

GPIOに関して

GPIOは General Purpose Input/Output の略で 汎用I/Oポートとも呼ばれ、入力にも出力にも使える便利なデジタル信号の出入り口です。 デジタル出力のセンサーや、スイッチの ON/OFFをマイコンに伝える入力や、マイコンの演算結果を LEDに表示したり、モーターを動かす信号を出力したりする時などに、利用します。

GPIOが 汎用と呼ばれる 大きな理由は、端子を入力用にも 出力用にも 使えるからです。初期のマイコンでは、端子の役割が入力か、出力に固定されていましたが、 今では多くのマイコンで入出力を 自由に設定可能です。仮にGPIO端子が 8本であれば、4本を 入力、4本を出力として 使うこともできますし

入力は1本で 出力は7本といった使い方も可能です。

で、今回のESP32の場合というか Arduino IDE の環境で プログラム開発を行う場合、入出力の機能を 関数アクセスで実現するようになっており ESP32と、Arduino UNOで、I/O周りの関数は 共通の名前の関数が使えます。では、GPIOをアクセスする関数を紹介します。

① ピンのモードを設定する

GPIOピンのモードを ピン毎に独立して 入力または 出力に設定する事が出来ます。これは、以下の関数で行います。

```
void pinMode( 番号、モード );
```

例えば、3番ピンを 出力に設定する場合だと `pinMode(3, OUTPUT);` に なります。

モードは

1-1 OUTPUT ピンを出力に設定する。

1-2 INPUT ピンを入力に設定する。

1-3 INPUT_PULLUP ピンを入力に設定して内部のプルアップ抵抗を使う。

の 3つが あります。

② デジタル出力を行う。

デジタル出力は、LEDの点灯、消灯など HIGHか LOWの 2択の出力です。

これは、以下の関数で行います。

```
void digitalWrite( ピン番号、値 );
```

値は、LOWか HIGH です。

たとえば、3番ピンを HIGH にするには次のように書きます。

```
digitalWrite( 3、HIGH );
```

2番ピンを LOW にするには

```
digitalWrite( 2、LOW );
```

 です。

③ デジタル入力を行う。

デジタル入力は、スイッチの入力など HIGHか、LOWのどちらかの状態を得る入力です。これは、以下の関数で行います。

```
int digitalRead( ピン番号 );
```

便宜上、関数値を int にしましたが、関数値は、0 と 1 しか戻ってこないで、char でも boolean でも 受けられます。但し、boolean の場合は、比較する時 True か False になるかもしれません。

数値	High/Low	boolean
1	HIGH	True
0	LOW	False

例) 3ピンから デジタル入力を得て、変数 i に代入する場合は、次のようになります。

```
i = digitalRead( 3 );
```

GPIOピンの ピンアサイン

ESP32も 種類が いろいろあるので、私が持っている物で、ピンアサインを確認します。

まず、旧 ESP32というか DEV-KITの基板でピンアサインを確認します。ピン数は、30ピンの物と、38ピンの物があります。基板上のピンの横には、略した信号名が記載してあります。ピンの足番号というか、1番から連番で付けられている番号が無いので、足ピン位置を数えにくい要素があります。で、仮の足番号を付ける事にします。基板を 部品側から見て上を蛇行アンテナ側、下をUSBコネクタ側にし、左上(蛇行アンテナの左)から、下に向かい1, 2, 3 と連番を付けます。仮に 30ピンの基板であれば、一番下が 15ピンとなります。そして、右下から 16ピンで、上方向に連番を付けると、右の一番上は 30ピンとなります。

38ピンの基板であれば、左上を 1ピンとして一番下が 19ピンとなります。

そして、右下から 20ピンで、上方向に連番を付けると、右の一番上は 38ピンとなります。

GPIOの番号が **飛んでいる部分もあるので使用可能な GPIOピンの本数が、分かりにくい**

そして、GPIOの番号も 連番で並んでいないところがあるので、探さないといけない。

という事で、**GPIOの欠番もハッキリ分かる**ような対応表を作ろうと思っていました。

旧 ESP32 30ピン GPIOピンのピンアサイン表

EN	1		30	io-23
io-36	2		29	io-22
io-39	3		28	io-1
io-34	4		27	io-3
io-35	5		26	io-21
io-32	6		25	io-19
io-33	7		24	io-18
io-25	8		23	io-5
io-26	9		22	io-17
io-27	10		21	io-16
io-14	11		20	io-4
io-12	12		19	io-2
io-13	13		18	io-15
GND	14		17	GND
VIN	15		16	3V3

緑のI/Oポートは、
後で説明します。

使用出来る GPIOピン 25本

1	io-1	28	14	io-21	26
2	io-2	19	15	io-22	29
3	io-3	27	16	io-23	30
4	io-4	20	17	io-25	8
5	io-5	23	18	io-26	9
6	io-12	12	19	io-27	10
7	io-13	13	20	io-32	6
8	io-14	11	21	io-33	7
9	io-15	18	22	io-34	4
10	io-16	21	23	io-35	5
11	io-17	22	24	io-36	2
12	io-18	24	25	io-39	3
13	io-19	25			

GPIOは io-で、略してます。

赤線は、GPIOの番号が飛んでいる箇所です。
シリアル通信や、A/D入力、I2Cインタフェース等
を使用していると その分端子は 減少します。

3V3	1		38	GND
EN	2		37	io-23
io-36	3		36	io-22
io-39	4		35	io-1
io-34	5		34	io-3
io-35	6		33	io-21
io-32	7		32	GND
io-33	8		31	io-19
io-25	9		30	io-18
io-26	10		29	io-5
io-27	11		28	io-17
io-14	12		27	io-16
io-12	13		26	io-4
GND	14		25	io-0
io-13	15		24	io-2
FL-D2	16		23	io-15
FL-D3	17		22	FL-D1
FL-CMD	18		21	FL-D0
VIN	19		20	FL-CK

旧 ESP32 38ピン GPIOピンの ピンアサイン表

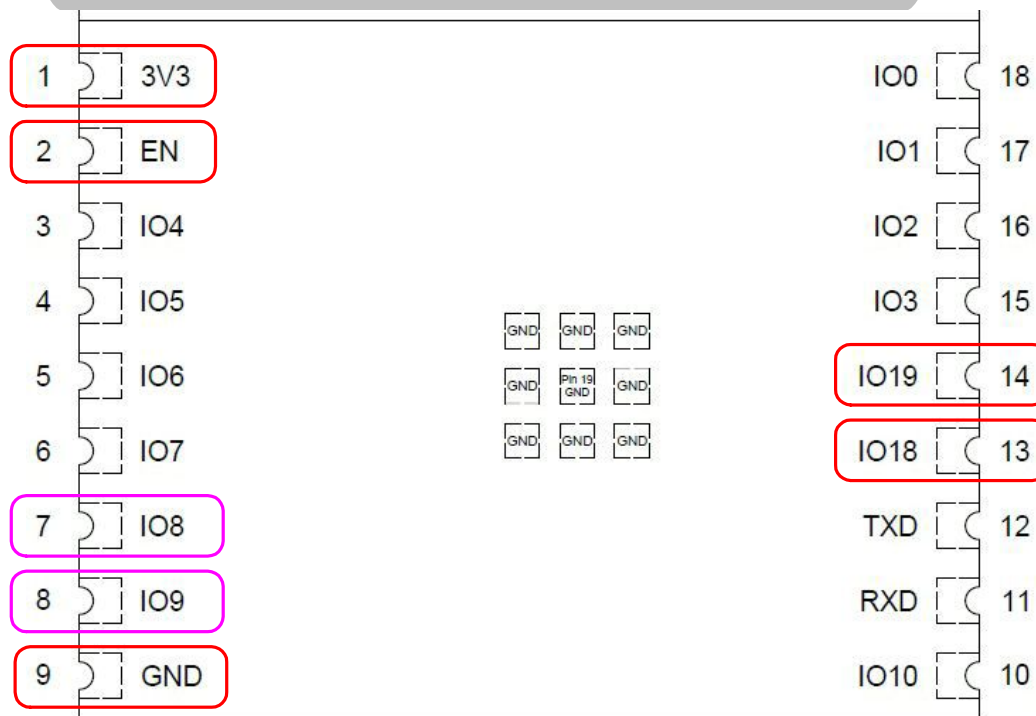
使用出来る GPIOピン 26本

1	io-0	25	10	io-15	23	19	io-26	10
2	io-1	35	11	io-16	27	20	io-27	11
3	io-2	24	12	io-17	28	21	io-32	7
4	io-3	34	13	io-18	30	22	io-33	8
5	io-4	26	14	io-19	31	23	io-34	5
6	io-5	29	15	io-21	33	24	io-35	6
7	io-12	13	16	io-22	36	25	io-36	3
8	io-13	15	17	io-23	37	26	io-39	4
9	io-14	12	18	io-25	9			

赤線は、GPIOの番号が飛んでいる箇所です。

30ピンのモジュールに比べ 8ピン増えているのに GPIOピンは 1本しか増えていません。その理由は、左のピンアサイン表の **FL-** が、Flash ROM信号線のため使えません。30ピンの物は、Flashの信号がピンに出て無いため Flashの事を気にせずに使えます。

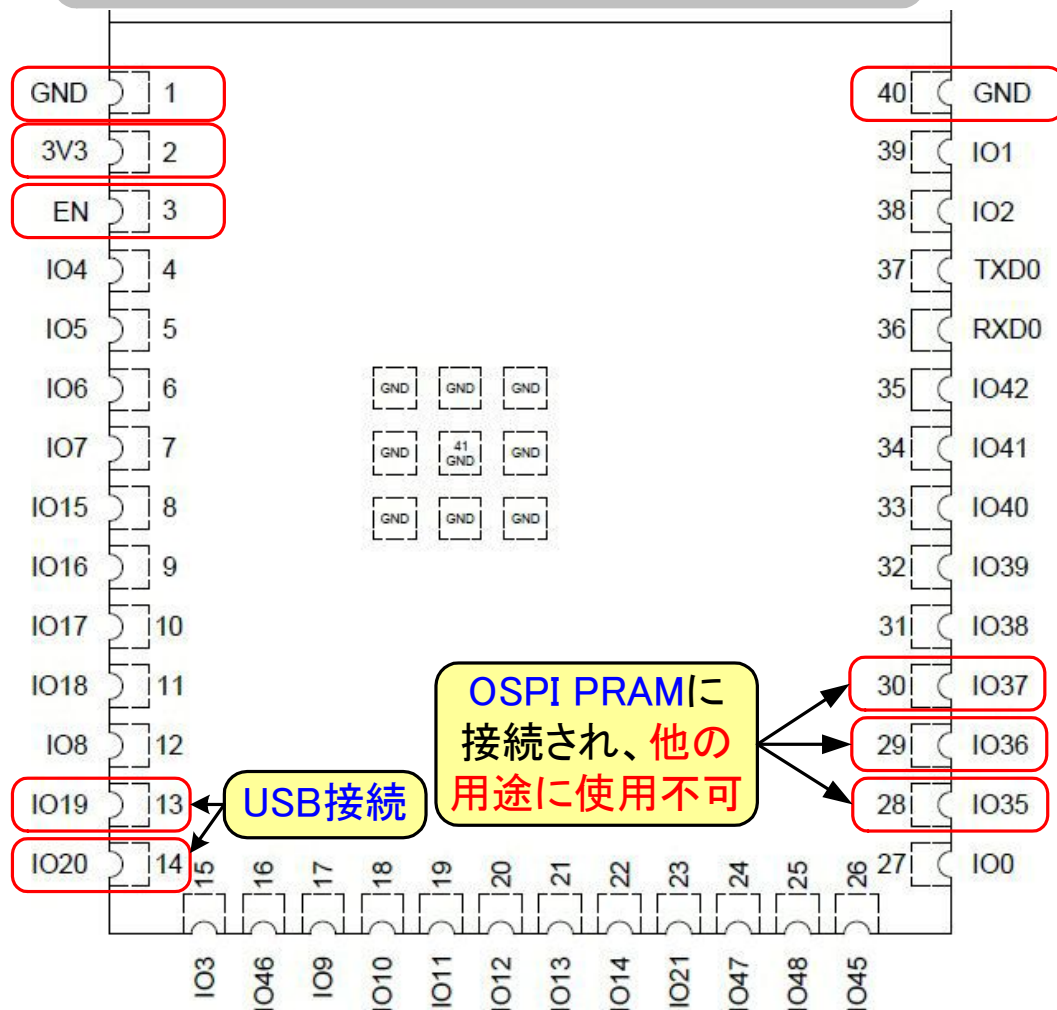
ESP32_C3_WROOM_02 ピンアサイン表



18ピンの端子がありますが、**赤枠**は専用用途で使えません。ピンクは書き込み時要注意の信号です。残りは多分GPIOとして使えると思います。

- [1番 **3V3**]は 3.3Vの電源供給端子です。
- [2番 **EN**]は、リセット入力信号です。
- [7番 **IO8**]は、常時10kΩでプルアップする必要があります。プルアップしないと、自動で書き込みモード、実行モードに切り替えられないです。
- [8番 **IO9**]は、ブートモード信号になります。
リセット直後 Lowであればブートモードです。
- [9番 **GND**]は、0 電位 グランドです。
- [11番 **RxD**]は GPIOに設定する場合 **GPIO 20** になります。
- [12番 **TxD**]は GPIOに設定する場合 **GPIO 21** になります。
- [13番、14番]は、**IO18、IO19**と書いてありますが **USBの高速伝送信号を接続**します。
13番が **D-**、14番が **D+** です。という事で、GPIOとして使えるのは、**IO-0 ~ IO-7、IO-10、IO-20、IO-21** になります。計 11本です。

ESP32_S3_WROOM_1 ピンアサイン表



ESP32 S3 WROOM 1 使用可能と思われるGPIOピン

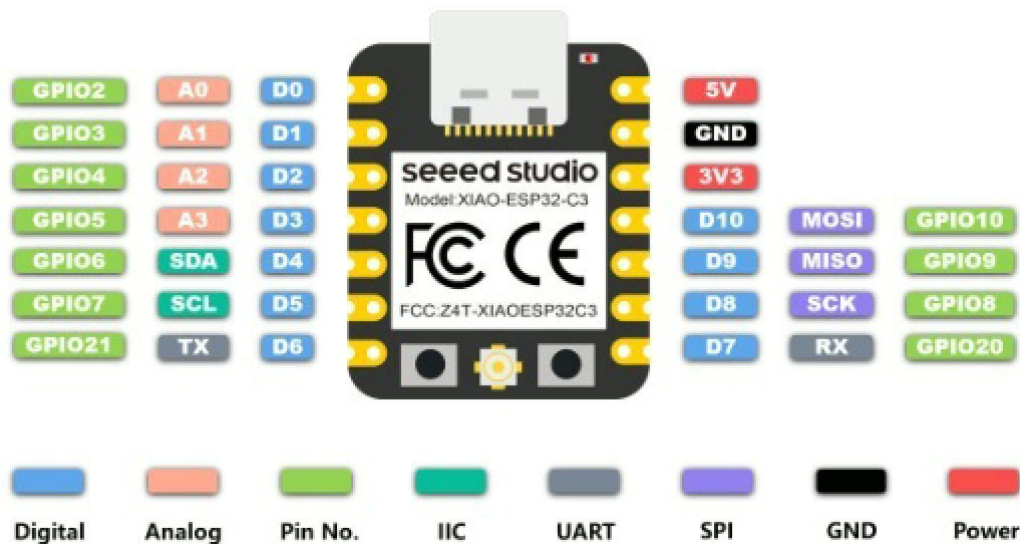
連番	ピン番	GPIO番号	連番	ピン番	GPIO番号
1	27	IO-00	17	9	IO-16
2	39	IO-01	18	10	IO-17
3	38	IO-02	19	11	IO-18
4	15	IO-03	20	23	IO-21
5	4	IO-04	21	31	IO-38
6	5	IO-05	22	32	IO-39
7	6	IO-06	23	33	IO-40
8	7	IO-07	24	34	IO-41
9	12	IO-08	25	35	IO-42
10	17	IO-09	26	37	IO-43 TXD0
11	18	IO-10	27	36	IO-44 RXD0
12	19	IO-11	28	26	IO-45
13	20	IO-12	29	16	IO-46
14	21	IO-13	30	24	IO-47
15	22	IO-14	31	25	IO-48
16	8	IO-15			

ESP32 S3_WROOM_1 の 現時点で 使用可能と思われる GPIOピンの一覧表です。

全部で、31ピンあります。これは予想外に 多かったですね。もしかして まだ何らかの制約で、使えなくなるピンが あるかもしれません。

AIXO ESP32 C3 S3 ピンアサイン表

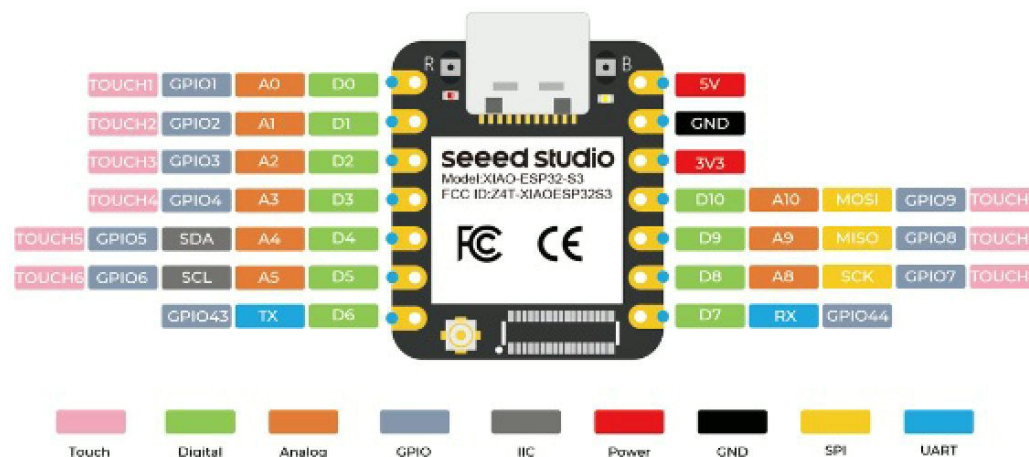
AIXOの ESP32モジュールの細かい事を記載した資料は、なかなか入手できないので、スイッチサイエンスのサイトで、表示されていたピンアサイン表をお借りして 説明します。



上は **AIXO ESP32 C3**モジュールの ピンアサイン表です。ピン数が少ない分 楽ですね。

D9と D8が 使えなかったと思いますので、
使える IOピンは、IO-0 ~ IO-7 と IO-10です。よって 計 9本です。

今度は、AIXO ESP32 S3です。



使える IOピンは、ESP32 S3 WROOM_1の資料を参考にしましたが、IO-0 ~ IO-7 及び IO-8 ~ IO-10は ピンの機能説明に デフォルトの設定が GPIOで 書いてありましたので 多分、使えると思います。 よって 計 11本です。

旧 ESP32で、GPIOを使う時の 注意事項

旧 ESP32の Dev-Kit WROOM-32の基板モジュールで I/Oポートの実験を行った際に発覚した現象で、IO34、IO35、IO36、IO39 が信号出力出来ない件にて、データシートを確認した結果、この4本のポートは、I/Oではなくて **I のみの入力専用ポート**である事が、分かりました。データシートの Pin Definitions の表で Typeの欄に 通常の IOピンは I/O と 記載してありますが、上記 IO34、IO35、IO36、IO39 は、**I** と 記載してあります。（右図参照）

よって IO34、IO35、IO36、IO39は、入力端子です。

ESP32 C3 S3も 確認しましたが、C3、S3に関しては、全ての IOと 書かれている端子の Typeは I/O でした。

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	High: On; enables the chip Low: Off; the chip power is off Note: Do not leave the pin floating
SENSOR_VP	4	I	GPIO36, ADC1_CH0, R
SENSOR_VN	5	I	GPIO39, ADC1_CH3, R
IO34	6	I	GPIO34, ADC1_CH6, R
IO35	7	I	GPIO35, ADC1_CH7, R
IO32	8	I/O	GPIO32, XTAL_32K_P (GPIO0), TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (GPIO0), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH1
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH2

リトル エンディアンと ビッグ エンディアン

ビッグエンディアン、リトルエンディアンとは何かというと 例えば 16bit 整数値をメモリの 100h番地から 書き込むとすると、

ビッグエンディアンの場合

100h 整数の 上位バイト 書込み
101h 整数の 下位バイト 書込み
になります。

リトルエンディアンの 場合

100h 整数の 下位バイト 書込み
101h 整数の 上位バイト 書込み
になります。

ビッグエンディアン、リトルエンディアンは、メモリに書き込む先頭番地に、上位バイトから 順に書き込むか、下位バイトから 順に書き込むかの 選択を 意味します。

因みに 4byte整数 (32bit)を、メモリの 100h番地から 書き込む場合は、

ビッグエンディアンの場合

100h 4byte整数の 最上位バイト 書込み
101h 4byte整数の 上中位バイト 書込み
102h 4byte整数の 中下位バイト 書込み
103h 4byte整数の 最下位バイト 書込み
になります。

リトルエンディアンの 場合

100h 4byte整数の 最下位バイト 書込み
101h 4byte整数の 中下位バイト 書込み
102h 4byte整数の 上中位バイト 書込み
103h 4byte整数の 最上位バイト 書込み
になります。

で、ESP32は ビッグエンディアンなのか、リトルエンディアンなのか、資料では 分からないので、実験で 確かめます。先に 結果を 伝えと ESP32は リトルエンディアンでした。

