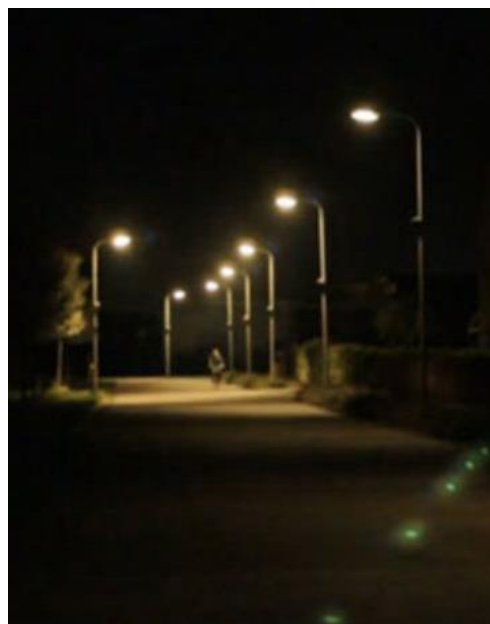


アナログ入力…CdS（硫化カドニウムセル）を使う

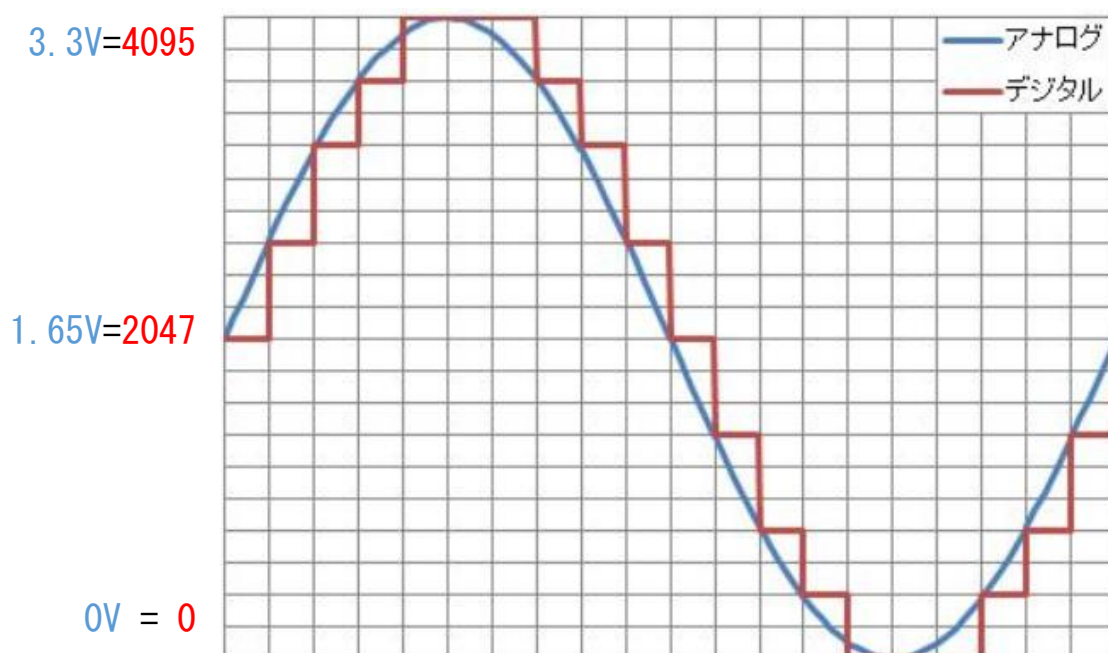
《何ができる？》アナログ入力は、ピン端子に 0V～3.3V のアナログの電圧値を入力することができます。様々なセンサー類（照度・温度・湿度・気圧など）は測定する対象に応じて細かなアナログ値を変化させる。

屋外に設置されている街灯は、外が暗くなると明かりをつける仕組みになっているが、ここでは照度センサ（光センサ）が使われている。



アナログ入力…ADC

ESP32 のアナログ入力は、ADC (Analog-Digital Converter) というものでセンサーの値をサンプリングした後 12 ビットで表現できる。12 ビットで表現できる数は $0 \sim 2^{12}-1$ なので、0～4095 の数値が ESP32 側のピンに入力される。



もし ESP32 のアナログ入力ピンに 3.3V の電圧が入ってきたら、アナログ入力の値は 4095 になる。

もし、2V の電圧が入力されてきたら、アナログ入力 a の値はどうなる？

比例配分で、 $3.3[V] : 4095 = 2[V] : a$ となるので：

$$a = \frac{2 \times 4095}{3.3} \doteq 2481 \text{ となる}$$

アナログ入力関数

アナログ入力には、`analogRead()` 関数が用いられます。

<code>uint16_t analogRead (uint8_t pin)</code>	
指定したピンでアナログ入力値を読み込む。	
【パラメータ】	<code>pin</code> : 利用するピン。
【戻り値】	12 ビットで表されるアナログ入力の値 (0-4095)。 <code>uint16_t</code> 型は符号なし 16 ビット整数型のこと。

アナログ入力には、ピンの入出力モードを設定する `pinMode()` は不要。ただし、アナログピンは以下に示すピン番号でしか使えません。

ESP8266 Developer で利用可能なピン（利用不可のピンは除く）…PWM で利用可能なピンと同じ。

アナログピン番号	GPIO 番号	アナログピン番号	GPIO 番号	アナログピン番号	GPIO 番号
A4	32	A12	2(Boot Mode)	A16	14
A5	33	A13	15	A17	27
A10	4	A14	13	A18	25
A11	0(Boot Mode)	A15	12	A19	26

《サンプル①》 I/O27 番ピンにアナログ入力を接続し、アナログ入力値を読み取りシリアルモニタに表示させる。シリアルモニタを表示させると 0~4095 までの数値が表示されます。

<pre>1 const int analogPin = A17; //GPIO I/O27pin 2 3 void setup() { 4 Serial.begin(115200); 5 6 } 7 8 void loop() { 9 int a = analogRead(analogPin); 10 Serial.println(a); 11 delay(10); 12 }</pre>	シリアルモニタ 3223 ←通常 3000 前後 3235 (部屋による) ... 3194 1439←CdS センサーを 1456 指でつまむ 1456 3199 3206 ... 3359 3985←スマホのライト 3945 で照らす 3967 4019
--	--

演習

【目標】

ESP32 ボードに CdS センサーをつなぎ、アナログ入力を読みとる。アナログ入力の値に応じて制御を行うプログラムを考える。

【1. ESP32 と電子工作部品との接続】

1. 1. 必要な部品

パーツ名	必要個数
CdS センサー	1 個
100k Ω 抵抗	1 個（茶黒橙金）
ジャンパーワイヤ	2 個（黄色・橙色）
ジャンパーコード	3 本（赤・黒・青色）

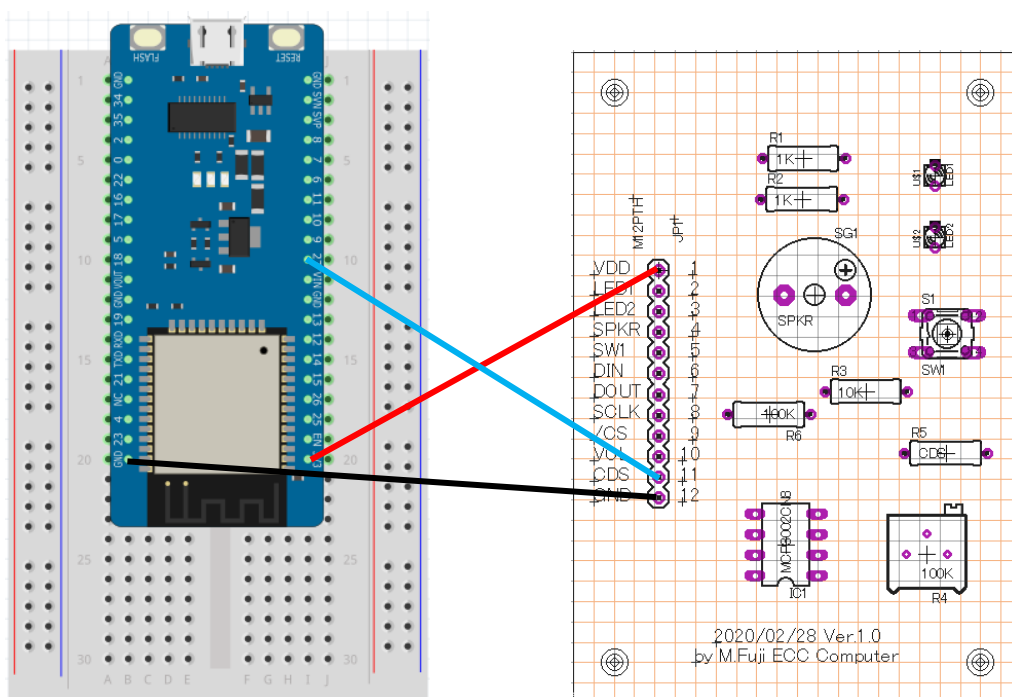
GPIO ボードに搭載
されているので不要

以下の接続を行う。

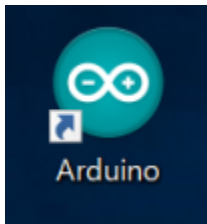
- ① ジャンパーコード（赤）ESP32 の 3V3 － GPIO ボード VDD (1 番ピン)
- ② ジャンパーコード（黒）ESP32 の GND － GPIO ボード GND (12 番ピン)
- ③ ジャンパーコード（青）ESP32 の IO27 － GPIO ボード CDS (11 番ピン)

※LED 出力、スイッチ入力を行う場合 (kad12, kad13) は以下の接続を追加する必要あり。

- ④ ジャンパーコード（黄）ESP32 の IO16 － GPIO ボード LED1 (2 番ピン)
- ⑤ ジャンパーコード（黄）ESP32 の IO4 － GPIO ボード LED2 (3 番ピン)
- ⑥ ジャンパーコード（白）ESP32 の IO17 － GPIO ボード SW1 (5 番ピン)




【2. Arduino スケッチのサンプルプログラムをプログラム実行】



デスクトップのアイコンをダブルクリックして
Arduino IDE を起動する。

【課題 11】プロジェクト名「kad11_CdS」



まずは、《サンプル①》を打ち込んで、動作確認をする。 シリアルモニタを開き CdS センサーに手を
かざして光を遮ったり、CdS センサーをつまんだりしてアナログ入力の値が下がるかどうか確認する。
可能ならスマホのライトを CdS センサーにあててアナログ入力の値が上がるか確認する。

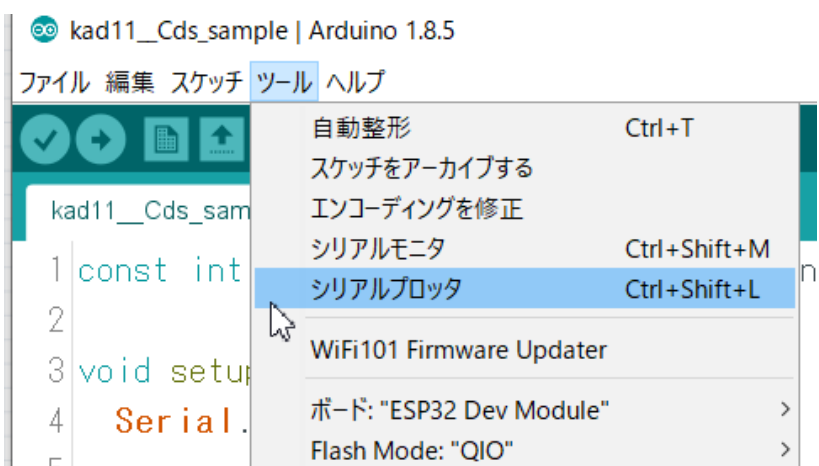
<pre>1 const int analogPin = A17;//GPIO IO27pin 2 3 void setup() { 4 Serial.begin(115200); 5 6 } 7 8 void loop() { 9 int a = analogRead(analogPin); 10 Serial.println(a); 11 delay(10); 12 }</pre>	<p>シリアルモニタ</p> <p>3223 ←通常 3000 前後 3235 (部屋による) ... 3194 1439←CdS センサを 1456 指でつまむ 1456 急激に下がる 3199 3206 ... 3359 3985←スマホのライト 3945 で照らす 3967 急激に上がる 4019</p>
--	--

【シリアルプロッタを使ってみる】

シリアルモニタでアナログ入力の値
が思った値を表示しているのを確認
したら、シリアルプロッタでアナロ
グ入力の値をグラフ表示させてみよ
う。

まず、シリアルモニタを閉じておい
て、[ツール]メニューから[シリアル
プロッタ]を選択する。

※シリアルモニタを開いている間はシ
リアルプロッタを使えないので注意！





シリアルプロッタの動作が確認できたら、シリアルモニタで以下の様なアナログ入力の電圧値が表示されるようにせよ。書式指定としては、以下の通り

“アナログ入力値は右寄せ 4 桁表示=電圧値は少数第 2 位まで表示[改行]”

`Serial.printf("%3d¥n", duty);` C 言語の `printf()` 関数と同じ書式指定子が使え。

`"%3d¥n"` は、右揃え 3 桁の整数+改行となる。小数点以下の桁数表示に使う書式指定子は？

アナログ入力値が 0 ならば電圧値は 0V で、アナログ入力値が 4095 なら電圧値は 3.3V となる。

アナログ入力値が a ならば、電圧値は x はいくらになりますか？

比例配分で、 $3.3[V] : 4095 = x[V] : a$ となるので、自分で式をたててみましょう。

以下の様な出力になれば OK !

シリアルモニタ	シリアルモニタ (つづき)
2843=2. 29[V] ←通常 3000 前後	624=0. 50[V]
2843=2. 29[V] (部屋による)	658=0. 53[V]
2832=2. 28[V]	692=0. 56[V]
...	1809=1. 46[V] ←指をはなすと元の値に戻る
2812=2. 27[V]	2255=1. 82[V]
2611=2. 10[V]	...
2303=1. 86[V]	2799=2. 26[V]
1582=1. 27[V] ←CdS センサを指でつまむ	3215=2. 59[V] ←スマホのライト
954=0. 77[V] 急激に下がる	3287=2. 65[V] で照らすと
658=0. 53[V] 3 桁になっても右寄せ	3551=2. 86[V] 急激に上がる
...	4095=3. 30[V] ←最大値

【課題 12】 プロジェクト名「kad12_CdS2LED」

CdS センサーの値が小さくなった時＝暗くなった時、LED が点灯するようなスケッチを作りたい。

CdS センサーを指でつまんだ時＝暗くなったとして、いままでの課題で接続済みの LED が両方とも点灯するようにスケッチを作成せよ。

GPI016 番につながった赤色 LED はデジタル出力そのまま、GPI04 番 (=A10 番) ピンにつながった青色 LED もデジタル出力にして点灯することとする。pinMode() 関数の指定を忘れないようにする。

2つの LED を点灯させる「暗さのしきい値」は場所によるが、グローバルな変数として以下の様に宣言しておくといよい。

```
int darkLevel = 1000; //暗さのしきい値
```

【課題 13】 プロジェクト名「kad13_CdSFading」

CdS センサーの値に応じて、I04 番ピンに接続した青色 LED が 0%～100%のデューティ比で点灯するスケッチを作成せよ。

《例えば》

- ・ CdS センサーのアナログ入力値が 0 のとき、青色 LED の PWM のデューティが 255 になり、
- ・ CdS センサーのアナログ入力値が 4095 のとき、青色 LED の PWM のデューティが 0 になる。

ほかにも

- ・ CdS センサーのアナログ入力値が 2047 のとき、青色 LED の PWM のデューティが 127 (50%) になる等。

比例配分 $0 : 255 = 4095 : 0$ となるような式が立てることができればよい。

ただし、整数値での比例配分には map() 関数という便利な関数がある。

Arduino 日本語リファレンス

map(value, fromLow, fromHigh, toLow, toHigh)

<http://www.musashinodenpa.com/arduino/ref/index.php?f=0&pos=2743>

できればこの map() 関数を使ってほしい。

動作確認するとき、実際の LED 点灯状況だけでは、明るさがわかりにくいので、シリアルモニタにも出力するようにする。loop() 関数内では以下の Serial.printf() 関数を使うといよい

```
Serial.printf("%d,%d\n", a, pwm);
```

ここで変数 a はアナログ入力値、変数 pwm は、比例配分が済んだデューティの値(変数名は自由)。

シリアルモニタ	シリアルモニタ (つづき)
2839, 79 ←通常の光だと青色LEDは少し光る	1981, 132
2559, 96	2106, 124
2511, 99	2237, 116
2431, 104	2352, 109
2367, 108	2415, 105
2320, 111	2449, 103
...	2491, 100
1136, 185 ←光を遮ったとき PWM は増える	2683, 88
1120, 186	2559, 96
1154, 184	2589, 94
1023, 192	3723, 24 ← スマホのライトをあてると
994, 194	3675, 27 どんどん PWM の値が減っていく
778, 207	3803, 19
682, 213	3815, 18
560, 221	4063, 2
544, 222 ←センサを指でつまむと	4005, 6
...	4027, 5
720, 211	4047, 3
726, 210	4087, 1
791, 206	4095, 0
1030, 191 ←指を離すとアナログ入力値は上がり	4085, 1
1520, 161 PWM は小さくなる	4095, 0 ←アナログ入力値が最大の 4095 に
1819, 142	4095, 0 なれば PWM は 0 になる。

最後にシリアルプロッタの画面を表示して以下の様な 2 点プロットのグラフになるか確認してください。

