

# 電気回路工作実習

ー マイコンを用いたタスクトレーニング回路製作 ー

トレーニングコーステキスト

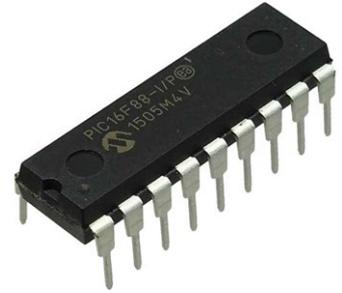
認知行動発達機構研究部門

## 1.目的

マイコンを用いた回路製作をすることでマイコンへの理解を深めるとともに回路工作に必要なハンダ付け、配線技術の習得を行う。

## 2.ワンチップマイコンとは

マイコン(マイクロコンピュータ)の一種で、ひとつの IC チップ上に CPU から RAM、ROM、各種入出力装置などを搭載した処理装置のことである。ワンチップマイコンは、汎用的な処理を行うことはできないが、小さな IC 回路のみで特定機能の処理を一手に行うことができる。そのためコンピュータ制御を必要とする装置の多くに組み込まれている。炊飯器や自動車の制御システムに採用されたり、あるいはマウスやキーボードにおける入力情報の制御などにもワンチップマイコンが用いられている。



### 2-1.代表的なワンチップマイコン

PIC と AVR が代表的なワンチップマイコンである。PIC (ピック) とは、Peripheral Interface Controller (ペリフェラル インターフェイス コントローラ) の略称であり、マイクロチップ・テクノロジー社 (Microchip Technology Inc.) が製造しているマイクロコントローラ (制御用 IC) 製品群の総称である。Atmel AVR (アトメル AVR) は、Atmel 社が製造している、RISC ベースの 8 ビットマイクロコントローラ (制御用 IC) 製品群の総称である。1996 年に開発された。

### 2-2.ワンチップマイコンを扱うには

IC 動作に最低限必要な、素子類 (発振・コンデンサ等) をつなげる必要があり、また単体でプログラム操作が出来ないため、PIC ライター等の装置が必要になる。



## 3.ワンボードマイコン (Arduino を例に)

ワンボードマイコンとは、むき出しの一枚(ワン)のプリント基板(ボード)の上に、電子部品と最低限の入出力装置を付けただけの極めて簡素なマイクロコンピュータである。

### 3-1.代表的なワンボードマイコン

最近多くのユーザーが使用している Arduino がある。

Arduino は、AVR マイコン、入出力ポートを備えた基板、C++ 風の Arduino 言語とその統合開発環境から構成されるシステムである。Arduino LLC および Arduino SRL が設計・製造を行い、登録商標を持っている。



### 3-2.ソフトウェア (統合開発環境) Arduino IDE

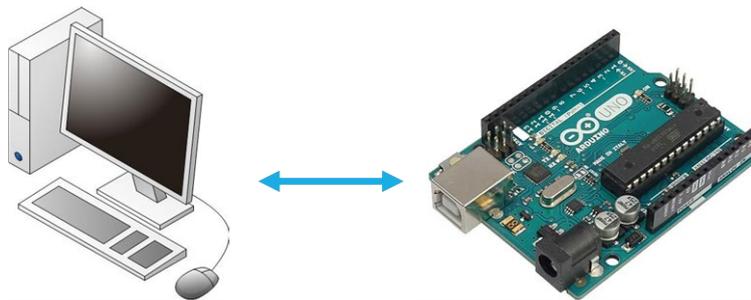
Arduino の統合開発環境はクロスプラットフォームの Java アプリケーションであり、エディタ、

コンパイラ、基板へのファームウェア転送機能などを含む。その内部ではC言語のコンパイラgccやアップロードプログラムavrdudeが使用されている。独有用語：Arduinoではプログラムをスケッチと呼ぶ。



### 3-3.ワンボードマイコン (Arduino) を扱うには

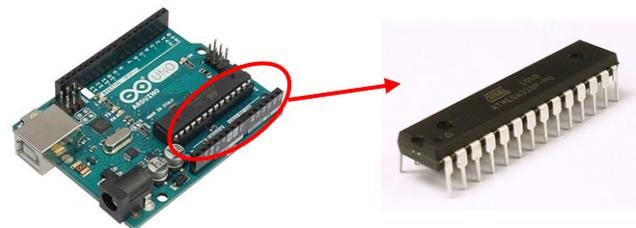
プログラムはUSB経由でArduinoにすることが出来る。USBでパソコンと接続すると電源供給も行い単体のボードで動作可能となる。



## 4.Arduino と AVR (ワンチップマイコン)

Arduino Uno はワンチップマイコン IC (AVR) の ATmega328P がメイン IC として搭載されている。

ATmega328P に Arduino Uno のブートローダーを書き込むと Arduino Uno のプログラムを実行することが可能となる。



Arduino Uno を IC ライターとして使用し、ATmega328P に Arduino Uno の統合開発環境からプログラムが可能となり、ATmega328P 単体で Arduino Uno のワンチップマイコンとして使用可能となる。(最低限必要な素子を接続する必要がある)

### 4-1.ブートローダー

「ブート」とはマイコンが起動することを意味する。「ローダ」とはプログラムやデータなどを、メモリに読み込むプログラムを意味する。それらが合わさった「ブートローダー」とは何を意味するのだろうか？結論から言うと、マイコンの起動時に、外部からプログラムをメモリに読み込むプログラムの事をブートローダーと言う。

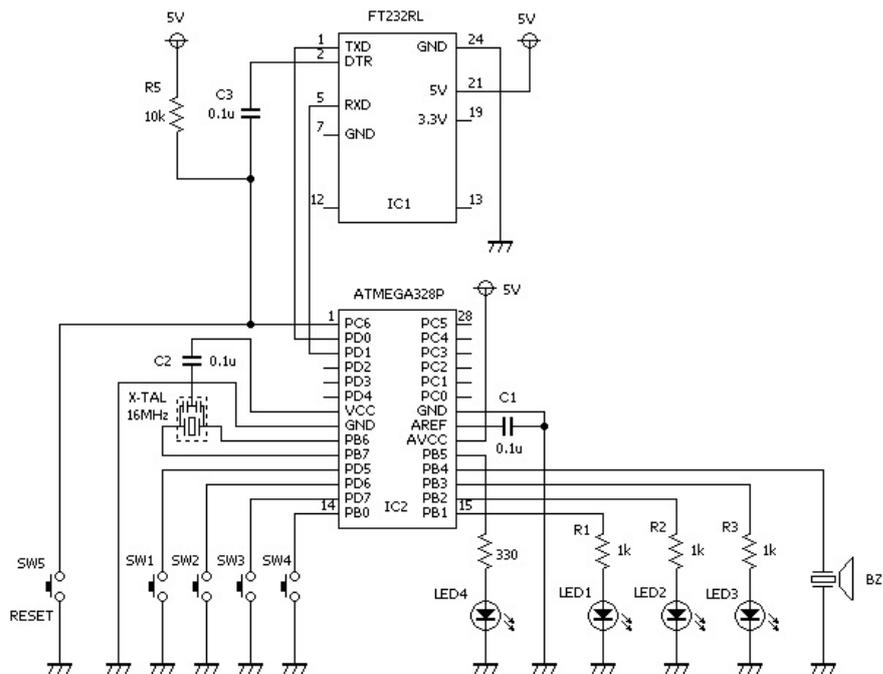
### 4-2.ブートローダーの書き込み

Arduino Uno を用いて ATmega328P にブートローダーを書き込む方法が一般的に多いようである。Arduino Uno 用ブートローダライターキットも販売している。

ウェブで検索するとその方法が多数出てくるので、色々試してみるとよい。

## 5.回路実習

今回、製作する回路を下記に示す。ワンチップマイコン ATmega328P を利用してタスクトレーニング回路を製作する。ここでは、スイッチ、LED、ブザーを利用してタスクのテスト回路を構成する。ATmega328P だけでは回路としては動作しないため、動作する回路を実際に製作してハンダ付け練習と ATmega328P を用いた回路実習、プログラム実習を行う。



製作する回路図

Arduino uno	ATmega328
Digital 0	2 (PD0)
Digital 1	3 (PD1)
Digital 2	4 (PD2)
Digital 3	5 (PD3)
Digital 4	6 (PD4)
Digital 5	11 (PD5)
Digital 6	12 (PD6)
Digital 7	13 (PD7)
Digital 8	14 (PB0)
Digital 9	15 (PB1)
Digital 10	16 (PB2)
Digital 11	17 (PB3)
Digital 12	18 (PB4)
Digital 13	19 (PB5)

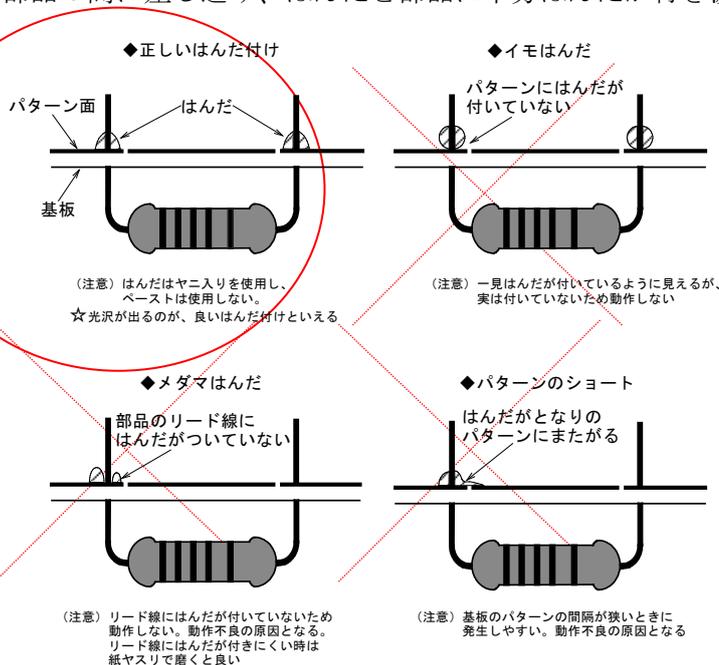
Arduino uno	ATmega328
GND	8,22 (GND)
AREF	21 (AREF)
SDA	27 (PC4/ADC4)
SCL	28 (PC5/ADC5)
Analog in 0	23 (PC0/ADC0)
Analog in 1	24 (PC1/ADC1)
Analog in 2	25 (PC2/ADC2)
Analog in 3	26 (PC3/ADC3)
Analog in 4	27 (PC4/ADC4)
Analog in 5	28 (PC5/ADC5)

Arduino uno と ATmega328 ピン配置対応表

## 6-1.はんだ付けの基本

はんだ付けする部品（リード線、電線、パターンなど）に一度コテ先をあててしばらく（数秒）待ったら、速やかにはんだをコテ先と部品の間に差し込み、はんだと部品に十分はんだが行き渡ったら直ちにはんだごてを離すのがコツである。この間数秒以内（はんだごてを部品にあてたり離したりする動作は、取り付け部やはんだの温度が十分に上がらずはんだ付け不良の原因となる）

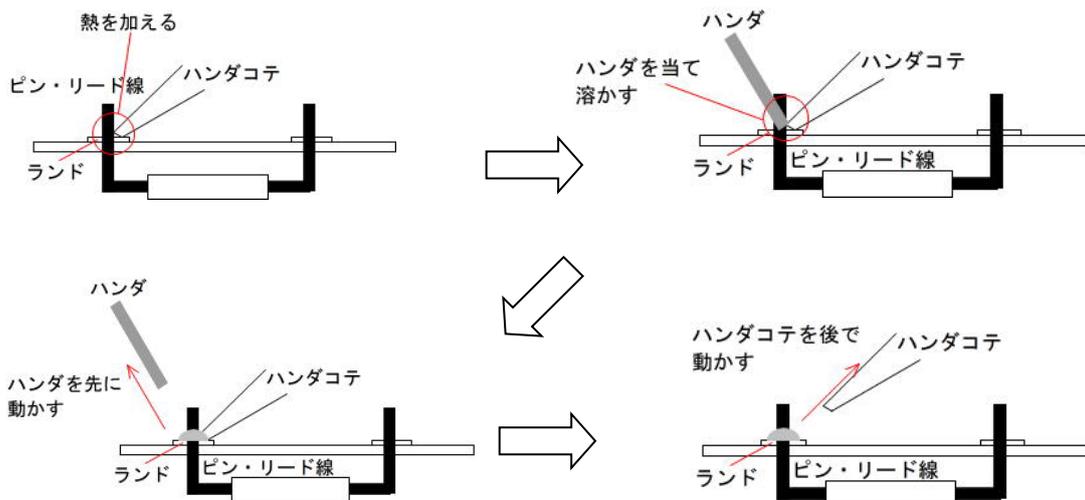
特に熱に弱いICやトランジスタはできるだけ手早く行うことが重要、はんだ付けをうまく行うには、数をこなすことが一番である。しかし、正しいはんだ付けを知らないとただ不良はんだを増産することになりかねない。右図に正しいはんだ付けと良くないはんだ付けの模式図を示す。



## 6-2.はんだ付けの実践

ハンダ面のランドとICソケットのピンにはんだコテ先を当てるように注意する。

熱が伝わったらハンダコテは動かさずに、ハンダをコテ先に当てハンダを溶かし、ハンダ付けが出来たら、先にハンダを動かしてからハンダコテを動かす。



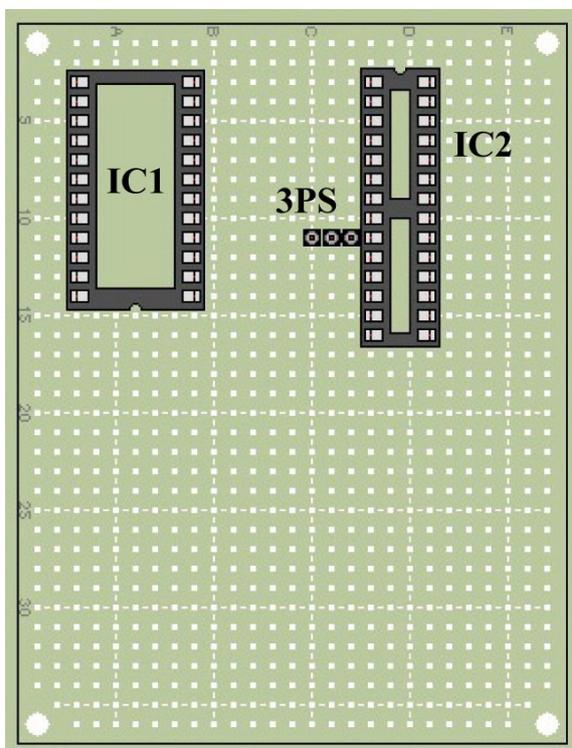
## 7.回路製作手順

### 手順1. ソケットの取り付け-電源ラインの接続

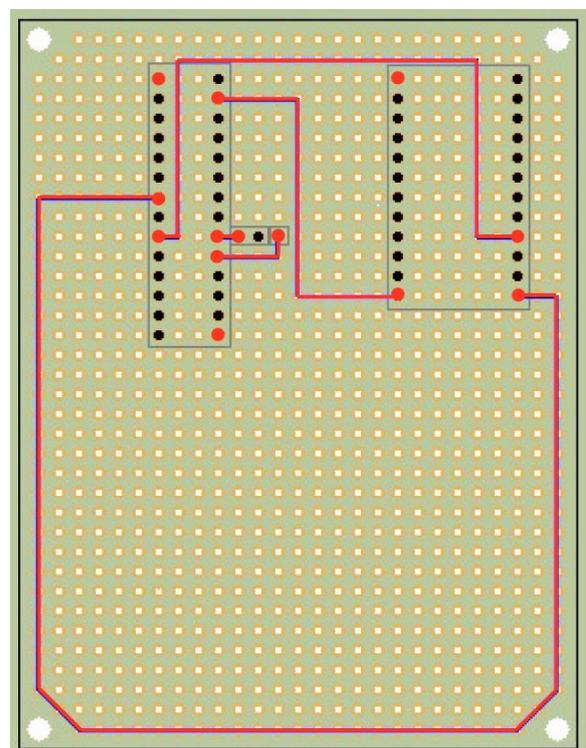
- ① ICソケット、PINソケットを図に示す所定の場所に配置し、対角の2点をハンダ付けする。
  - ・取り付け後、ソケットの浮きがないかチェックする。
- ② スズメッキ線を使い電源ラインを作る。
  - ・スズメッキ線は、ホールに沿わせできるだけまっすぐに取り付ける。
  - ・ソケットの電源ピンも一緒にハンダ付けする。

使用する部品

部品名	規格	個数	記号	備考
プリント基板	秋月 AE/B2 CEM-3	1		片面プリント
ICソケット	2列-26ピン端子	1	IC1	方向注意
ICソケット	2列-28ピン端子	1	IC2	方向注意
PINソケット	1列-3ピン端子	1	3PS	
スズメッキ線				



表面

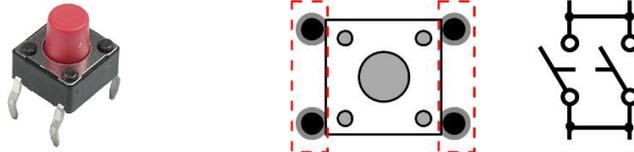


裏面

## 手順2. タクトスイッチの取り付け

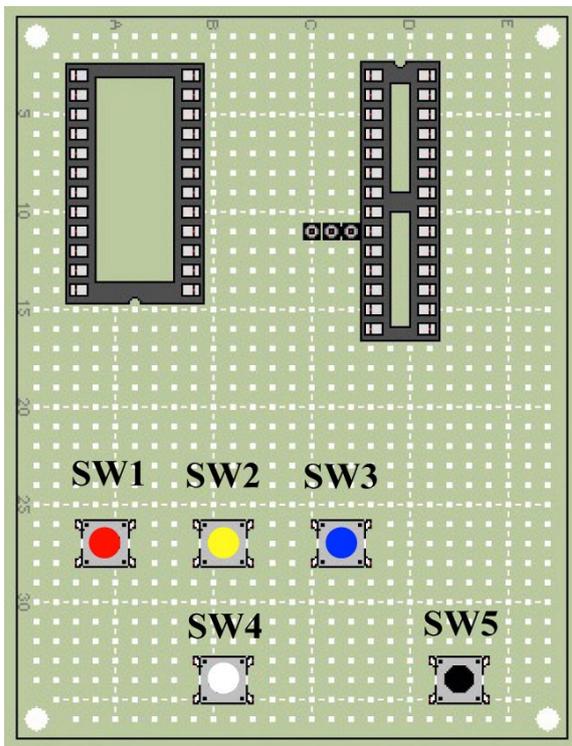
① タクトスイッチを図に示す所定の場所に配置しハンダ付けする。

- ・タクトスイッチは、色と方向を注意して取り付ける。
- ・取り付け後、IC端子とスイッチをスズメッキ線で結ぶ。
- ・スイッチとGNDラインをスズメッキ線で結ぶ。

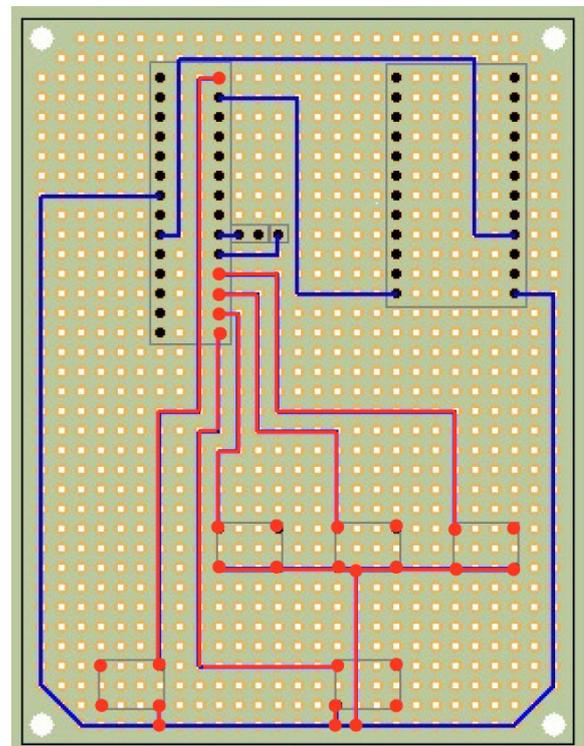


使用する部品

部品名	規格	個数	記号	備考
タクトスイッチ	赤,黄,青,白,黒	5	SW1-5	色、方向注意
スズメッキ線				



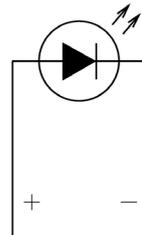
表面



裏面

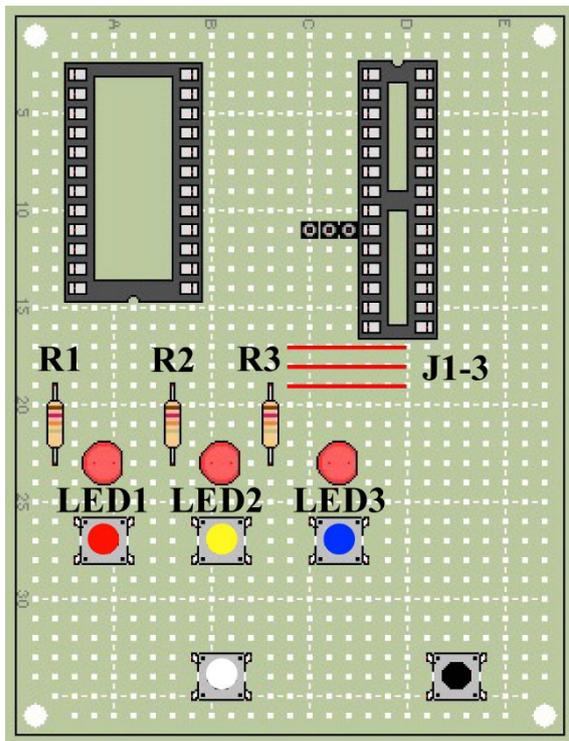
### 手順3. LEDと抵抗の取り付け

- ① LEDを図に示す所定の場所に配置しハンダ付けする。
  - ・LEDの端子は、極性があるので取り付け方向に注意する。（下図参照）
  - ・LEDの足を配線方向に折り曲げ、適当な長さで切る。
- ② 抵抗を取り付ける。
  - ・抵抗の足を配線方向に折り曲げ、適当な長さで切りハンダ付けする。
- ③ スズメッキ線でジャンパー線を作りJ1-3を取り付ける。
  - ・ジャンパー線は、ホールの長さに合わせてあらかじめ曲げておくとよい。

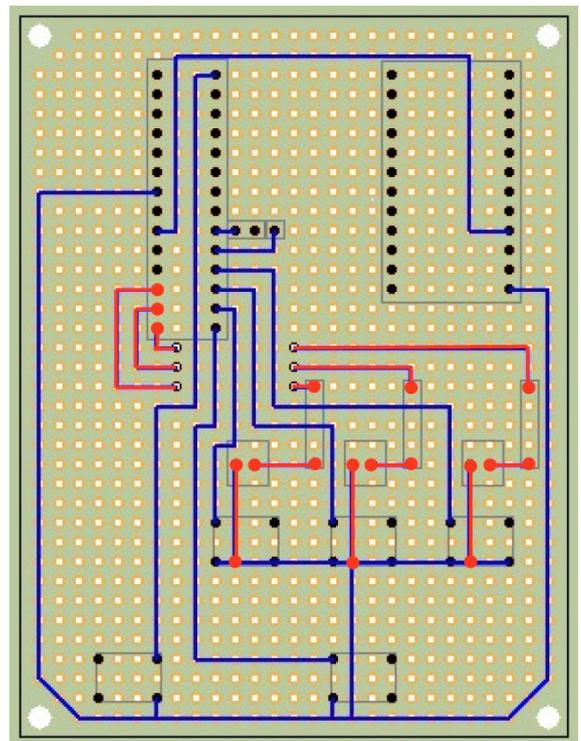


使用する部品

部品名	規格	個数	記号	備考
抵抗	1kΩ	3	R1-3	茶黒黒茶茶
LED	赤色	3	LED1-3	極性あり
ジャンパー線		3	J1-3	
スズメッキ線				



表面



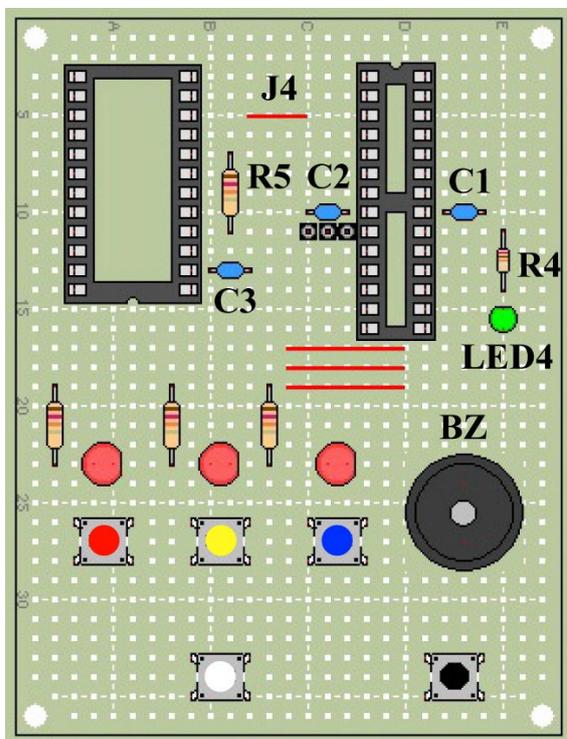
裏面

#### 手順4. 残りの部品の取り付け

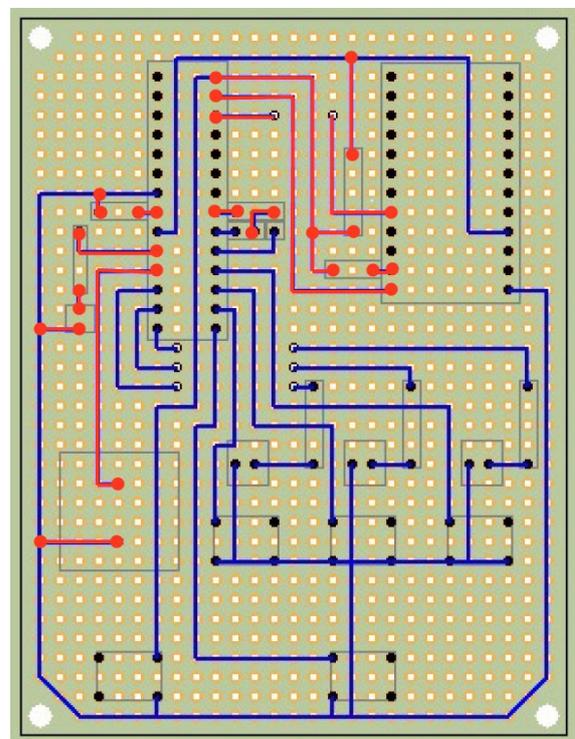
- ① ICソケット周りのコンデンサ、抵抗を所定の場所に配置しハンダ付けする。
  - ・抵抗、コンデンサの足を配線方向に折り曲げ、適当な長さで切りハンダ付けする。
- ② スズメッキ線でジャンパー線を作りJ4を取り付ける。
  - ・ジャンパー線は、ホールの長さに合わせてあらかじめ曲げておくとよい。
- ③ LED4を図に示す所定の場所に配置しハンダ付けする。
  - ・LEDの端子は、極性があるので取り付け方向に注意する。
- ④ 圧電ブザーを所定の場所に配置しハンダ付けする。
  - ・圧電ブザーの端子は、極性があるので取り付け方向に注意する。

使用する部品

部品名	規格	個数	記号	備考
抵抗	10k $\Omega$	1	R9	茶黒黒赤茶
抵抗	330 $\Omega$	1	R10	燈燈黒黒茶
コンデンサ(積層セラミック)	0.1 $\mu$ F	3	C1-3	104
LED		1	LED4	極性あり
圧電ブザー		1	BZ	極性あり
スズメッキ線				



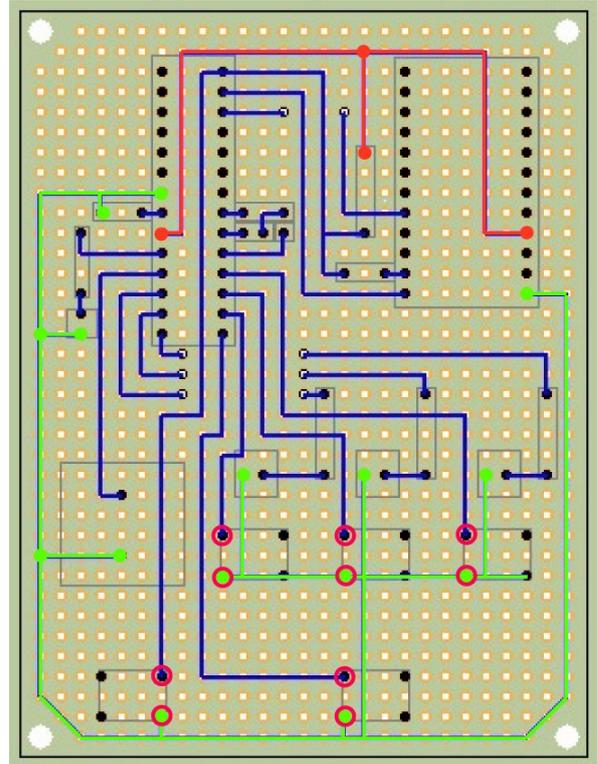
表面



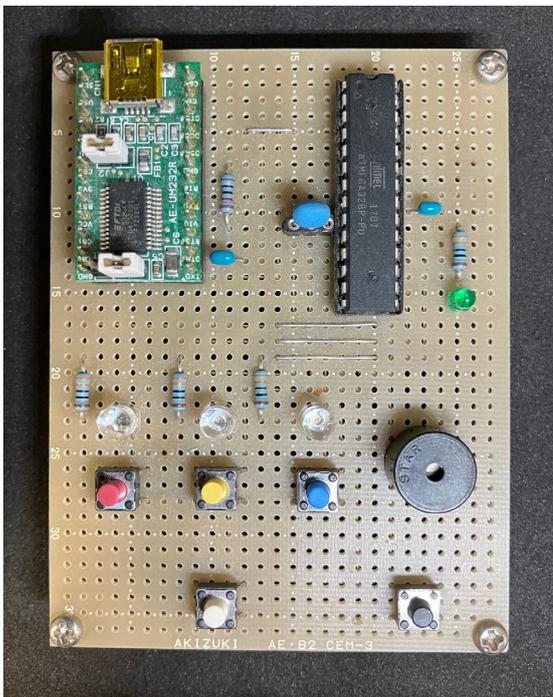
裏面

## 8.回路確認

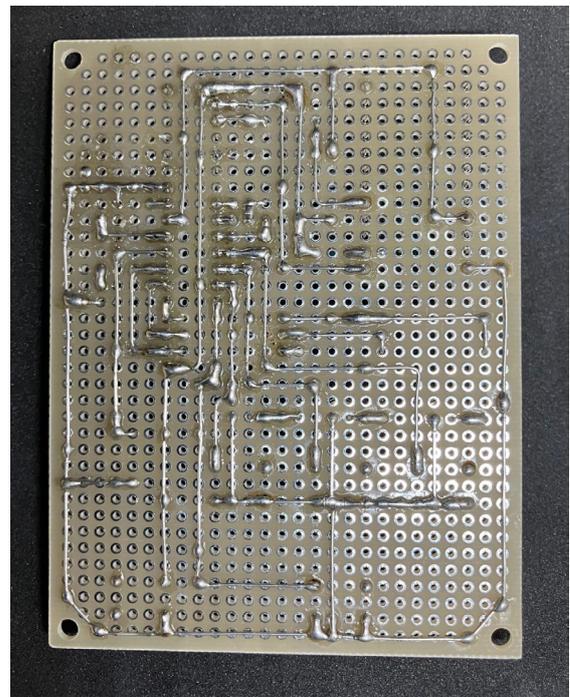
- 1) . 製作した回路と図面を参照し間違いがないか確認をする。
- 2) . テスターを抵抗にセットし、+5V電源ライン、グラウンドラインがショート（短絡）していないかを確認する。
- 3) . テスターを利用し、タクトスイッチの導通を確認する。（右図の赤丸印）
- 4) . USB ケーブルをパソコンと製作した回路に接続し、発光ダイオードが点灯するか確認をする。
- 5) . +5V電源ラインとグラウンドライン間の電圧を測定する。+5V付近であれば良い。



## 9.完成基板



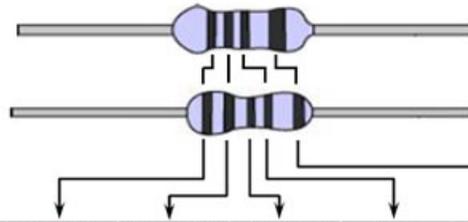
表面



裏面

## 9.部品表示の読み方

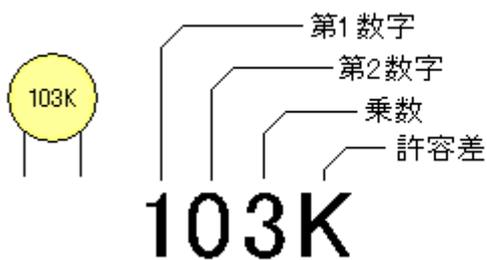
抵抗カラーコード表



色 Color	第1数字 1st figure	第2数字 2nd figure	第3数字 3rd figure	倍率 Multiplier	抵抗値許容差 Tolerance	記号 Symbol
黒 Black	0	0	0	1		
茶 Brown	1	1	1	10 <sup>1</sup>	±1%	F
赤 Red	2	2	2	10 <sup>2</sup>	±2%	G
橙 Orange	3	3	3	10 <sup>3</sup>		
黄 Yellow	4	4	4	10 <sup>4</sup>		
緑 Green	5	5	5	10 <sup>5</sup>	±0.5%	D
青 Blue	6	6	6	10 <sup>6</sup>	±0.25%	C
紫 Violet	7	7	7	10 <sup>7</sup>	±0.1%	B
灰 Gray	8	8	8		±0.05%	A
白 White	9	9	9			
金 Gold				10 <sup>-1</sup>	±5%	J
銀 Silver				10 <sup>-2</sup>	±10%	K
無 Plain					±20%	M

黒	0	黒い礼服
茶	1	小林一茶
赤	2	赤いエンジン
橙	3	ミカン
黄	4	四季
緑	5	五月ミドリ
青	6	青の6号
紫	7	紫式部
灰	8	ハイヤー
白	9	釧路

コンデンサの表示の読みかた



左記の例では

$$\begin{aligned}
 10 \times 10^3 &= 10 \times 1000 \text{ pF} \\
 &= 10000 \text{ pF} \\
 &= 0.01 \mu\text{F}
 \end{aligned}$$

許容差 ±10%

許容差の表示

F	±1%
G	±2%
H	±3%
J	±5%
K	±10%
M	±20%
N	±30%

P	0 ~ +100%
Q	-10 ~ +30%
S	-20 ~ +50%
T	-10 ~ +50%
Z	-20 ~ +80%

(10pF以下のもの)

B	±0.1 pF
C	±0.25 pF
D	±0.50 pF
E	±2.00 pF
F	±1.00 pF

## 10.Arduino スケッチプログラミング

Arduino アイコンをダブルクリックし、立ち上げる。

※ 開発環境(Arduino IDE)は <http://www.arduino.cc/>から無料で手に入れることができる。

Arduino IDE の画面説明を下記に示す。

### サンプルスケッチの書き込み (LEDの点灯)



**各機能**

- ✓ コンパイル
- ➡ ボードに書き込む (コンパイルも実行)
- 📄 新規作成
- 🔍 開く
- 💾 保存

**STEP③ Arduino Unoに書き込み**

- ➡ をクリックしボードに書き込む

**STEP④ 動作確認**

- 動作確認用LEDが点滅しているかを確認

**※書き込みエラーがでるときは**

- [ツール]-[シリアルポート]のCOM番号が右下の表示と一致しているかを確認し、一致していない場合は修正する

10-1.次のスケッチを入力しスイッチ入力と LED の点灯を確認しよう。

```
01 void setup()
02 {
03     pinMode(8,INPUT);           //8番ピンを入力として利用することを宣言
04     pinMode(10,OUTPUT);        //10番ピンを出力として利用することを宣言
05 }
06
07 void loop()
08 {
09     if(digitalRead(8)==HIGH);  //if文 8番ピンに値を読み取り条件分岐
10     {
11         digitalWrite(10,HIGH); //8番ピンがHIGHなら10番ピンのLEDを点ける
12     }
13     else
14     {
15         digitalWrite(10,LOW); //8番ピンがLOWなら10番ピンのLEDを消す
16     }
17 }
```

### `pinMode(pin, mode)`

ピンの動作を入力か出力かに設定する。

[パラメータ]

pin : 設定したいピンの番号

mode : INPUT (入力) か OUTPUT (出力)

### `digitalWrite(pin, value)`

HIGH または LOW を指定したピンに出力する。

[パラメータ]

pin : 出力するピンの番号

value : HIGH か LOW

### `digitalRead(pin)`

指定したピンの値を読み込む。結果の値は HIGH か LOW となる。

[パラメータ]

pin : 読みたいピン番号

### `delay(ms)`

プログラムを指定した時間だけ停止する。

[パラメータ]

ms : 一時停止する時間。単位はミリ秒。1秒は1000ミリ秒。

## 10-2.ピン対応表をもとにタスクプログラムを作ってみよう。

ピン対応表

入出力	対応ピン番号
ホームスイッチ (HOME_SW)	8
ターゲットレフトスイッチ (SW_L)	5
ターゲットセンタースイッチ (SW_C)	6
ターゲットライトスイッチ (SW_R)	7
ターゲットレフト LED (LED_L)	9
ターゲットセンターLED (LED_C)	1 0
ターゲットライト LED (LED_R)	1 1
圧電ブザー (buzz)	1 2