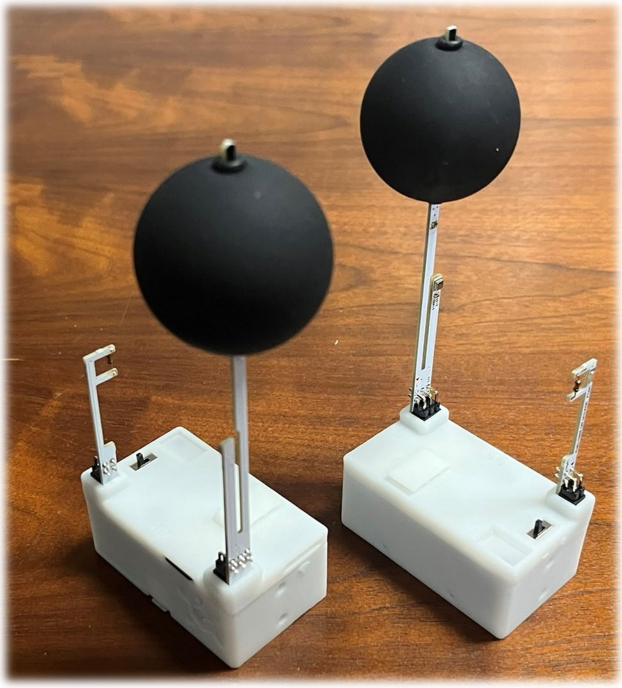
M-Logger System Documentation

5/25/2024



Index

Chapter 1 Introduction

1.1 Overview of the M-Logger System

M-Logger is a wireless measurement and logging device designed to monitor the indoor thermal environment of buildings. The "M" stands for "Mini" and "Makunouchi," signifying its compact size that fits in the palm of your hand while being packed with capabilities to measure various physical quantities such as dry-bulb temperature, relative humidity, globe temperature, low air velocity, and illuminance. "Makunouchi" is a Japanese term meaning a bento box (Japanese lunch box) with a variety of items. Below are its main features:

- All specifications and technical documents are freely available on the web (open-source hardware).
- Compact size of 64×40×30 mm.
- Capable of measuring dry-bulb temperature, relative humidity, globe temperature, and low air velocity necessary for evaluating thermal comfort.
- Can calculate and display thermal comfort indices (PMV, PPD, SET*) in real-time.
- Measures illuminance to infer room conditions.
- Runs on two AA batteries (no need for a power outlet).
- Can perform continuous measurements for several weeks on batteries (over one month without measuring low air velocity).
- The cost per unit is less than $100.
- Initial setup and data collection can be done wirelessly.
- Compatible with Windows, Mac, and Linux.
- Can also be used with a smartphone (iPhone, Android).

1.2 Structure of this document

Chapter 2 describes the specifications of M-Logger and provides an overview of the measurement system using it. Chapter 3 explains how to use M-Logger with a smartphone. Chapter 4 details how to use M-Logger with a PC, such as Windows. Chapter 5 covers how to build a system for remotely monitoring measurement results using Windows or Raspberry Pi. Chapter 6 explains the communication protocol with M-Logger.

M-Logger is open-source hardware with all development information publicly available. In addition to this document, more information is available on the website (https://www.mlogger.jp). Please refer to the website as needed.

Chapter 2 How to Use the M-Logger System

2.1 Configuration of the Thermal Environment Measurement System

The M-Logger system consists of a "sensor unit (M-Logger)" that measures physical quantities such as temperature and humidity, and a "control unit" that manages the measurement settings of the sensor units and stores the measured data. The sensor units and the control unit are connected via wireless communication (Fig. 2.1). The control unit can be a smartphone or a personal computer (PC).

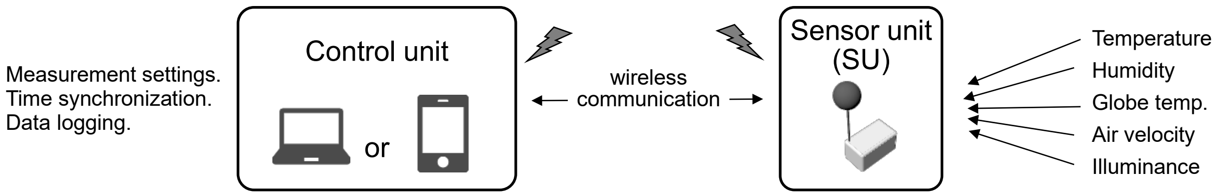


Fig. 2.1 Configuration of the M-Logger system

When using a smartphone as the control unit, the control unit and the sensor units communicate via Bluetooth Low Energy (BLE). In this case, only one sensor unit can be connected (Fig. 2.2). The measurement method for this configuration is explained in Chapter 3.

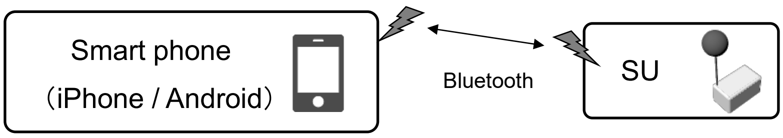


Fig. 2.2 Configuration when using a smartphone as the control unit

When using a PC as the control unit, it communicates with the sensor units via Zigbee. In this case, multiple sensor units can be connected simultaneously (Fig. 2.3). The measurement method for this configuration is explained in Chapter 4.

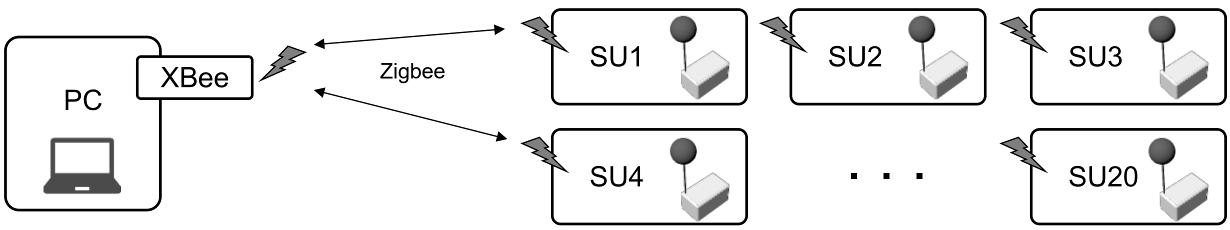


Fig. 2.3 Configuration when using a PC as the control unit

Setting up a PC as the control unit is slightly more complex compared to using a smartphone. Therefore, if you want to easily monitor the indoor thermal environment, it is recommended to use a smartphone. For more comprehensive multi-point measurements, using a PC is advisable.

2.2 Structure of the Sensor Unit (M-Logger)

The sensor unit can measure dry-bulb temperature, relative humidity, globe temperature, low air velocity, and illuminance. The components of the sensor unit are shown in Fig. 2.4.

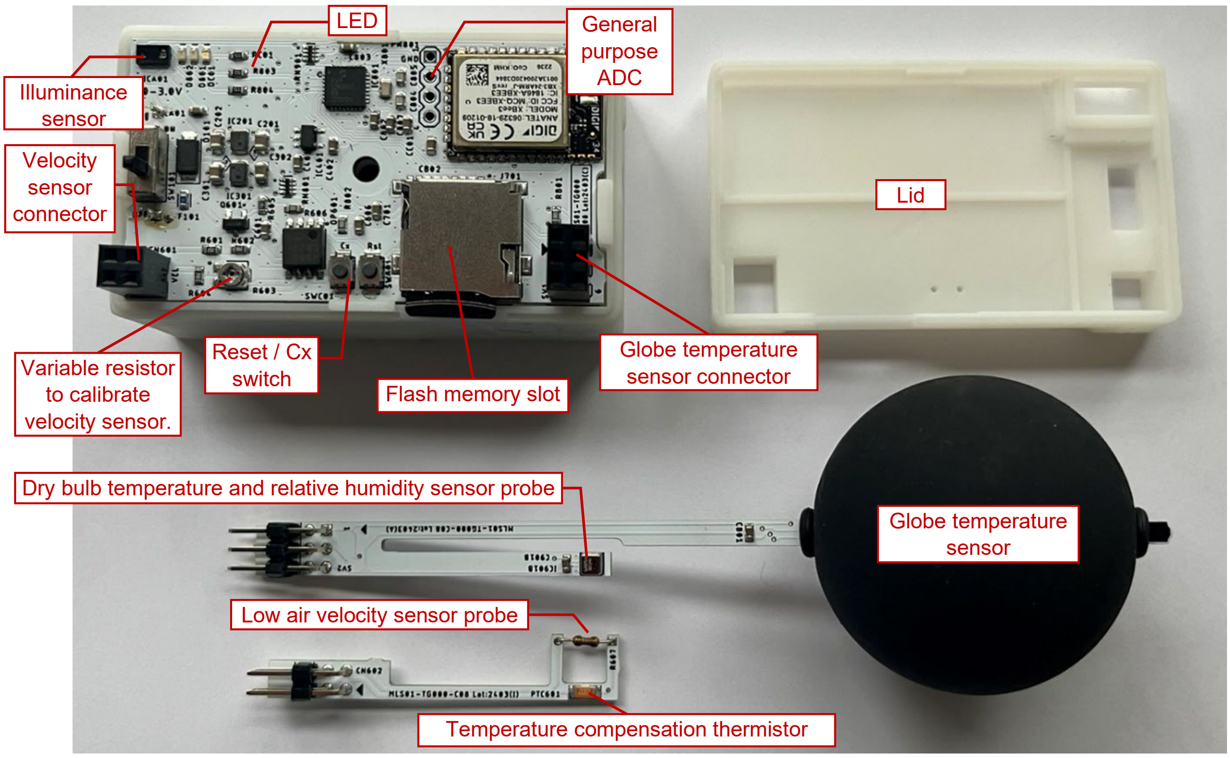


Fig. 2.4 The components of the sensor unit (M-Logger)

The globe temperature sensor and low air velocity sensor are separate from the main board. When transporting the sensor unit, they are secured by hooking them onto the tabs on the lid, as shown in Fig. 2.5. During measurement, they are inserted into the connectors, as shown in Fig. 2.6.

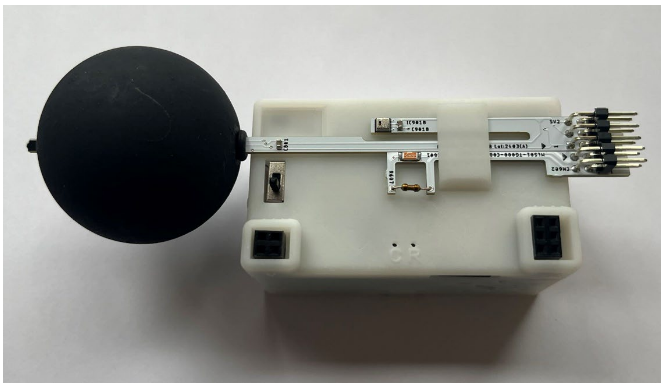


Fig. 2.5 Hooking sensors onto the tabs

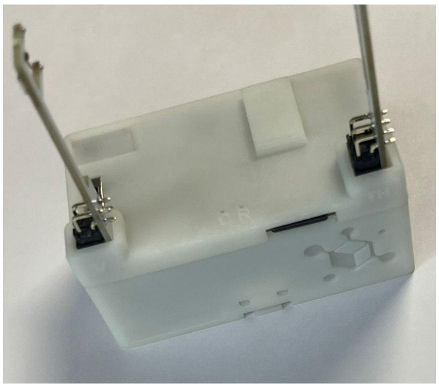


Fig. 2.6 Inserting sensors into the connectors

The illuminance sensor is directly attached to the main board and is located under the translucent acrylic plate at the corner of the lid. Therefore, when measuring illuminance, be careful not to cover this part.

The variable resistor is used to calibrate the low air velocity sensor. Therefore, do not rotate it to change the resistance value.

After inserting the low air velocity sensor and the globe temperature sensor, insert AA batteries into the battery box on the back and turn on the switch. The LED will start blinking at one-second intervals, indicating that the sensor unit is ready to receive communication from the control unit. The LED will blink twice if a flash memory card is inserted and recognized, otherwise, it will blink once. When measurement starts successfully upon receiving a command from the control unit, the LED will change to blinking once every five seconds.

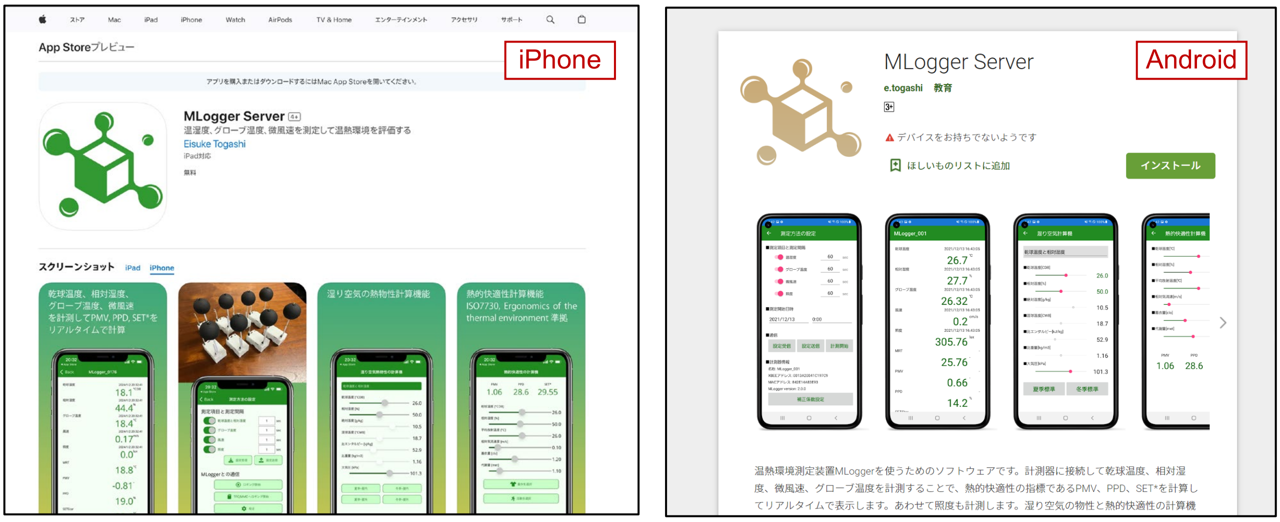
When measuring low air velocity, the temperature of the low air velocity probe will rise (up to about 40°C, which is not dangerous). Be careful not to let objects touch the probe, as this can cause incorrect readings.

The operating time of the sensor unit on batteries depends on the measurement settings. The most power-consuming component is the low air velocity probe, which requires heating. In tests using fully charged Eneloop Pro batteries, the measurement intervals of 1 min, 5 min, 15min, and 1 hour resulted in measurement times of 90 hours, 250 hours, 350 hours, and 420 hours, respectively. However, these times vary depending on the ambient temperature around the anemometer, with lower temperatures causing faster battery consumption.

Chapter 3　Measurement Using Smartphone

3.1 Preparations

When using a smartphone as the control unit, there is no need to add any additional hardware. For iPhone, simply connect to the Apple Store, and for Android, connect to Google Play, search for *MLogger Server*, and download and install the software to start measurements immediately (Fig. 3.1).



Apple Store:　https://apps.apple.com/us/app/mlogger-server/id1599907037

Google Play:　https://play.google.com/store/apps/details?id=net.hvacsimulator.mls

Fig. 3.1 Downloading the software

The control unit and sensor units communicate using Bluetooth Low Energy (BLE), so you need to grant the software permission to use BLE in the smartphone settings (Fig. 3.2). Particularly for Android, note that BLE will not be enabled unless "Location" is also permitted.

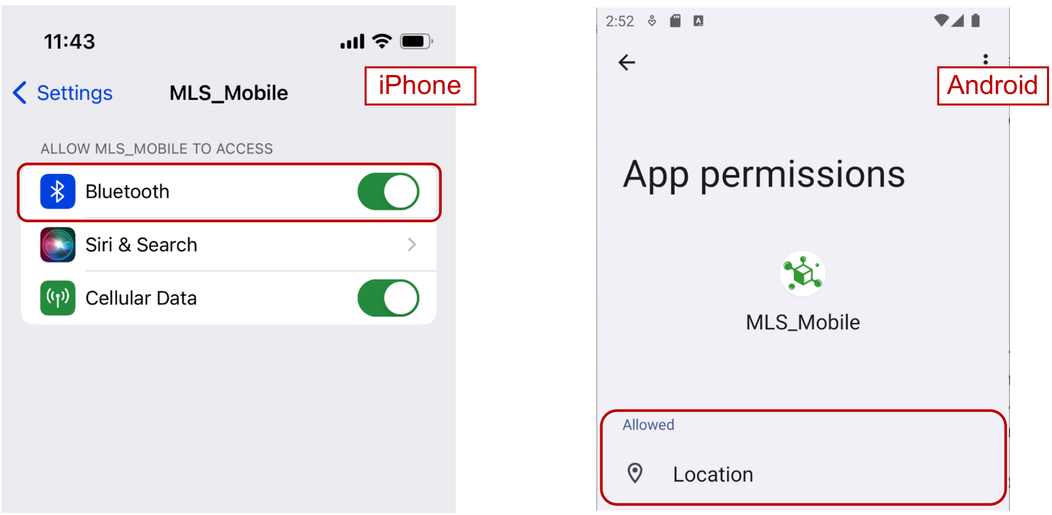


Fig. 3.2 Granting permission to use Bluetooth LE communication

3.2 How to use the software

1) Overview

The software has four functions, which can be switched using the tabs at the bottom (Fig. 3.3). The first function is scanning the sensor units (Fig. 3.3.(1)), the second is viewing and managing the logged data (Fig. 3.3.(2)), the third is calculating thermal comfort (Fig. 3.3.(3)), and the fourth is calculating psychrometrics (Fig. 3.3.(4)). Of these, the third and fourth functions do not require the sensor units and are designed to assist with the measurement tasks.

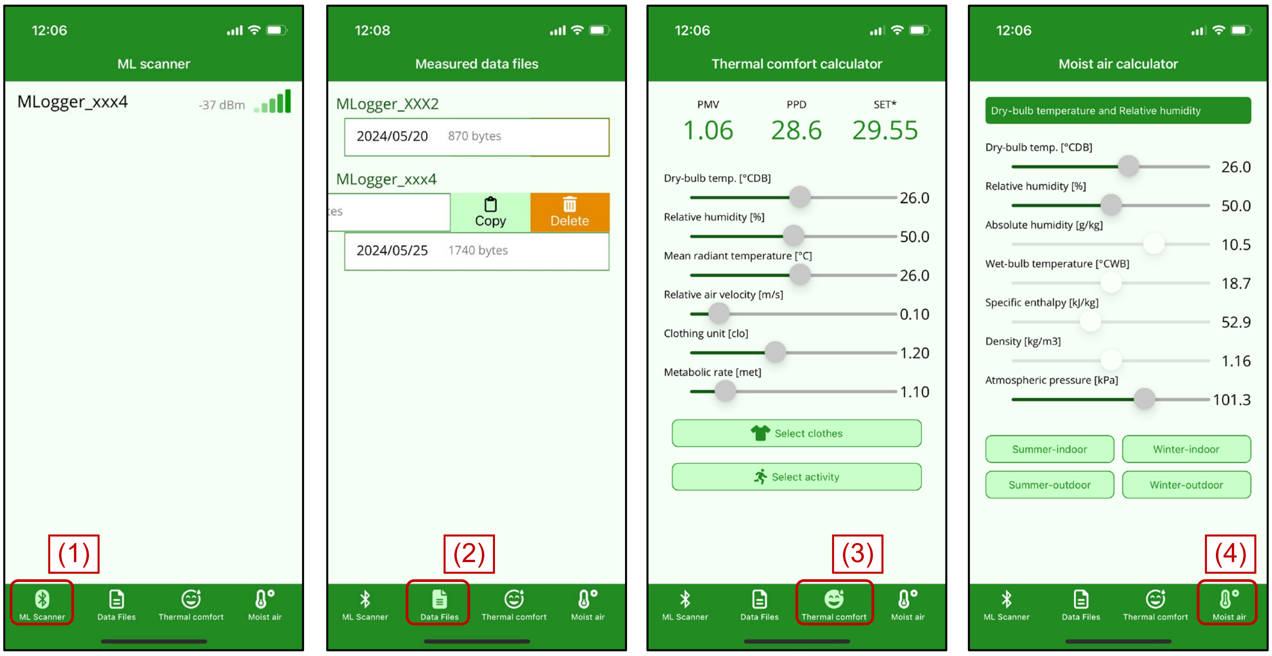


Fig. 3.3 Four functions of the software

2) Measurement settings

In the *ML Scanner* tab, pulling down the screen will search for new sensor units and update the list of discovered units (Fig. 3.4.(1)). Tapping on one of the listed sensor units will take you to the settings screen. Each sensor unit stores measurement-related settings internally, and you can change these settings on this screen.

At the top, there is a *Target and time interval* section (Fig. 3.4.(2)). Here, you can set whether to measure dry-bulb temperature, relative humidity, globe temperature, air velocity, and illuminance, as well as the measurement intervals and start date and time.

These settings are saved internally in the sensor unit. Tapping *Load* will retrieve the settings stored in the sensor unit. Conversely, if you change the settings and tap *Save*, the new settings will overwrite the sensor unit's settings. If you change the settings to values different from those stored in the sensor unit, the items will be displayed in red (Fig. 3.4.(3)).

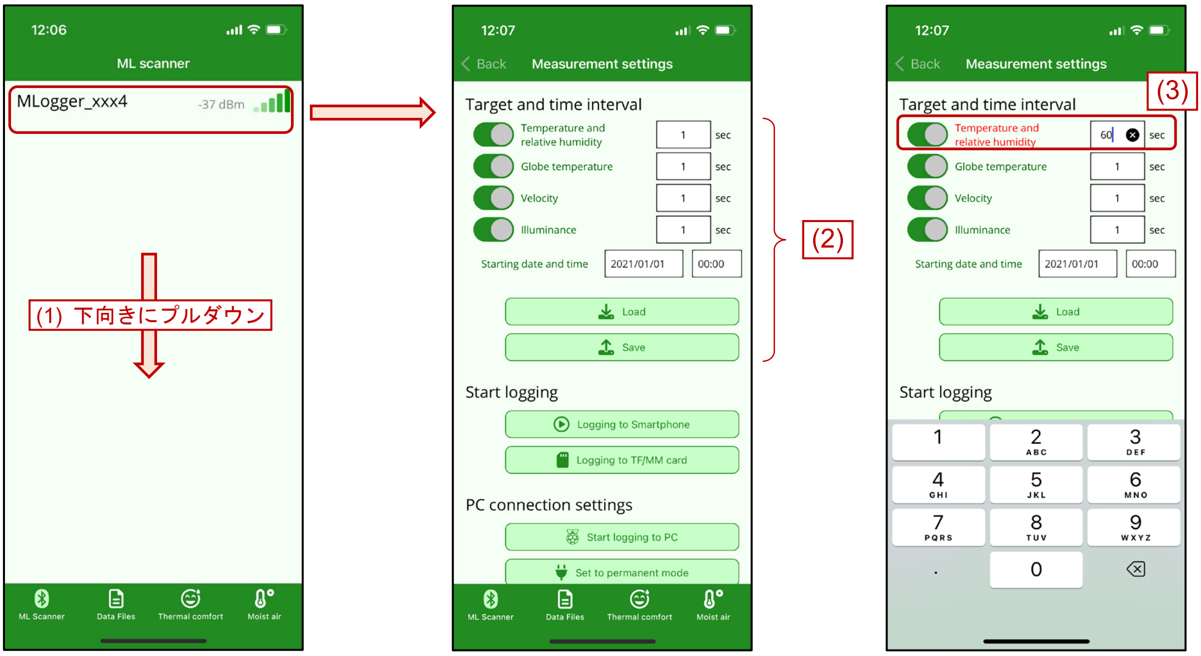


Fig. 3.4 Connecting to sensor unit

3) Record measured data

The values measured by the sensor unit can be saved in two ways: either wirelessly transmitted to the smartphone or recorded on a flash memory inserted into the sensor unit.

To send the data to the smartphone, tap *Logging to Smartphone* button (Fig. 3.5.(1)).

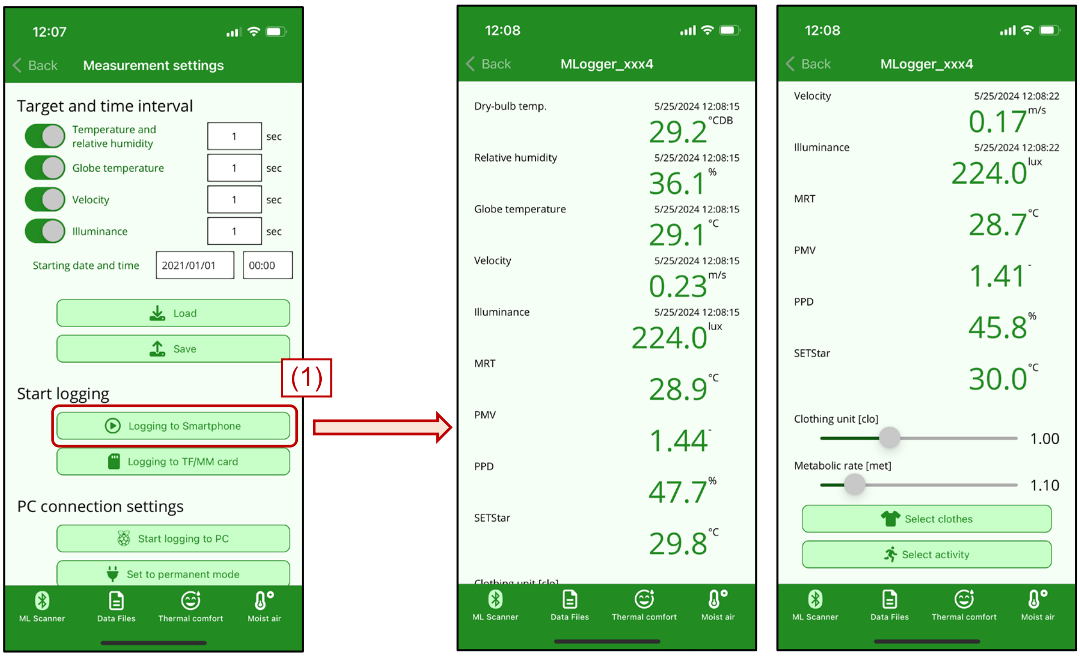


Fig. 3.5 Logging measured data to smartphone

The screen will switch, displaying the measured values for dry-bulb temperature, relative humidity, globe temperature, air velocity, and illuminance. Additionally, the calculated values for MRT, PMV, PPD, and SET* based on these measurements will be shown. The LED on the sensor unit will change to blinking once every five seconds.

To calculate thermal comfort, it is necessary to assume values for clothing units and metabolic rate. To change these values, tap *Select clothes* or *Select activity* button. Further instructions will be provided later.

Tap *Back* button to return to the previous screen and stop the measurement. Once the measurement is successfully stopped, the LED on the sensor unit will return to blinking once per second.

4) Viewing logged data

The data received by the smartphone is automatically logged and can be viewed in the *Data files* tab. In the *Data files* tab, the data is listed by sensor unit ID and date. Swiping an item to the left allows you to copy or delete the data (Fig. 3.6.(1)). Tapping an item displays the data in a table format (Fig. 3.6.(2)). On this screen, you can copy or delete the data by tapping the corresponding buttons (Fig. 3.6.(3)). The data copied to the clipboard is in CSV format, which can be edited using spreadsheet software.

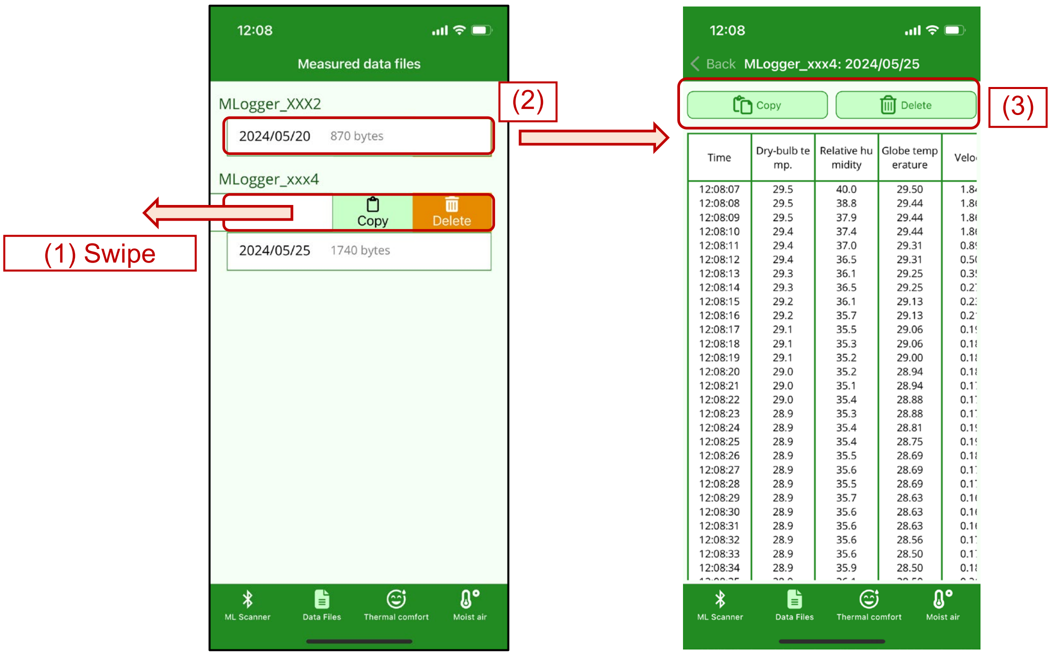


Fig. 3.6 Viewing of logged data

5) Logging on flash memory

The flash memory card types supported are TF (Trans Flash) cards, MM (Multi Media) cards, or compatible cards. To log data to these cards, you first need to ensure the sensor unit recognizes the card. Insert the card into the slot shown in Fig. 3.7. When the card is correctly recognized, the LED will change from blinking once to blinking twice. Note that only the FAT32 file format is supported. SDXC cards larger than 32GB are typically formatted with exFAT, so if using such a card, you will need to format it to FAT32 using a tool like Windows.

Once the card is correctly recognized, tap *Log to TF/MM Card* button (Fig. 3.7.(1)). When logging to the card begins, the connection with the smartphone will be terminated, and you will be returned to the initial screen (sensor unit list). The sensor unit will enter sleep mode as much as possible to conserve battery. Therefore, if you want to reconnect with the smartphone, you will need to turn off the sensor unit and restart it.

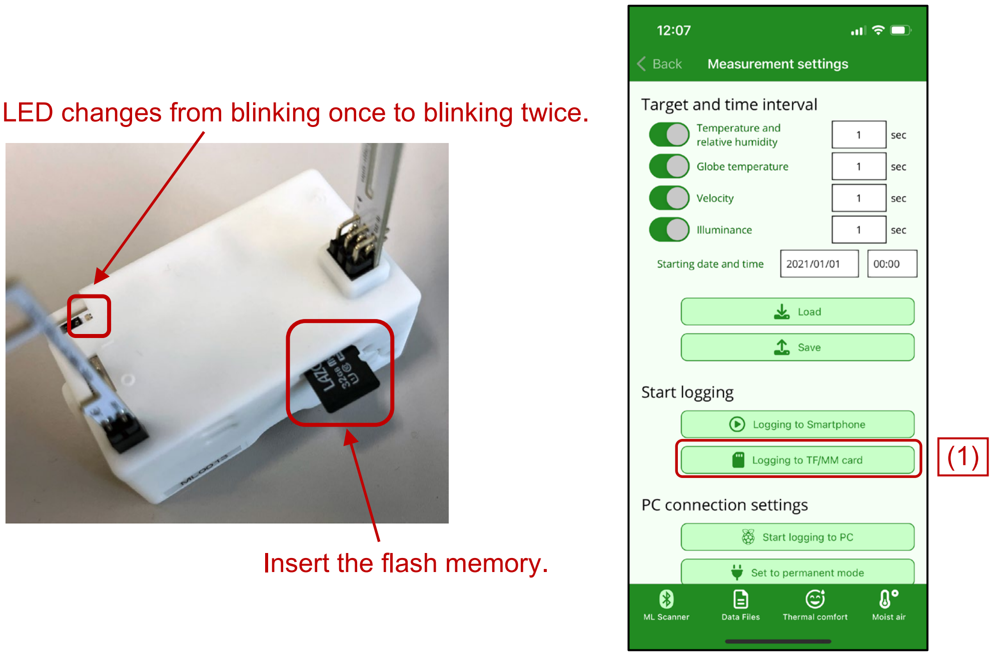


Fig. 3.7 Logging on flash memory

6) Connecting to PC

The *PC connection settings* section is used when a PC is the control unit. This will be explained later in Chapter 4.

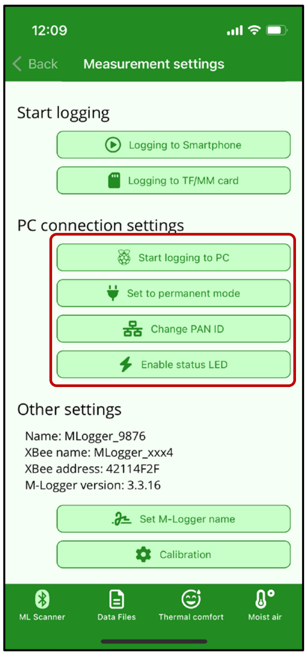


Fig. 3.8 PC connection settings

7) Setting name of the sensor unit

The sensor unit has a user-configurable *Name* in addition to its ID (*XBee name*). You can change this *Name* by tapping *Set M-Logger name* button (Fig. 3.9.(1)).

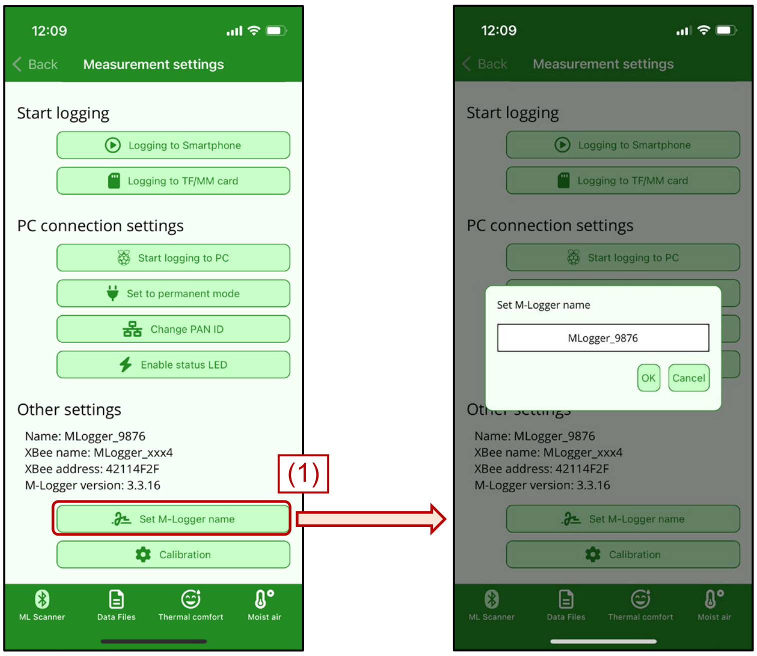


Fig. 3.9 Setting name of the sensor unit

8) Calibration

The product comes with basic calibration, except for illuminance, but you can also calibrate it yourself. When you tap *Calibration* button, several calibration options will appear (Fig. 3.10.(1)). Tapping *Set Correction Factor* button allows you to set correction factors for linear calibration in the form of "y=Ax+B" (Fig. 3.10.(2)). Tapping *Save* button sends the correction factors to the sensor unit, and tapping *Load* button retrieves the current correction values.

Automatic Temperature Calibration (Fig. 3.10.(3)) is a function used to match the values of the dry-bulb temperature sensor and the globe temperature sensor. If you place the measuring device in a thermally stable box with sufficient insulation, the values of the dry-bulb temperature sensor and the globe temperature sensor should be nearly identical. When you tap *Automatic Temperature Calibration* button with the measuring device inside such a box, the measurement will continue for the duration set as the *Calibrating Time*. After completion, the correction factor will be automatically set to make the globe temperature sensor's value match the dry-bulb temperature sensor's value.

Automatic Velocity Calibration (Fig. 3.10.(4)) is a function used to initialize the voltage of the anemometer when there is no air flow. As shown in Fig. 3.11, covering the probe of the anemometer with a low-emissivity metal clip and sealing the ends with tape or similar material to block the air flow will result in a nearly 0 m/s reading. In this state, tapping *Automatic Velocity Calibration* will measure the low air velocity for the duration set as the *Calibrating time*, and the voltage at no air flow will be initialized. The initial voltage for no air flow is set to 1.45V (Fig. 3.10.(5)).

However, this is a simplified calibration method. If you have access to test equipment that can create a stable air flow, we recommend the manual calibration method described below.

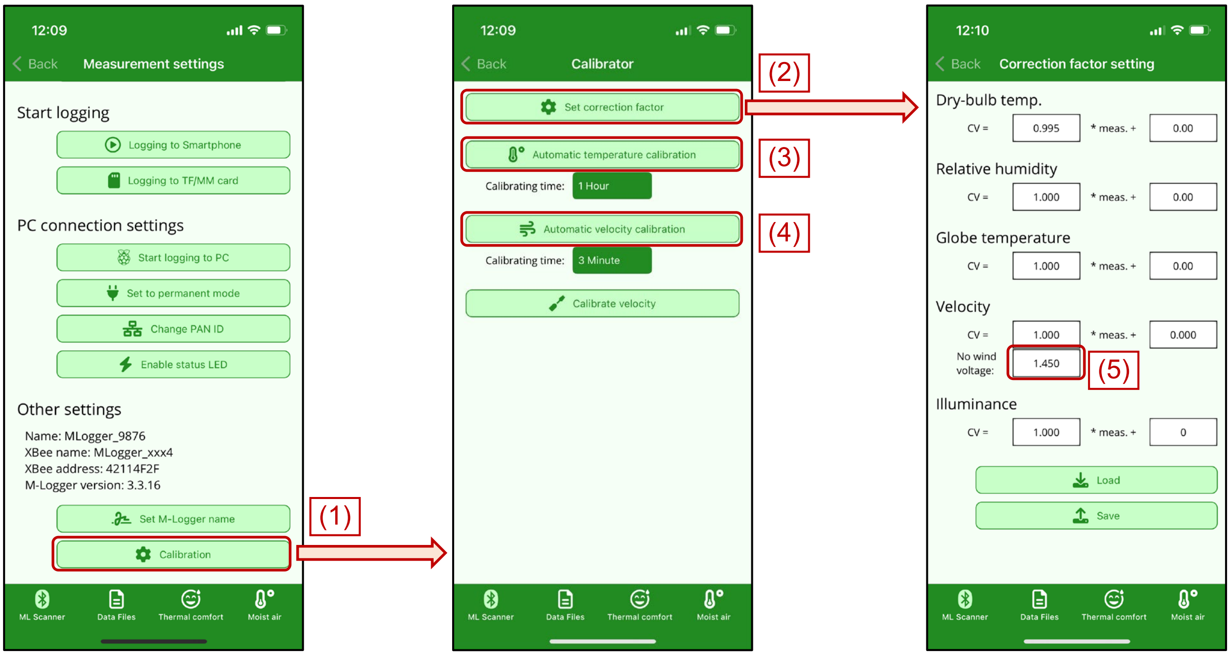


Fig. 3.10 Calibrating the sensor unit

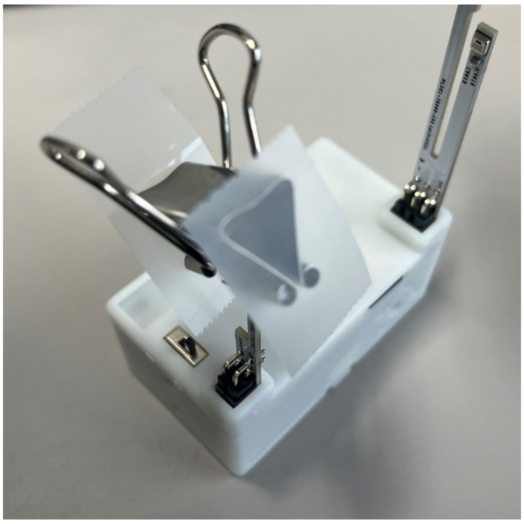


Fig. 3.11 Create windless conditions

When you tap *Calibrate Velocity* (Fig. 3.12.(1)), the measurement of air velocity begins, and both the instantaneous and 10-second average values for air velocity and voltage are displayed. The anemometer initially shows extremely high values upon startup, so it requires some time to stabilize. Therefore, the elapsed time since startup is displayed at the top of the screen. The measured values generally stabilize within about 30 seconds. Rotate the variable resistor (see Fig. 2.4) on the board to match the air velocity value to the velocity created by the test equipment. Turning the variable resistor to the right decreases the measured value, while turning it to the left increases it.

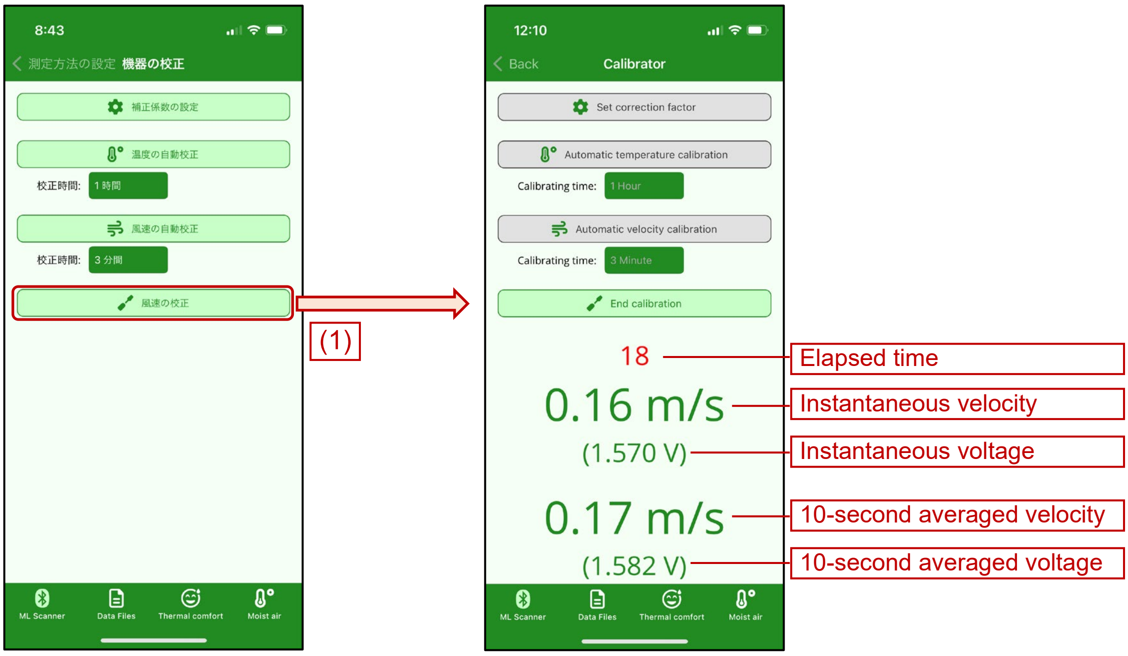


Fig. 3.12 Manual velocity calibration

9) Thermal comfort calculation function

In the *Thermal Comfort* tab, you can calculate PMV, PPD, and SET* by specifying the six thermal comfort factors. By tapping the *Select clothes* or *Select activity* buttons, you can specify the type of clothing or activity on a separate screen. For clothing, select one or more items and tap *Apply*. For activity, simply tap an item, and the corresponding Clothing unit [clo] and metabolic rate [met] will be applied.

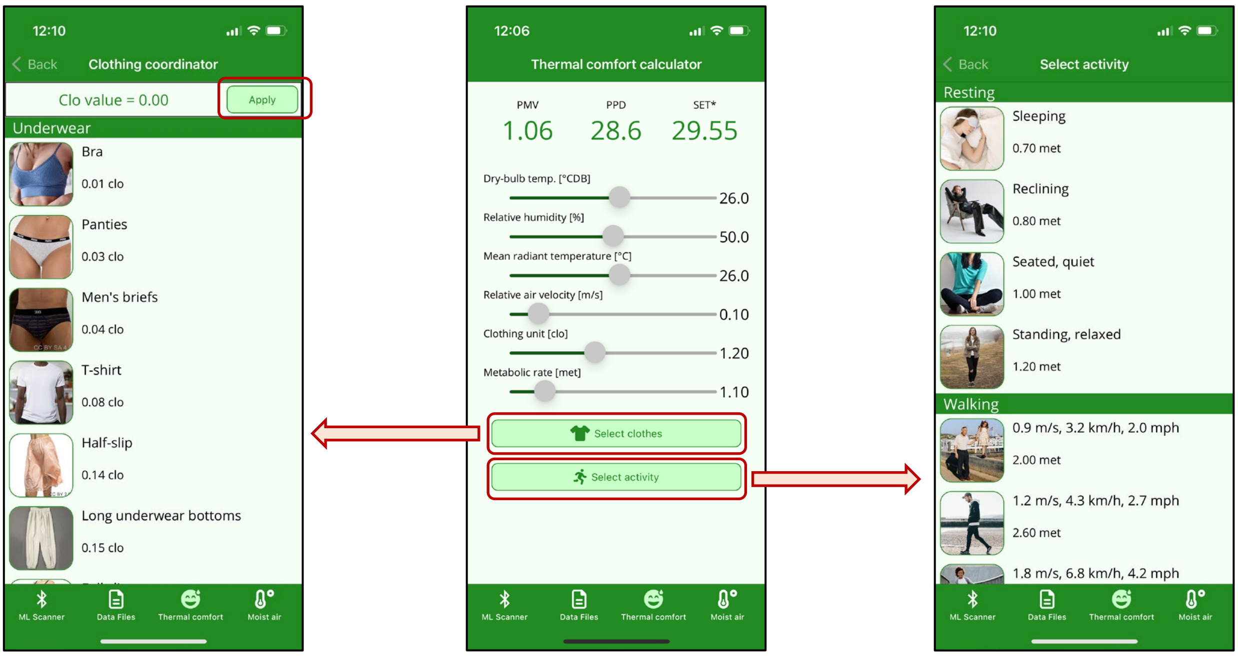


Fig. 3.13 Thermal comfort calculation function

10) Psychrometrics calculation function

In the *Moist Air* tab, you can calculate the remaining properties of moist air from two given properties. By selecting from the combo boxes at the top, you can choose the combination of two input properties (Fig. 3.14.(1)). Tapping the button at the bottom will reset the conditions to typical temperature and humidity values (Fig. 3.14.(2)).

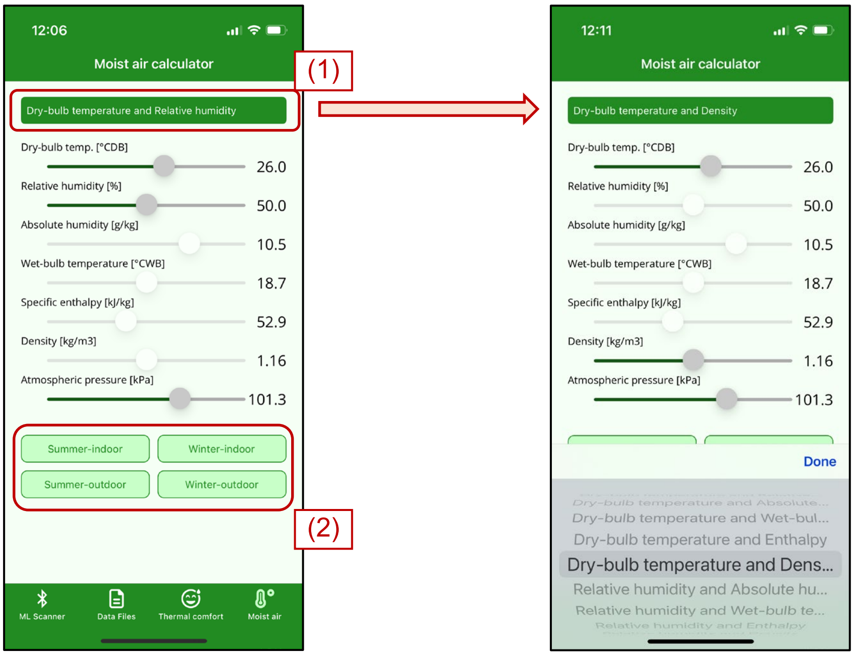


Fig. 3.14 Psychrometrics calculation function

Chapter 4 Measurement Using PC

4.1 Preparation

1) Hardware

When using a PC as the control unit, data from the sensor units is received via Zigbee communication using XBee module. Therefore, you need to attach and configure the XBee on the PC. Typically, the XBee is connected using a USB port, as shown in Fig. 4.1.

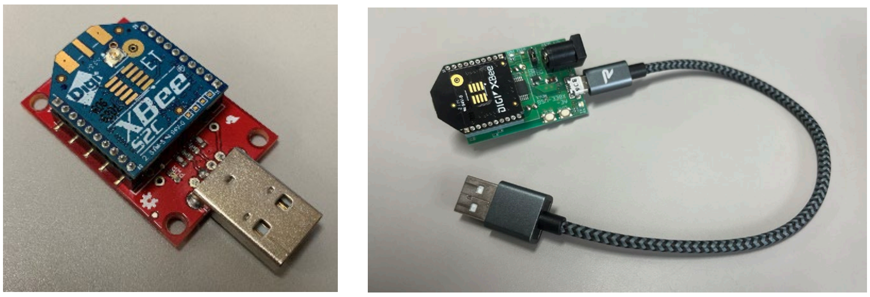


Fig. 4.1 USB connected XBee module

Currently, most versions of XBee available on the market are S2C and 3.0, both of which can be used. Additionally, the following USB adopter module have been tested for compatibility.

https://akizukidenshi.com/catalog/g/gK-06188
https://flashtree.com/products/11697

Typically, to use a USB adapter module, you need to install the D2XX driver from FTDI (www.ftdichip.com). For more details, please refer to the manual for each adapter module.

XBee has various settings, and the default settings at the time of purchase will not allow communication with the sensor units. To change these settings, you need to install the XCTU software (Fig. 4.2) provided by Digi, the manufacturer of XBee.

https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu

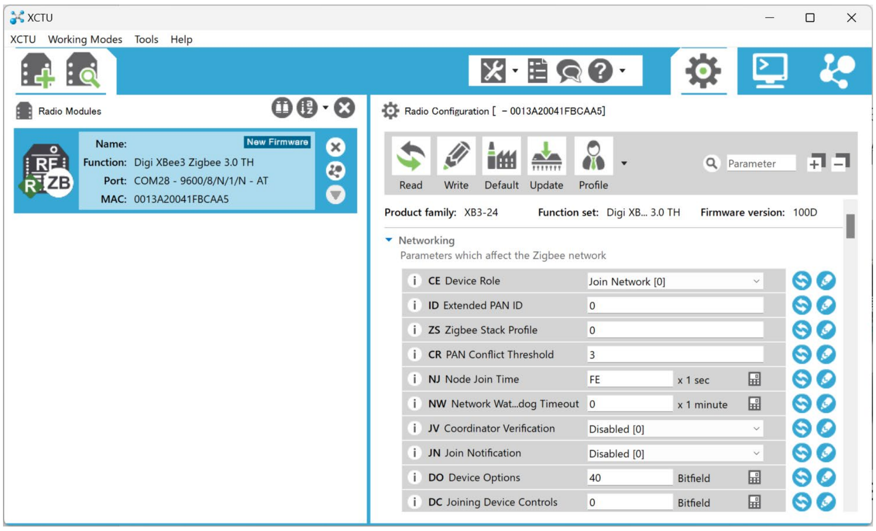


Fig. 4.2 XCTU

The settings of the items shown in Table 4.1 must be changed.

Table 4.1 Parameter values to be changed

Parameter		Value
ID	PAN ID	19800614
SP	Cyclic Sleep Period	0x64 (=1000 msec)
SN	Number of Cyclic Sleep Periods	3600 (sec)
CE	Coordinator Enable	Enabled
SM	Sleep Mode	No sleep
AP	API Enable	API enabled

Each parameter item is explained below.

- ID

Each Zigbee network has a unique ID called a *PAN ID*. Regardless of distance, devices with different PAN IDs do not communicate with each other. In the M-Logger system, the standard PAN ID used is "0x19800614" in hexadecimal. If you want to create multiple networks using the M-Logger system at the same measurement site, you will need to change this value for each network.

- SP

This is the time the control unit holds the message for the sensor unit. The sensor unit typically sleeps for up to 1 second during communication, so this value is set to 1 second to match.

- SN

The network between the control unit and the sensor units is maintained for three times the value of SN. Therefore, once the connection is established, the network will continue for 3 hours unless a specific command is given.

- CE

In XBee terminology, the control unit is called the "Coordinator." "CE" is the setting for determining whether it is a control unit; setting it to “Enabled” makes it the control unit, while setting it to “Disabled” makes it a sensor unit.

- SM

This setting determines whether the XBee enters sleep mode. The control unit should not enter sleep mode and must always remain active, so set it to “No Sleep.”

- AP

This setting indicates the communication method between XBee devices. Since the sensor units communicate using API by default, the control unit should also be set to "API enabled" to match the sensor units.

According to XBee specifications, a Coordinator can connect to up to 20 XBee devices. Therefore, by default, the control unit can only connect to 20 sensor units. However, you can increase this number by adding XBees configured as "Routers." Each Router can manage up to 20 XBee devices, so adding one Router increases the number of connectable sensor units by 20.

To configure an XBee as a Router, use the same settings as shown in Table 4.1 and change only the "CE: Coordinator Enable" to "Disabled." As shown in Fig. 4.3, there are also XBees available that can be plugged directly into a power outlet. This is convenient when using them as Routers.

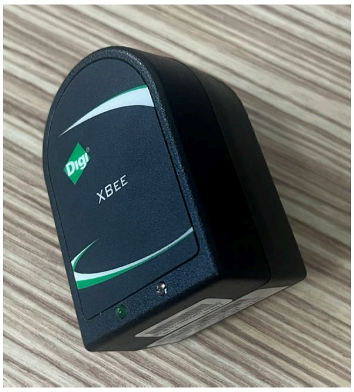


Fig. 4.3 XBee Wall Router

(https://akizukidenshi.com/catalog/g/gM-10502)

Increasing the number of Routers allows for more sensor units, but it also increases the risk of wireless congestion and data loss. In tests conducted in a small room, the limit was collecting measurements from 80 sensor units at 3-second intervals. Additionally, the risk of data loss depends on the surrounding wireless conditions; for example, significant communication interference can occur when a microwave oven is in use. Therefore, it is necessary to adjust the settings according to the conditions of the actual measurement site.

2) Software

The latest version of the software for the control unit can be downloaded from the website (https://github.com/et0614/mlogger/releases). Since it runs on Microsoft .NET 8.0 or higher, it works not only on Windows PCs but also on Mac and Linux. For example, using a Raspberry Pi, you can create a compact thermal environment measurement system on site. The .NET runtime can be downloaded from Microsoft's website if needed.

https://dotnet.microsoft.com/download/dotnet

When you extract the downloaded zip file, you will get the directory shown in Fig. 4.4. The main software is *MLServer.exe,* and there are also several other programs included to assist with measurements.

```
MLServer                    : Top directory
 ├─ data
 │   ├ index.htm            : Top page of the Web server
 │   ├ XBeeAddress.csv      : Logged data (Separate csv files for each sensor unit)
 │   └ .........            : Other files used by the web server
 ├─ SimpleWebServer.exe     : Simple web server software
 ├─ MLServer.exe            : Main software
 ├─ mlnames.txt             : File for setting the name of the sensor units
 ├─ DDNSUpdater.exe         : Software for using DDNS service
 ├─ setting.ini             : Initial setting file for data logging software
 ├─ ddns.ini                : Initial setting file for DDNS service
 └─ .........               : Other files required for execution
```

Fig. 4.4 Directory structure of MLServer

4.2 How to use the software

Connect the XBee to the PC via USB as explained in the previous section, and launch *MLServer*. The startup screen is shown in Fig. 4.5.

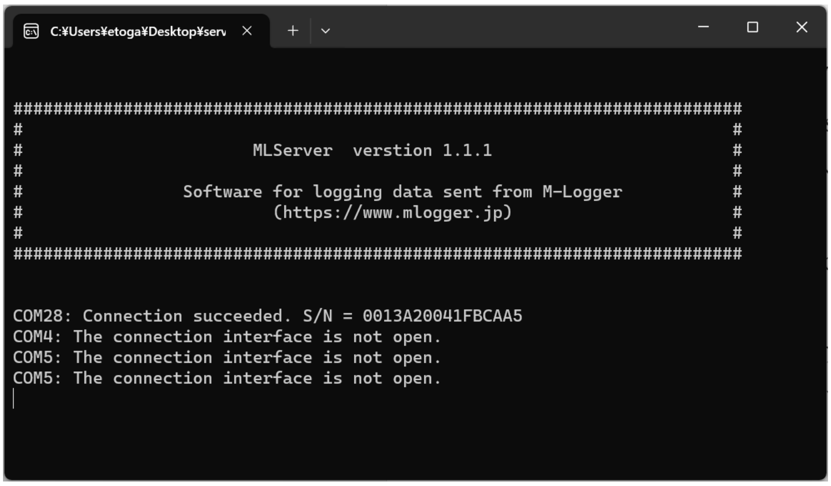


Fig. 4.5 Startup screen of the MLServer

When the XBee is recognized via USB, the following message will be displayed:

COM*n*: Connection succeeded. S/N = 0013A200*xxxxxxxx*

Here, the value of *n* in COM*n* varies depending on which port is used by the PC. The "0013A200*xxxxxxxx*" represents the unique address of the XBee, and the value of *x* will be different for each XBee.

With this, the XBee for the control unit is ready for communication. Now, once the sensor units are powered on, communication will begin.

In the *PC connection settings* section of the smartphone screen, tap *Start logging to PC* button (Fig. 4.6.(1)) to have the sensor units start sending measurements to the control unit. When the control unit receives data from the sensor units, the information is transferred to the *MLServer* screen, as shown in Fig. 4.7. The figure shows data being received from a sensor unit with the XBee address "4207BD4A."

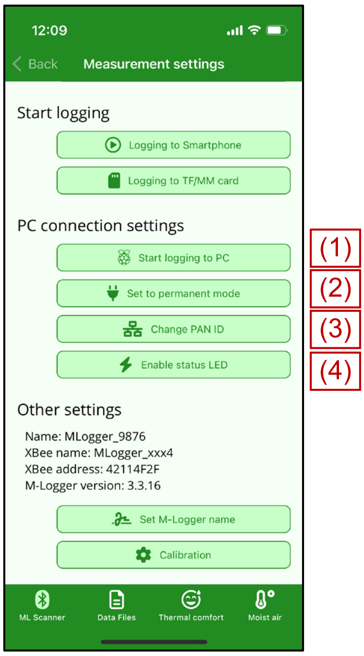


Fig. 4.6 PC connection settings

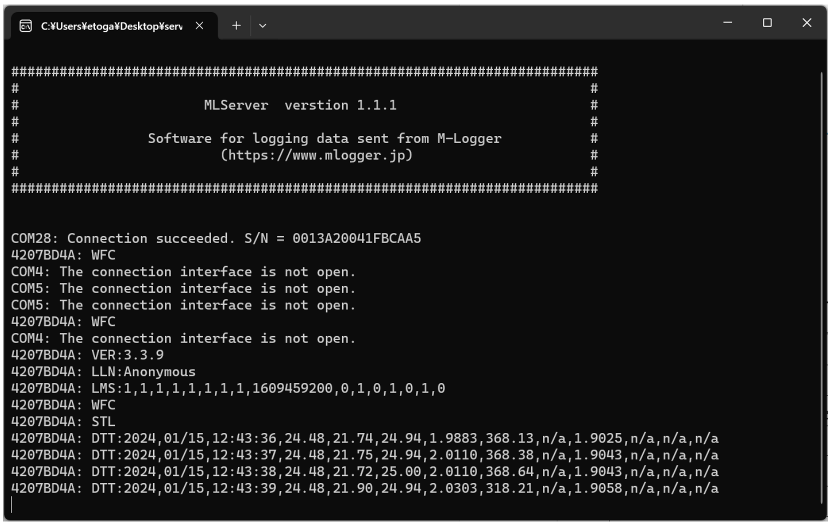


Fig. 4.7 Receiving data from sensor units

The data received by the control unit is saved as a CSV file in the *data* directory. Table 4.2 shows an example of the measurement data. The information represented by each column, from left to right, is the date and time the control unit received the data, the date and time the sensor unit measured the data, dry-bulb temperature [°C], relative humidity [%], globe temperature [°C], low air velocity [m/s], illuminance [lx], NaN, voltage for low air velocity measurement [V], and general-purpose ADC voltage [V].

Table 4.2 Example of the measurement data

2024/1/15 12:50	2024/1/15 12:50	24.51	21.30	24.94	0.1706	371.46	NaN	1.579	0
2024/1/15 12:50	2024/1/15 12:50	24.51	21.28	24.88	0.1612	372.22	NaN	1.572	0
2024/1/15 12:50	2024/1/15 12:50	24.50	21.34	24.88	0.1544	371.97	NaN	1.567	0
2024/1/15 12:50	2024/1/15 12:50	24.49	21.31	24.88	0.1697	371.97	NaN	1.578	0
2024/1/15 12:50	2024/1/15 12:50	24.51	21.30	24.88	0.1614	371.46	NaN	1.572	0
2024/1/15 12:50	2024/1/15 12:50	24.51	21.30	24.94	0.1626	371.71	NaN	1.573	0
2024/1/15 12:50	2024/1/15 12:50	24.51	21.24	24.88	0.1616	370.94	NaN	1.573	0
2024/1/15 12:50	2024/1/15 12:50	24.52	21.23	24.88	0.1450	371.71	NaN	1.560	0
2024/1/15 12:50	2024/1/15 12:50	24.51	21.25	24.94	0.1727	371.71	NaN	1.580	0
2024/1/15 12:50	2024/1/15 12:50	24.52	21.29	24.88	0.1796	371.97	NaN	1.585	0

When conducting long-term measurements on-site, the sensor unit's battery may run out during the measurement process. Resetting the settings every time the battery is replaced can be very troublesome. Therefore, a mode called *permanent mode* is available, which allows the sensor unit to resume measurements immediately with the previously configured settings when turned back on after the battery has been replaced.

Tap *Set to Permanent Mode* button (Fig. 4.6.(2)) to enable this mode. In this mode, the sensor unit will start measuring as soon as it is turned on. However, the sensor unit will no longer accept commands from the smartphone. To make the sensor unit operable from the smartphone again, you need to forcibly exit this mode. To do this, press and hold the reset switch shown in Fig. 2.4 for more than 3 seconds. The LED will blink three times, indicating that permanent mode has been disabled.

Note that when the sensor unit is turned off, the date and time are reset. Therefore, when restarting in permanent mode, use the first column in Table 4.2 for the date and time instead of the second column.

When sensor units are permanently installed for long-term measurements, replacing the batteries can be troublesome. If you can use a power outlet, an alternative is to use dummy batteries.

Dummy batteries are AC adapters with battery-shaped terminals, as shown in Fig. 4.8. Since the sensor unit has a built-in DC/DC converter, it will operate normally with an internal voltage boost if supplied with 2.0 to 3.0V. Therefore, make sure to use an AC adapter that provides voltage within this range.

Fig. 4.8 Dummy batteries

The *Change PAN ID* button in Fig. 4.6.(3) is used to change the network-specific ID explained in Table 4.1. The default value is "0x19800614," and usually, there is no need to change it.

The *Enable Status LED* button in Fig. 4.6.(4) allows you to set whether to display the communication status of the XBee with a blue LED.

When conducting on-site measurements, it is convenient to be able to remotely check the measurement status. Therefore, this chapter explains how to build a remote monitoring system using a PC.

1) Directory to be published on the web server

As explained in the previous chapter, *MLServer* saves the measured data as CSV files in the *data* directory. Therefore, if you can make this directory accessible via a web server, you will be able to check the measurement results remotely.

In addition to the CSV files, an HTML file (index.htm) representing the current measurement status is also generated in the *data* directory. When you open this file in a browser, a page like Fig. 5.1 will be displayed. This page shows the addresses and names of the sensor units, the measured values, the last measurement time, and thermal comfort indices (SET*, PMV, PPD) calculated from these physical quantities.

最終接続	測定器名▼	アドレス	計測日時	乾球温度[C]	相対湿度[%]	計測日時	グローブ温度[C]	計測日時	微風速[cm/s]	計測日時	照度[lx]	SET	PMV	PPD[%]	DL
3/4 20:01:30	ap07	41B5F4EA	3/4 20:01:53	21.5	41.5	3/4 20:01:53	21.68	3/4 20:01:53	25.7	3/4 20:01:53	0.55	27.1	0.46	9.3	41B5F4EA.csv
3/4 20:01:32	ap06	41B5F4FC	3/4 20:01:55	21.8	40.7	3/4 20:01:55	23.37	3/4 20:01:55	19.7	3/4 20:01:55	0.51	27.8	0.64	13.5	41B5F4FC.csv
3/4 20:01:34	ap04	41B5F525	3/4 20:01:52	21.7	37.0	3/4 20:01:52	22.24	3/4 20:01:52	17.5	3/4 20:01:52	0.46	27.3	0.56	11.6	41B5F525.csv
3/4 20:01:33	ap03	41B5F47D	3/4 20:01:52	21.6	39.4	3/4 20:01:52	23.30	3/4 20:01:52	8.2	3/4 20:01:52	0.54	27.6	0.75	16.9	41B5F47D.csv
3/4 20:01:35	ap02	41B5E814	3/4 20:01:52	21.7	36.2	3/4 20:01:52	21.54	3/4 20:01:52	9.3	3/4 20:01:52	0.48	27.0	0.61	12.9	41B5E814.csv

Fig. 5.1 Web page showing the current measurement status

You can set the names of the sensor units to be displayed by entering a list of XBee lower address/name pairs in *mlnames.txt* at the same level as *MLServer*, as shown below.

```
42114F7E:SUName-01
42114F57:SUName-02
420BCCD1:SUName-03
420D3FC1:SUName-04
…
```

To calculate thermal comfort indices such as PMV and PPD, you need to set the clothing index and metabolic rate. These values are configured in the *setting.ini* file located at the same level as *MLServer*.

Fig. 5.1 displays only the latest measurements. If you want to refer to past values, you can download the CSV files for each sensor unit by following the links in the rightmost column of the table.

Fig. 5.2 shows the method for remote monitoring. By connecting a portable WiFi router to the control unit, you can monitor the status of the sensor units over the internet.

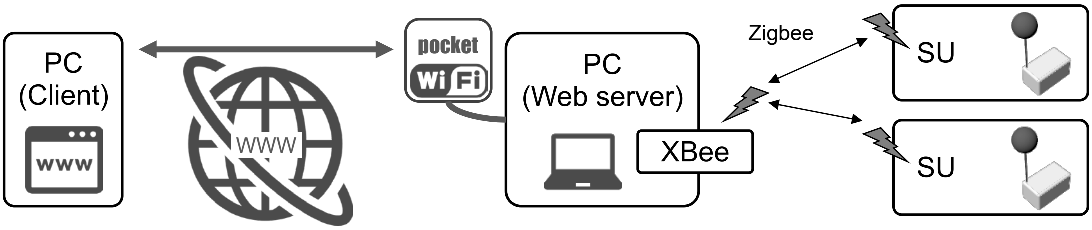


Fig. 5.2 The method for remote monitoring

2) Remote monitoring system with Windows

SimpleWebServer, located in the same directory as *MLServer*, is software for building a simple web server. When you launch it, the contents of the *data* directory will be served as the web root. However, you need to reserve the URL in advance using *netsh*. For more details, please refer to:
https://docs.microsoft.com/ja-jp/dotnet/framework/wcf/feature-details/configuring-http-and-https.

If the web server is running correctly, entering “127.0.0.1” in your browser will display a Basic Authentication screen as shown in Fig. 5.3. Enter *user* as the username and *pass* as the password to display the page shown in Fig. 5.1. You can change the username and password by editing the contents of *setting.ini* located in the same directory as *SimpleWebServer*.

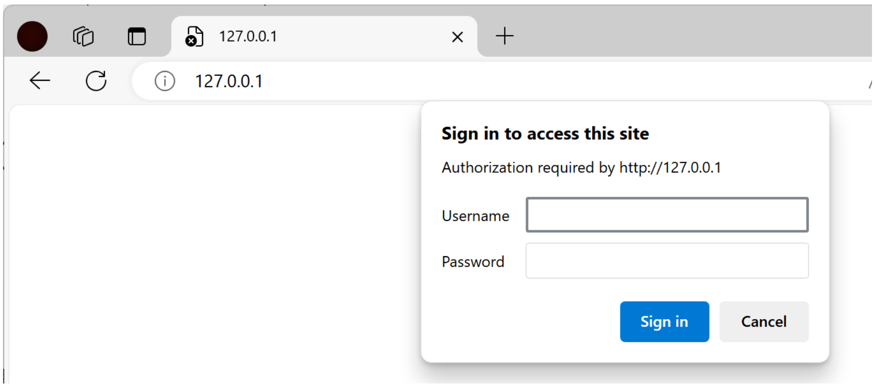


Fig. 5.3 Basic authentication of simple web server

To connect to the control unit's web server, you need the IP address of the control unit. If the control unit has a static IP address, there is no problem, but typically the IP address of a portable WiFi router may change. This means there is a risk of the IP address changing during on-site measurements, which would prevent connection.

To link a changing IP address with a fixed hostname, various companies offer Dynamic DNS (DDNS) services. *DDNSUpdater*, located in the same directory as *MLServer*, is software for using such DDNS services. The supported DDNS services are:

- dynamic DO (ddo.jp)
- NoIP (www.noip.com)

- DDNSNow (ddns.kuku.lu)

Register with one of these services, configure the registration details in *ddns.ini*, and launch *DDNSUpdater*. This will periodically send the current IP address to the DDNS, allowing you to connect to the control unit using the hostname.

For example, if you register the domain *mydomain.ddo.jp* with *dynamic DO* and set the password to *mypass*, you would configure *ddns.ini* as follows. The "#" at the beginning of each line indicates a comment.

With this setup, entering *http://mydomain.ddo.jp* in a browser will allow you to connect to the control unit.

```
## ddo.jp ##
service=dynamicDO;
pwd=mypass;                //password
host=mydomain.ddo.jp;      //domain, host name

## www.noip.com ##
#service=NoIP;
#usr=userid;               //ID
#pwd=password;             //password
#host=xxx.ddns.net;        //domain, host name

## ddns.kuku.lu ##
#service=DDNSNow;
#usr=userid;               //ID
#pwd=password;             //password

update=300;                //time interval to update [sec]
gip=https://api.ipify.org; //address of IP display service
```

Fig. 5.4 Example of ddns.ini configuration

3) Remote monitoring system with Raspberry Pi

Using a Raspberry Pi as the control unit allows you to utilize a robust web server for Linux, and it can continue running on battery power for extended periods in the event of a power outage at the measurement site. Below are the steps for configuring this setup.

The initialization of the Raspberry Pi will not be covered here. In the following explanation, we assume the username is *pi* and there is an *MLServer* directory on the desktop.

- Power Connection to the Raspberry Pi

When conducting long-term measurements on-site, there is a risk of power outages, so you cannot always rely on a continuous power supply from an outlet. To make the system more robust, place a battery between the outlet and the Raspberry Pi so that in case of a power outage, the battery will provide power (Fig. 5.5).

The function that allows simultaneous charging and discharging is called *pass-through charging*. Many batteries do not have this function, so please be careful when purchasing a battery.

The power consumption of the Raspberry Pi is about 600 mA at 5V, which is approximately 3W. Therefore, with a capacity of about 80 Wh, it can run for 24 hours on battery power alone. A 20,000 mAh battery with 3.7V (74 Wh) and pass-through charging can be obtained for less than $100.

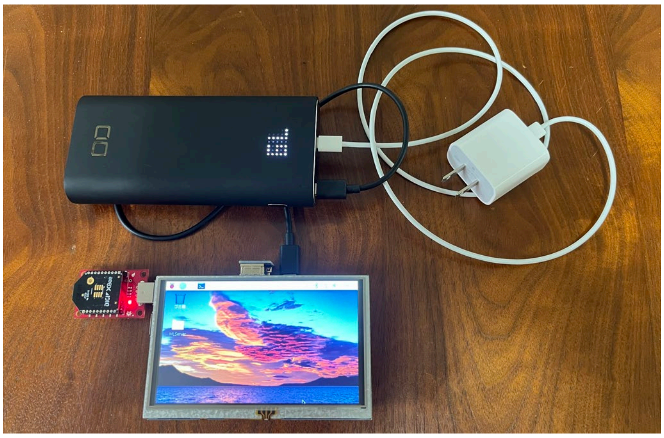


Fig. 5.5 Power connection via battery

- Installing the .NET Runtime

The Raspberry Pi uses a Linux-based OS called Raspberry Pi OS. Therefore, to run *MLServer*, you need to install the .NET runtime for Raspberry Pi OS. You can install the .NET runtime by entering the following commands in the console.

```
$ curl -sSL https://dot.net/v1/dotnet-install.sh | bash /dev/stdin --channel Current
```

Enter the following commands to add the path to dotnet.

```
$ echo 'export DOTNET_ROOT=$HOME/.dotnet' >> ~/.bashrc
$ echo 'export PATH=$PATH:$HOME/.dotnet' >> ~/.bashrc
$ source ~/.bashrc
```

If the installation is successful, the following command will display the .NET version.

```
$ dotnet –version
```

- Settings for reboot

Even with a battery, there is a risk that the Raspberry Pi may reboot unexpectedly. Therefore, we need to configure it to automatically resume data measurement upon reboot.

Create the following shell script to automatically start *MLServer* and *DDNSUpdater* upon reboot.

/home/*pi*/Desktop/MLServer/MLServer.sh

```
#!/bin/bash
dotnet /home/pi/Desktop/MLServer/MLServer.dll
```

/home/*pi*/Desktop/MLServer/DDNSUpdater.sh

```
#!/bin/bash
dotnet    /home/pi/Desktop/MLServer/DDNSUpdater.dll
```

If you create the above shell script on Windows, the line endings will be "¥r¥n (CRLF)," which may not be recognized correctly on the Linux side. Be sure to use "¥n (LF)" for line endings. Grant execute permissions to the created shell script with the following command.

```
$ chmod    755    /home/pi/Desktop/MLServer/MLServer.sh
$ chmod    755    /home/pi/Desktop/MLServer/DDNSUpdater.sh
```

Create the directory and file (*autostart*) to register the shell script for automatic execution at startup.

```
$ mkdir    -p    ~/.config/lxsession/LXDE-pi
$ cp    /etc/xdg/lxsession/LXDE-pi/autostart    ~/.config/lxsession/LXDE-pi/
```

Add the shell scripts you made to the last line of *autostart*.

/home/*pi*/.config/lxsession/LXDE-pi/autostart

```
@lxpanel    –profile    LXDE-pi
@pcmanfm    --desktop    --profile    LXDE-pi
@xscreensaver    -no-splash

@lxterminal    -e    "/home/pi/Desktop/MLServer/MLServer.sh"                    //Added.
@lxterminal    -e    "/home/pi/Desktop/MLServer/DDNSUpdater.sh"                 // Added.
```

- Installing Apache web server

Apache is one of the most widely used web servers in the world, and it can be easily installed if you are using a Raspberry Pi as the control unit. Install it with the following command.

```
$ sudo    apt    install    apache2    -y
```

To publish the *data* directory in the *MLServer* directory placed on the desktop, add the following settings.

/etc/apache2/sites-available/000-default.conf

```
DocumentRoot    /home/pi/Desktop/MLServer/data
<Directory /home/pi/Desktop/MLServer/data/>
    Options Indexes FollowSymLinks
```

```
    AllowOverride None
    Require all granted
</Directory>
```

Restart Apache with the following command.

```
$ sudo  service  apache2  restart
```

- Set up periodic reboots

Periodically reboot the Raspberry Pi to ensure *MLServer* restarts if it crashes unintentionally. Open crontab with the following command.

```
$ sudo crontab -e
```

Add the following to the last line to restart the system at midnight every night.

```
0  0  *  *  *  /sbin/reboot
```

4) Developing remote monitoring web pages

As shown in Fig. 5.6, *MLServer* outputs the latest measured values to a JSON file (*latest.json*). The web page displayed for remote monitoring is created by loading and processing this JSON file using JavaScript.

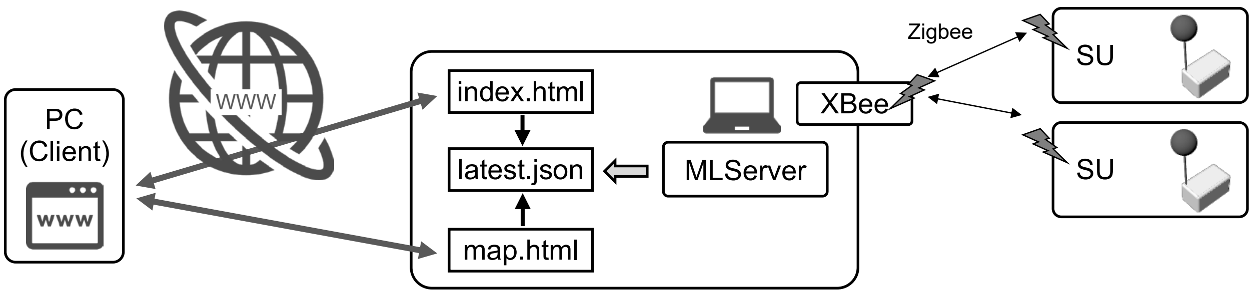


Fig. 5.6 How remote monitoring Web pages are created

Therefore, users can extend the functionality of the remote monitoring web page using HTML and JavaScript. By default, in addition to *index.html* shown in Fig. 5.1, there is also *map.html* for displaying measurement results on a heat map.

As shown in Fig. 5.7, *map.html* is a tool for displaying a thermal environment heat map by combining measurement values with floor plans. However, some preprocessing is required to enable the thermal environment heat map. You need to prepare the background floor plan, determine which sensor units correspond to which areas of the plan, and specify the positions where the measurement values will be displayed.

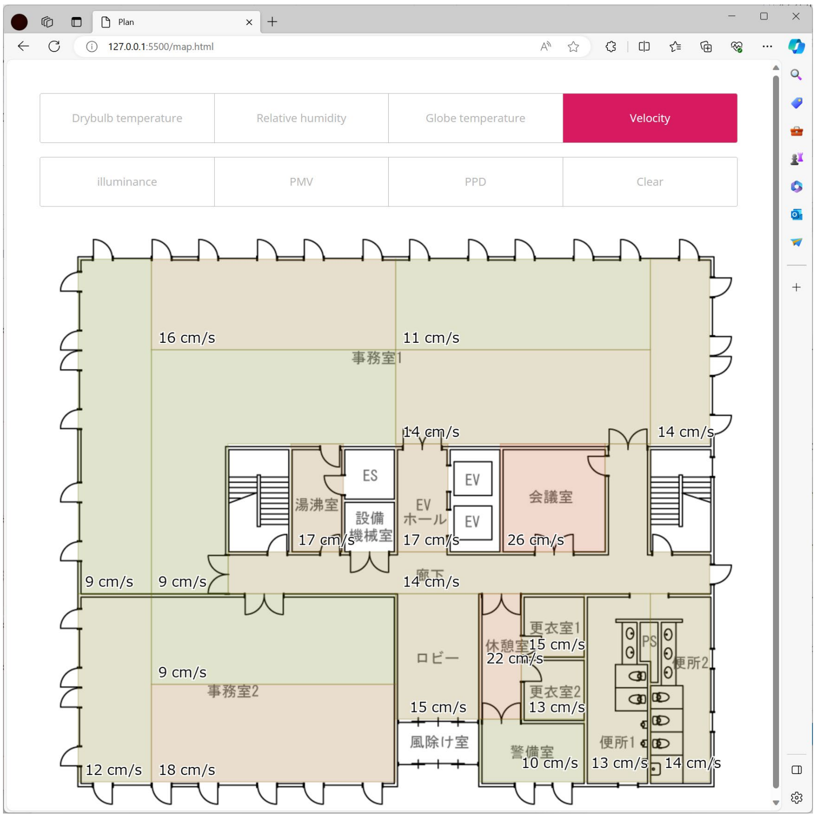


Fig. 5.7 Example of thermal environment heat map

- Prepare background floor plan

The background floor plan should be a PNG image with a width of 1000 px. The height will be adjusted automatically, so it can be any value. Save this image in the *data* directory with the name *background.png* (Fig. 5.8).

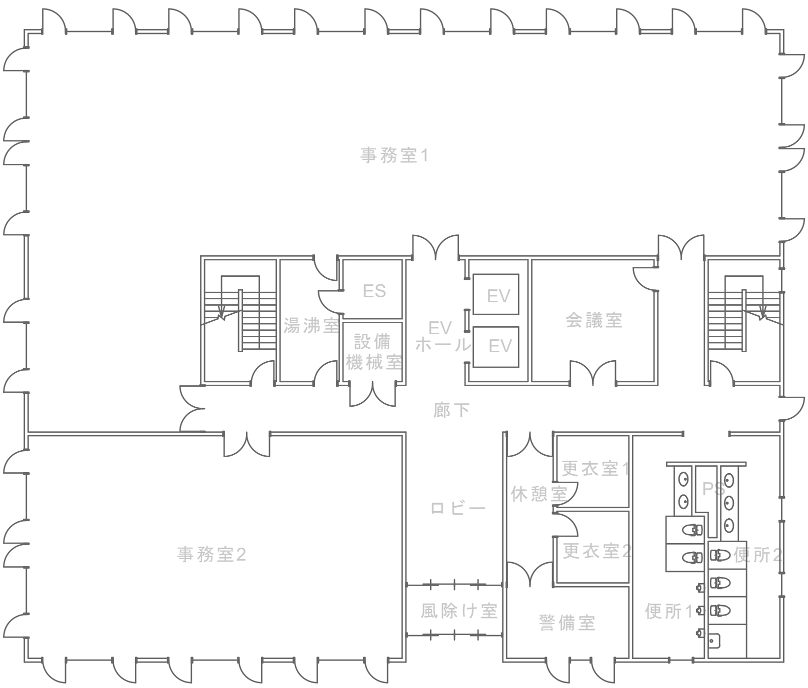


Fig. 5.8 Example of background floor plan

- Setting area

To associate the measurement values, set areas on the background floor plan. Areas can be defined as rectangles or polygons, so you need to determine the coordinates of the vertices. The coordinate system uses the top-left corner of the image as (0,0), with positive values extending to the right and down. Fig. 5.8 is divided into 20 areas, and Fig. 5.9 shows examples of the coordinates determined for each area.

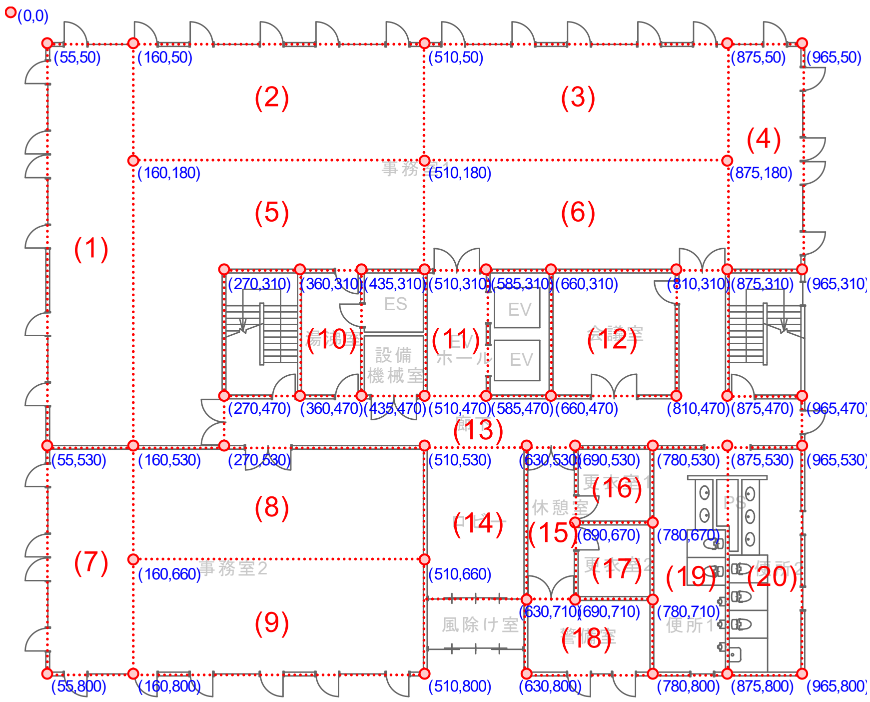


Fig. 5.9 Vertices for drawing areas

- Editing java script

To draw areas as rectangles or polygons, edit the *draw.js* file in the *data* directory. Add the XBee addresses of the sensor units used in the measurements to the switch statement in the *drawRegion* method, and specify where to draw each one.

To draw a rectangle, use the x and y coordinates of two diagonal points and write $rect(x_0, y_0, x_1, y_1)$. For example, in lines 4 and 5, a rectangle is assigned to the sensor unit with the address "42114F57" from (55, 50) to (160, 530). This corresponds to area (1) in Fig. 5.9.

To draw a polygon with n vertices, add "$vertex(x_n, y_n)$" statements between the *beginShape*() and *endShape*(*CLOSE*) methods. Lines 16 to 24 show an example of assigning area (5) in Fig. 5.9.

This feature is implemented using the *p5.js* library, so the drawing methods are not strictly limited to rectangles and polygons. Users can extend the functionality themselves.

```
function drawRegion(mloggerID){
  rectMode(CORNERS);
  switch(mloggerID){
    case "42114F57":
      rect(55, 50, 160, 530);
      break;
    case "420BCD82":
      rect(160, 50, 510, 180);
      break;
    case "42114EB8":
      rect(510, 50, 875, 180);
      break;
    case "42114EF0":
      rect(875, 50, 960, 315);
      break;
    case "420BCCD1":
      beginShape();
      vertex(160,180);
      vertex(510,180);
      vertex(510,315);
      vertex(270,315);
      vertex(270,530);
      vertex(160,530);
      endShape(CLOSE);
      break;
    case "42114F8F":
      rect(510, 180, 875, 315);
      break;
    case "420D3FC1":
      rect(55, 530, 160, 800);
      break;
    case "420BCD79":
      rect(160, 530, 510, 660);
      break;
    case "420BCCDA":
      rect(160, 660, 510, 800);
      break;
    case "420BCD0B":
      rect(360, 315, 435, 470);
      break;
    case "42114F00":
      rect(510, 315, 585, 470);
      break;
    case "42114E92":
      rect(660, 315, 810, 470);
      break;
    case "420BCD7E":
      beginShape();
      vertex(270,470);
      vertex(810,470);
      vertex(810,315);
      vertex(875,315);
      vertex(875,470);
      vertex(960,470);
      vertex(960,530);
      vertex(270,530);
      endShape(CLOSE);
      break;
    case "42114E95":
      rect(510, 530, 630, 710);
      break;
    case "420BCDC3":
      rect(630, 530, 690, 710);
      break;
    case "42114E40":
      rect(690, 530, 780, 620);
      break;
    case "42114F7E":
      rect(690, 620, 780, 710);
      break;
    case "42114EFA":
      rect(630, 710, 780, 800);
      break;
    case "420BCDD7":
      rect(780, 530, 875, 800);
      break;
    case "42114F78":
      rect(875, 530, 960, 800);
      break;
    default:
      break;
  }
}
```

Fig. 5.10 Example of *draw.js*

6.1 LED signals

The sensor units are equipped with blue, green, and red LEDs. The blue LED indicates the status of wireless data transmission and reception, blinking during communication. This status display can be enabled or disabled via software. The green and red LEDs indicate the hardware status through the number of blinks and the interval between blinks, as shown in Fig. 6.1. Table 6.1 shows the various states.

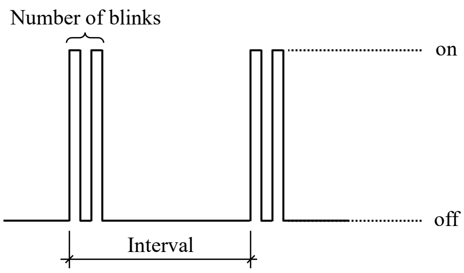


Fig. 6.1 Status indication by LED

Table 6.1 Status Based on Number of Blinks and Blink Intervals

Color	Number	Intervals	Status
Green	1	1	Waiting for commands from the control unit. Flash memory not recognized.
Green	2	1	Waiting for commands from the control unit. Flash memory has recognized.
Green	1	5	Logging in progress.
Both	1	1	Auto-calibration in progress.
Red	1	1	Logging is in progress but flash memory is not recognized.
Red	1	2	Low battery.
Red	2	2	XBee initialization error.
Red	3	-	Successfully deactivated the permanent mode by long pressing the reset switch.

6.2 Protocols

The control unit and sensor units communicate by sending and receiving strings via Bluetooth or Zigbee. The sensor units are passive and only respond to commands from the control unit. When a command from the control unit is successfully received by a sensor unit, the sensor unit will send a response.

Commands from the control unit consist of a three-letter reserved word followed by parameters. Table 6.2 shows a list of reserved words.

Table 6.2 List of reserved words

Reserved word		Function
VER	VERsion	Get the version of the sensor unit.
STL	STart Logging	Start measuring and logging.
CMS	Change Measurement Settings	Change measurement settings.
LMS	Load Measurement Settings	Load measurement settings.
ENL	ENd Logging	End measuring and logging.
SCF	Set Correction Factor	Set correction factors.
LCF	Load Correction Factor	Load correction factors.
CLN	Change Logger Name	Change name of sensor unit.
LLN	Load Logger Name	Load name of sensor unit.
CBV	Calibrate Velocity Voltage	Automatically calibrates voltage at no wind.
CBT	Calibrate Temperature	Automatically calibrates dry bulb and globe temperature.
SCV	Start Calibrate Velocity Voltage	Start manual calibration of voltage at no wind.
ECV	End Calibrate Velocity Voltage	End manual calibration of voltage at no wind.
UCT	Update Current Time	Update current date and time.

Below are the details of each command. Commands sent by the control unit, including the reserved word and parameters, must be within 72 characters. To indicate the end of a command, append "¥r" (Carriage Return) at the end.

- VER (VERsion) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
V	E	R	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “Vers:*int.int.int*”

int is an integer value representing the version.

- STL (STart Logging) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	T	L	Current date and time										t/f/e	t/f	t/f	¥r							
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Parameters:

3~12: The current time expressed in UNIX time of type *time_t* (*long*).

13: Whether to send the measurement values via Zigbee. Send with *t*, do not send with *f*. Set *e* to enter permanent mode.

14: Whether to send the measurement values via Bluetooth. Send with *t*, do not send with *f*.

15: Whether to write the measurement values to Flash memory. Write with *t*, do not write with *f*.

Response from sensor unit: “STL”

Subsequently, a periodic response in the format "*DTT:yyyy,MM/dd,HH:mm:ss, float, float, float, float, float, float, float, float*" will follow.

The first part "*yyyy,MM/dd,HH:mm:ss*" represents the date and time. The eight floats in the latter part represent the following measurements: dry-bulb temperature [°C], relative humidity [%], globe temperature [°C], low air velocity [m/s], illuminance [lx], globe temperature voltage [V] (always 0), low air velocity voltage [V], and general-purpose ADC voltage [V].

- CMS (Change Measurement Settings) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	M	S	t/f	Temperature and humidity					t/f	Globe temperature					t/f	Velocity					t/f	Illuminance	
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Illuminance			Measurement start date and time										t/f	ADC					f	0	0	0	0
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
0	f	0	0	0	0	0	f	¥r															

Parameters:

3: Whether to measure the temperature and humidity. Measure with *t*, do not measure with *f*.

4~8: Time interval [sec] of temperature and humidity measurement. An integer not exceeding 5 digits.

9: Whether to measure the globe temperature. Measure with *t*, do not measure with *f*.

10~14: Time interval [sec] of globe temperature measurement. An integer not exceeding 5 digits.

15: Whether to measure the velocity. Measure with *t*, do not measure with *f*.

16~20: Time interval [sec] of velocity measurement. An integer not exceeding 5 digits.

21: Whether to measure the illuminance. Measure with *t*, do not measure with *f*.

22~26: Time interval [sec] of illuminance measurement. An integer not exceeding 5 digits.

27~36: The measuring start time expressed in UNIX time of type *time_t* (*long*).

37: Whether to measure the voltage of general-purpose ADC. Measure with *t*, do not measure with *f*.

38~42: Time interval [sec] of voltage of general-purpose ADC measurement. An integer not exceeding 5 digits.

43~55: Parameters used in past versions. Fixed with "f00000f00000f".

Response from sensor unit: “CMS:*bool*,*int*,*bool*,*int*,*bool*,*int*,*bool*,*int*,*long*,*bool*,*int*,*bool*,*int*,*bool*,*int*,*bool*”

Bool represents the true/false status of the measurement with 1/0, int is an integer value representing the measurement interval [sec], and long is the UNIX time indicating the measurement start date and time. From the beginning, the settings are for temperature and humidity, globe temperature, low air velocity, illuminance, measurement start date and time, and general-purpose ADC voltage. The trailing “*bool,int,bool,int,bool*” was used in past versions and no longer has any meaning.

- LMS (Load Measurement Settings) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L	M	S	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “LMS:*bool*,*int*,*bool*,*int*,*bool*,*int*,*bool*,*int*,*long*,*bool*,*int*,*bool*,*int*,*bool*,*int*,*bool*”

The meaning of LMS parameters is the same as for CMS.

- ENL (ENd Logging) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	N	L	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “ENL”

・SCF (Set Correction Factor) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	C	F	*A’* (DB temp)				*B’* (DB temp)				*A’* (Humidity)				*B’* (Humidity)				*A’* (Globe temp)				*B’*
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
(Globe temp)			*A’* (Illuminance)				*B’* (Illuminance)				*A’* (Velocity)				*B’* (Velocity)				No wind voltage				¥r
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

†A' and B' are the coefficients used in the following equation: Corrected value = A × Measured value + B.

Parameters:

3~6: Correction coefficient for dry-bulb temperature. $A = 0.001A'$. The range of A' is set to 800~1200.

7~10: Correction coefficient for dry-bulb temperature. $B = 0.01B'$. The range of B' is set to -300~300.

11~14: Correction coefficient for relative humidity. $A = 0.001A'$. The range of A' is set to 800~1200.

15~18: Correction coefficient for relative humidity. *B* = 0.01*B'*. The range of *B*' is set to -999~999.

19~22: Correction coefficient for globe temperature. *A* = 0.001*A'*. The range of *A*' is set to 800~1200.

23~26: Correction coefficient for globe temperature. *B* = 0.01*B'*. The range of *B*' is set to -300~300.

27~30: Correction coefficient for illuminance. *A* = 0.001*A'*. The range of *A*' is set to 800~1200.

31~34: Correction coefficient for illuminance. *B* = *B'*. The range of *B*' is set to -999~999.

35~38: Correction coefficient for low air velocity. *A* = 0.001*A'*. The range of *A*' is set to 800~1200.

39~42: Correction coefficient for low air velocity. *B* = 0.001*B'*. The range of *B*' is set to -500~500.

43~46: 1000 times the voltage of the wind speed probe when there is no wind. The range is set to 1400~1500.

Response from sensor unit: "SCF: *float,float,float,float,float,float,float,float,float,float,float*"

Correction coefficients *A* and *B* for dry bulb temperature, relative humidity, globe temperature, illuminance, and low air velocity are listed in order. The float at the end is the voltage at no wind.

- LCF (Load Correction Factor) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L	C	F	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: "LCF: *float,float,float,float,float,float,float,float,float,float,float*"

The meaning of LCF parameters is the same as for SCF.

- CLN (Change Logger Name) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	L	N	Name of the sensor unit																				¥r
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: "CLN: Name"

- LLN (Load Logger Name) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
L	L	N	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: "LLN: Name"

- CBV (Calibrate Velocity Sensor) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	B	V	Calibrating time [sec]					¥r															
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “CBV: Remaining seconds”

After waiting 60 seconds to stabilize the low air velocity sensor’s readings, the voltage values from the sensor are measured for the duration specified by the parameters. The average of these measurements is used to initialize the voltage in the absence of wind. The measurements are considered valid only if the voltage is within the range of 1.4 V to 1.55 V. The remaining seconds until the calibration is complete are periodically transmitted from the sensor unit.

- CBT (Calibrate Temperature Sensor) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	B	T	Calibrating time [sec]					¥r															
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “CBT: Remaining seconds”

After waiting for 60 seconds to stabilize the thermometer's readings, the dry-bulb temperature and globe temperature are measured for the duration specified by the parameters. The regression coefficients are determined using the dry-bulb temperature as the dependent variable and the globe temperature as the independent variable. These regression coefficients are then used to initialize the correction factor for the globe temperature sensor. The results are considered valid only if the slope of the regression line is between 0.7 and 1.3 and the intercept is between -2 and 2. The remaining seconds until the calibration is complete are periodically transmitted from the sensor unit.

- SCV (Start Calibrate Velocity Voltage) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	C	V	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “SCV: Voltage [V]”

Every second, the voltage from the low air velocity sensor is measured and transmitted from the sensor unit.

- ECV (End Calibrate Velocity Voltage) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	C	V	¥r																				
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71

Response from sensor unit: “ECV”

- UCT (Update Current Time) command

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
U	C	T	Current date and time										¥r	-	-	-	-	-	-	-	-	-	-
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Parameters:

3~12: The current time expressed in UNIX time of type *time_t* (*long*).

Response from sensor unit: “UCT”

References

1) Digi XBee 3 RF Module, Hardware Reference Manual, Revision V, 2022.3
2) HT77xxBA, 5V/200mA PFM, Asynchronous Step-up Converter, Datasheet, Holtek, Rev. 1.20, 2021.8.17
3) MCP9700/9700A/MCP9701/9701A, Low-Power Linear Active Thermistor™ ICs, Microchip, Datasheet, DS20001942F
4) ATMega328P, 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, Datasheet, Atmel, Rev. 7810D–AVR–01/15
5) AVR32DB28/32/48, AVR DB Family, Datasheet, Microchip, Revision DS40002301A
6) VCNL4030X01, Datasheet, Vishay, Revision 1.2, Document Number 84250, 2020.3.18
7) 富樫英介: 室内温熱環境の廉価な無線計測記録装置の開発, 日本建築学会技術報告集, Vol. 28, No.68, pp. 267-272, 2022.2, DOI: 10.3130/aijt.28.267
8) 濱原和明: 超お手軽無線モジュール XBee, QC 出版株式会社, 2012.3.1