

ESP32-SLIM-R1
(ESP-WROOM-32 開発ボード)
取扱説明書

マイクロファン

<https://store.shopping.yahoo.co.jp/microfan/>
<https://www.amazon.co.jp/s?me=A28NHPRKJDC95B>

2022 年 11 月
Copyright © 2022 MicroFan,
All Rights Reserved.

目次

第 1 章	ESP32-SLIM-R1 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	3
第 2 章	ESP32-SLIM-R1 の特徴	4
2.1	USB インターフェース	4
2.2	電源回路	4
2.2.1	電圧レギュレータ	4
2.2.2	外部への電力供給と注意点	5
2.2.3	外部からの電力供給と注意点	5
2.3	ブレッドボードの利用	5
2.4	ディスプレイの利用	6
2.4.1	OLED ディスプレイ	6
2.4.2	TFT ディスプレイ	7
第 3 章	利用の準備	9
3.1	部品表	9
3.2	ESP32-SLIM-R1 の動作確認	9
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	10
3.4.1	概要と注意点	10
3.4.2	ピンの加工と基板への取り付け	10
3.5	端子等のはんだ付け	11
3.5.1	半田ごての状態の管理	11
3.5.2	実装時のヒント	12
3.5.3	ピンヘッダー SV1, SV2	12
3.5.4	ディスプレイ CN2, CN3	12
第 4 章	Arduino スケッチ環境の整備	13
4.1	ESP32 用 Arduino 開発環境のインストール	13

4.1.1	基本となる Arduino IDE のインストール	13
4.1.2	ESP32 用の開発機能の追加	13
4.2	ESP32 用 Arduino 開発環境の設定	14
4.3	サンプルスケッチの実行	14
4.3.1	BLINK:LED の単純な点滅	14
4.3.2	スケッチのコンパイルと ESP32-SLIM-R1 への書き込み	14
4.4	OLED ディスプレイの利用	15
4.4.1	U8g2 ライブラリのインストール	15
4.4.2	U8g2 ライブラリを利用	16
4.5	TFT ディスプレイの利用	17
4.5.1	Adafruit-ST7735 ライブラリのインストール	17
4.5.2	Adafruit-ST7735 ライブラリを利用	17
第 5 章	資料	19
5.1	ESP32-SLIM-R1 の回路図	19
5.2	基板上の入出力	20
5.3	ブレッドボード用端子	21
5.4	ディスプレイ搭載用端子	22
5.4.1	OLED ディスプレイ	22
5.4.2	TFT ディスプレイ	23
第 6 章	購入および問い合わせ先	25
6.1	ご協力をお願い	25
6.2	販売：ネットショップ	25
6.3	製品情報	25
6.4	問い合わせ先	25
6.5	所在地	26

表目次

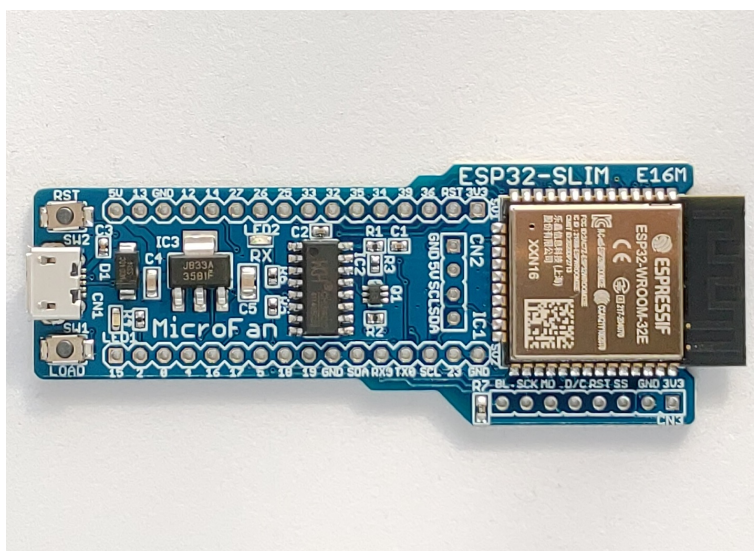
3.1	部品表	9
5.1	部品表	20
5.2	スイッチと LED	21
5.3	SV1,SV2 ピン配置	21
5.4	CN2(OLED ディスプレイ) ピン配置	22
5.5	CN3(TFT ディスプレイ) ピン配置	23

図目次

2.1	ブレッドボードに乗せた ESP32-SLIM-R1	6
2.2	OLED ディスプレイを搭載した ESP32-SLIM-R1	7
2.3	TFT ディスプレイを搭載した ESP32-SLIM-R1	7
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	11
4.1	BLINK:LED の単純な点滅	15
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	16
4.3	OLED ディスプレイ (SSD1306) 用のコンストラクタ	16
4.4	ライブラリマネージャを利用した Adafruit-ST7735 ライブラリの導入	17
4.5	端子指定の変更	18
4.6	端子指定の変更	18
5.1	ESP32-SLIM-R1 の回路図	19
5.2	ESP32-SLIM-R1 の部品配置	20
5.3	OLED ディスプレイ	22
5.4	TFT ディスプレイ	24

第 1 章

ESP32-SLIM-R1 の紹介



1.1 製品概要

近年様々なモノをインターネットに接続してサービスの高度化を図るモノのインターネット「IoT *¹」が注目されており、IoT サービスを実現するための様々な開発や実験が、企業はもちろん個人でも行われています。その IoT 装置を実現する中核部品として、WiFi や Bluetooth が標準装備された ESP-WROOM-32 が広く使用されています。

ESP32-SLIM-R1 は ESP-WROOM-32 を利用した IoT 機器の開発や実験を、ブレッドボード上で手軽に行うための開発ボードとして開発されました。ESP32-SLIM-R1 は Arduino の基本機能を一通り習得し、無線 LAN 機能を活用した応用に取り組みたい人に最適な開発ボードです。

ESP32-SLIM-R1 は以下のような特徴を持っています。

*¹ Internet of Things

- ESP32 の E バージョンの 16M 版を搭載しています。
- 高性能の 32 ビットマイクロプロセッサを搭載することで、Arduino UNO R3 などと比較して高速な処理が行えるとともに、大容量の FLASH(16M) と RAM を利用できます。
- ネットワークと接続するための WiFi や Bluetooth のネットワーク機能を利用できます。
- 電子工作で広く利用されている Arduino などの無償、便利、高機能な開発環境を利用してソフトウェアを開発できます。
- Arduino IDE で作成したスケッチを書き込むための USB インターフェースを装備しています。
- ドロップアウトが 300mV と少ない 1.5A ^{*2}の電圧レギュレータを搭載し、ESP-WROOM-32 に安定した電源を供給できます。
- ESP-WROOM-32 の信号線がピンヘッダーを取り付け可能な端子列に引き出されており、ブレッドボードに挿して利用できます。
- 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狭く（40 ピン DIP と同じ間隔）設計されています。
- 様々な情報を表示できる OLED ディスプレイ (別売) や TFT ディスプレイを搭載することができます。

1.2 購入・利用上の注意

ESP32-SLIM-R1 をご購入の際には、下記項目をご確認ください。

- ESP-WROOM-32 の未接続端子
内部のフラッシュメモリに接続されている ESP-WROOM-32 の 17-22 ピンは、使用上注意が必要なため未接続となっています。
- 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>
- ピンヘッダーは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/pin-header-40px2.html>
- ブレッドボードは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/breadboard-63.html>

^{*2} 放熱の制限で、継続的に 1.5A の電流を使用することはできません

1.3 マニュアルの記載内容に関して

ESP-WROOM-32 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、ESP32-SLIM-R1 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

ESP32-SLIM-R1 の特徴

2.1 USB インターフェース

Arduino IDE で作成したスケッチを ESP-WROOM-32 に書き込むために、USB インターフェース (microB) 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。

- ESP32-SLIM-R1 への電力供給。
USB からは 5V の電力が供給され、ESP32-SLIM-R1 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。
- ESP32-SLIM-R1 へのスケッチ (プログラム) の書き込み。
スケッチの書き込み時や、USB を介した受信時には、赤色の LED2 (RX) が点滅します。
- ESP32-SLIM-R1 と PC 間のシリアル通信。

2.2 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。ESP32-SLIM-R1 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。

ESP-WROOM-32 は WiFi 機能を稼働させる際に、突入電流として多くの電流を消費することが知られています。このため、電源が貧弱だと、ESP-WROOM-32 の動作が不安定になることがあります。

2.2.1 電圧レギュレータ

ESP-WROOM-32 は無線機能の利用時に 300mA 程度の電流を消費します。さらに、瞬間的ではありますが、突入電流として 1A 以上を消費することもあるようです。ESP32-SLIM-R1 で

利用している電圧レギュレータ BL8071 は、少なくとも 1.5A 以上の電流を供給^{*1}できますので ESP-WROOM-32 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 程度となります。ESP-WROOM-32 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。

2.2.2 外部への電力供給と注意点

ESP32-SLIM-R1 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、ESP32-SLIM-R1 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、ESP32-SLIM-R1 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。

2.2.3 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。

2.3 ブレッドボードの利用

ESP32-SLIM-R1 の 2 列の信号の引き出し端子は、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、他の ESP32 開発ボードと比較して狭い間隔（40 ピン DIP IC と同じ）で配置されています。他の ESP32 開発ボード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、ESP32-SLIM-R1 では図 2.1 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。

^{*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 1.5A の電流を使用することはできません。

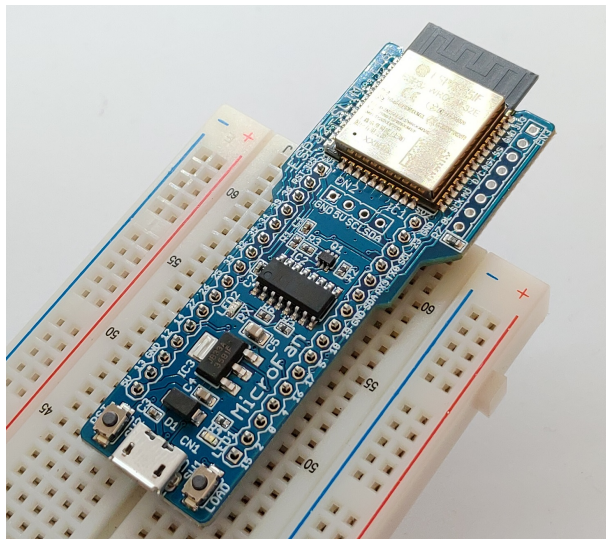


図 2.1 ブレッドボードに乗せた ESP32-SLIM-R1

2.4 ディスプレイの利用

ESP32-SLIM-R1 には OLED ディスプレイや TFT ディスプレイの接続端子が装備されているため、それらを基板に搭載して手軽に使用することができます。

ネット上などで公開されている ESP-WROOM-32 のサンプルスケッチでは、IP アドレスや様々な情報を PC 上でシリアルモニタに表示する例が多いですが、実際の運用では ESP-WROOM-32 を PC に接続して使用することは少ないため、運用時に必要な情報を確認することができないという問題があります。

ESP32-SLIM-R1 では、面倒な配線等を行うことなく開発ボード上に OLED ディスプレイや TFT ディスプレイを搭載できるため、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できます。

2.4.1 OLED ディスプレイ

ESP32-SLIM-R1 には OLED ディスプレイの接続端子が装備されているため、図 2.2 に示すように手軽に接続し、利用することができます。

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。

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。

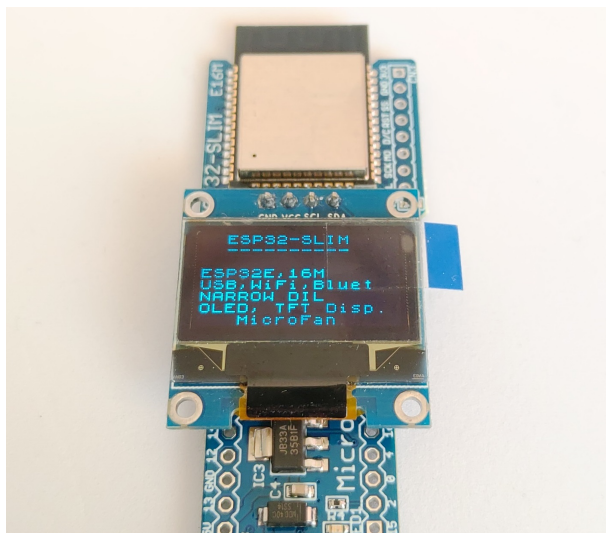


図 2.2 OLED ディスプレイを搭載した ESP32-SLIM-R1

2.4.2 TFT ディスプレイ

ESP32-SLIM-R1 には TFT ディスプレイの接続端子が装備されているため、図 2.3 に示すように手軽に接続し、利用することができます。

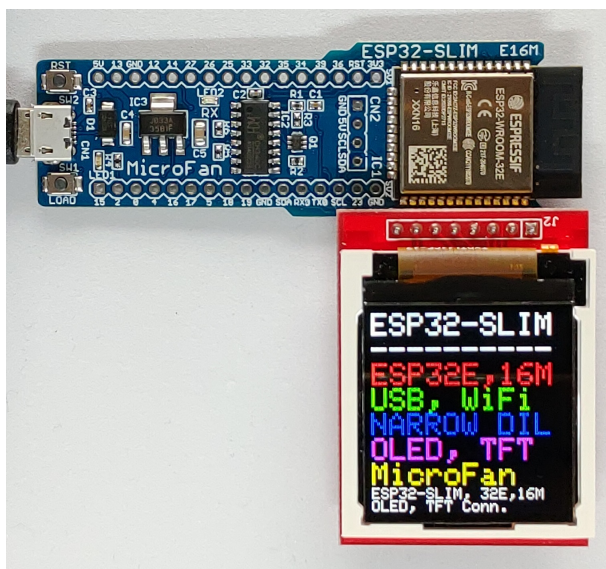


図 2.3 TFT ディスプレイを搭載した ESP32-SLIM-R1

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。

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、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。

第 3 章

利用の準備

ESP32-SLIM-R1 の利用に先立って、必要に応じて、ピンヘッダやコネクタ類のはんだ付けを行います。

ピンヘッダやコネクタ（ピンソケット）等は、必要に応じて別途ご入手ください。

3.1 部品表

ESP32-SLIM-R1 の部品表を表 3.1 に示します。表 3.1 には関連部品を含めて示していますが、ESP32-SLIM-R1 の基本的な商品構成は開発基板 1 つのみです。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 部品表

部品	シンボル	規格等	個数
プリント基板	ESP32-SLIM	Rev.1	1
OLED ディスプレイ	CN2	4 ピン	別売り
TFT ディスプレイ	CN3	8 ピン	別売り
ピンヘッダ	SV1, SV2	1x15PIN X2	別売り

3.2 ESP32-SLIM-R1 の動作確認

ESP32-SLIM-R1 は、製造時の基本的な動作確認として、LED1（青色）を点滅させるスケッチを書き込んで動作確認を行っています。

ESP32-SLIM-R1 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。LED1 が点滅し、ESP32-SLIM-R1 が稼働することが確認できます。

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

ESP32-SLIM-R1 には、OLED ディスプレイや TFT ディスプレイを取り付けて使用することができます

開発基板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。

ディスプレイのピンを直接基板にはんだ付けしても、ピンソケットをはんだ付けしても、基板が少しかさばるようになります。それが望ましくない場合には、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 ESP32-SLIM-R1 の CN2, CN3 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し、取り外したときには基板にピンソケットなどが残ってかさばることもありません。

ただし、基板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^{*1}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。

上下への変位量が大きすぎると、CN2, CN3 にピンを差し込むことが非常に難しくなりますし、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

^{*1} 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 3.1 ディスプレイのピンの加工

基板の端子差し込むと、ピンの根元ではピン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。

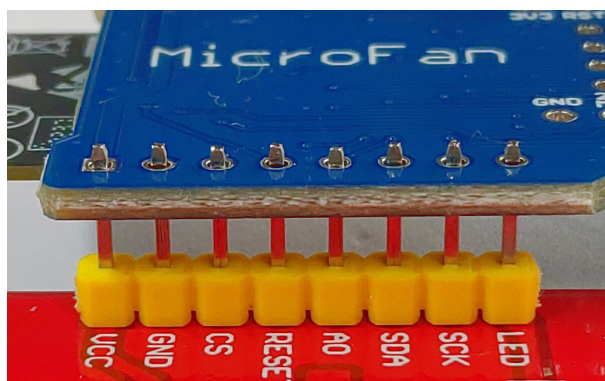


図 3.2 ディスプレイのピンの挿入

前章の図 2.3 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。

3.5 端子等のはんだ付け

3.5.1 半田ごての状態の管理

開発基板にピンヘッダ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。

3.5.2 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。

ESP32-SLIM-R1 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。

3.5.3 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にご注意ください。

3.5.4 ディスプレイ CN2, CN3

ディスプレイをピンソケットを介して ESP32-SLIM-R1 に接続する場合には、CN2 もしくは CN3 に 4 ピンもしくは 8 ピンのピンソケットを取り付けます。

ディスプレイを ESP32-SLIM-R1 に直接取り付ける場合には、OLED ディスプレイの場合には、裏側のコンデンサや抵抗の端子が基板上の部品に接触しないように、必要であれば少し浮かせた状態ではんだ付けするようご注意ください。

第 4 章

Arduino スケッチ環境の整備

4.1 ESP32 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
- ESP32 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。

4.1.1 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。

4.1.2 ESP32 用の開発機能の追加

ESP32 用の Arduino は以下の WEB ページで公開されています。

<https://github.com/espressif/arduino-esp32>

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 ESP32 用の Arduino のインストールを行ってください。

4.2 ESP32 用 Arduino 開発環境の設定

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 ESP32-SLIM-R1 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 esp32 の中から、初めの方に表示されている [ESP32 Dev Module] を選択してください。

また、[ポート:] は、ESP32-SLIM-R1 が接続されているポートが COMX(X は数値) の形式で表示されているので、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 などで確認してください。

最後に、フラッシュサイズに関連する以下の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大丈夫です。

基本的には、16M のフラッシュサイズに適合させます。

- Flash Size
16MB を選択
- Partition Scheme
2 種類ある [16M Flash (—)] のどちらかを選択

4.3 サンプルスケッチの実行

ESP32 用の Arduino をインストールすると、Arduino IDE の [ファイル] ⇒ [スケッチの例] に、ESP32 用の多くのサンプルスケッチが追加されます。これらのサンプルスケッチを試すことで、ESP32 のプログラミングを学ぶことができます。

ESP32-SLIM-R1 に初めから書き込まれている LED の点滅スケッチとかがありますが、ここでは、ESP32-SLIM-R1 の動作確認のために、LED 点滅スケッチの実行（更新）を試してみましょう。

4.3.1 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。ESP32-SLIM-R1 の LED は 2 番ピンに接続されているので、スケッチの `pinMode()`, `digitalWrite()` の第 1 引数の `LED_BUILTIN` を 2 に変更します。

4.3.2 スケッチのコンパイルと ESP32-SLIM-R1 への書き込み

スケッチの入力・修正が終わったら、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。

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(2, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(2, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

図 4.1 BLINK:LED の単純な点滅

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多数の行のメッセージが出て、スケッチの書き込みが開始されます。

この時、ESP32-SLIM-R1 の LED2:RX が赤色で点滅し、スケッチの書き込みのための受信が行われていることが示されます。

スケッチが ESP32-SLIM-R1 に正しく書き込まれたら、ボード上の LED が点滅します。

LED2:RX が点滅し、スケッチが更新されたことは確認できますが、先に書き込まれていたスケッチと同じなので、代り映えがせず本当にスケッチが更新されたか分かりにくいと思われる場合には、`delay()` の引数を変更して、LED の点滅の間隔などを変えてみるとよいでしょう。

4.4 OLED ディスプレイの利用

4.4.1 U8g2 ライブラリのインストール

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。この他にも、よく利用されるライブラリとして Adafruit の SSD1306 などがあります。

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。関連するライブラリも合わせてインストールするか聞かれた場合には、それらもすべてインストールしてください。

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>

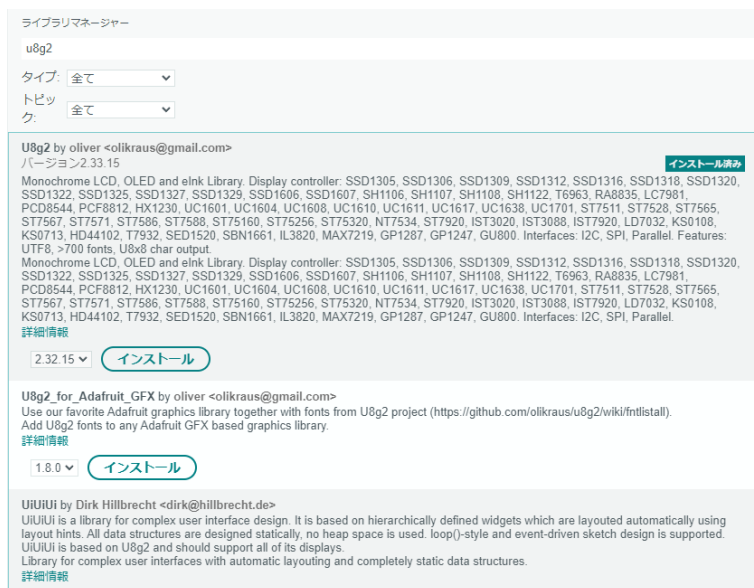


図 4.2 ライブラリマネージャを利用した U8g2 ライブラリの導入

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>

4.4.2 U8g2 ライブラリの利用

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、リストの下の方に 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。

例として [u8g2] ⇒ [full_buffer] ⇒ [GraphicsTest] を選択します。このスケッチは、デモ用の簡単な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 65 行あたりの、以下のコンストラクタを有効にしてください。

```
U8G2_SSD1306_128X64_NONAME_F_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

このコンストラクタを有効にすることにより、スケッチのコンパイル、実行ができるようになります。

4.5 TFT ディスプレイの利用

4.5.1 Adafruit-ST7735 ライブラリのインストール

TFT ディスプレイを利用するためのライブラリとして、Adafruit-ST7735 ライブラリを使用する例を示します。Adafruit-ST7735 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 [ST7735] を入力して絞り込むと、図 4.4 のように表示されます。

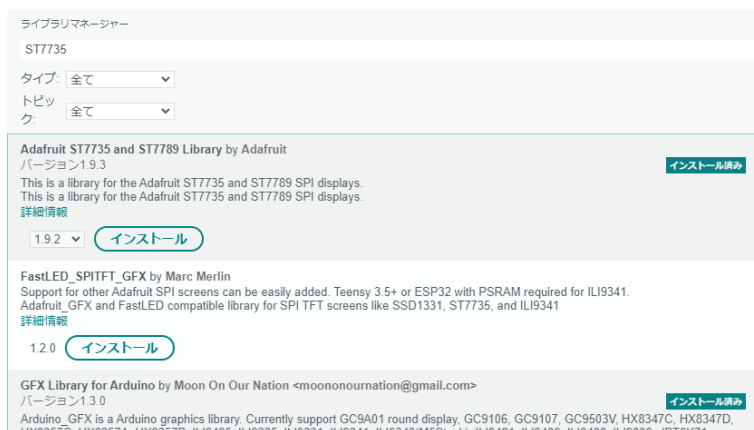


図 4.4 ライブラリマネージャを利用した Adafruit-ST7735 ライブラリの導入

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。関連するライブラリも合わせてインストールするか聞かれた場合には、それらもすべてインストールしてください。

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/adafruit/Adafruit-ST7735-Library>

4.5.2 Adafruit-ST7735 ライブラリの利用

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、リストの下の方に [Adafruit ST7735 and ST7789 Library] フォルダが追加されているのが確認できます。この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。

例として [Adafruit ST7735 and ST7789 Library] ⇒ [graphicstest] を選択します。このスケッチは、デモ用の簡単な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、ESP32-SLIM-R1 に合わせた端子の設定や、使用するディスプレイの設定、向きなどを変更する必要があります。

まず、スケッチの 53 行あたりの端子指定の内容を以下のように書き換えます。

```
// もともとの指定内容
#define TFT_CS      10
#define TFT_RST     9 // Or set to -1 and connect to Arduino RESET pin
#define TFT_DC      8

// ESP32-SLIM-R1 用の指定内容
#define TFT_CS      5
#define TFT_RST     -1
#define TFT_DC      17
```

図 4.5 端子指定の変更

次に、ここで利用する TFT ディスプレイは、コントローラとして ST7735 の GREENTAB 仕様のものを使用し、サイズが 1.44 インチなので、サンプルスケッチの 89 行あたりの初期設定のメソッド呼び出しを、少し下の行 (95 行) あたりにコメントアウトされているものに変更します。

```
// もともとの指定内容
tft.initR(INITR_BLACKTAB);      // Init ST7735S chip, black tab

// ESP32-SLIM-R1 用の指定内容
tft.initR(INITR_144GREENTAB); // Init ST7735R chip, green tab
tft.setRotation(2) ; // ディスプレイの表示向きを変更
```

図 4.6 端子指定の変更

さらに、`setRotation()` のメソッド呼び出しを追加しディスプレイの表示の向きを変更します。引数は、0-3 を指定できます。ご自身が TFT ディスプレイを使用する状況に合わせて、表示の向きを調整してください。

これらの設定で、TFT ディスプレイを使用できるようになります。

第 5 章

資料

5.1 ESP32-SLIM-R1 の回路図

ESP32-SLIM-R1 の回路図を図 5.1、部品表を表 5.1 に示します。

内部のフラッシュメモリに接続されている ESP-WROOM-32 の 17-22 ピンは、取扱に注意を要するため未接続となっています。

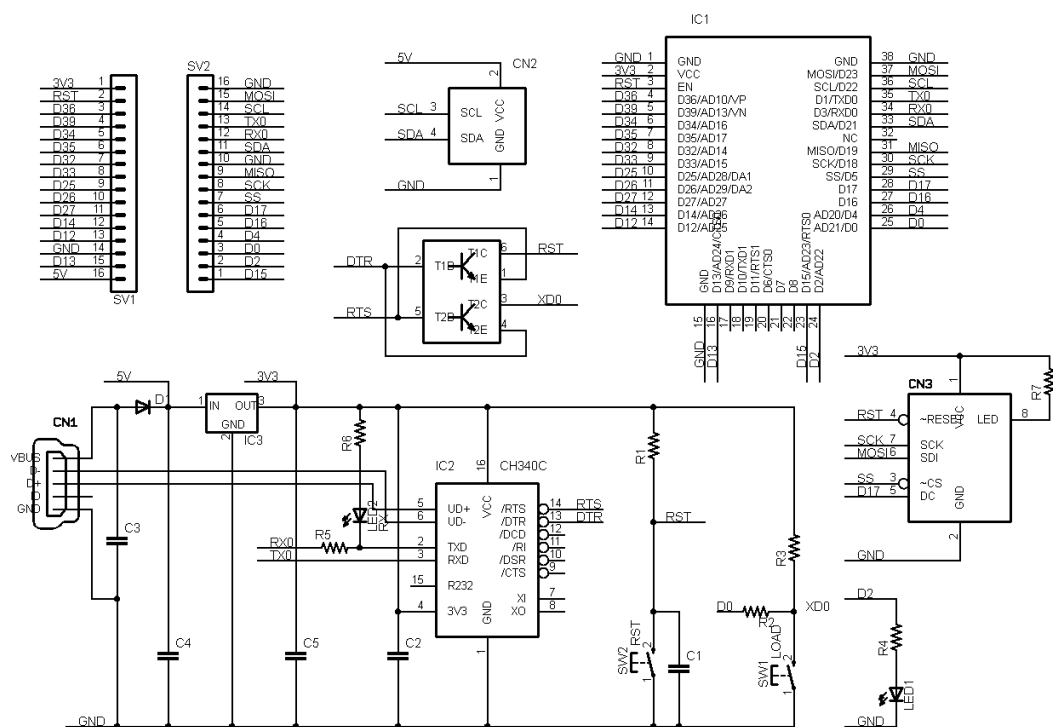


図 5.1 ESP32-SLIM-R1 の回路図

表 5.1 部品表

部品	シンボル	規格等	1
プリント基板	ESP32-SLIM	Rev.1	1
IC	IC1	ESP-WROOM-32(E, 16M)	1
	IC2	CH340C	1
	IC3	BL8071CLATR33	1
トランジスタ	Q1	UMH3N	1
ショットキーダイオード	D1	SS14	1
発光ダイオード	LED1	青	1
	LED2	赤	1
抵抗	R1, R3	10K Ω	2
	R2, R6	1K Ω	2
	R4, R5	470 Ω	2
	R7	4.7 Ω	1
セラミックコンデンサ	C1, C2, C3	0.1 μ F	3
	C4	10 μ F	1
	C5	47 μ F	1
タクトスイッチ	SW1, SW2	2 端子	2
USB	CN1	microB	1
OLED ディスプレイ	CN2	4 ピン	別売り
TFT ディスプレイ	CN3	8 ピン	別売り
ピンヘッダ	SV1, SV2	16PIN X2	別売り

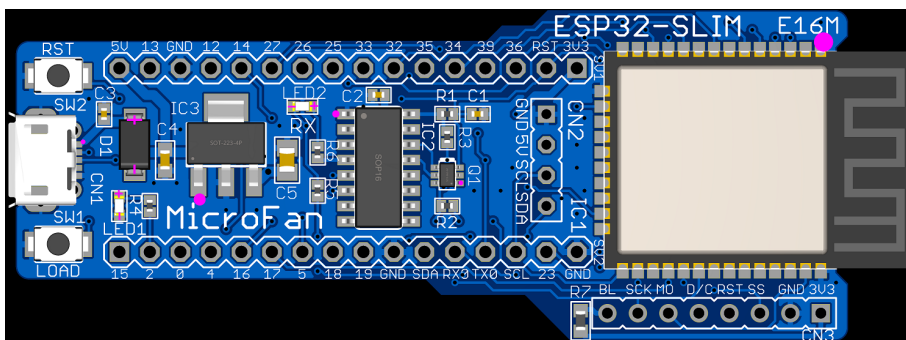


図 5.2 ESP32-SLIM-R1 の部品配置

5.2 基板上の入出力

ESP32-SLIM-R1 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（Arduino IDE などからのスケッチ書き込み）の切り替え用ですが、スケッチが走り始めたあとは、一般的な入力用のスイッチとして利用することができます。

表 5.2 スイッチと LED

シンボル	信号線	備考
SW1	D0	ブートモード移行用
SW2	EN	リセット用
LED1	D2	

5.3 ブレッドボード用端子

ESP32-SLIM-R1 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
	3.3V	16	GND	
EN	RST	15	D23	MOSI
VP/A0	D36	14	D22	SCL
VN/A3	D39	13	D1	TX0
A6	D34	12	D3	RX0
A7	D35	11	D21	SDA
A4	D32	10	GND	
A5	D33	9	D19	MISO
DA1/A18	D25	8	D18	SCK
DA2/A19	D26	7	D5	SS
A17	D27	6	D17	
A16	D14	5	D16	
A15	D12	4	D4	A10
	GND	3	D0	A11
A14	D13	2	D2	A12
	5V	1	D15	A13

5.4 ディスプレイ搭載用端子

5.4.1 OLED ディスプレイ

基板上に OLED ディスプレイを搭載するための CN2 端子を備えています。CN2 のピン配置を表 5.4 に、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>

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	GND	
2	5V	
3	D22	SCL
4	D21	SDA

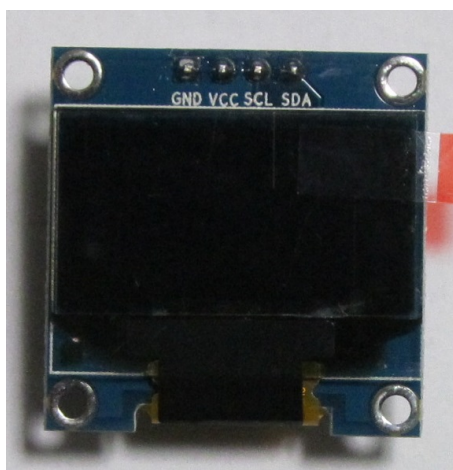


図 5.3 OLED ディスプレイ

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。
- ESP32-SLIM-R1 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。

- ESP32-SLIM-R1 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 TFT ディスプレイ

基板上に TFT ディスプレイを搭載するための CN3 端子を備えています。CN3 のピン配置を表 5.5 に、推奨する TFT ディスプレイを図??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>

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	VCC
2	GND	
3	D3	CS
4	RST	RESET
5	D17	DC
6	D23	MOSI
7	D18	SCK
8	3V3	LED

TFT 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 5.5 の順になっていること。

^{*1} 害害はないのですが、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。



図 5.4 TFT ディスプレイ

- 全ての信号線が 3.3V 対応であること。
- ESP32-SLIM-R1 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。

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 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