

I2C-PLUS-R1 利用マニュアル

マイクロファン

<http://www.microfan.jp/>

<http://store.shopping.yahoo.co.jp/microfan/>

<https://www.amazon.co.jp/s?merchant=A28NHPRKJDC95B>

2017年7月

Copyright © 2017 MicroFan,
All Rights Reserved.

目次

第 1 章	I2C-PLUS-R1 の紹介	1
1.1	製品概要	1
1.2	Arduino でのスケッチ	2
1.3	マニュアルの記載内容に関して	2
第 2 章	部品一覧	4
2.1	部品表	4
2.2	別売品	4
2.2.1	OLED ディスプレイ	6
2.2.2	BME280 環境センサー	7
2.2.3	MPU6050 加速度・ジャイロセンサー	8
2.2.4	DS3231 リアルタイムクロック	8
第 3 章	I2C-PLUS-R1 の作成手順	10
3.1	抵抗	10
3.2	FET	10
3.3	ピンソケット SV3, SV4	11
3.4	ピンヘッダー	11
3.5	ピンソケット SV5	11
3.6	I2C-PLUS-R1 の作成例	11
第 4 章	環境整備とサンプルスケッチ	12
4.1	Arduino UNO ボードとの接続	12
4.2	ライブラリのインストール	12
4.2.1	OLED ディスプレイのライブラリ	12
4.2.2	BME280 のライブラリ	14
4.2.3	リアルタイムクロックのライブラリ	15
4.3	ライブラリの利用例	16
4.3.1	OLED ディスプレイ	16
4.3.2	BME280 環境センサー	18
4.3.3	リアルタイムクロック	19

第 5 章	資料	21
5.1	I2C-PLUS-R1 の回路図	21
5.2	Arduino ボードのピン接続	21
5.3	I2C 用コネクタ	22
5.3.1	CN1: 5V デバイス用	22
5.3.2	CN2, CN3: 3.3V デバイス用	23
5.4	I2C モジュールの搭載コネクタ	23
5.4.1	CN4: BME280	23
5.4.2	CN6: DS3231	24
5.4.3	CN7:MPU6050	24
第 6 章	購入および問い合わせ先	26
6.1	ご協力をお願い	26
6.2	販売：ネットショップ	26
6.3	製品情報	26
6.4	問い合わせ先	26
6.5	所在地	27

目次

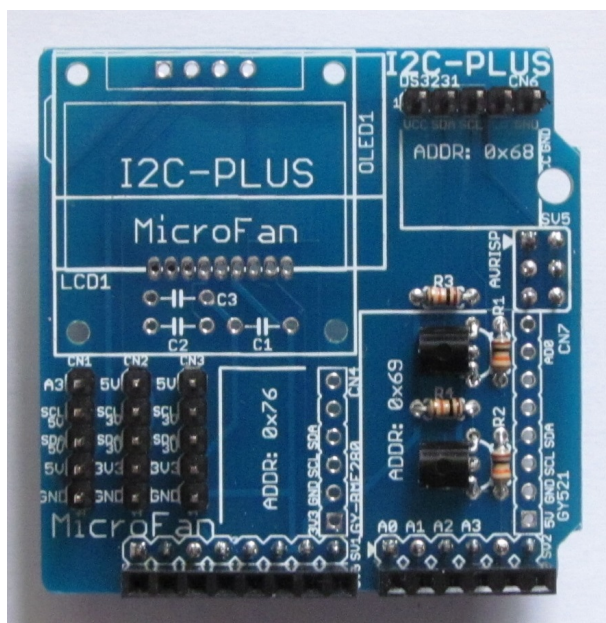
1.1	I2C-PLUS-R1 の Arduino UNO への接続例	2
2.1	部品 : 写真	5
2.2	I2C-PLUS-R1 とオプション部品の利用例	6
2.3	OLED ディスプレイ (表面)	6
2.4	OLED ディスプレイ (裏面)	7
2.5	環境センサー	7
2.6	加速度・ジャイロセンサー	8
2.7	DS3231 モジュール	9
3.1	I2C-PLUS-R1 の作成例	11
4.1	OLED ディスプレイと BME280 モジュールの搭載例	13
4.2	ライブラリマネージャを利用した U8g2 ライブラリの導入	13
4.3	OLED ディスプレイ (SSD1306) 用のコンストラクタ	14
4.4	OLED ディスプレイ (SSD1306) 用の文字表示専用コンストラクタ	14
4.5	ライブラリマネージャを利用した BME280 ライブラリの導入	15
4.6	OLED ディスプレイのサンプルスケッチ	16
4.7	OLED ディスプレイの文字専用出力	17
4.8	BME280 環境センサーのサンプルスケッチ	18
4.9	リアルタイムクロックのサンプルスケッチ	19
4.10	リアルタイムクロックのサンプルスケッチ	20
5.1	I2C-PLUS-R1 の回路図	21

表目次

2.1	部品表	4
5.1	Arduino ボードの電源ピン	22
5.2	Arduino ボードのアナログピン	22
5.3	CN1 ピン配置	22
5.4	CN2 ピン配置	23
5.5	CN3 ピン配置	23
5.6	CN4 ピン配置	24
5.7	CN6 ピン配置	24
5.8	CN7 ピン配置	25

第 1 章

I2C-PLUS-R1 の紹介



1.1 製品概要

I2C-PLUS-R1 は、信号線が 3.3V の I2C デバイスを Arduino UNO で手軽に利用できるようにするミニシールドです。通信線として I2C を利用するセンサーや入出力デバイスが増えていますが、信号線が 3.3V となっているものが多く、Arduino UNO の信号は 5V となっているために、そのままでは手軽に接続することができませんでした。

I2C-PLUS-R1 の特徴を以下に示します。

- 5V の I2C 信号を 3.3V に変換し、3.3V に対応した I2C デバイスを接続できます。
- 3.3V の信号線の I2C デバイスを接続できる端子を 2 個装備しています。
- 5V の信号線の I2C デバイスを接続できる端子を 1 個装備しています。

- 環境センサーモジュール GY-BME280 を基板上に搭載できます。
- 加速度・ジャイロセンサーモジュール GY521 を基板上に搭載できます。
- リアルタイムクロックモジュール DS3231 を基板上に搭載できます。
- 128x64 ピクセルの OLED ディスプレイもしくは、8 文字 x2 行の文字 LCD ディスプレイを基板上に搭載できます。
- SCL, SDL として使用されている A4, A5 を除き、デジタルピン D0-D13, アナログピン A0-A3 は未使用なので自由に使用することができます。
- 電子工作で広く利用されている Arduino のミニシールドとして開発されており、手軽に利用できる Arduino の IDE を利用して独自のスケッチを行えます。

I2C-PLUS-R1 は、Arduino MEGA など、Arduino UNO 以外の Arduino ボードでは利用できませんのでご注意ください。

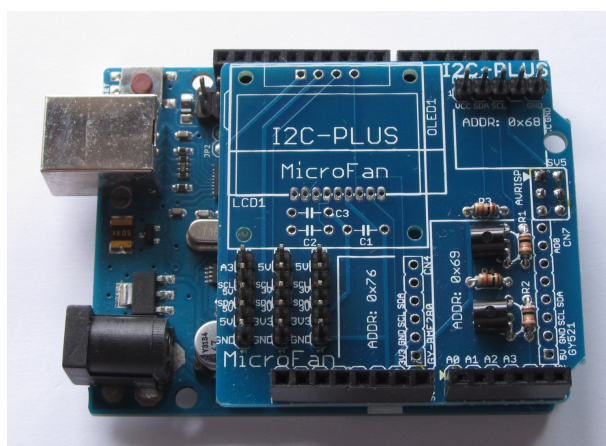


図 1.1 I2C-PLUS-R1 の Arduino UNO への接続例

1.2 Arduino でのスケッチ

本製品は、Arduino のシールドとして開発されています。このためプログラミングは、多くの利用者が親しんでいる Arduino の IDE を利用して手軽に行うことができます。

1.3 マニュアルの記載内容に関して

本文書の一部もしくは全部を無断で複写、複製、配布することは、法律で認められた場合を除き、著作権の侵害となります。

本文書に記載されている製品名などは、一般的にそれぞれの権利者の登録商標または商標です。

お伝えする内容と本質的な問題がないと判断した場合には、本マニュアルには、旧バージョンの製品の写真や他製品の写真などがそのまま使用されている場合がありますのでご

承知おきください。

本文書は最善の注意を払って作成されていますが、本書に記載されている内容の誤り、本書に記載されている内容に基づく作業、運用などにおいて、いかなる損害が生じても、弊社および著者をはじめとする本文書作成関連者は、一切の責任を負いませんのであらかじめご了承ください。

第2章

部品一覧

2.1 部品表

I2C-PLUS-R1 キットの部品一覧を表 2.1, 図 2.1 に示します。部品が不足、破損している場合には、キットを組み立てる前にマイクロファンにお問い合わせください。

表 2.1 部品表

部品	シンボル	規格等	個数
プリント基板	I2C-PLUS-R1		1
FET	Q1, Q2	2N7000	2
抵抗	R1-R4	10K Ω	4
ピンソケット	SV3	8 ピン	1
ピンソケット	SV4	6 ピン	1
ピンソケット	SV5	2x3 ピン	1
ピンヘッダー	SV1	10 ピン, 40 ピンから切り出し	1
ピンヘッダー	SV2	8 ピン, 40 ピンから切り出し	1
ピンヘッダー	CN1,CN2,CN3	5 ピン, 40 ピンから切り出し	4
OLED ディスプレイ	OLED1	128 × 64 ピクセル I2C	別売り
環境センサー	CN4	GY-BME280	別売り
リアルタイムクロック	CN6	DS3231	別売り
加速度・ジャイロセンサー	CN7	MPU6050	別売り

2.2 別売品

本製品は Arduino UNO のシールドとして開発されています。このため、本製品を利用するためには下記製品が別途必要です。

- Arduino UNO あるいはそれらの互換品 (必須)

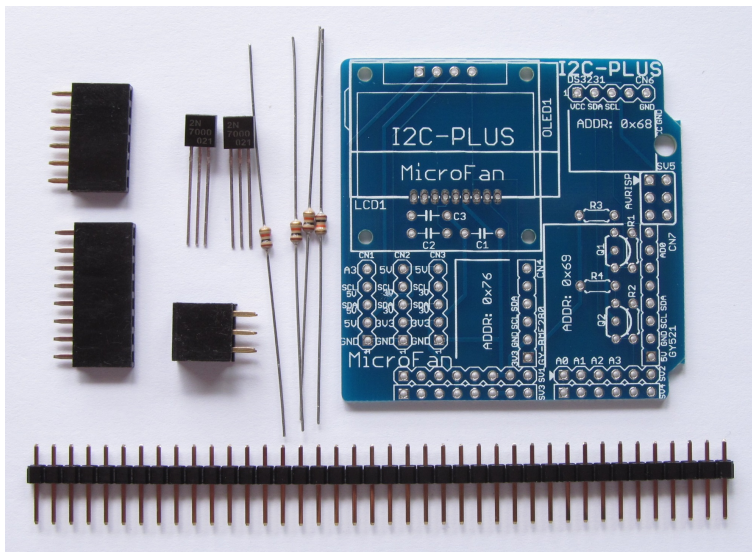


図 2.1 部品：写真

I2C-PLUS-R1 は、SCL, SDL 信号をアナログピンソケットの A4, A5 から取得しており、SCL, SDA がデジタルピンと同じ列のソケットに出力されていない R2 以前の製品でも使用できます。一方、Arduino MEGA など、Arduino UNO 以外の Arduino ボードでは利用できませんのでご注意ください。

I2C-PLUS-R1 には、以下の部品を取り付けて利用できるように設計されていますが、それらの部品そのものはキットには含まれていませんので、別途ご用意ください。

- OLED ディスプレイ
<https://store.shopping.yahoo.co.jp/microfan/oled096-128x64-i2c-blue.html>
<https://www.amazon.co.jp/dp/B06Y4TKL1F/>
- DS3231 リアルタイムクロック
<https://store.shopping.yahoo.co.jp/microfan/ds3231-rtc-module.html>
- 環境センサー GY-BME280
<https://store.shopping.yahoo.co.jp/microfan/gy-bme280.html>
- 加速度・ジャイロセンサー GY521
<https://store.shopping.yahoo.co.jp/microfan/gy-521.html>

これらのオプションを I2C-PLUS-R1 に搭載した例を図 2.2 に示します。

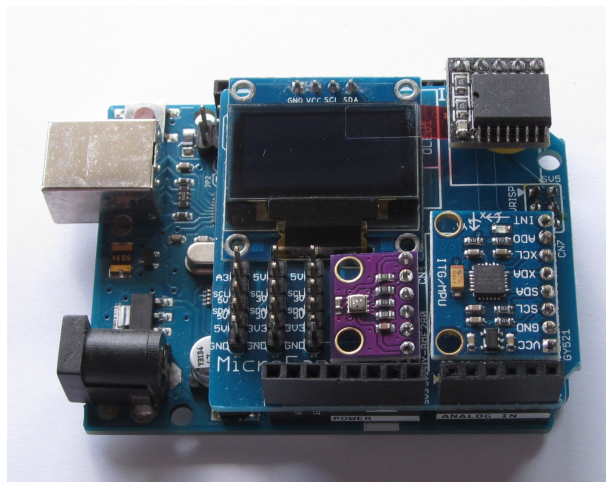


図 2.2 I2C-PLUS-R1 とオプション部品の利用例

2.2.1 OLED ディスプレイ

I2C-PLUS-R1 に搭載できる OLED ディスプレイモジュール*1の表面を図 2.3、裏面を図 2.4 に示します。

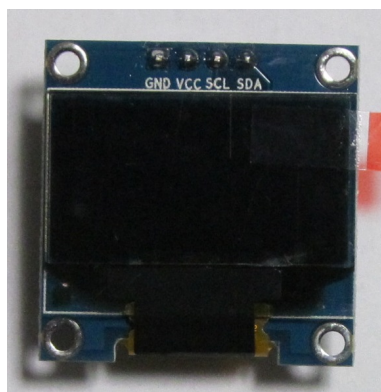


図 2.3 OLED ディスプレイ（表面）

OLED は発光ダイオードの一種で、液晶とは異なり自身で発光するためバックライトを必要としません。OLED を使用すると薄型で消費電力の低い表示装置を作成できるという特徴があります。

I2C-PLUS-R1 に搭載できる OLED ディスプレイは、コントローラーに SSD1306 を使用しており、コンパクトながら 128x64 ピクセルの解像度を持っています。Arduino UNO との接続は 2 線式の I2C インターフェースを利用します。この OLED ディスプレ

*1 製品のロットにより詳細が異なることがあります。

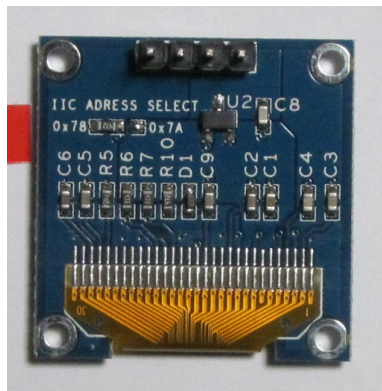


図 2.4 OLED ディスプレイ (裏面)

イは、モノクログラフィックスディスプレイ用の表示ライブラリである U8g2lib で操作することができます。

モジュール内の OLED ディスプレイの電源や信号線は 3.3V を利用するようになっています。モジュールの基板には、3.3V の電圧レギュレータが組み込まれているため、外部からの電源は 5V を供給して使用できます。

2.2.2 BME280 環境センサー

I2C-PLUS-R1 は、BME280 を搭載した環境センサーモジュール (図 2.5) を装着できます。BME280 は温度、湿度、気圧センサーが組み込まれた IC です。

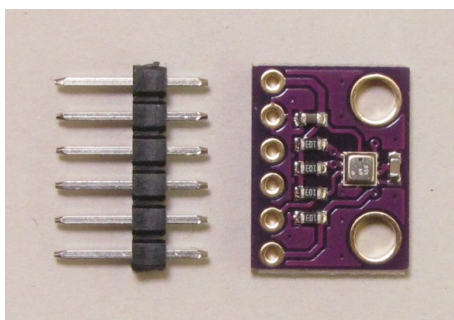


図 2.5 環境センサー

BME280 の概要は、下記の WEB ページに掲載されています。

https://www.bosch-sensortec.com/bst/products/all_products/bme280

また、この WEB ページから BME280 のデータシートをダウンロードできます。

2.2.3 MPU6050 加速度・ジャイロセンサー

I2C-PLUS-R1 は、MPU6050 を搭載した加速度・ジャイロセンサーモジュール (図 2.6) を装着できます。MPU6050 は加速度センサーとジャイロセンサーが組み込まれた IC で、XYZ の 3 方向の加速度と、3 方向の回転角を調べることができます。この機能により、特別な操作をする必要なく I2C-PLUS-R1 の傾きや回転などの動きを取得することができます。また、付加機能として、温度センサーも内蔵しています。

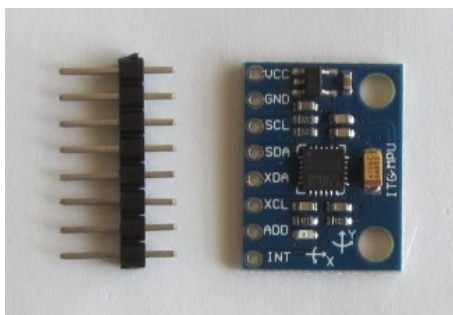


図 2.6 加速度・ジャイロセンサー

MPU6050 の概要は、下記の WEB ページに掲載されています。

<http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
また、この WEB ページから MPU6050 のデータシートをダウンロードできます。

2.2.4 DS3231 リアルタイムクロック

I2C-PLUS-R1 は、DS3231 にバックアップ電池を付加したモジュール (図 2.7) を搭載することができます。DS3231 は、秒、分、時間、曜日、日付、月、および年の情報を管理する高精度のリアルタイムクロック IC です。

I2C-PLUS-R1 に DS3231 モジュールを搭載すると、Arduino 本体の電源が切れている間も正確に動き続ける時計機能を利用できるようになります。また、付加機能として、温度センサーも内蔵しています。

DS3231 の概要は、下記の WEB ページに掲載されています。

<https://www.maximintegrated.com/jp/products/digital/real-time-clocks/DS3231.html>

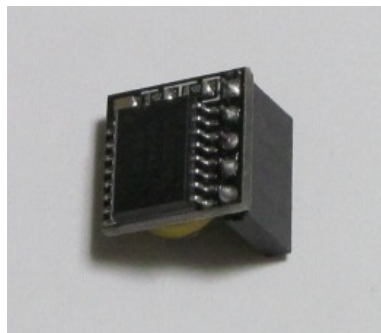


図 2.7 DS3231 モジュール

第3章

I2C-PLUS-R1 の作成手順

I2C-PLUS-R1 キットの標準的な作成手順を以下に示します。基本的には、一般的な電子工作の手順と同様で、背の低い部品から実装していきます。

- 抵抗の実装
- FET の実装
- ピンソケットの実装 SV3, SV4
- ピンヘッダーの実装
- ピンソケットの実装 SV5

3.1 抵抗

抵抗 R1-R4 をプリント基板上の所定の位置に取り付けはんだ付けします。はんだ付けは、片方の足をはんだ付けし、抵抗の取り付け姿勢などを必要に応じて修正してからもう片方の足をはんだ付けすると、抵抗の配置をきれいに整えて取り付けることができます。はんだ付け後に、余った足をニッパーなどで切り取ります。

3.2 FET

FET Q1, Q2 を所定の位置に取り付けはんだ付けします。

FET Q1, Q2 は、取り付け穴の間隔に合うように、本体に大きな力が加わらないように注意してラジオペンチなどで事前に足の幅を広げて基板に取り付けてください。トランジスタ本体の切欠き部分とシルクの切りかけ部分の方向を合わせて取り付けてください。

なお、Q1, Q2 は、必要に応じてその上部にかぶさるように MPU6050 モジュールを装着するため、図 3.1 に示すように、FET のプラスチックモールドの切欠き面を下にして、基板に張り付くように取り付けてください。

3.3 ピンソケット SV3, SV4

ピンソケット SV3, SV4 をプリント基板上の所定の位置に取り付けはんだ付けします。ピンソケットの端子のはんだ付けは、まず、それぞれのピンソケットの両端の端子をはんだ付けし、ピンソケットの取り付け姿勢などを必要に応じて修正してから残りの端子をはんだ付けすると、ピンソケットの配置をきれいに整えて取り付けることができます。

3.4 ピンヘッダー

ピンヘッダーから、ニッパーなどで必要な長さのピンヘッダーを切り出し、プリント基板の CN1, CN2, CN3, CN6 に取り付けはんだ付けします。また、プリント基板の SV1, SV2 の裏面に取り付けはんだ付けします。ピンヘッダーの端子のはんだ付けは、まず、それぞれのピンヘッダーの両端の端子をはんだ付けし、ピンヘッダーの取り付け姿勢などを必要に応じて修正してから残りの端子をはんだ付けすると、ピンヘッダーの配置をきれいに整えて取り付けることができます。

3.5 ピンソケット SV5

ピンソケット SV5 をプリント基板上の所定の位置の裏側に取り付けはんだ付けします。

3.6 I2C-PLUS-R1 の作成例

I2C-PLUS-R1 の作成例を図 3.1 に示します。お疲れ様でした。

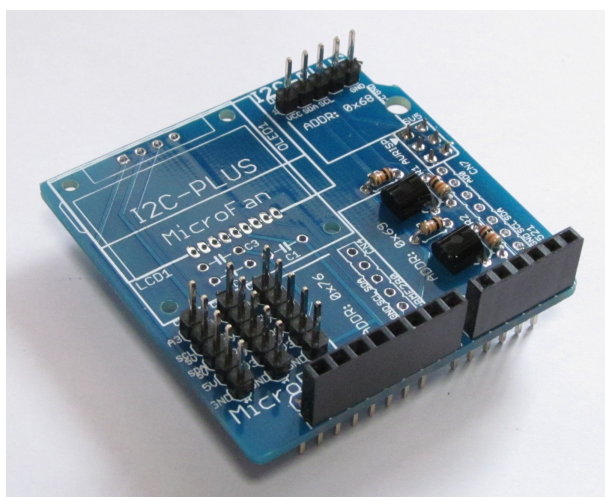


図 3.1 I2C-PLUS-R1 の作成例

第 4 章

環境整備とサンプルスケッチ

4.1 Arduino UNO ボードとの接続

I2C-PLUS-R1 の背面に延びているピンヘッダーとピンソケットを利用して、Arduino UNO ボードに接続します。

4.2 ライブラリのインストール

I2C-PLUS-R1 を利用するためには、必要に応じて Arduino IDE に以下のデバイス用のライブラリをインストールする必要があります。

- OLED ディスプレイ
- DS3231 リアルタイムクロック
- 環境センサー GY-BME280
- 加速度・ジャイロセンサー GY521

Arduino IDE へのライブラリの一般的なインストール法は、下記の WEB ページに説明されています。

<https://www.arduino.cc/en/Guide/Libraries>

4.2.1 OLED ディスプレイのライブラリ

I2C-PLUS-R1 の OLED1 に OLED ディスプレイを搭載した例を図 4.1 に示します。

OLED ディスプレイを利用するためのライブラリとして、U8g2 ライブラリを使用する例を示します。U8g2 ライブラリは、Arduino IDE のライブラリマネージャを利用してインストールすることができます。ライブラリマネージャの検索フィルタに [U8g2] を入力して絞り込むと、図 4.2 のように表示されます。

ライブラリマネージャのダイアログ上でインストールするライブラリの欄をクリックすると、インストールボタンが表示されるので、最新バージョンを選択して、ライブラリをインストールします。

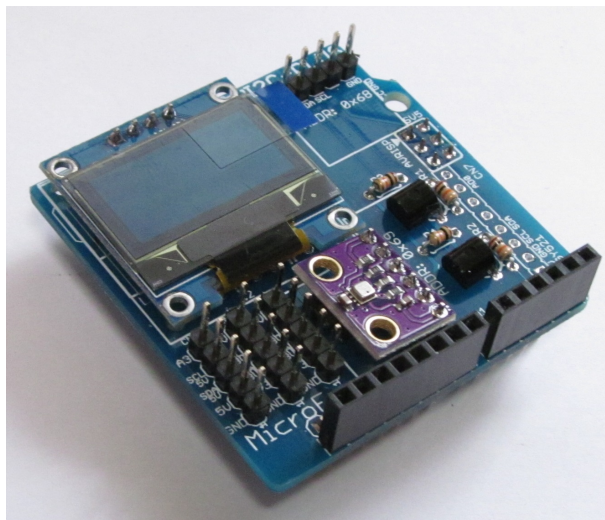


図 4.1 OLED ディスプレイと BME280 モジュールの搭載例

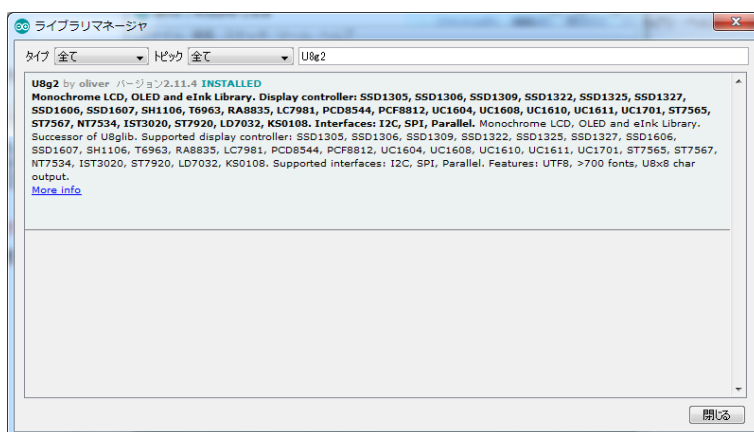


図 4.2 ライブラリマネージャを利用した U8g2 ライブラリの導入

このライブラリは、以下の URL で取得することもできます。

<https://github.com/olikraus/u8g2>

また、マニュアルは、以下の URL で参照することができます。

<https://github.com/olikraus/u8g2/wiki>

ライブラリのインストール後、Arduino IDE メニューから [ファイル] ⇒ [スケッチの例] を選択すると、リストに U8g2 フォルダが追加されているのが確認できます。U8g2 フォルダの中を確認するといくつかのサンプルスケッチがあり、選択して実行することができます。

デモ用のスケッチの中には、[full_buffer]、[page_buffer] の分類があります。[full_buffer] に分類されているデモは、RAM メモリが十分にある MEGA などを対象にしたもので、[page_buffer] に分類されているデモは、RAM メモリが少ない UNO を対象にしたもの

です。

例として [u8g2] ⇒ [page.buffer] ⇒ [GraphicsTest] を選択します。このスケッチは、デモ用のグラフィック表示を行うものです。

このサンプルスケッチをコンパイル・実行するためには、コメントアウトされたりリストの中から適切な u8g2 コンストラクタを選択するか、自分自身で追加する必要があります。ここで利用する OLED ディスプレイは、コントローラとして SSD1306 を使用しており、I2C インターフェースで接続されているので、サンプルスケッチの 60 行あたりに、以下のコンストラクタを追加してください。

```
U8G2_SSD1306_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE) ;
```

図 4.3 OLED ディスプレイ (SSD1306) 用のコンストラクタ

OLED ディスプレイに文字しか表示しない場合には、軽量な U8x8 ライブラリを使用することができます。U8x8lib は U8g2lib に組み込まれていますが、両方で利用できるコンストラクタやメソッドは異なっていますので、別のライブラリと考えて利用する必要があります。

U8x8 ライブラリを利用する場合には、以下のコンストラクタを使用します。

```
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(U8X8_PIN_NONE) ;
```

図 4.4 OLED ディスプレイ (SSD1306) 用の文字表示専用コンストラクタ

このコンストラクタの追加により、文字表示専用のライブラリを使用したスケッチのコンパイル、実行ができるようになります。

4.2.2 BME280 のライブラリ

I2C-PLUS-R1 の CN4 に BME280 モジュールを搭載した例は前節の図 4.1 を参照してください。

BME280 用のライブラリは、Arduino IDE のライブラリマネージャを利用してインストールすることができます。ライブラリマネージャの検索フィルタに [BME280] を入力して絞り込むと、BME280 用のライブラリは、この文書の作成時点では 3 個ほど列挙されます。

ここでは、[BME280 by Tyler Glenn] とタイトルが付けられているライブラリを選択してインストールする例を示します。

ライブラリマネージャのダイアログ上でインストールするライブラリの欄をクリックすると、インストールボタンが表示されるので、最新バージョンを選択して、ライブラリをインストールします。

このライブラリは、以下の URL で取得することもできます。

<https://github.com/finitespace/BME280>

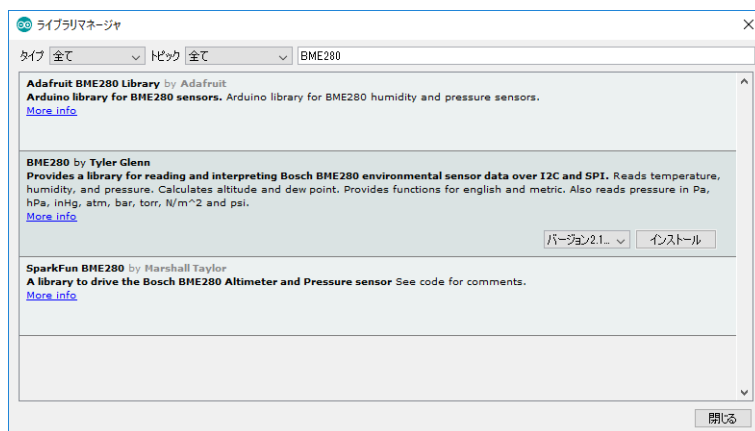


図 4.5 ライブラリマネージャを利用した BME280 ライブラリの導入

ライブラリのインストール後、Arduino IDE メニューから [ファイル] ⇒ [スケッチの例] を選択すると、リストに BME280 フォルダが追加されているのが確認できます。BME280 フォルダの中を確認するといくつかのサンプルスケッチがあり、選択して実行することができます。

I2C-PLUS-R1 では、BME280 モジュールは I2C で接続されているので、サンプルスケッチのうち [BME_280_I2C_Test] を選択します。このサンプルスケッチは、温度、湿度、気圧およびそれらから算出できる値を 0.5 秒ごとに取得してシリアルモニタに出力します。

サンプルスケッチはこのままコンパイル、書き込んで実行させることができますが、温度が華氏であったり標高がフィートだったりするので、サンプルスケッチの 46 行の `boolmetric = false;` の `false` を `true` に変更します。この変更により、温度は摂氏で、標高がメートルで表示されるようになります。

シリアルモニタの出力が文字化けする場合には、`setup()` 内で指定したボーレートと、シリアルモニタの右下で選択できるボーレートが合致していることを確認してください。

4.2.3 リアルタイムクロックのライブラリ

DS3231 を利用するためには下記の URL から ZIP 形式のライブラリをダウンロードします。

<https://github.com/Makuna/Rtc>

入手した `RtcDS3231` のライブラリをライブラリマネージャを利用して Arduino IDE にインストールします。

4.3 ライブラリの利用例

I2C-PLUS-R1 の利用例として、以下のようなサンプルスケッチを示します。

- OLED ディスプレイ
- BME280 環境センサー
- DS3231 リアルタイムクロック

4.3.1 OLED ディスプレイ

OLED ディスプレイを使用する際には、U8g2lib ライブラリを使用します。U8g2lib ライブラリの使用法は、以下の WEB ページに紹介されています。

<https://github.com/olikraus/u8g2/wiki/u8g2reference>

U8g2lib ライブラリを使用したサンプルスケッチを図 4.6 に示します。U8g2lib ライブラリを使用する際には、まず、ヘッダーファイル [U8g2lib.h] をインクルードします。

```
#include <U8g2lib.h>

U8G2_SSD1306_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE) ;

void setup(void)
{
  u8g2.begin() ;
  u8g2.setFont(u8g2_font_ncenB10_tr) ;
}

void loop(void)
{
  u8g2.firstPage() ;
  do {
    u8g2.drawStr(0, 30, "Hello, MicroFan!") ;
    u8g2.setCursor(50, 50) ;
    u8g2.print(millis()/100) ;
  } while( u8g2.nextPage() ) ;

  delay(100) ;
}
```

図 4.6 OLED ディスプレイのサンプルスケッチ

OLED ディスプレイに文字しか表示しない場合には、メモリの使用量が少ない軽量の U8x8 ライブラリを使用することができます。U8x8 ライブラリの詳細な使用法は、以下

のWEB ページに紹介されています。

<https://github.com/olikraus/u8g2/wiki/u8x8reference>

U8x8 ライブラリを使用したサンプルスケッチを図 4.7 に示します。U8x8 ライブラリを使用する際には、まず、ヘッダーファイル [U8x8lib.h] をインクルードします。

```
#include <U8x8lib.h>

// SSD1306 をテキストディスプレイとして利用するためのコンストラクタ
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(U8X8_PIN_NONE) ;

void setup(void)
{
    u8x8.begin() ;
    u8x8.setFont(u8x8_font_artossans8_r) ; // フォントの指定
}

void loop(void)
{
    u8x8.clear() ; // 画面をクリアしカーソルを左上に戻す

    u8x8.println("MicroFan") ; // println() は文字列の表示後に改行
    u8x8.println("I2C-PLUS\n") ;
    u8x8.print("OLED") ; // print() は改行しないので文字列を継続表示でき
る
    u8x8.print(" 128x64\n") ; // print() で改行したい場合には\nを指定
    u8x8.println(millis()) ; // 整数値の表示
    u8x8.println(123.45) ; // 実数値の表示

    delay(1000);
}
```

図 4.7 OLED ディスプレイの文字専用出力

4.3.2 BME280 環境センサー

BME280 環境センサーを使用する際には、BME280I2C ライブラリを使用します。

BME280I2C ライブラリを使用したサンプルスケッチを図 4.8 に示します。ライブラリのサンプルスケッチ BME_280_I2C_Test の簡略版です。

```
#include <BME280I2C.h>

BME280I2C bme;
#define PRESSURE_HPA 1 // ヘクトパスカル

void setup()
{
  Serial.begin(115200) ;

  while (!bme.begin()) {
    delay(1000) ;
  }
}

void loop()
{
  float temp ; // 温度
  float hum ; // 湿度
  float pres ; // 気圧

  // 気圧 (hPa)、温度 (摂氏)、湿度を取得
  bme.read(pres, temp, hum, true, PRESSURE_HPA) ;

  Serial.println("---") ;
  Serial.print("Temp: ") ;
  Serial.print(temp) ;
  Serial.println(" C") ;
  Serial.print("Humidity: ") ;
  Serial.print(hum) ;
  Serial.println(" % RH") ;
  Serial.print("Pressure: ") ;
  Serial.print(pres) ;
  Serial.println(" hPa") ;

  delay(1000) ;
}
```

図 4.8 BME280 環境センサーのサンプルスケッチ

4.3.3 リアルタイムクロック

CN1 にリアルタイムクロック DS3231 を接続して日時表示するスケッチを以下に示します。時間をシリアルモニタに出力する例を 4.9 に、OLED に表示する例を 4.10 に示します。

```
#include <RtcDS3231.h>

RtcDS3231 Rtc ;

void setup(void)
{
  Serial.begin(115200) ;

  Rtc.Begin() ;

  // DS3231 の日時を「それなりの値」に設定する場合に、以下の2行をコメントアウト
  // RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__) ;
  // Rtc.SetDateTime(compiled) ;
}

void loop ()
{
  char dbuff[12], tbuff[12] ;

  RtcDateTime dt = Rtc.GetDateTime() ;

  sprintf(dbuff, "%04u/%02u/%02u", dt.Year(), dt.Month(), dt.Day()) ;
  sprintf(tbuff, " %02u:%02u:%02u", dt.Hour(), dt.Minute(), dt.Second()) ;

  Serial.println("---") ;
  Serial.println(dbuff) ;
  Serial.println(tbuff) ;

  delay(1000) ;
}
```

図 4.9 リアルタイムクロックのサンプルスケッチ

リアルタイムクロックモジュールは、最初の状態では適切な日時が設定されておらず、でたらめな日時が表示されるようです。私の例では、2000 年の適当な日時が表示されました。

リアルタイムクロックモジュールに正確な日時を設定するのは少し面倒ですが、おおざっぱな時間を設定するのは簡単にできます。上記のスケッチの setup() 中の 2 行のコメントを外して実行できるようにしてコンパイルし、スケッチを Arduino ボードに書き込んで実行すると、コンパイルを行った日時がリアルタイムクロックに設定されます。スケッチのコンパイルとスケッチの実行では少し時間差があるので、リアルタイムクロックモジュールに 10-20 秒程度遅れた時間を設定することになりますが、それなりの日時を簡


```
#include <U8g2lib.h>
#include <RtcDS3231.h>

U8G2_SSD1306_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE) ;
RtcDS3231 Rtc ;

void setup(void)
{
  u8g2.begin() ;
  u8g2.setColorIndex(1) ; // pixel on
  u8g2.setFont(u8g_font_unifont) ;

  Rtc.Begin() ;

  // DS3231 の日時を「それなりの値」に設定する場合に、以下の2行をコメントアウト
  // RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__) ;
  // Rtc.SetDateTime(compiled) ;
}

void loop ()
{
  char dbuff[12], tbuff[12] ;

  RtcDateTime dt = Rtc.GetDateTime() ;

  sprintf(dbuff, "%04u/%02u/%02u", dt.Year(), dt.Month(), dt.Day()) ;
  sprintf(tbuff, " %02u:%02u:%02u", dt.Hour(), dt.Minute(), dt.Second()) ;

  u8g2.firstPage() ;
  do {
    u8g2.drawStr( 20, 32, dbuff) ;
    u8g2.drawStr( 20, 48, tbuff) ;
  } while( u8g2.nextPage() ) ;

  delay(100) ;
}
```

図 4.10 リアルタイムクロックのサンプルスケッチ

単に設定できます。この方法は、ライブラリのデモプログラムの中に記載されていたものです。

一度リアルタイムクロックに日時を設定すると、リアルタイムクロックは電池でバックアップされているため、Arduino 本体の電源を切っても日時は自動的に更新されます。

第 5 章

資料

5.1 I2C-PLUS-R1 の回路図

I2C-PLUS-R1 の回路図を図 5.1 に示します。

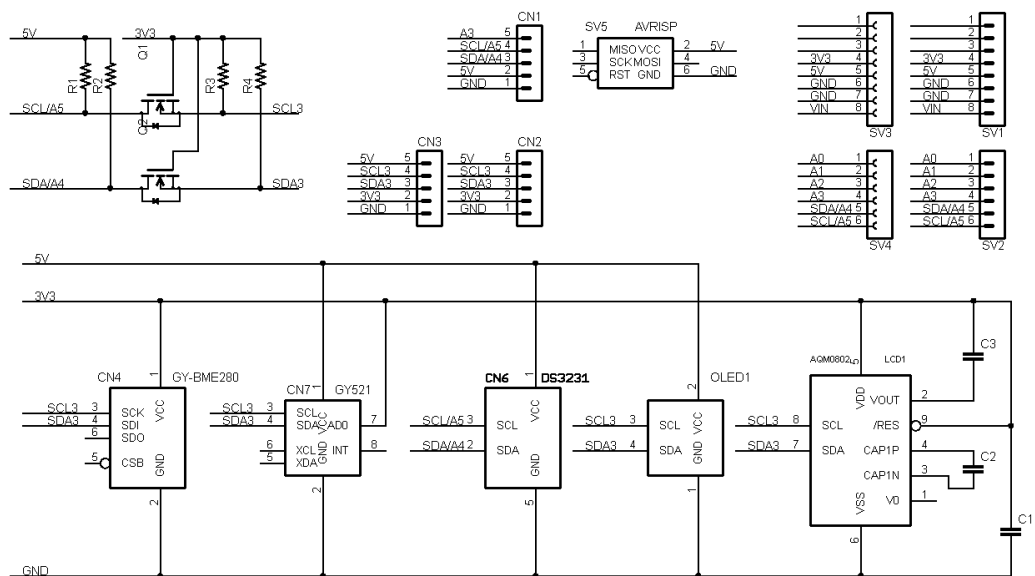


図 5.1 I2C-PLUS-R1 の回路図

5.2 Arduino ボードのピン接続

I2C-PLUS-R1 に接続する Arduino UNO のピン接続を表 5.2 に示します。

表 5.1 Arduino ボードの電源ピン

ピン番号	I2C-PLUS での用途	備考
VIN	-	
GND	GND	
GND	GND	
5V	5V	
3.3V	3.3V	
-	-	
-	-	

表 5.2 Arduino ボードのアナログピン

ピン番号	I2C-PLUS での用途	備考
A0	未使用	
A1	未使用	
A2	未使用	
A3	未使用	CN1 に接続
A4	SDA	
A5	SCL	

5.3 I2C 用コネクタ

5.3.1 CN1: 5V デバイス用

CN1 は、5V の I2C デバイスを接続するためのコネクタです。CN1 のピン配置を表 5.3 に示します。I2C 用の信号線は、Arduino ボードの信号線が直接接続されています。

ピン番号 5 には、追加で A3 が接続されているので、必要に応じて利用してください。

表 5.3 CN1 ピン配置

ピン番号	信号線	備考
1	GND	
2	5V	
3	SDA	A4
4	SCL	A5
5	A3	

5.3.2 CN2, CN3: 3.3V デバイス用

CN2, CN3 は、3.3V の I2C デバイスを接続するためのコネクタです。CN2, CN3 のピン配置を表 5.4, 5.5 に示します。I2C 用の信号線は、Arduino ボードの信号線を 3.3V に変換したものが接続されています。

ピン番号 5 には、追加で 5V の電源が接続されているので、必要に応じて利用してください。

表 5.4 CN2 ピン配置

ピン番号	信号線	備考
1	GND	
2	3V3	
3	SDA	A4
4	SCL	A5
5	5V	

表 5.5 CN3 ピン配置

ピン番号	信号線	備考
1	GND	
2	3V3	
3	SDA	A4
4	SCL	A5
5	5V	

5.4 I2C モジュールの搭載コネクタ

I2C-PLUS-R1 には、様々な I2C モジュールを搭載するためのコネクタが用意されています。

5.4.1 CN4: BME280

CN4 は、環境センサー BME280 を搭載したサブボードを接続するためのコネクタで、別途調達したセンサーモジュールを接続します。CN4 のピン配置を表 5.6 に示します。

Arduino ボードとの接続は I2C で、スレーブアドレスは 0x76 です。

電源は 3.3V を供給しており、センサーと I2C 信号線は 3.3V で駆動されます。

表 5.6 CN4 ピン配置

ピン番号	信号線	備考
1	3.3V	
2	GND	
3	SCL	A5
4	SDA	A4
5	-	
6	-	

5.4.2 CN6: DS3231

CN6 は、リアルタイムクロック (RTC) の DS3231 とバックアップ用の電池を搭載したモジュールを接続するためのコネクタで、I2C-PLUS-R1 キットとは別途調達した DS3231 モジュールを接続することができます。CN6 のピン配置を表 5.7 に示します。

Arduino ボードとの接続は I2C で、スレーブアドレスは 0x3E です。

電源は 5V から供給しており、DS3231 と I2C 信号線は 5V で駆動されます。

表 5.7 CN6 ピン配置

ピン番号	信号線	備考
1	3V3	
2	SDA	A4
3	SCL	A5
4	-	
5	GND	

5.4.3 CN7:MPU6050

CN7 は、加速度・ジャイロセンサ MPU6050 を搭載したサブボードを接続するためのコネクタで、別途調達したセンサーモジュールを接続します。CN7 のピン配置を表 5.8 に示します。

Arduino ボードとの接続は I2C で、スレーブアドレスは 0x69 です。

電源は 5V を供給していますが、センサーボードに 3.3V に変換する電源レギュレータが搭載されており、センサーと I2C 信号線は 3.3V で駆動されます。

表 5.8 CN7 ピン配置

ピン番号	信号線	備考
1	5V	
2	GND	
3	SCL	A5
4	SDA	A4
5	-	
6	-	
7	AD0	3.3V に固定
8	-	

第 6 章

購入および問い合わせ先

6.1 ご協力をお願い

製品をより良くし、多くの方々にお楽しみいただけるよう、製品の向上に努めて参ります。問題点やお気づきの点、あるいは製品の企画に対するご希望などございましたら、microfan_shop@yahoo.co.jp までご連絡いただけますようよろしくお願いいたします。末永くご愛顧いただけますよう、お願いいたします。

6.2 販売：ネットショップ

製品の販売はネットショップで行っています。対面販売は行っておりません。

- マイクロファン Yahoo!ショップ
WEB アドレス：<https://store.shopping.yahoo.co.jp/microfan/>
- アマゾン
WEB アドレス：<https://www.amazon.co.jp/s?merchant=A28NHPRKJDC95B>

6.3 製品情報

マイクロファン ラボ
WEB アドレス：<http://www.microfan.jp/>
マイクロファンの製品情報や活用情報を紹介しています。

6.4 問い合わせ先

株式会社ピープルメディア マイクロファン事業部
E-Mail: microfan_shop@yahoo.co.jp
TEL: 092-938-0450
お問い合わせは基本的にメールでお願いいたします。

6.5 所在地

株式会社ピープルメディア マイクロファン事業部
〒811-2316 福岡県糟屋郡粕屋町長者原西 2-2-22-503