

ESP8266-LEAF-R5
利用マニュアル

マイクロファン

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

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

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

2017年11月

Copyright © 2015-2017 MicroFan,
All Rights Reserved.

目次

第 1 章	ESP8266-LEAF-R5 の紹介	1
1.1	製品概要	1
1.1.1	ESP8266-LEAF-R5 の特徴	2
1.1.2	R4 からの変更点	3
1.1.3	32 ビットプロセッサ	3
1.1.4	WiFi 無線ネットワーク装置	3
1.2	電源回路	4
1.3	搭載・接続可能なセンサーと入出力装置	4
1.3.1	赤外線 LED	5
1.3.2	明るさセンサー	5
1.3.3	OLED ディスプレイ	5
1.3.4	環境センサー	6
1.3.5	加速度・ジャイロセンサー	6
1.3.6	NeoPixel(WS2812)/RC サーボ	7
1.4	回路の試作と機能拡張	8
1.5	購入・作成上の注意	8
1.5.1	プログラミング用ソフトウェア	10
1.5.2	PC の動作環境	10
1.6	マニュアルの記載内容に関して	10
第 2 章	利用の準備	11
2.1	梱包内容	11
2.2	動作確認	12
2.2.1	USB ドライバおよびインターフェース	12
2.2.2	リセット動作の確認	12
2.2.3	ESP-WROOM-02 の動作確認	13
2.3	同梱部品の取り付け	14
2.3.1	はんだ付けを行う際の注意	15
2.3.2	明るさセンサー	15
2.3.3	赤外線発光ダイオード	16

2.3.4	ピンヘッダー CN4(WS2812/RC サーボ)	16
2.3.5	ピンヘッダー SV4, SV5	17
2.3.6	ピンソケット SV1, SV2, SV3	17
第 3 章	Arduino スケッチ環境の整備	19
3.1	Arduino Core	19
3.2	ESP8266 用 Arduino 開発環境のインストール	19
3.2.1	Arduino IDE のバージョン確認	20
3.2.2	ボードマネージャを使用する準備	20
3.2.3	ボードマネージャを使用した ESP8266 用 Core のインストール	20
3.2.4	ボードとパラメーターの選択	21
3.3	サンプルスケッチの実行	22
3.3.1	BLINK:LED の単純な点滅	23
3.3.2	FADE:PWM による LED の点滅	24
3.3.3	明るさセンサーの利用	24
3.3.4	赤外線 LED の利用	24
3.3.5	WiFi WebServer	26
3.3.6	OLED ディスプレイの利用	27
3.3.7	環境センサー BME280 の利用	29
3.3.8	MPU6050 の利用	31
3.3.9	カラー LED WS2812 の利用	31
3.3.10	RC サーボの利用	33
3.4	ESP8266-LEAF の応用例	34
第 4 章	資料	35
4.1	ESP8266-LEAF の回路図	35
4.2	BOOT モードの選択	36
4.2.1	BOOT モードの選択端子	37
4.3	NodeMCU 方式のフラッシュ書き込み機能	37
4.4	ハードウェア拡張用コネクタ	37
4.5	モジュール搭載用コネクタ	38
4.5.1	CN2: OLED ディスプレイ	38
4.5.2	CN3: BME280/MPU6050 モジュール	39
4.5.3	I2C 端子としての CN2-CN3 の活用	39
4.6	電源端子	40
4.7	電源レギュレータの電流容量と電圧	40
4.8	CH340 のデバイスドライバのインストール	40
第 5 章	購入および問い合わせ先	41

5.1	ご協力をお願い	41
5.2	販売：ネットショップ	41
5.3	製品情報	41
5.4	問い合わせ先	41
5.5	所在地	42

目次

1.1	ESP-WROOM-02	4
1.2	OLED ディスプレイ	5
1.3	OLED ディスプレイの搭載例	6
1.4	BME280 モジュール	6
1.5	環境センサーを搭載した ESP8266-LEAF	7
1.6	MPU6050 モジュール	7
1.7	加速度・ジャイロセンサーを搭載した ESP8266-LEAF	8
1.8	ブレッドボードへの接続	9
2.1	ESP8266-LEAF の梱包内容	11
2.2	シリアルモニタの出力	13
2.3	RST コマンド	14
2.4	GMR コマンド	15
2.5	同梱部品の取り付け例	15
2.6	ピンソケットの取り付け例	18
3.1	Board Manager を利用するための設定	21
3.2	ボードマネージャ	22
3.3	ESP8266 の構成	22
3.4	BLINK:LED の単純な点滅	23
3.5	FADE:PWM による LED の点滅	25
3.6	明るさセンサー値のシリアル出力	26
3.7	WiFi WebServer	27
3.8	OLED ディスプレイの搭載例	28
3.9	ライブラリマネージャを利用した U8g2 ライブラリの導入	28
3.10	OLED ディスプレイ (SSD1306) 用のコンストラクタ	29
3.11	環境センサーを搭載した ESP8266-LEAF	30
3.12	ライブラリマネージャを利用した BME280 ライブラリの導入	30
3.13	加速度・ジャイロセンサーを搭載した ESP8266-LEAF	31
3.14	カラー LED ライブラリ	32

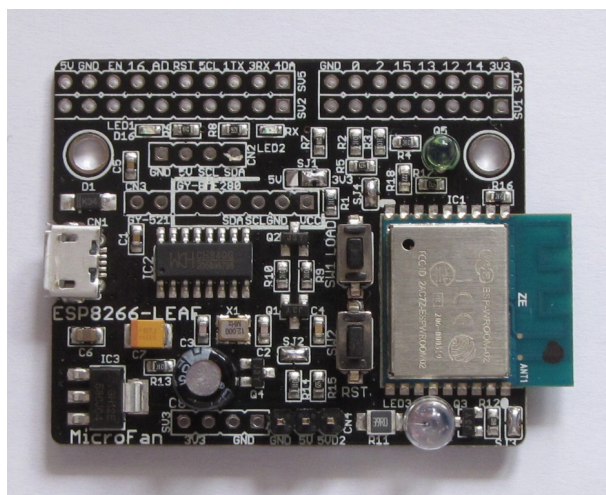
3.15	RC サーボの利用	33
4.1	ESP8266-LEAF の回路図	35

表目次

2.1	ESP8266-LEAF の梱包内容	11
4.1	部品表	36
4.2	BOOT モード	37
4.3	SV1,SV4 ピン配置	38
4.4	SV2,SV5 ピン配置	38
4.5	CN2(OLED ディスプレイ) ピン配置	39
4.6	CN3(BME280/MPU6050 モジュール) ピン配置	39

第 1 章

ESP8266-LEAF-R5 の紹介



1.1 製品概要

ESP8266-LEAF は ESP-WROOM-02 を単なる WiFi の無線 LAN 装置としてではなく、32 ビットプロセッサを搭載した無線 LAN 機能付の MCU として利用するために開発されたマイコンボードです。Arduino ボードなどの他の MCU なしに、このボードだけで無線 LAN 機能が付いた MCU のプログラミングを楽しむことができます。

また、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、ESP-WROOM-02 が注目されています。

ESP8266-LEAF は 家庭用の IoT 機器の開発や実験の中核となる機能を ESP-WROOM-02 と共にコンパクトな基板にまとめており、IoT 機器の試作や実験を行う人

*1 Internet of Things

が、高機能な IoT 装置を手軽に作成するためにも利用できます。

ESP8266-LEAF は Arduino の入門者にはもちろんですが、特に、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ボードです。

1.1.1 ESP8266-LEAF-R5 の特徴

ESP8266-LEAF は以下のような特徴を持っています。

ESP-WROOM-02 ボードとしての特徴

- 高性能の 32 ビットマイクロプロセッサを搭載しており、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 と RAM を利用可能
- ネットワークと接続するための WiFi ネットワーク機能を搭載
- 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可能
- USB インターフェースを装備しており、Arduino IDE で作成したスケッチを簡単に書き込むことが可能
- NodeMCU 方式の自動リセット・フラッシュ書き込み機能を搭載しています。
- 使用者の自由な発想でハードウェア的に機能拡張できるように、ESP-WROOM-02 の信号線が基板上部の一列のピンヘッダー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。
- 1A でドロップダウンが 0.45V の高性能電圧レギュレータを搭載し、ESP-WROOM-02 に安定した電源を供給可能

センサーの接続機能等の特徴

ESP8266-LEAF-R5 は、他の多くの ESP-WROOM-02 ボードと異なり、標準で様々なセンサーや出力装置を搭載・接続できるようになっており、単体で基本的な IoT 装置を構成できます。

- 家電機器をリモコン操作できる赤外線発光ダイオードを搭載
- 周囲の明るさを測ることができる明るさセンサー（フォトトランジスタ）を搭載
- 鮮やかな表示ができる WS2812 カラー LED や RC サーボの接続端子（信号の 5V 変換回路付）を装備
- 装置の周辺の温度、湿度、明るさなどを取得できる BME280 環境センサー（別売り）の搭載端子*2を装備
- 装置の加速度や回転などの動きを取得できる MPU6050 加速度・ジャイロセンサー（別売り）の搭載端子を装備
- 様々な情報を表示できる OLED ディスプレイ（別売り）の搭載端子を装備

*2 BME280 センサーと下記の MPU6050 センサーの搭載端子は共用

1.1.2 R4 からの変更点

前バージョンからの変更点は以下の通りです。

- BME280 環境センサーの取り付け位置を ESP-WROOM-02 の発熱の影響が少ない場所に変更。
ただし、ESP-WROOM-02 をスリープモードを併用するなどして発熱を抑えた運用をしないと、計測値は依然として ESP-WROOM-02 の発熱の影響を受けます。
- MPU6050 の端子は、BME280 の端子と共用に変更。
- フルカラーシリアル LED WS2812B と RC サーボを接続・制御できる 5V 信号出力の端子を追加。
- 明るさセンサー（フォトトランジスタ）を追加。
- 電圧レギュレータを電圧低下が少ない (0.45V@1A)AP2114H に変更して電源の安定度を向上。

1.1.3 32 ビットプロセッサ

ESP8266-LEAF の中核となる ESP-WROOM-02 には、高機能的な 32 ビットプロセッサと Arduino UNO R3 などと比較して大容量の Flash(プログラムメモリ)と RAM (データメモリ)が搭載されており、性能が要求されたり、ある程度のプログラムやデータ規模が要求される様々なアプリケーションに余裕をもって対応することができます。

ESP8266-LEAF で利用できる代表的なプログラミング環境/言語としては、下記のようなものがあります。

- Arduino IDE
- PlatformIO
- MicroPython
- NodeMCU

本文書の 3 章では、Arduino プログラミング環境のインストール法を説明しています。

1.1.4 WiFi 無線ネットワーク装置

ESP8266-LEAF の中核となる ESP-WROOM-02 モジュールを図 1.1 に示します。

無線装置としての日本国内での利用に必要な技適を取得しており、安心して利用することができます。ESP-WROOM-02 の製造元の WEB ページは以下の通りです。

<http://espressif.com/en/products/hardware/esp-wroom-02/overview>

ESP8266-LEAF の中核となる ESP-WROOM-02 には、WiFi の無線ネットワーク機能が標準で組み込まれています。従来の MCU をネットワークに接続しようとする、ネットワーク接続用の周辺機器を追加する必要がありましたが、ESP-WROOM-02 はそ



図 1.1 ESP-WROOM-02

れのみでネットワークに接続することができます。

また、ESP-WROOM-02 は、WiFi のクライアントとしてアクセスポイントに接続したり、自分自身がアクセスポイントになってスマートフォンなどの接続を受け入れることができます。

このため、ESP8266-LEAF は Arduino の入門者にはもちろんですが、特に、Arduino の基本機能を一通り習得し、無線機能を活用した応用に取り組みたい人に最適なボードです。

1.2 電源回路

ESP-WROOM-02 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-02 の動作が不安定になることがあります。

ESP8266-LEAF はドロップダウンが $0.45\text{V}@1\text{A}$ と小さく、1A 供給できる電源レギュレータを搭載し、ESP-WROOM-02 に十分な電流を供給して、安定して動作できるように設計されています。

1.3 搭載・接続可能なセンサーと入出力装置

ESP8266-LEAF-R5 は、他の多くの ESP-WROOM-02 ボードと異なり、標準で様々なセンサーや出力装置を搭載・接続できるようになっており、単体で基本的な IoT 装置を構成できます。

1.3.1 赤外線 LED

赤外線 LED を装備（同梱部品をはんだ付けする必要があります）しています。赤外線 の発光強度を上げるため、トランジスタを利用した赤外線 LED の駆動回路が組み込まれ ています。ESP8266-LEAF-R5 を IoT デバイスとして利用する際には、この赤外線 LED で様々な家電製品を制御することができます。

1.3.2 明るさセンサー

フォトトランジスターを使用した明るさセンサーを装備（同梱部品をはんだ付けす る必要があります）しています。ESP-WROOM-02 の AD 変換端子に接続されており、 ESP8266-LEAF-R5 周辺の明るさを読み取ることができます。

1.3.3 OLED ディスプレイ

図 1.2 に示す OLED ディスプレイ*³（別売り）を接続端子 CN2 に接続（図 1.3）する ことができます。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。

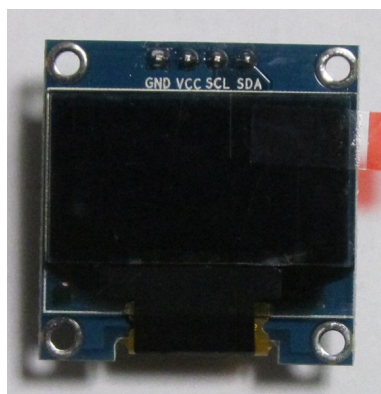


図 1.2 OLED ディスプレイ

WEB や書籍で公開されている ESP-WROOM-02 のサンプルスケッチでは、IP アド レスや様々な情報を PC の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-02 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。

ESP8266-LEAF では、基板上に OLED 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 OLED に表示し確認することができ

*³ <https://store.shopping.yahoo.co.jp/microfan/oled096-128x64-i2c-blue.html>

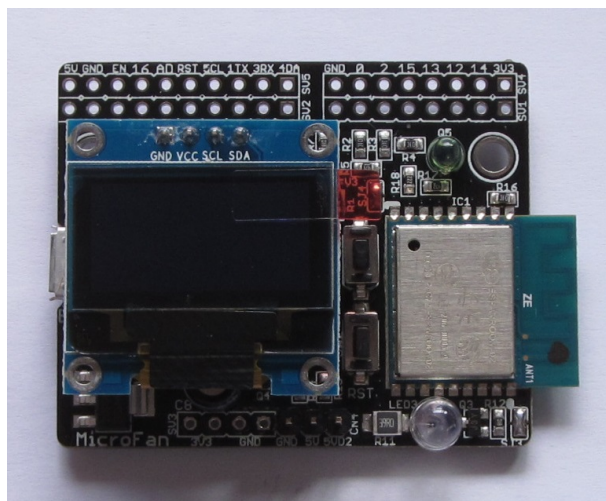


図 1.3 OLED ディスプレイの搭載例

ます。

1.3.4 環境センサー

図 1.4 に示す BME280 環境センサーモジュール*4 (別売り) を CN3 の接続端子に搭載 (図 1.5) することにより、温度、湿度、気圧を取得し、それらの環境情報をネットワークを通じて遠隔地の利用者に通知することができます。

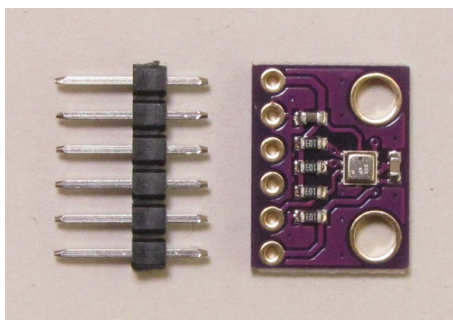


図 1.4 BME280 モジュール

1.3.5 加速度・ジャイロセンサー

図 1.6 に示す別売りの MPU6050 加速度・ジャイロセンサーモジュール*5 を CN3 の接続端子に搭載 (図 1.7) することにより、加速度・回転情報を取得し、それらの情報をネッ

*4 <https://store.shopping.yahoo.co.jp/microfan/gy-bme280.html>

*5 <https://store.shopping.yahoo.co.jp/microfan/gy-521.html>

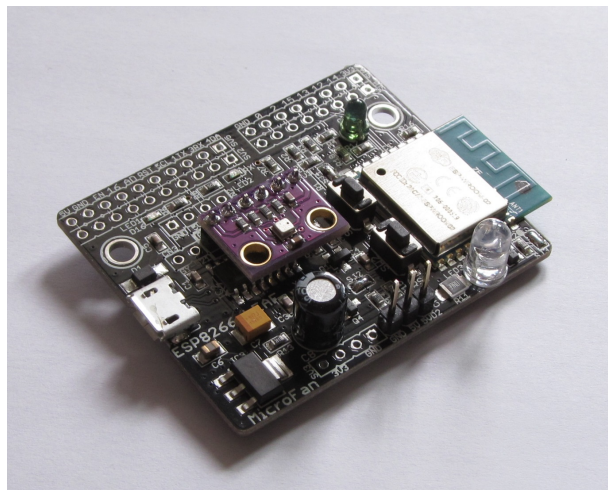


図 1.5 環境センサーを搭載した ESP8266-LEAF

トワークを通じて遠隔地の利用者に通知することができます。

なお、CN3 は BME280 と MPU6050 で共用です。

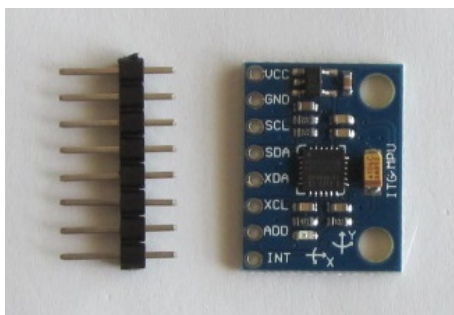


図 1.6 MPU6050 モジュール

1.3.6 NeoPixel(WS2812)/RC サーボ

NeoPixel(WS2812) カラー LED や RC サーボを接続できる CN4 端子を装備しています。

NeoPixel(WS2812) カラー LED や RC サーボは、制御用の信号の電圧が 5V 程度である必要があります。しかしながら ESP-WROOM-02 の信号線の電圧は 3.3V であるため、これらを接続しても安定して動かすことができません。CN4 端子には、信号線の出力電圧を 5V に変換する回路が組み込まれており、5V 程度の信号線を必要とする外部装置を安定して制御することができるようになっています。

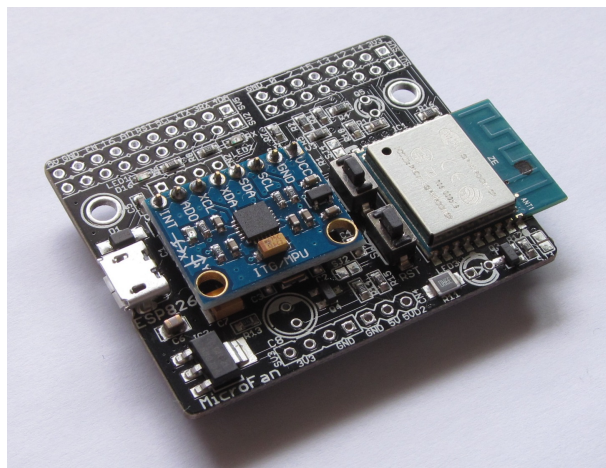


図 1.7 加速度・ジャイロセンサーを搭載した ESP8266-LEAF

1.4 回路の試作と機能拡張

開発者ごとに試作や開発したい IoT デバイスの機能は異なるため、画一的な一つの装置でそれらのすべてに対応することはできません。

ESP8266-LEAF は、IoT デバイス開発者の広範囲な要求に応えるために、3 種類の機能拡張法を用意しています。

- ディスプレイ、環境センサー、加速度・ジャイロセンサーなどの IoT デバイスに典型的な機能を拡張したい場合には、ESP8266-LEAF のボード上に接続端子が装備されているため、手軽に機能拡張を行うことができます。
- 拡張回路の開発などの際に試行錯誤をする場合には、基板の端に用意されている端子部分にヘッダーピンをはんだ付けし、図 1.8 に示すように ESP8266-LEAF をブレッドボードに接続して試作を行うことができます。
- 確定した回路で機能拡張する場合には、基板の端に用意されている端子部分にピンソケットをはんだ付けし、十分なサイズの拡張基板を接続して独自の高性能な IoT デバイスを作成することができます。

1.5 購入・作成上の注意

ESP8266-LEAF-R5 をご購入の際には、下記項目をご確認ください。

- 付属の明るさセンサー（フォトトランジスタ）、赤外線 LED、電解コンデンサー、ピンヘッダーは、基板にはんだ付けされていないので、各自の利用方法に合わせて取り付けてください。

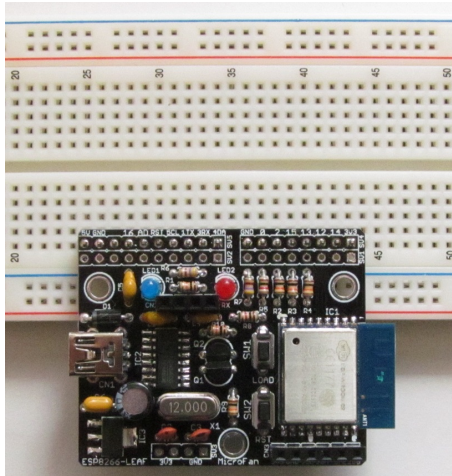


図 1.8 ブレッドボードへの接続

- ピンソケットは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/pin-socket-set-10844.html>
- 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/>
- BME280 センサーモジュールは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/gy-bme280.html>
- MPU6050 センサーモジュールは別売りです。
<https://store.shopping.yahoo.co.jp/microfan/gy-521.html>
- WS2812B/NeoPixel カラー LED は別売りです。
<https://store.shopping.yahoo.co.jp/microfan/ws2812-matrix-8x8.html>
<https://store.shopping.yahoo.co.jp/microfan/cjmcu-ws2812-ring-24.html>
<https://store.shopping.yahoo.co.jp/microfan/cjmcu-ws2812-ring-16.html>
<https://store.shopping.yahoo.co.jp/microfan/cjmcu-ws2812-ring-12.html>

1.5.1 プログラミング用ソフトウェア

ESP-WROOM-02 は標準的にはシリアル接続で AT コマンドで操作する無線 LAN 装置として構成されていますが、この文書では、Arduino IDE でプログラミングするための方法を示しています。

Arduino IDE や NodeMCU はオープンソースソフトウェアとして開発と提供が行われており、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、ESP8266-LEAF 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。

1.5.2 PC の動作環境

現時点では、ESP8266-LEAF 用の Arduino IDE は、Windows 7/10 上のみで動作確認を行っております。このソフトウェアは Windows の他のバージョンや MacOS や Linux 上でも問題なく稼働する*6と考えられますが、動作確認は行っていません。ESP8266-LEAF を MacOS や Linux 上の Arduino IDE で利用される場合には、各自の責任において動作確認をお願いいたします。

1.6 マニュアルの記載内容に関して

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。

*6 WEB 上に MacOS や Linux での ESP-WROOM-02/ESP8266 の利用例が多く報告されています。

第2章

利用の準備

2.1 梱包内容

ESP8266-LEAF の梱包内容を表 4.1、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 ESP8266-LEAF の梱包内容

部品	シンボル	規格等	個数
組み立て済基板	ESP8266-LEAF	Rev.5	1
明るさセンサー	Q5	フォトトランジスタ 3mm 灰緑色	1
赤外線発光ダイオード	LED3	5mm 透明	1
電解コンデンサー	C8	100 μ F	1
ピンヘッダ	SV4, SV5, CN4	1列 40PIN	1

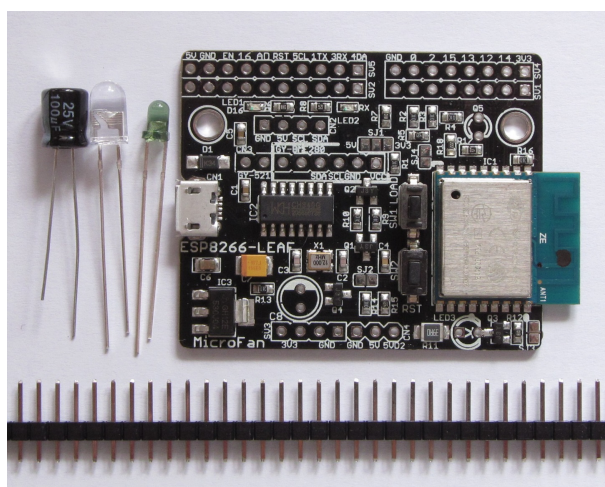


図 2.1 ESP8266-LEAF の梱包内容

2.2 動作確認

ESP8266-LEAF を入手したら、最初に USB ケーブルで PC に接続して動作確認を行ってください。

動作確認を行う前に、PC には CH340G のデバイスドライバーをインストール^{*1}しておく必要があります。ドライバーに関しては、4.8 節をご参照ください。

2.2.1 USB ドライバおよびインターフェース

一度 ESP8266-LEAF を PC から外して、Arduino IDE を起動して [ツール] ⇒ [シリアルポート] メニューを確認してください。COM1 など、ESP8266-LEAF を接続していない現時点で PC で利用できるシリアルポートが確認できます。

次に ESP8266-LEAF を PC に接続すると、正常な場合には USB デバイスとして認識されます。Arduino IDE で再度 [シリアルポート] メニューを確認すると、ESP8266-LEAF を接続していなかった際にはなかった COMX(X はポートに割り振られた数値)を確認できます。ESP8266-LEAF を利用するために、この COM ポートを選択します。

ESP8266-LEAF の USB インターフェースを認識できない、あるいは正常な USB デバイスではないなどと PC 画面に警告が出る場合には、製品に問題があるかもしれません。

製品以外の主な問題としては以下のようなものが考えられます。

- USB ケーブルの接続不良
- USB デバイスドライバの未導入もしくは失敗

問題が発生した場合には、以下のような対応をしてみてください。また、インターネット上で対応法を調査することも役立ちます。

- USB ケーブルの抜き差し
- USB ケーブルの交換
- USB デバイスドライバの再インストール
- PC の再起動
- 他の PC での確認

確認作業を行っても問題が解決されない場合には、マイクロファンに電子メール (microfan_shop@yahoo.co.jp) でお問い合わせください。

2.2.2 リセット動作の確認

ESP-WROOM-02 のリセット動作の確認を行います。

ESP8266-LEAF を PC に接続した状態で、Arduino IDE を起動し [ツール] ⇒ [ポー

^{*1} Windows 10 では、最初に PC に接続したときに自動的にインストールされます。

ト]メニューで ESP8266-LEAF に対応した適切な COM ポートを選択してください。

ウィンドウの右上方の虫メガネのようなアイコンを押して、シリアルモニタを表示してください。シリアルモニタウィンドウの右下で通信速度を設定できるので、115200bps に設定してください。

この後に、ESP8266-LEAF のリセットボタンを1度押してください。図 2.2 に示すように、シリアルモニタに意味のない文字列がうのようによと出力された後に ready と出力されます。

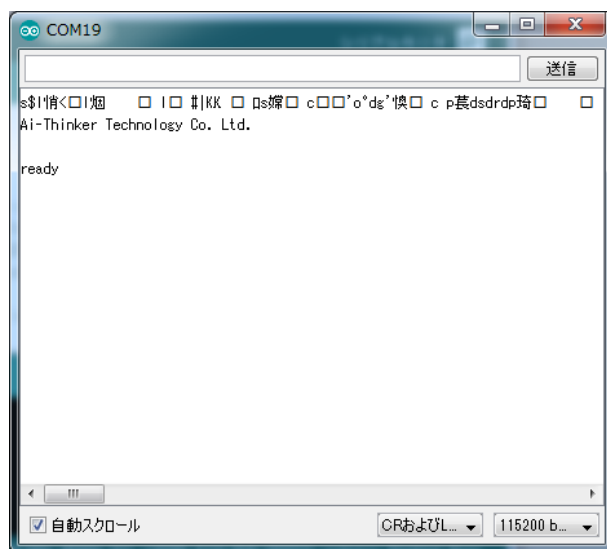


図 2.2 シリアルモニタの出力

これと似たような出力が得られれば、ESP-WROOM-02 と USB インターフェースが正常に機能していることが確認できます。

2.2.3 ESP-WROOM-02 の動作確認

次に、AT コマンドを利用して、ESP-WROOM-02 の無線モデムとしての動作確認を行います。

シリアルモニタの上部の入力エリアに、以下の文字列を入力して、それぞれ右側の送信ボタンを押してください。

- AT+RST
- AT+GMR

それぞれのコマンド入力に対し、図 2.3, 図 2.4 のような出力が得られれば、ESP8266-LEAF が PC から AT コマンドを受信できていることと、その応答処理ができていることが確認できます。

これまでの動作確認に問題がある場合には、再度動作確認を行った上で、作業内容と状

```
AT+RST

OK

ets Jan 8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 2408, room 16
tail 8
chksum 0xe5
load 0x3ffe8000, len 776, room 0
tail 8
chksum 0x84
load 0x3ffe8310, len 632, room 0
tail 8
chksum 0xd8
csum 0xd8

2nd boot version : 1.6
  SPI Speed      : 40MHz
  SPI Mode       : DIO
  SPI Flash Size & Map: 16Mbit(512KB+512KB)
jump to run user1 @ 1000

[***ここには文字化けした文字列が表示されています***]
ready
```

図 2.3 RST コマンド

況を取りまとめて電子メールでマイクロファンにお問い合わせください。

2.3 同梱部品の取り付け

ESP8266-LEAF は購入された状態で、Arduino IDE などでプログラミングを楽しむことができますが、必要に応じて同梱の部品や別売りのセンサー等をはんだ付けしてご利用ください。

未実装部品の取り付けは、ESP8266-LEAF の動作確認後に行ってください。

未実装部品の取り付け例を図 2.5 に示します。

```
AT+GMR
AT version:1.3.0.0(Oct 28 2016 11:29:39)
SDK version:2.0.0(6ccde98)
compile time:Mar 31 2017 14:50:03
OK
```

図 2.4 GMR コマンド

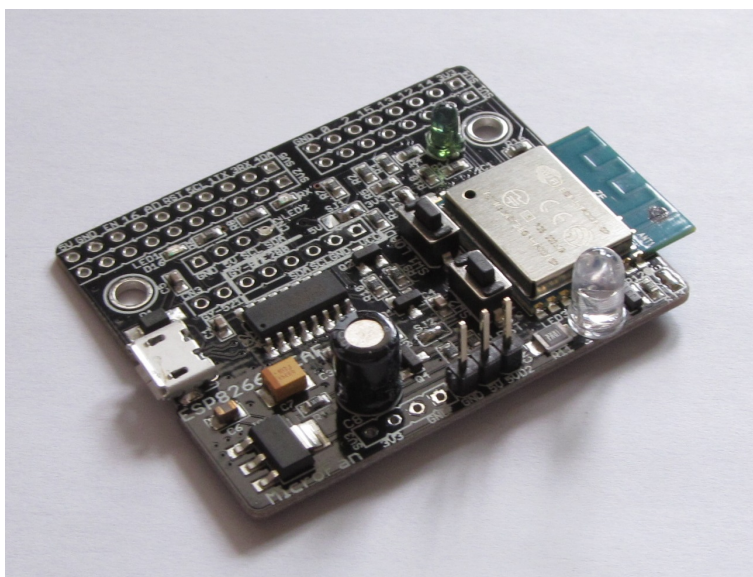


図 2.5 同梱部品の取り付け例

2.3.1 はんだ付けを行う際の注意

ESP8266-LEAF 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時間がかかるため、他の端子と比較して少し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。

2.3.2 明るさセンサー

取り付け

明るさセンサー（フォトトランジスタ）Q5 を所定の位置に取り付けはんだ付けします。

明るさセンサーの受光強度には指向性があるため、明るさセンサーの向きを計測したい発光源の方に向ける必要があります。このため、明るさセンサーを基板に取り付ける際には、利用法に合わせて足の長さや明るさセンサーの向きを調整してください。

明るさセンサーの端子には極性があり、長い脚のほうがアノード (+)、短い脚のほうがカソード (-) で、カソード側のプラスチックモールドの基部に切り欠きがあります。その切り欠きを基板のシルク印刷の切り欠きに合わせて取り付けはんだづけします。

明るさセンサーは、まず片方の足をはんだ付けし、部品の取り付け姿勢などを必要に応じて修正してからもう片方の足をはんだ付けすると、部品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、基板裏面から伸びて余っている足をニッパーなどで切り取ります。

ジャンパー SJ4

明るさセンサーは、ESP-WROOM-02 の AD 変換端子に接続して利用しますが、初期状態では、ジャンパー SJ4 で信号線が切り離されています。明るさセンサーを取り付け利用する場合には、SJ4 にはんだを盛ってショートさせ、AD 変換端子と明るさセンサーが接続されるようにしてください。

2.3.3 赤外線発光ダイオード

取り付け

赤外線発光ダイオード LED3 を所定の位置に取り付けはんだ付けします。

赤外線発光ダイオードの発光強度には指向性があるため、発光ダイオードの向きを操作する家電機器等に向ける必要があります。このため、赤外線発光ダイオードを基板に取り付ける際には、利用法に合わせて足の長さや発光ダイオードの向きを調整してください。

発光ダイオードの端子には極性があり、長い脚のほうがアノード (+)、短い脚のほうがカソード (-) で、カソード側のプラスチックモールドに切り欠きがあります。その切り欠きを基板のシルク印刷の切り欠きに合わせて取り付けはんだづけします。

発光ダイオードのはんだ付け法は、明るさセンサーのそれに準じます。

ジャンパー SJ3

赤外線発光ダイオードは、GPIO15 で制御しますが、初期状態では、ジャンパー SJ3 で信号線が切り離されています。赤外線発光ダイオードを取り付け利用する場合には、SJ3 にはんだを盛ってショートさせ、GPIO15 と赤外線発光ダイオードが接続されるようにしてください。

2.3.4 ピンヘッダー CN4(WS2812/RC サーボ)

取り付け

WS2812/RC サーボを使用する場合には、CN4 にピンヘッダーを取り付けます。

まず CN4 用に、3 ピンのピンヘッダーを、ピンヘッダーの台座部分の切れ込みをニッパーなどで挟んで切り離します。

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。

ジャンパー SJ2

CN4 は GPIO2 で制御しますが、初期状態ではジャンパー SJ2 で信号線が切り離されています。CN4 に外部装置を接続して利用する場合には、SJ2 にはんだを盛ってショートさせ、GPIO2 と CN4 が接続されるようにしてください。

電解コンデンサー C8

CN4 に WS2812 を接続して利用する場合、WS2812 への電流供給を安定化させるために電解コンデンサー C8 を取り付けます。

電解コンデンサーの端子には極性があり、電解コンデンサーの白い帯と、基板の三日月状の白いシルク印刷を合わせて取り付けはんだ付けします。

2.3.5 ピンヘッダー SV4, SV5

ESP8266-LEAF をブレッドボードに接続して利用する場合には、SV4, SV5 にピンヘッダーを取り付けます。

まず、8 ピンと 10 ピンのピンヘッダーを、ピンヘッダーの台座部分の切れ込みをニッパーなどで挟んで切り離します。

ピンヘッダーはブレッドボードに挿せるように、SV4, SV5 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。

2.3.6 ピンソケット SV1, SV2, SV3

ピンソケット SV1, SV2, SV3 は ESP8266-LEAF に拡張ボードを搭載する場合など、必要に応じてピンヘッダーを調達して取り付けてください。ピンソケットの取り付け例を図 2.6 に示します。

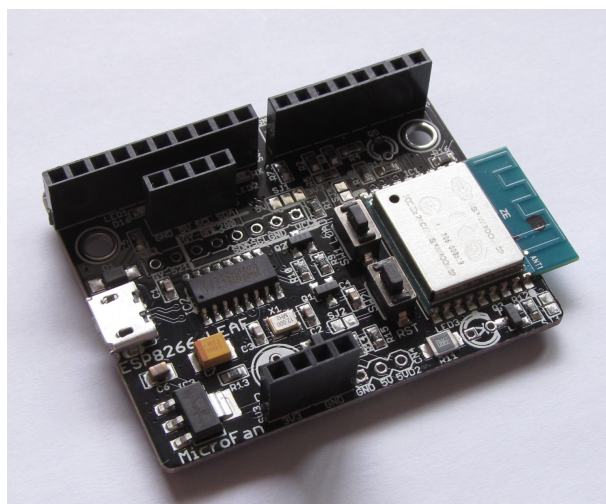


図 2.6 ピンソケットの取り付け例

第 3 章

Arduino スケッチ環境の整備

3.1 Arduino Core

ESP8266-LEAF のプログラム開発環境として、広く利用されている Arduino IDE を利用することができます。

標準的な Arduino ボード Arduino UNO R3 は MCU として ATmega328P を搭載しており、スケッチの作成環境である Arduino IDE には、ATmega328P 用のコンパイラ等が組み込まれています。Arduino には UNO 以外にも複数のボードがあり、それぞれに異なる MCU が搭載されています。Arduino IDE では、このような個別の MCU 用のコンパイラ等を [Core] と呼んでおり、ATmega328P 以外の MCU 用の Core をインストールできるようになっています。

Arduino IDE を利用して ESP-WROOM-02 用のスケッチを開発するためには、Arduino IDE に ESP8266 用の Core をインストールする必要があります。Core のインストール法は、下記の WEB ページに示されています。

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

Installing additional Arduino Cores

この手順に基づいた ESP8266 用 Core のインストール法を以下に示します。

3.2 ESP8266 用 Arduino 開発環境のインストール

ESP8266 用の Core は ESP8266 community によって開発され、以下の WEB ページで LGPL ライセンスで公開^{*1}されています。

<https://github.com/esp8266/Arduino>

^{*1} 高機能で利便性の高い Core を LGPL ライセンスで公開されている ESP8266 community に感謝いたします。

3.2.1 Arduino IDE のバージョン確認

このドキュメントの作成時点 (2017/11) では、<https://github.com/esp8266/Arduino> には、Arduino IDE の 1.8 より新しいものを利用するように示されています。

現状では Arduino のサイトで確認できる Arduino IDE の最新版は 1.8.5 となっていますので、1.8.5 の Arduino IDE をダウンロードしてインストールします。

なお、ESP8266 用 Core は改良が続けられており、対象となる Arduino IDE のバージョンも今後変わっていくものと考えられます。したがって、Arduino IDE の対象バージョンは ESP8266 用コアのページ <https://github.com/esp8266/Arduino> を参照して確認してください。

3.2.2 ボードマネージャを使用する準備

Arduino IDE に新しい Core をインストールするには、ボードマネージャを利用します。

Arduino IDE のメニューバーから、[ファイル] ⇒ [環境設定] を選択します。すると、図 3.1 に示すようなダイアログが表示されます。

<https://github.com/esp8266/Arduino> のインストール手順に示されているように、ダイアログの下方の [追加のボードマネージャの URL] の入力枠に下記の URL を入力し、ダイアログの右下の [OK] ボタンを押してダイアログを閉じます。

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

これで、ボードマネージャを利用して ESP8266 用 Core をインストールする準備ができました。

3.2.3 ボードマネージャを使用した ESP8266 用 Core のインストール

Arduino IDE のメニューバーから、[ツール] ⇒ [ボード : …] ⇒ [ボードマネージャ...] を選択します。すると、図 3.2 に示すようなダイアログが表示されます。

このダイアログのリストの中から、左上に [esp8266 by ESP8266 Community] と表示されている項目 (このドキュメントの記載時にはリストの最下部) をクリックして選択します。

項目を選択するとその項目の右下に [インストール] ボタンが表示されるので、それをクリックします。これにより ESP8266 用コアのインストールが開始され、インストールの進行状況を示すプログレスバーが表示されます。

インストールが完了したら、ダイアログ右下の [閉じる] ボタンを押してダイアログを閉じます。

ESP8266 用のコアがインストールされると、Arduino IDE のメニューバーから、[ツ

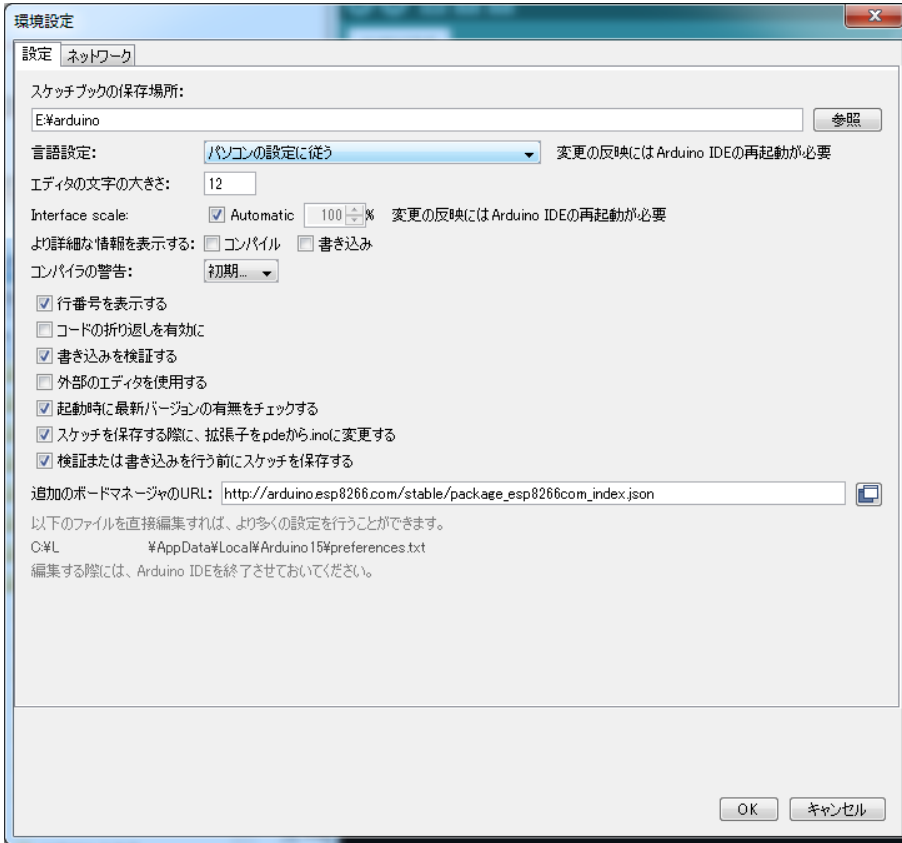


図 3.1 Board Manager を利用するための設定

ル] ⇒ [ボード: …] を選択します。すると、選択肢として開発対象のボードのリストが表示されます。そのリストの下の方に、複数の ESP8266 ボードの選択肢が表示され、ESP8266 用コアがインストールされていることが確認できます。

3.2.4 ボードとパラメーターの選択

メニューから [ツール] ⇒ [ボード: …] を選択します。すると、選択肢として開発対象のボードのリストが表示されます。そのリストの下の方から、[Generic ESP8266 Module] を選択します。

選択後、再度 [ツール] を選択すると、ESP8266 の構成を選択するための項目が追加されているのが確認できます。これらの項目の値を、図 3.3 に設定します。

特に、[Reset Method:] を "nodemcu" に設定し、[シリアルポート:] を ESP8266-LEAF のポート (一般的に COM1 以外) に設定することを忘れないでください。



図 3.2 ボードマネージャ



図 3.3 ESP8266 の構成

3.3 サンプルスケッチの実行

ESP8266用のCoreをインストールすると、Arduino IDEの[ファイル]⇒[スケッチの例]に、ESP8266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、ESP8266のプログラミングを学ぶことができます。

ここでは、それらのスケッチとは別に、ESP8266-LEAF の動作確認のために、いくつかのスケッチの実行を試してみましょう。

- 発光ダイオードの点滅
- 明るさセンサーの利用
- 赤外線発光ダイオードの利用
- WebServer
- OLED ディスプレイの利用
- 環境センサー BME280 の利用
- カラー LED WS2812/NeoPixel の利用
- RC サーボの利用

3.3.1 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。ESP8266-BLADE/LEAF の LED は 16 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 13 を 16 に変更します。

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(16, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(16, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(16, LOW); // turn the LED off by making the voltage LOW
  delay(1000);          // wait for a second
}
```

図 3.4 BLINK:LED の単純な点滅

スケッチのコンパイルとマイコンボードへの書き込み

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。

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矢印マーク [マイコンボードに書き込む] 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橙色の文字でメッセージが出て、ピリオド（点）が連続して表示され、スケッチの書き込みが開始されます。

スケッチが ESP8266-LEAF に正しく書き込まれたら、ボード上の LED が点滅します。

3.3.2 FADE:PWM による LED の点滅

Arduino IDE の [ファイル] ⇒ [スケッチの例] ⇒ [01.Basics] から Fade を選択してください。ESP8266-BLADE/LEAF の LED は 16 番ピンに接続されているので、スケッチの変数 led を 9 から 16 に変更します。

通常の Arduino の analogWrite() の第 2 引数で指定する PWM の強度は 0-255 ですが、ESP8266 の Arduino の実装の初期値では 0-1023 になっています。この上限の値 1023 は、analogWriteRange() を使用して変更することができるので、setup() の中で analogWriteRange(255) を実行して、PWM の強度の範囲を通常の Arduino に合わせます。

スケッチのコンパイルとマイコンボードへの書き込みは、3.3.1 節を参照してください。

スケッチが ESP8266-LEAF に正しく書き込まれたら、ボード上の LED が徐々に明るくなったり暗くなったりします。

3.3.3 明るさセンサーの利用

明るさセンサーを利用する場合には、2.3.2 節を参照して、まず基板への明るさセンサーの取り付けと、SJ4 のショート（結線）をしてください。

明るさセンサーの値は、analogRead(0) で簡単に読み取ることができます。analogRead(0) の値は、0~1023 で、暗いときに 0 となります。

スケッチのコンパイルとマイコンボードへの書き込みは、3.3.1 節を参照してください。

スケッチが ESP8266-LEAF に正しく書き込まれたら、シリアルモニタを開いてください。シリアルモニタに、明るさの計測値が出力されます。

3.3.4 赤外線 LED の利用

赤外線 LED を利用する場合には、2.3.3 節を参照して、まず基板への赤外線 LED の取り付けと、SJ1 のショート（結線）をしてください。

赤外線 LED は、IRremoteESP8266 ライブラリを使用すると手軽に利用できます。IRremoteESP8266 は、Arduino で広く利用されている IRremote を ESP8266 に移植したもので、以下の URL で公開されています。

<https://github.com/markszabo/IRremoteESP8266>

ライブラリのインストール

IRremoteESP8266 は、ライブラリマネージャを利用して簡単にインストールできます。

Arduino IDE のメニューから、[スケッチ] => [ライブラリをインクルード] => [ライブ

```
int led = 16;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 16 to be an output:
  pinMode(led, OUTPUT);

  analogWriteRange(255) ; // for ESP8266
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 16:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

図 3.5 FADE:PWM による LED の点滅

ラリを管理...] を選択すると、ライブラリマネージャのダイアログウィンドウが表示され、インストール可能なライブラリの一覧が表示されます。この中から、IRremoteESP8266 を選択すると、リストの右下に [インストール] ボタンが表示されるのでそれをクリックします。これで、ライブラリがインストールされ、スケッチで利用できるようになります。

ライブラリの利用

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 IRremoteESP8266 フォルダが追加されているのが確認できます。IRremoteESP8266 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。


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

void loop(void)
{
  Serial.println(analogRead(0)); // 明るさセンサーの値

  delay(1000);
}
```

図 3.6 明るさセンサー値のシリアル出力

赤外線 LED でリモコンコードを送信する場合には、赤外線 LED が割り当てられている 15 番ピンを指定して `irsend(15)` コンストラクタを使用してスケッチを作成します。

3.3.5 WiFi WebServer

ESP8266 Core のサンプルスケッチの中に、いくつかの Web サーバーの例があります。それらはまず ESP-WROOM-02 を既存の WiFi のアクセスポイントに接続させておいて、クライアントはそのアクセスポイント経由で ESP-WROOM-02 の Web サーバーにアクセスするようになっています。

ここでは、ESP-WROOM-02 をアクセスポイントに設定して、スマートフォンなどで直接 ESP-WROOM-02 に接続して利用する例を示します。

- SSID ESP8266
- パスワードなし

スケッチのコンパイルとマイコンボードへの書き込みは、3.3.1 節を参照してください。ESP8266-LEAF にスケッチを書き込み、Arduino IDE のシリアルモニタを開くと ESP-WROOM-02 で動作している WebServer のアドレスが表示されます。

スマートフォンなどで、まず ESP-WROOM-02 のアクセスポイント（この場合は SSID:ESP8266）に WiFi 接続します。この次に、接続したスマートフォンなどの WEB ブラウザで、シリアルモニタに表示された WebServer のアドレスに接続すると、[Hello MicroFan!] と書かれた WEB ページが表示されます。

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

const char ssid[] = "ESP8266";

ESP8266WebServer server(80);

void handleRoot()
{
  String msg = "<html><head><title>ESP8266 WebSerber</title>"
              "</head><body><h1>Hello MicroFan!</h1>" ;

  server.send(200, "text/html", msg + "<hr>"
              "<h1>" + millis() + "</h1></body></html>");
}

void setup()
{
  WiFi.softAP(ssid); // パスワードなし
  server.on("/", handleRoot);
  server.begin();

  Serial.begin(115200);
  Serial.println();
  Serial.print("WebServer IP address: ");
  Serial.println(WiFi.softAPIP());
}

void loop()
{
  server.handleClient();
}
```

図 3.7 WiFi WebServer

3.3.6 OLED ディスプレイの利用

ESP8266-LEAF は、多様な情報の表示装置として OLED ディスプレイを搭載することができます。基板上に OLED ディスプレイを搭載した例を図 3.8 に示します。

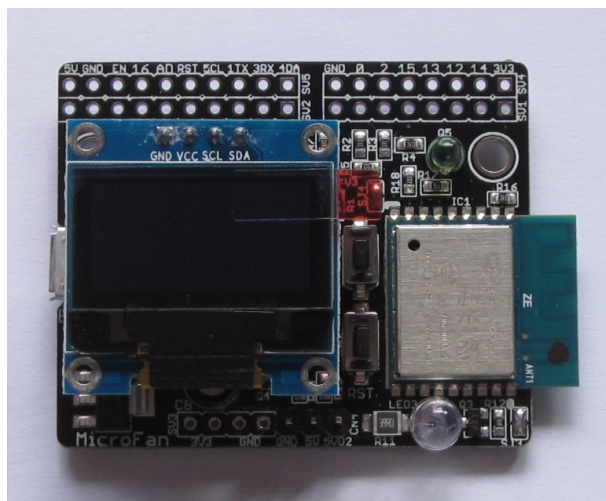


図 3.8 OLED ディスプレイの搭載例

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。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 3.9 のように表示されます。

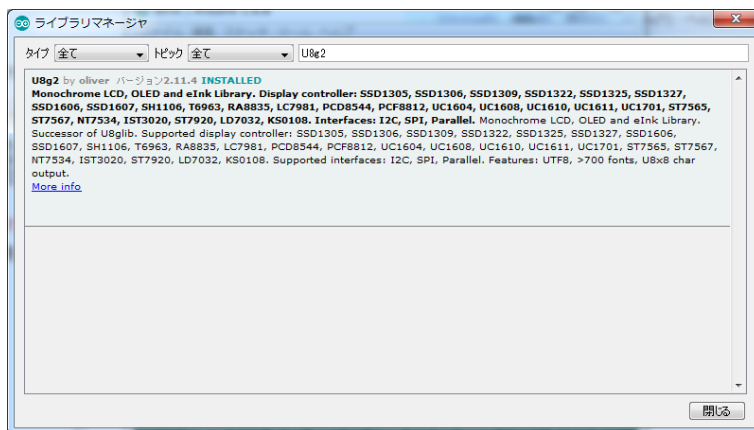


図 3.9 ライブラリマネージャを利用した U8g2 ライブラリの導入

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。

なお、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_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);
```

図 3.10 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の追加により、スケッチのコンパイル、実行ができるようになります。

3.3.7 環境センサー BME280 の利用

ESP8266-LEAF は、温度、湿度、気圧を測る環境センサーとして、BME280 モジュールを CN3 に搭載することができます。BME280 モジュールを利用する場合には、モジュールに 3.3V 電源を供給するために、SJ1 の 3.3V 側をはんだでショートさせるようにしてください。ESP8266-LEAF に BME280 モジュールを搭載した例を図 3.11 に示します。

ライブラリのインストール

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。

今回は、[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

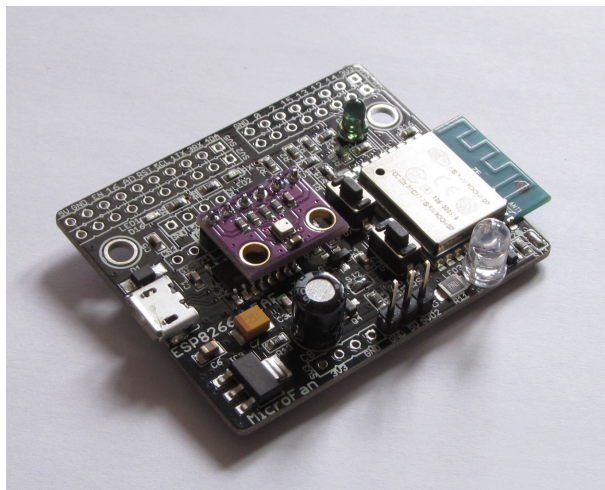


図 3.11 環境センサーを搭載した ESP8266-LEAF

インストールします。

なお、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>

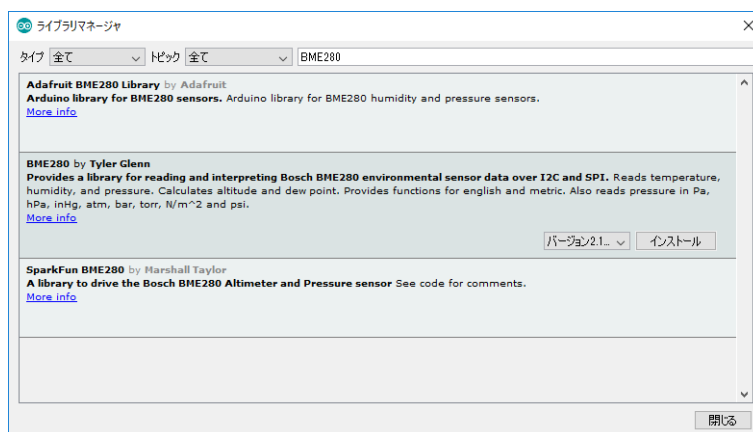


図 3.12 ライブラリマネージャを利用した BME280 ライブラリの導入

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。

ESP8266-LEAF 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。

3.3.8 MPU6050 の利用

ESP8266-LEAF は、加速度・ジャイロセンサーとして、CN3 に MPU6050 モジュールを搭載することができます。ESP8266-LEAF に MPU6050 モジュールを搭載した例を図 3.13 に示します。

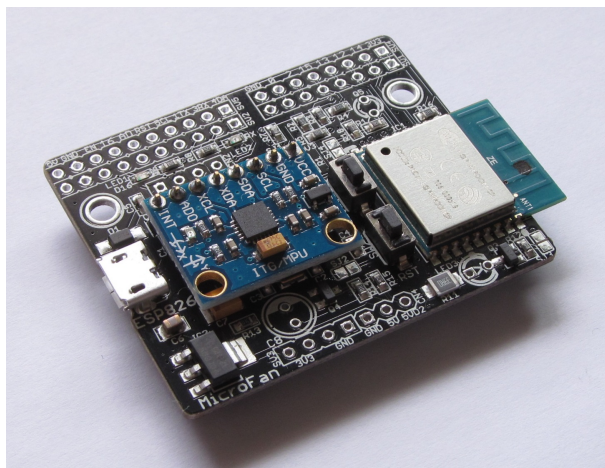


図 3.13 加速度・ジャイロセンサーを搭載した ESP8266-LEAF

MPU6050 モジュールを利用する場合には、モジュールに 5V 電源を供給するために、SJ1 の 5V 側をはんだでショートさせるようにしてください。

MPU6050 用に使えるライブラリの例を以下に示します。

<https://www.i2cdevlib.com/devices/mpu6050>

3.3.9 カラー LED WS2812 の利用

WS2812/NeoPixel は CN4 に接続して利用することができます。WS2812/NeoPixel を使用する場合には、電解コンデンサ C8 も基板に取り付けてください。

ESP8266-LEAF で利用できるカラー LED:WS2812B には、LED の発光制御用の IC が組み込まれています。この IC のおかげで、カラー LED の色や明るさを 1 本の信号線

で制御できるのですが、その制御信号として、高速かつ正確なタイミングの制御信号を生成する必要があります。このようなプログラムを作成するためには、MCU のマシン語を用いたプログラミングの知識なども必要になり少々面倒なのですが、Adafruit 社がオープンソースライセンスで公開している NeoPixel ライブラリを利用すると、WS2812B を簡単に制御することができるようになります。

NeoPixel ライブラリは以下の URL で公開されています。

https://github.com/adafruit/Adafruit_NeoPixel

Adafruit の NeoPixel に関する Web ページは次のとおりです。

<https://learn.adafruit.com/adafruit-neopixel-uberguide/overview>

ライブラリのインストール

NeoPixel ライブラリは、Arduino IDE のライブラリ管理機能を利用して簡単にインストールすることができます。

Arduino IDE のメニューバーから、[スケッチ] ⇒ [ライブラリをインクルード] ⇒ [ライブラリを管理...] を選択します。図 3.14 に示すような [ライブラリマネージャ] ダイアログが開きますので、[Adafruit NeoPixel Library] の項目を見つけてクリックします。なお、項目をスクロールして見つけるのが面倒であれば、ダイアログの右上の [検索をフィルタ...] と表示されている入力枠に [NeoPixel] と入力すると、候補が絞られます。項目の右下に [インストール] ボタンが出てきますので、それをクリックするとライブラリがインストールされます。

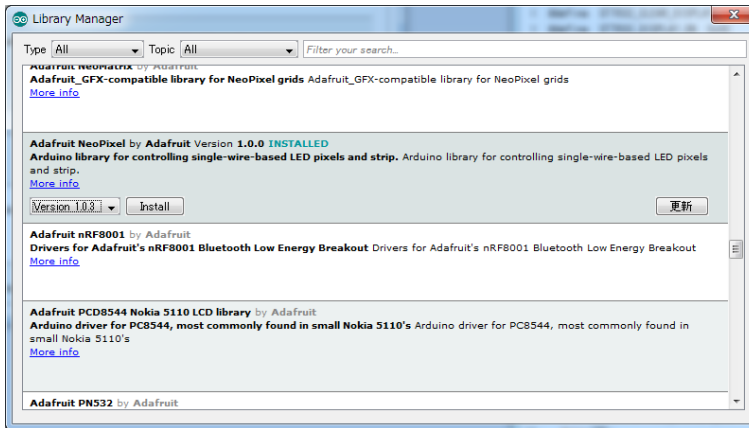


図 3.14 カラー LED ライブラリ

ライブラリの利用

ライブラリがインストールされると、メニューバーの [ファイル] ⇒ [スケッチの例...] を選択すると、インストールされた NeoPixel ライブラリに関する項目が追加されていることがわかります。その項目を選択すると、いくつかのデモプログラムが表示されます。

ライブラリのテストとしては、カラー LED の鮮やかな色の変化を楽しめる strandtest がよいでしょう。スケッチを開いたら、`#define` で 6 が設定されている信号線 PIN を 2 に設定します。また、`Adafruit_NeoPixel` コンストラクタの 1 番目の引数の 60 を、実際に接続するカラー LED の個数に変更します。

3.3.10 RC サーボの利用

RC サーボも、前節の WS2812/NeoPixel と同様に、CN4 に接続して利用することができます。

RC サーボ用の `Servo` ライブラリは、基本的なライブラリとして ESP8266 用のコアに組み込まれており、改めてライブラリをインストールしなくても利用できます。

広く利用されている Arduino UNO などの `Servo` ライブラリと同様に利用することができますので、その利用法を参考にしてください。

<https://www.arduino.cc/en/Reference/Servo>

ESP8266 用のコアに含まれている Sweep スケッチを参考に示します。

```
#include <Servo.h>

Servo myservo;

void setup()
{
  myservo.attach(2); // RC サーボを接続する CN4 は 2 番端子
}

void loop()
{
  int pos;

  for(pos = 0; pos <= 180; pos += 1) { // 0 度から 180 度へ
    myservo.write(pos);
    delay(15);
  }
  for(pos = 180; pos >= 0; pos -= 1) { // 180 度から 0 度へ
    myservo.write(pos);
    delay(15);
  }
}
```

図 3.15 RC サーボの利用

3.4 ESP8266-LEAF の応用例

ESP8266-LEAF の活用に有用な情報を WEB 上で公開しています。記載内容が不十分な部分が多々ありますが、整備を進めてまいりますのでご活用ください。

ESP8266-LEAF や ESP8266-LEAF を活用したスケッチに関しては以下のページをご参照ください。

- <http://www.microfan.jp/esp8266/esp8266-leaf>
- <http://www.microfan.jp/esp8266/workshop/index>

ESP8266-LEAF の IoT への活用に関しては、以下のページをご参照ください。

- <http://www.microfan.jp/iot/blynk/esp8266/index>
- <http://www.microfan.jp/iot/thingspeak/esp8266/index>

第 4 章

資料

4.1 ESP8266-LEAF の回路図

ESP8266-LEAF の回路図を図 4.1 に、部品表を表 4.1 に示します。

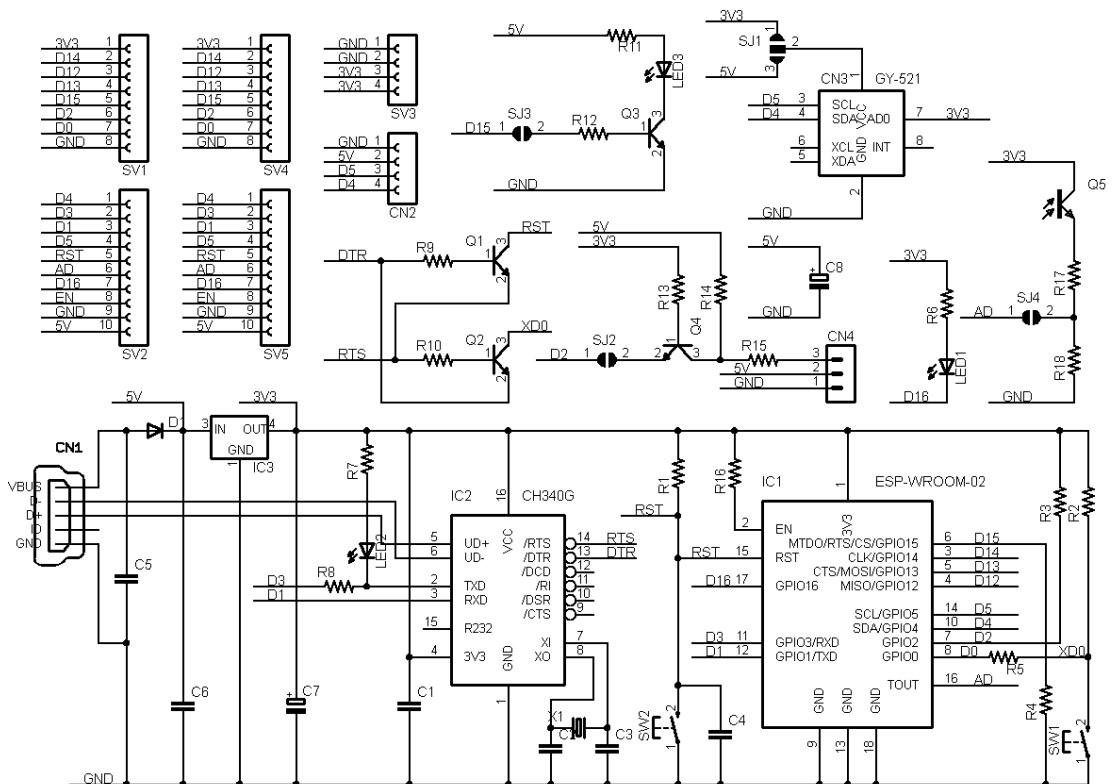


図 4.1 ESP8266-LEAF の回路図

表 4.1 部品表

部品	シンボル	規格等	個数
プリント基板	ESP8266-LEAF	Rev.4	1
IC	IC1	ESP-WROOM-02	1
	IC2	CH340G	1
	IC3	AP2114H-3.3	1
トランジスタ	Q1, Q2, Q3, Q4	S8050	4
明るさセンサー	Q5	フォトトランジスタ	1
ショットキーダイオード	D1	1N5820	1
発光ダイオード	LED1, LED2		2
赤外線発光ダイオード	LED3		1
水晶発振子	X1	12MHz	1
抵抗	R1-R4, R9, R10, R13, R14, R16	10K Ω	9
	R6, R7, R12	1K Ω	3
	R5, R8, R15	470 Ω	3
	R11	39 Ω	1
	R17	4.7K Ω	1
	R18	2.2K Ω	1
	セラミックコンデンサ	C1, C5	0.1 μ F
C2, C3		20pF	2
C4		470pF	1
C6		10 μ F	1
タンタルコンデンサ	C7	100 μ F	1
電解コンデンサ	C8	100 μ F	1
タクトスイッチ	SW1, SW2	2 端子	2
USB コネクタ	CN1	micro	1
OLED ディスプレイ	CN2	0.96 インチ, I2C	別売
センサー端子	CN3	GY-BME280/GY-521	別売
ピンヘッダ	CN4	3PIN	
ピンヘッダ	SV4, SV5	8PIN, 10PIN	
ピンソケット	SV1	1x8	別売
ピンソケット	SV2	1x10	別売
ピンソケット	SV3	1x4	別売

4.2 BOOT モードの選択

ESP-WROOM-02(ESP8266) には、下記の 2 種類のブート（起動）モードがあります。

- Flash Boot Mode
- UART Download Mode

Flash Boot Mode が一般的なブートモードで、ESP-WROOM-02 のフラッシュメモリに書き込まれたプログラムに基づいて処理が行われます。

UART Flash Boot Mode は、ESP-WROOM-02 のフラッシュメモリの内容を書き換えるためのモードで、Arduino IDE を利用したスケッチの書き込みや、NodeMCU などの firmware の書き込みに利用されます。

4.2.1 BOOT モードの選択端子

Flash Boot Mode と UART Download Mode の切り替えは、ESP-WROOM-02 の起動時に、以下に示す 3 端子の状態を設定することによって行います。両者の違いは GPIO(D0) だけで、GPIO0(D0) の状態は ESP8266-LEAF の SW1(Load) スイッチで切り替えることができます。SW1(Load) スイッチが押されていない場合には GPIO0(D) は H、押されている場合には L となります。

表 4.2 BOOT モード

ピン番号	Flash Boot Mode	UART Download Mode
GPIO15(D15)	L	L
GPIO2(D2)	H	H
GPIO0(D0)	H	L

4.3 NodeMCU 方式のフラッシュ書き込み機能

ESP8266-LEAF には、NodeMCU に採用されているフラッシュ書き込み機能が組み込まれています。したがって、Arduino IDE を利用する際に、「Reset Method」で nodemcu を選択すると、Arduino IDE で「マイコンボードに書き込む」を指定するだけで、スケッチを ESP8266-LEAF に書き込むことができます。

4.4 ハードウェア拡張用コネクタ

ESP8266-LEAF には、ハードウェアの拡張を行えるように、ESP-WROOM-02 の信号線の引き出しコネクタが用意されています。信号線は、SV1, SV2, SV4, SV5 としてまとめられ、ヘッダーピンやピンソケットで引き出せるようになっています。SV1,4 のピン配置を表 4.3 に、SV2,5 のピン配置を表 4.4 に示します。

拡張用の信号線は様々な応用に対応できるように、デジタル入出力線、SPI の信号線、アナログ入力線などから構成されています。

ESP8266-LEAF-R4 まで無接続だった 8 番ピンが ESP-WROOM-02 の EN に接続されています。

表 4.3 SV1,SV4 ピン配置

ピン番号	信号線	備考
1	3.3V	
2	D14	SCK
3	D12	MI
4	D13	MO
5	D15	SJ3 で赤外線 LED に接続
6	D2	SJ2 で CN4 に接続
7	D0	SW1
8	GND	

表 4.4 SV2,SV5 ピン配置

ピン番号	信号線	備考
1	D4	SDA
2	D3	RX
3	D1	TX
4	D5	SCL
5	RESET	SW2
6	AD	SJ4 で明るさセンサーに接続
7	D16	LED1
8	EN	
9	GND	
10	5V	

4.5 モジュール搭載用コネクタ

4.5.1 CN2: 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 4.5 に示します。

OLED ディスプレイを CN2 に取り付ける際に、OLED ディスプレイを基板にくっつく様に押し込んで取り付けると、OLED ディスプレイの裏面の部品が本ボードの USB コネクタと接触して回路がショートする可能性があります。このような問題が生じないように、OLED ディスプレイを取り付ける際には基板から少し浮かせて取り付けるようにするか、CN2 にまずピンソケットを取り付け、そのピンソケットに OLED ディスプレイを取り付けるようにしてください。

表 4.5 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	D5	SCL
4	D4	SDA

4.5.2 CN3: BME280/MPU6050 モジュール

基板上に BME280 モジュールもしくは MPU6050 モジュールを搭載するための CN3 端子を備えています。CN3 のピン配置を表 4.6 に示します。

BME280 モジュールはピン番号 1-6 に、MPU6050 モジュールはピン番号 1-8 に取り付けます。また、SJ1 を使用して、使用するモジュールに合致する電源電圧を選択します。BME280 モジュールを使用する場合には 3.3V を、MPU6050 モジュールを使用する場合には 5V を選択します。

表 4.6 CN3(BME280/MPU6050 モジュール) ピン配置

ピン番号	信号線	備考
1	3V3/5V	SJ1 で選択
2	GND	
3	D5	SCL
4	D4	SDA
5	-	CSB/XDA
6	-	SDO/XCL
7	3V3	-/AD0
8	-	-/INT

4.5.3 I2C 端子としての CN2-CN3 の活用

CN2-CN3 は、I2C 信号を利用しています。それぞれの端子に対応したディスプレイやモジュールを接続する通常の利用法のほかに、I2C 用の接続端子として利用することもできます。その場合、各端子にピンソケットかピンヘッダーを取り付け、外部の I2C デバイスに接続して利用するとよいでしょう。

各端子には、I2C の SDA,SCL の信号とともに、それぞれ下記のように電源が供給されています。

- CN2: 5V

- CN3: 3.3V/5V

利用する I2C デバイスが電源として 3.3V を利用するのであれば CN3 を、I2C デバイスに電源レギュレータなどが搭載されているようであれば CN2, CN3 を利用することができます。

4.6 電源端子

ESP8266-LEAF を USB で PC に接続している場合には、ESP8266-LEAF は USB から供給される電力で稼働し、SV2,SV5 の 5V 端子には USB からの 5V の供給電圧がショットキーダイオードによって少し低下した状態で出力されています。

PC から USB で供給できる電流は USB2 で最大で 500mA、USB3 で 900mA となっているので、ESP8266-LEAF と外部の回路の消費電流の合計がこの制限を超えない範囲で SV2,SV5 の 5V 端子を外部回路の電源として利用できます。

ESP8266-LEAF をブレッドボードなどに挿して利用する場合には、SV2,SV5 の 5V 端子に 5V 程度の電源を供給して利用します。

4.7 電源レギュレータの電流容量と電圧

ESP-WROOM-02 は 200-300mA 程度の電流を消費します。ESP8266-LEAF で利用している電源レギュレータ AP2114H-3.3 は、最大で 1A 供給*1できますので ESP-WROOM-02 を余裕をもって稼働させることができます。

AP2114H-3.3 の入力電圧から出力電圧のドロップダウンは 0.45V 程度で、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 0.9V 程度となります。USB からの供給電圧が定格の 5V をある程度下回っても、安定した電圧を維持することができます。

4.8 CH340 のデバイスドライバのインストール

ESP8266-LEAF の USB 通信チップとして CH340 *2を利用しています。

ESP8266-LEAF を USB で PC に接続した際に、ドライバーがないとメッセージが出た場合には、PC に CH340 専用のデバイスドライバをインストールする必要があります。

CH340 の WEB ページは <http://www.wch.cn/product/CH340.html> です。また、CH340 のデバイスドライバは、http://www.wch.cn/download/CH341SER_ZIP.html からダウンロードできます。

*1 ただし USB からの電源供給は USB2 で最大で 500mA、USB3 で 900mA です。

*2 CH340 のメーカーのホームページは <http://www.wch.cn/> です。

第5章

購入および問い合わせ先

5.1 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。

5.2 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>

5.3 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。

5.4 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。

5.5 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