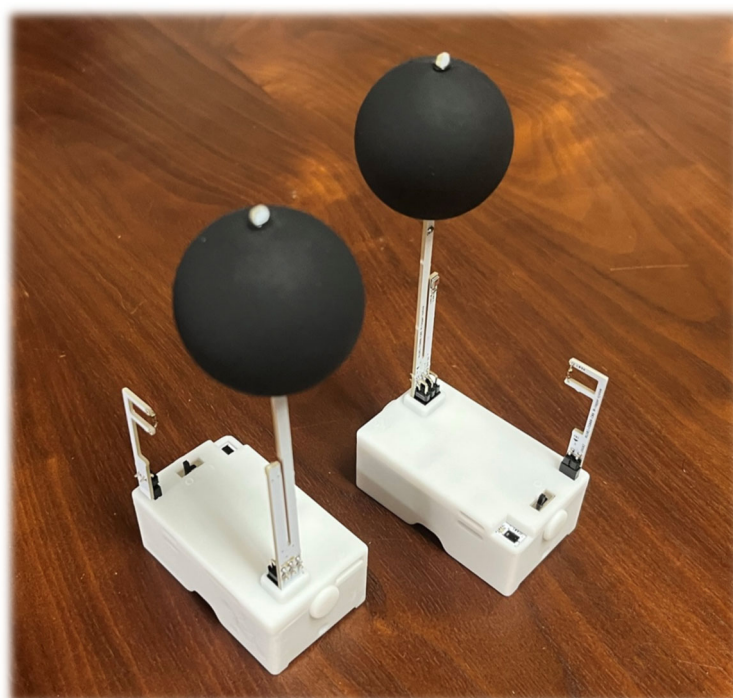


M-Logger システム解説書

2024年1月18日版



目次

第1章	はじめに	1
1.1	M-Logger システムとは	1
1.2	本資料の構成	1
第2章	M-Logger システムの使い方	2
2.1	温熱環境測定システムの構成	2
2.2	子機 (M-Logger) の構造	3
第3章	スマートフォンを使った計測	5
3.1	準備	5
3.2	ソフトウェアの使い方	6
1)	全体の構成	6
2)	計測方法の設定	6
3)	スマートフォンへの記録	7
4)	記録データの閲覧	8
5)	フラッシュメモリへの記録	8
6)	PC との接続	9
7)	その他の設定	10
8)	熱的快適性の計算機能	10
9)	湿り空気物性の計算機能	11
第4章	PC などを使った計測	12
4.1	準備	12
1)	ハードウェア	12
2)	ソフトウェア	15
4.2	ソフトウェアの使い方	16
第5章	遠隔監視システムの構築	20
1)	Web サーバーで公開するディレクトリ	20
2)	Windows による遠隔監視システム	21
3)	Raspberry Pi による遠隔監視システム	22
4)	遠隔監視画面の調整	26
第6章	通信仕様	29

第1章 はじめに

1.1 M-Logger システムとは

M-Logger は建物の室内温熱環境を把握するための無線計測記録装置です。「M」は「Mini」と「Makunouchi」が由来で、手のひらに乗る小型寸法でありながら、計測できる物理量は乾球温度、相対湿度、グローブ温度、微風速、照度と、盛り沢山です。以下に主な特徴を挙げます。

- ・仕様書や技術資料がすべて無償で公開されている（オープンソース ハードウェア）
- ・60×40×20 mm の小型な外形
- ・熱的快適性を評価するために必要な乾球温度、相対湿度、グローブ温度、微風速が計測できる
- ・熱的快適性指標（PMV, PPD, SET*）をリアルタイムで演算して表示できる
- ・残業状況やペリメータへの日射入射の状況を推測するための照度が計測できる
- ・コンセント接続せずに単3電池×2本で駆動できる
- ・数週間の連続測定ができる（微風速計測無の場合には1ヶ月超）
- ・1台あたり約5,000円で製造することができる
- ・無線でデータ収集ができる
- ・親機（PC）からGUIを使って無線で初期設定ができる
- ・Raspberry PiなどのLinux系OSを親機とすることができる
- ・親機をWebサーバー化して遠隔監視ができる
- ・計測時間間隔、計測開始日時、計測対象の物理量は任意に設定できる
- ・スマートフォン（iPhone, Android）を使ってデータを収集できる

1.2 本資料の構成

第2章では、M-Loggerの仕様と、これを使った計測システムの全体像について解説します。

第3章では、スマートフォンを使ってM-Loggerを操作する方法について解説します。

第4章では、WindowsなどのPCを使ってM-Loggerを操作する方法について解説します。

第5章では、WindowsやRaspberry Piなどを使って、遠隔で計測結果を監視するシステムを構築する方法について解説します。

第6章では、M-Loggerと通信するための仕様を解説します。

M-Loggerは開発に必要な情報が全て公開されたオープンソースハードウェアで、本資料の他にも多くの情報をWebサイト（<https://www.mlogger.jp>）で公開しています。必要に応じてWebも参照してください。

第2章 M-Logger システムの使い方

2.1 温熱環境測定システムの構成

M-Logger システムは、温湿度などの物理量を計測する「子機 (M-Logger)」と、子機の計測方法を管理したり、収集したデータを保存したりする「親機」とで構成されます。親機と子機は無線通信で接続されます (Fig. 2.1)。親機は、スマートフォンまたは一般の PC を使うことができます。

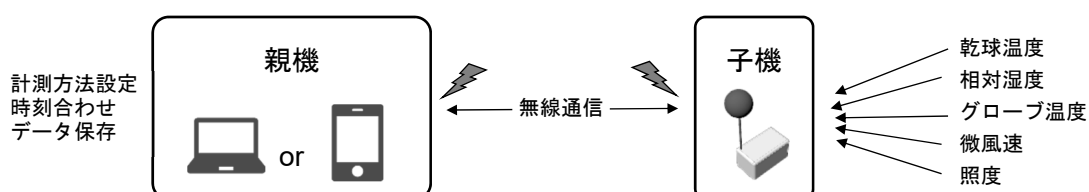


Fig. 2.1 M-Logger システムの構成

スマートフォンを親機にする場合には、親機と子機は Bluetooth Low Energy (BLE) で通信します。この場合には接続できる子機の台数は 1 台のみです (Fig. 2.2)。この操作は第 3 章で解説します。

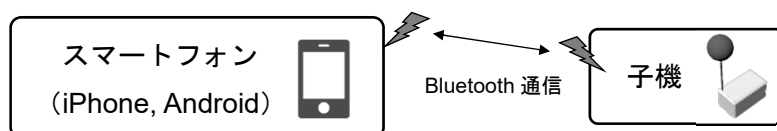


Fig. 2.2 スマートフォンを親機にする場合

PC などを親機にする場合には、親機と子機は Zigbee で通信します。この場合には複数の子機と接続してまとめてデータを収集することができます (Fig. 2.3)。この操作は第 4 章で解説します。

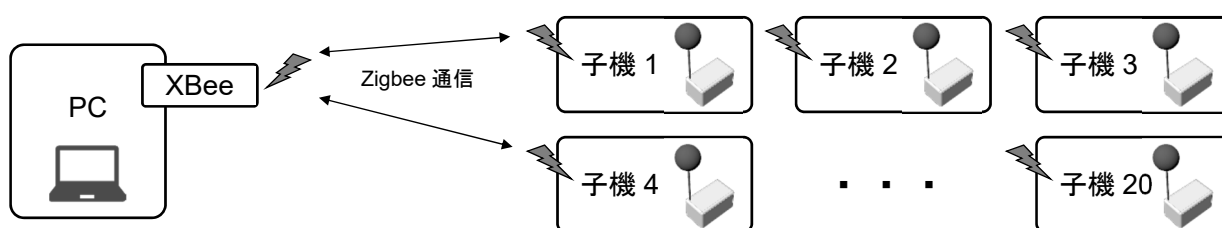


Fig. 2.3 PC などを親機にする場合

PC などを親機にする場合の設定は、スマートフォンを親機にする場合に比べてやや複雑です。従って、室内の温熱環境を簡易に把握したい場合にはスマートフォンを親機とし、本格的な多点計測をした場合には PC などを親機にすると良いでしょう。

2.2 子機 (M-Logger) の構造

子機は、乾球温度、相対湿度、グローブ温度、微風速、照度（または近接計）の計測機能を持ちます。

子機の外観を Fig. 2.4 に示します。グローブ温度計と風速計は基板と着脱でき、計測する際に接続端子につなぎます。なお、風速計の端子には向きがありません (Fig. 2.5)。

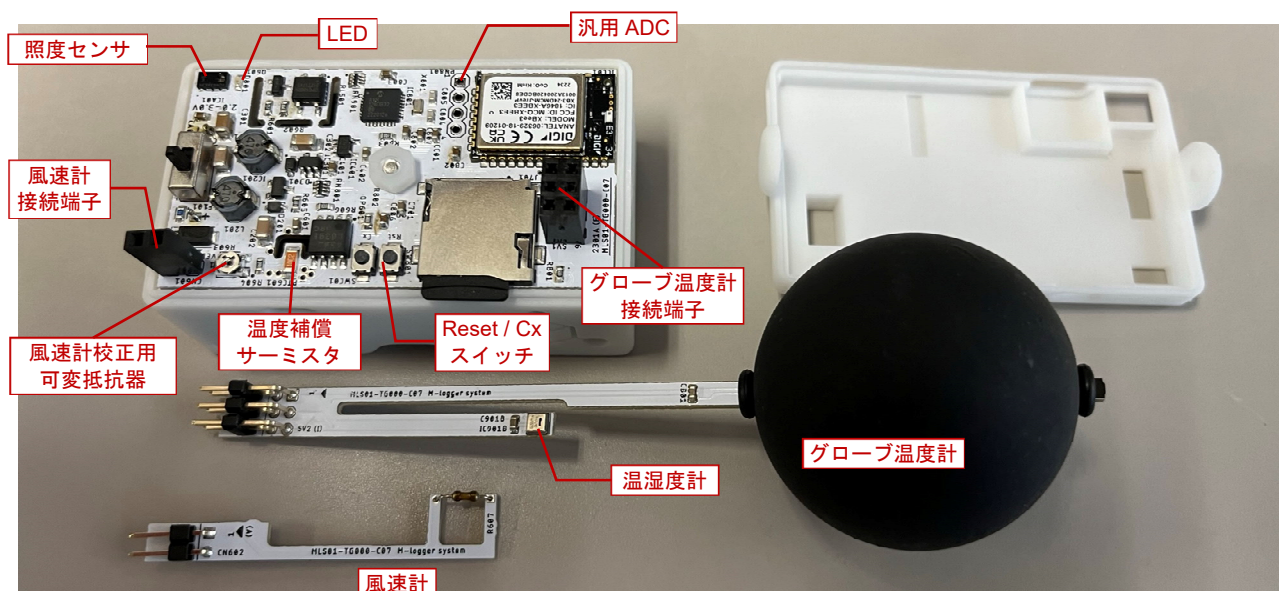


Fig. 2.4 子機 (M-Logger) の外観

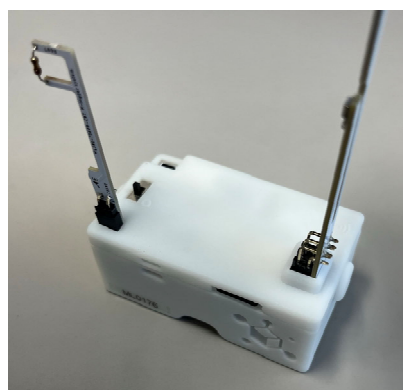


Fig. 2.5 風速計とグローブ温度計の接続

照度計は基板に直接に設置されているため、照度を計測する場合にはテープなどで塞がれないように注意してください。また、可変抵抗器は風速計を校正するために使います。従って、特別の理由が無い限り、回転させて抵抗値を変えないようにしてください。

風速計とグローブ温度計を接続した後に裏面の電池ボックスに単 3 電池を差し込み、スイッチを入れると、LED が 1 秒間隔で点滅を始めます。これは親機からの通信を受け入れる準備ができたことを表しています。点滅回数は 1 回ですが、フラッシュメモリカードが挿入されていて、カードが認識されている場合には 2 回点滅となります。親機から計測項目や計測時間間隔を設定し、正常に計測が始まると LED の点滅は 5 秒に 1 回に変わります。

微風速を計測するときには、風速計のプロープの温度が 40℃ 程度まであがるため、直接に手で触れぬ

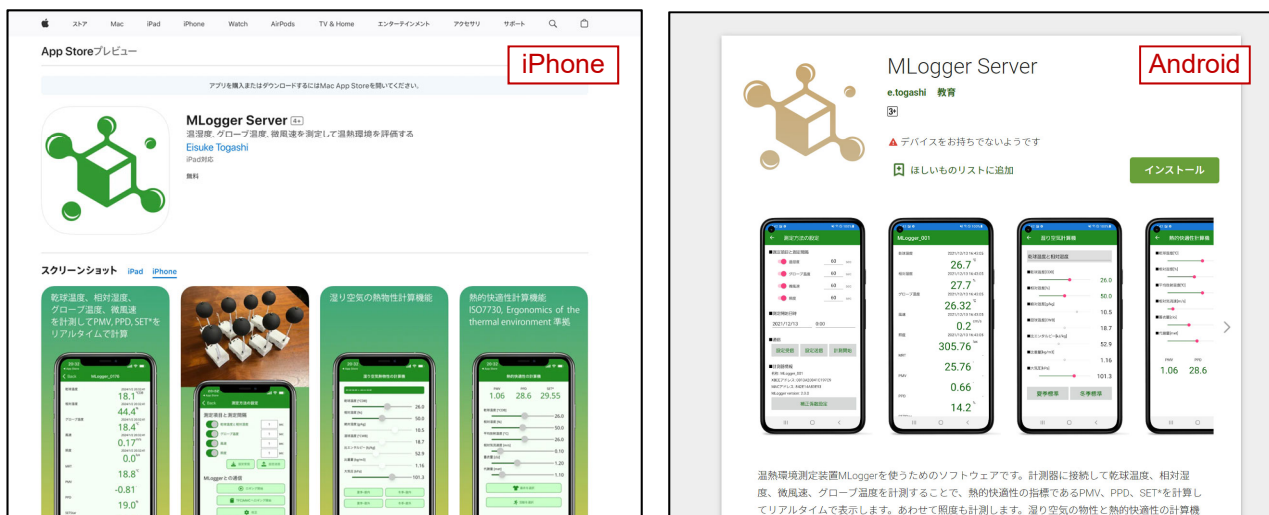
ように注意する必要があります。

子機が電池でどれだけ稼働できるかは、計測の設定内容に依存します。最も電池を消費するのはプローブを昇温させる必要がある風速計です。電池として完全に充電された Enloop Pro を使った実験では、計測間隔を 1sec、1 min、5 min、10 min とした場合、それぞれ 24 h、97 h、250 h、450 h の計測ができました。ただし、これらの時間は風速計周囲の温度に影響を受けて変わり、温度が低いほど、早く電池を消耗します。

第3章 スマートフォンを使った計測

3.1 準備

スマートフォンを親機にする場合には、特別なハードウェアを追加する必要はありません。iPhone の場合には Apple Store、Android の場合には Google Play に接続し、「mlogger server」で検索して必要なソフトウェアをダウンロード・インストールすればすぐに使えます (Fig. 3.1)。



Apple Store: <https://apps.apple.com/jp/app/mlogger-server/id1599907037>

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

Fig. 3.1 ソフトウェアのダウンロード

親機と子機は Bluetooth Low Energy (BLE) を使って通信するため、スマートフォンの設定で、ソフトウェアに BLE による通信許可を与える必要があります (Fig. 3.2)。特に Android の場合には、仕様により BLE を有効にするには「位置情報」の許可も与える必要がある点に注意が必要です。



Fig. 3.2 Bluetooth LE 通信権限の付与

3.2 ソフトウェアの使い方

1) 全体の構成

ソフトウェアには4つの機能があり、下部のタブで切り替えます (Fig. 3.3)。Fig. 3.3.(1)は子機への接続、Fig. 3.3.(2)は測定データの閲覧・操作、Fig. 3.3.(3)は熱的快適性の計算、Fig. 3.3.(4)は湿り空気の計算、です。この内、(3)と(4)は子機を必要としない補助機能です。

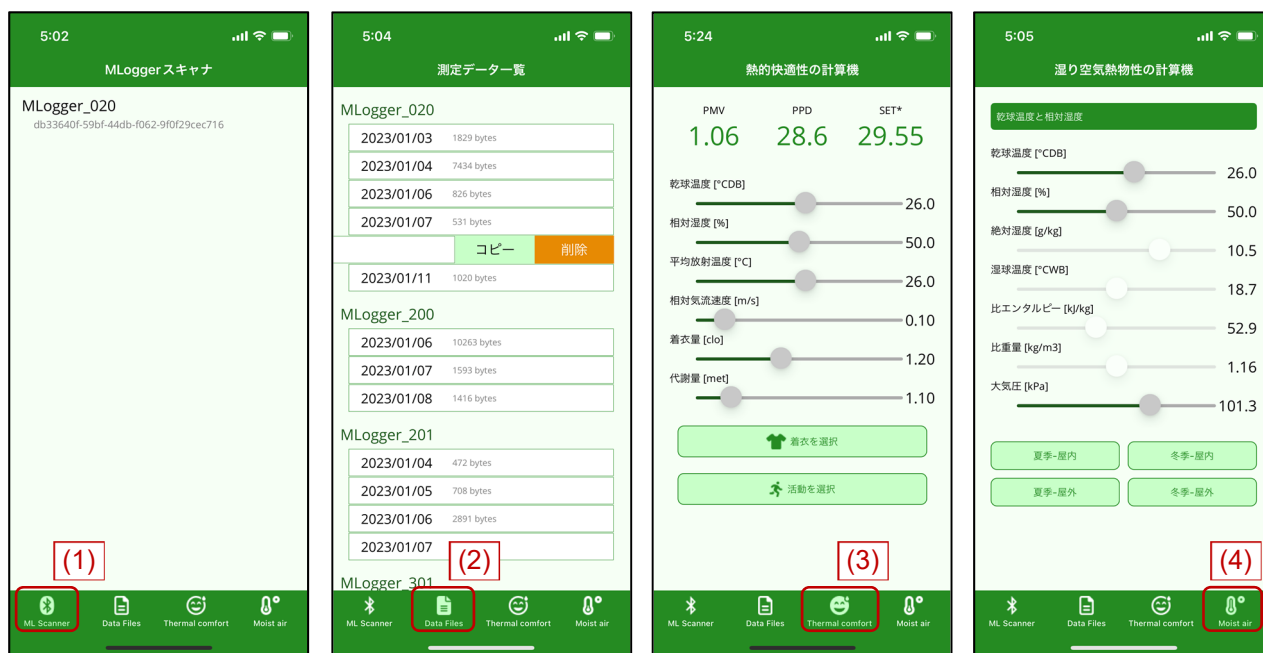


Fig. 3.3 4つの計算機能

2) 計測方法の設定

子機への接続タブで下方にプルダウンすれば新しい子機が探索され、起動している子機のリストが更新されます (Fig. 3.4.(1))。

リスト表示された子機の1つをタップすると、詳細設定画面に移ります。それぞれの子機の内部には計測に関わる設定が保存されていて、この画面でその設定内容を変更することができます。

上部には「測定項目と測定間隔」の欄があります (Fig. 3.4.(2))。ここでは、乾球温度、相対湿度、グローブ温度、風速、照度のそれぞれを計測するか否か、計測する場合の時間間隔、計測開始日時を設定できます。

これらの設定は子機の内部に記録されています (ただし「計測開始日時」は Version 3.2.12 以降)。「M-Logger から設定を受信」をタップすれば子機に保存されている設定を受信できます。逆に、設定を変えて「M-Logger へ設定を送信」をタップすれば新しい設定で子機の情報を書き込みます。子機の情報と異なる値に設定を変えた場合には、その項目は赤字で表されます (Fig. 3.4.(3))。

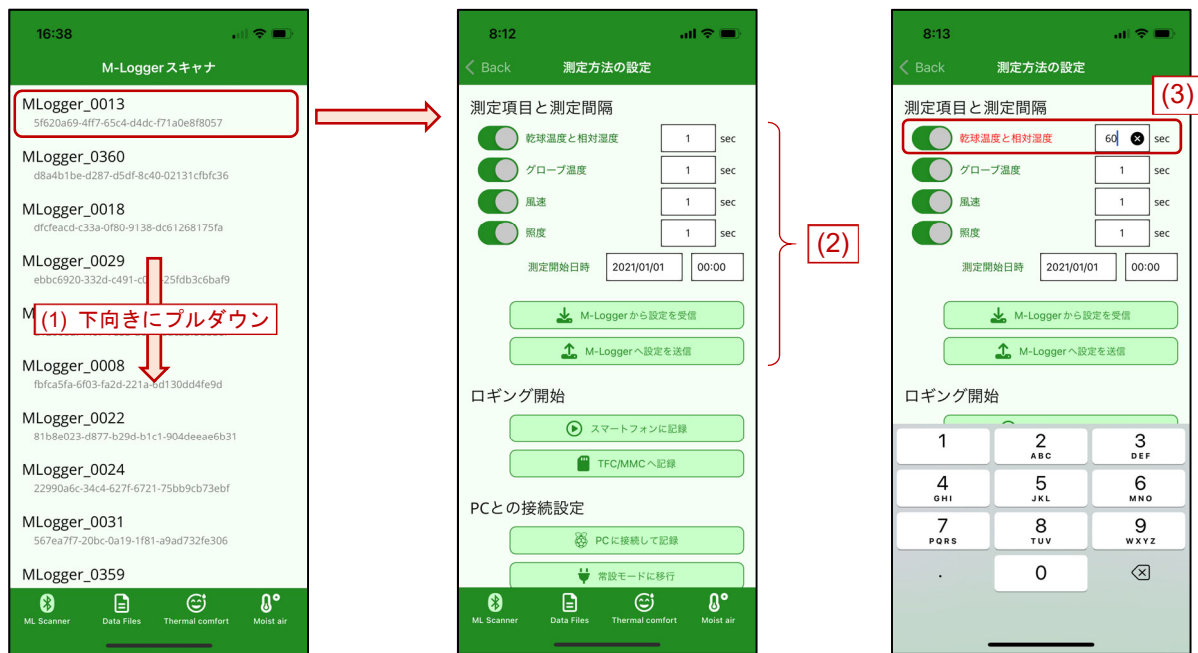


Fig. 3.4 子機への接続

3) スマートフォンへの記録

子機が計測した値は、無線でスマートフォンへ送信するか、計測器に挿入したフラッシュメモリに記録するか、2通りの方法で保存できます。

スマートフォンに送信する場合には、「スマートフォンに記録 (Fig. 3.5.(1))」をタップします。

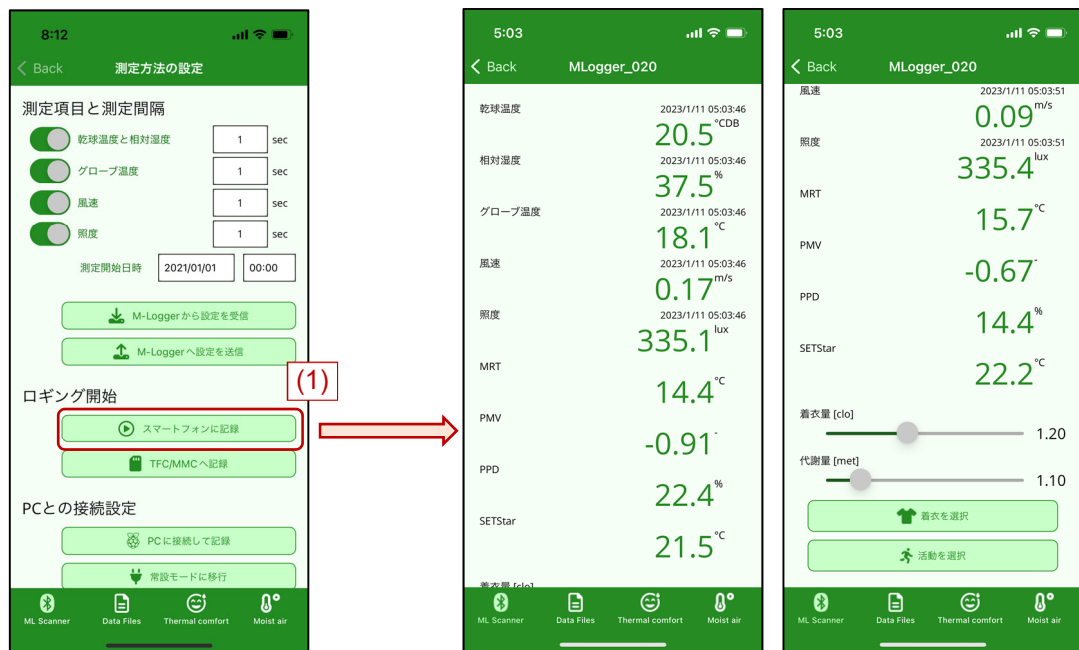


Fig. 3.5 スマートフォンに記録

画面が切り替わり、乾球温度、相対湿度、グローブ温度、風速、照度が表示されます。また、これらの

計測値にもとづいて演算された MRT、PMV、PPD、SET*も表示されます。

熱的快適性の計算のためには着衣量と代謝量の値を仮定する必要があります。これらを変更するには「着衣を選択」または「活動を選択」をタップしますが、その操作の詳細は後述します。

Back をタップして前の画面に戻れば自動で計測は終了されます。正常に終了すると子機の LED は再び点滅状態に戻ります。

4) 記録データの閲覧

スマートフォンで受信したデータは自動で記録され、「測定データの閲覧・操作」タブで表示させることができます。「測定データの閲覧・操作」タブには、子機の ID ごと、日付ごとにデータが一覧表示されます。項目を左にスワイプ (Fig. 3.6.(1)) すれば、データをコピーまたは削除できます。また、項目をタップ (Fig. 3.6.(2)) すればデータの中身をテーブル表示することができます。この画面でボタンをタップ (Fig. 3.6.(3)) すればデータのコピーまたは削除ができます。クリップボードにコピーされるデータは CSV 形式のテキストデータで、表計算ソフトウェアなどを使えば編集できます。

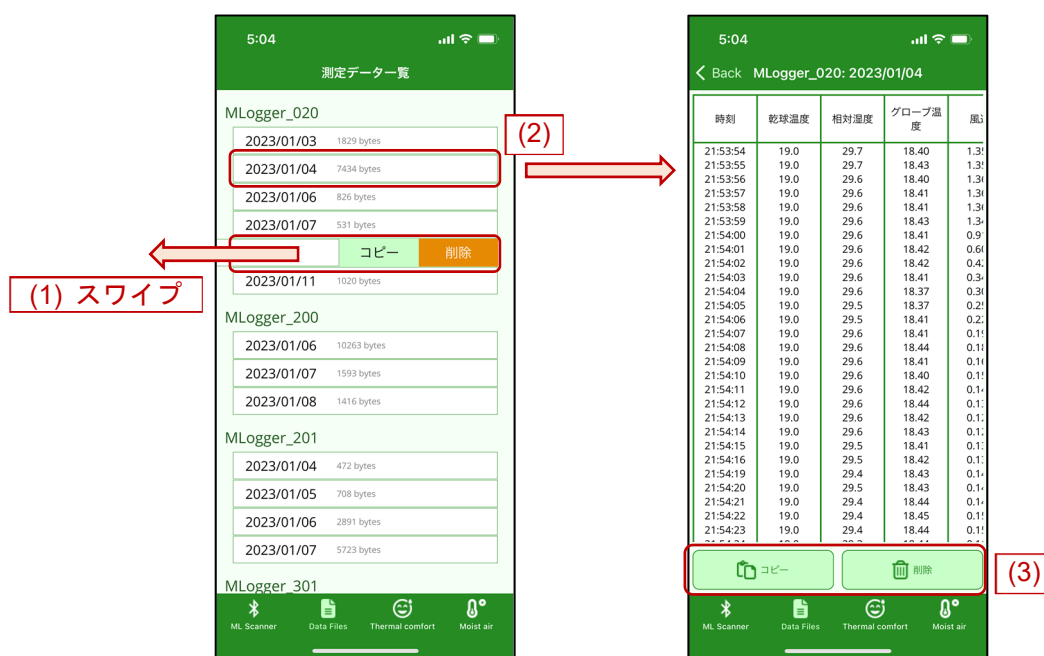


Fig. 3.6 記録された計算値の閲覧

5) フラッシュメモリへの記録

フラッシュメモリカードとしては、TF (Trans flash) カード、MM (Multi media) カードまたはこれらの互換カードに対応しています。これらのカードに記録するには、まず、子機にカードを認識させる必要があります。Fig. 3.7 に示す位置にカードを差し込み、カードが正常に認識されると LED の点滅が 1 回から 2 回に変わります。なお、本計測器は FAT32 形式にしか対応していません。一般に 32GB を超える SDXC 形式のカードは exFAT でフォーマットされているため、このようなカードを使う場合には一旦、Windows などでカードを FAT32 でフォーマットする必要があります。

カードが正常に認識されたら「TFC/MMC へ記録 (Fig. 3.7.(1))」をタップします。カードへの記録が始まるとスマートフォンとの接続は解除され、初期画面 (M-Logger 一覧) に戻ります。計測器は電池を節

約するためにできる限りスリープするようになりますので、再びスマートフォンと接続するためには、計測器を一度 Off にして再起動する必要があります。



Fig. 3.7 TFC/MMC へ記録

6) PC との接続

「PC との接続設定 (Fig. 3.8)」の欄は、PC などを親機とする場合に使うため、後ほど第 4 章で解説します。



Fig. 3.8 PC との接続設定

7) その他の設定

子機はIDである「XBee 名称」の他、ユーザーが自由に設定できる「名称」を持っています。「M-Logger の名称を設定 (Fig. 3.9.(1))」をタップすれば、この名称を変更できます。

計測値は「 $y=Ax+B$ 」の形で線形補正をすることができます。これを設定する場合には「校正 (Fig. 3.9.(2))」をタップします。「設定送信」で子機に補正係数が送信され、「設定受信」で現在の補正値が受信できます。



Fig. 3.9 子機の補正係数と名称の設定

8) 熱的快適性の計算機能

「熱的快適性の計算」タブでは熱的六要素を指定して PMV、PPD、SET*を計算することができます。「着衣量を選択」または「活動を選択」ボタンをタップすると着衣や活動の種類を別画面で指定することができます。着衣は1つ以上の項目を選択して「Apply」をタップ、活動は1つの項目をタップすればそれぞれ対応する Clo 値と Met 値が反映されます。

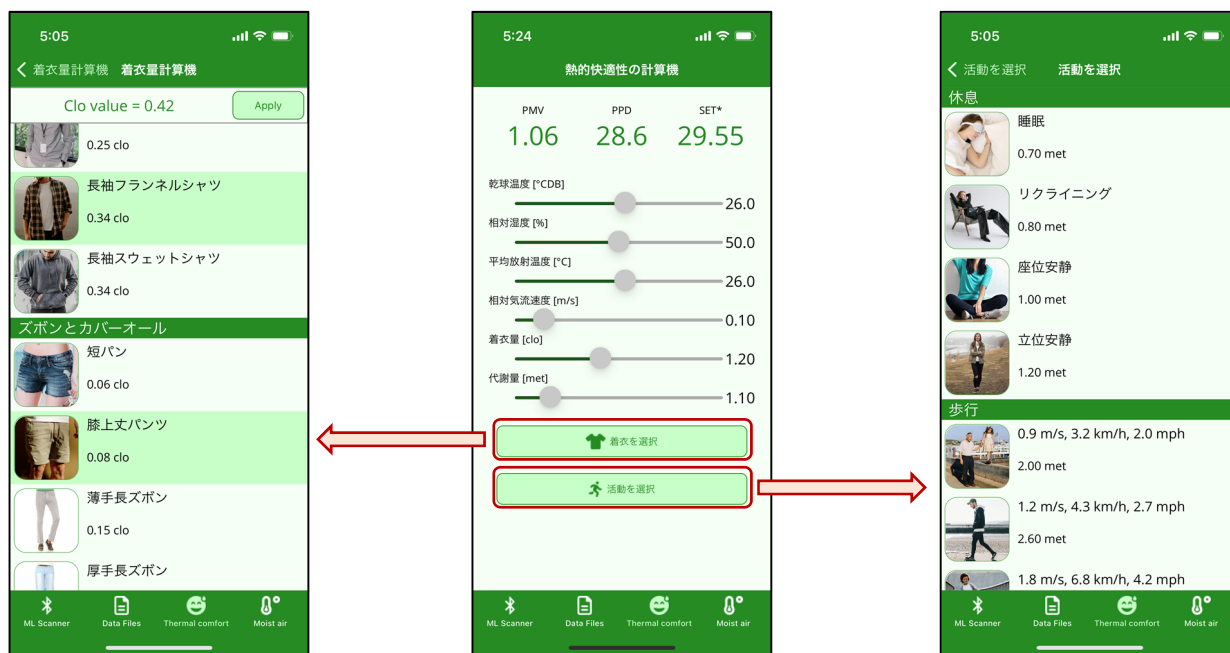


Fig. 3.10 熱的快適性の計算機能

9) 湿り空気物性の計算機能

「湿り空気の計算」タブでは湿り空気の2つの物性から残りの物性を計算できます。上部のコンボボックスを選択(1)すれば、条件とする2つの物性の組み合わせを選択できます。下部のボタンをタップ(2)すれば典型的な温湿度条件へ初期化することができます。



Fig. 3.11 熱的快適性の計算機能

第4章 PC などを使った計測

4.1 準備

1) ハードウェア

PCなどを親機として使う場合には、子機からのデータをXBeeによるZigbee通信で受け取ります。このため「XBee モジュール」をPCなどに取り付けて認識させる必要があります。

XBeeはFig. 4.1に示すように、USB端子を使って取り付けることが一般的です。

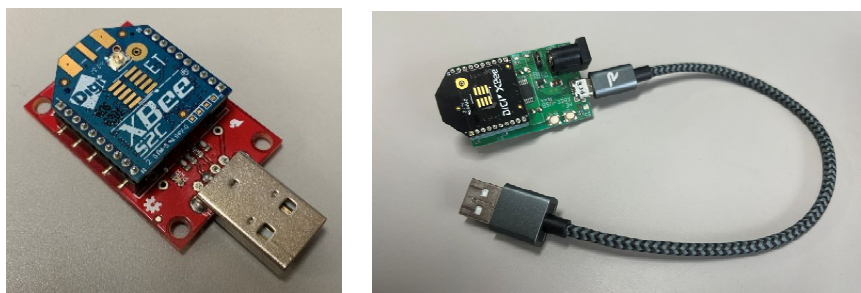


Fig. 4.1 XBee (USB 接続)

現在、市場に出回っているXBeeのバージョンの多くは、S2Cと3.0ですが、いずれも使用することができます。また、以下のUSB接続モジュールについては動作確認済みです。

<https://akizukidenshi.com/catalog/g/gK-06188>

<https://flashtree.com/products/11697>

ただし、上記のUSB接続モジュールを使うためには、FTDI社(www.ftdichip.com)のD2XXドライバをインストールする必要があります。

XBeeには購入時点では初期値が書き込まれており、そのままでは子機と通信できません。この設定を変えるためには、XBeeの製造元であるDigiが公開しているXCTU (Fig. 4.2) というソフトウェアをインストールして操作する必要があります。

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

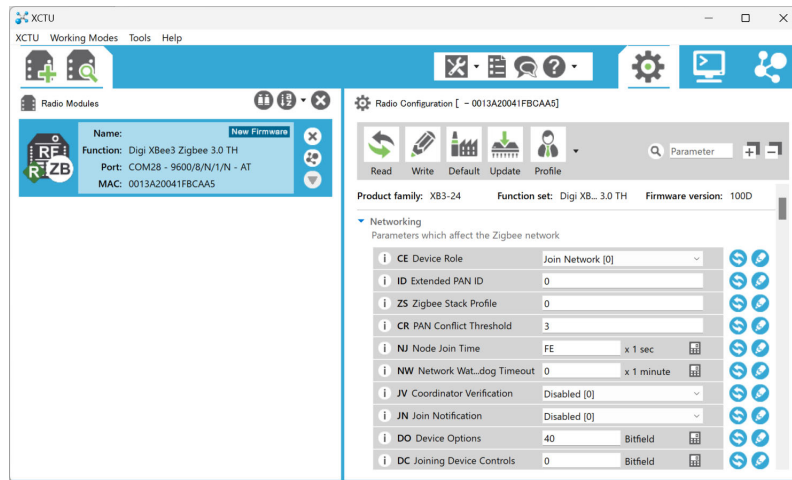


Fig. 4.2 XCTU

具体的には、Table 4.1 に示す項目の設定内容を変える必要があります。

Table 4.1 XBee 端末の設定内容

項目		変更する内容
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

以下で各項目について解説します。

• ID

Zigbee のそれぞれのネットワークは固有の ID を持っており、「PAN ID」と呼ばれています。距離に関わらず、異なる PAN ID を持つ機器は相互に通信しません。M-Logger システムでは標準の「PAN ID」として 16 進数で「0x19800614」を使います。もしも実測現場で M-Logger システムを使った複数のネットワークを作りたい場合には、ネットワークごとにこの値を変更する必要があります。

• SP

親機が子機へのメッセージを保持する時間です。M-Logger の子機は標準では通信時に最大で 1 秒のスリープをするため、その値に合わせて 1 秒とします。

• SN

親機と子機のネットワークはこの値の 3 倍の時間、維持されます。従って、一旦、親子関係が確立すると特別な指令をしない限り、3 時間はネットワークが続きます。

• CE

XBee の用語では親機を「Coordinator」と呼びます。「CE」は親機か否かの設定で、Enabled とすれば親機になります。

• SM

XBee がスリープモードに入る設定で、親機は常時起動する必要があるため、「No Sleep」とします。

- AP

XBee 同士の通信の方式を表しています。子機は標準で API で通信するため、親機も子機に合わせて API enabled とします。

なお、XBee ネットワークの仕様上、Coordinator に接続できる Xbee は 20 台までです。従って、標準では親機に接続できる子機の台数は 20 台が上限となります。この数を増やすには Coordinator を補助する「Router」と呼ばれる XBee を追加する必要があります。Router も 20 台の XBee を管理できるため、Router を 1 台増やせば 40 台の子機から成るネットワークを構築できます。

XBee を Router として機能させるには、Table 4.1 と同じ設定をし、「CE: Coordinator Enable」の項目のみ「Disabled」に変更します。Fig. 4.3 に示すように、コンセントに直接に挿して使う XBee も販売されています。



Fig. 4.3 XBee Wall Router

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

Router の数を増やせば子機の数も増やせますが、無線が混雑してデータを取りこぼす危険性も増加します。小部屋での試験では全部で 80 台の子機を設置した場合、3 秒間隔での計測が限界でした。また、データを取りこぼす可能性は周囲の無線状況にもより、例えば電子レンジが使われているときにはかなり通信に障害が生まれます。

2) ソフトウェア

親機で動かすソフトウェアの最新版は Web サイト (<https://github.com/et0614/mlogger/releases>) からダウンロードできます。

これらは Microsoft .NET 8.0 以上で動作するため、一般の Windows PC だけではなく、Mac や Linux を使うこともできます。Raspberry Pi を使えば実測現場で省スペースに熱環境計測システムを構築できるでしょう。.NET ランタイムは、必要に応じて Microsoft 社の Web サイトからダウンロードしてください。

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

ダウンロードした圧縮ファイルを解凍すると、Fig. 4.4 のディレクトリが得られます。「MLServer.exe」が中心となるソフトウェアですが、その他にも計測を補助するいくつかのソフトウェアが用意されています。

MLServer	: トップディレクトリ
├── data	
│ ├── index.htm	: Web サーバー用のトップページ
│ ├── XBee アドレス.csv	: 収集したデータ (子機ごとの csv ファイル)
│ └──	: その他の Web サーバー用ファイル
├── SimpleWebServer.exe	: 簡易 Web サーバーソフトウェア
├── MLServer.exe	: データ収集を再開するソフトウェア
├── mlnames.txt	: 子機の名称設定用ファイル
├── DDNSUpdater.exe	: DDNS サービスを使うためのソフトウェア
├── setting.ini	: データ収集ソフトウェアの初期設定ファイル
├── ddns.ini	: DDNS サービスの初期設定ファイル
└──	: その他の実行に必要なファイル

Fig. 4.4 ソフトウェアのディレクトリ構成

4.2 ソフトウェアの使い方

前節で解説した XBee を PC などに USB 接続し、MLServer を起動します。Fig. 4.5 に起動画面を示します。

```
#####
#
#           MServer version 1.1.1           #
#                                           #
#           Software for logging data sent from M-Logger #
#           (https://www.mlogger.jp)         #
#                                           #
#####
COM28: Connection succeeded. S/N = 0013A20041FBCAA5
COM4: The connection interface is not open.
COM5: The connection interface is not open.
COM5: The connection interface is not open.
```

Fig. 4.5 MLServer の起動画面

USB 経由で XBee が認識されると、Fig. 4.5 に示すように

COMn: Connection succeeded. S/N = 0013A200xxxxxxx

と表示されます。ただし、COMn の「n」の値は PC などでのどのポートを使うかによって異なります。また「0013A200xxxxxxx」は XBee の固有のアドレスを表しており、「x」の値は XBee ごとに異なります。

これで親機の XBee は通信準備ができたため、子機を立ち上げれば通信が始まります。

スマートフォンの「PC との接続設定」で「PC に接続して記録 (Fig. 4.6.(1))」をタップすると、子機は親機の XBee に向けて計測値を送信し始めます。

親機が子機からのデータを受信すると、その内容は Fig. 4.7 に示すように MLServer の画面に転送されます。図では「4207BD4A」という XBee アドレスを持つ子機からデータ送信を受けています。



Fig. 4.6 PC との接続設定

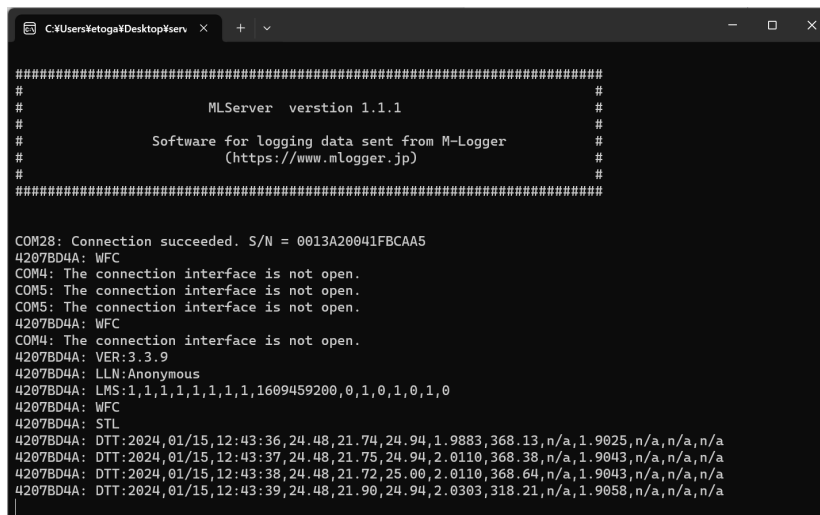


Fig. 4.7 データの受信

親機が受信したデータは「data」ディレクトリの中に csv 形式のファイルで保存されます。

Table 4.2 は計測データの例です。列が表す情報は左から順に、親機がデータを受信した日時、子機が計測した日時、乾球温度[°C]、相対湿度[%]、グローブ温度[°C]、微風速[m/s]、照度[lx]、グローブ温度（電圧）[V]、微風速（電圧）[V]、汎用電圧[V]です（グローブ温度の電圧は M-Logger version3 より廃止）。

Table 4.2 計測データの例

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

現場で長期に M-Logger を設置したい場合には、計測の途中で電池がなくなることがあります。このときに毎回計測の設定をやり直すことは煩雑です。そこで、電池の入れ替えなどで M-Logger を一旦 Off にした場合にも、起動と同時にすぐに以前に設定した内容で計測を開始させるモードが用意されています。

Fig. 4.6.(2)の「常設モードに移行」がそれで、これをタップすると、M-Logger を起動するとすぐに計測が始まるようになります。逆に、スマートフォンからの操作は受け付けなくなります。従って、再びスマートフォンからの操作ができるようにするためには、強制的にこのモードを抜け出す必要があります。このためには Fig. 2.4 に示した Reset スイッチを 3 秒以上、長押しします。LED が 3 回点滅して常設モードが解除されます。

なお、子機は off になると日時が初期化されてしまいます。このため常設モードで再起動した場合には、Table 4.2 の第 1 列に示した、親機がデータを受信した日時を使います。データの送受信によって数秒のズレは発生する可能性があります、相対的には無視しても良い誤差と言えます。

ところで、このように M-Logger を常設して長期計測する場合には、電池の入れ替えが発生すること自体が煩雑です。もしもコンセント電源が手に入るのであれば、ダミー電池を使うという方法もあります。

これは**エラー! 参照元が見つかりません**。に示すような電池形状の端子を持った AC アダプタです。M-Logger には DC/DC コンバータが仕込まれているため、2.0~3.0V の電圧を与えれば内部で電圧が調整されて適正に動きます。従って、AC アダプタもこの範囲にあるものを用意すれば動きます。



Fig. 4.8 ダミー電池による電源供給

Fig. 4.6.(3)の「PAN ID の変更」は Table 4.1 で解説したネットワーク固有の ID を変更するためのボタンです。通常は変更は不要です。

Fig. 4.6.(4)の「通信状況 LED 表示の有効化/無効化」は XBee の通信状況を青色 LED で表示するか否かの設定です (M-Logger version 3 以上のみ)。

第5章 遠隔監視システムの構築

現場での実測の場合、遠隔で計測状況を確認できると便利です。そこで、本章では PC などを使って遠隔監視システムを構築する方法を解説します。

1) Web サーバーで公開するディレクトリ

前章で解説したように、MLServer は計測したデータを「data」ディレクトリに csv 形式でのファイルで保存していきます。従って、このディレクトリを Web サーバーで公開できれば、外部から現在までの計測結果を確認できるようになります。

なお、「data」ディレクトリの中には、csv 形式のファイルの他に、現在の計測状況を表す html ファイル (index.htm) も生成され、これをブラウザで開くと Fig. 5.1 のような表示が得られます。子機のアドレス、名称、計測値と最終の測定時刻の他、これらの物理量から求めた熱的快適性指標 (SET*, PMV, PPD) も表示されます。



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	風速[m/s]	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 「data」ディレクトリ内の「index.htm」

子機の名称は、MLServer と同階層にある「mlnames.txt」に以下のように XBee の下位アドレスと名称の組をリストで記入することで反映できます。

```
42114F7E:SIH-01
42114F57:SIH-02
420BCCD1:SIH-03
420D3FC1:SIH-04
...
```

PMV や PPD などの熱的快適性を計算するためには、着衣量と代謝量を設定する必要がありますが、その値は MLServer と同階層にある「setting.ini」で設定します。

Fig. 5.1 には最終の計測値のみが表示されていますが、過去の値を参照したい場合には、表の最右端のリンクをたどることで、子機ごとの csv ファイルをダウンロードできます。

Fig. 5.2 に遠隔監視の方法を示します。親機に Pocket Wifi を接続し、インターネット越しに子機の状態を監視できるようにします。

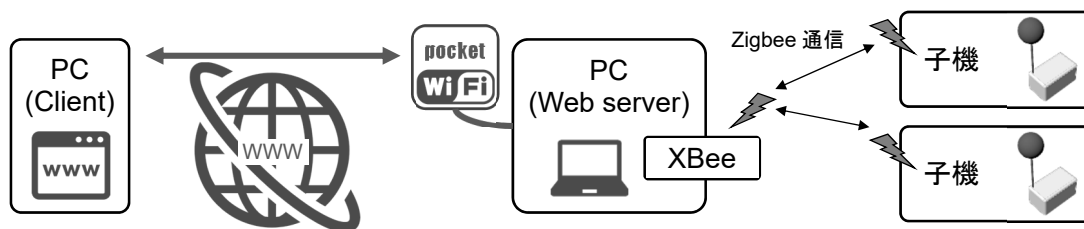


Fig. 5.2 遠隔監視の方法

2) Windows による遠隔監視システム

MLServer と同階層にある「SimpleWebServer」は簡易 Web サーバーのソフトウェアです。これを起動すると「data」ディレクトリの中身をルートとする Web が公開されます。ただし、事前に「netsh」を使って URL を予約する必要があります。詳細は、

<https://docs.microsoft.com/ja-jp/dotnet/framework/wcf/feature-details/configuring-http-and-https>

を参照してください。

正常に Web サーバー稼働すれば、ブラウザで「127.0.0.1」を入力すると Fig. 5.3 のように Basic 認証画面が表示されます。ユーザー名に「user」、パスワードに「pass」を入力すると Fig. 5.1 が表示されます。ユーザー名とパスワードは、SimpleWebServer と同階層にある「setting.ini」の中身を書き換えれば変更できます。



Fig. 5.3 簡易 Web サーバーの Basic 認証画面

親機の Web サーバーに接続するためには、親機の IP アドレスが必要になります。固定 IP が手に入るのであれば問題は無いですが、一般に Pocket Wifi は IP アドレスが変動します。このため、現場での実測を続けている間に IP アドレスが変わって接続できなくなる危険性があります。

このような変動する IP アドレスと固定のホスト名とを紐づける Dynamic DNS (DDNS) というサービスが各社から提供されています。MLServer と同階層にある「DDNSUpdater」は、このような DDNS サービスを使うためのソフトウェアです。対応している DDNS サービスは

- dynamic DO (ddo.jp)
- NoIP (www.noip.com)
- DDNSNow (ddns.kuku.lu)

の3つです。それぞれのサービスに登録し、登録内容を「ddns.ini」に設定し、「DDNSUpdater」を起動すると、定期的に現在のIPがDDNSサービスに送られ、固定ホスト名で親機に接続できるようになります。

例えば dynamic DO で「mydomain.ddo.jp」というドメインに登録し、「mypass」というパスワードを設定した場合には、ddns.iniに以下のように設定します。各行の冒頭の「#」はコメント行を表します。

これでブラウザにIPアドレスではなく「http://mydomain.ddo.jp」を入力することで、親機に接続できるようになります。

```
#DDNS サービスの種類を指定

## ddo.jp ##
service=dynamicDO;
pwd=mypass;           //パスワード
host=mydomain.ddo.jp; //ドメイン・ホスト名

## www.noip.com ##
#service=NoIP;
#usr=user id;         //ID
#pwd=password;       //パスワード
#host=xxx.ddns.net;  //ドメイン・ホスト名

## ddns.kuku.lu ##
#service=DDNSNow;
#usr=user id;         //ID
#pwd=password;       //パスワード

update=300;           //更新時間間隔[sec]
gip=https://api.ipify.org; //IP表示Webサービスのアドレス
```

Fig. 5.4 ddns.ini の設定例

3) Raspberry Pi による遠隔監視システム

遠隔監視のための親機として Raspberry Pi を使えば、Linux 用の堅牢な Web サーバーを使えますし、実測現場で停電が起きた場合にも長時間、電池で駆動を続けることができます。以下に、このための設定を説明します。

Raspberry Pi の初期化については、説明を省きます。以下の説明ではユーザー名を「pi」とし、デスクトップに「MLServer」ディレクトリがあることを前提にします。

・Raspberry Pi への電源接続

現場での長期実測の場合、一般の停電や法定点検のための停電などがあるため、コンセントからの電源供給が確実に続くという保証はありません。このため、コンセントと Raspberry Pi の間に電池を挟むことで、停電時には電池から電源が供給されるようにすればシステムがさらに堅牢になります (Fig. 5.5)。

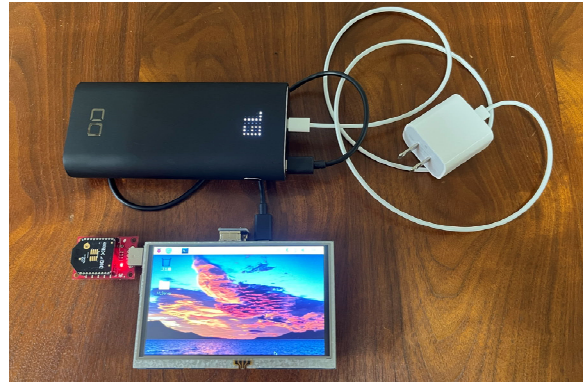


Fig. 5.5 電池経由の電源接続

なお、充電と放電とを同時にする機能を「パススルー機能」と呼びます。パススルー機能を備えていない電池も多いため、電池の購入にあたっては注意が必要です。

Raspberry Pi の消費電力は約 600 mA で 5V のため、3 W 程度です。従って、80 Wh 程度の容量があれば、電池のみで 24 時間の連続稼働ができます。20,000 mA で 3.7 V (74 Wh)、パススルー機能付きの電池であれば 1 万円を下回る金額で調達できます。

・ .NET 実行環境の導入

Raspberry Pi では Linux 系の OS である Raspberry Pi OS が使われています。従って、MLServer を動かすために Raspberry Pi OS 用の .NET 実行環境を導入する必要があります。コンソールで以下のコマンドを入力すれば、.NET 実行環境が導入できます。

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

以下のコマンドで dotnet へのパスを通します。

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

正常に導入できていれば、以下のコマンドで .NET のバージョンが表示されます。

```
$ dotnet --version
```

・再起動時の設定

電池があつたとしても予期しない再起動が起きる危険性があります。このため、再起動時に自動でデータ計測が再開されるように設定します。

再起動時に「MLServer」と「DDNSUpdater」を自動で起動するために、以下のシェルスクリプトを用意します。

```
/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
```

上記のシェルスクリプトを Windows で作成すると改行コードが「`¥r¥n` (CRLF)」となり、Linux 側では正常に認識されないことがあります。「`¥n` (LF)」に変換する必要があります。
作成したシェルスクリプトに対して、以下のコマンドで実行権限を与えます。

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

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

起動時に自動実行するシェルスクリプトを登録するためのディレクトリとファイル (autostart) を用意します。

```
$ mkdir -p ~/.config/lxsession/LXDE-pi
```

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

autostart ファイルの最終行に、先程作ったシェルスクリプトを追加します。

```
/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" //この行を追加
```

```
@lxterminal -e "/home/pi/Desktop/MLServer/DDNSUpdater.sh" //この行を追加
```

・ Apache の導入

Apache は世界で最も使われている Web サーバーで、Raspberry Pi を親機とするのであれば簡単に導入できます。以下のコマンドでインストールします。

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

デスクトップに置かれた「MLServer」ディレクトリ内の「data」ディレクトリを公開するため、以下のよう設定を追加します。

```
                                                                    /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>
```

以下のコマンドで Apache を再起動します。

```
$ sudo service apache2 restart
```

・ 定期的な再起動の設定

Raspberry Pi を定期的に再起動させることで、MLServer が意図せずにクラッシュした場合にも再起動するようにします。以下のコマンドで `crontab` を開きます。

```
$ sudo crontab -e
```

開いたエディタで最終行に以下を追記し、毎日夜 0 時 0 分に再起動させます。

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

4) 遠隔監視画面の調整

MLServer は Fig. 5.6 に示すように、JSON 形式のファイルを使って現在値を書き出しています。遠隔監視で表示される Web ページは、HTML+JavaScript でこの JSON 形式のファイル参照することで作られています。

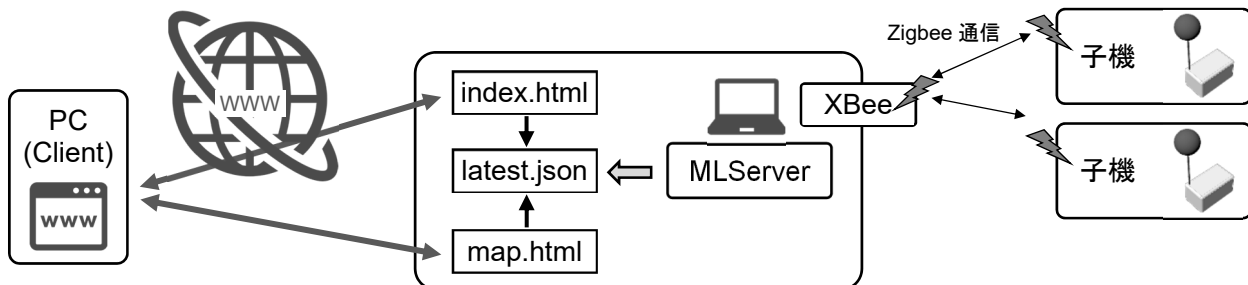


Fig. 5.6 Web ページの生成方法

従って、ユーザーは HTML と JavaScript を使って遠隔監視画面の機能を拡張することができます。標準では、Fig. 5.1 に示した「index.html」の他に、計測結果をマップで表示するための「map.html」が用意されています。

「map.html」は Fig. 5.7 に示すように、計測値と図面を組み合わせ、熱環境のマップを表示させるための道具です。

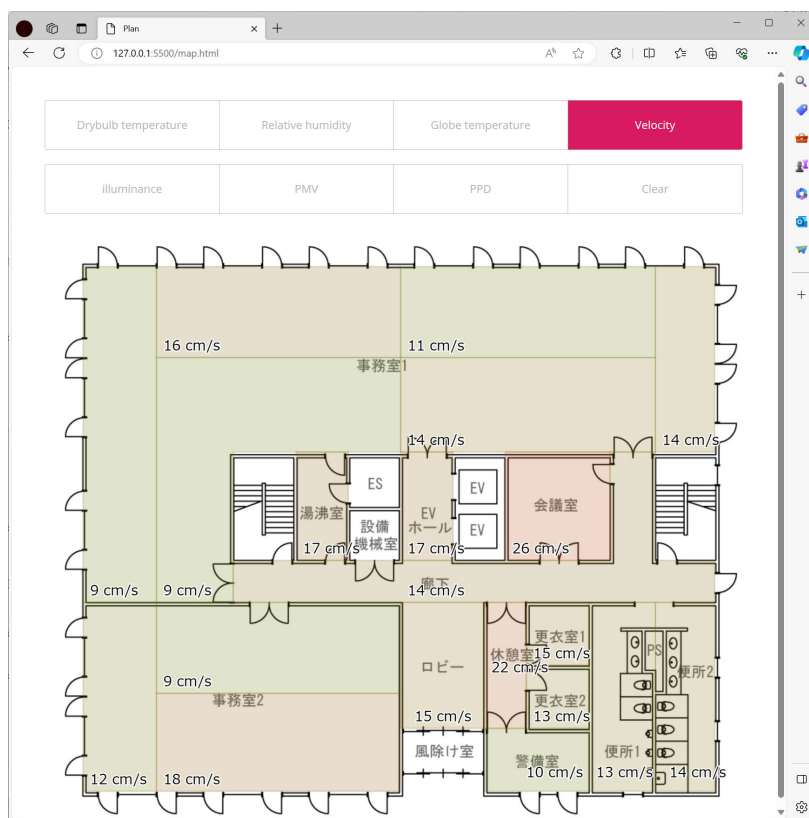


Fig. 5.7 熱環境マップ表示の例

ただし、熱環境マップを有効にするには下処理が必要です。背景となるマップを用意して、マップのどの領域にどの計測器を対応させるかを決め、計測値を書き出す文字の位置を指定する必要があります。

・背景マップの用意

背景とするマップは、横幅 1000 px で png 形式の画像とします。縦幅は自動で調整されるため、任意です。これを「background.png」という名称で「data」ディレクトリに格納します (Fig. 5.8)。

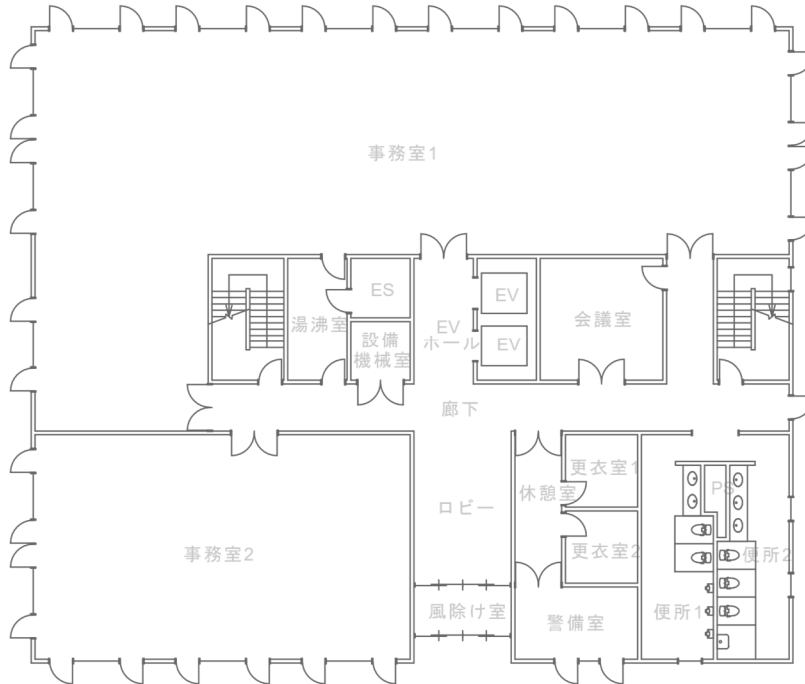


Fig. 5.8 「background.png」の例

・領域の決定

計測値を割り当てるために、背景画像に領域を設定します。領域は長方形または多角形として設定するため、頂点の座標を調べます。座標系は画像の左上を(0,0)とし、右と下を正の値とします。Fig. 5.8 を 20 の領域に分け、それぞれの座標を調べた結果を Fig. 5.9 に示します。

・javascript の編集

領域を長方形または多角形で描画するため、「data」ディレクトリの中の「draw.js」を編集します。「drawRegion」というメソッドの中の switch 文に計測で使っている M-Logger の XBee アドレスを追加し、どの位置に描画するのかを記述します。

長方形の場合には、対角する 2 つの点の x,y 座標を使い、「 $rect(x_0,y_0,x_1,y_1)$ 」と書きます。例えば 4 行と 5 行では「42114F57」というアドレスを持つ M-Logger に対して、(55, 50)から(160, 530)へ向かう長方形領域を割り当てています。これは Fig. 5.9 の(1)の領域です。

n 個の頂点を持つ多角形の場合には「beginShape()」と「endShape(CLOSE)」メソッドの間に頂点を表す「vertex(x_n,y_n)」を追加していきます。16~24 行は Fig. 5.9 の(5)の領域を割り当てた例です。

なお、この機能は「p5.js」を使って実装しており、厳密には、描画の方法は長方形と多角形に限定されません。

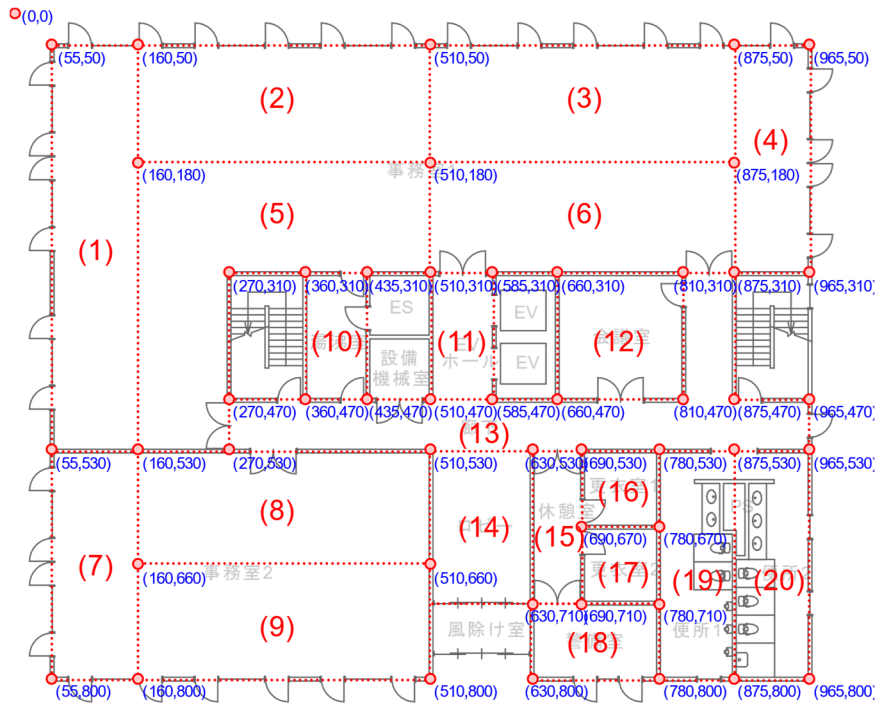


Fig. 5.9 領域を示す頂点の把握

<pre> 1 function drawRegion(mloggerID){ 2 rectMode(CORNERS); 3 switch(mloggerID){ 4 case "42114F57": 5 rect(55, 50, 160, 530); 6 break; 7 case "420BCD82": 8 rect(160, 50, 510, 180); 9 break; 10 case "42114EB8": 11 rect(510, 50, 875, 180); 12 break; 13 case "42114EF0": 14 rect(875, 50, 960, 315); 15 break; 16 case "420BCCD1": 17 beginShape(); 18 vertex(160,180); 19 vertex(510,180); 20 vertex(510,315); 21 vertex(270,315); 22 vertex(270,530); 23 vertex(160,530); 24 endShape(CLOSE); 25 break; 26 case "42114F8F": 27 rect(510, 180, 875, 315); 28 break; 29 case "420D3FC1": 30 rect(55, 530, 160, 800); 31 break; 32 case "420BCD79": 33 rect(160, 530, 510, 660); 34 break; 35 case "420BCCDA": 36 rect(160, 660, 510, 800); 37 break; 38 case "420BCD0B": 39 rect(360, 315, 435, 470); 40 break; 41 case "42114F00": 42 rect(510, 315, 585, 470); </pre>	<pre> 43 break; 44 case "42114E92": 45 rect(660, 315, 810, 470); 46 break; 47 case "420BCD7E": 48 beginShape(); 49 vertex(270,470); 50 vertex(810,470); 51 vertex(810,315); 52 vertex(875,315); 53 vertex(875,470); 54 vertex(960,470); 55 vertex(960,530); 56 vertex(270,530); 57 endShape(CLOSE); 58 break; 59 case "42114E95": 60 rect(510, 530, 630, 710); 61 break; 62 case "420BCDC3": 63 rect(630, 530, 690, 710); 64 break; 65 case "42114E40": 66 rect(690, 530, 780, 620); 67 break; 68 case "42114F7E": 69 rect(690, 620, 780, 710); 70 break; 71 case "42114EFA": 72 rect(630, 710, 780, 800); 73 break; 74 case "420BCDD7": 75 rect(780, 530, 875, 800); 76 break; 77 case "42114F78": 78 rect(875, 530, 960, 800); 79 break; 80 default: 81 break; 82 } 83 } 84 </pre>
--	--

Fig. 5.10 「draw.js」 の例

第6章 通信仕様

親機と子機は Bluetooth または Zigbee 通信で文字列を送受信することで会話をします。ただし、子機は受け身で、親機からのコマンドに応答を返すだけです。親機からのコマンドが正常に子機に伝わると、子機は何かしらの応答を返します。

親機からのコマンドは 3 文字の予約語と、これに続く補足情報で構成されます。予約語の一覧を Table 6.1 に示します。

Table 6.1 予約語一覧

予約語		機能
VER	<u>VER</u> sion	子機のバージョンを取得する
STL	<u>ST</u> art <u>L</u> ogging	計測とロギングを始める
CMS	<u>C</u> hange <u>M</u> easurement <u>S</u> ettings	計測の設定を変更する
LMS	<u>L</u> oad <u>M</u> easurement <u>S</u> ettings	計測の設定を取得する
ENL	<u>EN</u> d <u>L</u> ogging	計測とロギングを終える
SCF	<u>S</u> et <u>C</u> orrection <u>F</u> actor	補正係数を設定する
LCF	<u>L</u> oad <u>C</u> orrection <u>F</u> actor	補正係数を取得する
CLN	<u>C</u> hange <u>L</u> ogger <u>N</u> ame	機器の名称を変更する
LLN	<u>L</u> oad <u>L</u> ogger <u>N</u> ame	機器の名称を取得する
CBV	<u>C</u> alibrate <u>V</u> elocity <u>V</u> oltage	無風電圧を自動校正する
CBT	<u>C</u> alibrate <u>T</u> emperature	温度を自動校正する
SCV	<u>S</u> tart <u>C</u> alibrate <u>V</u> elocity <u>V</u> oltage	無風電圧の手動校正を始める
ECV	<u>E</u> nd <u>C</u> alibrate <u>V</u> elocity <u>V</u> oltage	無風電圧の手動校正を終える

以下に、それぞれのコマンドの詳細を記します。親機が送信するコマンドは予約語と補足情報を含めて 72 文字以内です。1 つのコマンドの終了を判断するため、最後に「 \r (Carriage Return)」をつけます。

・ VER (VERsion) コマンド

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

子機の応答: 「Vers: *int.int.int*」

(*int* はバージョンを表す整数値)

・ STL (STart Logging) コマンド

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	現在時刻										t/f	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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

補足情報:

3~12: 現在の時刻。time_t (long) 型の UNIX 時間。

13: 結果を Zigbee 通信で送信するか否か。「t」で送信する、「f」で送信しない。「e」で無限計測モードへ。

14: 結果を Bluetooth 通信で送信するか否か。「t」で送信する、「f」で送信しない。

15: 結果を MicroSD に書き出すか否か。「t」で書き出す、「f」で書き出さない。

子機の応答: 「STL」

以降、定期的に「DTT:yyyy,MM/dd,HH:mm:ss,float,float,float,float,float,float,float,float」の応答が続きます。

前半の「yyyy,MM/dd,HH:mm:ss」は日付と時刻を表します。後半の7つのfloatはそれぞれ、乾球温度 [°C]、相対湿度 [%]、グローブ温度[°C]、風速 [m/s]、照度 [lx]、グローブ温度電圧 [V]、微風速電圧 [V]、汎用電圧 1~3[V]、を表します。

・ CMS (Change Measurement Settings) コマンド

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	温湿度					t/f	グローブ温度					t/f	風速					t/f	照度		
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
照度			測定開始時刻										t/f	AD1				t/f	AD2					
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	
AD2	t/f	AD3					t/f	¥r																

補足情報:

3: 温湿度を計測するか否か。「t」は計測する、「f」は計測しない。

4~8: 温湿度の計測時間間隔。5桁以内の整数。

9: グローブ温度を計測するか否か。「t」は計測する、「f」は計測しない。

10~14: 温湿度の計測時間間隔。5桁以内の整数。

15: 風速を計測するか否か。「t」は計測する、「f」は計測しない。

16~20: 温湿度の計測時間間隔。5桁以内の整数。

21: 照度を計測するか否か。「t」は計測する、「f」は計測しない。

22~26: 温湿度の計測時間間隔。5桁以内の整数。

27~36: 測定開始時刻。time_t (long) 型の UNIX 時間。

37: AD1 を計測するか否か。「t」は計測する、「f」は計測しない。

38~42: AD1 の計測時間間隔。5桁以内の整数。

43: AD2 を計測するか否か。「t」は計測する、「f」は計測しない。

44~48: AD2 の計測時間間隔。5桁以内の整数。

49: AD3 を計測するか否か。「t」は計測する、「f」は計測しない。

50~54: AD3 の計測時間間隔。5桁以内の整数。

55: 近接センサを有効にするか否か

子機の応答: 「CMS:bool,int,bool,int,bool,int,bool,int,long,int,long,int,long,int」

bool は計測の真偽を表す 0/1、int は計測間隔を表す整数値、long は計測開始時刻を表す UNIX 時間で、前から順に、温湿度、グローブ温度、風速、照度、汎用電圧 1~3、近接センサの有効無効、の設定です。

・ LMS (Load Measurement Settings) コマンド

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

子機の応答: 「LMS:bool,int,bool,int,bool,int,bool,int,long,int,long,int,long,int」

bool は計測の真偽を表す 0/1、int は計測間隔を表す整数値、long は計測開始時刻を表す UNIX 時間で、前から順に、温湿度、グローブ温度、風速、照度、汎用電圧 1~3、近接センサの有効無効、の設定です。

・ ENL (ENd Logging) コマンド

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

子機の応答: 「ENL」

・ SCF (Set Correction Factor) コマンド

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	乾球温度 A				乾球温度 B				相対湿度 A				相対湿度 B				グローブ A			グ	
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
ローブ B			照度 A				照度 B				風速 A				風速 B				風速無風電圧			¥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

補足情報:

共通: 補正值=A×実測値+B

3~6: 乾球温度の補正係数 A×1000。A の範囲は 0.800~1.200 とします。

7~10: 乾球温度の補正係数 B×100。B の範囲は-3.00~3.00 とします。

11~14: 相対湿度の補正係数 A×1000。A の範囲は 0.800~1.200 とします。

15~18: 相対湿度の補正係数 B×100。B の範囲は-9.99~9.99 とします。

19~22: グローブ温度の補正係数 A×1000。A の範囲は 0.800~1.200 とします。

23~26: グローブ温度の補正係数 B×100。B の範囲は-3.00~3.00 とします。

27~30: 照度の補正係数 $A \times 1000$ 。A の範囲は 0.800~1.200 とします。

31~34: 照度の補正係数 $B \times 1$ 。B の範囲は-999~999 とします。

35~38: 風速の補正係数 $A \times 1000$ 。A の範囲は 0.800~1.200 とします。

39~42: 風速の補正係数 $B \times 1000$ 。B の範囲は-0.500~0.500 とします。

43~46: 風速の無風電圧 $\times 1000$ 。範囲は 1.400~1.500 とします。

・ LCF (Load Correction Factor) コマンド

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

・ CLN (Change Logger Name) コマンド

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	機器の名称																			¥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

子機の応答: 「CLN:機器名称」

・ LLN (Load Logger Name) コマンド

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

子機の応答: 「LLN:機器名称」

・ CBV (Calibrate Velocity Sensor) コマンド

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	秒数				¥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

子機の応答: 「CBV:残り秒数」

立ち上げのために 60 秒間待機した後、設定した秒数だけ、風速計の電圧値が計測され、その平均値を使って無風電圧が初期化されます。ただし、 $1.4V < \text{無風電圧} < 1.55V$ の範囲にある場合のみ、有効とされます。子機からの送信される残り秒数は、補正完了までの待ち時間です。

・ CBT (Calibrate Temperature Sensor) コマンド

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	秒数					¥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	

子機の応答: 「CBT:残り秒数」

立ち上げのために 60 秒間待機した後、設定した秒数だけ、乾球温度計とグローブ温度計の状態値が計測され、乾球温度計の値を目的変数、グローブ温度計を説明変数とした回帰係数が求められます。これらの回帰係数によってグローブ温度計の補正係数が初期化されます。ただし、回帰直線の傾きが 0.7~1.3、切片が-2~2 の範囲にある場合のみ、有効とされます。子機からの送信される残り秒数は、補正完了までの待ち時間です。

・ SCV (Start Calibrate Velocity Voltage) コマンド

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

子機の応答: 「SCV:電圧[V]」

毎秒、風速電圧が計測されます。

・ ECV (End Calibrate Velocity Voltage) コマンド

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

子機の応答: 「ECV」

【参考文献】

- 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/ajjt.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