ESP8266-CORE-R1 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

2018年8月 Copyright © 2017-2018 MicroFan, All Rights Reserved.

目次

第1章	ESP8	266-CORE-R1 の紹介	1
1.1	製品構	既要	1
	1.1.1	ESP8266-CORE-R1 の特徴	2
	1.1.2	32 ビットプロセッサ	2
	1.1.3	プログラミング環境..............................	2
	1.1.4	WiFi 無線ネットワーク装置	2
1.2	電源回	可路	3
1.3	回路の	D試作と機能拡張	3
	1.3.1	ブレッドボードの利用..............................	4
	1.3.2	汎用実験プラットフォーム:ESP8266-WORKBENCH	4
	1.3.3	IoT 実験:ESP8266-IoT-BASE	4
	1.3.4	模型自動車作成:ESP8266-MOTOR-BASE	5
	1.3.5	ロボット作成:ESP8266-SERVO-BIT	5
	1.3.6	赤外線リモコン作成:ESP8266-IR-BIT	6
1.4	購入	・作成上の注意	6
	1.4.1	プログラミング用ソフトウェア	7
	1.4.2	PC の動作環境	7
1.5	マニ:	ュアルの記載内容に関して	8
第2章	利用の	D準備	9
2.1	梱包区	内容	9
2.2	動作碩	雀認	0
	2.2.1	USB ドライバおよびインターフェース	0
	2.2.2	リセット動作の確認	0
	2.2.3	ESP-WROOM-02 の動作確認 1	1
2.3	同梱音	邵品の取り付け	2
	2.3.1	はんだ付けを行う際の注意 1	2
	2.3.2	ピンヘッダー SV1, SV2 1	3

第3章 Arduino スケッチ環境の整備

14

3.1	Arduino Core	14
3.2	ESP8266 用 Arduino 開発環境のインストール	14
	3.2.1 Arduino IDE のバージョン確認	14
	3.2.2 ボードマネージャを使用する準備	15
	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	16
	3.2.4 ボードとパラメーターの選択	16
3.3	サンプルスケッチの実行	17
	3.3.1 BLINK:LED の単純な点滅	17
	3.3.2 FADE:PWM による LED の点滅	18
	3.3.3 WiFi WebServer	19
箪₄音	答約	22
4 1	ESP8266-COBEの回路図	22
4.2	BOOT チードの選択	22
1.2	491 BOOT モードの選択端子	20 23
43	4.2.1 BOOT C 「の医尿漏」	$\frac{20}{24}$
4.0	NoteMote 7月407777721102070082	24
4.5	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	24 25
4.0	电磁	20 25
	4.5.1 电砂油 J	20 25
	4.5.2 OSD コインタル 500 电抗	20 25
	4.5.5 电ボレイエレ への穴力电圧	20 26
4.6	1.0.4 電源レイエレースの電流音量と電圧牌「 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	20 26
-		-
第5章	購入および問い合わせ先	27
5.1	ご協力のお願い....................................	27
5.2	販売:ネットショップ	27
5.3	製品情報	27
5.4	問い合わせ先....................................	27
5.5	所在地	28



1.1	ESP-WROOM-02	3
1.2	ブレッドボードへの接続..................................	4
1.3	ESP8266-WORKBENCH	5
1.4	ESP8266-IoT-BASE	5
1.5	ESP8266-MOTOR-BASE	6
1.6	ESP8266-SERVO-BIT の利用	6
1.7	ESP8266-SERVO-BIT の利用	7
2.1	ESP8266-CORE の梱包内容	9
2.2	シリアルモニタの出力 1	.1
2.3	RST コマンド 1	.2
2.4	GMR コマンド	3
2.5	ピンヘッダーの取り付け例1	.3
3.1	Board Manager を利用するための設定 1	5
3.2	ボードマネージャ 1	.6
3.3	ESP8266 の構成	.7
3.4	BLINK:LED の単純な点滅	.8
3.5	FADE:PWM による LED の点滅	.9
3.6	WiFi WebServer	20
4.1	ESP8266-CORE の回路図	22

表目次

2.1	ESP8266-CORE の梱包内容	9
4.1	部品表	23
4.2	BOOT モード	24
4.3	SV1 ピン配置	24
4.4	SV2 ピン配置	25

第1章

ESP8266-CORE-R1 の紹介



1.1 製品概要

ESP8266-CORE 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-CORE には、IoT 機器の開発や様々な応用で必要となる拡張機能を手軽に利用でき るように、様々な拡張ボードが提供されています。

ESP8266-CORE 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} Internet of Things

1.1.1 ESP8266-CORE-R1の特徴

ESP8266-CORE は以下のような特徴を持っています。

- 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 を利用できます。
- 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しています。
- 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。
- 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できます。
- 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。
- 600mAでドロップダウンが0.25Vの高性能電圧レギュレータを搭載し、ESP-WROOM-02 に安定した電力を供給できます。
- 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、細い形状の両側に二列の ピンヘッダーを組み込めるようになっており、ブレッドボードに挿して利用できます。
- 様々な試作や応用に対応した拡張ボードが提供されています。

1.1.2 32 ビットプロセッサ

ESP8266-CORE の中核となる ESP-WROOM-02 には、高機能な 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。

1.1.3 プログラミング環境

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



⊠ 1.1 ESP-WROOM-02

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-CORE 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と、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-CORE 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-CORE はドロップダウンが 0.25V@600mA と小さく、600mA の電流を供給できる 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 回路の試作と機能拡張

開発者ごとに試作や開発したい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-CORE は、開発者の広範囲な要求に応えるために、様々な拡張ボードを用意しています。

1.3.1 ブレッドボードの利用

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 1.2 に示すように ESP8266-CORE 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。

https://store.shopping.yahoo.co.jp/microfan/breadboard-30.html https://store.shopping.yahoo.co.jp/microfan/breadboard-63.html



図 1.2 ブレッドボードへの接続

1.3.2 汎用実験プラットフォーム: ESP8266-WORKBENCH

ESP8266-WORKBENCH は、様々な実験・試作で必要とされる OLED ディスプレイ、スイッ チ、LED などを搭載するとともに、それらの機能を I2C を利用して接続し、ESP-WROOM-02 の少ない信号線を様々な実験・試作で効果的に利用できるように構成した拡張ボードキットです。 https://store.shopping.yahoo.co.jp/microfan/esp8266-workbench-r1-kit.html

1.3.3 IoT 実験: ESP8266-IoT-BASE

ESP8266-IoT-BASE は、IoT システムに多用されるセンサー類とデバイス類を搭載するとともに、外部デバイスの接続機能をまとめた拡張ボードキットです。

https://store.shopping.yahoo.co.jp/microfan/esp8266-iot-base-r1-kit.html



⊠ 1.3 ESP8266-WORKBENCH



 $\boxtimes 1.4$ ESP8266-IoT-BASE

1.3.4 模型自動車作成: ESP8266-MOTOR-BASE

ESP8266-MOTOR-BASE は、ESP8266-CORE に DC モーターや RC サーボを接続できるようにする拡張ボードキットです。

https://store.shopping.yahoo.co.jp/microfan/esp8266-motor-base-r1-kit.html

1.3.5 ロボット作成: ESP8266-SERVO-BIT

ESP8266-SERVO-BIT は、ESP8266-CORE に 8 個の RC サーボを接続できるようにする拡張 ボードです。

https://store.shopping.yahoo.co.jp/microfan/esp8266-servo-bit-r1-kit.html



⊠ 1.5 ESP8266-MOTOR-BASE



図 1.6 ESP8266-SERVO-BIT の利用

1.3.6 赤外線リモコン作成: ESP8266-IR-BIT

ESP8266-IR-BIT は、ESP8266-CORE に赤外線リモコン機能を追加する拡張ボードです。 https://store.shopping.yahoo.co.jp/microfan/esp8266-ir-bit-r1-kit.html

1.4 購入・作成上の注意

ESP8266-CORE-R1 をご購入の際には、下記項目をご確認ください。



図 1.7 ESP8266-SERVO-BIT の利用

1.4.1 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-CORE 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。

1.4.2 PC の動作環境

弊社では、ESP8266-CORE 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^{*2}と考えられますが、弊社では動作確認は行っていません。ESP8266-CORE 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。

^{*&}lt;sup>2</sup> 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。

1.5 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。

第2章

利用の準備

2.1 梱包内容

ESP8266-CORE 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。

部品	シンボル	規格等	個数
組み立て済基板	ESP8266-CORE	Rev.1	1
ピンヘッダ	SV1, SV2	1列 8PIN	2

表 2.1 ESP8266-CORE の梱包内容



図 2.1 ESP8266-CORE の梱包内容

2.2 動作確認

ESP8266-CORE 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、4.6 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

ー度 ESP8266-CORE 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-CORE 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-CORE 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-CORE 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-CORE を利用するために、この COM ポートを選択します。

ESP8266-CORE 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
- 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 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イル (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-CORE を PC に接続した状態で、Arduino IDE を起動し [ツール] \Rightarrow [ポート] メ ニューで ESP8266-CORE に対応した適切な COM ポートを選択してください。

^{*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。

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-CORE 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にょうにょと出力された後に 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
- \bullet 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-CORE 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、作業内容と状況を取り まとめて電子メイルでマイクロファンにお問い合わせください。

```
AT+RST
UK
 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

2.3 同梱部品の取り付け

ESP8266-CORE 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ピンヘッダーをボードにはんだ付けしてご利用ください。 ピンヘッダーの取り付けは、ESP8266-CORE の動作確認後に行ってください。 ピンヘッダーの取り付け例を図 2.5 に示します。

2.3.1 はんだ付けを行う際の注意

ESP8266-CORE のプリント基板はベたアースになっており、GND に接続された端子の熱容量 が大きくなっております。このため、ピンヘッダーの GND 端子をはんだ付けする際には基板の 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 ピンヘッダーの取り付け例

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 ピンヘッダー SV1, SV2

ESP8266-CORE をブレッドボードに接続して利用する場合には、SV1, SV2 にピンヘッダーを 取り付けます。

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。

第3章

Arduino スケッチ環境の整備

3.1 Arduino Core

ESP8266-CORE 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

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} 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。

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、 $[ファイル] \Rightarrow [環境設定] を選択します。すると、図 3.1 に$ 示すようなダイアログが表示されます。

環境設定		
設定 ネットワーク		
スケッチブックの保存場所:		
E:¥arduino		参照
言語設定:	パソコンの設定に従う 🗸	変更の反映にはArduino IDEの再起動が必要
エディタの文字の大きさ:	12	
Interface scale:	🔽 Automatic 🛛 100 킂 🕅 変更の反映には Arduino	IDEの再起動が必要
より詳細な情報を表示する:	□ コンパイル □ 書き込み	
コンパイラの警告:	ネフフ其月 ▼	
☑ 行番号を表示する		
 コードの折り返しを有効 フーキジスカムを検証する 	ē.	
● 各さたのを快知する	3	
■ 記動時に最新バージョン	- /の有無をチェックする	
☑ スケッチを保存する際に	、拡張子をpdeから.inoに変更する	
▼ 検証または書き込みを行	テラ前にスケッチを保存する	
追加のボードマネージャのUF	RL: http://arduino.esp8266.com/stable/package_esp8	266com_index.json
以下のファイルを直接編集す C:¥L ¥AppD:	れば、より多くの設定を行うことができます。 stationalWarduinalSXareferences.tvt	
編集する際には、Arduino I	Na+cocai+miduino is+preferences.com	
		OK キャンセル

図 3.1 Board Manager を利用するための設定

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、 $["] \rightarrow [" - "] \rightarrow [" - "] \rightarrow [" - "] \rightarrow [" - "] \rightarrow "]$ 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。



図 3.2 ボードマネージャ

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]l ボタンが表示すされるので、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-CORE 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 3.3 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-CORE 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
- WebServer

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 $[7 r \ell \lambda] \Rightarrow [スケッチの例] \Rightarrow [01.Basics] から Blink を選択して$ ください。ESP8266-CORE 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-CORE 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 $[7r \ell \lambda] \Rightarrow [\lambda f \gamma f \sigma M] \Rightarrow [01.Basics] から Fade を選択してください。$ ESP8266-CORE 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、analog-WriteRange() 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-CORE 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。

```
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

3.3.3 WiFi WebServer

ESP8266 Arduino 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
```

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
const char ssid[] = "ESP8266";
ESP8266WebServer server(80);
void handleRoot()
ſ
  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()
ł
  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()
ſ
  server.handleClient();
}
```

⊠ 3.6 WiFi WebServer

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-CORE 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。

第4章

資料

4.1 ESP8266-COREの回路図

ESP8266-CORE 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-CORE の回路図

部品	シンボル	規格等	個数
プリント基板	ESP8266-CORE	Rev.1	1
IC	IC1	ESP-WROOM-02	1
	IC2	CH340C	1
	IC3	AP2112K-3.3	1
トランジスタ	Q1	UMH3N	1
ショットキーダイオード	D1	SS34	1
発光ダイオード	LED1, LED2		2
抵抗	R1-R4	10K Ω	4
	R5, R8	470 Ω	2
	R6, R7	1Κ Ω	2
セラミックコンデンサ	C1, C6	$22 \ \mu$ F	2
	C2, C4	$0.1 \ \mu$ F	2
	C3	$1\mathrm{nF}$	1
	C5	10 μ F	1
タクトスイッチ	SW1, SW2	2 端子	2
USB コネクタ	CN1	micro	1
ピンヘッダ	SV1, SV2	8pin	2

表 4.1 部品表

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-CORE 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。

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

表 4.2 BOOT モード

4.3 NodeMCU 方式のフラッシュ書き込み機能

ESP8266-CORE 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-CORE に書き込むことができます。

4.4 ハードウェア拡張用端子

ESP8266-CORE には、拡張ボードやブレッドボードとの接続用に、ESP-WROOM-02 の信号 線の引き出し端子が用意されています。信号線は、SV1, SV2 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 のピン配置を表 4.3 に、SV2 のピン配置を 表 4.4 に示します。SV1 の1番ピンは、端子の並びの ESP-WROOM-02 側、SV2 の1番ピンは、 端子の並びの USB コネクタ側となっています。

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。

信号線	備考
3.3V	出力のみ
D14	SCK
D12	MI
D13	МО
D15	
D2	
D0	SW1
GND	
	信号線 3.3V D14 D12 D13 D15 D2 D0 GND

表 4.3 SV1 ピン配置

ピン番号	信号線	備考
1	5V	入出力可
2	D4	SDA
3	D3	RX
4	D1	ΤХ
5	D5	SCL
6	RESET	SW2
7	AD	
8	D16	LED1

表 4.4 SV2 ピン配置

4.5 電源

4.5.1 電源端子

ESP8266-CORE を USB で PC に接続している場合には、ESP8266-CORE は USB から供給 される電力で稼働し、SV2 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。

ESP8266-CORE を USB 接続なしで利用する場合には、SV2 の 5V 端子に 5V 程度の電源を供給して利用します。

なお、SV1の3.3V 端子は出力専用で、外部から3.3Vの電源供給をすることはできません。

4.5.2 USB コネクタからの電流

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-CORE 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 の 5V 端 子を外部回路の電源として利用できます。

USB コネクタに接続されているショットキーダイオードは最大で 3A の電流を流すことができ るため、USB バッテリーや AC アダプタなどから USB ケーブルを通じて ESP8266-CORE に電 力を供給する場合には、最大で 5V3A 程度の電流を流すことができます。このため、USB コネク タを通じて供給した電源で、ある程度の電流を必要とする RC サーボや、数十個程度のシリアル カラー LED を駆動することができます。

4.5.3 電源レギュレータの入力電圧

ESP8266-CORE で利用している電源レギュレータ AP2112K-3.3 の入力電圧の最大値は 6.5V となっています。USB コネクタから適切に電力を供給する場合には問題ありませんが、電池や AC アダプタなどから 5V 端子に電力を供給する場合には、入力電圧が 6.5V を超えないようにご 注意ください。

4.5.4 電源レギュレータの電流容量と電圧降下

ESP-WROOM-02 は 200mA 程度の電流を消費します。ESP8266-CORE で利用している電 源レギュレータ AP2112K-3.3 は、3.3V の電源を最大で 600mA 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。

AP2112K-3.3 の入力電圧から出力電圧のドロップダウンは 0.25V 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 0.6V 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。

4.6 CH340 のデバイスドライバのインストール

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

ESP8266-CORE 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。

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イ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