



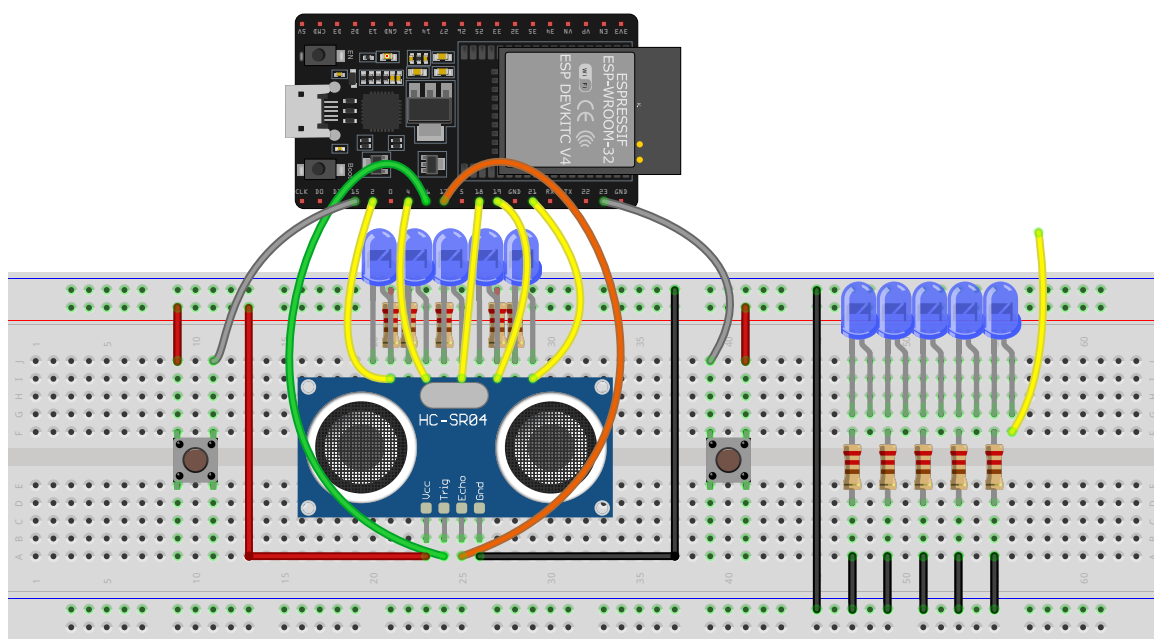
装置名	ULTRASONIC
装置番号	icc2025-003
作問者	Norimasa TAKANA
作問日	2025-11-08
制限時間	15m00s

1. 作問者より一言

2つの目玉のようなものが付いた、またもや謎の部品。一体なにを測っているんだ……？ プログラムを見てみると、何かの信号が返ってくるのを待ってる？

2. 回路

回路の配線図を図2に示す。



↑LEDの部分は分かりづらいですがこうなってます。

fritzing

図2: 配線図

図2の下側にある青色の大きな部品が、実際の装置の緑色の部品（形は同じ）に相当する。図と実際のワイヤの位置が違うが、ブレッドボードの縦5つの穴は全て繋がっているので電気的には同じである。

また、図の都合上LEDの配線が見づらいので図2の右側に等価な回路を載せた。マイコンの出力ピン → LEDの+ → LEDの- → 抵抗 → GNDとなっている。

回路に使われている部品の一覧を表 1 に示す.

表 1: 部品一覧

部品名	個数	概要
タクトスイッチ	2	別紙参照
LED	5	—
???	1	図 3 と別紙参照

謎の部品については解き方を見ながらその正体を一緒に考えてほしいが,LED がヒントになっている. LED の色は特に意味があるわけではない.

3. ソースコード

装置に書き込まれたプログラムを以下に示す.

cpp

```
1 struct Challenge Ultrasonic = {
2     .gaming = ultrasonic,
3     .setup_pin = setup_dist,
4     .time_limit = 900,
5 };
6
7 // giver pin assign
8 const uint8_t TRIG = 16;
9 const uint8_t ECHO = 17;
10 const uint8_t LED1 = 2;
11 const uint8_t LED2 = 4;
12 const uint8_t LED3 = 18;
13 const uint8_t LED4 = 19;
14 const uint8_t LED5 = 21;
15 const uint8_t LEFT_BUTTON = 15;
16 const uint8_t RIGHT_BUTTON = 23;
17
18 void setup_dist(void) {
19     pinMode(TRIG, OUTPUT); // 出力のためのピン
20     pinMode(LED1, OUTPUT); // 出力のためのピン
21     pinMode(LED2, OUTPUT); // 出力のためのピン
22     pinMode(LED3, OUTPUT); // 出力のためのピン
23     pinMode(LED4, OUTPUT); // 出力のためのピン
24     pinMode(LED5, OUTPUT); // 出力のためのピン
25     pinMode(ECHO, INPUT); // 入力のためのピン
26     pinMode(LEFT_BUTTON, INPUT_PULLDOWN); // ボタンの入力を読むためのピン
27     pinMode(RIGHT_BUTTON, INPUT_PULLDOWN); // ボタンの入力を読むためのピン
28 }
29
30 void ultrasonic(void *pvParameters) {
31     bool flag1 = false;
32     bool flag2 = false;
33     bool flag3 = false;
34     float duration, cm;
35
36     while(1) {
```

```

37 flag1 = (digitalRead(RIGHT_BUTTON) == HIGH);
38 flag2 = (digitalRead(LEFT_BUTTON) == HIGH);
39
40 digitalWrite(TRIG, LOW);
41 delayMicroseconds(2);
42 digitalWrite(TRIG, HIGH);
43 delayMicroseconds(10);
44 digitalWrite(TRIG, LOW);
45
46 // ECHO が HIGH になっている時間を返す (マイクロ秒).
47 duration = pulseIn(ECHO, HIGH);
48 // duration を 2 で割って係数 0.0344 をかける.
49 // 係数 0.0344 は  $331.5 * (0.61 * t)$  (t は摂氏温度, 今回は20度と想定)
50 // の結果の単位を合わせたもの.
51 // これは一般的に約 340 m/s と習う「あの」速さを
52 // 温度に合わせて厳密に計算したものです.
53 cm = (duration / 2) * 0.0344;
54
55 // 先に LED1~LED5 までのピンを全部 LOW にしておく
56 digitalWrite(LED1, LOW);
57 digitalWrite(LED2, LOW);
58 digitalWrite(LED3, LOW);
59 digitalWrite(LED4, LOW);
60 digitalWrite(LED5, LOW);
61 if (10 <= cm) { // もし cm の値が 10 以下だったら
62 |   digitalWrite(LED1, HIGH); // LED1 番のピンを HIGH に
63 | }
64 if (15 <= cm) { // もし cm の値が 15 以下だったら
65 |   digitalWrite(LED2, HIGH);
66 | }
67 if (20 <= cm) { // もし cm の値が 20 以下だったら
68 |   digitalWrite(LED3, HIGH);
69 | }
70 if (25 <= cm) { // もし cm の値が 25 以下だったら
71 |   digitalWrite(LED4, HIGH);
72 | }
73 if (30 <= cm) { // もし cm の値が 30 以下だったら
74 |   digitalWrite(LED5, HIGH);
75 | }
76
77 if (flag1 && 15 < cm && cm <= 20) {
78 |   flag3 = true;
79 | }
80
81 // succeeded
82 if(flag3) {
83 |   succeeded();
84 | }
85
86 // failed
87 if(flag1 && flag2) {
88 |   failed();

```

```
89     }  
90  
91     delay(100);  
92 } // while (true) { の行まで戻る  
93 }
```

cm の値が爆弾解除の鍵のようですが, 具体的に今 cm がいくつなのか分かりません. これでは解きづらいので今回は LED とリンクさせてみました. 是非試してみてください.

4. 解き方

4.1. 配線と部品を確認する

今回はかなり配線の数が多く苦労しそうです。表1を見ると、謎の部品・LEDが5つとボタンが2つという構成のようです。この問題もボタンを押したり、謎の部品を使って解除の条件を満たしていけば良さそうです。

やってみよう

- 手元の装置を見て、謎の部品・左のボタン・右のボタンがマイコンのピンの何番に繋がっているか調べよう。
 - ▶ 謎の部品: _____
 - ▶ 左のボタン: _____
 - ▶ 右のボタン: _____
- プログラムを見て、SENSOR, BUTTON_LEFT, BUTTON_RIGHT がそれぞれ何番を表すのか調べよう。
 - ▶ SENSOR: _____
 - ▶ BUTTON_LEFT: _____
 - ▶ BUTTON_RIGHT: _____

次にボタンとワイヤの関係も見ておきましょう。ボタンを押すと電気はどちらに流れていきますか？

やってみよう

ペンを使って Vcc からピンまでの流れを塗ってみよう。

4.2. 解除成功・失敗の条件を調べる

次に、書かれているプログラムを見てみると、flag1 と flag2 と flag3 があります。flag1, flag2, flag3 はデータを入れる変数というものでした。この変数には true / false のどちらかが入ります。

flag1 と flag2 は前の問題と同じような形ですが、flag3 は多少複雑です。

```
30 if (flag1 && 15 < cm && cm <= 20) {  
31     flag3 = true;  
32 }
```

cpp

&& は“かつ”（英語で AND）という意味です。例えば A && B とあったら A と B 両方が true であるときに true という意味です。今回は && が2つあるので A && B && C となり A と B と C 全部が true であるときに true という意味になります。

つまり、以下の条件全てが true（成り立つ）とき flag3 が true になります。

条件 1: flag1 が true

条件 2: cm に入っている値が 15 より大きい。

条件 3: cm に入っている値が 20 以下である。

同様に失敗の条件も探してみましょう。

4.3. 謎の部品の正体を探る

後はこの 2 つの目玉のようなものが付いた部品の正体です。そもそこの部品（センサ？）はどうやって値を取得しているのでしょうか？ さっきの問題は片方が Vcc(+) に片方が入力ピンに接続されていて、条件が揃うと Vcc と 入力ピンが繋がって HIGH が入力されるという仕組みでした。今回のセンサは出力と入力のピン両方が繋がっています。

cpp

```
1 pinMode(TRIG, OUTPUT); // 出力のためのピン
2 pinMode(ECHO, INPUT); // 入力のためのピン
```

TRIG が出力 ECHO が入力です。これはセンサの正面にも書かれています。 TRIG はおそらく“trigger”の略で、銃の引き金や何かを動作させるための信号のことを指します。ECHO は“echo”，つまり山びこのことです。

cpp

```
40 digitalWrite(TRIG, LOW);
41 delayMicroseconds(2);
42 digitalWrite(TRIG, HIGH);
43 delayMicroseconds(10);
44 digitalWrite(TRIG, LOW);
```

まず TRIG を LOW にして $2\mu s$ 待ってそこから $10\mu s$ の間 HIGH にします。この $10\mu s$ が trigger となる信号なのです。

cpp

```
45 // ECHO が HIGH になっている時間を返す（マイクロ秒）。
46 duration = pulseIn(ECHO, HIGH);
47 // duration を 2 で割って係数 0.0344 をかける。
48 // 係数 0.0344 は  $331.5 * (0.61 * t)$  (t は摂氏温度、今回は20度と想定)
49 // の結果の単位を合わせたもの。
50 // これは一般的に約 340 m/s と習う「あの」速さを
51 // 温度に合わせて厳密に計算したものです。
52 cm = (duration / 2) * 0.0344;
```

そして今度は ECHO の入力を待ってます。 pulseIn は信号が HIGH になっている長さを返す関数です。データシートの図に意味を加えるようになってます。

The principle of distance measurement

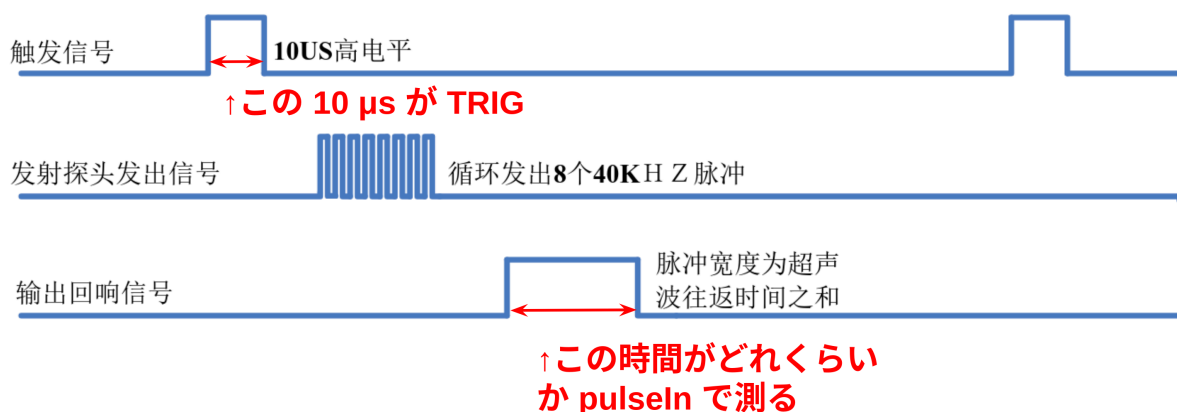


図 3: 信号の意味

謎の部品から返ってくる信号の長さ（図3一番下の山の幅）は何を表しているのでしょうか？ それはこの問題の名前に隠されています。“ultrasonic” これはズバリ「超音波」という意味です。

部品の正面に“T”と“R”という文字があるのに気づいたでしょうか？ これは“Transmit”（送信）と“Receive”（受信）という意味です。この部品は TRIG に信号を受け取ると“T”の方から 40 kHz の超音波を 8 回飛ばします。そして超音波はなにかにぶつかって返ってきます。これを“R”で検知します。

検知したら“T”から超音波を飛ばして“R”に返ってくるまでの時間を信号として ECHO に返します。謎の部品から返ってくる信号の長さ（図3一番下の山の幅）の正体はこれです。この時間がプログラムの中だと duration という変数に入っています。

ここまで来ると何故 duration が 2 で割られるのか、謎の係数 0.0344 とは何なのか、cm という変数が何を表しているのか分かります。

やってみよう

謎のセンサの仕組みが分かったら以下の空欄を埋めてみよう。

- duration が 2 で割られるのは、_____ からである。
- 0.0344 という係数は、_____ の速さと時間をかけることで _____ を求めるためのものである。
- つまり、このセンサは、_____ を測るためのものである。