

# Arduino を使って ミニ電光掲示板をつくらう

神奈川工科大学 OpenCampus ミニ体験授業

2011 年 10 月 15 日(土)

神奈川工科大学 情報学部

情報メディア学科 助教

小坂 崇之

# 1 必要なもの

## 1.1 必要なもの(ハードウェア)

体験授業に必要なハードウェアは以下の通りです。必要なものが揃っているか確認してください。不足している場合は、お知らせください。

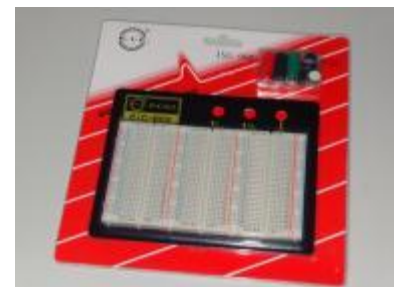
必要機材	型番	価格	個数
Arduino	Arduino UNO	2995 円	1 台
USB ケーブル	A-B タイプ	-	1 本
ブレッドボード	EIC-803	900 円	1 台
ジャンパーワイヤー	-	-	適量
LED	-	-	適量
LED マトリックス	SZ421288K	100 円	1 台



Arduino UNO



USB ケーブル



ブレッドボード



ジャンパーワイヤー



LED



LED マトリックス

これらの部品は、下記のサイトで購入することができます。

スイッチサイエンス	<a href="http://www.switch-science.com/">http://www.switch-science.com/</a>
秋月電子通商	<a href="http://akizukidenshi.com/catalog/default.aspx">http://akizukidenshi.com/catalog/default.aspx</a>
aitendo@shopping	<a href="http://www.aitendo.co.jp/">http://www.aitendo.co.jp/</a>

## 1.2 必要なもの(ソフトウェア)

実習を始めるのに、「Arduino Software」というソフトウェアが必要になります。今回の体験授業ではコンピュータに既にインストールされています。このソフトを用いて Arduino の開発を行うことができます。Windows, MacOS X, Linux などの異なる OS でも開発することができます。

「Arduino Software」は、下記のサイトでダウンロードすることができます。

Arduino 0022

<http://arduino.cc/en/>

[Buy](#) [Download](#) [Getting Started](#) [Learning](#) [Reference](#) [Hardware](#) [FAQ](#)

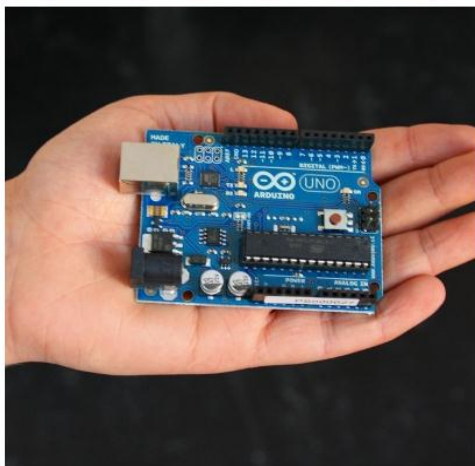


Photo by the Arduino Team

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

The boards can be **built by hand** or **purchased** preassembled; the software can be **downloaded** for free. The hardware reference designs (CAD files) are **available** under an open-source license, you are free to **adapt them** to your needs.

<http://arduino.cc/en/>

## 2 セットアップ

### 2.1 Arduino と PC の接続

1. Arduino を箱から取り出します。



2. USB ケーブルを Arduino と接続します。

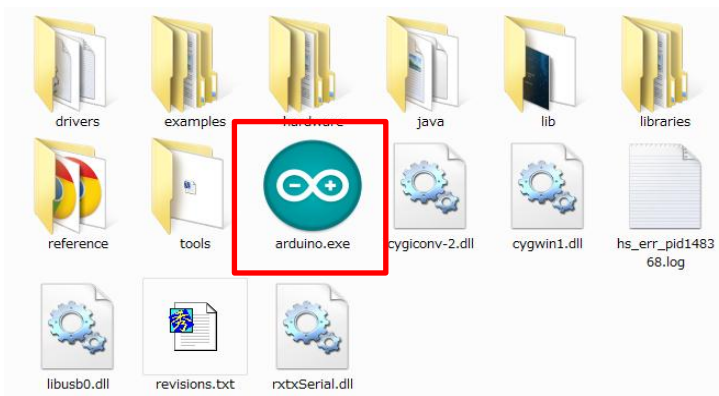


3. もう片方の USB ケーブルを PC 側に接続します。



### 2.2 ArduinoSoftware の起動

1. デスクトップにある「arduino-0022」フォルダを開きます。
2. 「arduino.exe」を実行します。

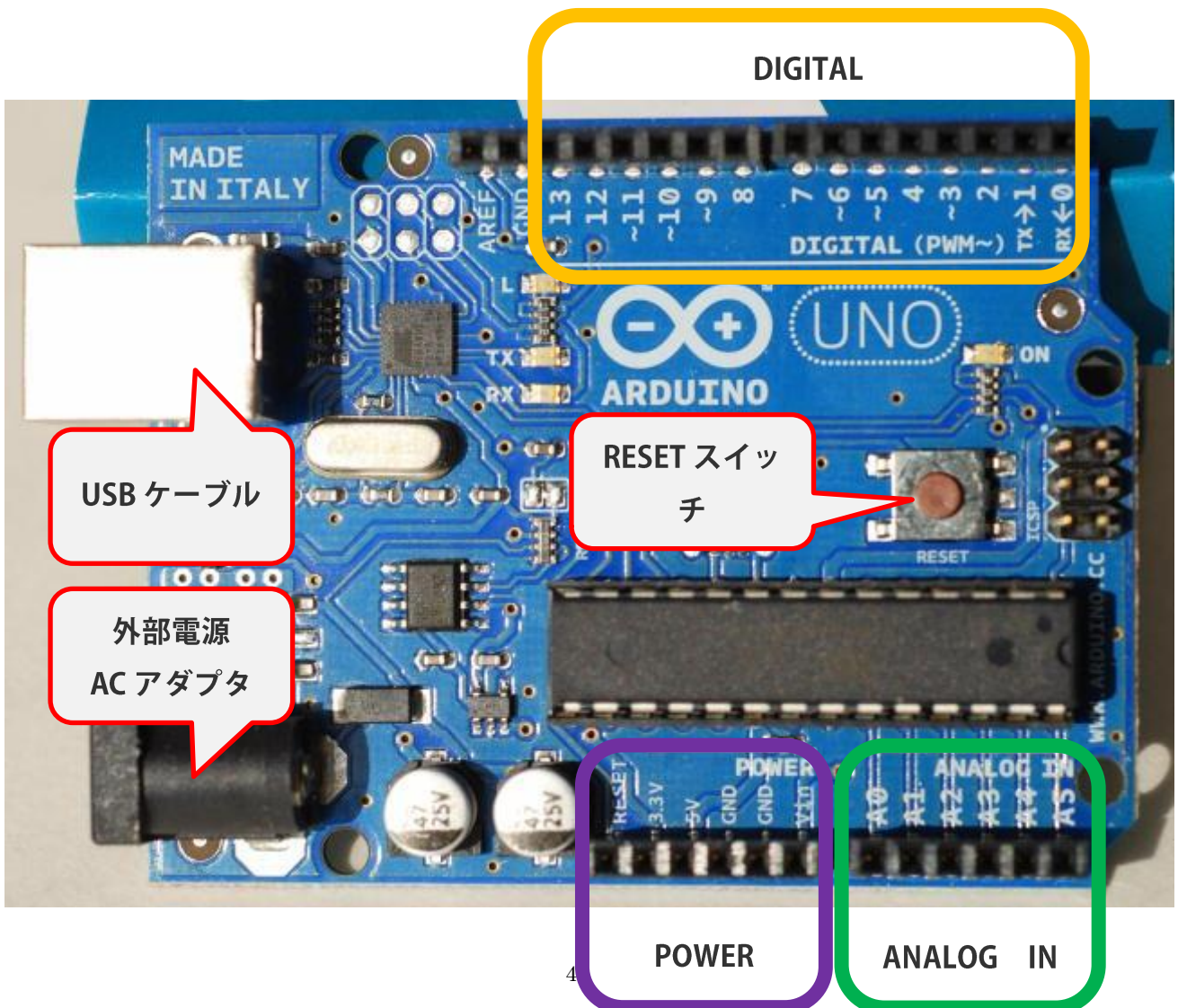


## 3 見てみよう

### 3.1 Arduino UNO を見てみよう

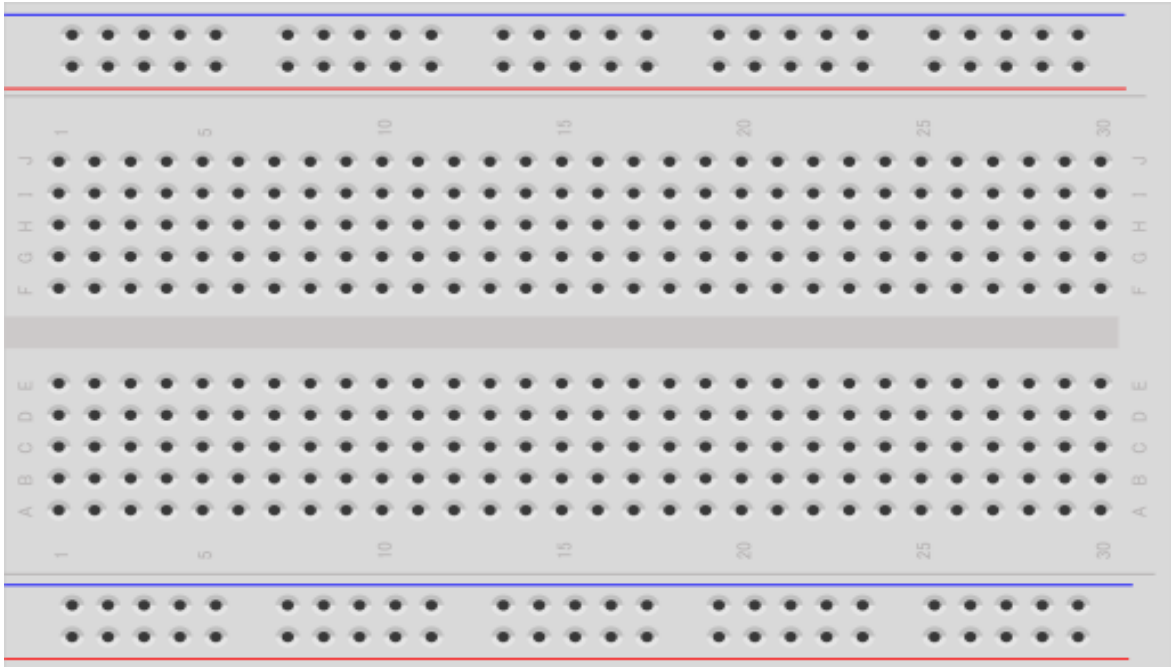
Arduino(アルドゥイーノ)は、AVR マイコン、入出力ポートを備えた基板、C 言語風の Arduino 言語とその統合開発環境から構成されるシステム。Arduino はスタンドアロン型のインタラクティブデバイス開発だけでなく、ホストコンピュータ上のソフトウェア（例えば、Adobe Flash、Processing、Max/MSP、Pure Data、SuperCollider）で制御することもできる。組み立て済みの基板を購入することもできるが、オープンソースハードウェアであり、ハードウェア設計情報の EAGLE ファイルは無料で公開されており、誰でも自分の手で Arduino を組み立てることができる。

Wikipedia より参照

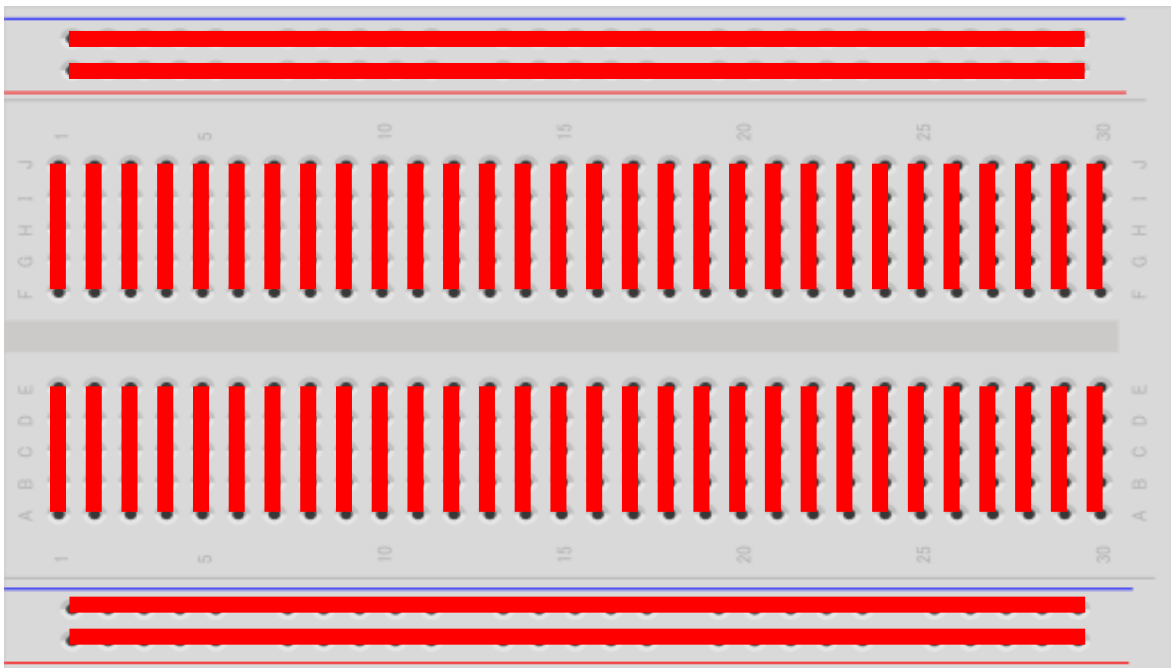


# 3.2

## ブレッドボードを見てみよう



ブレッドボード外観



ブレッドボード内部 結線図

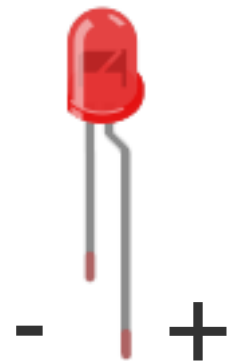
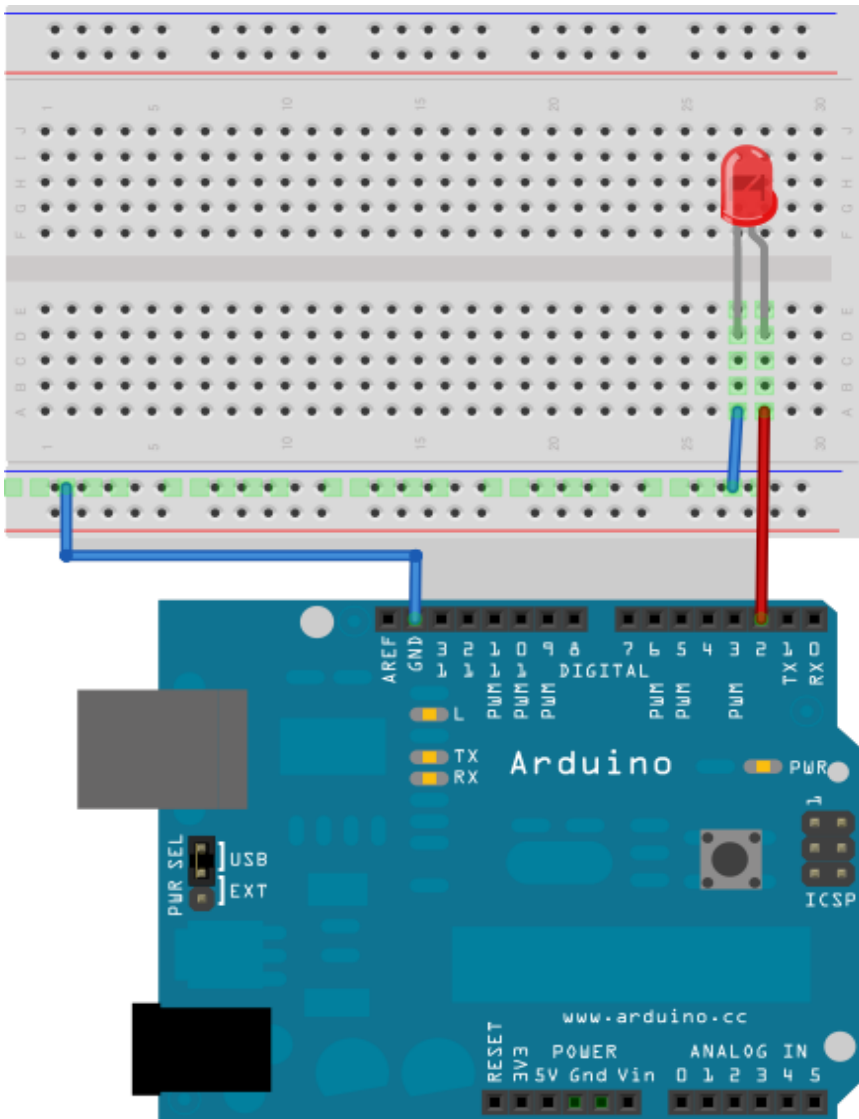
# 4

## 演習問題

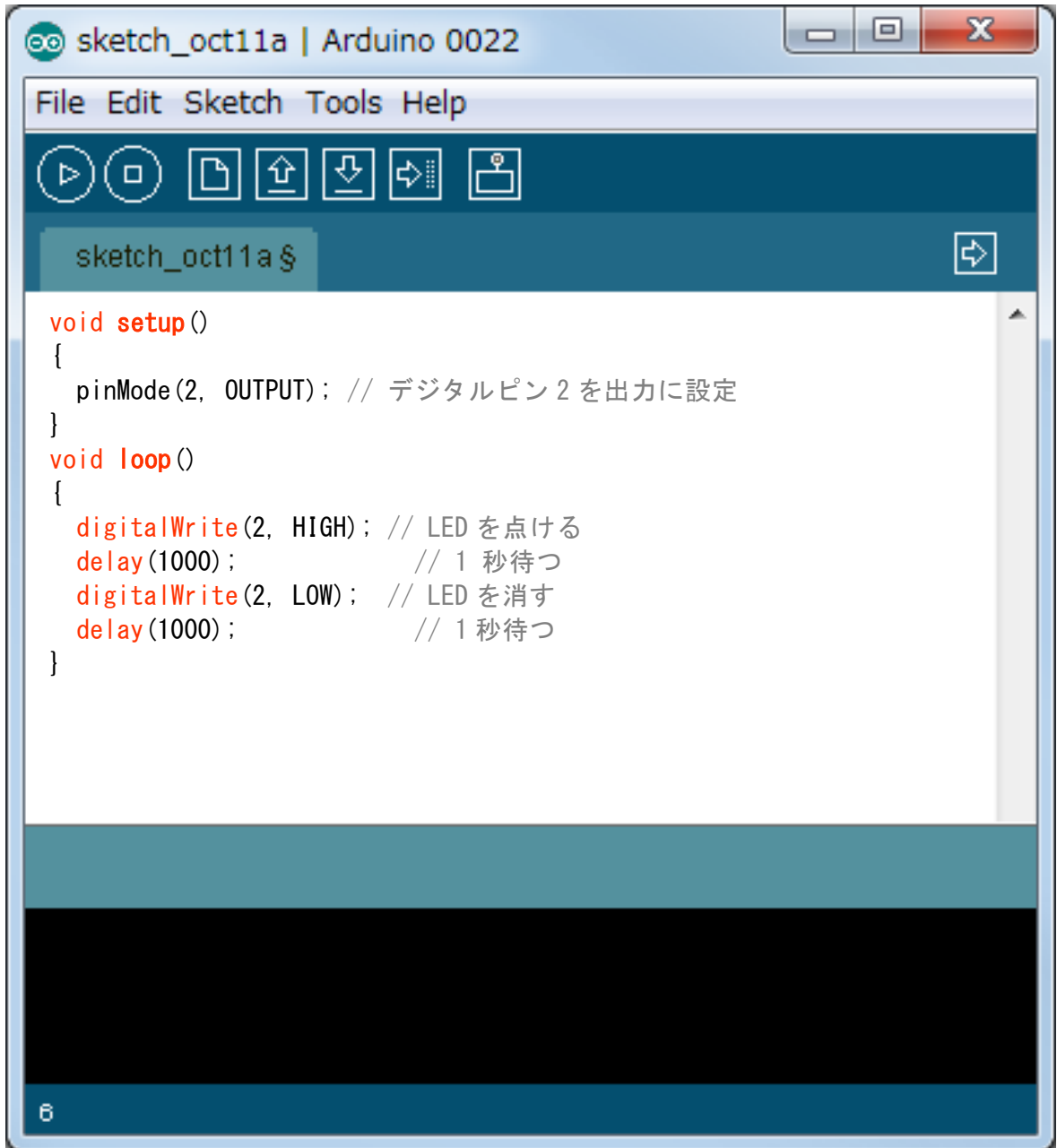
### 4.1



## LED を光らせてみよう。その1

1. 次の図のように Arduino とブレッドボードを配線してください。



- 次に ArduinoSoftware に以下のように入力します。



- ArduinoSoftware の  ボタンをクリックします。  
エラーが表示されなければ次に進んでください。もし、エラーが表示された場合は、プログラムの打ち間違いです。よく確認してください。
- ArduinoSoftware の  ボタンをクリックします。

## Setup 関数

プログラム実行時に最初の一回だけ呼び出される関数。

初期設定を行う場所

```
void setup()  
{  
  
}
```

## Loop()関数

プログラムのメイン関数

ここに記述したプログラムが繰り返し実行される。

```
void loop()  
{  
  
}
```

## pinMode()関数

指定したピン番号の設定

**OUTPUT:**出力専用

**INPUT:**入力専用

```
pinMode(2, OUTPUT); // デジタルピン 2 を出力に設定
```

## digitalWrite()関数

指定したピン番号の設定

**HIGH:** ON 出力

**LOW:** OFF 出力

```
digitalWrite(2, HIGH); // LED を点ける
```

## delay()関数

指定した時間だけプログラムを止める

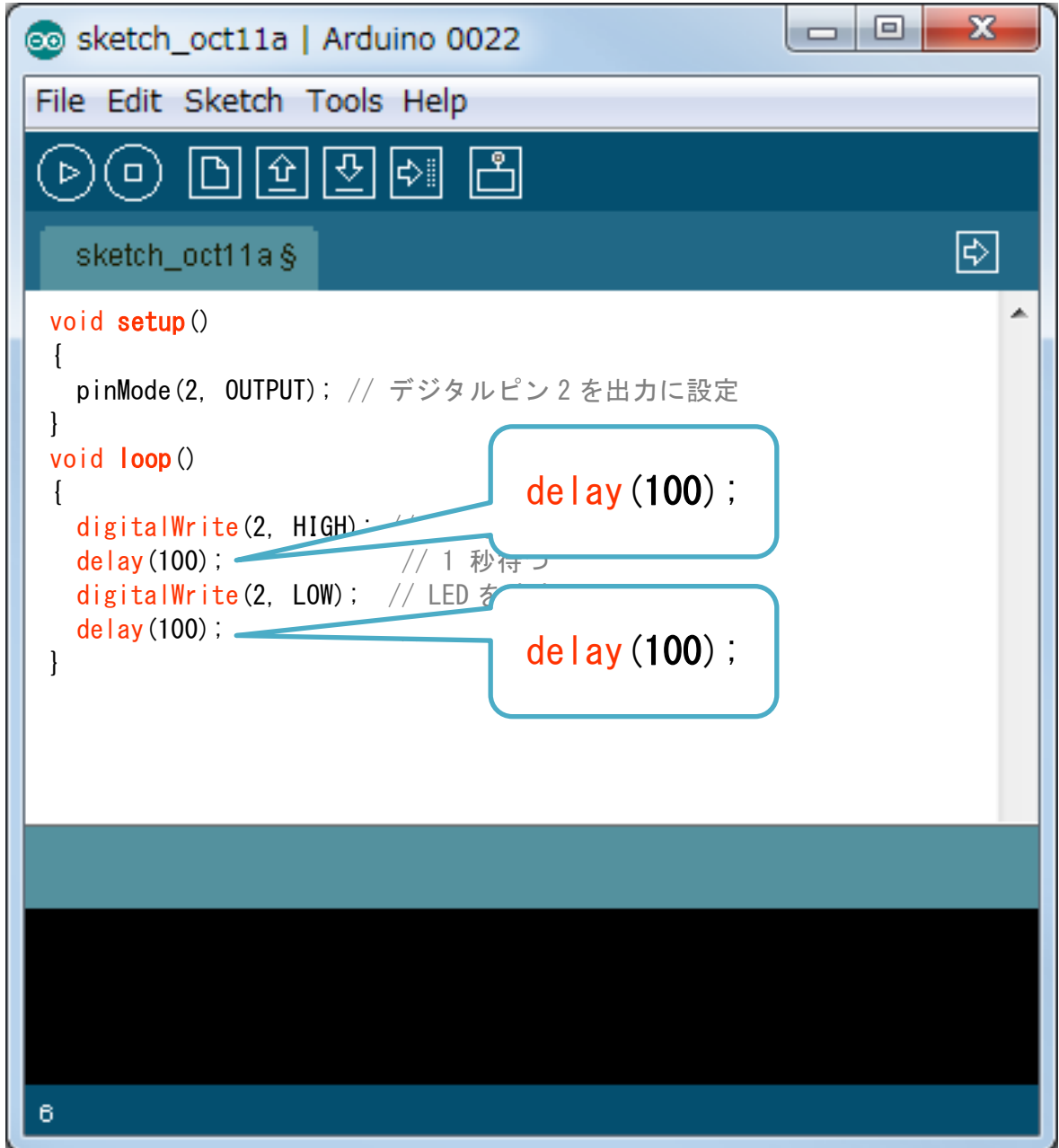
単位は、ミリ秒

1 秒は、1000 ミリ秒、10 秒は 10000 ミリ秒、0.5 秒は 500 ミリ秒

```
delay(1000); // 1 秒待つ
```

## 4.2 LED を光らせてみよう。その2

1. 次に ArduinoSoftware に以下のように入力します。



```
sketch_oct11a | Arduino 0022
File Edit Sketch Tools Help
sketch_oct11a $
void setup()
{
  pinMode(2, OUTPUT); // デジタルピン 2 を出力に設定
}
void loop()
{
  digitalWrite(2, HIGH); // LED を点灯
  delay(100); // 1 秒待つ
  digitalWrite(2, LOW); // LED を消灯
  delay(100);
}
```

2. ArduinoSoftware の  ボタンをクリックします。

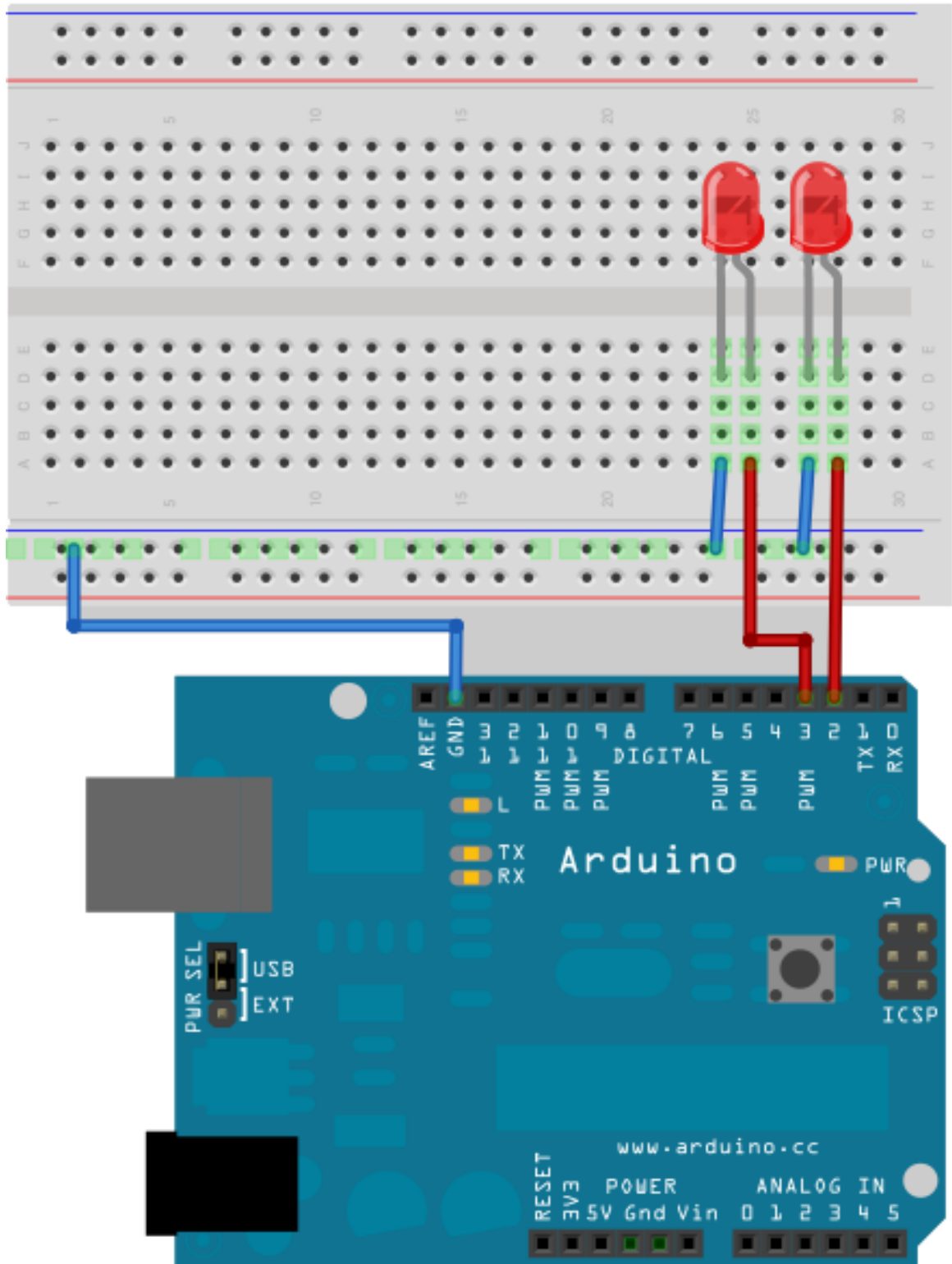
エラーが表示されなければ次に進んでください。もし、エラーが表示された場合は、プログラムの打ち間違いです。よく確認してください。

3. ArduinoSoftware の  ボタンをクリックします。

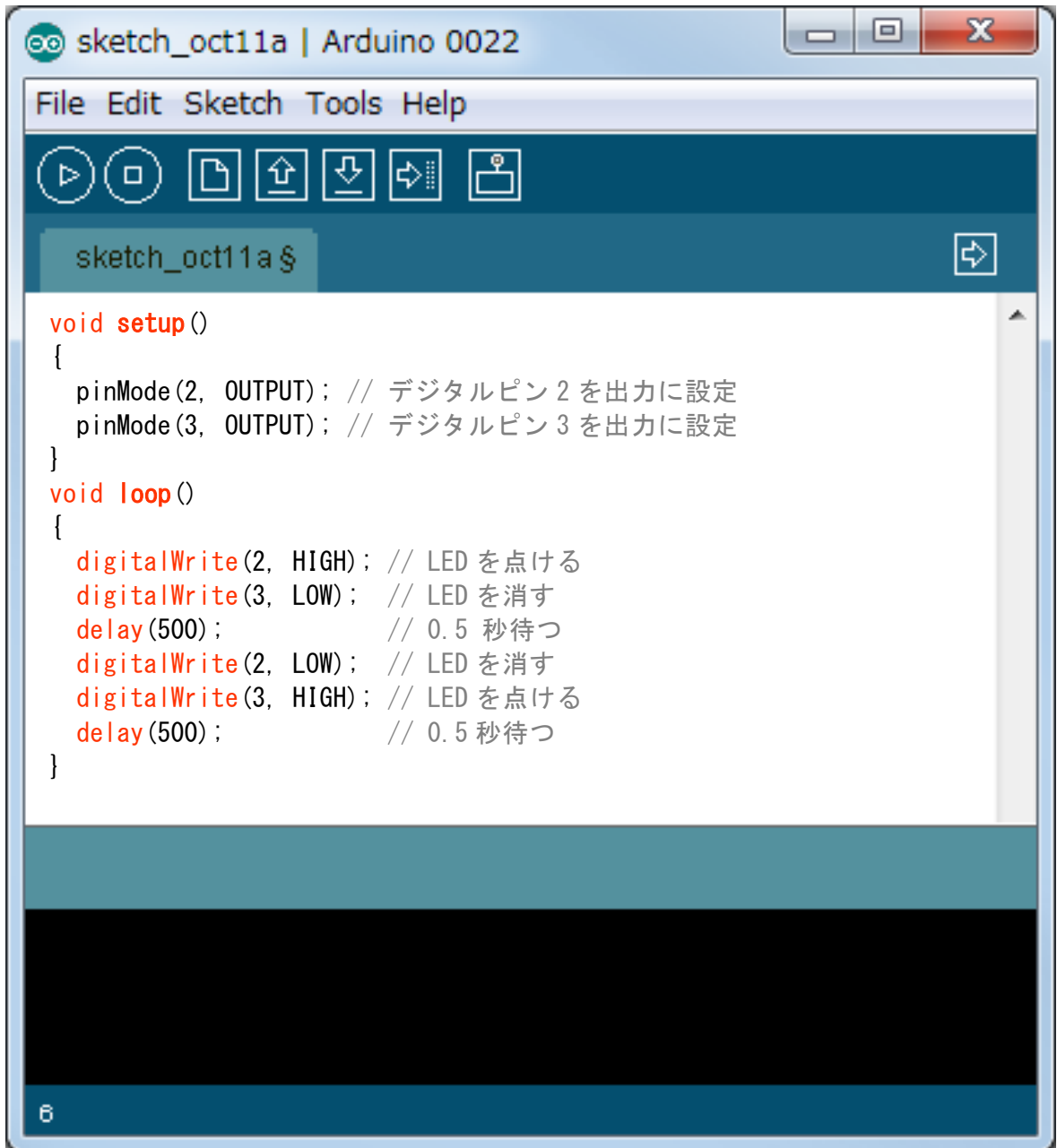
# 4.3

## LED を光らせてみよう。その3

1. 次の図のように Arduino とブレッドボードを配線してください。




2. 次に ArduinoSoftware に以下のように入力します。



```

sketch_oct11a | Arduino 0022
File Edit Sketch Tools Help
sketch_oct11a $
void setup()
{
  pinMode(2, OUTPUT); // デジタルピン 2 を出力に設定
  pinMode(3, OUTPUT); // デジタルピン 3 を出力に設定
}
void loop()
{
  digitalWrite(2, HIGH); // LED を点ける
  digitalWrite(3, LOW); // LED を消す
  delay(500); // 0.5 秒待つ
  digitalWrite(2, LOW); // LED を消す
  digitalWrite(3, HIGH); // LED を点ける
  delay(500); // 0.5 秒待つ
}

```

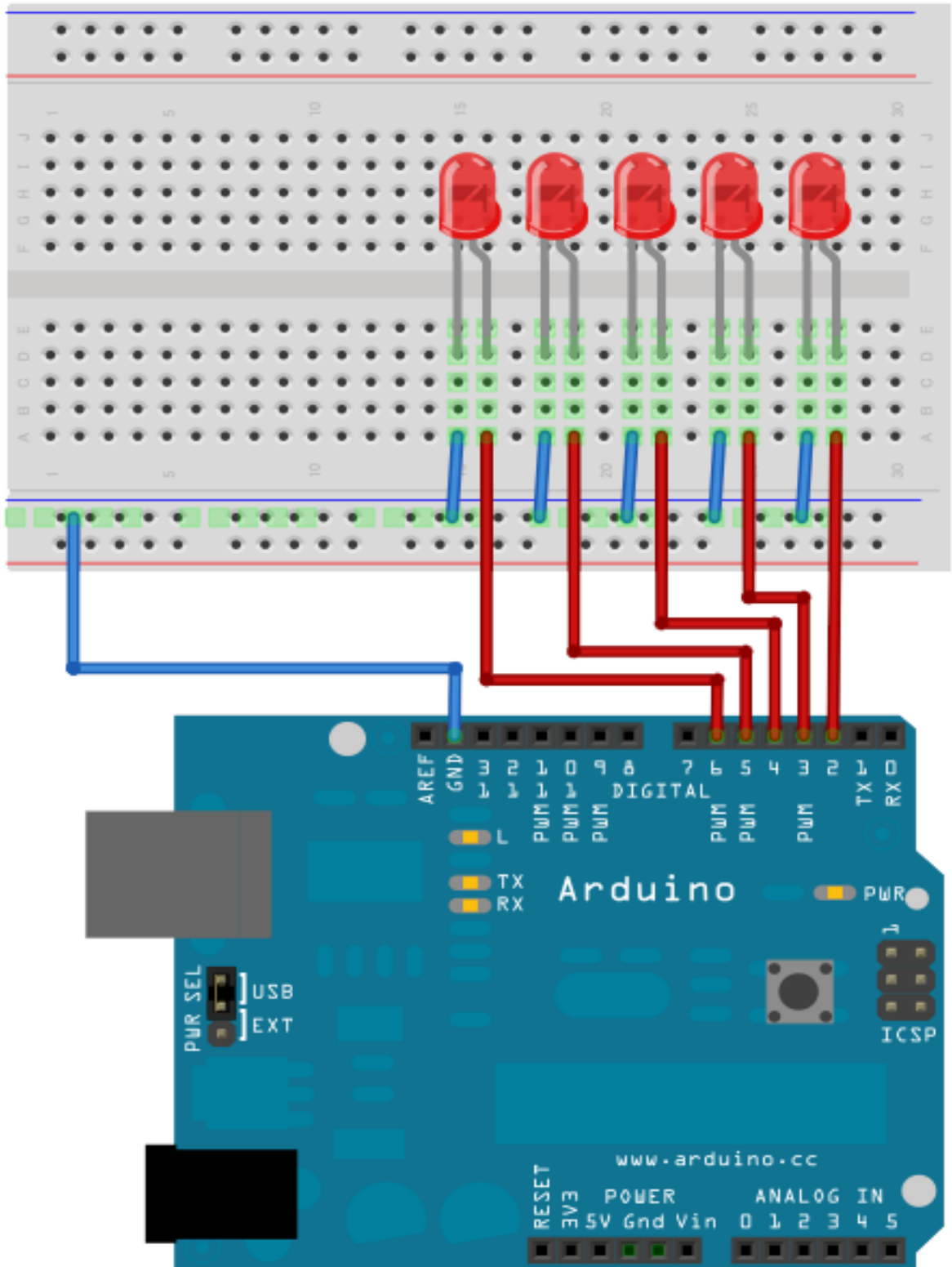
3. ArduinoSoftware の  ボタンをクリックします。  
エラーが表示されなければ次に進んでください。もし、エラーが表示された場合は、プログラムの打ち間違いです。よく確認してください。

4. ArduinoSoftware の  ボタンをクリックします。

# 4.3

## LED を光らせてみよう。その 4

1. 次の図のように Arduino とブレッドボードを配線してください。



## 2. 次に ArduinoSoftware に以下のように入力します。

```
void setup()
{
  pinMode(2, OUTPUT); // デジタルピン 2 を出力に設定
  pinMode(3, OUTPUT); // デジタルピン 3 を出力に設定
  pinMode(4, OUTPUT); // デジタルピン 4 を出力に設定
  pinMode(5, OUTPUT); // デジタルピン 5 を出力に設定
  pinMode(6, OUTPUT); // デジタルピン 6 を出力に設定
}
void loop()
{
  digitalWrite(2, HIGH); // LED を点ける
  digitalWrite(3, LOW); // LED を点ける
  digitalWrite(4, LOW); // LED を点ける
  digitalWrite(5, LOW); // LED を点ける
  digitalWrite(6, LOW); // LED を点ける
  delay(100); // 1 秒待つ

  digitalWrite(2, LOW); // LED を点ける
  digitalWrite(3, HIGH); // LED を点ける
  digitalWrite(4, LOW); // LED を点ける
  digitalWrite(5, LOW); // LED を点ける
  digitalWrite(6, LOW); // LED を点ける
  delay(100); // 1 秒待つ

  digitalWrite(2, LOW); // LED を点ける
  digitalWrite(3, LOW); // LED を点ける
  digitalWrite(4, HIGH); // LED を点ける
  digitalWrite(5, LOW); // LED を点ける
  digitalWrite(6, LOW); // LED を点ける
  delay(100); // 1 秒待つ

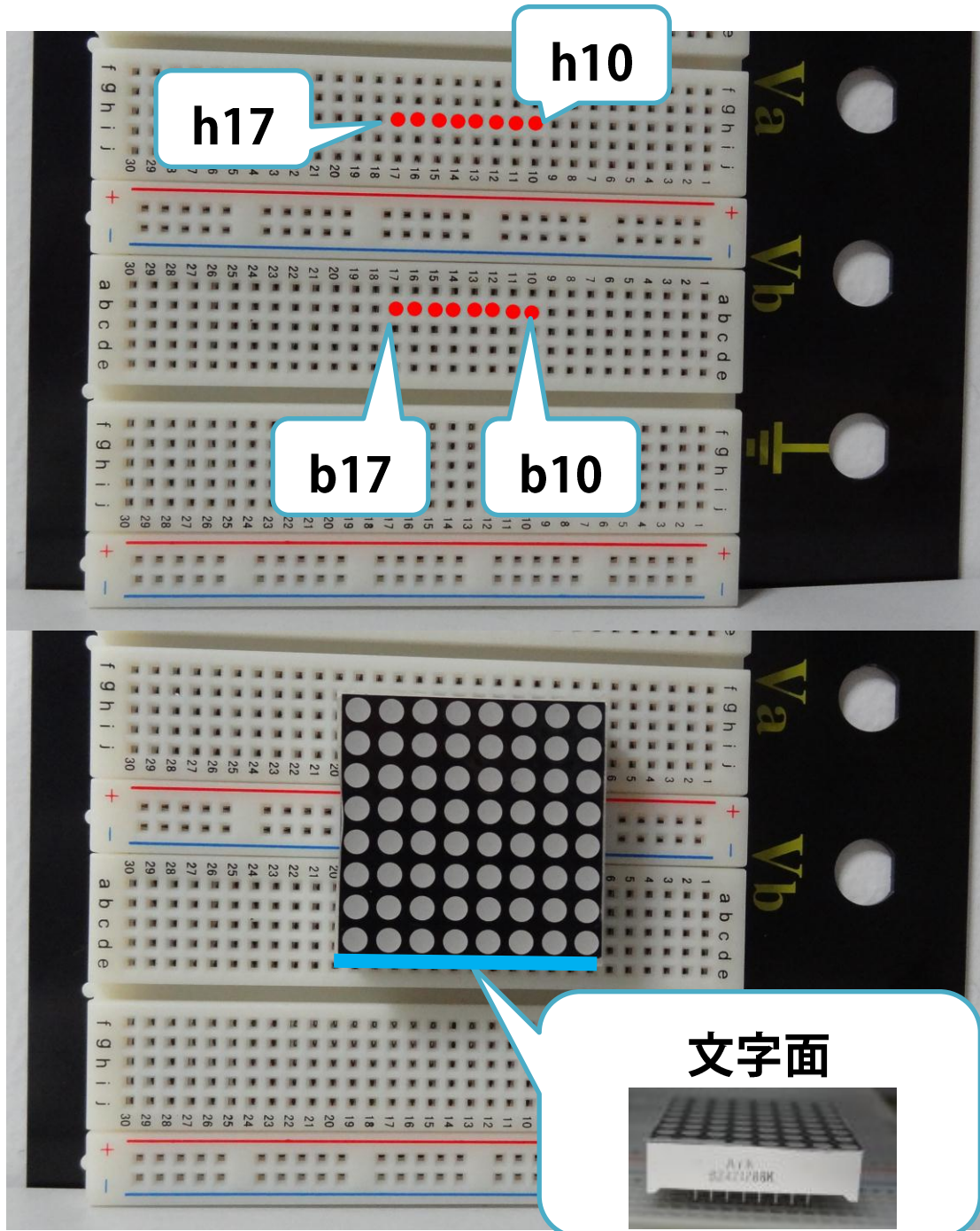
  digitalWrite(2, LOW); // LED を点ける
  digitalWrite(3, LOW); // LED を点ける
  digitalWrite(4, LOW); // LED を点ける
  digitalWrite(5, HIGH); // LED を点ける
  digitalWrite(6, LOW); // LED を点ける
  delay(100); // 1 秒待つ

  digitalWrite(2, LOW); // LED を点ける
  digitalWrite(3, LOW); // LED を点ける
  digitalWrite(4, LOW); // LED を点ける
  digitalWrite(5, LOW); // LED を点ける
  digitalWrite(6, HIGH); // LED を点ける
  delay(100); // 1 秒待つ
}
```

## 5 LED マトリックス

### 5.1 LED マトリックス その1

1. 次の図のように LED マトリックスを配線してください。





### 3. 次に ArduinoSoftware に以下のように入力します。

```
void setup() {  
  //16本のピン(2~17)を出力に設定  
  for(int i=2; i<=17; i++) {  
    pinMode(i, OUTPUT);    //ピンを出力に設定  
  }  
}  
  
void loop() {  
  //行(横)の繰り返し処理  
  for(int i=2; i<=9; i++) {    //行(2~9番ピン)  
    digitalWrite(i, HIGH);    //HIGHで点灯  
  
    //列(縦)の繰り返し処理  
    for(int j=10; j<=17; j++) { //列(10~17番ピン)  
      digitalWrite(j, LOW);  //LOWで点灯  
      delay(100);           //点灯時間  
      digitalWrite(j, HIGH); //列をオフにする  
    }  
  
    digitalWrite(i, LOW);    //行をオフにする  
  }  
}
```

## 5.2

## LED マトリックス その2 文字の描写

## 1. ArduinoSoftware に以下のように入力します。

```
boolean matrix[8][8]={
  {0,0,0,1,1,0,0,0},
  {0,0,1,0,0,1,0,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,1,1,1,1,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0}
};

void setup() {
  for(int i=2;i<=17;i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
}

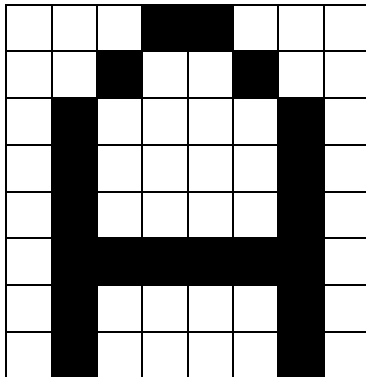
void loop() {
  for(int i=2;i<=9;i++) {
    digitalWrite(i, HIGH); //行 : HIGH で点灯
    for(int j=10;j<=17;j++) {
      if(matrix[i-2][j-10]==1) { //点灯条件
        digitalWrite(j, LOW); //列 : LOW で点灯
      }

      delayMicroseconds(300); //0.03 秒点灯
      digitalWrite(j, HIGH); //オフにする
    }
    digitalWrite(i, LOW); //オフにする
  }
}
```

# 5.3

## LED マトリックス その2 任意文字の描写

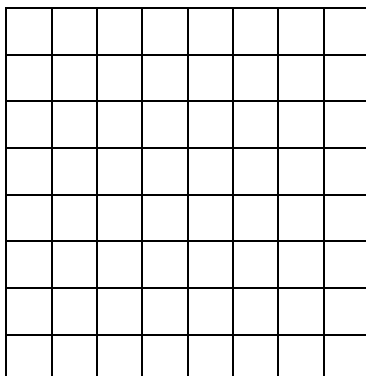
1. 8X8 のマス目に任意の文字や記号を書いてみましょう
2. 左のマス目に LED の発光部分を黒で塗りつぶしてください。
3. 右のマス目に「黒を 1」「白を 0」にして記入してください。



→黒を 1→

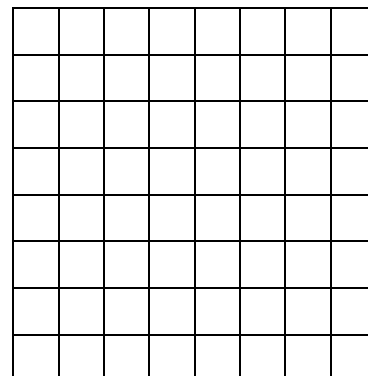
→白を 0→

0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0



→黒を 1→

→白を 0→

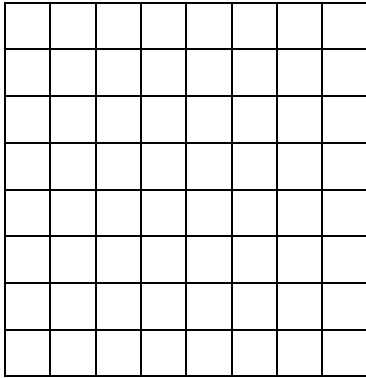


※ミスした場合は、次のページに予備のマス目シートがあります。

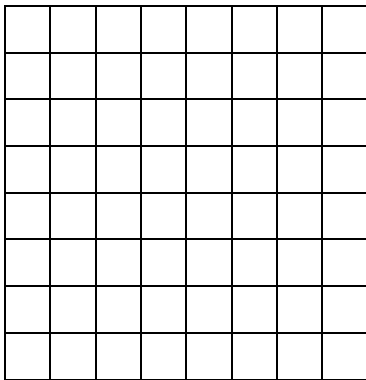
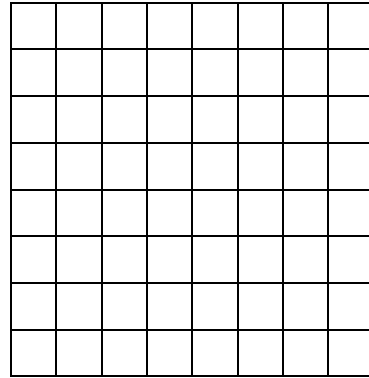
4. 5.3 のプログラム上部の matrix[8][8]の 0,1 を任意に置き換えてください。

```
boolean matrix[8][8]={
  {0,0,0,1,1,0,0,0},
  {0,0,1,0,0,1,0,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,1,1,1,1,1,0},
  {0,1,0,0,0,0,1,0},
  {0,1,0,0,0,0,1,0}
};
```

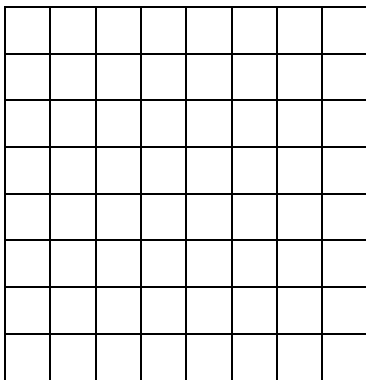
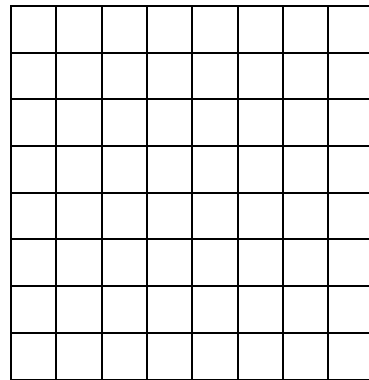
予備シート



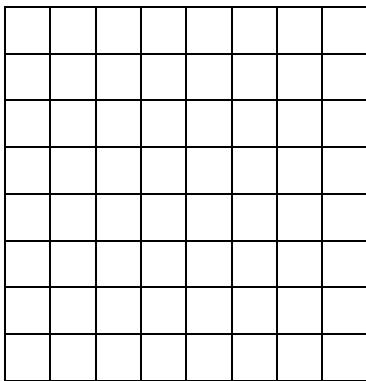
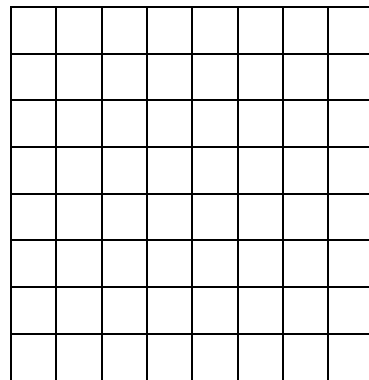
→黒を 1→  
→白を 0→



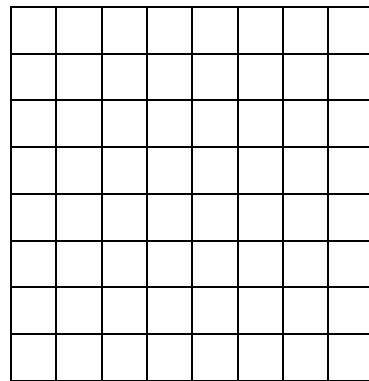
→黒を 1→  
→白を 0→



→黒を 1→  
→白を 0→



→黒を 1→  
→白を 0→



## 5.4

## LED マトリックス その3 流れる文字

```

#define MAX_LEN 24 //8x3 文字 = 24
//8x25 の配列にする (25 列目は余白)
boolean matrix[8][ MAX_LEN+1]={
  {0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0},
  {0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0},
  {0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
  {0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
  {0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
  {0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0},
  {0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0}
};

void setup() {
  for(int i=2; i<=17; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
}

void loop() {
  int count=5;
  while(count>0) {
    for(int i=2; i<=9; i++) {
      digitalWrite(i, HIGH);
      for(int j=10; j<=17; j++) {
        digitalWrite(j, !matrix[i-2][j-10]);
        delayMicroseconds(300);
        digitalWrite(j, HIGH); //LED OFF
      }
      digitalWrite(i, LOW); //LED OFF
    }
    count--;
  }

  //カスケードリング
  for(int k=0; k<8; k++) {
    //以下の配列数の値を変更しておく
    for(int l=0; l<=MAX_LEN; l++) {
      if(l== MAX_LEN) {
        matrix[k][ MAX_LEN ]=matrix[k][0];
      }else{
        matrix[k][l]=matrix[k][l+1];
      }
    }
  }
}

```

解説

3文字の場合

```

#define MAX_LEN 24 //8x3 文字 = 24
//8x25 の配列 (25列目は余白)
boolean matrix[8][MAX_LEN+1]=
    {
        {0,0,0,1,1,0,0,0,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0},
        {0,0,1,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0},
        {0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0,1,1,1,1,1,0,0,0,1,0,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0},
        {0,1,1,1,1,1,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0},
        {0,1,0,0,0,0,0,1,0,0,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0}
    };
    
```

**8列×3文字 =24列 +1列 =25列**

4文字の場合

```

#define MAX_LEN 33 //8x4 文字 = 32
//8x25 の配列にする (25列目は余白)
boolean matrix[8][MAX_LEN+1]=
    {
        {0,0,0,1,1,0,0,0,0,0, .. 略.. ,1,1,1,1,1,0,0,0,0,0,1,1,1,1,0,0,0},
        {0,0,1,0,0,1,0,0,0,0, .. 略.. ,1,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0},
        {0,1,0,0,0,0,0,1,0,0, .. 略.. ,1,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0, .. 略.. ,1,1,1,1,1,0,0,0,1,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0, .. 略.. ,1,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0},
        {0,1,1,1,1,1,1,0,0, .. 略.. ,1,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0},
        {0,1,0,0,0,0,0,1,0,0, .. 略.. ,1,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0},
        {0,1,0,0,0,0,0,1,0,0, .. 略.. ,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0}
    };
    
```

**8列×4文字 =32列 +1列 =33列**

## 5.5 LED マトリックス その4 自分の名前の描写

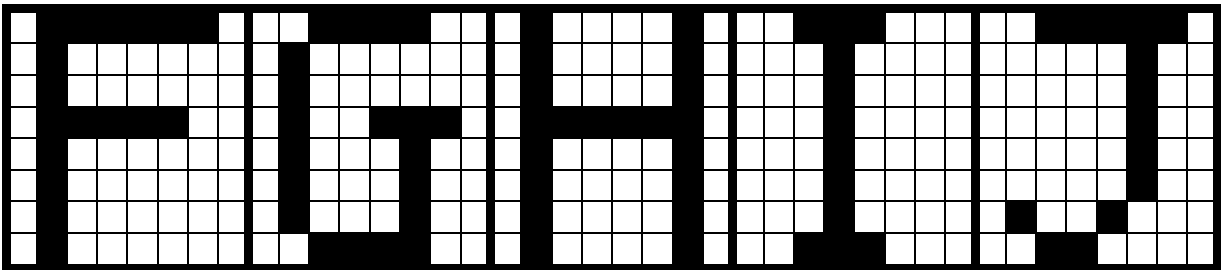
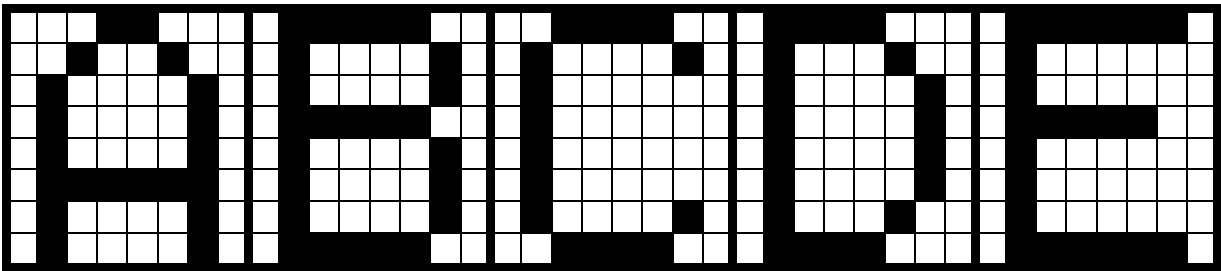
自分の名前 (3~7桁) を表示してみよう。

たかゆき → TAKAYUKI

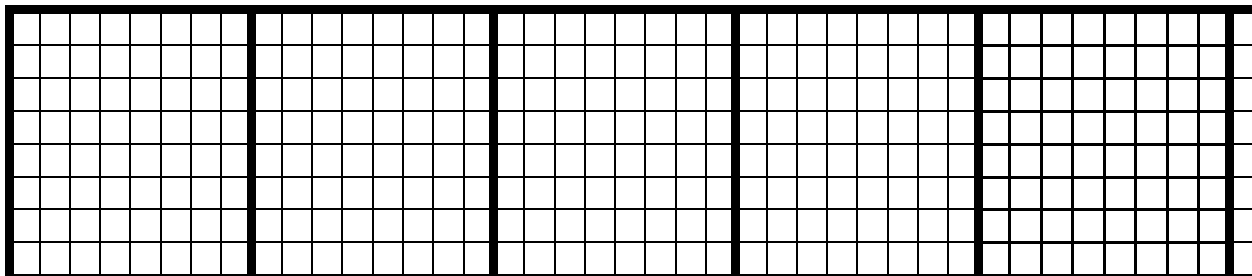
さなえ → SANAE

あきひこ → AKIHIKO

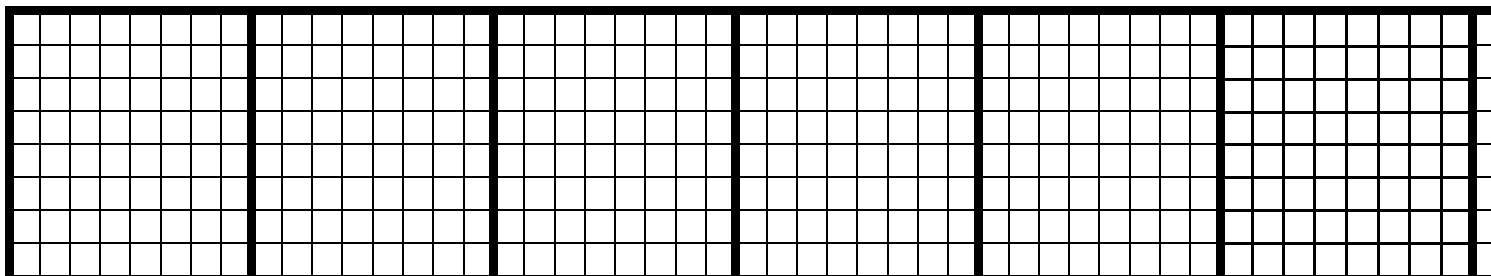
ゆき → YUKI



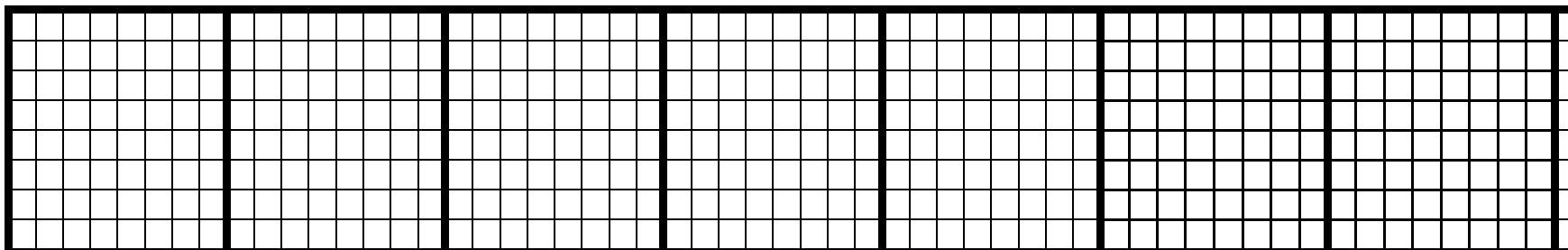
**5文字用** #define MAX\_LEN 40 //8x5文字 = 40



**6文字用** #define MAX\_LEN 48 //8x6文字 = 48



**7文字用** #define MAX\_LEN 56 //8x7文字 = 56





## 6

## Arduino でできること

今回は、Arduino を用いてミニ電光掲示板の演習を行いました。思ったより簡単にできたと驚かれたかと思います。昔は、このような電光掲示板を作成するためには、プログラミングの知識だけでなく電気の知識などが必要でした。最近では、Arduino があれば専門知識がなくても簡単に作ることができます。

この演習では、LED マトリックスで出力のみを行いました。各種センサの値を取得することも可能です。例えば、今回のミニ電光掲示板に距離センサを取り付ければ、距離に応じて再生速度を変化させることも可能ですし、放射線を測定するガイガーカウンターを取り付ければ放射線の値を表示することも簡単にできてしまいます。様々なセンサ（距離センサ、赤外線センサ、圧力センサ、色センサなど）を用いることで、私たち人間の行動データを取得することができます。Wii リモコンや Kinect のようなセンサも簡単に作ることができます。みなさんのアイデアと発想次第でどのようなものを作り出すことができるのです。

情報メディア学科 小坂研究室では、このようなセンサを用いて新しいデバイスの開発なども行なっています。

神奈川県立 神奈川工科大学 情報学部 情報メディア学科 助教  
小坂崇之

<http://www.kosaka-lab.com>