

RP2040-SLIM
(RP2040 開発ボード)
取扱説明書

マイクロファン

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

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

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

2023 年 11 月

Copyright © 2023 MicroFan,
All Rights Reserved.

目次

第 1 章	RP2040-SLIM の紹介	1
1.1	製品概要	1
1.2	購入・利用上の注意	2
1.3	マニュアルの記載内容に関して	2
第 2 章	RP2040-SLIM の特徴	4
2.1	Raspberry Pi Pico との相違点	4
2.2	専用の MicroPython ファームウェアの提供	4
2.3	USB インターフェース	5
2.4	電源回路	5
2.4.1	電圧レギュレータ	5
2.4.2	外部への電力供給と注意点	6
2.4.3	外部からの電力供給と注意点	6
2.5	用途別の接続端子の配置	6
2.5.1	ブレッドボードの利用	7
2.5.2	拡張端子への SENSOR-PLUS の接続	8
2.5.3	拡張端子へのディスプレイの接続	9
第 3 章	利用の準備	12
3.1	製品内容	12
3.2	RP2040-SLIM の動作確認	12
3.3	端子等のはんだ付け	13
3.3.1	半田ごての状態の管理	13
3.3.2	実装時のヒント	13
3.3.3	ピンヘッダー SV1, SV2	13
3.3.4	ピンソケット CN2, CN3	13
第 4 章	プログラミング環境の整備	14
4.1	MicroPython の実行環境の整備	14
4.1.1	MicroPython のファームウェアのダウンロード	14
4.1.2	MicroPython のファームウェアの書き込み	14

4.1.3	Thonny: IDE のインストール	15
4.2	MicroPython のプログラム例	16
4.2.1	MicroPython のプログラミング情報	16
4.2.2	LED の点滅	16
4.3	Arduino の実行環境の整備	17
4.3.1	RP2040 用 Arduino 開発環境のインストール	17
4.3.2	Arduino 開発環境の設定	18
4.3.3	Arduino のサンプルスケッチの実行	18
4.3.4	BLINK:LED の単純な点滅	18
4.3.5	スケッチのコンパイルと RP2040-SLIM への書き込み	19
第 5 章	資料	21
5.1	RP2040-SLIM の回路図	21
5.2	基板上的入出力	23
5.3	ブレッドボード用端子	23
5.4	拡張端子	24
5.4.1	CN2:OLED ディスプレイ	24
5.4.2	CN3:TFT ディスプレイ	25
5.4.3	SENSOR-PLUS	27
第 6 章	購入および問い合わせ先	29
6.1	ご協力をお願い	29
6.2	販売：ネットショップ	29
6.3	製品情報	29
6.4	問い合わせ先	29
6.5	所在地	30

表目次

3.1	製品内容	12
5.1	部品表	22
5.2	スイッチと LED	23
5.3	SV1,SV2 ピン配置	23
5.4	CN2(OLED ディスプレイ) ピン配置	24
5.5	CN3(TFT ディスプレイ) ピン配置	26
5.6	SENSOR-PLUS のコネクタと RP2040-SLIM のコネクタとの接続	28

目次

2.1	RP2040-SLIM の用途別の接続端子	7
2.2	ブレッドボードに乗せた RP2040-SLIM	8
2.3	SENSOR-PLUS と接続した RP2040-SLIM	8
2.4	OLED ディスプレイを搭載した RP2040-SLIM	10
2.5	TFT ディスプレイを搭載した RP2040-SLIM	10
4.1	LED の点滅	16
4.2	端子に割り当てられた名前	17
4.3	BLINK:LED の単純な点滅	19
5.1	RP2040-SLIM の回路図	21
5.2	RP2040-SLIM の部品配置	22
5.3	OLED ディスプレイ	25
5.4	TFT ディスプレイ	26
5.5	SENSOR-PLUS	27

- WS2812B タイプのカラー LED を搭載しています。
- ドロップアウトが 300mV と少ない 800mA ^{*1}の電圧レギュレータを搭載し、RP2040 に安定した電源を供給できます。
- RP2040 の信号線をブレッドボード上で使用するものと、拡張端子を通じて他のモジュール等と接続して利用する 2 種類に分け、プロトタイピングの回路作成を効率よく行えるようにしています。
- ブレッドボードで使用する信号線が接続されている端子列の幅は、ブレッドボードを効果的に活用できるように Raspberry Pi Pico と比べて狭く設計されています。
- 主要な入出力やセンサーを取りまとめた SENSOR-PLUS を拡張端子に接続して利用することができます。
- 様々な情報を表示できる OLED ディスプレイ (別売) や TFT ディスプレイ (別売) を拡張端子に直接接続して利用することができます。

1.2 購入・利用上の注意

RP2040-SLIM をご購入の際には、下記項目をご確認ください。

- SENSOR-PLUS は別売りです。
 - <https://www.amazon.co.jp/dp/B0C6FW4HWQ/>
- OLED ディスプレイは別売りです。
 - <https://store.shopping.yahoo.co.jp/microfan/oled096-128x64-i2c-blue.html>
 - <https://www.amazon.co.jp/dp/B06Y4TKL1F/>
- TFT ディスプレイは別売りです。
 - <https://store.shopping.yahoo.co.jp/microfan/TFT144-128x128.html>
 - <https://www.amazon.co.jp/dp/B0BN5XGRTW/>
- ブレッドボードは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/breadboard-63.html>

1.3 マニュアルの記載内容に関して

RP2040 やそれに関連するハードウェアやソフトウェアは、機能の追加や改良が頻繁に行われているため、本文書で提供している情報は、RP2040-SLIM の購入者の利用時にはすでに古い情報になっている可能性があります。そのため、本文書で示している内容と異なる部分があったり、本文書で示している手順ではうまく動作しないことがあることと、その場合には、各自で対処方法を調査・確認していただく必要があることをご承知おきください。

本マニュアルの記載内容と、ご提供するソフトウェア、ハードウェアに差異がある場合には、ご

^{*1} 放熱の制限で、継続的に 800mA の電流を使用することはできません

指摘によりマニュアルの迅速な訂正を心がけますが、ご提供するソフトウェア、ハードウェアの現品の仕様が優先されます。

お伝えする内容と本質的な問題がない場合には、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。

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。

第 2 章

RP2040-SLIM の特徴

2.1 Raspberry Pi Pico との相違点

RP2040-SLIM は基本的には Raspberry Pi Pico と同じように利用できますがいくつかの相違点があります。

- Raspberry Pi Pico のフラッシュメモリは 2MB ですが、RP2040-SLIM のフラッシュメモリは 8MB です。
- RP2040-SLIM には Raspberry Pi Pico にはないリセットボタンが装備されています。
- Raspberry Pi Pico の USB コネクタは microB ですが、RP2040-SLIM の USB コネクタは Type-C です。
- Raspberry Pi Pico で VBUS の監視用に使用されていた GPIO24 は、RP2040-SLIM では WS2812B カラー LED に接続されています。
- RP2040-SLIM では、Raspberry Pi Pico でシステム用に使用されていた GPIO23、29(ADC3) をユーザーが使用できます。
- Raspberry Pi Pico で端子に引き出されている ADC_VREF は RP2040-SLIM では引き出されていません。
- Raspberry Pi Pico と RP2040-SLIM の接続端子の配置と幅は異なります。
- Raspberry Pi Pico の電源回路は昇圧も可能で、1.8V-5.5V の幅広い入力に対応していますが、RP2040-SLIM の電源回路は 4.5-5.5V の電源入力が必要です。

2.2 専用の MicroPython ファームウェアの提供

MicroPython は様々な MCU の開発ボードに対応しており、オリジナルのファームウェアは、以下のページからダウンロードすることができます。

- <https://micropython.org/download/>

RP2040-SLIM で MicroPython を利用する場合には、このページからダウンロードできる Raspberry Pi Pico 用のファームウェアを書き込んで利用することができますが、以下の様な制

約があります。

- 8MB に拡張されたフラッシュメモリの増加分を有効に利用できない。
- 開発ボードで用途が定まっている端子に分かりやすい名前を付けて利用できない。
- 頻繁に利用する入出力やセンサーのドライバ・ライブラリが組み込まれていない。

マイクロファンは、これらの問題を解決した RP2040-SLIM 専用のファームウェアを作成し、以下のページで配布しています。

- <https://www.microfan.jp/micropython-firmware/>

RP2040-SLIM にこの専用ファームウェアを書き込んで使用することにより、MicroPython を使用する上での利便性を大幅に向上させることができます。RP2040-SLIM で MicroPython を利用する場合には、この専用ファームウェアをご利用ください。

2.3 USB インターフェース

開発ボードと PC を接続するために、USB インターフェース (Type-C) を備えています。

USB インターフェースは以下のような用途で使用されます。

- RP2040-SLIM への電力供給。
USB からは 5V の電力が供給され、RP2040-SLIM では、基板上の電圧レギュレータで 3.3V に変換され使用されます。
- RP2040-SLIM と PC 間のシリアル通信。
Arduino のシリアル入出力や、MicroPython の REPL の入出力に使用されます。
- RP2040-SLIM へのスケッチ (プログラム) やファームウェアの書き込み。
- PC に対するマストストレージ機能の提供 (ブートローダーモード)。

2.4 電源回路

ブレッドボード上で回路の試作や実験を行うためには、電源回路が必要になります。RP2040-SLIM には USB から必要な電力を取得する電源回路が組み込まれており、USB で PC に接続して開発を行う場合には、外部に別途電源を用意する必要がありません。

2.4.1 電圧レギュレータ

RP2040-SLIM で利用している電圧レギュレータ ME6217 は、少なくとも 800mA 以上の電流を供給*1できますので RP2040 を余裕をもって稼働させることができます。

また、ME6217 の入力電圧から出力電圧のドロップダウンは 300mV 程度で、USB から電力を

*1 ただし PC に接続された USB2.0 からの供給電流は最大で 500mA、USB3.0 からは 900mA です。また、放熱の制限で継続して 800mA の電流を使用することはできません。

取得する場合、ショットキーダイオードの順方向電圧降下と合わせると電圧低下は 0.8V 程度となります。RP2040 が瞬間的に大きな電流を必要としている際に、USB からの供給電圧が定格の 5V をある程度下回っても、安定した電源電圧 3.3V を維持することができます。

2.4.2 外部への電力供給と注意点

RP2040-SLIM が組み込まれているブレッドボード上の回路や、試作基板上の消費電力の小さな回路に対しては、RP2040-SLIM の 5V 端子や 3.3V 端子からそれぞれ電力を取得して使用することができます。

PC に接続された USB 端子からは、500mA あるいは 900mA 程度の電流しか取り出すことができないので、RP2040-SLIM の使用分も含めて、使用する電力の使用量が過大にならないようにご注意ください。

2.4.3 外部からの電力供給と注意点

USB コネクタを通して電力供給を行わない場合には、5V の端子に電源を接続してください。

- 5V 端子に接続する電源の電圧は 6V を超えないようにしてください。6V を超えると電圧レギュレータをはじめとする回路が破損する可能性があります。
- 外部電源にそれ自身に対する保護回路がついている様であれば、USB コネクタが PC 等に接続された状態でも、5V 端子に外部電源の接続を行って構いません。（逆流防止用のダイオードがついているので、PC 等に外部電源からの電圧・電流が逆流することはありません。）
- 3.3V の端子は出力専用で、外部から 3.3V の電源を接続しないでください。3.3V 端子に外部から電力を供給すると回路が破損する可能性があります。

2.5 用途別の接続端子の配置

開発ボードやブレッドボードなどを使用して、プロトタイピングや様々な回路の実験を行う際に、ディスプレイやスイッチなど、いつも使用する入出力回路の配線を毎回行うのを面倒に思ったり、それらの回路のつまらない接続ミスや、接続が不安定で実験がうまくいかないなどの様々なトラブルに見舞われたことはないでしょうか。

プロトタイピングや実験を行うために作成する回路は、以下の 2 種類に分類することができます。

- 主要な実験対象となる回路
- 実験を効率よく行うための、入力や出力・表示などの支援回路

後者の回路は、標準的で接続が安定した回路がいつでも利用できるようになっていて、プロトタイピングや実験を行う際に、前者の回路のみに集中して回路を作成することができれば、プロトタ

イピングや実験を効率よく安定して行うことができます。

この様な考えで RP2040-SLIM の接続端子は、実験回路の作成に利用するための端子群と、標準的な入出力などの支援回路に利用する端子群に分けて配置されています。

- SV1, SV2 はブレッドボード上で実験を行いたい回路の作成に利用
- CN2, CN3 は、入出力モジュールやディスプレイなど、プロトタイピングで標準的に必要とする入出力回路を接続するために利用

図 2.1 に示すように、SV,1,SV2 には開発ボードの裏側にピンヘッダーを取り付けてブレッドボードで利用、CN2, CN3 には開発ボードの表側にピンソケットを取り付けて入出力モジュール等を接続するために利用することを想定しています。

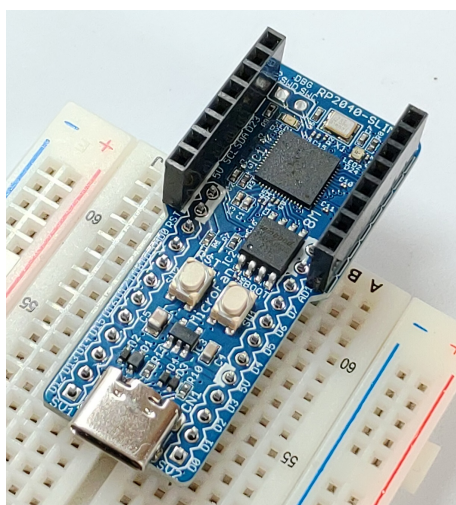


図 2.1 RP2040-SLIM の用途別の接続端子

必要であれば、もちろん NC2, CN3 のピンソケットから、ブレッドボード上に配線することもできます。

2.5.1 ブレッドボードの利用

RP2040-SLIM の 2 列の信号の引き出し端子 SV1, SV2 は、ブレッドボードの部品配置領域を効果的に利用できるように、Raspberry Pi Pico と比較して狭い間隔で配置されています。Raspberry Pi Pico では部品の配置が難しかった一般的なブレッドボードでも、RP2040-SLIM では図 2.2 に示すように基板の両横に余裕をもって部品を配置し利用することができます。

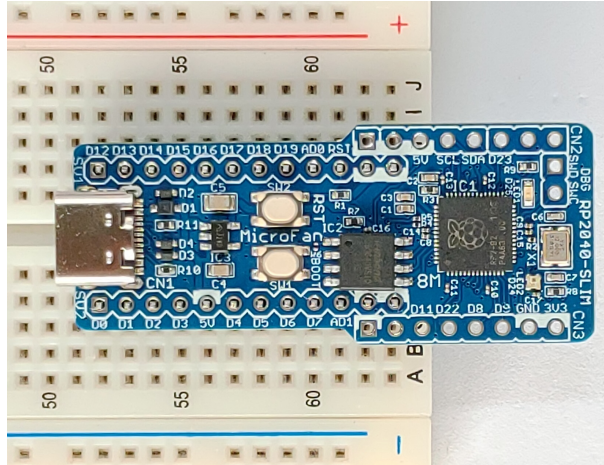


図 2.2 ブレッドボードに乗せた RP2040-SLIM

2.5.2 拡張端子への SENSOR-PLUS の接続

マイクロファンは、拡張端子 CN2, CN3 に接続して利用できる入出力モジュール SENSOR-PLUS を提供しています。SENSOR-PLUS の接続例を図 2.3 に示します。

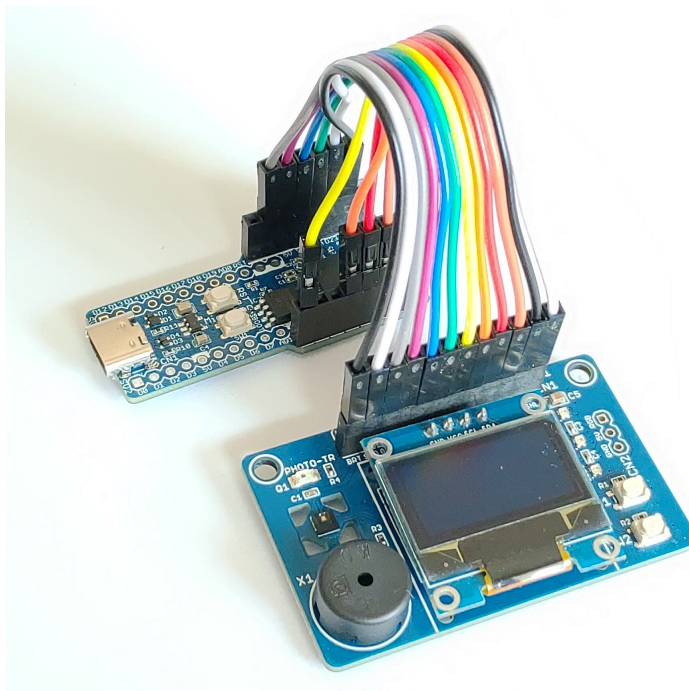


図 2.3 SENSOR-PLUS と接続した RP2040-SLIM

- <https://www.microfan.jp/2023/05/sensor-plus/>

ご購入

- <https://www.amazon.co.jp/dp/B0C6FW4HWQ/>

SENSOR-PLUS には以下に示すような利用頻度の高い入出力装置やセンサーを搭載しており、SENSOR-PLUS を拡張端子に接続することで、安定したプロトタイピングの支援環境を効率よく構築することができます。

- OLED ディスプレイ
- タクトスイッチ x2
- 圧電スピーカー
- 明るさセンサー
- 気温、湿度センサー
- 3 軸加速度センサー
- W2812B カラー LED x3

拡張端子に SENSOR-PLUS を接続することで、多くのプロトタイピング実験で必要となる主要な入出力機能がまとめて組み込まれるため、ブレッドボード上では試したい回路の構成のみに集中して作業を行うことができます。

2.5.3 拡張端子へのディスプレイの接続

プロトタイピングの支援機能として、情報の表示機能さえあればよいという場合には、拡張端子にディスプレイを直接接続して手軽に利用することができます。

RP2040-SLIM の拡張端子 CN2, CN3 の接続端子の並びは、一般的な OLED ディスプレイや TFT ディスプレイの端子の並びに合わせているため、それらのディスプレイを面倒な配線等を行うことなく拡張端子に直接接続することができます。このため、RP2040-SLIM を PC と切り離して単独で運用している場合でも、様々な情報をそれらのディスプレイに表示し確認することが簡単に行えるようになります。

CN2: OLED ディスプレイ

CN2 には図 2.4 に示すように OLED ディスプレイを直接接続することができます。

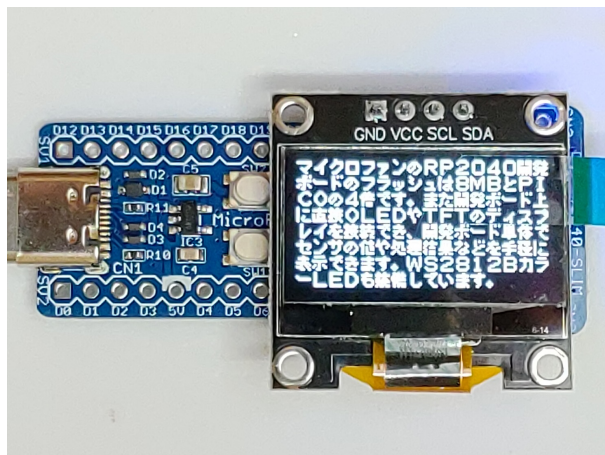


図 2.4 OLED ディスプレイを搭載した RP2040-SLIM

OLED ディスプレイは、128x64 ドットのグラフィックディスプレイになっており、ボードの稼働状態や利用者に伝えたい情報を、画像や文字で分かり易く表示できるようになります。

OLED ディスプレイの端子と表示モジュールに関しては、5.4 節をご参照ください。

CN3: TFT ディスプレイ

CN3 には図 2.5 に示すように TFT ディスプレイを直接接続することができます。



図 2.5 TFT ディスプレイを搭載した RP2040-SLIM

TFT ディスプレイは、128x128 ドットのカラーグラフィックディスプレイになっており、開発ボードの稼働状態や利用者に伝えたい情報を、カラー画像や文字で分かり易く表示できるように

なります。

TFT ディスプレイの端子と表示モジュールに関しては、5.4 節をご参照ください。

第 3 章

利用の準備

RP2040-SLIM の利用に先立って、必要に応じて、ピンヘッダやピンソケットのはんだ付けを行います。

3.1 製品内容

RP2040-SLIM の製品内容を表 3.1 に示します。表 3.1 には開発ボード以外の関連部品を含めて示しています。基板が破損している場合には、ご利用になる前にマイクロファンにお問い合わせください。

表 3.1 製品内容

部品	シンボル	備考	個数
プリント基板	RP2040-SLIM	Rev.1	1
ピンヘッダ	SV1, SV2	1x12 ピン	2 または相当品
ピンソケット	CN2, CN3	1x8 ピン	2

3.2 RP2040-SLIM の動作確認

RP2040-SLIM をご購入なさったら、利用に先立ち基板を USB ケーブルで PC 等に接続してください。まず、USB インターフェースが認識され、次に、RP2040-SLIM がディスクドライブとして認識されます。

PC の設定にもよりますが、ディスクの内容がエクスプローラーに表示され、INDEX.HTM, INFO_UF2.TXT などが確認できます。

ここで問題があれば、マイクロファンにお問い合わせください。

3.3 端子等のはんだ付け

3.3.1 半田ごての状態の管理

開発ボードにピンヘッダやピンソケットをはんだ付けされる方のために、まず最初に、はんだ付けを行う際の一般的な半田ごての状態の管理に関して示します。

はんだ付けを行う直前に、スポンジなどのこて先クリーナーで半田ごてをクリーニングしてフラックスや酸化膜などの汚れを取り除き、こて先が銀色に輝く状態ではんだ付けを行います。また、こて先にほとんどはんだが乗っておらず乾いていると、こて先から部品のピンや基板のパターンなどに熱が伝わりにくいので、こて先に少し（薄く）はんだを付けてこて先がはんだで濡れた状態にしてはんだ付けを行います。

3.3.2 実装時のヒント

ピンソケット、ピンヘッダーなどの複数の端子を持つ部品のはんだ付けは、端子の端の1ピン、もしくは両端か対角上の2ピンをはんだ付けし、部品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、部品の姿勢をきれいに整えて取り付けることができます。

RP2040-SLIMのプリント基板はベタアースになっており、熱容量が大きくなっております。このため、各部品のGND端子をはんだ付けする際には基板の端子部分（ランド）の温度が上がりはんだが融けるまで少し時間がかかるため、他の端子と比較して長めにはんだごてを当てておく必要がありますのでご注意ください。

3.3.3 ピンヘッダー SV1, SV2

ピンヘッダーはブレッドボードに挿せるように、SV1, SV2の基板の裏側に取り付けます。ピンヘッダーは基板裏面からピンの短いほうをプリント基板に取り付け、プリント基板の表面（おもてめん）ではんだ付けします。

ピンヘッダーを開発ボードにはんだ付けする際に、はんだごてが基板上の部品に接触しないようご注意ください。

3.3.4 ピンソケット CN2, CN3

CN2, CN3にピンソケットをはんだ付けすると、SV1, SV2のはんだ付けが非常に行いにくくなるので、先にSV1, SV2のはんだ付けをお勧めします。

ピンソケットは、CN2, CN3の基板の表側に取り付けます。

第 4 章

プログラミング環境の整備

RP2040 にはいくつかのプログラミング言語・環境が提供されていますが、ここでは人気の高い MicroPython と Arduino のプログラミング環境の整備法を紹介します。

4.1 MicroPython の実行環境の整備

4.1.1 MicroPython のファームウェアのダウンロード

RP2040-SLIM で MicroPython をご利用になる場合には、MicroPython のファームウェアを書き込む必要があります。RP2040-SLIM には、以下の WEB ページからダウンロードできる Raspberry Pi Pico 用のファームウェアを使用することができます。

- https://micropython.org/download/RPI_PICO/

しかしながら、この標準的なファームウェアでは、Raspberry Pi Pico と異なる発想で開発された RP2040-SLIM を効果的に使用することができません。

マイクロファンは、RP2040-SLIM の利便性を高めるように構成された MicroPython のファームウェアを作成し、以下の WEB ページで配布しています。

- <https://www.microfan.jp/micropython-firmware/>

上記の WEB ページで、RP2040-SLIM 用の最新のファームウェアをダウンロードしてご利用ください。このファームウェアには、RP2040-SLIM を快適に使用するための様々な拡張が施されています。

4.1.2 MicroPython のファームウェアの書き込み

RP2040-SLIM へのファームウェアの書き込み法は、下記のように Raspberry Pi Pico への書き込みと同様です。

- BOOT ボタンを押した状態で PC に接続された USB コネクタを RP2040-SLIM に接続する。あるいは、PC と RP2040-SLIM がすでに USB コネクタで接続されている場合には、

BOOT ボタンを押した状態で RST ボタンを押して放す。

- RP2040-SLIM が USB ディスクと認識され、その内容を示すファイルエクスプローラーが開くので、そこにダウンロードしておいた MicroPython のファームウェアをドラッグアンドドロップあるいはコピーする。
- フォルダへのファームウェアのコピーが終了すると、ファームウェアが自動的に RP2040-SLIM に書き込まれ再起動される。Thonny などを使用している場合には、RP2040-SLIM に割り当てられた USB シリアルデバイス (COM ポート) を選択して RP2040-SLIM を操作する。

4.1.3 Thonny: IDE のインストール

MicroPython のファームウェアを書き込んだ RP2040-SLIM を PC から操作する場合には、PC に TeraTerm などのシリアル通信用のアプリケーションをインストールして使用することができます。

しかしながらこの方法では、MicroPython の利用には不便な点が多いので、MicroPython を手軽に利用するための様々な支援機能が組み込まれた IDE を利用しましょう。MicroPython の IDE にはいくつかの候補がありますが、ここでは Raspberry Pi のソフトウェアパッケージに Python 用 IDE としてあらかじめ組み込まれている Thonny を紹介します。

Thonny の情報サイトを以下に示します。

- <https://thonny.org/>

Thonny には様々な機能がありますが、例えば以下のような機能が、RP2040-SLIM 上の MicroPython を快適に使用するために大変役立ちます。

- Python の文法を理解した組込みエディタが使える。
- MicroPython が RP2040 上に作成しているファイルシステムの操作を IDE のエディタなどと連携して行える。これにより、Python プログラムの RP2040 への書き込み・保存や、ライブラリ・モジュール等のデバイスへの登録が簡単に行える。
- RP2040 を使用した開発ボードへの MicroPython のファームウェア書き込み機能が組み込まれている。
- PC 上で稼働する Python がインストールされるため、RP2040 の MicroPython だけでなく、PC の標準的な Python を使用できる。

Thonny のインストール法と利用法は、下記の WEB ページで説明しているのでご参照ください。

- <https://www.microfan.jp/2023/04/thonny/>

4.2 MicroPython のプログラム例

ここでは、RP2040-SLIM での MicroPython のプログラム例を示します。

4.2.1 MycroPython のプログラミング情報

RP2040 の信号線を操作するために、PC 上で Python を使用している方にはなじみのないライブラリ・モジュールを使用していますが、以下の文書を参考にしてみてください。

- <https://micropython-docs-ja.readthedocs.io/ja/latest/rp2/quickref.html>
RP2040 用のクイックリファレンスがあり、RP2040 で MicroPython を使用する際に大変役立ちます。
- <https://micropython-docs-ja.readthedocs.io/ja/latest/library/index.html>
MicroPython のライブラリに関しては、このページにまとめられています。

また、マイクロファンのファームウェアを使用した MycroPython での RP2040-SLIM のプログラミングは、様々な例を下記の WEB ページに示しているのご活用ください。

- <https://www.microfan.jp/2023/09/rp2040-uno-plus-micropython/>

4.2.2 LED の点滅

ここでは、電子工作で定番の LED の点滅プログラムの実行例を紹介します。RP2040-SLIM に搭載されている LED1 を点滅させるプログラムです。

Thonny を使用している場合には、ウィンドウの上側の編集ペインにプログラムを入力し、緑色の実行ボタンを押して実行させることができます。この方法をとると、プログラムに誤りがあった場合には、その部分だけを修正して再度実行ボタンを押すとプログラムの動作を確認できるのでとても便利です。

```
from machine import Pin
import time

led1 = Pin('LED1', Pin.OUT) # LED1 を出力に設定, 25 でもよい

while True:
    led1.on()
    time.sleep_ms(1000) # 1000ms(1秒) 待つ
    led1.off()
    time.sleep_ms(1000)
```

図 4.1 LED の点滅

LEDのピンを'LED1'と指定できるのは、マイクロファンが提供するMicroPythonのファームウェアを使用した場合で、オリジナルのMicroPythonファームウェアを使用した場合には、'LED1'と指定することはできません。

プログラムは無限ループですが、CTRL-Cの入力や、Thonnyの赤色のSTOPアイコンをクリックしてで中断させることができます。

マイクロファンが提供するMicroPythonのファームウェアで端子にどのような名前が割り当てられているかは、以下の入力で確認することができます。

```
>>> from machine import Pin
>>> help(Pin.board)
```

図 4.2 端子に割り当てられた名前

4.3 Arduinoの実行環境の整備

4.3.1 RP2040用Arduino開発環境のインストール

Arduinoの開発環境のインストールは以下の2段階の手順で行います。

- 基本となるArduino IDEのインストール
- RP2040用の開発機能の追加

この後は、必要に応じて、各種のライブラリの追加インストールを行います。

下記のインストール法がわかりにくい様であれば、WEBで検索をするとインストール法を示したページが複数見つかるので、ご自身がわかりやすいと思うページを参照してインストールを行ってください。

基本となるArduino IDEのインストール

以下のページからダウンロードオプションで、ご自身が使用しているOS用のインストールパッケージを選択しダウンロードしインストールしてください。

- <https://www.arduino.cc/en/software>

Arduino IDEがインストールできたら起動してください。

メニュー等を日本語化するために、Arduino IDEの[File] ⇒ [Preferences...] ⇒ [Settings] タブの[Language:]を日本語に設定してください。

RP2040用の開発機能の追加

RP2040用のArduinoは以下のWEBページで公開されています。

- <https://github.com/earlephilhower/arduino-pico>

インストール方法も示されているので、示されている手順に従って RP2040 用の Arduino のインストールを行ってください。

4.3.2 Arduino 開発環境の設定

RP2040-SLIM と PC を USB で接続してください。

RP2040-SLIM に MicroPython のファームウェアなどを書き込んでいない初期状態であれば、次の操作に進んでください。すでに MicroPython のファームウェアなどを書き込んだ状態であれば、RST ボタンと BOOT ボタンを一緒に押し、先に RST ボタンを離し、次に BOOT ボタンを離すようにしてください。この操作で、RP2040-SLIM がブートローダーモードに移行します。

なお、RP2040-SLIM に Arduino のスケッチをいったん書き込むと、以降のスケッチの書き込みでは、ブートローダーモードに移行する操作をする必要はなくなります。

Arduino IDE のメニューの「ツール」を選択してメニューを表示してください。このメニューの中から、まず、開発用のボードと PC と RP2040-SLIM の接続を行うポートを設定します。

開発ボードの選択は、[ボード:] メニューの [Raspberry Pi Pico/RP2040] の中から、初めの方に表示されている [Raspberry Pi Pico] を選択してください。

また、[ポート:] は、[UF2 Board] という選択肢があるので、それを選択して設定してください。MAC の場合には、ポートの選択方法は WEB などを確認してください。

面倒でなければ、以下の様な設定で、RP2040-SLIM の 8MB フラッシュを最大限に使用できるようになります。

開発ボードの選択は、[ボード:] メニューの [Raspberry Pi Pico/RP2040] の中から、最後の方に表示されている [Generic RP2040] を選択してください。

この設定では、フラッシュサイズに関連する以下の追加設定をメニューから行ってください。他の設定は既定の状態の問題ありません。

基本的には、8M のフラッシュを使用できるようにします。

- Boot Stage 2
W25QV64JV QSPI /4 を選択
- Flash Size 4種類ある [8MB (—)] のどのどれかを選択、当面は (Sketch: 4MB, FS: 4MB) にしておくといでしょう。

4.3.3 Arduino のサンプルスケッチの実行

RP2040-SLIM の動作確認のために、LED 点滅スケッチの実行を試してみましょう。

4.3.4 BLINK:LED の単純な点滅

電子工作界の hello world、LED の点滅スケッチを実行しましょう。

Arduino IDE の [ファイル] ⇒ [スケッチの例] ⇒ [01.Basics] から Blink を選択してください。RP2040-SLIM の LED は Raspberry Pi Pico と同様 25 番ピンに接続されているので、スケッチ

の `pinMode()`, `digitalWrite()` の第 1 引数に指定されている `LED_BUILTIN` を変更せずに使用できます。

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

図 4.3 BLINK:LED の単純な点滅

4.3.5 スケッチのコンパイルと RP2040-SLIM への書き込み

スケッチのコンパイルを行う前に、COM ポートの設定がされているか確認してください。RP2040-SLIM にまだ何も書き込みが行われていないか、BOOT ボタンを押した状態で USB コネクタを接続した場合には、RP2040-SLIM はブートローダーモードにあります。

ブートローダーモードでは、Arduino IDE のポート設定の候補には、[UF2 Board] という項目が表示されているので、ポートとしてそれを選択します。また、RP2040-SLIM にすでに Arduino のスケッチを書き込んでいる場合には、[COMX(X は数値)] という項目が表示されているので、ポートとしてそれを選択します。

サンプルスケッチを開いたら、まず問題なくコンパイルを行えるかどうか、Arduino IDE の左上部のチェックマーク [検証] のアイコンをクリックして、スケッチをコンパイルします。

問題なくコンパイルできたならば、先ほどのアイコンの右隣の右矢印マーク [書き込み] のアイコンをクリックします。スケッチの再コンパイルの後に、Arduino IDE の下部のメッセージエリアに白色の文字で数行のメッセージが出て、スケッチの書き込みが行われます。

スケッチが RP2040-SLIM に正しく書き込まれたら、ボード上の LED が点滅します。

スケッチの書き込みが失敗するようであれば、COM ポートの設定がされているか確認してください。

いったんスケッチを書き込むと、以降はスケッチの書き込みの際にブートローダーモードに移行する操作をする必要はなくなります。また、ポートも、[COMX(X は数値)] の形式に変化しているため、それを選択して設定します。

ブートローダーモードでの最初のスケッチの書き込みが終了したら、ポートの選択肢に [UF2 Board] がなくなり、代わりに COMX が表示されるようになるので、その COMX をポートに再

設定するようにしてください。

第 5 章

資料

5.1 RP2040-SLIM の回路図

RP2040-SLIM の回路図を図 5.1、部品表を表 5.1 に示します。

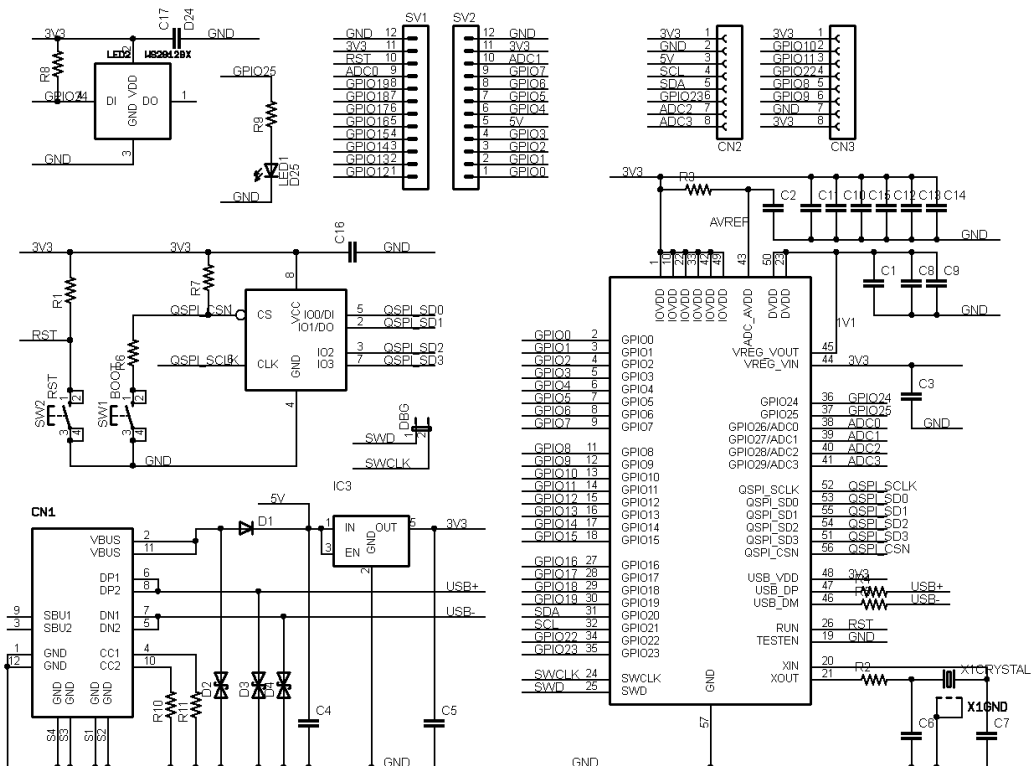


図 5.1 RP2040-SLIM の回路図

表 5.1 部品表

部品	シンボル	規格等	個数
プリント基板	RP2040-SLIM	Rev.1	1
IC	IC1	RP2040	1
	IC2	W25Q64JV	1
	IC3	ME6217	1
水晶	X1	12MHz	1
ショットキーダイオード	D1	1N5819WS	1
ESD ダイオード	D2,D3,D4	LESD5D5.0CT1G	3
発光ダイオード	LED1	青	1
	LED2	WS2812B	1
抵抗	R1, R7, R8	10K Ω	3
	R2, R6	1K Ω	2
	R3	200 Ω	1
	R4, R5	27 Ω	2
	R9	470 Ω	1
	R10, R11	5.1K Ω	2
セラミックコンデンサ	C1-C3	2.2 μ F	3
	C4	10 μ F	1
	C5	47 μ F	1
	C6,C7	22pF	2
	C8-C17	0.1 μ F	10
タクトスイッチ	SW1, SW2	2 端子	2
USB	CN1	Type-C	1
ピンヘッダ	SV1, SV2	12 ピン	2
ピンソケット	CN2, CN3	8 ピン	2

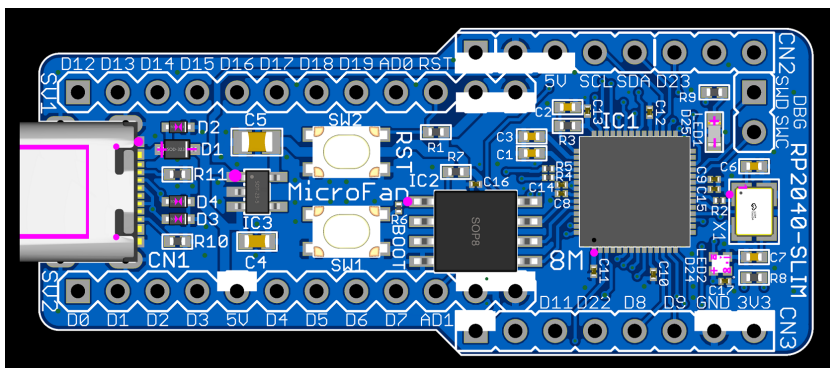


図 5.2 RP2040-SLIM の部品配置

5.2 基板上の入出力

RP2040-SLIM の基板上のスイッチと LED を表 5.2 に示します。

SW1 は、リセット時のブートモード（スケッチやファームウェアの書き込み）の切り替え用です。通常の入力信号として利用することはできません。

表 5.2 スイッチと LED

シンボル	信号線	備考
SW1	-	BOOT ローダーモード移行用
SW2	RST	リセット用
LED1	D25	正論理
LED2	D24	WS2812B

5.3 ブレッドボード用端子

RP2040-SLIM には、ブレッドボードに挿して利用するためのピンヘッダー用端子 SV1, SV2 が用意されています。SV1,SV2 のピン配置を表 5.3 に示します。

表 5.3 SV1,SV2 ピン配置

備考	SV1 信号線	ピン番号	SV2 信号線	備考
	GND	12	GND	
	3V3	11	3V3	
	RST	10	D27	ADC1/SCL1
SDA1/ADC0	D26	9	D7	
MOSI0	D19	8	D6	
SCK0	D18	7	D5	SCL
SS0	D17	6	D4	SDA
MISO0	D16	5	5V	
MOSI1	D15	4	D3	
SCK1	D14	3	D2	
SS1	D13	2	D1	RX1
MISO1	D12	1	D0	TX1

表 5.3 に示されている I2C や SPI のピン配置は、Arduino でのデフォルト設定に準拠しています。

- <https://github.com/earlephilhower/arduino-pico/blob/master/variants/>

rpipico/pins_arduino.h

I2C や SPI を使用する場合には、表 5.3 以外のピンを明示的に割り当てて使用することができます。

5.4 拡張端子

ブレッドボードへの接続を想定して配置された端子群 SV1, SV2 とは別に、ディスプレイや外部拡張ボードとの接続用に端子群 CN2, CN3 が配置されています。CN2, CN3 には、ピンソケットを接続することが想定されています。

これらの拡張端子には、OLED ディスプレイや TFT ディスプレイを接続するほかに、SENSOR-PLUS を接続することもできます。

5.4.1 CN2:OLED ディスプレイ

CN2 のピン配置を表 5.4 に示します。

CN2 は、OLED ディスプレイを直接接続できるように、OLED ディスプレイの端子配置に合わせて GND, 5V, SCL, SDA が隣接して配置されています。

また、OLED の直接の接続端子としてではなく、汎用的な拡張端子としての使用に備えて、CN2 には OLED ディスプレイ接続用の端子の他に、ADC 入力として使用できる端子 2 個を含む 3 端子が配置されています。

表 5.4 CN2(OLED ディスプレイ) ピン配置

ピン番号	信号線	備考
1	3V3	
2	GND	
3	5V	
4	D21	SCL
5	D20	SDA
6	D23	
7	D28	ADC2
8	D29	ADC3

CN2 に OLED ディスプレイを直接接続する場合には、CN2 の 2-5 番ピンに間違いなく接続するように十分に注意してください。接続位置を間違えると、OLED ディスプレイあるいは RP2040-SLIM が損傷することがあります。

推奨する OLED ディスプレイを図 5.3 に示します。また、ネットショップ URL を以下に示します。

- <https://store.shopping.yahoo.co.jp/microfan/oled096-128x64-i2c-blue>.

html

- <https://www.amazon.co.jp/dp/B06Y4TKL1F>

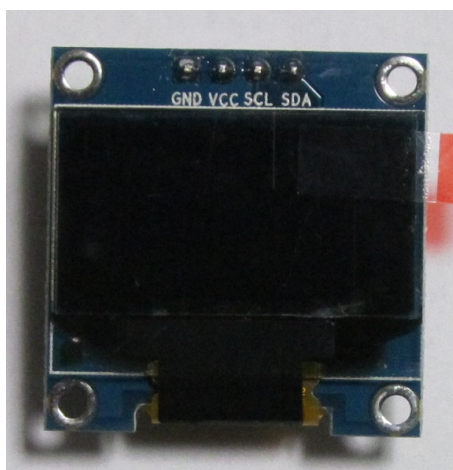


図 5.3 OLED ディスプレイ

OLED ディスプレイに要求される機能を以下に示します。

- モジュールを直接コネクタに刺すためには、信号線の並びが表 5.4 の順になっていること。
- SCL, SDA の信号線が 3.3V 対応であること。
- RP2040-SLIM には I2C 用のプルアップ抵抗が組み込まれていないため、SCL, SDA の信号線にプルアップ抵抗が付与されていること。
- RP2040-SLIM からの電源として 5V を供給しているため、3.3V の電圧レギュレータが内蔵されていること。
- 使用するライブラリにもよりますが、コントローラに SSD1306 か SH1106 を使用していること。

推奨する OLED ディスプレイの SDA, SCL 信号線には、4.7K-10K Ω のプルアップ抵抗が組み込まれています。このため、CN2 に OLED ディスプレイを接続している場合には、ブレッドボード上で I2C デバイスを使用する際に、SDA, SCL に別途プルアップ抵抗を接続する必要はありません。(付けた場合には、OLED ディスプレイのプルアップ抵抗との合成抵抗値となります。)

5.4.2 CN3:TFT ディスプレイ

CN3 のピン配置を表 5.5 に示します。

CN3 は、TFT ディスプレイを直接接続できるように、TFT ディスプレイの端子配置に合わせて端子が配置されています。

表 5.5 CN3(TFT ディスプレイ) ピン配置

ピン番号	信号線	備考
1	3V3	
2	GND	
3	D9	CS
4	D8	RESET
5	D22	DC
6	D11	MOSI
7	D10	SCK
8	3V3	LED

推奨する TFT ディスプレイを図 5.4 に示します。また、ネットショップ URL を以下に示します。

- <http://store.shopping.yahoo.co.jp/microfan/tft144-128x128.html>
- <https://www.amazon.co.jp/dp/B0BN5XGRTW>

また、下記の 1.8 インチ 128x160 ピクセルの TFT ディスプレイもコントローラと端子の並びが共通しているので、スケッチでの初期化法を少し変更^{*1}することにより利用可能です。

- <https://www.amazon.co.jp/dp/B0BN5XKFFL>



図 5.4 TFT ディスプレイ

TFT ディスプレイに要求される機能を以下に示します。

^{*1} 実害はないのですが、ライブラリによりませんが、画面の端に少しノイズが出ます。これを除去するためには、ライブラリのソースコードを少し変更する必要があります。

- モジュールを直接コネクタに刺すためには、信号線の並びが表 5.5 の順になっていること。
- 全ての信号線が 3.3V 対応であること。
- RP2040-SLIM からの電源として 3.3V を供給しているため、モジュールに内蔵の 3.3V の電圧レギュレータを無効にできること。(保証はできませんが、多くの場合、無効にしなくても問題なく動くようです)
- 使用するライブラリにもよりますが、コントローラに ST7735 を使用していること。

5.4.3 SENSOR-PLUS

CN2, CN3 の端子を組み合わせ、よく利用する入出力とセンサーを組み込んだ SENSOR-PLUS を接続して利用することができます。

製品紹介

- <https://www.microfan.jp/2023/05/sensor-plus/>

ご購入

- <https://www.amazon.co.jp/dp/B0C6FW4HWQ/>

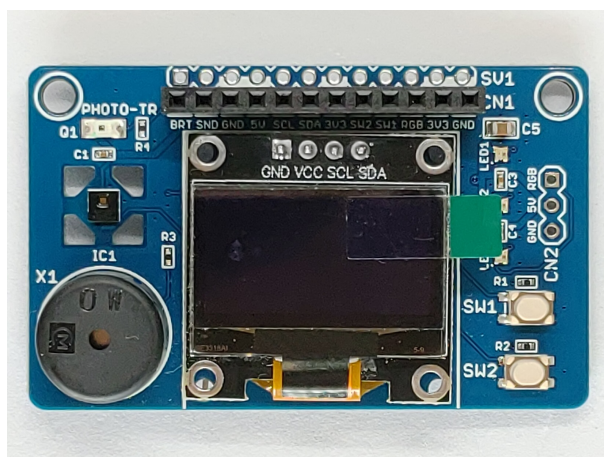


図 5.5 SENSOR-PLUS

SENSOR-PLUS のコネクタと RP2040-SLIM のコネクタとの接続方を 5.6 に示します。

表 5.6 SENSOR-PLUS のコネクタと RP2040-SLIM のコネクタとの接続

SENSOR-PLUS ピン番号	RP2040-SLIM コネクタ：ピン番号	信号線番号、信号線機能名	備考
1	CN2:7	28, BRTE, ADC2, GPIO28	明るさセンサー
2	CN2:6	23, SNDE, GPIO23	圧電スピーカー
3	CN2:2	-	GND
4	CN2:3	-	5V
5	CN2:4	21, SCL0, GPIO21	I2C0
6	CN2:5	20, SDA0, GPIO20	I2C0
7	CN3:1	-	3.3V
8	CN3:6	11, SW2E, GPIO11	
9	CN3:7	10, SW1E, GPIO10	
10	CN3:5	22, RGBE, GPIO22	
11	-		
12	-		

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。対面販売は行っていません。

- 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