
プログラミング演習Ⅱ

フィジカルコンピューティング

第2回 アナログ入出力とシリアル通信

中村, 小松, 小林, 鹿喰
(監修: 橋本直)

今日の内容

- **アナログ入力とアナログ出力**
 - センサの値を読み取る
 - LEDの明るさを変える
- **シリアル通信**
 - データの送信 (Arduino→Processing)
 - センサからの入力値をPCに送る
 - データの受信 (Processing→Arduino)
 - PCからの文字を受け取ってArduinoを制御する
- **色々なセンサーを試してみる**
- **カラーLED**

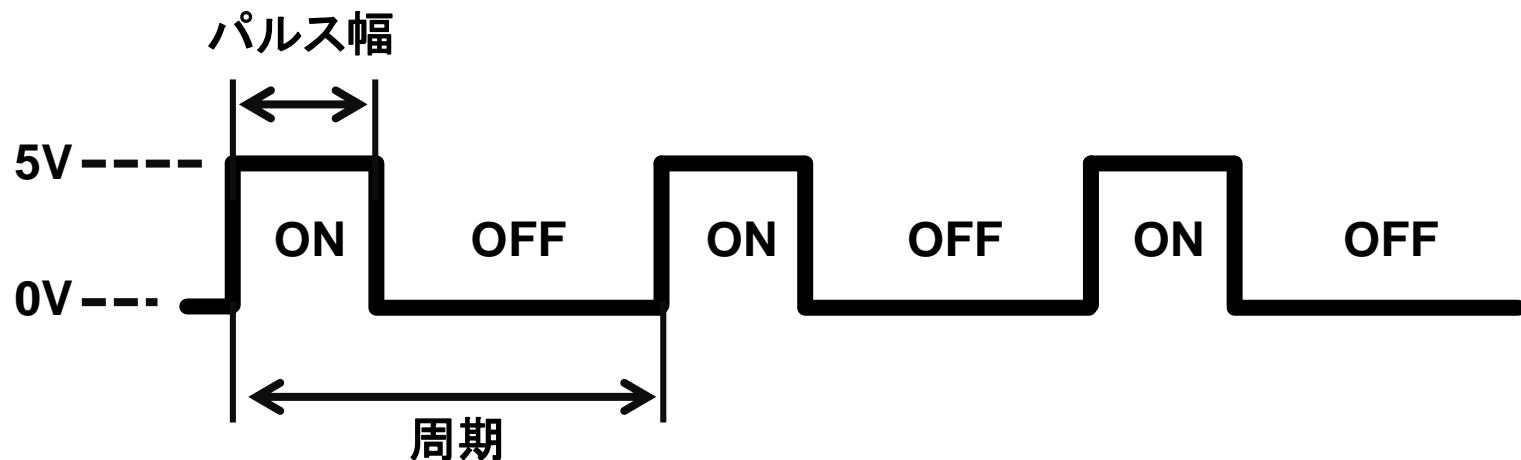
アナログ出力

Arduinoにおけるデジタル出力: LOWかHIGH か (0V or 5V)

Arduinoにおけるアナログ出力: PWM信号 (256段階)

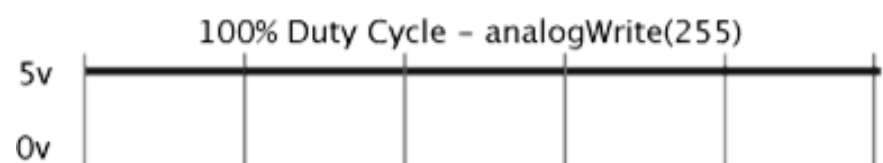
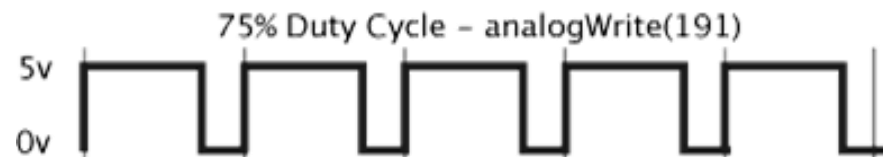
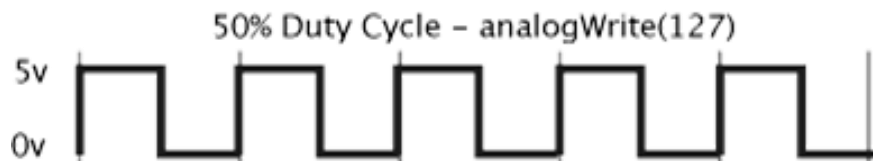
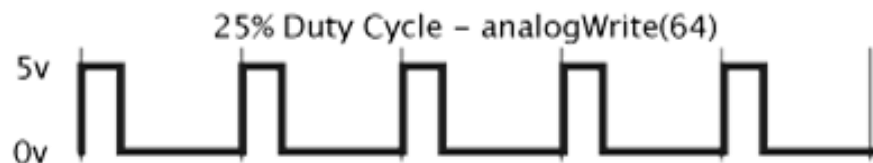
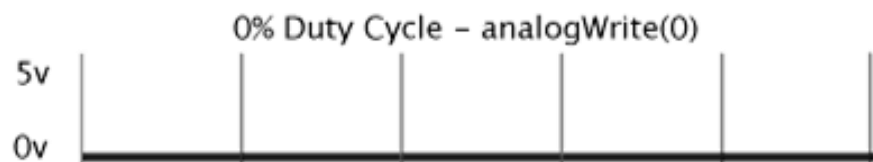
PWM = Pulse Width Modulation (パルス幅変調)

- ON・OFFを高速に切り替える手法
- ON時間とOFF時間の比(デューティ比)を変えてコントロール



アナログ出力の命令

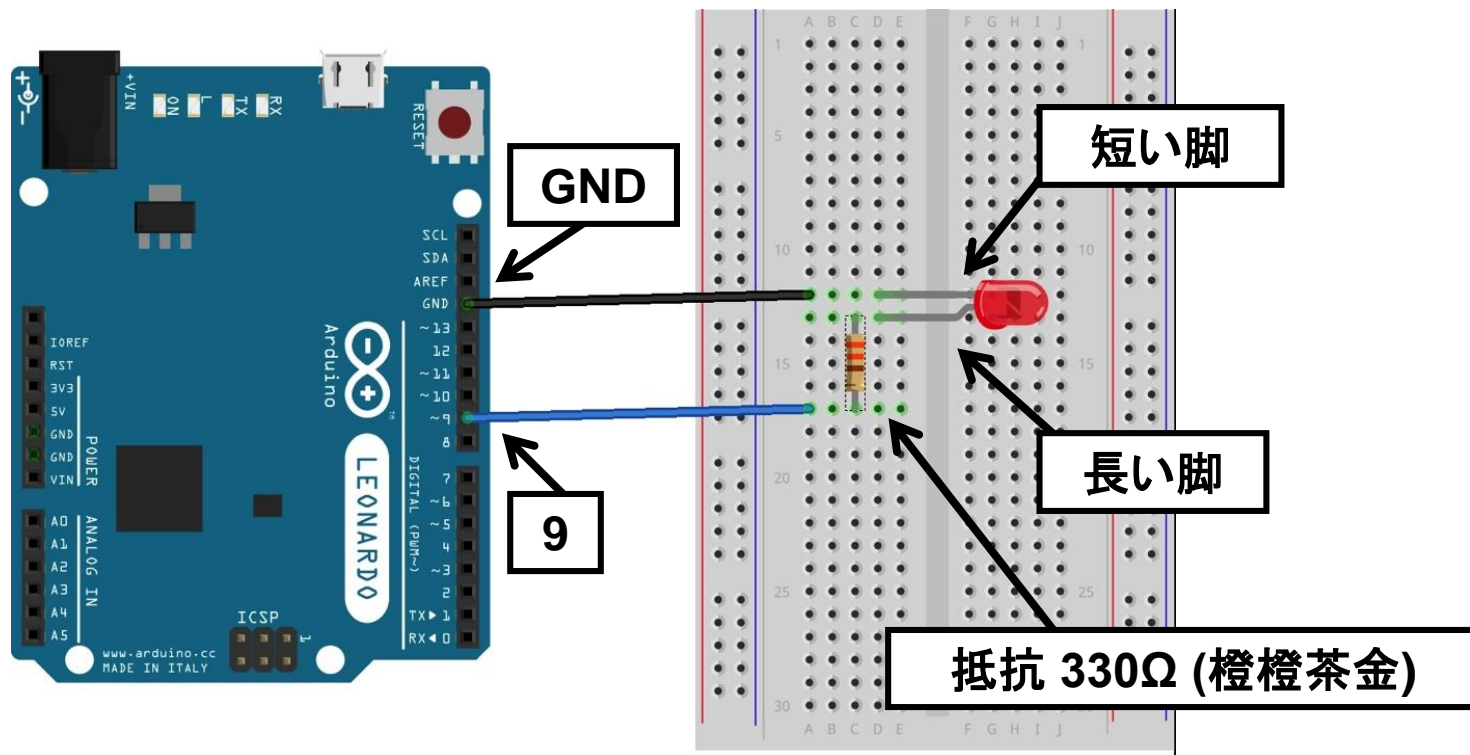
- **analogWrite(ピン番号, 出力値)**
 - 指定したピンに対してPWM出力する
 - 出力値には0 ~ 255を指定する
 - PWM出力できるのは3,5,6,9,10,11,13※ピン



※Arduino Unoでは13はPWM非対応

PWMでLEDの明るさを制御する回路

- 「~」マークがついた端子がPWM出力できる端子
- 抵抗は 330Ω ※前回と違うのはより明るくするため



LEDの明るさが滑らかに変化するプログラム

```
void setup() {  
  pinMode( 9, OUTPUT ); ← 9番ピンを出力に設定  
}  
  
void loop() {  
  for ( int i=0; i<=255; i++ ) { ← 0から255まで1ずつカウントアップ  
    analogWrite( 9, i ); ← 9番ピンでアナログ出力  
    delay(10);  
  }  
  for ( int i=255; i>=0; i-- ) { ← 255から0まで1ずつカウントダウン  
    analogWrite( 9, i ); ← 9番ピンでアナログ出力  
    delay(10);  
  }  
}
```

アナログ入力

Arduinoにおけるデジタル入力: LOW か HIGH か (0V or 5V)

Arduinoにおけるアナログ入力: 0V~5Vを多段階で刻んだ値

入力にはアナログ入力端子 A0~A5 を使う

- 電圧の変化を1024段階で読み取れる
- $5V/1024 = 4.9mV$ 刻み

アナログ入力の命令

- `analogRead(ピン番号)`
 - 指定したピンの入力電圧を読み取る
 - ピン番号は 0～5 を指定 (A0～A5に対応)
 - Leonardoでは, 6～11も使用可能 (デジタル入力端子の4, 6, 8, 9, 10, 12に対応)
 - int値 (0～1023) を返す

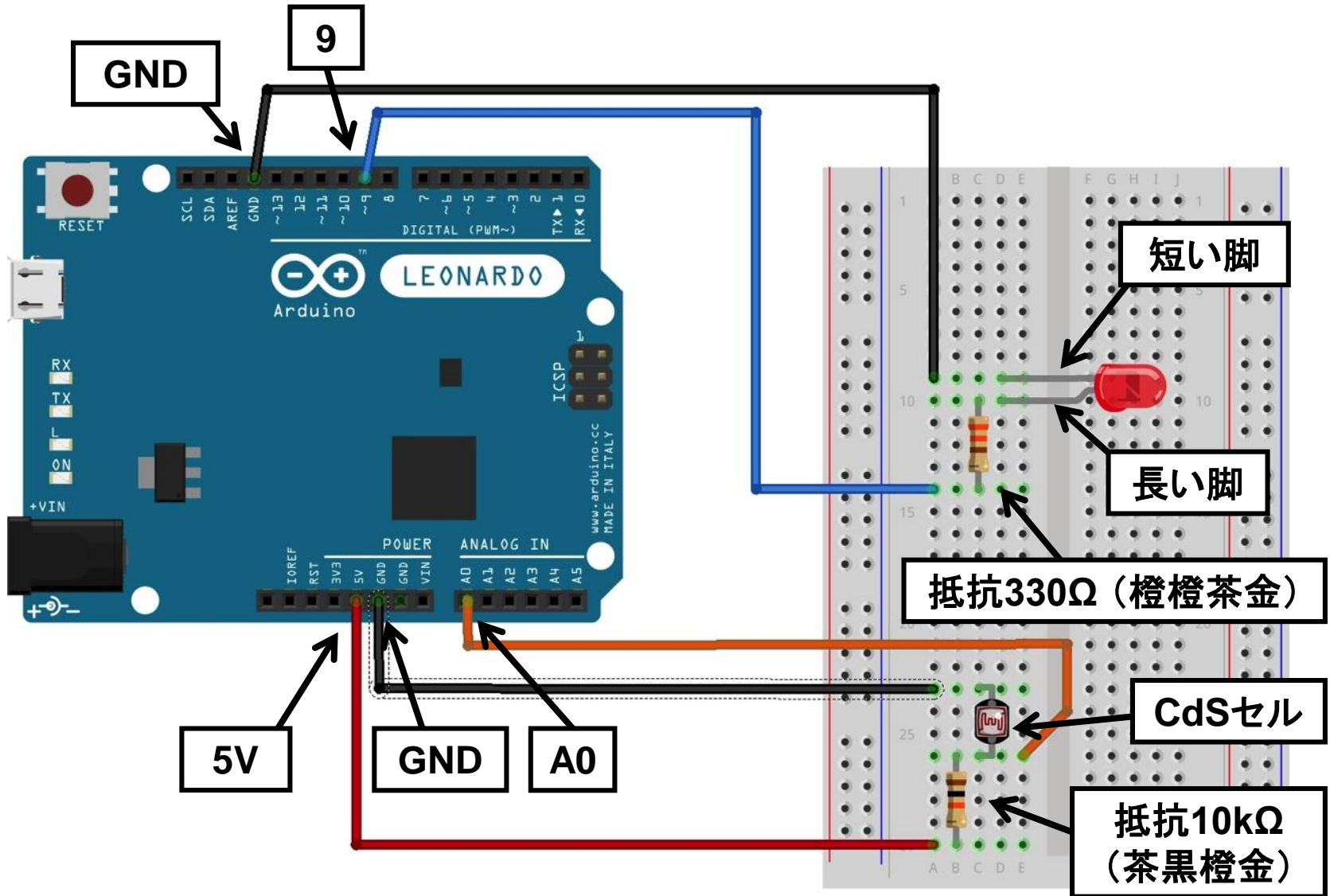
光センサ（CdSセル）

- あたる光が強くなると電気抵抗が低下する素子
- 硫化カドミウム（CdS）の性質を利用
- フォトレジスタ、光伝導セルとも呼ぶ

抵抗値の変化は、電気回路では
電圧値の変化として読み取る！



光の強さをセンシングする回路



周囲の明るさでLEDの点滅周期が変化するプログラム

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

9番ピンを出力に設定

アナログ入力端子(A0)についてはpinMode()不要

```
void loop() {  
  int value = analogRead(0);
```

アナログの0番ピンを読み取る

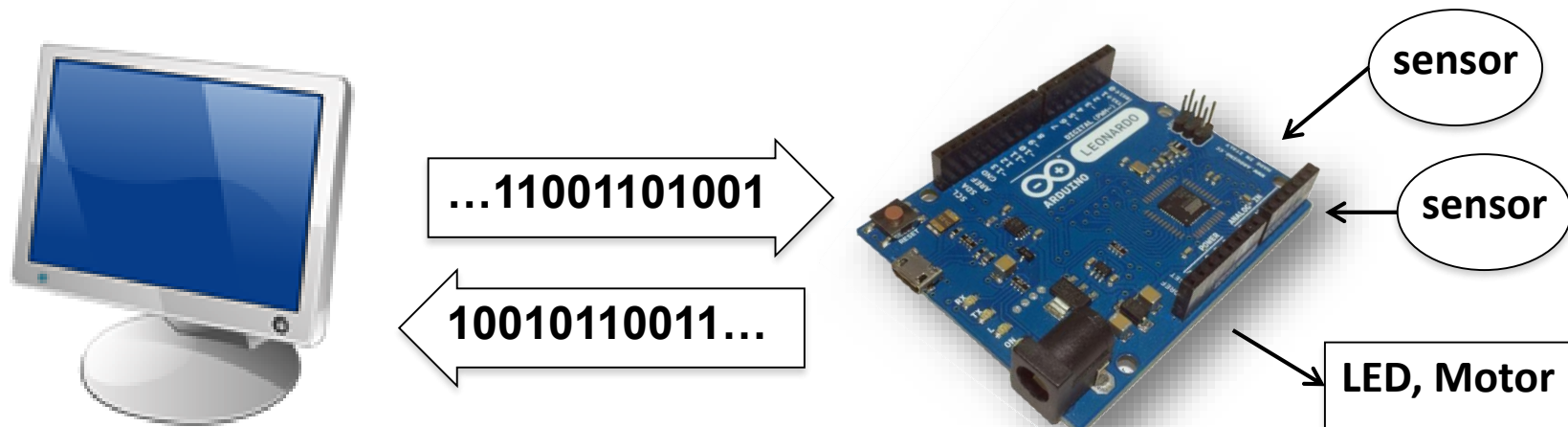
```
  digitalWrite( 9, HIGH );  
  delay( value );  
  digitalWrite( 9, LOW );  
  delay( value );
```

LEDを点滅させる

アナログ入力の値に応じて待ち時間を設定

```
}
```

シリアル通信



- 10010110011...のようなデータ信号を1ビットずつ送る通信方式
- ArduinoとPCを接続して、互いに数値や文字列のデータをやりとりできる
- Arduinoで計測したセンサ情報をPCに送ったり、PCからのコマンド送信によってLEDやモータを制御したりできる

シリアル通信に挑戦

- 回路は光センシングの回路そのまま

【データの送信】(Arduino→PC)

- 光センサの入力値をPCに送信してモニタリング

【データの受信】(PC→Aruidno)

- PCからコマンドを送ってLEDをコントロール

【データの送信】

光センサの入力値をPCに送信するプログラム

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

シリアル通信を開始する
(通信速度:9600bps)

```
void loop() {  
  int value = analogRead(0);  
  Serial.println(value);  
}
```

アナログの0番ピンを
読み取る

データをPCに送信する

命令の意味

- **Serial.begin(通信速度)**

- シリアル通信を開始する。通信速度は以下のいずれか
300bps, 1200bps, 2400bps, 4800bps, 9600bps, 19200bps,
57600bps, 115200bps

単位: bps = bits per second (1秒間に送るビット数)

- **Serial.print(データ) ... 改行なし**

- **Serial.println(データ) ... 改行あり**

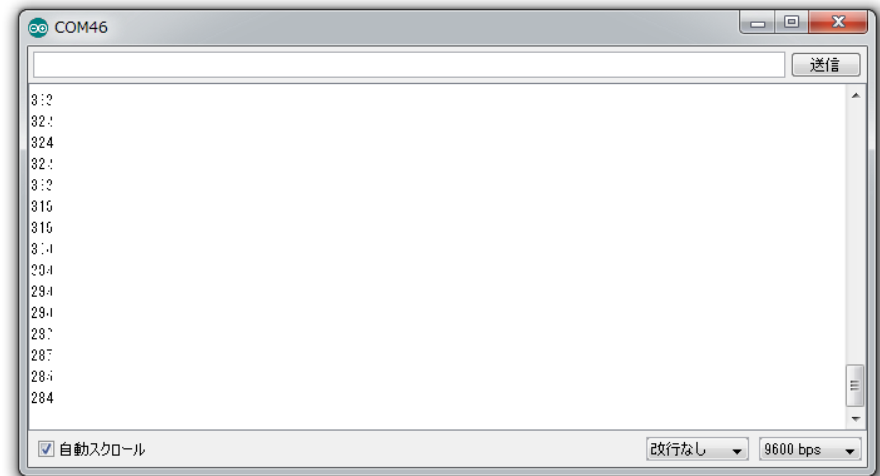
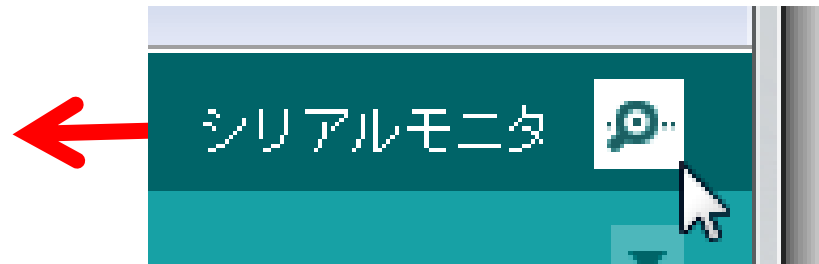
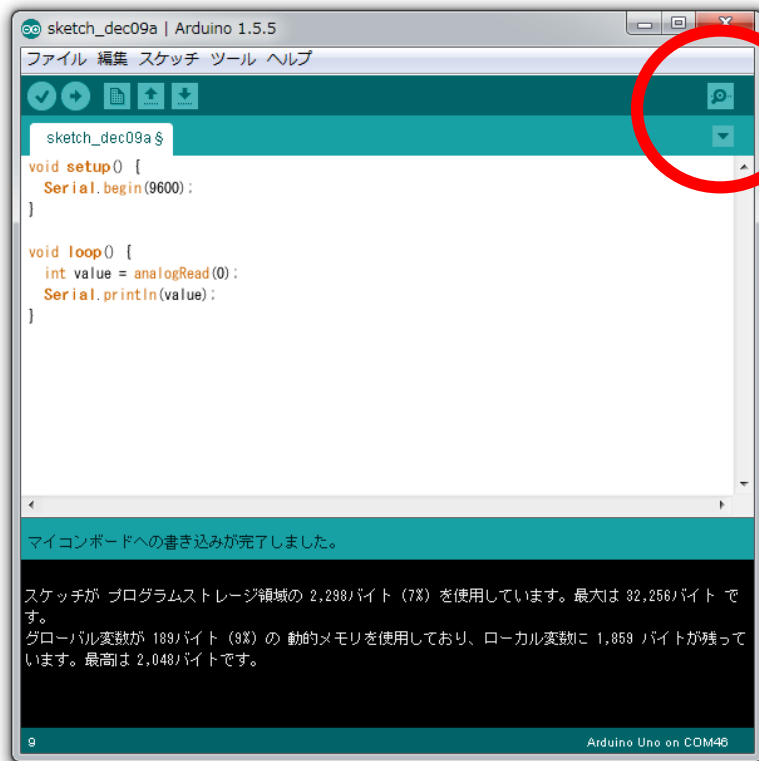
- 数値や文字列を送れる
- Arduinoでは「+」による文字列の結合はできないので注意

Serial.print()による文字の出力

- Serial.print(78) - “78”が出力される
- Serial.print(1.23456) - “1.23”が出力される
- Serial.print('N') - “N”が出力される
- Serial.print(“Hello world.”) - “Hello world.”と出力される

データのモニタリング

- ウィンドウ右上にある「シリアルモニタ」ボタンをクリック
- シリアルモニタが起動してデータが流れてくる



【データの受信】 コマンドでLED制御

```
void setup() {  
  Serial.begin( 9600 );  
  pinMode( 9, OUTPUT );  
}
```

← シリアル通信を開始する

← 9番ピンを出力に設定

```
void loop() {  
  if ( Serial.available() > 0 ) {  
    int data = Serial.read();  
    if ( data == 'a' ) {  
      digitalWrite( 9, HIGH );  
    }  
    else if ( data == 'b' ) {  
      digitalWrite( 9, LOW );  
    }  
  }  
}
```

← データがきているかチェック

← データを1バイト読み込む

← データが「a」だったら
LEDをONにする

← データが「b」だったら
LEDをOFFにする

命令の意味

- **Serial.available()**
 - 受信したデータのバイト数を返す
 - 1以上の値を返せばデータが着ていることになる
 - Arduinoは一度に128バイトまで保持する

- **Serial.read()**
 - 受信したデータの最初の1バイトを返す
 - 戻り値は int型

PCからのデータの送信

- シリアルモニタを起動する
- 入力欄にコマンド(「a」または「b」)を入力して送信ボタン(またはエンターキー)を押す
- 「a」を送った時にLEDがONになり、「b」を送った時にLEDがOFFになる

(補足)文字「a」を送るということ

- 文字にはそれぞれ「文字コード」と呼ばれる識別番号が割り当てられている。

(例) 「a」は97、「A」は65、「1」は49

参考: http://www9.plala.or.jp/sgwr-t/c_sub/ascii.html

- 「a」の文字コード(97)を2進数で表すと「01100001」
- この1/0の並びをON/OFFの電気信号に置き換えて伝送している。

(補足) int型での受信データの判定

- Serial.read()は受信データをint型で返す
- 文字を受信したときは？
 - 文字コードが受信される
 - すなわち、「a」という文字がきた時は、97というint型の値を受け取っている
 - なので、このように判定することもできる ('a' は 97 と置換可能)

```
int data = Serial.read();  
if ( data == 97 ) {  
    digitalWrite( 9, HIGH );  
}
```

曲げセンサ

- 曲げ量に応じて抵抗値が変化するセンサ
- 光センサを取り外して曲げセンサで置き換えてみよう(根元が折れやすいので注意！)
- 曲げに応じて値はどのように変化するだろうか



感圧センサ

- 押す力に応じて抵抗値が変化するセンサ
- 曲げセンサをはずして感圧センサで置き換えてみよう(根元が折れやすいので注意！)
- 力に応じて値はどのように変化するだろうか

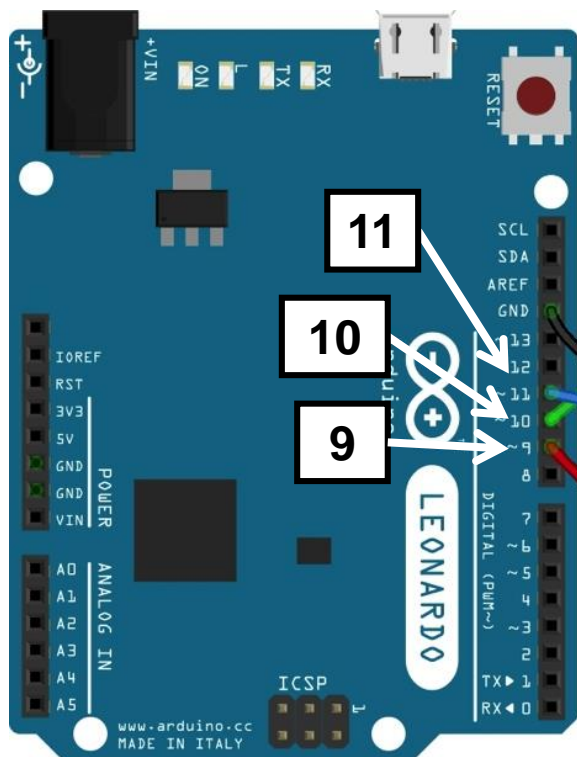


フルカラーLED

- R,G,B 3色のLEDが1つになった素子。
- 各色の強度を調整して色々な色が出せる



フルカラーLEDを点灯する回路



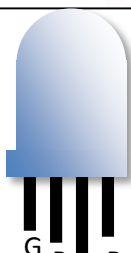
GND

抵抗 330Ω (橙橙茶金)

フルカラーLED

一番長いピン

抵抗 330Ω (橙橙茶金)



コモンカソード

フルカラーLEDのピン配置

フルカラーLED(アナログ出力の応用)

```
void setup() {  
  pinMode( 9, OUTPUT );  
  pinMode( 10, OUTPUT );  
  pinMode( 11, OUTPUT );  
}
```

← 9,10,11番ピンを出力に設定
どれも、~が付いているピン

```
void loop() {  
  analogWrite( 9, random(0,255) );  
  analogWrite( 10, random(0,255) );  
  analogWrite( 11, random(0,255) );  
  delay(1000);  
}
```

← 赤の強度を設定
← 緑の強度を設定
← 青の強度を設定

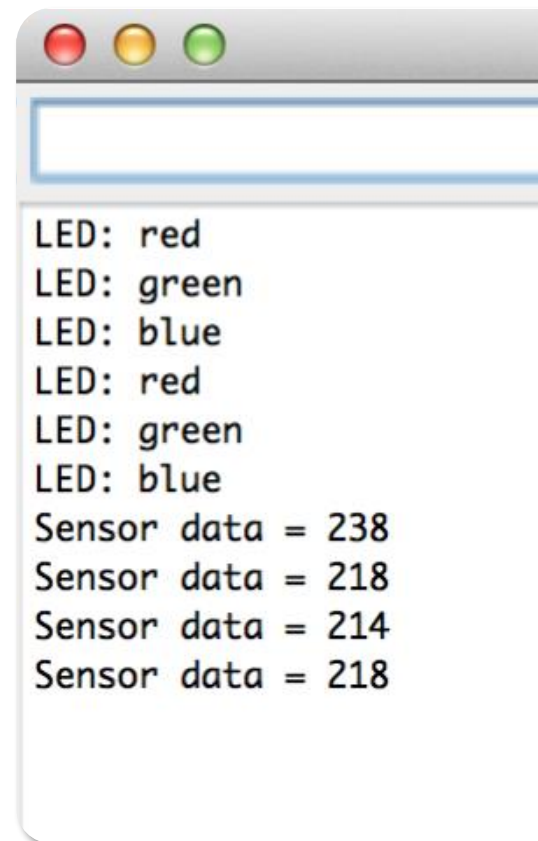
練習問題1

- コマンド入力によって以下の反応を返すプログラムにしてみよう

- 「r」という文字を送信したらカラーLEDが赤く点灯して「LED: red」という文字列を返す
- 「g」という文字を送信したらカラーLEDが緑に点灯して「LED: green」という文字列を返す
- 「b」という文字を送信したらカラーLEDが青く点灯して「LED: blue」という文字列を返す

- 「c」という文字を送信したら、光センサの値を「sensor data= ???」という形式で返す

「rgbrgbr」と送信すると、どうなるだろう？



```
LED: red
LED: green
LED: blue
LED: red
LED: green
LED: blue
Sensor data = 238
Sensor data = 218
Sensor data = 214
Sensor data = 218
```

練習問題2

- 光センサに手をかざしたときだけLEDが点灯するプログラムを作ってみよう
 - 【ヒント1】 if文によって、「センサの値がいくつ以上のときに点灯(or消灯)する」という分岐を作ってみよう。
 - 【ヒント2】 シリアル通信を使って、手をかざしているときに、センサの入力値がどのくらいになっているかチェックして閾値(しきいち)を決めよう。
- 光センサに手をかざすと、カラーLEDの光り方が変化するプログラムを作ってみよう