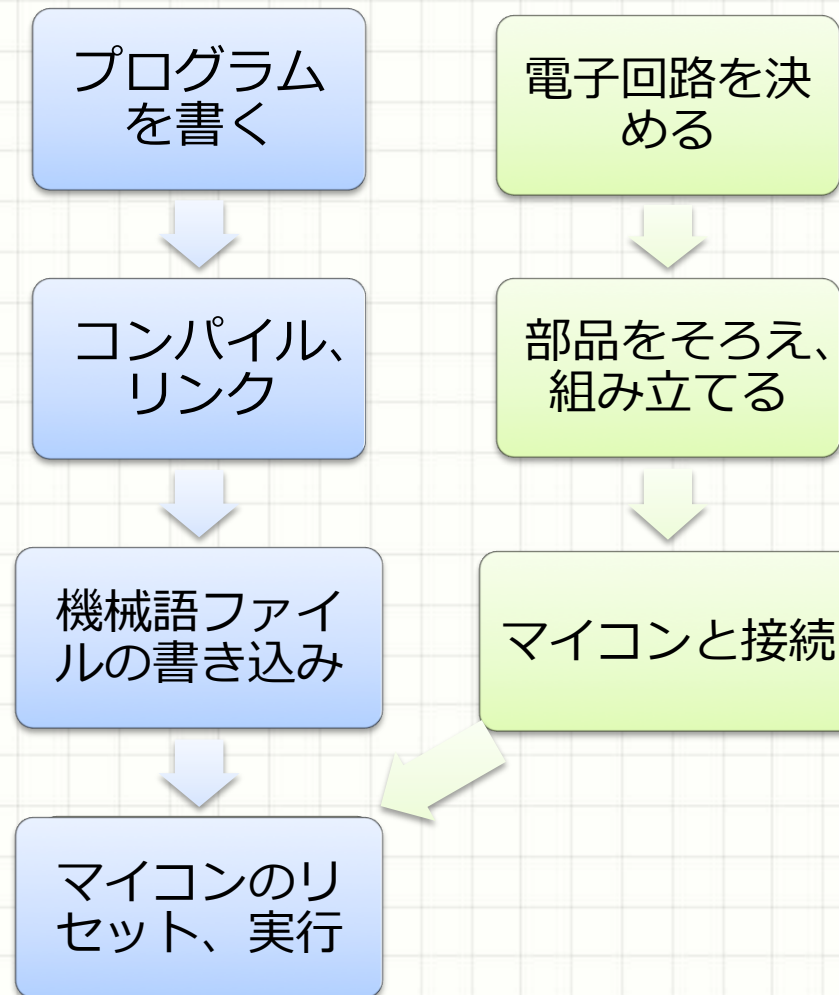


計算機システムII ARDUINO プログラミング入門



I. 準備

1. Arduino のプロジェクト作成手順



PCのソフトウェアを確認



- ① Arduino IDE ソフトウェアを「開く」.
 - ✓ 空の、または前回編集していたスケッチ(プログラム)が開く.
- ② メニュー「ツール(Tools)」→「ボード(Board)」に自分のボードのモデルが表示され、選択(チェック)されている
 - ✓ Arduino UNO、Arduino Duemilanove 328 など
- ③ ボードを接続すると、
「ツール(Tools)」→「シリアルポート(Serial Ports)」のリストに自分のボードのシリアルポートが追加される
 - ✓ Windows なら COM??.

Arduino ボードの接続テスト

① USBケーブルでPCと接続

- ボードの電源がONになり Power On LEDが点灯
- プログラムが書き込まれている場合、動作がはじまる(LEDの点滅など).

② Arduino IDE でシリアルポートを確かめる

- シリアルポートのリストに新しいCOMポートが出現
- COM??かは決まっていない.
接続後メニューにあらわれる番号をおぼえておく

接続、スケッチ作成、アップロード

- ① メニューのシリアルポートのリストから
ボードのCOMポートを選ぶ.

- チェックが入る

- ② スケッチを書く／開く

- Filesファイル > Examples スケッチの例 > 01. Basics > Blink を開いてみる

- ③ メニューの File > Upload または Upload ボタンで、プログラムをボードに書き込む.

- コンパイル完了メッセージの後、ボードの送受信LEDが点滅.
その後、ゆっくりユーザーLEDが点滅

2. すこしずつ、いろいろ変更してみる

① Blink のプログラムを変更してみる

- loop() の中の delay(1000); (2カ所)の数値 1000 をそれぞれ変更すると、どうなるか.
- 点滅が見えなくなくなるまで、高速に点滅させみる. 明るさを調整してみる.

② 回路を変更してみる

- 別のピンにLEDをつないで増やしてみる
- LEDのかわりに圧電(ピエゾ)スピーカを接続してみる

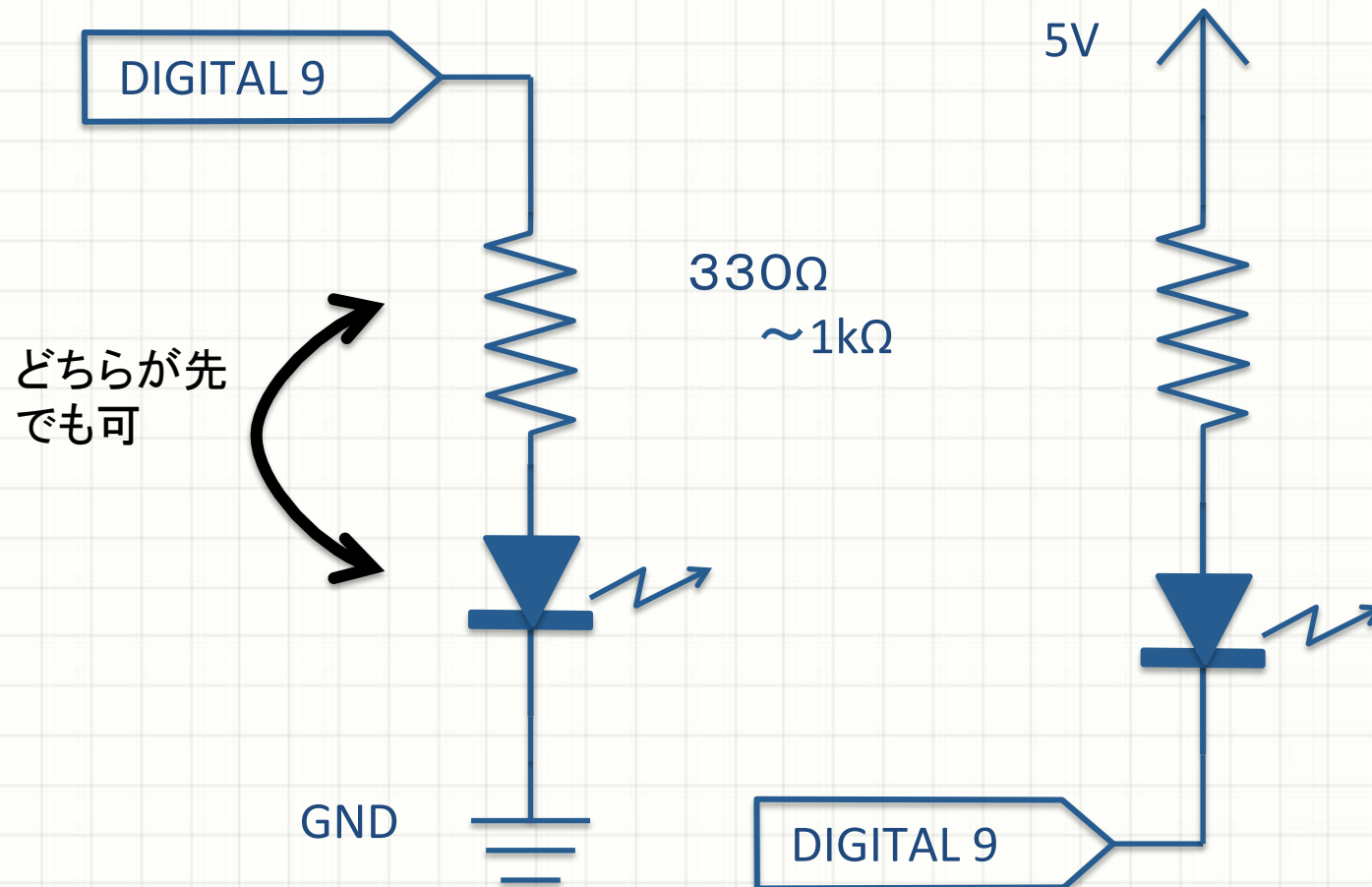


II. 入出力と プログラミング

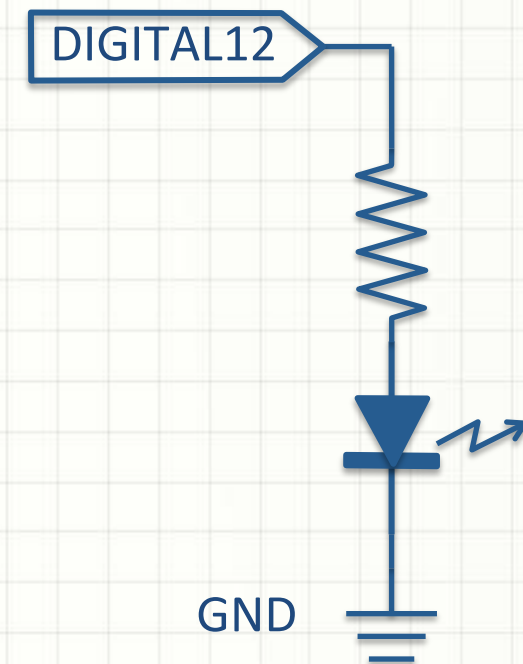
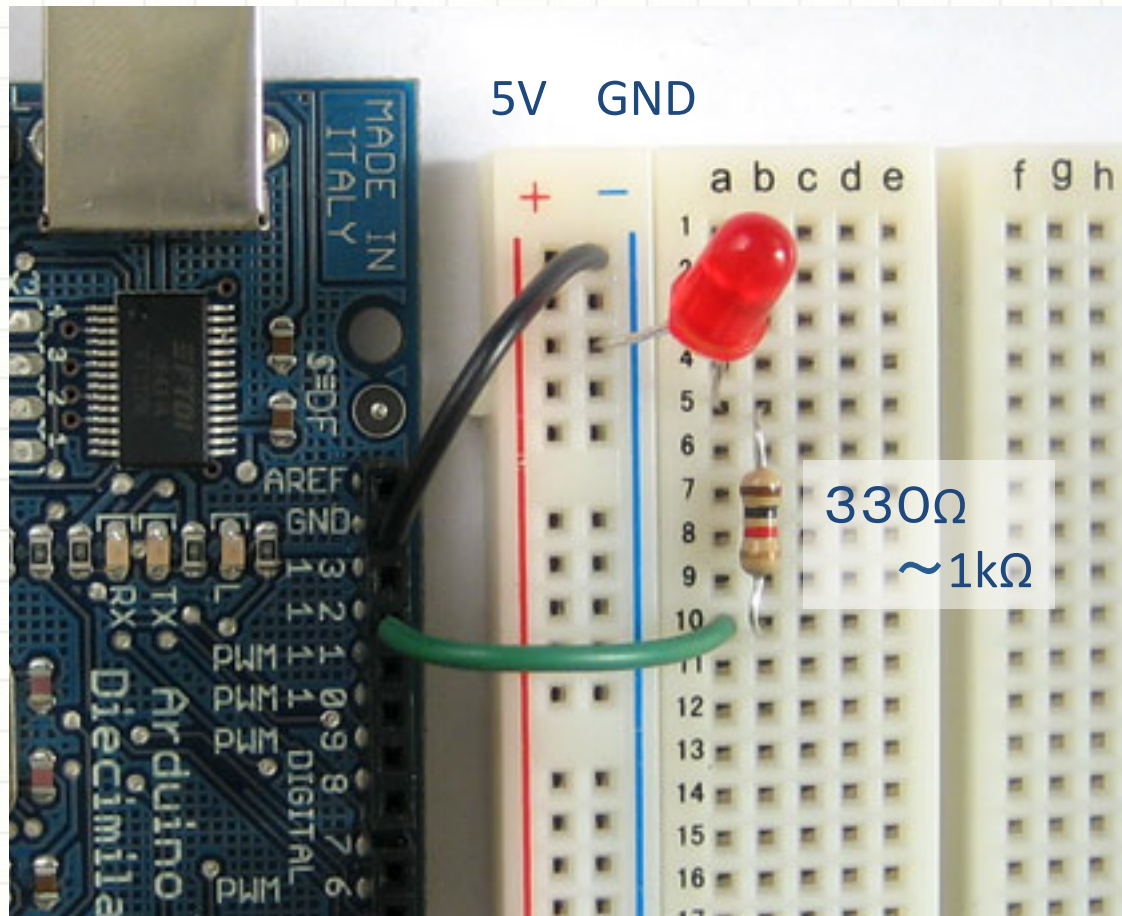
3. LED（発光ダイオード）とデジタル出力

正論理（HIGHで点灯）

負論理（LOWで点灯）

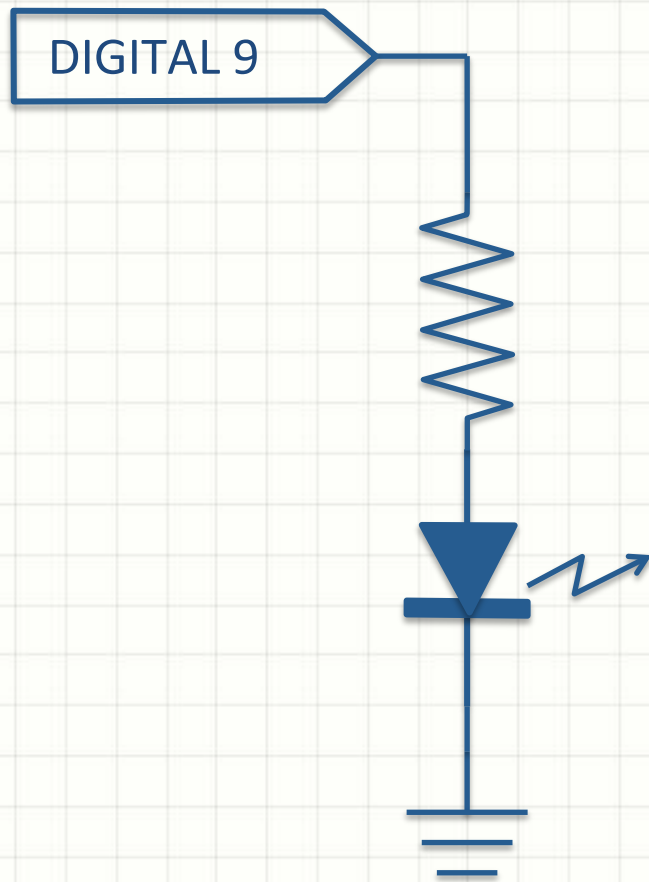


ブレッドボードでのLEDの配線



<http://www.ladyada.net/learn/arduino/lesson3.html>

デジタル出力のスケッチ(プログラミング)



```
void setup() {  
    pinMode(9, OUTPUT);  
}
```

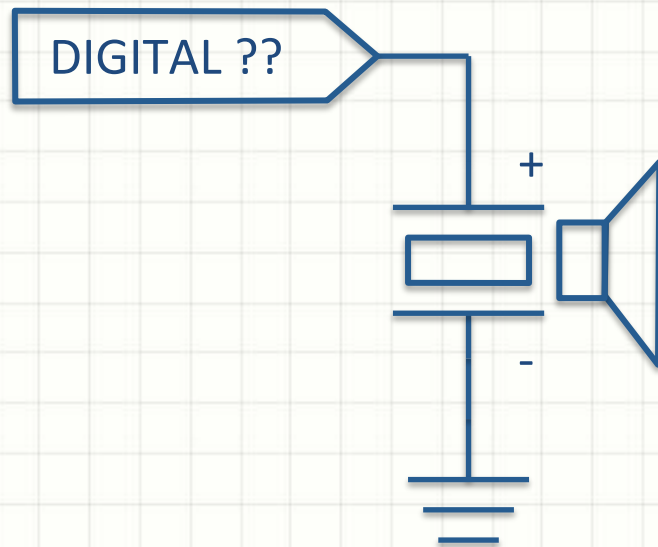
```
void loop() {  
    ...  
    digitalWrite(9, HIGH);  
    ...  
    digitalWrite(9, LOW);  
    ....  
}
```

演習課題 II-1

①〔踏切シグナル〕

LED 二つをデジタルピン10 と 11に接続する回路を作り、交互に点灯(0.5秒程度)するスケッチを書きなさい.

ピエゾスピーカの接続



```
pinMode(9, OUTPUT);
```

```
tone(9, 1500, 500);  
/* ピン9 に 1500Hz を  
500 ミリ秒出力)
```



まとめ：

デジタル出力1ビットでできること

- ① スイッチON／OFF
- ② 電飾の点滅、映像の描画
- ③ 赤外線リモコンなど、信号の送信
- ④ パルス幅変調・・・明るさ調整、モーター回転数
(力)の制御、サーボモーター(位置)制御
- ⑤ 音の発生：ブザー、電子音(矩形波)

4. シリアルポートでの文字入出力

1) PCと文字通信する

- Files ファイル> Examplesスケッチの例 > 04.Communication> ASCII Table を開く
- プログラムを Upload マイコンボードに書き込む

2) Terminal シリアルモニタボタンをクリックして ウィンドウを開く



- ウィンドウ右下のPC側通信速度(ボーレート)を確認.
マイコン側と速度が違くと文字化けする

プログラムでの使い方

1) setup() で通信速度セット

- ボーレートはシリアルモニタのメニューを参考に選ぶ

```
void setup() {  
    Serial.begin(9600);
```

```
    ...
```

```
}
```

2) begin() 後はいつでも利用可能

- 変数... char, char の文字列配列, int, long, float.
整数の場合, BIN (二進), HEX (十六進) など基数を指定できる

```
Serial.print(変数);  
/* 改行なし */
```

```
Serial.println(変数);  
/* 改行あり */
```

シリアルポートを使ったプログラム

1) シリアルモニタに Hello world! と出力するプログラムを書いてみる

➤ `Serial.println("Hello world!");`



2) 受信の手順をサンプルでみる

➤ MorseTransmitter ... 受け取った文字のモールス信号を
光と音で発信

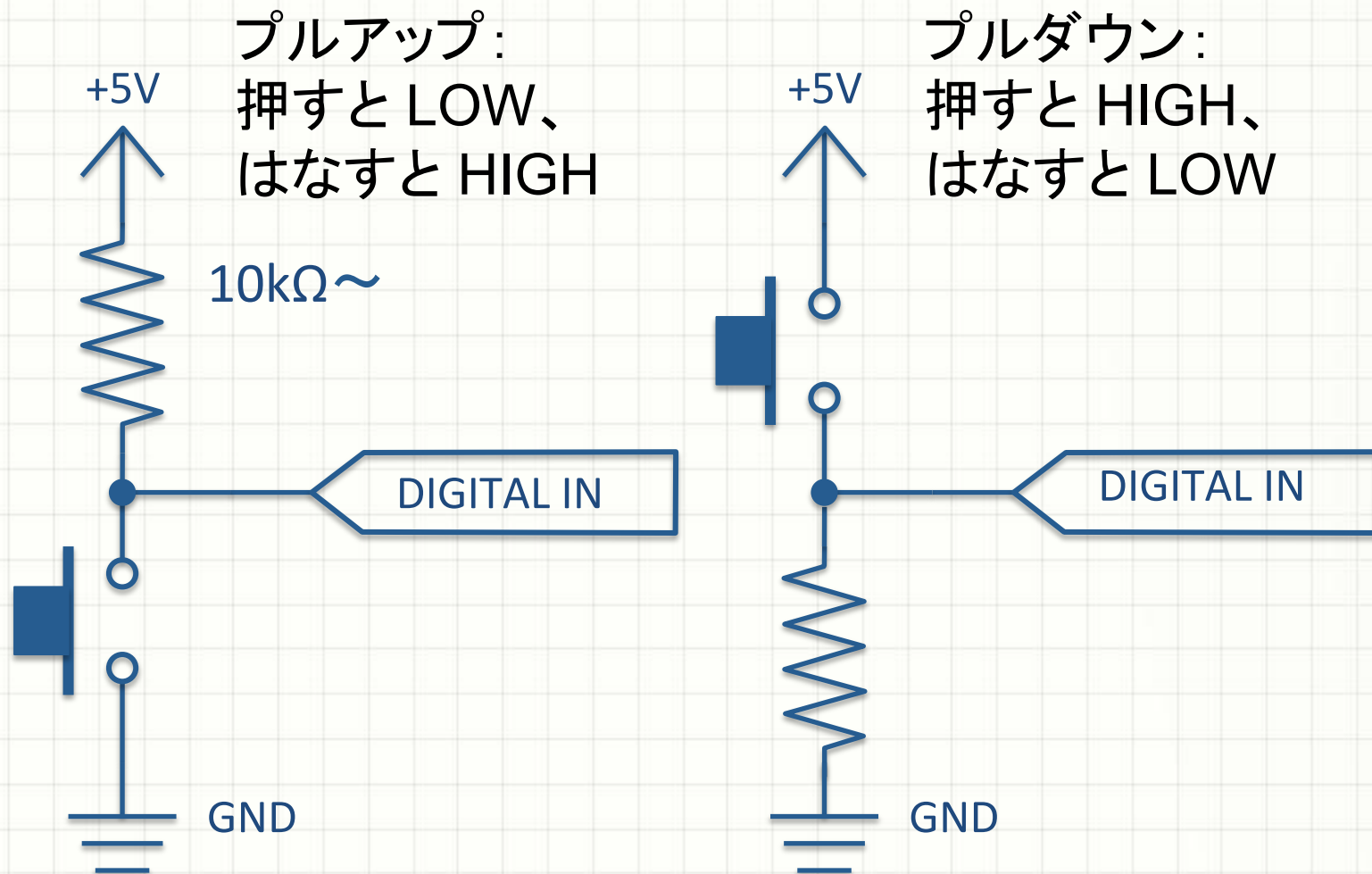
演習課題 II-2

①〔2進／16進計算機〕

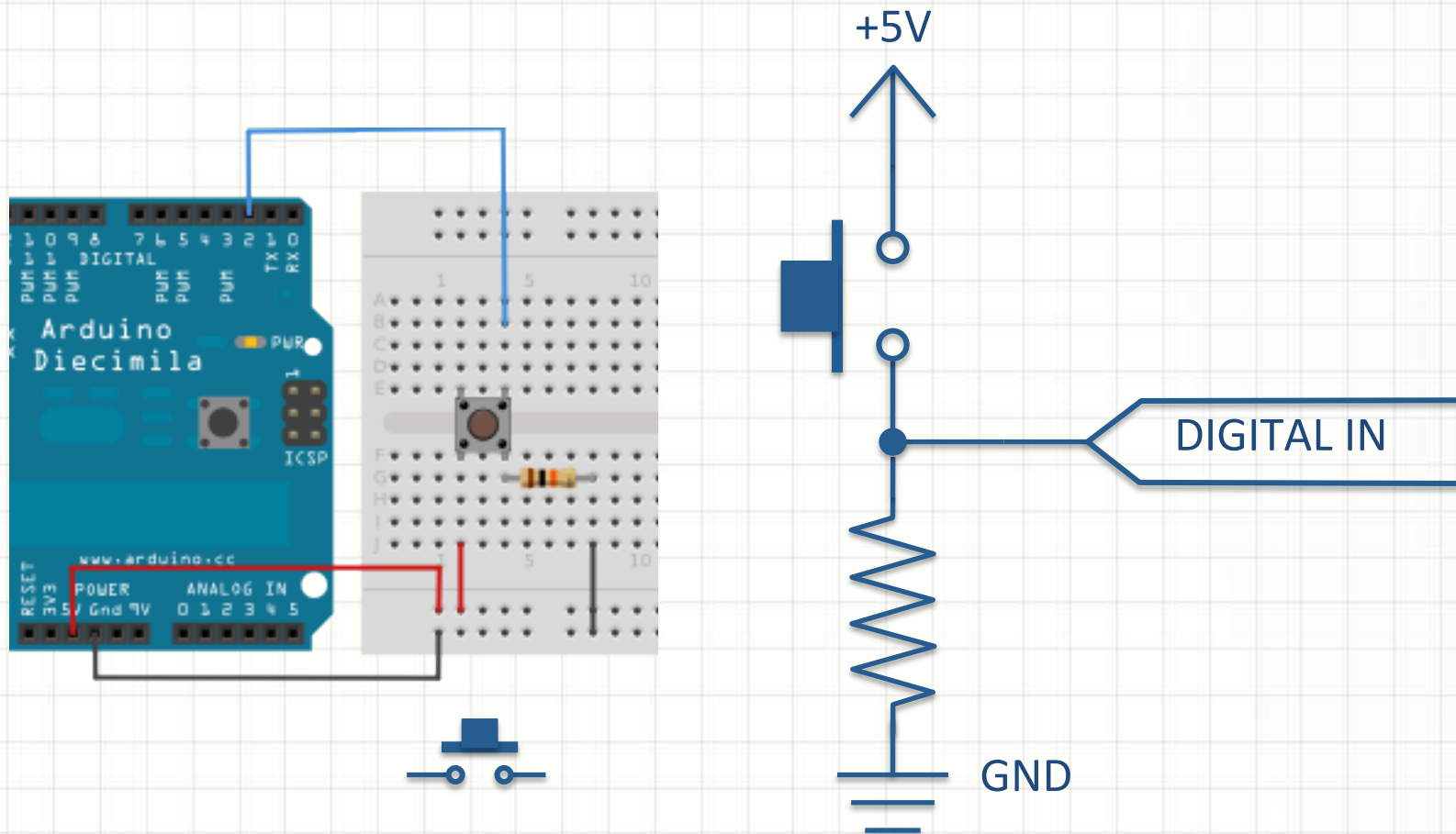
二つの int 型変数に異なる整数値を10進数で代入し、それぞれの2進数表現(BIN)と16進数表現(HEX)をシリアルモニタで確認しなさい。

② 上記の二つの値のビットごとの AND(&)、OR(|)、XOR(^) を求めて表示させ、結果を確認しなさい。

6. スイッチとデジタル入力



ブレッドボードでのスイッチの配線： プルダウン型



演習課題 II-3

- ① スイッチを押してから、10秒間だけLEDが点滅するスケッチを書きなさい.

時間経過を知るには、`millis()` 関数を使いなさい.


6. アナログ入力の使い方

1) アナログ入力ピン AIN 0 ~ 5 に印加された電圧を A/D コンバータで整数値 0 ~ 1023 として取得

➤ 入力できる電圧は 0 V (= GND) から 5 V (=電源電圧) まで

2) アナログタイプのセンサの出力を利用できる

➤ 温度センサ、照度センサ、距離センサ、加速度センサ
etc...



1. Windows、UNIX/Linux、iOS など
マルチタスクOS

1. プログラムの起動、実行、終了をOSが制御

2. 同時に動く他のプログラムを気にせずプログラミング

3. ハードウェアアクセスはOS経由に限る

1. マイコン、組込みリアルタイムOS

1. 動作するプログラムは一つ。振るまいはプログラムしだい

2. 作業の数、優先順位、メモリの使い方はプログラマが計画を立てる

3. 実際のハードウェアにビット単位で直接アクセス

プログラムのソースコードを書く

エディタでテキストファイルを編集

実行形式(機械語)のファイルに変換

コンパイルしてバイナリファイルを生成

コマンド等として実行

OSがプログラムをメモリにロード。与えたコマンド引数がmain関数に渡される

main 関数が終了で実行終了

OSにエラーコードを返す

プログラムのソースコードを書く

エディタでテキストファイルを編集

実行形式(機械語)のファイルに変換

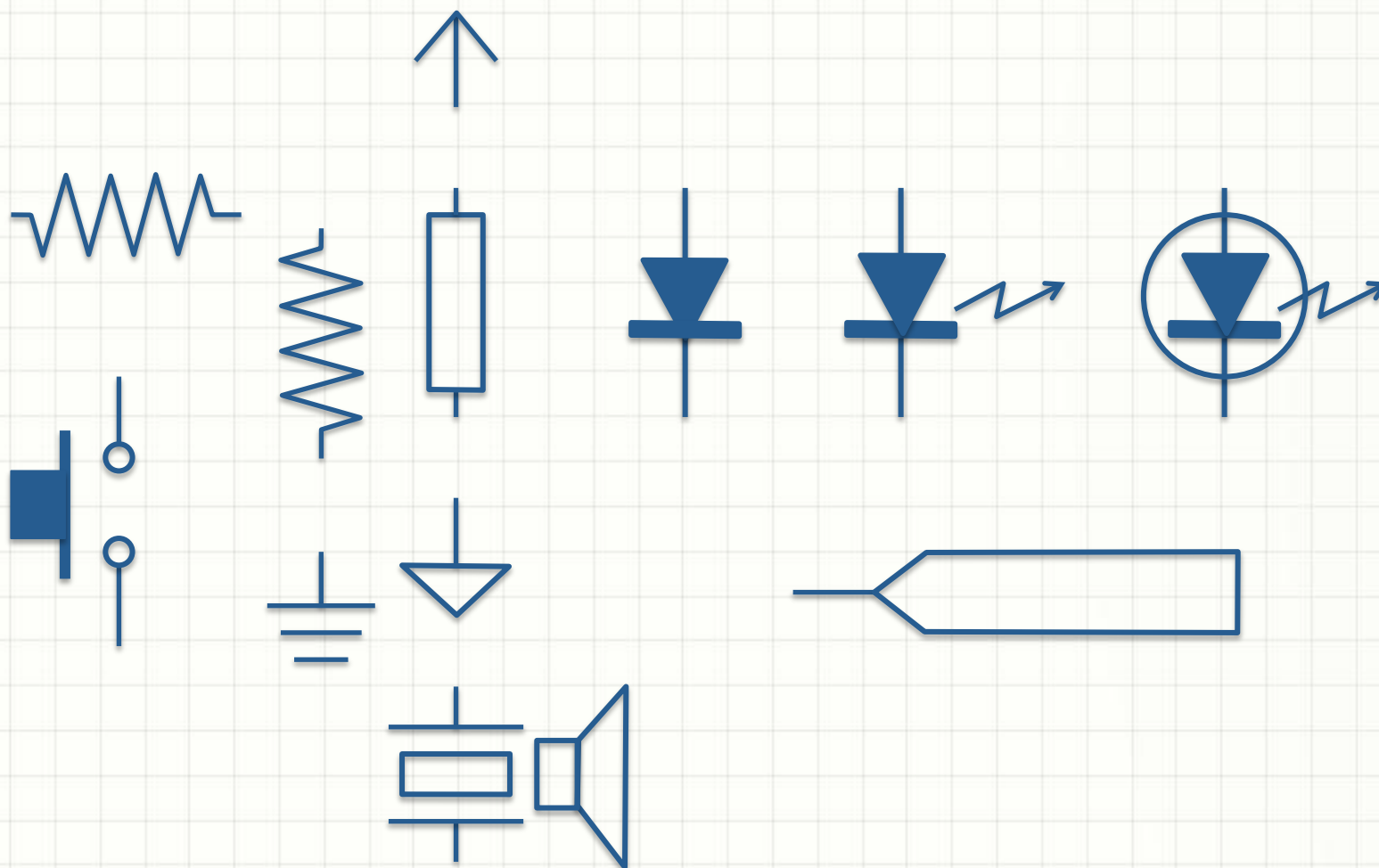
コンパイルしてバイナリファイルを生成

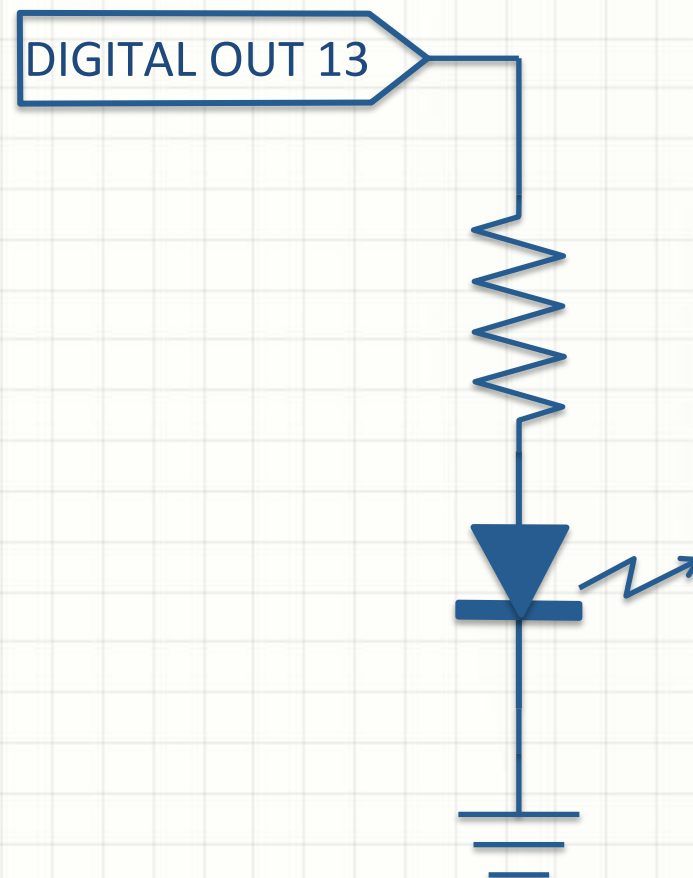
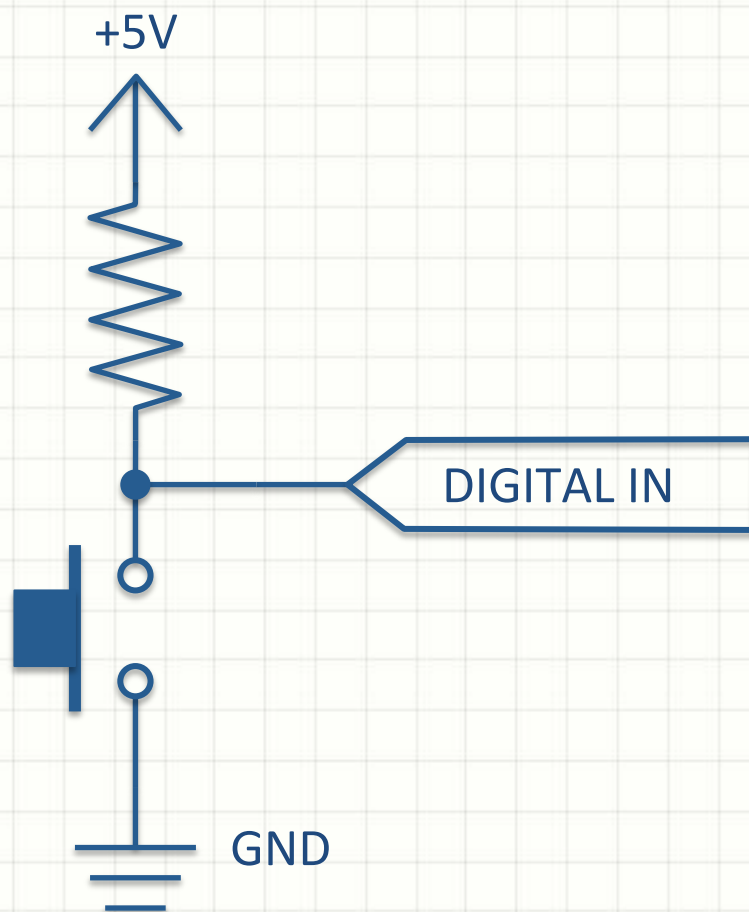
対象マイコンのプログラムメモリに書き込み

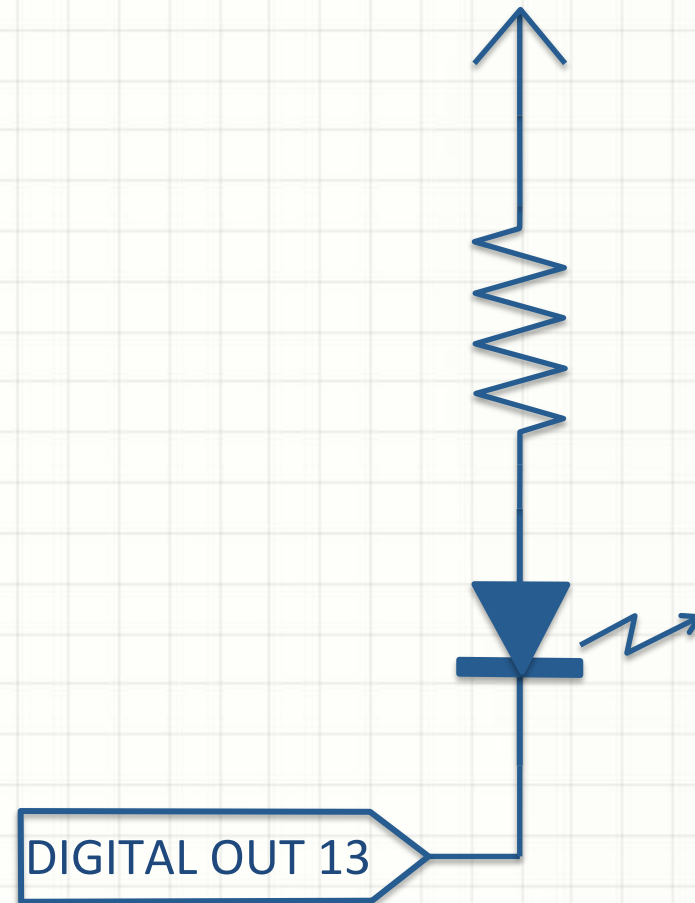
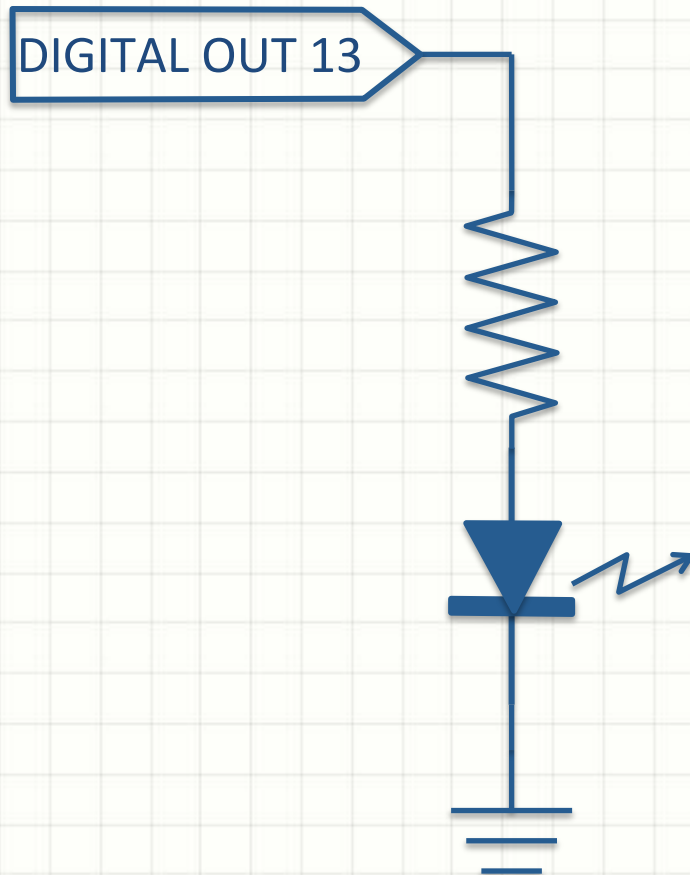
OSが必要な場合自分で書く

電源投入、リセットで main関数呼び出し

電源を切るかリセットまで動き続ける

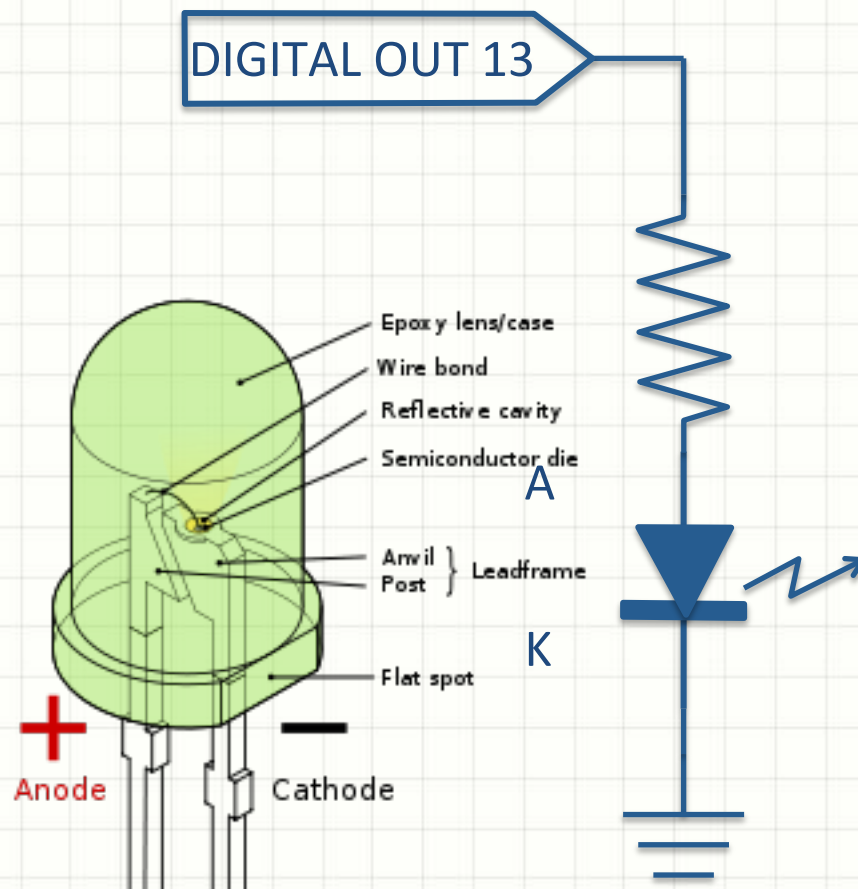






LEDを光らせるにはコツがある

なんと！ダイオードはオームの法則に従わない



- 1) 電流は一方通行
 - ダイオードはアノード A → K カソード
- 2) 電流が流れたとき生じる電位差は一定
 - 小さなLED1個は 2V 強 程度
- 3) 流せる電流に上限あり
(※流しすぎ危険！)
 - 15mA でじゅうぶん明るい

自分のPCでうまくいかない場合

1) ソフトはインストールされている？

➤ ダウンロードして、適切なディレクトリにおく

2) ボードのLEDは点灯する？

➤ ボードの故障、ケーブルの不具合、接続を確認

3) Arduino のCOMポート用ドライバはインストールされている？

➤ インストールが必要

Arduino IDE にドライバも同梱されている