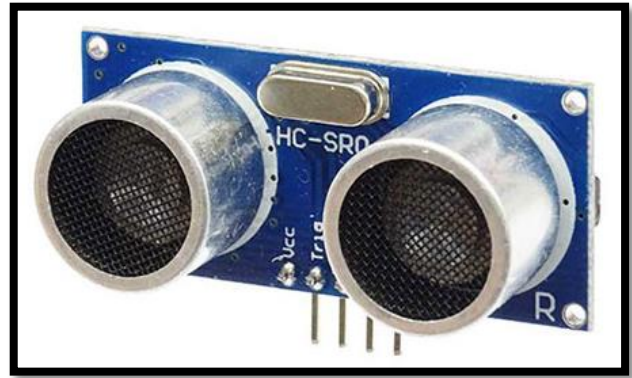


超音波距離センサーとは

超音波距離センサーとは、超音波の反射時間を利用して非接触で測距が行える電子部品のこと。Trigger により超音波を送信し、対象物に当たって反射した超音波を Echo が受信する。その受信した時間から距離へと変換することで距離を測定する。それに電源と GND を加えて 4 本の線で制御する。



マクロとは

プログラミング言語におけるマクロとは、プログラム内の文字列をあらかじめ定義した規則に従って置き換えられる機能のこと。`#define` 修飾子でマクロ化ができ、主にピン番号を定義するのに使用する。変数や定数とは違いデータ型を持たず、コンパイル前に置き換えられるので「値を保持する」ことは出来ないが、その分メモリ領域を割かず別な文字列として扱うことができ、可読性が向上する。

```
1 #define LED_PIN 25          // マクロ
2 int ledPin = 25;            // 変数
3 const int LED_PIN = 25;     // 定数
4
```

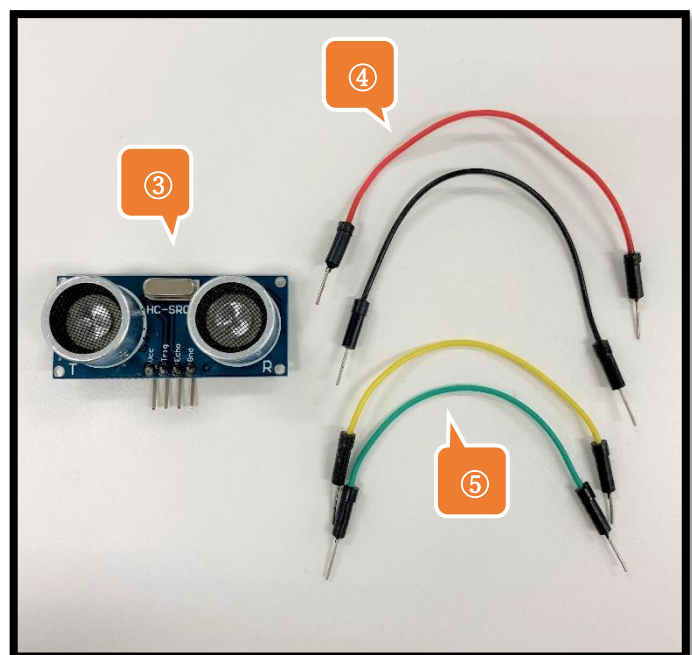
LED のピン番号を定義した例
プログラム内で LED_PIN と記述
すると、25 に置き換えられる

機材準備

今回使用する機材を所定の位置から借りて持って来る。※授業終了時には必ず元の場所に戻すこと。

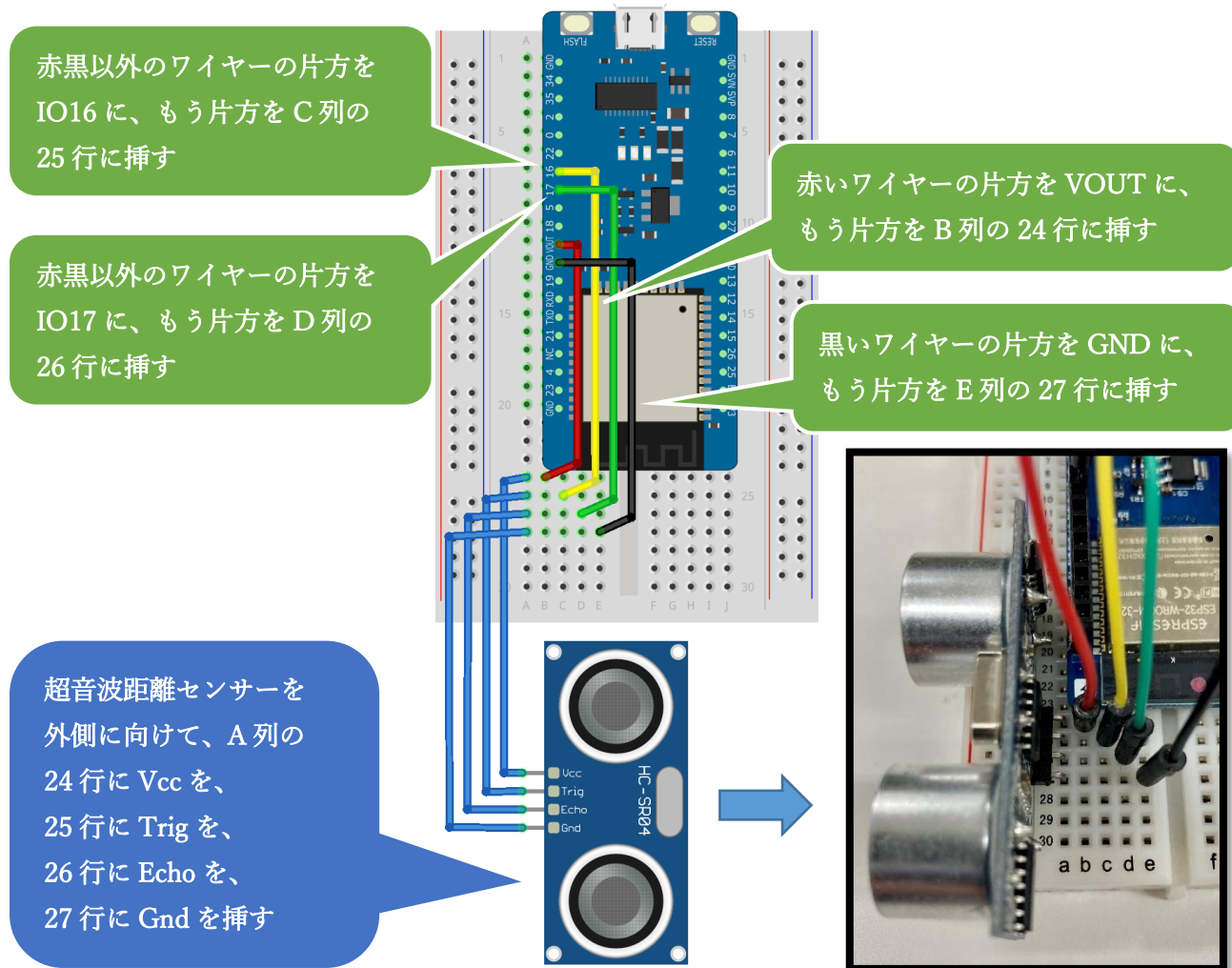
- ① ESP32 (ブレッドボード付)
- ② Micro-B USB ケーブル
- ③ 超音波距離センサー
- ④ ジャンパーワイヤー (赤と黒を 1 本ずつ)
- ⑤ ジャンパーワイヤー (赤・黒以外を 2 本)

※右の図にはないが、①と②は毎回必要
自分の番号のものを持って来ること



練習 (Distance)

ESP32 に Micro-B USB ケーブルを差し込まずに以下の図のとおり配線を行いなさい。



配線を行った後、Micro-B USB ケーブルを ESP32 と実習機に繋ぎ通電する。

その後、以下のサンプルコードを記述し、ESP32 に書き込みを行う。

超音波距離センサーから 1 秒毎に距離を取得し、シリアルモニタ上に出力されているか確認すること。

```
11:09:39.849 -> 距離 : 42.76cm
11:09:40.835 -> 距離 : 42.69cm
11:09:41.866 -> 距離 : 56.59cm
11:09:42.843 -> 距離 : 57.78cm
11:09:43.876 -> 距離 : 56.54cm
11:09:44.860 -> 距離 : 54.45cm
```

☒ 自動スクロール ☒ タイムスタンプを表示 CRおよびLF 115200 bps

```
/*
  Distance
  Date : 2022/01/01
  Author : IE1A 99 K.Murakami
*/

// ピン番号をマクロで定義
#define TRIG_PIN 16 // トリガー（送信）
#define ECHO_PIN 17 // エコー（受信）

double duration = 0; // 受信までの時間
double distance = 0; // 物との距離

// 起動時に一度だけ呼び出されるメソッド
void setup() {
  Serial.begin(115200); // シリアル通信の転送レート（bps）を設定

  // ピンの入出力設定
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  // トリガーを LOW に落としておく
  digitalWrite(TRIG_PIN, LOW);
  delay(10); // 落ちきるまで少し待機
}

// 超音波送信メソッド
void sendTrigger() {
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10); // 送信してから一瞬だけ待機
  digitalWrite(TRIG_PIN, LOW);
}

// メインループメソッド
void loop() {
  sendTrigger(); // 超音波送信

  duration = pulseIn(ECHO_PIN, HIGH); // 受信した時間を代入
```

マクロはヘッダコメントの下、
プログラムの最上部に定義する
マクロにセミicolon(;)は不要

```
// 値が0より大きい場合（受け取れていたら）
if (duration > 0) {
    duration /= 2; // 往復の距離を片道に変換
    // 音速を 340m/s として受信時間から距離に変換
    distance = duration * 340 / 10000;
    // シリアルモニタに出力
    Serial.print("距離: ");
    Serial.print(distance);
    Serial.println("cm");
}

delay(1000); // 1 秒待機
}
```

Java とは違い、連結演算子 (+) が使用できないので
3 行に分けて記述している (println は出力後に改行)
1 行にまとめることも出来るので、気になった人は
printf 文や strcat 関数を検索してみましょう

書き込んだら超音波距離センサーの前に手をかざしたり、物を置いたりして距離を測ってみましょう。
反射の都合上、1~2cm 未満は上手く距離が取得できず、1m を超えると誤差が生じてきます。

課題 (DistanceLit)

前述の練習を **別名保存** して超音波距離センサーから 1 秒間隔で物との距離を取得した後、その距離が **30cm 以内であれば LED を点灯し、30cm を超えていたら LED を消灯する** プログラムに変更しなさい。
LED の回路を追加する必要がある為、追加で必要な部品を考えて所定の位置から持って来ること。
配線を行う際は Micro-B USB ケーブルを ESP32 に **差し込まず** に行うこと。

※前回の課題同様、LED の回路は **ある部分に注意しないと LED 本体が壊れてしまいます**。

絶対に壊さないという自信がある人以外は、配線後に講師を呼んで配線のチェックをしてもらうこと。
また、以下の 5 つの条件を満たすこと。

- ① 超音波距離センサーの回路は練習のまま変更しないこと
- ② LED は IO25 番から出力して制御すること
- ③ LED のピン番号はマクロ名 (LED_PIN) で定義して使用すること
- ④ LED の抵抗値は 1kΩ (茶・黒・**赤**・金) にすること
- ⑤ LED 回路のジャンパーワイヤーの色は出力側を赤黒**以外**にして、GND 側を黒にすること

※課題が完成した人は ESP32 にプログラムを書き込んだ状態で講師を呼び、チェックしてもらうこと。

授業終了前に

前回同様に使用した機材や部品類は元あった場所に必ず戻し現状復帰すること。

ブレッドボード上の部品類やワイヤーは外し忘れることが多い為、しっかり確認して元に戻すように。