

RP2040-UNO  
(RP2040 開発ボード)  
取扱説明書

マイクロファン

<https://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

<b>第 1 章</b>	<b>RP2040-UNO の紹介</b>	<b>1</b>
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
<b>第 2 章</b>	<b>RP2040-UNO の特徴</b>	<b>4</b>
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	拡張端子 . . . . .	6
2.5.1	拡張端子への SENSOR-PLUS の接続 . . . . .	7
2.5.2	拡張端子へのディスプレイの接続 . . . . .	8
<b>第 3 章</b>	<b>利用の準備</b>	<b>10</b>
3.1	製品内容 . . . . .	10
3.2	RP2040-UNO の動作確認 . . . . .	10
3.3	ディスプレイの取り付け . . . . .	10
3.4	ディスプレイの簡易取り付け . . . . .	11
3.4.1	概要と注意点 . . . . .	11
3.4.2	ピンの加工と基板への取り付け . . . . .	11
<b>第 4 章</b>	<b>プログラミング環境の整備</b>	<b>13</b>
4.1	MicroPython の実行環境の整備 . . . . .	13
4.1.1	MicroPython のファームウェアのダウンロード . . . . .	13
4.1.2	MicroPython のファームウェアの書き込み . . . . .	13
4.1.3	Thonny: IDE のインストール . . . . .	14
4.2	MicroPython のプログラム例 . . . . .	15

4.2.1	MycroPython のプログラミング情報 . . . . .	15
4.2.2	LED の点滅 . . . . .	15
4.3	Arduino の実行環境の整備 . . . . .	16
4.3.1	RP2040 用 Arduino 開発環境のインストール . . . . .	16
4.3.2	Arduino 開発環境の設定 . . . . .	17
4.3.3	Arduino のサンプルスケッチの実行 . . . . .	17
4.3.4	BLINK:LED の単純な点滅 . . . . .	17
4.3.5	スケッチのコンパイルと RP2040-UNO への書き込み . . . . .	18
<b>第 5 章</b>	<b>資料</b>	<b>20</b>
5.1	RP2040-UNO の回路図 . . . . .	20
5.2	基板上的入出力 . . . . .	22
5.3	シールド等接続端子 . . . . .	22
5.4	拡張端子 . . . . .	25
5.4.1	CN2, CN3:RC サーボ . . . . .	25
5.4.2	CN4:SENSOR-PLUS . . . . .	25
5.4.3	CN5:TFT ディスプレイ . . . . .	26
5.4.4	CN6:OLED ディスプレイ . . . . .	28
<b>第 6 章</b>	<b>購入および問い合わせ先</b>	<b>30</b>
6.1	ご協力をお願い . . . . .	30
6.2	販売：ネットショップ . . . . .	30
6.3	製品情報 . . . . .	30
6.4	問い合わせ先 . . . . .	30
6.5	所在地 . . . . .	31

# 表目次

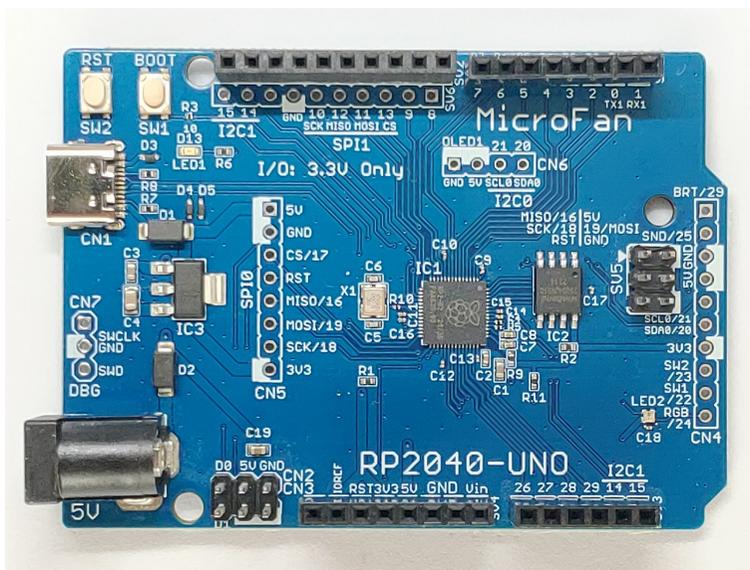
3.1	製品内容 . . . . .	10
5.1	部品表 . . . . .	21
5.2	スイッチと LED . . . . .	22
5.3	SV1: デジタル信号コネクタ 1 . . . . .	23
5.4	SV2: デジタル信号コネクタ 2 . . . . .	23
5.5	SV3: アナログ信号コネクタ . . . . .	23
5.6	SV4: 電源コネクタ . . . . .	24
5.7	SV5 . . . . .	24
5.8	SV6: デジタル信号コネクタ 3 . . . . .	24
5.9	CN4(SENSOR-PLUS) ピン配置 . . . . .	26
5.10	CN5(TFT ディスプレイ) ピン配置 . . . . .	27
5.11	CN6(OLED ディスプレイ) ピン配置 . . . . .	28

# 目次

2.1	SENSOR-PLUS . . . . .	7
2.2	TFT ディスプレイを搭載した RP2040-UNO . . . . .	8
2.3	OLED ディスプレイを搭載した RP2040-UNO . . . . .	9
3.1	ディスプレイのピンの加工 . . . . .	11
3.2	ディスプレイのピンの挿入 . . . . .	12
4.1	LED の点滅 . . . . .	15
4.2	端子に割り当てられた名前 . . . . .	16
4.3	BLINK:LED の単純な点滅 . . . . .	18
5.1	RP2040-UNO の回路図 . . . . .	20
5.2	RP2040-UNO の部品配置 . . . . .	22
5.3	SENSOR-PLUS . . . . .	26
5.4	TFT ディスプレイ . . . . .	27
5.5	OLED ディスプレイ . . . . .	29

# 第 1 章

## RP2040-UNO の紹介



### 1.1 製品概要

RP2040 を搭載した Raspberry Pi Pico がその高い性能と低コストが注目され、広く利用されるようになってきました。RP2040-UNO は、様々な開発や実験を手軽に行うために、Arduino UNO 形状のプリント基板に RP2040 を搭載した開発ボードとして開発されました。RP2040-UNO は RP2040 を使用し、様々な応用に取り組みたい人に最適な開発ボードです。

RP2040-UNO は以下のような特徴を持っています。

- RP2040 を Arduino UNO 形状の開発ボードに搭載しています。ただし、Arduino UNO と異なり、信号線の電圧は 3.3V です。5V の回路を接続しないようにしてください。
- Raspberry Pi Pico の 4 倍の 8MB のフラッシュを搭載しています。
- 機能と利便性を高めた RP2040-UNO 専用の MicroPython ファームウェアを使用できます。

- Arduino IDE でプログラム (スケッチ) を開発できます。
- WS2812B タイプのカラー LED を搭載しています。
- 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。
- RC サーボモーター 2 台かそのほかの外部モジュールを接続できる端子を装備しています。
- 使用頻度の高い入出力やセンサーをまとめた SENSOR-PLUS を接続する端子を装備しており、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。
- ドロップアウトが 300mV と少ない 1.3A \*<sup>1</sup>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。

## 1.2 購入・利用上の注意

RP2040-UNO をご購入の際には、下記項目をご確認ください。

- RP2040-UNO の端子は 3.3V の信号線となっており、5V の入力に対する耐性はありません。
- 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/>

## 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-UNO 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

---

\*<sup>1</sup> 放熱の制限で、継続的に 1.3A の電流を使用することはできません

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-UNO の特徴

### 2.1 Raspberry Pi Pico との相違点

RP2040-UNO 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-UNO のフラッシュメモリは 8MB です。
- RP2040-UNO 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-UNO 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-UNO では WS2812B カラー LED に接続されています。
- RP2040-UNO 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-UNO では引き出されていません。
- 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO への電力供給。  
USB からは 5V の電力が供給され、RP2040-UNO 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-UNO 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-UNO 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

ブレッドボードなど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-UNO 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-UNO で利用している電圧レギュレータ BL8071 は、少なくとも 1.3A 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、BL8071 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

---

\*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 1.3A の電流を使用することはできません。

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-UNO と接続されているシールドやブレッドボード上の回路などで消費電力の小さな回路に対しては、RP2040-UNO 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-UNO 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、DC ジャックあるいは SV4 の Vin 端子に 5V の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。
- USB コネクタで PC 等と接続された状態でも、DC プラグソケットあるいは SV4 の Vin 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 拡張端子

RP2040-UNO には通常の UNO タイプの開発ボードのピンソケットなどの他に、CN2-CN6 の拡張端子が装備されています。

- CN2, CN3  
RC サーボモータの接続端子
- CN4  
SENSOR-PLUS の接続端子
- CN5  
TFT ディスプレイの接続端子
- CN6  
OLED ディスプレイの接続端子

拡張端子の標準的な用途は上記の通りですが、電源端子や信号端子の機能や用途を十分に理解

していれば、これらの拡張端子を標準的な用途外に活用することもできます。

### 2.5.1 拡張端子への SENSOR-PLUS の接続

マイクロファンは、拡張端子 CN4 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。

製品紹介

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

ご購入

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

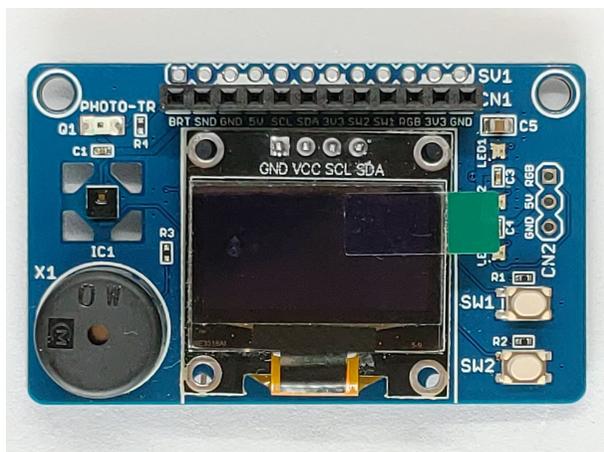


図 2.1 SENSOR-PLUS

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 CN4 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 CN4 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ようになります。

## 2.5.2 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-UNO の拡張端子 CN5, CN6 は、一般的な TFT ディスプレイや OLED 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-UNO 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できるようになります。

### CN5: TFT ディスプレイ

CN5 には図 2.2 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 2.2 TFT ディスプレイを搭載した RP2040-UNO

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。

### CN6: OLED ディスプレイ

CN6 には図 2.3 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 2.3 OLED ディスプレイを搭載した RP2040-UNO

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。

## 第3章

# 利用の準備

### 3.1 製品内容

RP2040-UNO の製品内容を表 3.1 に示します。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-UNO	Rev.1	1

### 3.2 RP2040-UNO の動作確認

RP2040-UNO 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-UNO 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 ディスプレイの取り付け

RP2040-UNO の CN5, CN6 には、TFT ディスプレイや OLED ディスプレイを取り付けて使用することができます

開発ボードへのディスプレイの取り付けは、基本的には、ディスプレイのピンを基板に直接はんだ付けするか、ピンソケットをまず基板の端子部分に取り付け、そのピンソケットにディスプレイのピンを接続するかの方法を取ります。

CN5, CN6 はピンソケットを取り付けることで、ディスプレイの取り付け以外の様々な用途への利用ができますので、端子にディスプレイを直接はんだ付けすることはお勧めしません。

CN5, CN6に取り付けるピンソケットが手元にはない場合には、必ずしも安定した接続法ではありませんが、次の節に示すようにディスプレイのピンを少し加工することにより、はんだ付けなしに基板に実用的に接続できるようになるので、必要に応じて試してみてください。

## 3.4 ディスプレイの簡易取り付け

### 3.4.1 概要と注意点

OLED ディスプレイや TFT ディスプレイのピンを少し加工することにより、多くの場合ははんだ付けせずに RP2040-UNO の CN5, CN6 端子にディスプレイを取り付けて利用できるようになります。

以下に示す方法は、少し変則的な方法になりますが、はんだ付けもピンソケットも必要のない便利な方法ですので、必要に応じてご活用ください。この方法では、端子をはんだ付けしていないので不必要な時にはディスプレイを取り外せます。

ただし、基板の端子のスルーホールはこのような使い方を想定して作られてはいないため、安定して使用できないことも考えられますし、さらには、ピンの挿抜を繰り返すと、接触不良やスルーホールの銅箔が剥離する可能性もあることをご承知<sup>\*1</sup>の上ご活用ください。

### 3.4.2 ピンの加工と基板への取り付け

具体的には、ディスプレイのピンの先端部分を図 3.1 に示すように、ピンの並びの中心線から交互に上下に少しずつずらすように、ピンを根元から少し曲げます。

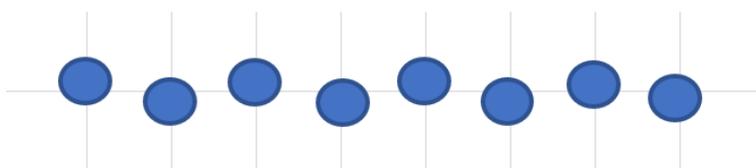


図 3.1 ディスプレイのピンの加工

上下への変位量が大きすぎると、CN5, CN6 にピンを差し込むことが難しくなりますし、変位量が小さいと、ピンがばねの効果でスルーホールの壁面に押し付けられて接触する力が弱くなり、接触不良となる可能性が高くなるので少し慣れや調整が必要です。

ピンを加工したディスプレイを基板の端子に取り付ける際には、図 3.2 に示すように、ピンの先が基板の端子の裏側に少し抜けた程度の位置で止めてください。ディスプレイのピンの根元まで基板の端子差し込むと、ピンの根元でばねを曲げて得られたばねの効果がないので、接触不良となる可能性が高くなります。

<sup>\*1</sup> 弊社はこの方法でディスプレイを接続し利用できることや、この方法で端子等に何らかの問題が発生しないことを推奨・保証するものではありません

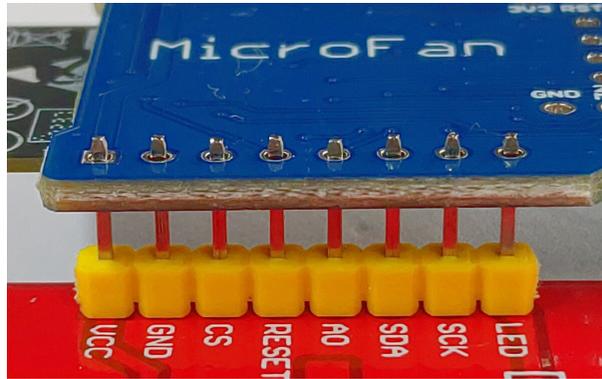


図3.2 ディスプレイのピンの挿入

前章の図2.2は、この様にして接続したTFTディスプレイに表示を行うスケッチを動かした例です。

ここで説明した例はTFTディスプレイですが、OLEDディスプレイも同様の手順で取り付けることができます。

## 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-UNO 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-UNO 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/](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 Arduino UNO 形状の RP2040-UNO を、端子名の指定などを含めて、UNO らしく使用することができません。

マイクロファンは、RP2040-UNO 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-UNO 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-UNO 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-UNO 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-UNO に接続する。あるいは、PC と RP2040-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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でご活用ください。

- <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-UNO 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, 'D13', 10 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'や'D13'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'や'D13'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-UNO と PC を USB で接続してください。

RP2040-UNO 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-UNO がブートローダーモードに移行します。

なお、RP2040-UNO 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-UNO 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 などでも確認してください。

面倒でなければ、以下の様な設定で、RP2040-UNO 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でも問題ありません。

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-UNO 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-UNO の LED は UNO の D13 に相当する 10 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 の定義を 10 に変更します。

```
#define LED_BUILTIN 10
// 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-UNO 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-UNO 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-UNO 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-UNO 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-UNO 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ので、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-UNO の回路図

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

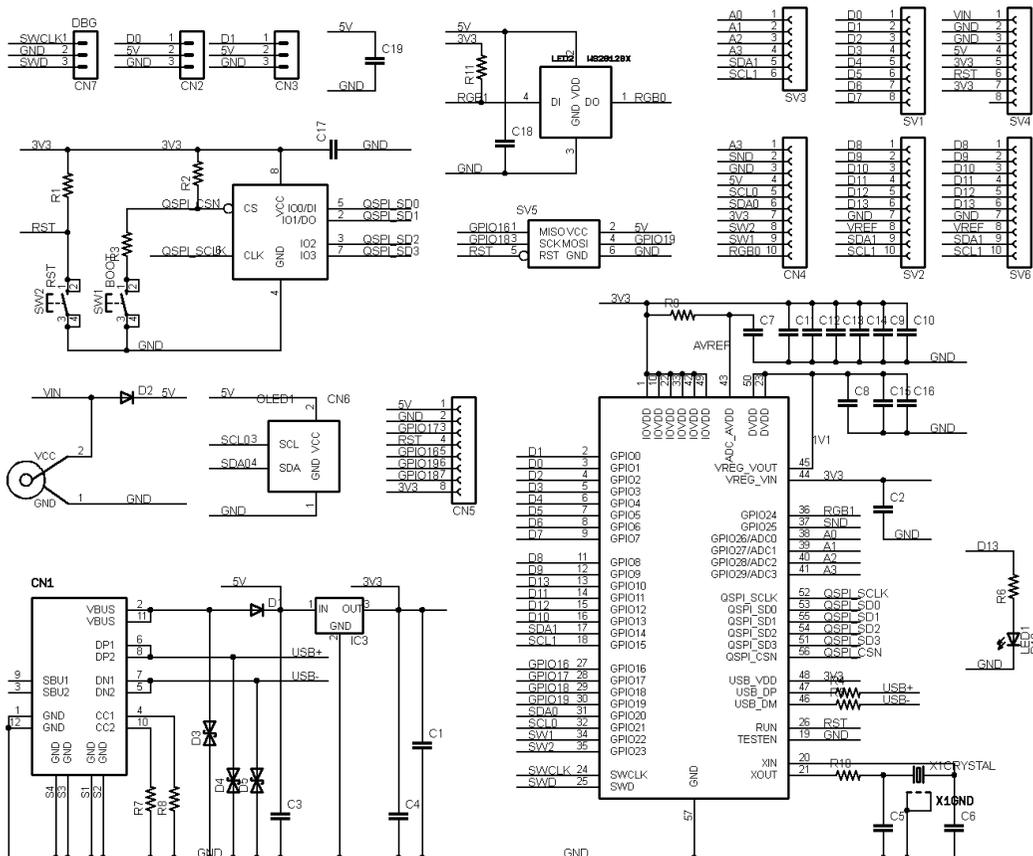


図 5.1 RP2040-UNO の回路図

表 5.1 部品表

部品	シンボル	規格等	個数
プリント基板	RP2040-UNO	Rev.1	1
IC	IC1	RP2040	1
	IC2	W25Q64JV	1
	IC3	BL8071	1
水晶	X1	12MHz	1
ショットキーダイオード ESD ダイオード	D1,D2	SS14	2
	D3,D4,D5	LESD5D5.0CT1G	3
発光ダイオード	LED1	青	1
	LED2	WS2812B	1
抵抗	R1, R2, R11	10K $\Omega$	3
	R3, R10	1K $\Omega$	2
	R4, R5	27 $\Omega$	2
	R6	470 $\Omega$	1
	R7, R8	5.1K $\Omega$	2
	R9	200 $\Omega$	1
セラミックコンデンサ	C1, C3, C19	10 $\mu$ F	3
	C2, C7, C8	2.2 $\mu$ F	3
	C4	47 $\mu$ F	1
	C5,C6	18pF	2
	C9-C18	0.1 $\mu$ F	10
タクトスイッチ	SW1, SW2	2 端子	2
USB	CN1	Type-C	1
ピンソケット	SV1, SV4	8 ピン	2
	SV2	10 ピン	1
	SV3	6 ピン	1
ピンヘッダー	SV5	2x3 ピン	1
	CN2, CN3	2x3 ピン	1
DC プラグ	PWR1		1

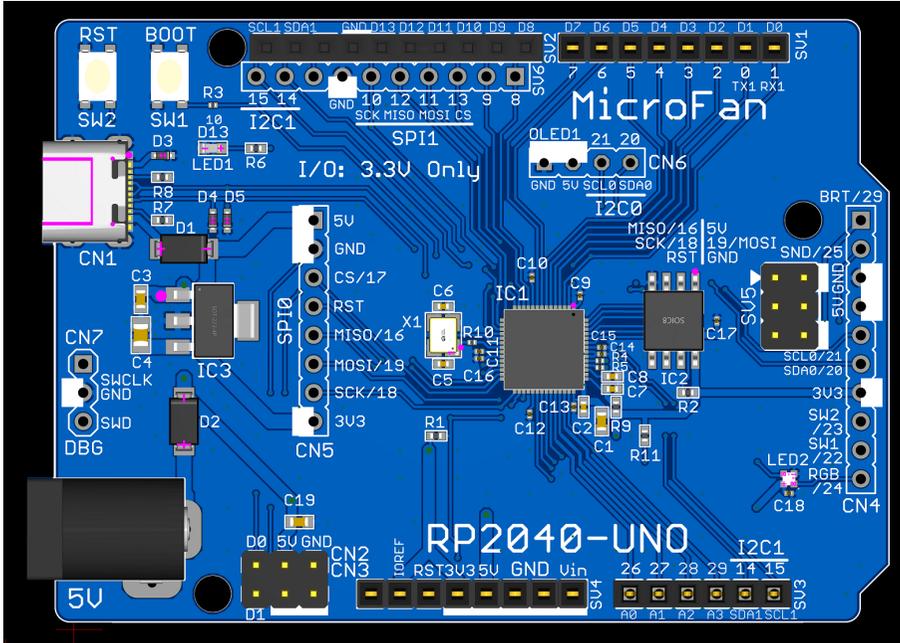


図 5.2 RP2040-UNO の部品配置

## 5.2 基板上的の入出力

RP2040-UNO の基板上的のスイッチと 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	D13, 0	正論理
LED2	RGB, 24	WS2812B

## 5.3 シールド等接続端子

ピンソケット SV1-SV6 の端子名等を表に示します。

端子名は、Arduino での一般的な端子の呼び名（一部異なります）、信号線番号は Arduino や MicroPython のプログラミングで使用できる信号線番号、信号線機能名は、マイクロファンの MicroPython ファームウェアで使用できる端子名を示しています。

表 5.3 SV1: デジタル信号コネクタ 1

ピン番号	端子名	信号線番号	信号線機能名	備考
1	D0	1	D0, GPIO1, GP1, RX1	
2	D1	0	D1, GPIO0, GP0, TX1	
3	D2	2	D2, GPIO2, GP2	
4	D3	3	D3, GPIO3, GP3	
5	D4	4	D4, GPIO4, GP4	
6	D5	5	D5, GPIO5, GP5	
7	D6	6	D6, GPIO6, GP6	
8	D7	7	D7, GPIO7, GP7	

表 5.4 SV2: デジタル信号コネクタ 2

ピン番号	端子名	信号線番号	信号線機能名	備考
1	D8	8	D8, GPIO8, GP8	
2	D9	9	D9, GPIO9, GP9	
3	D10	13	D10, GPIO13, GP13, SPI1_CS	
4	D11	11	D11, GPIO11, GP11, SPI1_MOSI	
5	D12	12	D12, GPIO12, GP12, SPI1_MISO	
6	D13	10	D13, GPIO10, GP10, SPI1_SCK	
7	GND			GND
8	-			無接続
9	SDA1	14	D18, GPIO14, GP14, SDA1	
10	SCL1	15	D19, GPIO15, GP15, SCL1	

表 5.5 SV3: アナログ信号コネクタ

ピン番号	端子名	信号線番号	信号線機能名	備考
1	A0	26	A0, D14, GPIO26, GP26, ADC0	ADC 可
2	A1	27	A1, D15, GPIO27, GP27, ADC1	ADC 可
3	A2	28	A2, D16, GPIO28, GP28, ADC2	ADC 可
4	A3	29	A3, D17, GPIO29, GP29, ADC3	ADC 可
5	SDA1	14	D18, GPIO14, GP14, SDA1	
6	SCL1	15	D19, GPIO15, GP15, SCL1	

表 5.6 SV4: 電源コネクタ

ピン番号	端子名	信号線番号	信号線機能名	備考
1	Vin	-		電源入力 (5V)
2	GND	-		
3	GND	-		
4	5V	-		電源出力 (5V)
5	3.3V	-		電源出力
6	RST	-		リセット
7	IOREF	-		MCU の信号線電圧 (3.3V)
8	-	-		無接続

表 5.7 SV5

ピン番号	端子名	信号線番号	信号線機能名	備考
1	MISO	16	SPI0_MISO, GPIO16, GP16	
2	5V	-		電源出力 (5V)
3	SCK	18	SPI0_SCK, GPIO18, GP18	
4	MOSI	19	SPI0_MOSI, GPIO19, GP19	
5	RST	-		リセット
6	GND	-		

SV6 は、SV2 の端子群を、Arduino のシールド専用のプリント基板ではなく、一般的な蛇の目基板で利用できるようにするための端子です。

表 5.8 SV6: デジタル信号コネクタ 3

ピン番号	端子名	信号線番号	信号線機能名	備考
1	D8	8	D8, GPIO8, GP8	
2	D9	9	D9, GPIO9, GP9	
3	D10	13	D10, GPIO13, GP13, SPI1_CS	
4	D11	11	D11, GPIO11, GP11, SPI1_MOSI	
5	D12	12	D12, GPIO12, GP12, SPI1_MISO	
6	D13	10	D13, GPIO10, GP10, SPI1_SCK	
7	GND	-		
8	-	-		無接続
9	SDA1	14	D18, GPIO14, GP14, SDA1	
10	SCL1	15	D19, GPIO15, GP15, SCL1	

## 5.4 拡張端子

S1-S4 のピンソケットとは別に、ディスプレイや外部拡張ボードを接続するための拡張端子 CN2-CN6 が配置されています。

### 5.4.1 CN2, CN3:RC サーボ

RC サーボを接続できる端子として、CN2, CN3 を装備しています。それぞれ、D0, D1 が接続されています。マイクロファンの MicroPython ファームウェアでは、D0, D1 にはそれぞれ SRV1, SRV2 の別名が付けられており、それらの名前指定して使用することができます。

RC サーボを操作するライブラリは、マイクロファンの MicroPython のファームウェアに組み込まれているので、簡単に RC サーボを制御することができます。

RC サーボの制御法は、下記の WEB ページをご参照ください。

- <https://www.microfan.jp/2023/10/micropython-rcservo-driver/>

D0, D1 は汎用的なデジタル入出力端子ですので、CN2, CN3 は、RC サーボの制御以外にも利用できます。ただし、CN2, CN3 の電源端子には、RC サーボでの利用を想定して、3.3V ではなく 5V が接続されている点に注意してください。

Arduino UNO では、D0, D1 は USB に接続された UART シリアル通信の端子となっており、基本的に PC との通信の障害にならないように、使用しないことが一般的でした。一方 RP2040 は、USB を介した PC とのシリアル通信は UART を介さず、USB 機能を直接利用しているため、D0, D1 は通常は用途がない状態となっています。このような状況を利用して、RP2040-UNO では D0, D1 を RC サーボの制御用に利用できるように、CN2, CN3 端子を用意しています。

### 5.4.2 CN4:SENSOR-PLUS

CN4 には、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/>

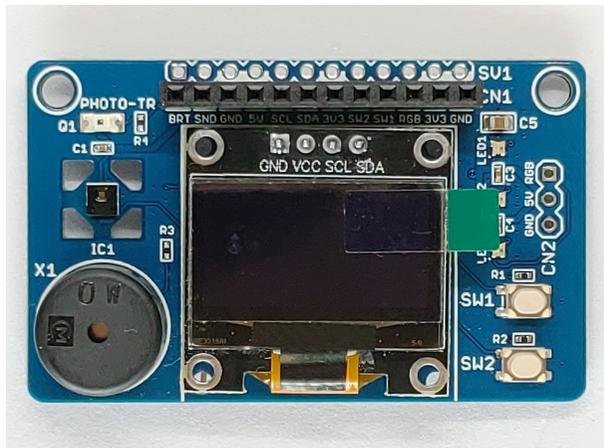


図 5.3 SENSOR-PLUS

表 5.9 CN4(SENSOR-PLUS) ピン配置

ピン番号	信号線番号	信号線機能名	備考
1	29	BRT, A3	明るさセンサー
2	25	SND	圧電スピーカー
3	GND	-	
4	5V	-	
5	21	SCL0	I2C0
6	20	SDA0	I2C0
7	3V3	-	
8	23	SW2	
9	22	SW1	
10	24	RGB	LED2 とダイジーチェーン

### 5.4.3 CN5:TFT ディスプレイ

CN5 のピン配置を表 5.10 に示します。

CN5 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。MicroPython では、CN5 の SPI は SPI(0) でインスタンス化できます。

表 5.10 CN5(TFT ディスプレイ) ピン配置

ピン番号	信号線番号	信号線機能名
1	5V	
2	GND	
3	17	SPI0_CS
4	RST	RESET
5	16	SPI0_DC
6	19	SPI0_MOSI
7	18	SPI0_SCK
8	3V3	

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<sup>\*1</sup>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。

<sup>\*1</sup> 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 5.10 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-UNO からの電源として 5V を供給しているため、モジュールに 3.3V の電圧レギュレータを内蔵していること。
- 使用するライブラリにもよりますが、コントローラに ST7735 を使用していること。

#### 5.4.4 CN6:OLED ディスプレイ

CN6 のピン配置を表 5.11 に示します。

CN6 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。MicroPython では、CN6 の I2C は I2C(0) でインスタンス化できます。

表 5.11 CN6(OLED ディスプレイ) ピン配置

ピン番号	信号線番号	信号線機能名
1	GND	
2	5V	
3	21	SCL0
4	20	SDA0

推奨する OLED ディスプレイを図 5.5 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 5.5 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 5.11 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-UNO 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-UNO 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。

## 第 6 章

# 購入および問い合わせ先

### 6.1 ご協力のお願い

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](mailto: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](mailto: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