

測定装置作成 マニュアル (ver1)

農業試験研究センター

企画調整部

企画・スマート農業研究担当

TEL:0952-45-2142

本マニュアルでは、佐賀県成果情報「スマートフォン等でハウス内の温度をリアルタイムに確認できる安価な測定装置の作成」に掲載の安価に自作できる測定装置の作成方法を紹介します。

本装置は、簡易に作成できる測定装置の一例です。

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

1. 概要説明 - 1

本体



設置道具



Wi-Fi環境

通知機能



クラウド機能



仕組み

- ・ マイコンにプログラムを書き込む。(少し専門の知識が必要)
- ・ マイコンがセンサーから取得したデータを無料のクラウド上にアップロード。
- ・ 低温遭遇時間をスマートフォン等に通知！

留意事項

- ・ マイコンは電池で動かない。
→ コンセント(電源)が必要！
- ・ Wi-Fiを利用する。
→ Wi-Fiが届くことが条件。

経費
約2.4万
円

1. 概要説明 - 2

経費内訳(一例)

	活用機器類	初期経費(概算)	維持経費(概算)
本体	M5stickCPlus (マイコン)	4,200円	0円
	ENVⅢ(環境センサ)	1,200円	0円
	M5Stack用GROVE互換ケーブル(200cm)	500円	0円
Wi-Fi環境	Aterm MPO2LN(モバイルルーター)	7,500円	0円
	プリペイドSIM	5,000円	5,000円
クラウド機能	Ambient	0円	0円
通知機能	LINE Notify	0円	0円
設置道具	ウォルボックス	1,200円	0円
	延長コード 10m	2,500円	0円
	園芸支柱(緑)	200円	0円
	USBコンセント	1,000円	0円
	プラスチックカップ	100円	0円
	紐	100円	0円
	合計	23,500円	5,000円

※クラウド機能や通知機能は無料で使用できますが、ソフトウェアの利用には別途通信料が発生します。

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

まずは完成写真！！



👉 箱の中の様子



2. 準備するもの、必要な環境 - 準備するもの

- ① M5StickCPlus
- ② 環境センサユニット ENV III
- ③ Grove互換ケーブル
- ④ データ転送 & 給電用のtypeC-USBケーブル
- ⑤ モバイルルーター
- ⑥ プリペイドSIM



～以下設置環境に合わせて準備～

ウォルボックス、延長コード、USBコンセント、プラスチックカップ、乾燥剤、ジッパー付袋、防水バッグ、結束バンド(紐)、園芸支柱、ビニールテープなど

2. 準備するもの、必要な環境

- 準備するもの

① M5StickCPlus (約4,200円)

バッテリーを内蔵しており、容量は120 mAhです。

満充電にかかる時間は、40分程度

満充電にした後、電源から外して動く時間は、2時間程度(※プログラム内容による。)

ですので、基本的には常時電源に繋いだ状態で使う必要があります。

【基本操作】

電源ボタン

- ・6秒長押しで電源OFF
- ・2秒長押しで電源ON



2. 準備するもの、必要な環境

- 準備するもの

②環境センサユニット ENVⅢ (約1,200円)

内部には、温湿度センサ「SHT30」と、
気圧センサ「QMP6988」
が内蔵されています。



2. 準備するもの、必要な環境

- 準備するもの

③ Grove互換ケーブル

10cm/20cm/50cm/1m/2mが販売されています。
ENVⅢには、20cmのケーブルがついてきます。



④ データ転送 & 給電用のtypeC-USBケーブル

マイコン(M5StickCPlus)に、プログラムを書き込む
ときには、データ転送用のものが必要になります。
書き込んだ後に、現場に設置する時は給電用でも
よいです。



(注) 給電しかできないケーブルではマイコンにデータを書き込めません。

2. 準備するもの、必要な環境

- 準備するもの(購入できるサイト)

スイッチサイエンス

“<https://www.switch-science.com/>”

秋月電子

“<https://akizukidenshi.com/catalog/top.aspx>”

2. 準備するもの、必要な環境

- 準備するもの

⑤ モバイルルーター

simフリーのモバイルWi-Fiルーターは、8,000～9,000円程度で購入できます。

※simフリーでないと、次のプリペイドsimに制限が出てきます。

(例)



2. 準備するもの、必要な環境

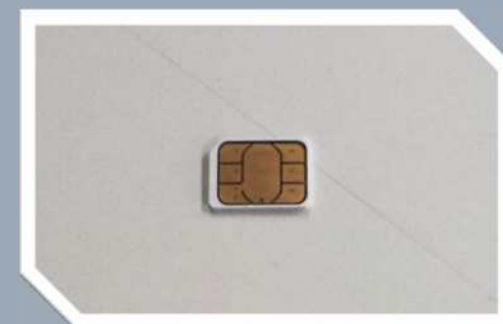
- 準備するもの

⑥プリペイドsim

GB数は気にせず、有効期間と価格を見て、一番安いプリペイドsimを購入します。1年間5,000円程度です。

※マイコンによるデータ通信量は超低容量です。
センサ6台を稼働させて、1日あたり1MB
(60日間で60.05MB)でした(試験場調査)。

(例)



2. 準備するもの、必要な環境

- 必要な環境

①電源となるコンセントがあること

マイコンを電池のみで作動させた場合、翌朝まで持ちません。

②Wi-Fiが届くこと

クラウド(Ambient)へのデータ送信時とLINE通知時に、Wi-Fiが必要です。
ただし、マイコンの画面で気温を確認する場合は、Wi-Fiは不要です。

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

3. Arduino IDEとドライバーFTDI Driversのダウンロード方法

- Arduino IDE、ドライバーFTDI Driversとは？

① Arduino IDE

マイコンにプログラムを書き込むソフトウェア

② ドライバーFTDI Drivers

Arduino IDEを動かすために必要なソフトウェア

3. Arduino IDEとドライバーFTDI Driversのダウンロード方法

- Arduino IDEのダウンロード※最新Verで構いません。

Arduino IDE のダウンロードサイトから、自分のPCにあったソフトウェアをダウンロードします。

※Windowsの場合は、以下のものをクリックしてダウンロードする。
「Windows win 10 and newer,64 bits」

3. Arduino IDEとドライバーFTDI Driversのダウンロード方法

- Arduino IDEのダウンロード

自分のパソコンに「**arduino-2.3.2-windows_64bit(1).exe**」というアプリケーションがダウンロードされるので、ダブルクリックします。

そのまま指示に従い、「**Arduino IDE**」をダウンロードします。

※特に設定を変える必要はありません。

すべて「はい」でよいです。いくつかダウンロードが行われます。

3. Arduino IDEとドライバーFTDI Driversのダウンロード方法 - ドライバーFTDI Driversのダウンロード

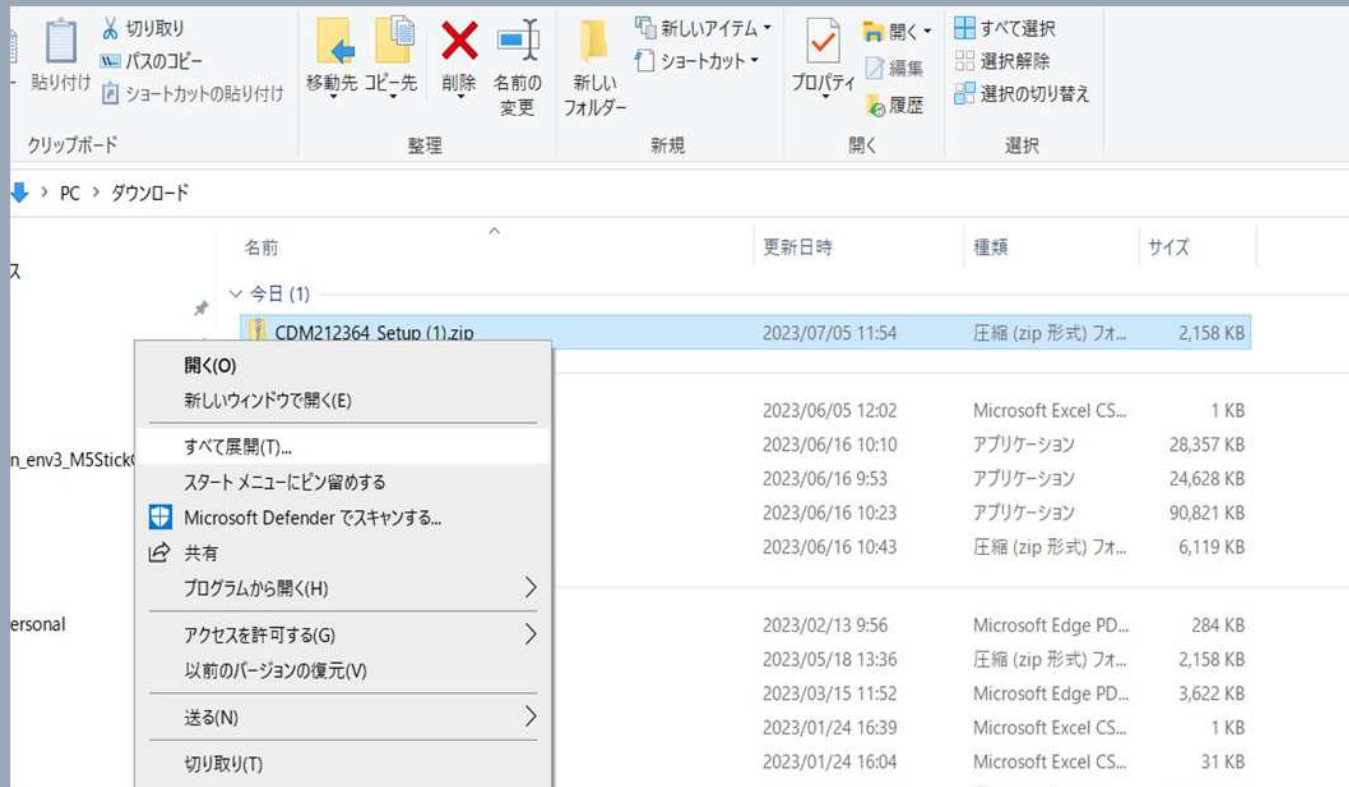
M5STACK公式サイト「SOFTWARE」→「Developer Tools」→USB DRIVER & OPEN SOURCE LIBRARYの「FTDI Drivers」をクリックし、FTDIChipという外部のサイトに飛びます。

FTDIChipのサイトで自分のPCにあったものをダウンロードします。

※Windowsの場合は、**Windows(Desktop)***にある「**setup executable**」をクリック。

3. Arduino IDEとドライバーFTDI Driversのインストール方法 - ドライバーFTDI Driversのダウンロード

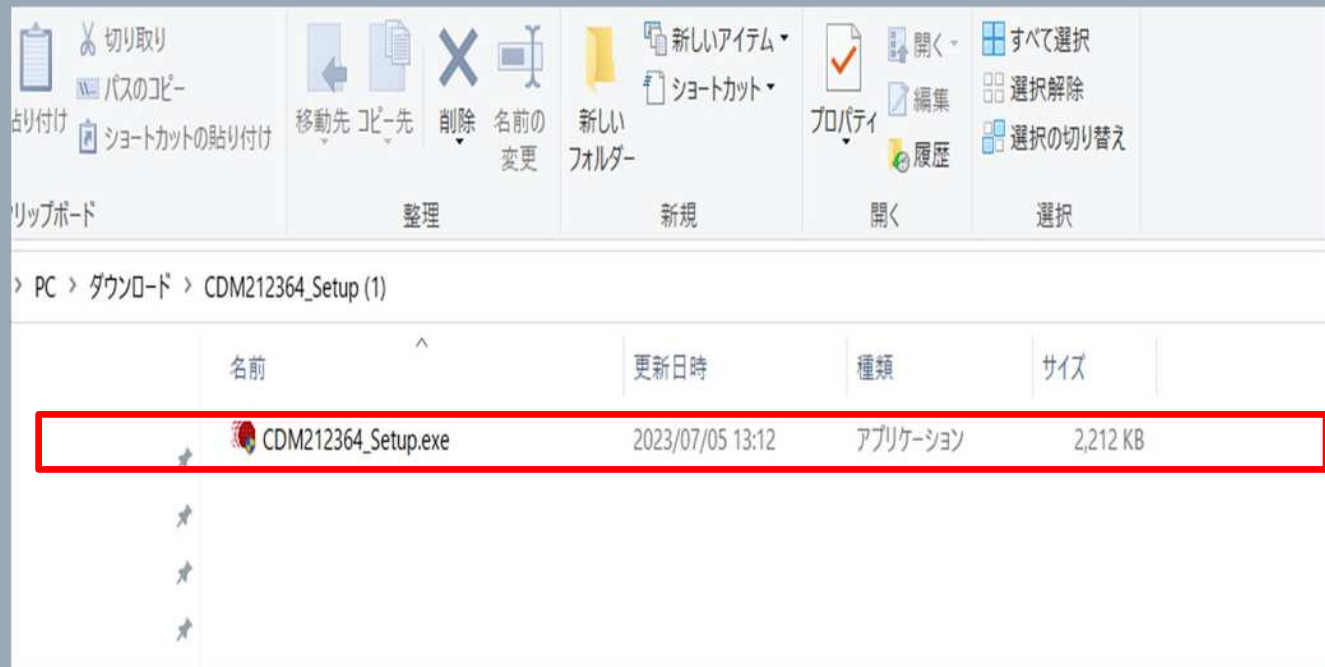
「CDM212364_Setup (1).zip」というzipファイルがダウンロードされるので、右クリックで「すべて展開」してください。



3. Arduino IDEとドライバーFTDI Driversのインストール方法 - ドライバーFTDI Driversのダウンロード

展開後に表示されるダウンローダーをダブルクリックします。
そのまま指示に従い、ドライバーFTDI Driversをダウンロードします。

※表示されるメッセージは
すべて「はい」でよい
です。



4. プログラムについて

プログラム編をご参照ください。

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

5. クラウドサービス「Ambient」

Ambientは、無料のクラウドサービスです。

8(チャンネル) × 8(データ) = 64データまでは無料で利用可能です。

- ・データをグラフ化し、リアルタイムに確認できます。
- ・データはcsvファイルで3ヶ月前までダウンロードすることが可能です。

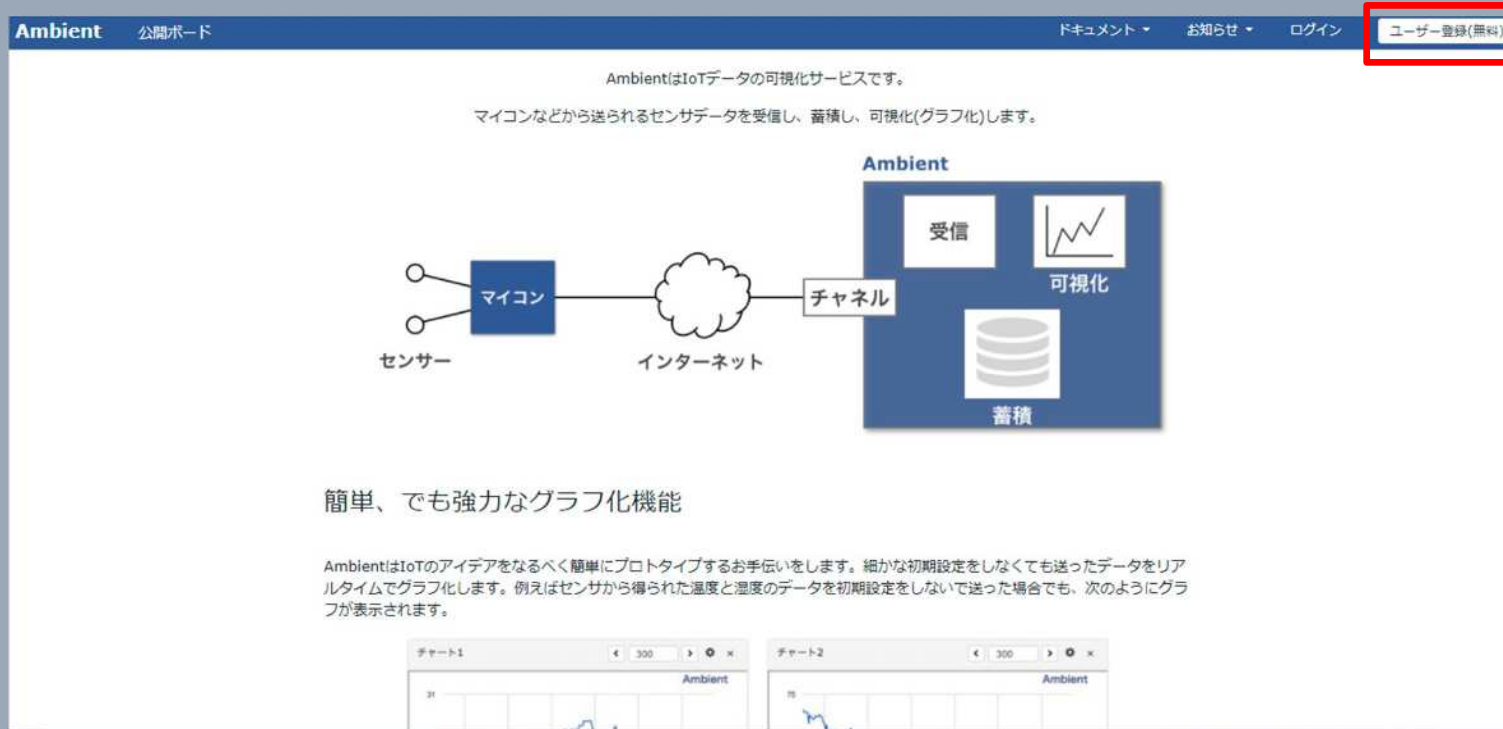
※1回のダウンロードで、最大35日分をダウンロードできます。



5. クラウドサービス「Ambient」 - 設定方法

<https://ambidata.io/>

上記URLにアクセスし、右上の「ユーザー登録(無料)」をクリックします。
メールアドレスと任意のパスワードで登録してください。



The screenshot shows the Ambient website interface. At the top right, the navigation menu includes 'ドキュメント', 'お知らせ', 'ログイン', and 'ユーザー登録(無料)', which is highlighted with a red rectangular box. The main content area features a diagram illustrating the data flow: 'センサー' (Sensors) connect to a 'マイコン' (Microcontroller), which sends data through 'インターネット' (Internet) via a 'チャンネル' (Channel) to the 'Ambient' cloud service. The cloud service is depicted as a blue box containing '受信' (Reception), '可視化' (Visualization), and '蓄積' (Storage) components. Below the diagram, the text reads 'AmbientはIoTデータの可視化サービスです。マイコンなどから送られるセンサーデータを受信し、蓄積し、可視化(グラフ化)します。' (Ambient is an IoT data visualization service. It receives sensor data from microcontrollers, stores it, and visualizes it (graphs it).). A section titled '簡単、でも強力なグラフ化機能' (Simple, but powerful graphing function) explains that Ambient makes it easy to prototype IoT ideas without initial settings. It shows two example graphs, 'チャート1' and 'チャート2', both displaying data trends over time.

5. クラウドサービス「Ambient」

- チャンネル、ライトキーについて

1アカウントにつき、チャンネルは8つ作成可能です。

それぞれのチャンネルには、チャンネルIDとリードキー、ライトキーが付与されます。

マイコンのプログラム作成の時には、チャンネルIDとライトキーを使うので、ここからコピー&ペーストして、プログラムに書き込んでください。

ユーザーキー: [REDACTED]

チャンネル名	チャンネルID	リードキー	ライトキー	ダウンロード	データ削除	設定
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	↓	🗑️	⋮
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	↓	🗑️	⋮
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	↓	🗑️	⋮
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	↓	🗑️	⋮

チャンネルを作る デバイスキーを設定する

5. クラウドサービス「Ambient」

- データ保存場所について

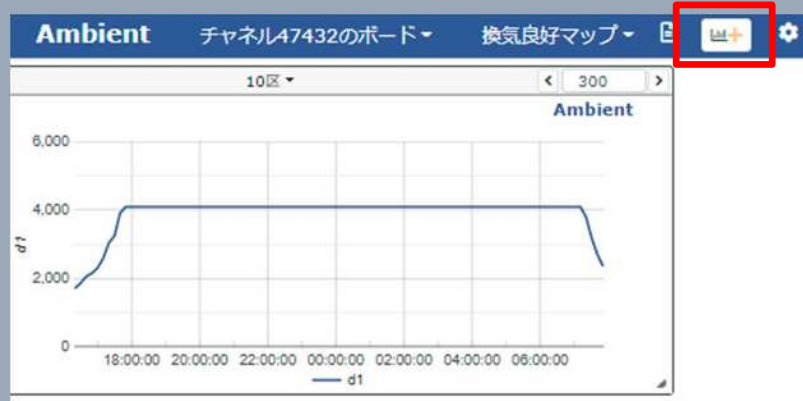
ひとつのチャンネル中に、データ保存場所が8つ付与されます。
プログラムで、このデータ保存場所(1~8)を指定します。
すると、指定したチャンネルの指定した保存場所にデータが送信されます。

データ-1	<input type="text" value="データ名1"/>	 ▼	データ-2	<input type="text" value="データ名2"/>	 ▼
データ-3	<input type="text" value="データ名3"/>	 ▼	データ-4	<input type="text" value="データ名4"/>	 ▼
データ-5	<input type="text" value="データ名5"/>	 ▼	データ-6	<input type="text" value="データ名6"/>	 ▼
データ-7	<input type="text" value="データ名7"/>	 ▼	データ-8	<input type="text" value="データ名8"/>	 ▼

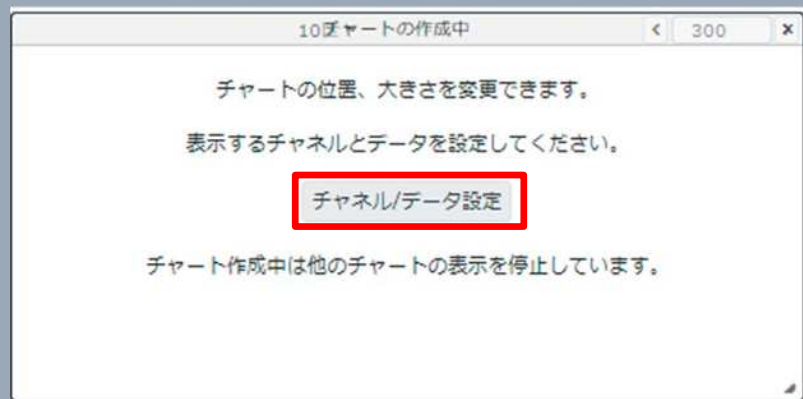
5. クラウドサービス「Ambient」

- グラフ化について

「グラフプラス」ボタンをクリックします。



「チャンネル/データ設定」をクリックします。



5. クラウドサービス「Ambient」

- グラフ化について

- ・「チャート名」を記入
- ・「チャンネル」を選択
- ・「データ番号」を選択
 - ※2つ以上のデータも選択可能です。
左軸と右軸に分けることも可能です。
- ・「設定する」をクリック

(参考)

- ・最大値、最小値を設定可能です。
- ・表示件数の変更も可能。最新のデータがこの数字の数だけグラフ化されます(最大3,000)。日付指定も可。

The screenshot shows the configuration page for a chart in the Ambient cloud service. The interface is in Japanese and includes the following elements:

- Channels:** A dropdown menu for 'チャンネル' (Channel) showing 'チャンネル [redacted] (チャンネルID: [redacted])'. There are also input fields for '透過度' (Transparency) and 'z-index'.
- Chart Type:** A dropdown menu for 'チャート種類' (Chart Type) set to '折れ線グラフ' (Line Graph).
- Data Series:** Eight data series (d1 to d8) are listed, each with radio buttons for '表示なし' (None), '左軸' (Left Axis), and '右軸' (Right Axis).
- Axes:** Input fields for '最小値' (Minimum) and '最大値' (Maximum) for both '左軸' (Left Axis) and '右軸' (Right Axis). There are also checkboxes for 'log?' (log scale).
- Display Settings:** A text input for '表示件数' (Number of items to display) set to '300', and a checkbox for '日付指定' (Date specification).
- Aggregation:** A dropdown for '集計' (Aggregation) followed by 'の' (of) and another dropdown.
- Buttons:** '設定せずに閉じる' (Close without settings) and '設定する' (Apply settings).

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. **通知サービス「LINE Notify」**
7. Wi-Fi環境の作り方
8. 設置

6. 通知サービス「LINE Notify」

- ・ **LINE Notify**の機能を使うことで、LINEに通知を送ることができます。
- ・ まずは自分のLINEで、LINEの検索から「LINE Notify」と検索し、友だち追加をしてください。
- ・ 次ページからトークンを発行します。
トークンとは、マイコンからLINEに通知を飛ばすときに必要な「LINE Notify」のパスワードで、マイコンのプログラムに書き込む必要があります。

6. 通知サービス「LINE Notify」

- トークン発行方法

<https://notify-bot.line.me/ja/>

- ①上記URLにパソコンからアクセスし、右上のログインからメールアドレスとパスワードでログインしてください。
(※LINEのメアドとPWは、「アカウント」から確認できます。)
- ②ログインしたら、マイページを開きます。
- ③アクセストークンの発行(開発者向け)から、「トークンを発行する」をクリックします。

6. 通知サービス「LINE Notify」

- トークン発行方法

- ・トークン名を記入します。
今回は、「自作温湿度センサ」
(後から変更不可)
- ・「1対1でLINE Notifyから通知を受け取る」を選択します。
※グループラインへの通知も可能です。
- ・「発行する」をクリックすると、トークンが発行されます。
※発行されたトークンはこの時しか確認できないため、
必ず保存してください！

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

7. Wi-Fi環境の作り方

- ・プリペイドsimをモバイルWi-Fiルーターに差します。
- ・購入したプリペイドsimによって、すぐに使えるもの、開通手続きが必要なものがあります。
- ・基本的には、モバイルWi-FiルーターでのAPN(アクセスポイント名)設定が必要です。説明書を見ながら、一つずつ行ってください。
- ・近くにWi-Fi環境があれば不要。



7. Wi-Fi環境の作り方

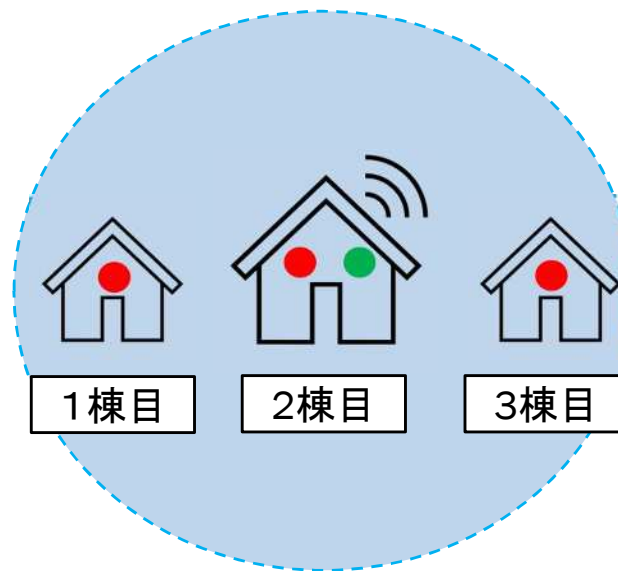
- Wi-Fiが届く範囲

1棟設置の場合



1棟目

3棟設置の場合



1棟目

2棟目

3棟目

● センサー

● Wi-Fiルーター

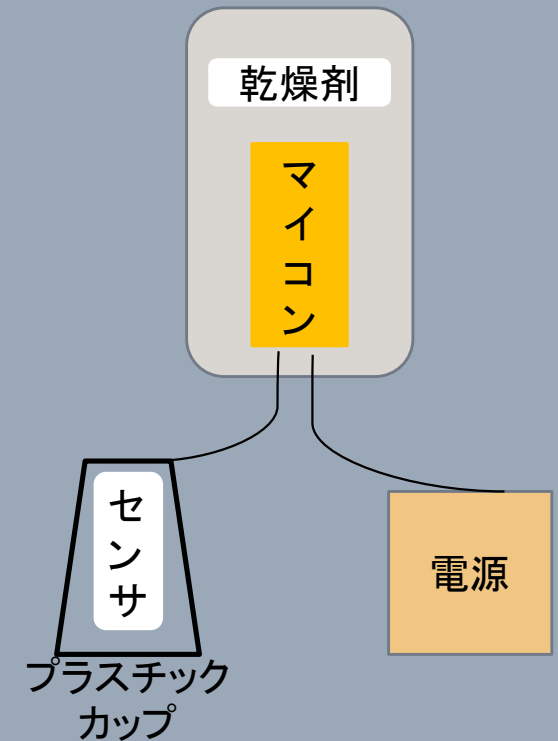
● Wi-Fiが届く範囲 ※遮蔽物など環境により範囲は変わる。

～目次～

1. 概要説明
2. 準備するもの、必要な環境
3. Arduino IDEとドライバーFTDIDriversのダウンロード方法
4. プログラムについて
5. クラウドサービス「Ambient」
6. 通知サービス「LINE Notify」
7. Wi-Fi環境の作り方
8. 設置

8. 設置

- ・センサは、プラスチックコップの底に穴をあけたり、植木鉢ポット等を活用して、簡単なカバーとして設置します。
※カバーの効果：直達光による温度上昇回避
粉塵防止
- ・接続部をできるだけ露出しないようにしてください。
(ビニールテープなどを使って対策を！)
- ・センサ以外はほぼすべてウォルボックスに入れます。
ウォルボックスの中には一緒に乾燥剤入れることをおすすめします。



8. 設置

- 防水について

【前提】自作での完璧な防水は困難です。
袋に入れると中で結露が生じてしまいます。

- ・地温を測定する場合は、センサを防水バッグに入れ、設置します。

(例) 市販防水バッグ (約14,000円)
Aquapac Wire Through Case S
(TC Clip付き) 型番 548



8. 設置

- Wi-Fiに繋ぐ

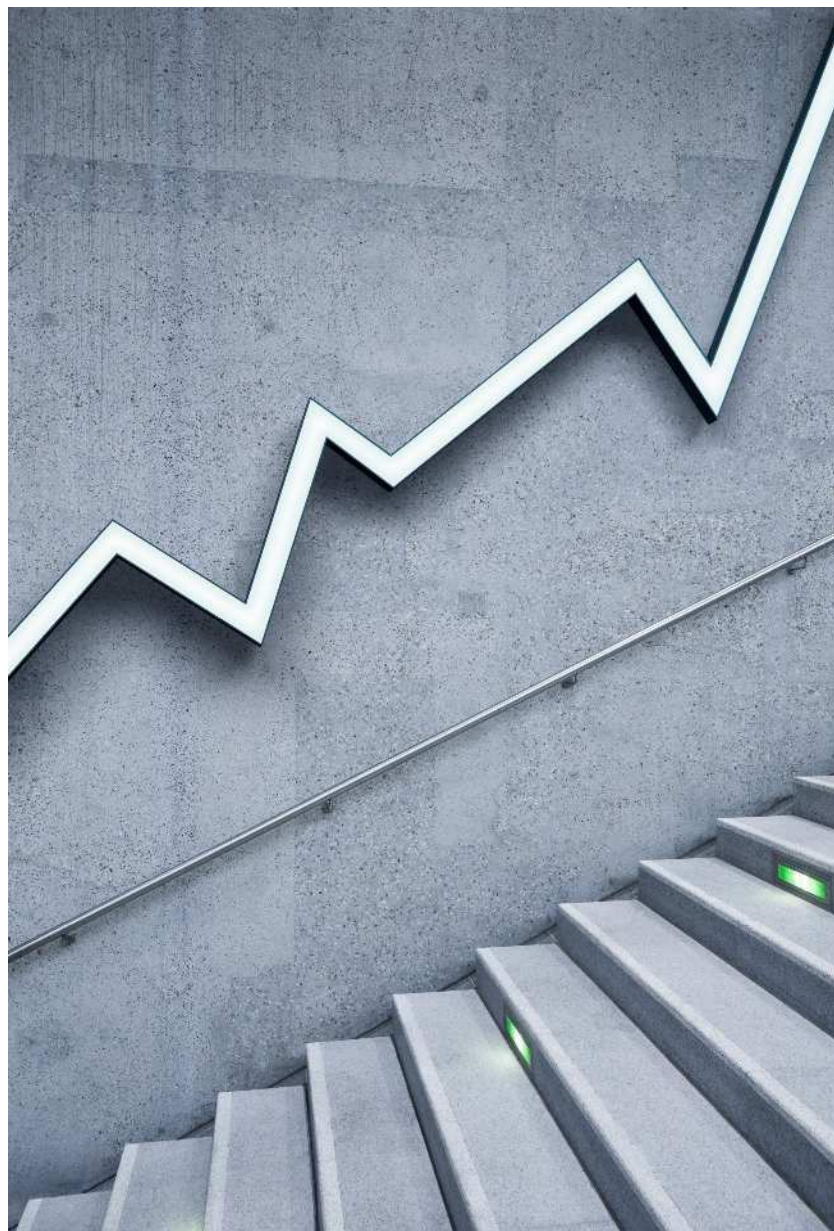
マイコンはUSBからの給電開始で電源ONにはなりません。
Wi-Fiにすぐには繋がらないことがあります。そのようなときは再起動してください。

【再起動の方法】

電源ボタン

- ①6秒長押しで電源OFF
- ②2秒長押しで電源ON





測定装置作成 マニュアル (プログラム編)

ver 1

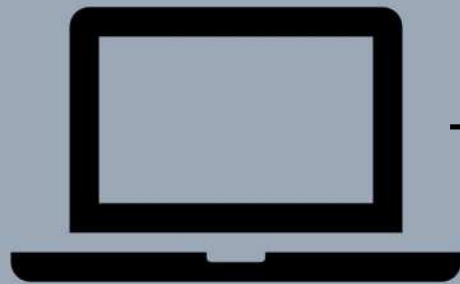
農業試験研究センター

企画調整部

企画・スマート農業研究担当

Arduino IDEで書いたプログラムをマイコンに書き込むことで、マイコンはその働きをします。

PCとマイコンは、USB-TypeCケーブルで接続し、書き込みを行います。



ArduinoIDE



M5StickCPlus

～目次～

1. Arduino IDEの初期設定

- 環境設定
- ボードの追加
- ライブラリの追加

2. プログラムの概要

3. マイコンへの書き込み方法

4. 今回のプログラムの内容について

1. Arduino IDEの初期設定

- 環境設定

「ファイル」→「環境設定」

「追加ボードマネージャのURL」の右端の小さな四角のアイコンをクリック

https://raw.githubusercontent.com/espressif/Arduino-esp32/gh-pages/package_esp32_index.json

上の一文をコピー&ペーストしてください。(GitHubサイトより引用)

※コピー&ペーストした後に、改行が入らないように注意してください!

1. Arduino IDEの初期設定

- ボードの追加

「ツール」→「ボード」
→「ボードマネージャ」

今回は以下のボードを検索して、
インストールしてください。

- esp32

ここは、Arduino IDEを使って開発するときには書き込み先のマイコンを設定する作業です。

1. Arduino IDEの初期設定

- ライブラリのインストール

使用するマイコンやセンサ等のライブラリをインストールします。

「ツール」 - 「ライブラリを管理」

ライブラリマネージャから必要なライブラリを検索してインストールします。

ライブラリとは、Arduino IDEの機能を拡張するための事前に書かれたコードや関数の集まりのことです。

1. Arduino IDEの初期設定

- ライブラリのインストール

今回は、以下2つのライブラリをインストールしてください。

- ・M5StickCPlus
- ・Ambient ESP32 ESP8266 lib (Ambient を使用する場合)

同時にインストールするライブラリが出てきますので、すべてインストールするようにしてください。

～目次～

1. Arduino IDEの初期設定

- 環境設定
- ボードの追加
- ライブラリの追加

2. プログラムの概要

3. マイコンへの書き込み方法

4. 今回のプログラムの内容について

2. プログラムの概要

- プログラムのルール

①コードはすべて半角で入力する。

※全角入力はエラーになります。

特に、全角スペースに要注意！！

②//○○○○

スラッシュの後の○○○○はプログラム自体に影響がない。

このスラッシュの記号を使うことで、後で自分が見たときや他者が見たときに、ここのプログラムがどのような意味なのかを、メモとして書いて残しておくことができます。

2. プログラムの概要

- プログラムの大きな流れ

すべてのプログラムは、

```
#include
```

で前提条件を入力し、

```
void setup() { }
```

で、Wi-Fi等の初期設定を行い、

```
void loop() { }
```

で、loop内のプログラムを永遠に繰り返します。

2. プログラムの概要

- プログラムの大きな流れ

```
#include          //前提条件をの入力
                  //今回はこのマイコン(このセンサ、Wi-Fi、Ambient)
                  //を使います など

void setup() {    //ここのプログラムは最初の1回だけ読まれます。
                  //今回使うWi-Fiの名前やPW など
}

void loop() {     //ここのコードが繰り返し読まれます。
                  //〇分おきにデータを取得する など
}
```

～目次～

1. Arduino IDEの初期設定

- 環境設定
- ボードの追加
- ライブラリの追加

2. プログラムの概要

3. マイコンへの書き込み方法

4. 今回のプログラムの内容について

3. マイコンへの書き込み方法

①ボードを選択します。

「ツール」→「ボード」 →「ESP32 Arduino」→「M5Stick-C」

※M5Stick-Cはかなり下の方にあります。

②マイコンが接続されているシリアルポートを選択します。

「ツール」 - 「シリアルポート」 - 「COM3」

※ポート番号(COM3)は適当に変わります。

③メニュー付近にある右矢印マークで書き込みます。

④画面下半分に「ボードへの書き込みが完了しました」と出たら、マイコンを取り外して大丈夫です。

～目次～

1. Arduino IDEの初期設定

- ・環境設定
- ・ボードの追加
- ・ライブラリの追加

2. プログラムの概要

3. マイコンへの書き込み方法

4. 今回のプログラムの内容について

プログラム

【変更する箇所】

1. 計測時刻
2. Wi-FiのSSID(ネットワーク名)とPW
3. AmbientのチャンネルIDとライトキー
4. Ambientのデータ番号
5. LINEトークン
6. アラートの通知設定

※ここだけ頭を使って考えるところです。

※LINEへの通知コメントに「°C」や「%」などの記号は使えません。

プログラム③ LINE通知機能を付ける(続き)

【変更する箇所】

6. アラートの通知設定

```
if ( tmp >= 35){ //もし気温35°C以上だったら
  alert_tmp = alert_tmp + tmp; //積算気温に気温を足す
} else { //35°C未満だったら
  count_tmp = count_tmp + 1; //気温カウントに1足す
}
if ( alert_tmp >= 70 ){ //もし積算気温が70を超えたら
  send("ハウス温度35度以上"); //LINEに「ハウス気温35度以上」と送る
  alert_tmp = 0; //そして積算気温をゼロに
  count_tmp = 0; //気温カウントもゼロにしてリセット
}
if ( count_tmp == 2 ){ //気温カウントが2になったら
  alert_tmp = 0; //積算気温をゼロにしてリセット
  count_tmp = 0; //気温カウントもリセットする
}
```

※マイコンは再起動など一度電源を切っても、書き込んだプログラムは忘れませんが、積算気温はリセットされます。

4. プログラムの内容について

次ページよりプログラムとなっています。
プログラムはそのままコピーしてお使いいただけますが、
前々ページの6つの**変更箇所**のみ変更をお願いします。


```
// 【機能ごとに色分け】
// M5StickCPlus に環境センサユニット (ENV III) を接続し、気温・湿度を測定する。
// 測定したデータを multiWiFi で Ambient に定刻にデータを送信する。
// 設定に従い、アラートを LINE に送信する。
// 必要ライブラリ：M5StickCPlus、M5UNIT_ENV
// 変更箇所：①計測時間、②Wi-Fi、③Ambient、④アラート
```

```
#include <M5StickCPlus.h>
#include <M5_ENV.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <Ambient.h>
#include <ssl_client.h>
#include <WiFiClientSecure.h>
#include <HTTPClient.h>
#include <time.h>
```

```
WiFiClient client;
WiFiMulti wifiMulti;
```

```
Ambient ambient;
```

```
unsigned int channelId = 000000 ;// Ambient のチャンネル ID
const char* writeKey = "00000000";// Ambient のライトキー
```

```
SHT3X sht30;
QMP6988 qmp6988;
```

```
float tmp = 0.0; //気温
float hum = 0.0; //湿度
int count_tmp = 0; //気温カウント
int count_hum = 0; //湿度カウント
float alert_tmp = 0; //積算気温
float alert_hum = 0; //積算湿度
```

```
// 日本時間を設定するための定義(9 時間ずれを秒数に直している)
```

```
#define JST      3600* 9
```

```
void send(String message) {
    const char* host = "notify-api.line.me";
    const char* token = "○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○"; //LINE
    のトークン
    WiFiClientSecure client;
    Serial.println("Try");
    //Line の API サーバに接続
    client.setInsecure();
    if (!client.connect(host, 443)) {
        Serial.println("Connection failed");
        return;
    }
    Serial.println("Connected");
    //リクエストを送信
    String query = String("message=") + message;
    String request = String("") +
        "POST /api/notify HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "Authorization: Bearer " + token + "\r\n" +
        "Content-Length: " + String(query.length()) + " \r\n" +
        "Content-Type: application/x-www-form-urlencoded\r\n\r\n" +
        query + "\r\n";
    client.print(request);

    //受信終了まで待つ
    while (client.connected()) {
        String line = client.readStringUntil('\n');
        Serial.println(line);
        if (line == "\r") {
            break;
        }
    }
}

String line = client.readStringUntil('\n');
```

```

Serial.println(line);
}

void setup() {
  M5.begin();
  M5.Axp.ScreenBreath(10); // 画面の輝度を少し下げる
  M5.Lcd.setRotation(3); // 左を上にする
  M5.Lcd.setTextSize(2); // 文字サイズを2にする
  M5.Lcd.fillScreen(BLACK); // 背景を黒にする

  wifiMulti.addAP("〇〇〇〇〇〇", "〇〇〇〇〇〇"); //Wi-Fi の SSID (ネットワーク名) と
  PW

  Serial.begin(115200);
  Serial.println("Connecting Wifi...");
  if(wifiMulti.run() == WL_CONNECTED) {
    Serial.println(".");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    M5.Lcd.setCursor(0, 70, 1);
    M5.Lcd.println("WiFi connected");
    M5.Lcd.println(WiFi.localIP());
  }

  ambient.begin(channelId, writeKey, &client); // チャンネル ID とライトキーを指定して
  Ambient を初期化

  //configTime( 時差 , サマータイム , 日本の NTP サーバー(複数可能) )
  configTime( JST, 0, "ntp.nict.jp", "ntp.jst.mfeed.ad.jp");

  Wire.begin();

  qmp6988.init();

```

```
}
```

```
void loop() {
```

```
    // 時間を格納する変数を設定
```

```
    time_t t;
```

```
    struct tm *tm;
```

```
    t = time(NULL);
```

```
    tm = localtime(&t);
```

```
    // tm->tm_hour, tm->tm_min, tm->tm_sec (左から時、分、秒)
```

```
    //if ((tm->tm_min == 0 && tm->tm_sec ==0) || (tm->tm_min == 10 && tm->tm_sec ==0) || (tm->tm_min == 20 && tm->tm_sec ==0) || (tm->tm_min == 30 && tm->tm_sec ==0) || (tm->tm_min == 40 && tm->tm_sec ==0) || (tm->tm_min == 50 && tm->tm_sec ==0)){
```

```
    if ( (tm->tm_sec == 0) || (tm->tm_sec == 20) || (tm->tm_sec == 40) ){
```

```
        if(sht30.get()==0){
```

```
            tmp = sht30.cTemp;
```

```
            hum = sht30.humidity;
```

```
        }else{
```

```
            tmp=0,hum=0;
```

```
        }
```

```
    //アラートの通知設定
```

```
    if ( tmp >= 35){
```

```
        //もし気温 35°C以上だったら
```

```
        alert_tmp = alert_tmp + tmp;
```

```
        //積算気温に気温を足す
```

```
    } else {
```

```
        //35°C未満だったら
```

```
        count_tmp = count_tmp + 1;
```

```
        //気温カウントに 1 足す
```

```
    }
```

```
    if ( alert_tmp >= 70 ){
```

```
        //もし積算気温が 70 を超えたら
```

```
        send("ハウス温度 35 度以上");
```

```
        //LINE に「ハウス気温 35 度以上」と送る
```

```
        alert_tmp = 0;
```

```
        //そして積算気温をゼロに
```

```
        count_tmp = 0;
```

```
        //気温カウントもゼロにしてリセット
```

```
    }
```

```
    if ( count_tmp == 2 ){
```

```
        //気温カウントが 2 になったら
```

```

    alert_tmp = 0;           //積算気温をゼロにしてリセット
    count_tmp = 0;         //気温カウントもリセットする
}

if ( hum >= 95){           //もし湿度 (humid) が 95%以上だったら
    alert_hum = alert_hum + hum; //積算湿度に湿度を足す
} else {                   //95%未満だったら
    count_hum = count_hum + 1; //湿度カウントに 1 足す
}

if ( alert_hum >= 4560 ){ //積算湿度が 4560 (95*6*8) (6 は 10 分おきの測定
//で、1 時間に 6 回) を超えたら
    send("ハウス湿度 95 パーセント 8 時間");//LINE に「ハウス湿度 95 パーセント 8 時
//間」と送る
    alert_hum = 0;         //そして積算湿度をゼロに
    count_hum = 0;        //湿度カウントもゼロにしてリセット
}

if (int(count_hum) == 5 ){ //もし湿度カウントが 5 になったら
    alert_hum = 0;         //積算湿度をゼロにしてリセット
    count_hum = 0;        //湿度カウントもリセットする
}

M5.Lcd.setCursor(0, 5, 1);
M5.Lcd.printf("temp: %4.1f°C¥r¥n", tmp);
M5.Lcd.printf("humid:%4.1f%%¥r¥n", hum);

ambient.set(1, tmp);      //気温
ambient.set(2, hum);      //湿度
ambient.send();

delay( 10 * 1000 );
}
}

```

自分でプログラムを作成される方へ

- ネットに多くの情報があるので、検索してみてください。
そこにある情報を組み合わせて、自分用のプログラムを作成します。
- 作成手順は、①プログラムを書く、②プログラムの検証、③プログラムを書き込む、④マイコンが正しく動作するか確認 になります。
なかなか成功しない場合は、この①～④を繰り返します。
- ②プログラムの検証は、メニュー付近にあるチェックマークです。
- プログラムの作成の際は、こまめな保存をおすすめします。
完成したプログラムは保存し、コピーしたファイルに新たな機能を追加するなど変更を行います。そうすることで、確実に通るプログラムを残しておくことができます。
- プログラムのフォルダは、ライブラリのフォルダと同じ場所にある必要があります。

その他

- メンテナンスについて

- ・コード接続部のほこりは定期的になぞってください。
- ・水などにぬらさないでください。

- 不具合について

○データがAmbientに飛んできていない
→Wi-Fiとつながっていますか？

○電源が入らない(画面が真っ暗)
→コードの接続はしっかりできていますか？(電気は来ていますか？)

○温度測定ができない、Wi-Fiとつながらない
→再起動してみてください。

最後に

本マニュアルは、安価な測定装置を自作するための機器やプログラムを紹介するもので、作成や利用にあたっては、個人の責任において行ってください。

成果情報「スマートフォン等でハウス内の温度をリアルタイムに確認できる安価な測定装置の作成」参照

問合せ先 農業試験研究センター 企画・スマート農業研究担当
TEL:0952-45-2142