number starts again from 1. Use a reboot to recover from a temporary problem.

Reset to Factory Default

Use a factory reset before moving the device to a new site or a new owner: it erases the saved Wi-Fi network and password, the MQTT broker and login, and the stored readings. Export any readings you still need with **Reminisce** first. Then, under **Troubleshooting**, tap **Factory reset** and slide to confirm. The connection ends and the device restores the factory settings below and restarts. Set up Wi-Fi and MQTT again: until then the device sends nothing.

Setting	Factory Value
Heartbeat	60 minutes
Wi-Fi network	Not set
MQTT Host	Not set
MQTT Port	1883
MQTT User Name and Password	Empty - no login
MQTT Topic	Macsync-WBS-TH-X1 (the device adds its device ID)
Trigger 1	Off - Temperature, -30.00 to 80.00 °C, checked every 30 s
Trigger 2	Off - Humidity, 0.00 to 100.00 %RH, checked every 30 s
Sampling	Off - Both, 2 samples, 60 s window

Note:

A factory reset does not change the Bluetooth name. The device keeps the name it had before the reset; to use the factory name again, set it under Name (General Setting).

Erased readings can no longer be exported. A factory reset also restarts the uplink and downlink counters.

A factory reset does not change the battery level (Chapter 7).

The firmware version and the Serial No never change with a reset.

Chapter 6. Installation

Installation Location

- Keep the device away from metal objects, metal cabinets and large obstacles between the device and the Wi-Fi router or access point.
- Mount the device as high as practical, with the sensor probe at the bottom, as shown in Chapter 3.
- For correct readings, keep the device out of direct sunlight and away from heaters, air-conditioner outlets and doors.
- Before fixing the device, use **Ping** at the mounting position to check the Wi-Fi signal (Chapter 5). Aim for Good or better: -65 dBm or stronger.

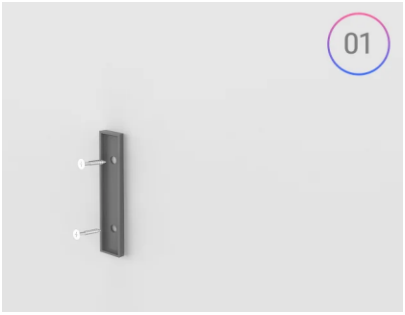
Do not install the device in the places below. They block or weaken the Wi-Fi signal, so uplinks may fail and the battery runs down faster: a weak signal keeps Wi-Fi on longer for every message (Chapter 7).

Avoid	Why	Do this instead
Inside a metal box, metal cabinet or metal panel	Metal blocks Wi-Fi. A closed box also gives wrong readings, because room air cannot reach the sensor.	Mount the device outside the box, in free air.
Near electrical panels (distribution boards, MCB boxes) and AC mains equipment	The metal panel blocks Wi-Fi, and the equipment gives off heat that changes the readings.	Keep the device at least 1 m away from them.
Near a microwave oven, wireless cameras, cordless phones, baby monitors or Bluetooth speakers	They use the same 2.4 GHz band as Wi-Fi and can disturb it.	Keep the device at least 1 m away from them.
Far from the Wi-Fi router, with many walls or floors in between	Wi-Fi reaches much less far than LoRaWAN. Every wall and floor weakens it.	Mount the device where Ping shows Good or better. If the signal does not reach, add an access point or extender near the device, or move the router.
On the 5 GHz band, far from the router	5 GHz reaches less far and goes through walls less well than 2.4 GHz.	For a device far from the router, connect it to the router's 2.4 GHz network.
Underground, in a basement, or behind thick concrete walls, steel structures or large water tanks	Wi-Fi hardly passes through soil, concrete, steel or water.	Mount the device above ground with a clear path to the router, or add an access point near the device.
Flat against a large metal surface (metal wall, door or cabinet side)	Metal right behind the device weakens the signal.	Mount it on a non-metal wall. On a metal pole, turn the device towards the router.
Covered with metal tape, foil, metallic paint or a metal cover	It blocks Wi-Fi like a metal box.	Do not paint or cover the device.
On a guest or hotel Wi-Fi with a sign-in page, or a WPA2-Enterprise network	The device cannot sign in to these networks, so it sends nothing.	Use a WPA2-Personal or WPA3-Personal network (Chapter 5).

Note:

After installation, tap Ping again at the final position. Aim for Good or better: -65 dBm or stronger. If the signal is Weak, move the device, move the router or add an access point near the device.

Wall Mounting



1. Fix the mounting clip



2. Slide the device down



3. Device mounted

1. Stick the white reference tape from the box on the wall where the device will go. Its two marked holes are the drilling template.
2. Drill the two holes at the marks on the tape.
3. Screw the mounting clip to the wall with the mounting screws supplied.
4. Slide the device down onto the clip until it clicks and locks in place.
5. Optional: for extra security, put a screw through the extra screw holes on the other side of the device.

Pole Mounting

1. Remove the mounting clip from the device.
2. Put metal or plastic zip ties through the mounting holes of the clip.
3. Hold the device, with the clip and the zip ties, against the pole at the height you want.
4. Wrap the zip ties around the pole and pull them tight until the device is held firmly. Plastic and stainless-steel zip ties can both be used.

Note:
Mount the device upright with the sensor probe pointing down. The enclosure is IP65; the operating range is -30 °C to +80 °C.

Chapter 7. Battery Information

The device runs on one non-rechargeable 3.6 V, 19 Ah Li-SOCl2 cell. The battery level is shown in the Maya app (Battery, under General) and sent in every heartbeat, sampling and boot message.

Item	Description
Battery type	ER34615: 3.6 V, 19,000 mAh Li-SOCl2 (lithium thionyl chloride), D size
Battery life	More than 10 years at the default 1-hour heartbeat with a good Wi-Fi signal, with no triggers or sampling.

How long the battery lasts depends mostly on how often the device wakes - to send heartbeats, trigger alarms and sampling batches, and to check triggers - and on the Wi-Fi signal. On a good signal a message takes about 3.5 seconds from wake-up to sleep; on a fair (yellow) signal the radio draws more, and each message costs about 1.7 times as much. Check yours with Ping (Chapter 5).

Battery life by heartbeat and Wi-Fi signal, with no triggers or sampling:

Heartbeat	Messages a Day	Good Wi-Fi Signal	Fair Wi-Fi Signal
1 minute	1,440	4.6 months	2.8 months
5 minutes	288	1.8 years	1.1 years
15 minutes	96	4.7 years	3.0 years
30 minutes	48	7.7 years	5.3 years
1 hour (default)	24	More than 10 years	8.6 years
6 hours	4	More than 10 years	More than 10 years
24 hours	1	More than 10 years	More than 10 years

Battery life for common needs, with a good Wi-Fi signal:

What You Need	Settings	Battery Life
A reading every minute, sent straight away	Heartbeat 1 minute	4.6 months
A reading every minute, sent every 12 minutes	Sampling: No of samples 12, Sampling Window 60 s	1.4 years
A reading every 5 minutes, sent straight away	Heartbeat 5 minutes	1.8 years
A reading every 5 minutes, sent once an hour	Sampling: No of samples 12, Sampling Window 300 s	5.7 years
A reading every hour	Heartbeat 1 hour (the default)	More than 10 years
Hourly readings, and an alarm within about 5 minutes of a limit being crossed	Heartbeat 1 hour; one trigger with a Trigger Time of 5 minutes	5.4 years
Hourly readings, and an alarm within about 30 seconds	Heartbeat 1 hour; one trigger with a Trigger Time of 30 seconds (the default)	11 months

What shortens the battery life:

Cause	What to Do
A short heartbeat	Choose the longest heartbeat that suits you. For frequent readings, use Sampling instead: the same readings, sent in batches, make the battery last three times as long or more.
A short Trigger Time	Choose the longest Trigger Time that fits how fast your conditions change, and switch off a trigger you do not need. Every check wakes the device and costs about a fifth of a message; two triggers checked every 30 seconds give about 8.5 months.
A weak Wi-Fi signal	Each message then costs more: about 1.7 times as much on a fair (yellow) signal, and more again on a weak (red) one. Check the signal with Ping before mounting the device and aim for Good or better: -65 dBm or stronger.
Wi-Fi or broker unreachable	Each failed attempt costs as much as about 6 messages with the router off, and more if the router works but the broker does not answer. After three failures in a row the device tries less often - up to an hour apart - and returns to normal after the next message gets through, or when the app connects. At a 1-hour heartbeat a router that stays off still drains the battery about three and a half times as fast, so fix the network promptly.
The app left connected	The device stays awake while the app is connected: one minute costs as much as 6 messages. Disconnect when you are finished. With no command for 5 minutes, the device goes back to sleep on its own.
Cold	Batteries of this type deliver less in the cold. Expect a shorter battery life where it is often below 0 °C.
Storage	A device that is not in use still sleeps with Bluetooth on, so that the Maya app can find it - about 4 % of the battery a year, plus about 1 % the cell loses on its own. Install the device soon after delivery.

Note:

The battery is fitted at the factory and cannot be replaced by the user.

The figures come from the current measured on the device, about 80 µA while it sleeps. A lithium thionyl chloride cell also loses about 1 % of its capacity a year on its own, so figures of 10 years or more are given as more than 10 years.

The battery level is an estimate: the device adds up the energy it uses - sleeping, Wi-Fi, checks, app connections - and subtracts it from the battery's usable capacity, rather than measuring the cell. Allow about ±5 % after months of use. The level reaches 0 % before the battery is completely empty.

A factory reset, a reboot and a firmware update all keep the battery level. It starts again from 100 % only after a new battery has been fitted, with the MQTT command new_battery (Chapter 8).

At the end of its life, dispose of the device according to local regulations for lithium batteries.

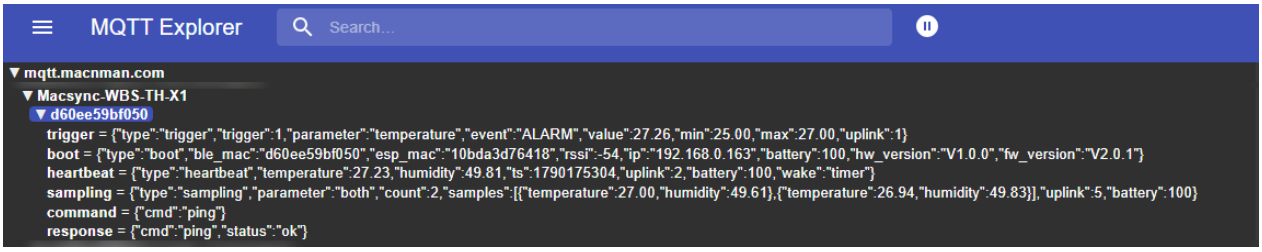
Chapter 8. Uplink and Downlink

Overview

The device publishes JSON messages (uplinks) to your MQTT broker and receives commands (downlinks) from it. Every topic starts with the device's topic root, written <root> in this guide: the base topic set in the app, then the device ID - the Serial No in lowercase without colons - e.g. Macsync-WBS-TH-X1/d60ee59bf050. Times are Unix time in seconds, UTC.

Topic	Direction	Message
<root>/boot	Uplink	Start-up information: once after every restart, with the first uplink
<root>/heartbeat	Uplink	Sensor reading: at every heartbeat, at every Ping and once straight after a reboot
<root>/trigger	Uplink	Trigger ALARM or CLEAR
<root>/sampling	Uplink	Sampling batch
<root>/response	Uplink	The device's replies to your commands
<root>/command	Downlink	Commands from your server, collected by the device at its next uplink

To watch everything one device sends, subscribe to <root>/# - e.g. Macsync-WBS-TH-X1/d60ee59bf050/# - in any MQTT client, such as MQTT Explorer (free, for Windows, macOS and Linux). Subscribe to Macsync-WBS-TH-X1/+/heartbeat for the heartbeats of all devices. Each topic shows the latest message published on it, including the last command you sent:



QoS, Retain and Sessions

Setting	Value	Why
QoS of the device's messages	1	The broker acknowledges every message; the device waits for that before it goes back to sleep.
QoS of your commands	1	The broker keeps the command while the device sleeps and delivers it at the next uplink.
Retain	Off - always	The device never retains its messages, and refuses retained commands: a retained command would be delivered again every time the device connects.
Session	Persistent (clean session off)	The broker remembers the device and keeps its commands while it sleeps or cannot connect.
Client ID	The device's Wi-Fi MAC address, e.g. 10bda3d76418	Fixed, so the broker finds the device's session at every connection.
Keep-alive	30 seconds	Only matters during the few seconds the device is connected.

Your broker must therefore:

- **Support persistent sessions** - standard in MQTT 3.1.1, and on by default in Mosquitto, EMQX and HiveMQ.
- **Keep queued messages through a broker restart** - otherwise a restart deletes every waiting command (Mosquitto: persistence true).
- **Not expire sessions too early** - a device that cannot reach the broker for a long time, for example during a long Wi-Fi outage, must still find its queue. Mosquitto keeps sessions unless persistent_client_expiration is set; hosted brokers often expire them, so check the setting.
- **Allow the client ID** - brokers that restrict client IDs must accept the device's Wi-Fi MAC address.
- **Accept plain MQTT over TCP** - usually on port 1883; the device does not use TLS (Chapter 5).

Uplink Data

Start-up Information (<root>/boot)

Published once after every restart, with the first uplink.

```
{"type":"boot","ble_mac":"d60ee59bf050","esp_mac":"10bda3d76418","rssi":-54,
 "ip":"192.168.0.163","battery":100,"hw_version":"V1.0.0","fw_version":"V1.0.0"}
```

Field	Description
ble_mac	The Serial No - the device ID used in the topics
esp_mac	The device's Wi-Fi MAC address - also its MQTT client ID
rssi	The Wi-Fi signal at that moment, in dBm
ip	The address the router gave the device
battery	Battery level, %
hw_version, fw_version	Hardware and firmware versions

Sensor Reading (<root>/heartbeat)

Published at every heartbeat, at every Ping, and once straight after a reboot, together with the start-up information.

```
{"type":"heartbeat","temperature":30.26,"humidity":61.18,
 "ts":1789907847,"uplink":18,"battery":98,"wake":"timer"}
```

Field	Description
temperature, humidity	The reading, in °C and %RH, with two decimals
ts	When the reading was taken, as Unix time in UTC (seconds since 1 January 1970) - 0 until the device's clock has been set
uplink	Message number - goes up by one for every message the broker receives from the device
battery	Battery level, %
wake	timer for a scheduled heartbeat, ping for a reading sent with Ping in the app

Note:
If the sensor cannot be read, a heartbeat carries "error" in place of the temperature and humidity.

Trigger Report (<root>/trigger)

Published when a trigger raises an ALARM or a CLEAR (Chapter 5).

```
{"type":"trigger","trigger":1,"parameter":"temperature","event":"ALARM",
 "value":36.20,"min":10.00,"max":35.00,"uplink":42}
```

Field	Description
trigger	1 or 2
parameter	temperature or humidity
event	ALARM when the value left the range, CLEAR when it came back
value	The reading that caused the event
min, max	The trigger's limits
uplink	Message number, as in the heartbeat

Sampling Batch (<root>/sampling)

Published after the last reading of each batch. The samples are listed oldest first; with Both, each sample carries temperature and humidity. Example with Humidity and 3 samples:

```
{"type":"sampling","parameter":"humidity","count":3,
 "samples":[{"humidity":62.33},{"humidity":62.35},{"humidity":62.41}],
 "ts":1789909047,"uplink":19,"battery":98}
```

Field	Description
parameter	temperature, humidity or both - the Select type
count	Number of samples in the message
samples	The readings, oldest first
ts	When the batch was completed - the time of its last sample, as Unix time (UTC); 0 until the device's clock is set
uplink, battery	Message number and battery level, as in the heartbeat

Note:

The other samples were taken one Sampling Window apart before ts, oldest first. The exact time of every reading is in the Reminisce export (Chapter 5).

Downlink Command

The device's settings can be changed from your server by publishing a command to <root>/command. There are seven commands: txtime (heartbeat), trigger, sampling, ota (firmware update), ota_source (firmware address), ping and new_battery. The device answers on <root>/response, with QoS 1 - normally with one reply per command.

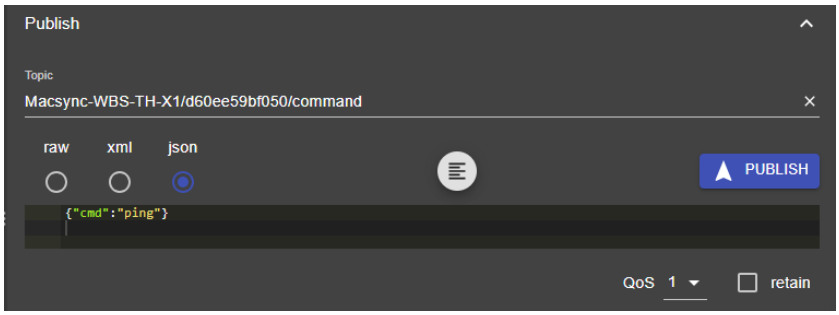
The device sleeps between uplinks, so a command is not delivered the moment you publish it. The broker keeps it and hands it over at the device's next uplink:

```
14:00  your server publishes {"cmd":"txtime","interval":300}   QoS 1, retain off
       the device is asleep: the broker keeps the command
14:30  the device wakes for its heartbeat and connects to Wi-Fi and the broker
       the broker delivers every waiting command; the device publishes its heartbeat
       the device listens until nothing new arrives for 1 s (5 s at most)
       it applies each command, saves it and replies on <root>/response
         {"cmd":"txtime","interval":300,"mode":"heartbeat"}
       then it disconnects and sleeps
14:35  the next heartbeat comes on the new 5-minute interval
```

- **How long a command waits** - depends on the heartbeat: with a 60-minute heartbeat, up to an hour. In sampling mode the device connects only when it sends a batch.
- **To apply a command at once** - tap **Ping** in the Maya app: the device connects straight away and collects it.
- **A command still arrives after a long outage** - if the device cannot reach the broker for a long time, for example because the Wi-Fi is down, the broker keeps the command until the device connects again.
- **A new device must send one uplink first** - the broker starts keeping commands for the device from its first connection.
- **Only the firmware update restarts the device** (ota) - all other commands apply at once, without a restart.

Sending a Command

1. In MQTT Explorer, open the **Publish** panel.
2. **Topic**: the device's command topic, e.g. Macsync-WBS-TH-X1/d60ee59bf050/command.
3. Choose **json** and type the command, e.g. {"cmd":"ping"}.
4. Set **QoS** to **1** and leave **retain** off.
5. Click **Publish**. After the device's next uplink, its reply appears under **response**.



Command List

Command	What It Does	In the App
txtime	Sets the heartbeat interval	Heartbeat
trigger	Sets up one trigger, or switches it on or off	Trigger
sampling	Sets up sampling, or switches it on or off	Sampling
ota	Downloads and installs the firmware (Macnman's, or yours via ota_source); the device restarts	Push firmware
ota_source	Sets the GitHub address the next firmware update downloads from	-
ping	Checks that commands reach the device and replies come back	-
new_battery	Sets the battery level back to 100 % after a new battery has been fitted	-

The cmd field is always required; txtime also needs interval, and trigger also needs trigger (the trigger number, 1 or 2). Any other field you leave out keeps the value already saved on the device.

Command	Field	Values	Description
txtime	interval	60 to 86400 - required	Heartbeat interval in seconds: 300 = 5 min, 1800 = 30 min, 3600 = 1 h. The next heartbeat comes one interval after the command is applied.
trigger	trigger	1 or 2 - required	Which trigger.
	index	0 = Temperature, 1 = Humidity	What it watches.
	min, max	Temperature -30 to 80, humidity 0 to 100; decimals allowed	The allowed range; min must not be above max. An alarm is raised below min or above max.
	timeslot	10 to 86400	Seconds between checks (the Trigger Time).
	enable	1 = on, 0 = off	Switches this trigger on or off. Switching a trigger on switches sampling off.
sampling	index	0 = Temperature, 1 = Humidity, 2 = Both	What goes into the message (Select type).
	count	2 to 12	Samples per batch.
	sample_time	60 to 3540	Seconds between samples (the Sampling Window).
	enable	1 = on, 0 = off	Switches sampling on or off. Switching it on switches both triggers off and pauses the heartbeat.
ota_source	url	A raw.githubusercontent.com address, up to 226 characters; "" = Macnman's address	Where the next ota downloads the firmware. Leave url out to see the address in use. See Firmware from Your Own GitHub.
ota, ping, new_battery	-	-	No fields.

Note:

Values must be plain numbers: "enable":1, not "enable":"1" or "enable":true.

A successful trigger command clears that trigger's alarm state; if the value is still outside the range, a new ALARM follows. A successful sampling command starts a new batch.

While sampling is on, a new interval is saved, but the heartbeat stays paused until sampling is switched off.

Ready-made commands:

What You Want	Publish This
Heartbeat every 30 minutes	{"cmd":"txtime","interval":1800}
Trigger 1 on: temperature 10 to 40 °C, checked every 60 s	{"cmd":"trigger","trigger":1,"index":0,"min":10,"max":40,"timeslot":60,"enable":1}
Trigger 2 on: humidity 30 to 70 %RH, checked every 120 s	{"cmd":"trigger","trigger":2,"index":1,"min":30,"max":70,"timeslot":120,"enable":1}
Trigger 1: change only the maximum, to 25.5 °C	{"cmd":"trigger","trigger":1,"max":25.5}
Trigger 1 off	{"cmd":"trigger","trigger":1,"enable":0}
Sampling on: both, 6 samples, one every 60 s	{"cmd":"sampling","index":2,"count":6,"sample_time":60,"enable":1}
Sampling off	{"cmd":"sampling","enable":0}
Firmware update	{"cmd":"ota"}
Firmware from your own GitHub	{"cmd":"ota_source","url":"https://raw.githubusercontent.com/<account>/<repository>/main/MS-WBSD-TH-X1.bin"}
Back to Macnman's firmware	{"cmd":"ota_source","url":""}
Link check	{"cmd":"ping"}
New battery fitted	{"cmd":"new_battery"}

Responses

Settings commands reply with all the settings now in effect - not an echo of what you sent - so the reply shows what the device is actually running:

Command	Reply When It Worked
txtime	{"cmd":"txtime","interval":1800,"mode":"heartbeat"}
trigger	{"cmd":"trigger","trigger":1,"index":0,"min":10.00,"max":40.00, "timeslot":60,"enable":1,"mode":"heartbeat+trigger"}
sampling	{"cmd":"sampling","index":2,"count":6,"sample_time":60,"enable":1,"mode":"sampling"}
ota	{"cmd":"ota","status":"updating","source":"default"}, then just before the restart {"cmd":"ota","status":"OTA done","rebooting"}
ota_source	{"cmd":"ota_source","url":"https://raw.githubusercontent.com/<account>/<repository>/main/MS-WBSD-TH-X1.bin","source":"saved"}
ping	{"cmd":"ping","status":"ok"}
new_battery	{"cmd":"new_battery","status":"ok","battery":100}

The mode field shows how the device runs after the command:

Mode	Heartbeat	Triggers	Sampling
heartbeat	Running	Off	Off
heartbeat+trigger	Running	At least one on	Off
sampling	Paused	Off	Running

Firmware update replies:

Status / Error	Meaning
updating	The download is starting. source shows where from: default (Macnman's address) or saved (your ota_source address).
OTA done	The new firmware was downloaded and checked; the device restarts. Its next boot message shows the new fw_version.
failed / begin failed	The firmware file could not be reached or opened - for example a wrong ota_source address.
failed / download failed	The connection was lost, or the download was incomplete.
failed / finish failed	The downloaded firmware did not pass the check.
failed / bluetooth connected	A phone is connected to the device. Disconnect it and send the command again.
failed, with "shifted_to":"default"	The update from your ota_source address failed, so the device cleared that address: the next ota installs Macnman's firmware.

Note:

A failed update replies with "status":"failed" and the reason in an error field, e.g. {"cmd":"ota","status":"failed","error":"begin failed"}.

The device downloads the firmware itself from GitHub (raw.githubusercontent.com) over HTTPS, port 443 - from Macnman's firmware repository, or from your own (Firmware from Your Own GitHub). So besides your MQTT broker, the device's Wi-Fi network must let it reach this address on the internet. The download fails with "begin failed" on a network without internet access, behind a firewall that blocks the address or port 443, where web traffic must go through a proxy server, or where a firewall inspects HTTPS traffic. In that case, ask your IT team to allow the address, or update the device over Bluetooth with the Maya app (Chapter 5, Updates).

The device installs whatever firmware Macnman currently publishes - there is no version check.

The update is refused while the app is connected, and stops if a phone connects during the download. After a failed update the device keeps its current firmware and does not retry: send the command again.

Errors

A rejected setting changes nothing. The reply carries an error field, normally after the settings still in effect:

```
{"cmd":"txtime","interval":3600,"mode":"heartbeat","error":"Minimum uplink interval is 1 minute"}
```

Error	Cause
Minimum uplink interval is 1 minute	interval below 60
Maximum uplink interval is 24 hours	interval above 86400
interval is required	txtime sent without interval
Invalid trigger - only trigger 1 and 2 exist	trigger missing, or not 1 or 2
Invalid parameter - choose Temperature or Humidity	trigger index not 0 or 1
Threshold out of sensor range, and min must not exceed max	min or max outside the sensor range, or min above max
Trigger time must be 10 s to 24 h	timeslot outside 10 to 86400
Invalid parameter - choose Temperature, Humidity or Both	sampling index not 0, 1 or 2
Sample count must be 2 to 12	count outside 2 to 12
Sampling window must be 60 s to 59 min	sample_time outside 60 to 3540
enable must be 0 or 1	enable not 0 or 1
invalid value for <field>	The field is text, true/false, null, or a fraction where a whole number is needed
url must be https://<server>/<file>	ota_source url does not start with https://, or has no server
url too long	ota_source url longer than 226 characters
url must not contain spaces or quotes	A space or a quote in the ota_source url
could not save	ota_source: the device could not store the address; nothing changed

Some errors come back with only the command and the error, without the settings: {"cmd":"<command>","error":"<text>"}.

A message that cannot be run at all gets a short error only:

Reply	Cause
{"error":"retained command ignored"}	Published with retain on
{"error":"command too long"}	Longer than 255 bytes
{"error":"empty command"}	The message is empty
{"error":"invalid JSON"}	Not a JSON object - e.g. plain text
{"error":"missing cmd"}	No cmd field
{"error":"unknown command"}	cmd is not one of the seven commands
{"error":"<n> more rejected commands not answered"}	More rejected commands arrived at once than the device answers one by one; <n> of them got no reply of their own

Several Commands at Once

- **Only the latest of each kind runs** - if several commands are waiting, the device keeps the latest txtime, trigger 1, trigger 2, sampling, ota, ota_source, ping and new_battery, and answers each command it replaced with, for example, {"cmd":"txtime","status":"superseded","count":2}.
- **Kinds run in the order they arrived** - a sampling ON followed by a trigger ON ends in trigger mode.
- **The firmware update always runs last** (ota) - because it restarts the device.
- **Nothing overflows** - commands piling up during a long outage can never fill the queue; the most recent request of each kind always runs.

Checking That a Command Worked

1. Publish {"cmd":"ping"}. When {"cmd":"ping","status":"ok"} appears on the response topic after the next uplink, the whole path works: broker, queue, device and reply.
2. Then publish {"cmd":"txtime","interval":300}. After the next uplink, the heartbeat speeds up to every 5 minutes.

> **Note:**
>
> A reply with an error field means nothing changed.
>
> After a firmware update, the boot message's fw_version shows the firmware now running.
>
> The Maya app shows the same settings: MQTT commands and the app change the same stored values, and whichever was set last applies.
>
> No reply at all? The command may have been lost - for example if the broker expired the device's session - so publish it again. If a "more rejected commands not answered" reply came, the command was rejected: check it against the error tables, then publish it again.

Firmware from Your Own GitHub

By default the device downloads new firmware from Macnman's repository on GitHub. You can keep the firmware file in your own GitHub repository instead - for example to decide yourself when your devices get a new version. Macnman sends you the .bin file.

1. On GitHub, create a new repository and choose **Public**.
2. Click **Add file > Upload files**, add the .bin file and commit it to the main branch.
3. Open the file and click **Raw**. Copy the address from the browser; it starts with https://raw.githubusercontent.com/.
4. Publish {"cmd":"ota_source","url":"<the copied address>"}. The reply repeats the address with "source":"saved".
5. Publish {"cmd":"ota"} and close the Maya app. The reply {"cmd":"ota","status":"updating","source":"saved"} shows that the device uses your file; after the restart, the boot message shows the new fw_version.

For a new version, upload the new file under the same name: the address stays the same, so only {"cmd":"ota"} is needed. Wait about 5 minutes after the upload - GitHub can keep sending the old file for that long.

Note:

The address is saved in each device separately and kept through reboots and firmware updates. A factory reset clears it.

{"cmd":"ota_source"} alone shows the address in use; {"cmd":"ota_source","url":""} goes back to Macnman's firmware.

If an update from your address fails, the device clears the address and adds "shifted_to":"default" to the reply: the next {"cmd":"ota"} installs Macnman's firmware. A "bluetooth connected" failure keeps the address - close the app and publish {"cmd":"ota"} again.

Devices with older firmware do not know ota_source: they reply {"error":"unknown command"} and always use Macnman's address.

What your repository needs:

- **Public** - the device cannot log in to GitHub: a private repository answers "404: Not Found" and the update fails with begin failed.
- **The Raw address** - not the address of the file's GitHub page (…/blob/…), which is a web page, not the file.
- **A normal upload** - not Git LFS; for an LFS file, GitHub sends a short pointer instead of the firmware.
- **Exact names** - the account, repository, branch, folders and file name must match the address exactly, capital letters included, with no spaces. The address can be up to 226 characters.
- **A file under 2 MB** - the size of the device's firmware slot.
- **Not too many devices at once** - GitHub limits downloads without login from one internet address. If many devices behind one router update together, some can fail with begin failed; update them in groups.

Note:

The device installs whatever firmware it finds at the address and does not check that it comes from Macnman. Upload only files from Macnman, and protect your GitHub account and your MQTT broker: whoever can change the file, or publish to the command topic, decides what your devices install.

Chapter 9. Services

Macnman provides customers with timely and comprehensive technical support. End users can contact their local dealer for technical support; distributors and resellers can contact Macnman directly.

Item	Contact
Technical support	support@macnman.com
Website	www.macnman.com

When you contact support, please have ready the Serial No and the hardware and software versions (About > System), the MQTT broker you use, and a description of what you saw - the LED, the Ping result and the last messages on the broker.

MACNMAN TECHNOLOGIES PVT LTD

Chapter 10. Coming Soon

These features are under implementation and will be described in a future edition of this guide. Power Saving and Security & Sharing are already shown in the Maya app (Chapter 5, The Device Screen); TLS will be added to the MQTT settings.

Feature	In the App	What It Will Do
Power Saving Mode	Power Saving	Wi-Fi Power Save Mode: the Wi-Fi radio sleeps between the router's signals while the device stays connected, to use less power.
TWT (Target Wake Time)	Power Saving	A Wi-Fi 6 feature: the device and the router agree when the device will wake, so it can stay connected while sleeping for longer.
Security	Security & Sharing	Choose who can connect to the device from the Maya app.
Share device	Security & Sharing	Share access to the device with other Maya app users.
TLS and certificate validation	MQTT Settings	TLS encrypts the connection to the broker (MQTT over TLS, usually port 8883), so readings and commands cannot be read on the way. Certificate validation checks the broker's certificate, so the device connects only to the genuine broker.

Note:

Until then, the device saves power by sleeping with its radio off between uplinks (Chapter 7).

Until TLS is available, use the broker's plain MQTT port, usually 1883 (Chapter 5, MQTT Settings).

These features will come with updates of the Maya app and the device firmware (Chapter 5, Maintenance).