Arduino入門勉強会 #2

【センサによる入出力】

平成27年7月7日

ソフトピアジャパン　ドリーム・コア1F　ネクストコア

センサ入出力

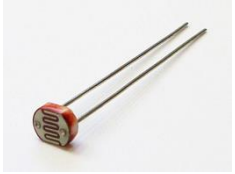
Arduinoにセンサなど様々な入出力部品を取り付け動作させてみます。
今回の勉強会で扱うのは以下の部品です。

入力：タクトスイッチ、ボリューム抵抗(半固定抵抗)、
光センサ(CDS)、温度センサ

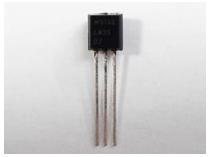
出力：LED(単色)、ピエゾスピーカー、サーボ、フルカラーLED

タクトスイッチ

ボリューム抵抗
（半固定抵抗）



光センサ(CDS)



温度センサ

LED(単色)

ピエゾスピーカー

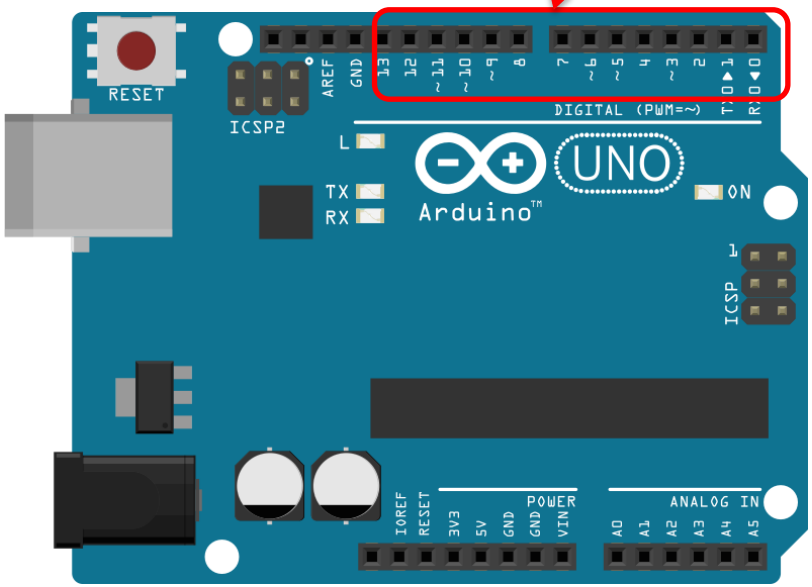
サーボ

フルカラーLED

デジタル入力

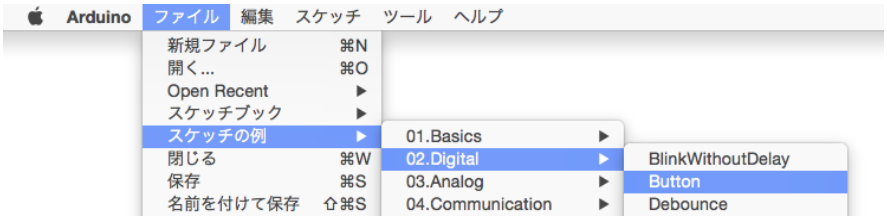
デジタル入力ピンではdigitalRead関数を用いて5V電圧のON/OFF状態を読み取ることができます。

デジタル入出力ピン(pin0～13)

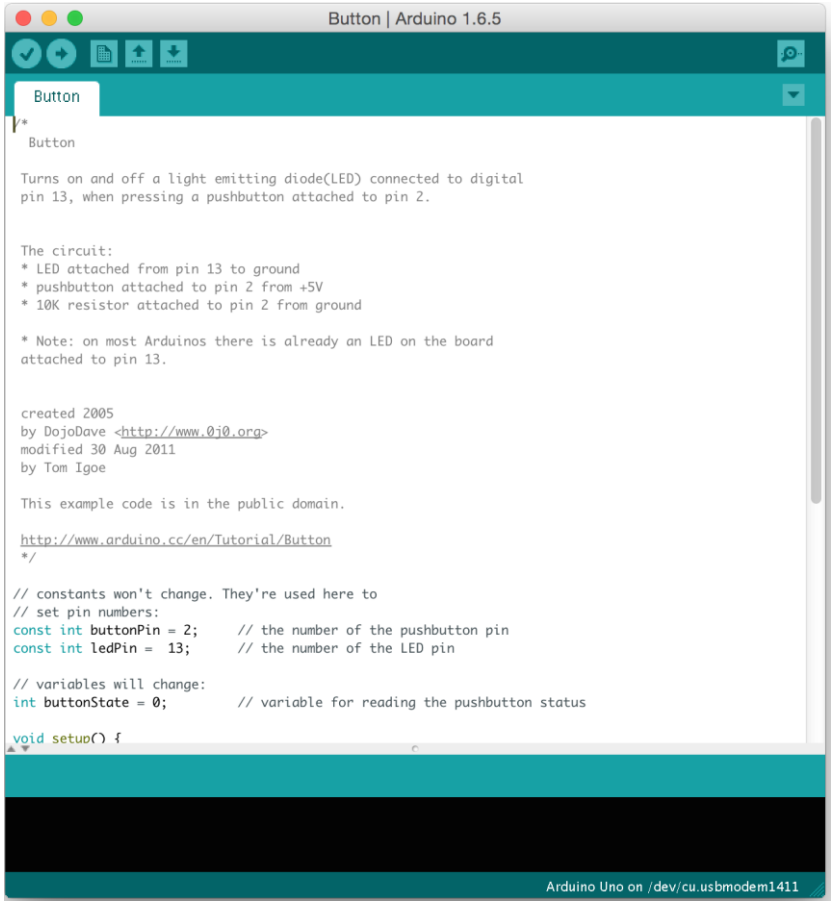


サンプル「Button」

タクトスイッチのボタンを押すとLEDが点灯するサンプルスケッチです。

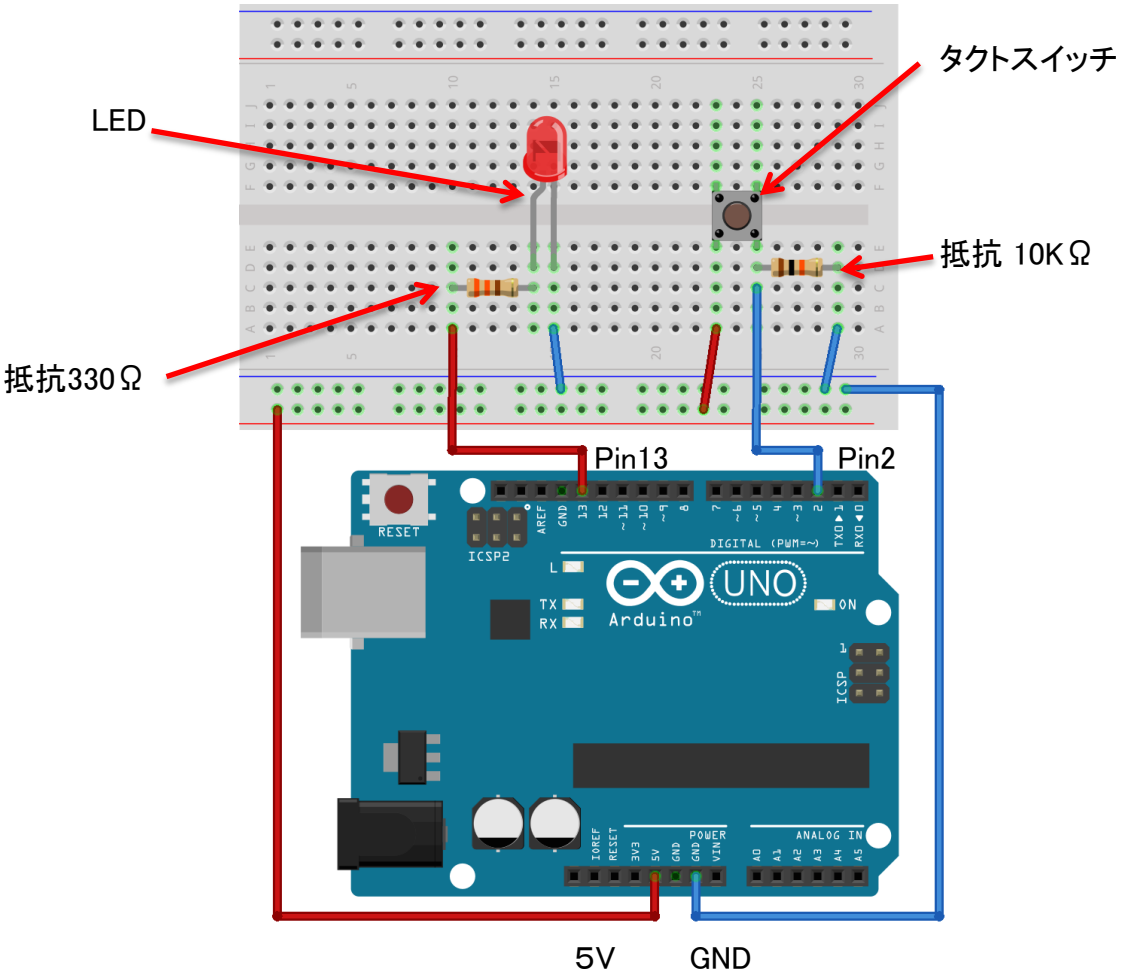


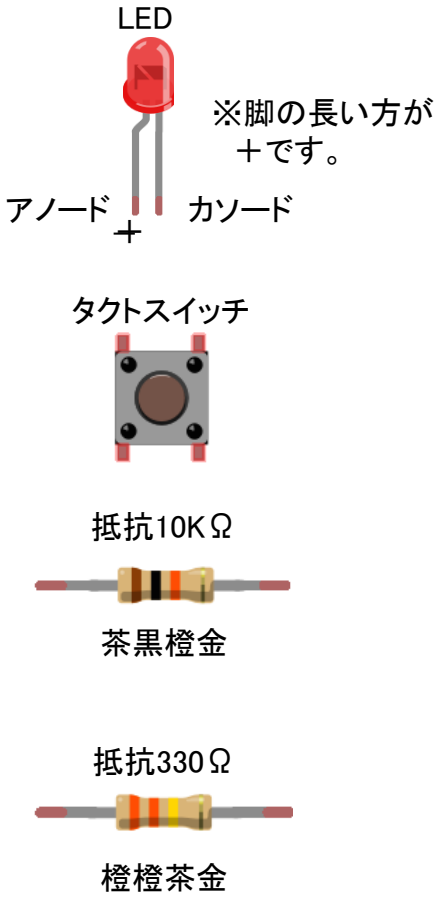
メニューから
「ファイル」→「スケッチの例」→「02.digital」→「Button」
と選択します。



サンプル「Button」:回路

スイッチのON／OFF状態を判断するための回路と、LEDを点灯するための回路の組み合わせです。





サンプル「Button」:コード

定数の宣言

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
```

ボタン(スイッチ)に使用するピン番号の定義

LEDに使うデジタルピンの番号の定義

変数の宣言

```
// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status
```

ボタンの状態を表す変数

初期化

```
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
```

LED用のピンを出力に設定

ボタン用のピンを入力に設定

メイン処理の記述

```
void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

ボタン用ピンにかかっている電圧がON(HIGH)かOFF(LOW)か読み取り

条件分岐:もしも、ボタン用ピンの電圧がON(HIGH)ならば

LEDを点ける(LEDピンの電圧をHIGHにする)

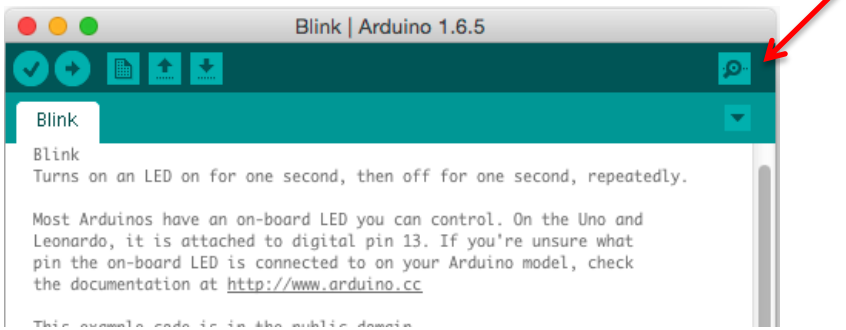
ボタンがON(HIGH)の場合

LEDを消す(LEDピンの電圧をLOWにする)

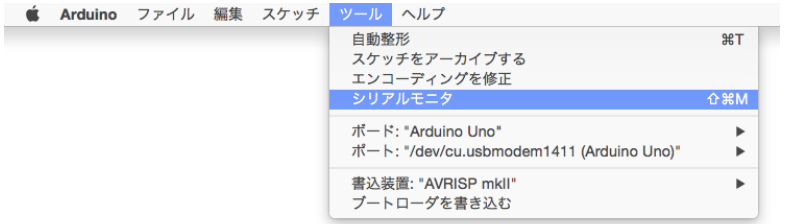
ボタンがOFF(LOW)の場合

シリアルモニタを使ってみよう

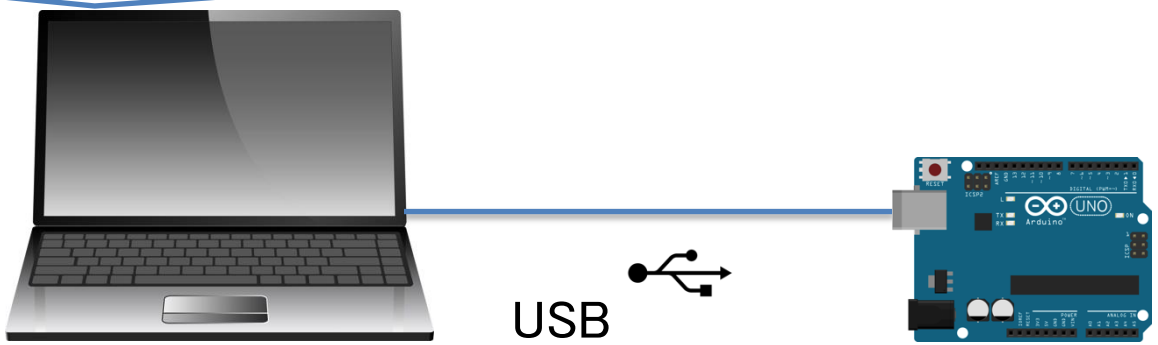
ArduinoとPCはUSBケーブルを通じてシリアル通信をすることができます。
PCとArduinoをUSBケーブルで接続した後、ウィンドウ右上のシリアルモニタアイコンをクリックします。



シリアルモニタアイコン



メニューからの場合
「ツール」→「シリアルモニタ」と選択します。

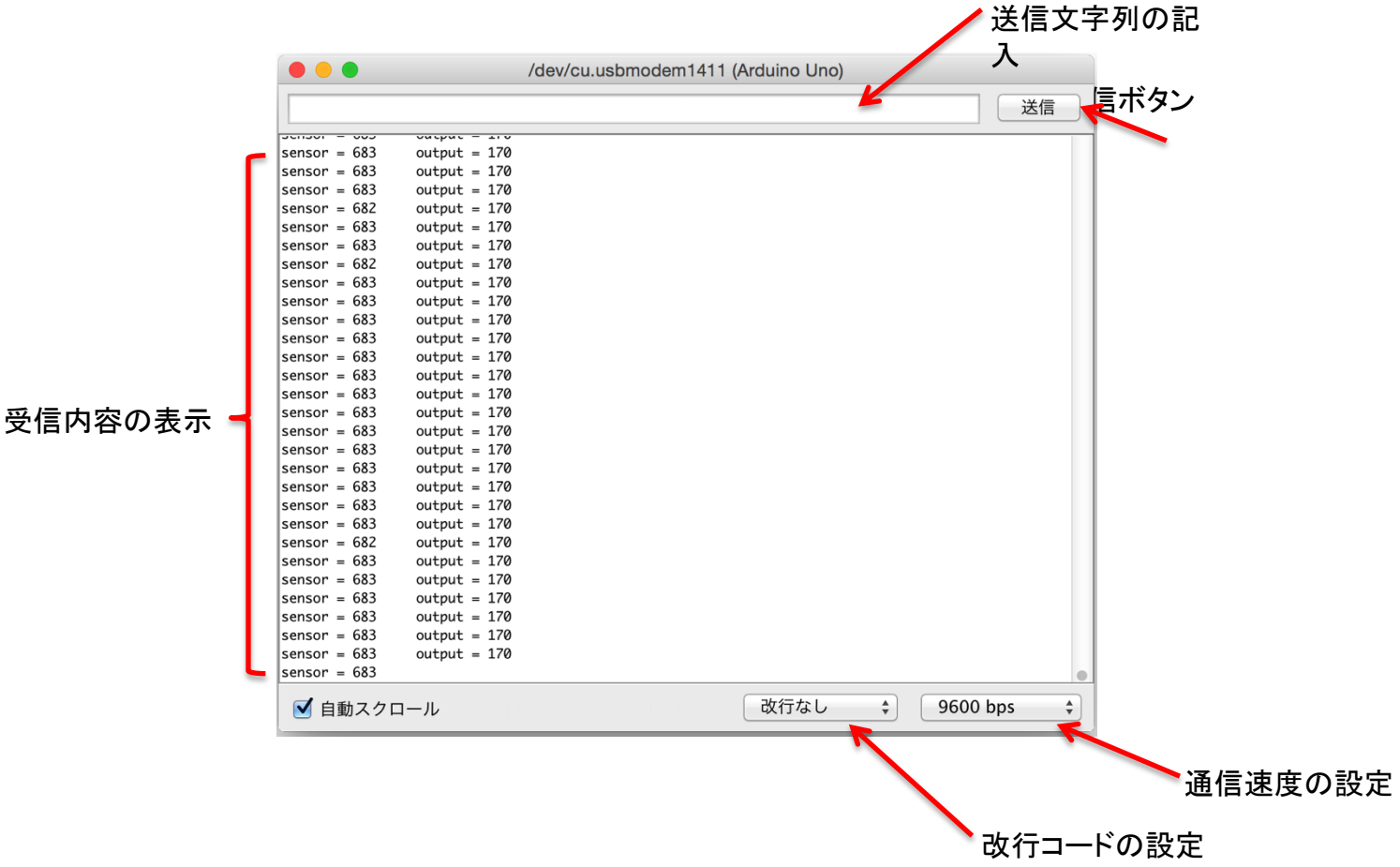


PC(Windows,Mac,Linux)

Arduinoボード

シリアルモニタ

シリアルモニタを使用するとAruduinoからシリアル通信で送信された情報を表示させることが出来ます。

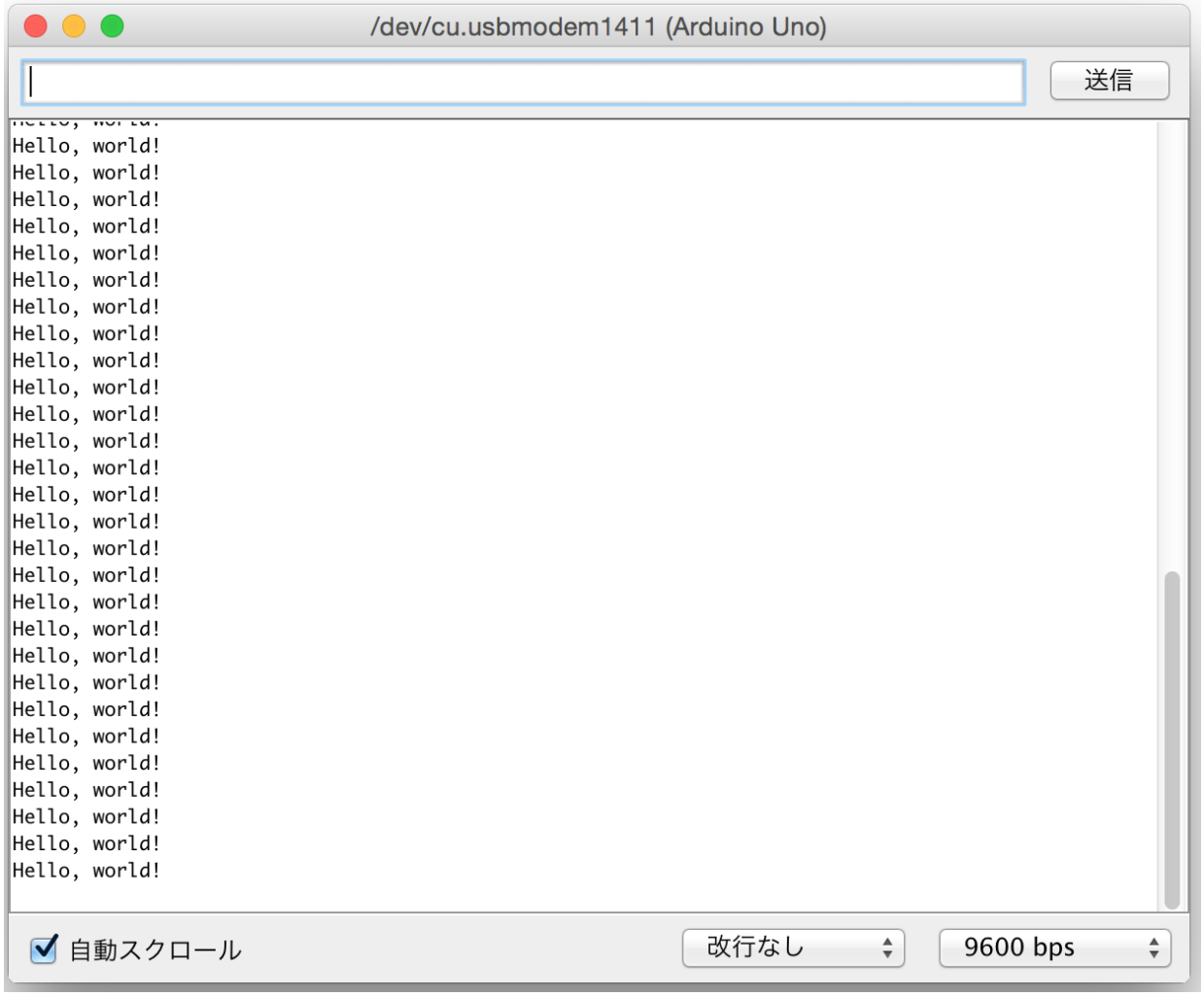


シリアルモニタ

以下のプログラムを書き込み、シリアルモニタで表示してみましょう

シリアルモニタ

Arduinoから送信された文字列 “Hello, world!” が表示されます。



コード解説

Arduinoではシリアル通信の処理が予め用意されており、
Serial.begin()関数を呼び出せば、以後シリアル通信が可能になります。

初期化

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}
```

シリアル通信の開始(通信速度9600bps)

メイン処理

```
void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("Hello, world!");
  delay(300);
}
```

シリアルに文字列 “Hello, world!” を書き出す

300ミリ秒待機する

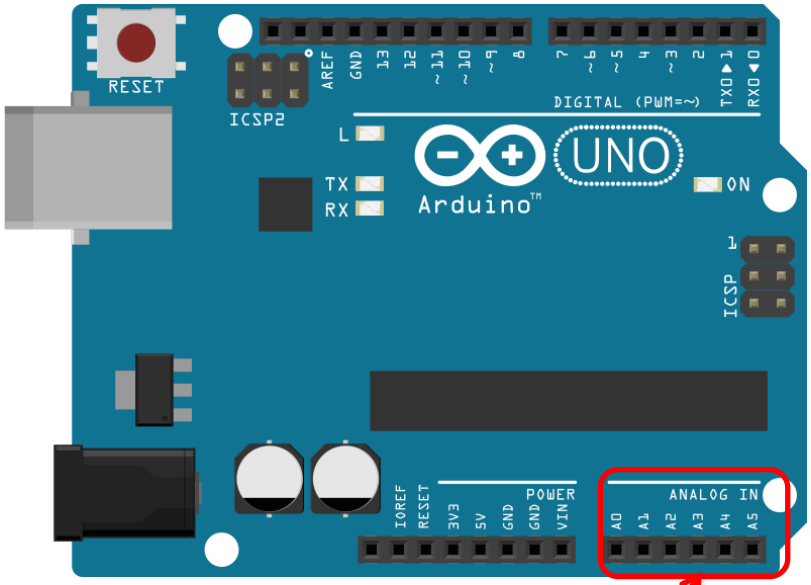
※シリアルへの書き出しにはいくつかの関数が用意されており、
文字列出力の場合は、改行有りと改行なしの2つが用意されています。

・改行有り：Serial.println(“Hello, world!”);

・改行無し：Serial.print(“Hello, world!”);

アナログ入力

・アナログ入力ピンではanalogRead関数を用いて0から5Vまでの電圧を0から1023までの値としてを読み取ることができます。

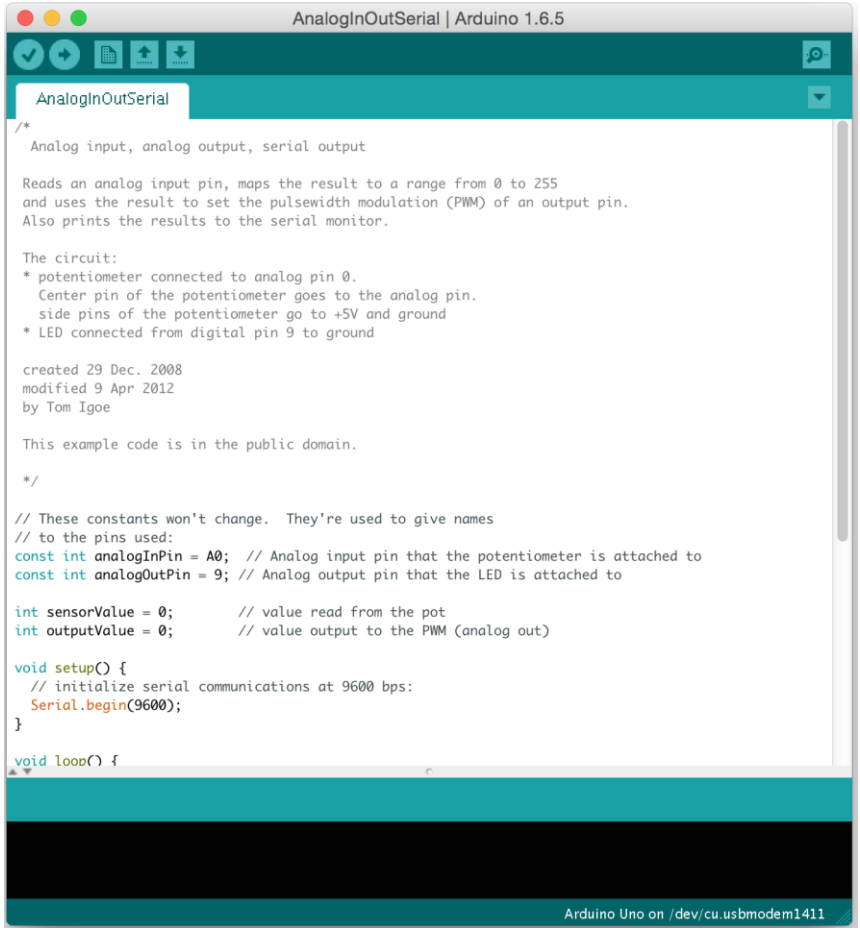


アナログ入力ピン(A0~A5)

サンプル「AnalogInOutSerial」

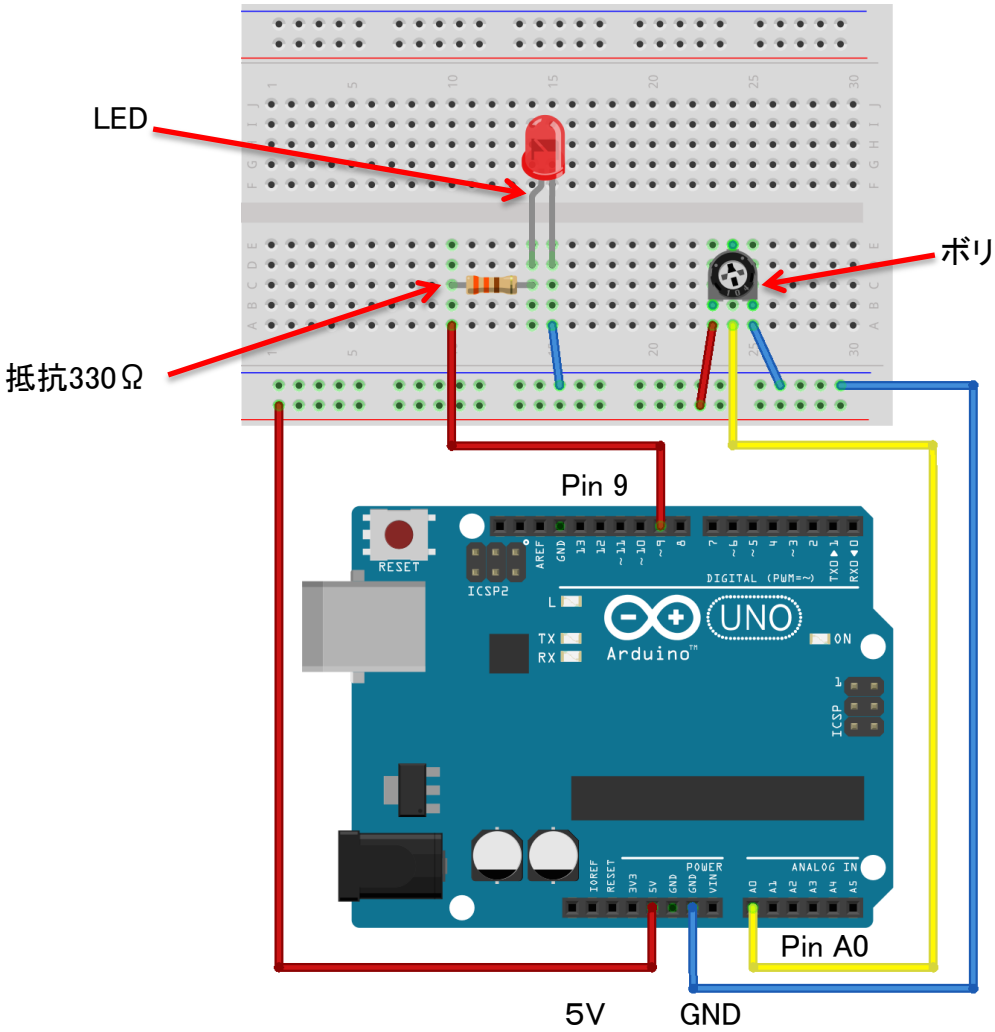
センサからのアナログ入力値に応じてLEDの明るさが変化するサンプルスケッチです。PCでセンサの入力値をチェックできるようシリアル通信も行います。

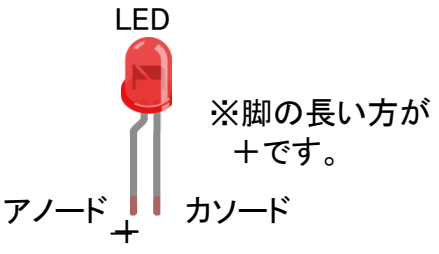
メニューから
「ファイル」→「スケッチの例」
→「03.Analog」→
「AnalogInOutSerial」
と選択します。



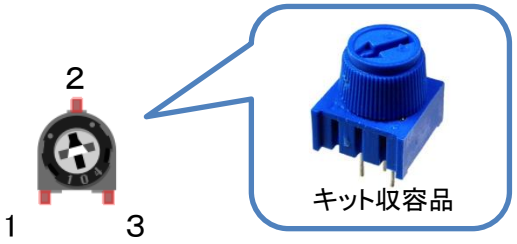
「AnalogInOutSerial」: 回路

センサの状態を読み取る回路と、LEDを点灯するための回路の組み合わせです。





ボリューム抵抗10KΩ



ピン1を電源(5V)、
ピン2をアナログ入力ピン、
ピン3をGNDへ繋ぎます。

抵抗330Ω

橙橙茶金

「AnalogInOutSerial」: コード

定数の宣言

```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
```

アナログ入力に使用するピン番号の定義

LEDに使うピンの番号の定義

変数の宣言

```
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
```

センサー入力値

LED用出力値

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```

シリアル通信の開始(通信速度9600bps)

メイン処理の記述

```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("¥t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```

アナログ入力値を読み取る(0～1023)

値の調整:0～1023のsensorValueを0～255に割り振ってoutputValueへ

アナログ出力ピンから出力

文字列「sensor =」をシリアルポートに書き出す

センサ読み取り値をシリアルポートに書き出す

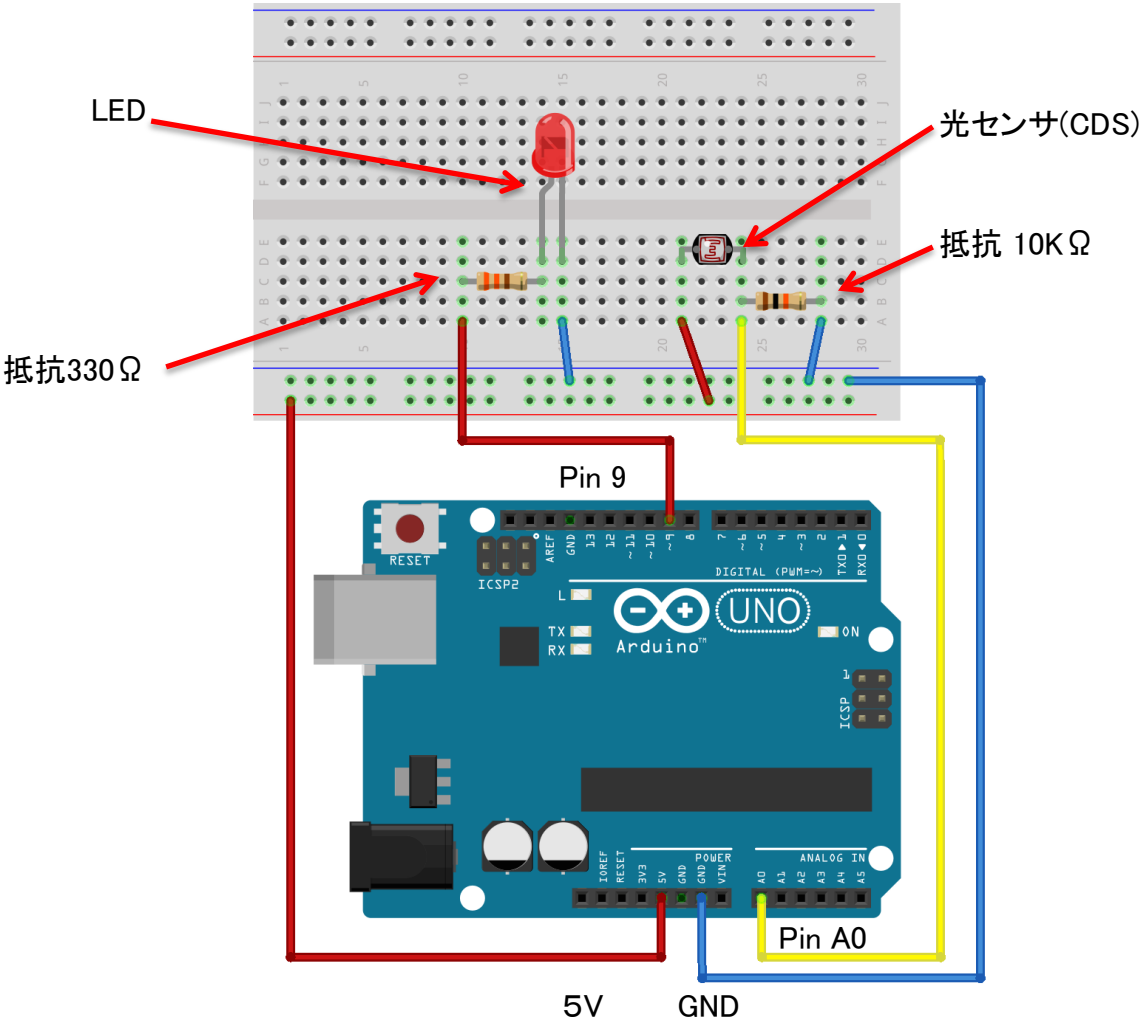
文字列「¥t output = 」をシリアルポートに書き出す

LED用出力値をシリアルポートに書き出す

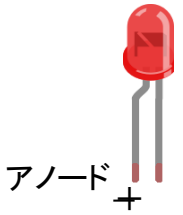
2ミリ秒待機

入力センサを変えてみよう

入力に用いるセンサをボリューム抵抗から光センサ(CSD)に変更します。



LED



※脚の長い方が＋です。

光センサ(CDS)

※極性はありません

抵抗10KΩ

茶黒橙金

抵抗330Ω

橙橙茶金

反応を良くしてみよう

シリアルモニタの情報を基にセンサの値をより有効に使用しよう。

定数の宣言

```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
```

変数の宣言

```
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
```

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```

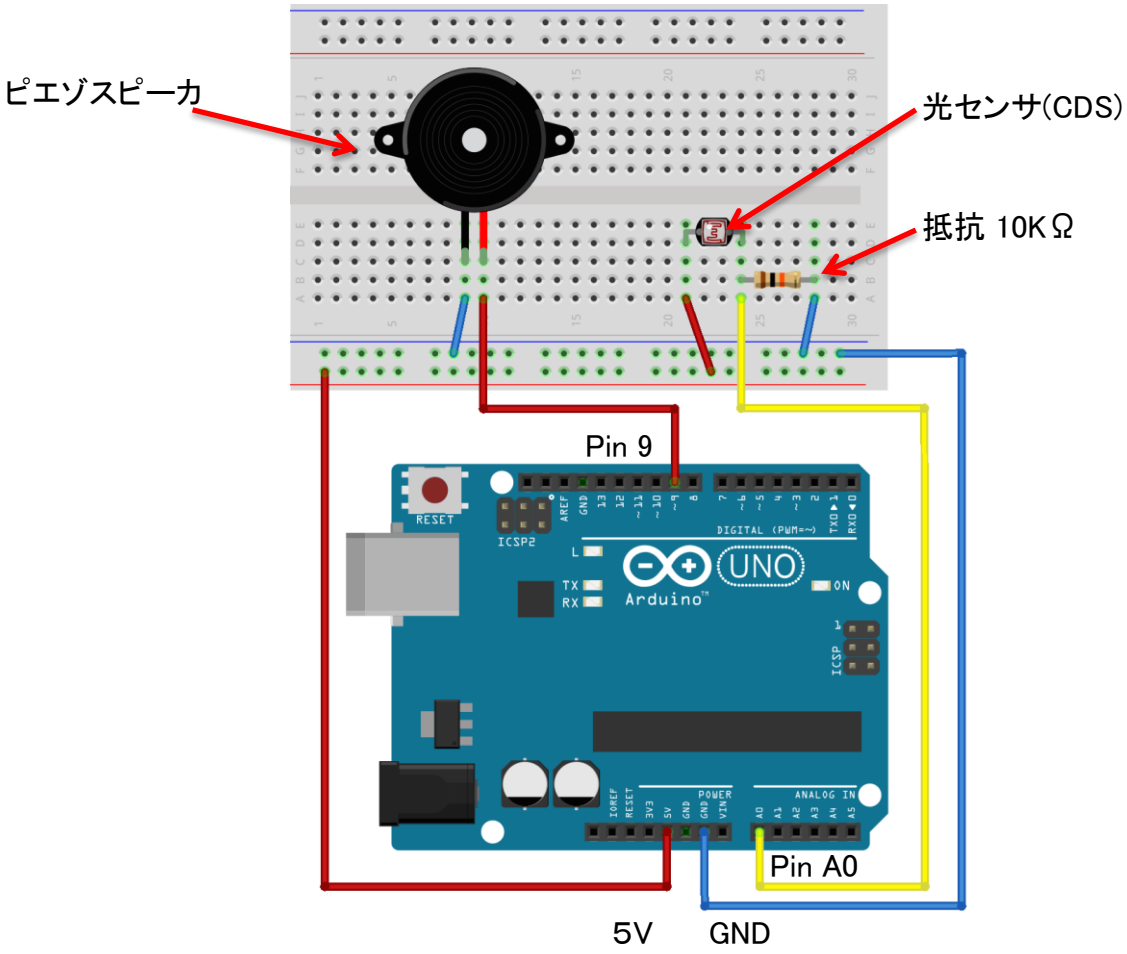
メイン処理の記述

```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 150, 700, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("¥t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```

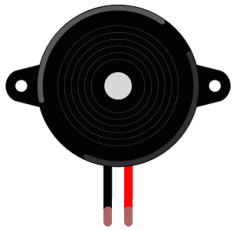
光センサからの入力値が150から700までなので
そこを有効な値として使用する。

出力装置を変えてみよう①

LEDをピエゾスピーカーと変えてみましょう。



ピエゾスピーカ



光センサ(CDS)

※極性はありません

抵抗10KΩ

茶黒橙金

tone()関数

tone(pin, frequency)
tone(pin, frequency, duration)

指定した周波数の矩形波を生成します。時間(duration)を指定しなかった場合、noTone()を実行するまで動作を続けます。出力ピンに圧電ブザーやスピーカに接続することで、一定ピッチの音を再生できます。

【パラメータ】
pin: トーンを出力するピン
frequency: 周波数(Hz)
duration: 出力する時間をミリ秒で指定できます(オプション)

スピーカー用にコード書き換え

定数の宣言

```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
```

変数の宣言

```
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
```

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```

メイン処理の記述

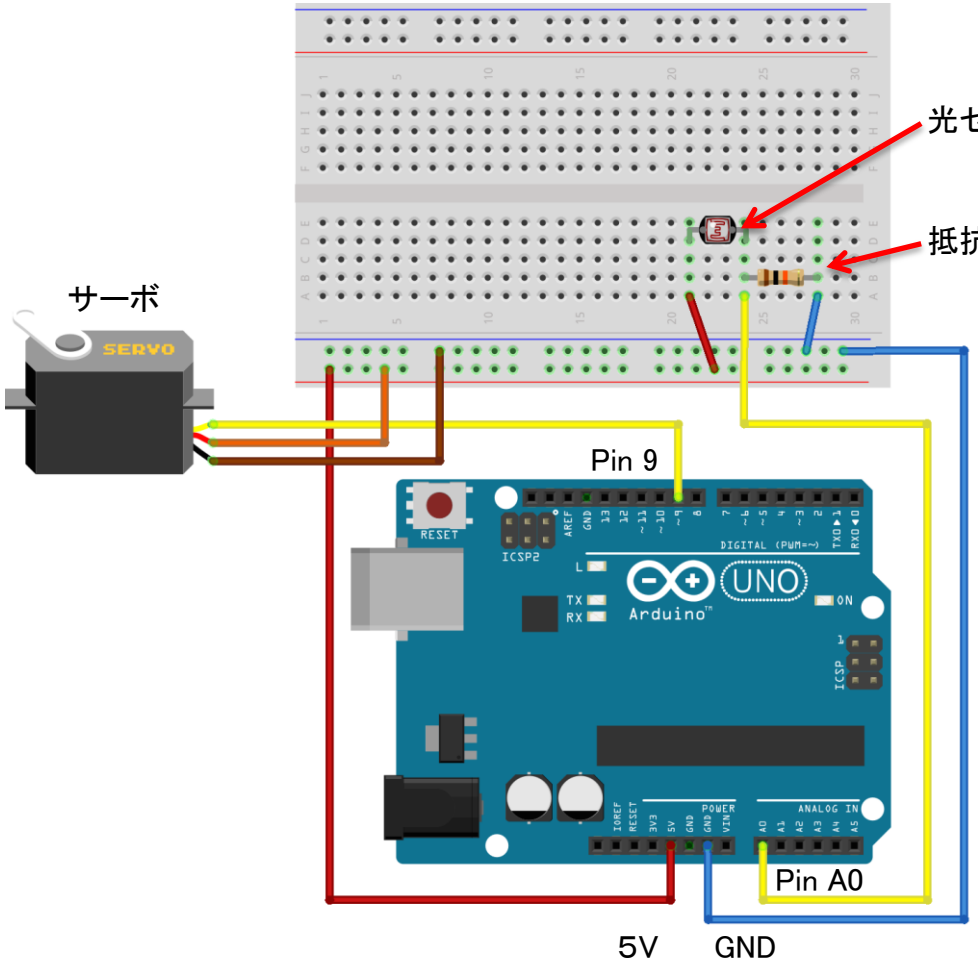
```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 150, 700, 20, 1000);
  // change the analog out value:
  tone(analogOutPin, outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("¥t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```

値の調整：150～700のsensorValueを20～1000に割り振ってoutputValueへ

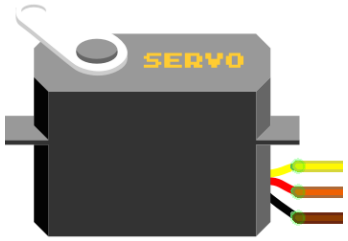
アナログ出力ピンから指定の周波数の波形出力する関数

出力装置を変えてみよう②

ピエゾスピーカーをサーボに変えてみましょう。



サーボ



黄:信号
赤:電源(5V)
茶:GND

光センサ(CDS)

※極性はありません

抵抗10KΩ

茶黒橙金

Servoクラス

このライブラリはRCサーボモータのコントロールに用います。標準的なサーボでは0から180度の範囲でシャフトの位置(角度)を指定します。

コーディング例：ピン9に接続されたサーボを90度にセットする。

```
#include <Servo.h>

Servo myservo;

void setup(){
        myservo.attach(9);
        myservo.write(90);
}

void loop() {}
```

Servoクラスのヘッダファイルを読み込むする。

Servoクラスのインスタンスをmyservoとして宣言

デジタル9番ピンをサーボ制御に指定

サーボのシャフトを90度にセットする

サーボ用にコード書き換え

ヘッダの宣言

```
#include <Servo.h>
```

サーボクラスのヘッダファイルを読み込み

```
// These constants won't change.  They're used to give names
// to the pins used:
```

定数の宣言

```
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
```

変数の宣言

```
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
Servo myservo;
```

サーボの変数名の宣言

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  myservo.attach(analogOutPin);
}
```

サーボ制御ピンを割り当て

メイン処理の記述

```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  myservo.write(outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("¥t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(100);
}
```

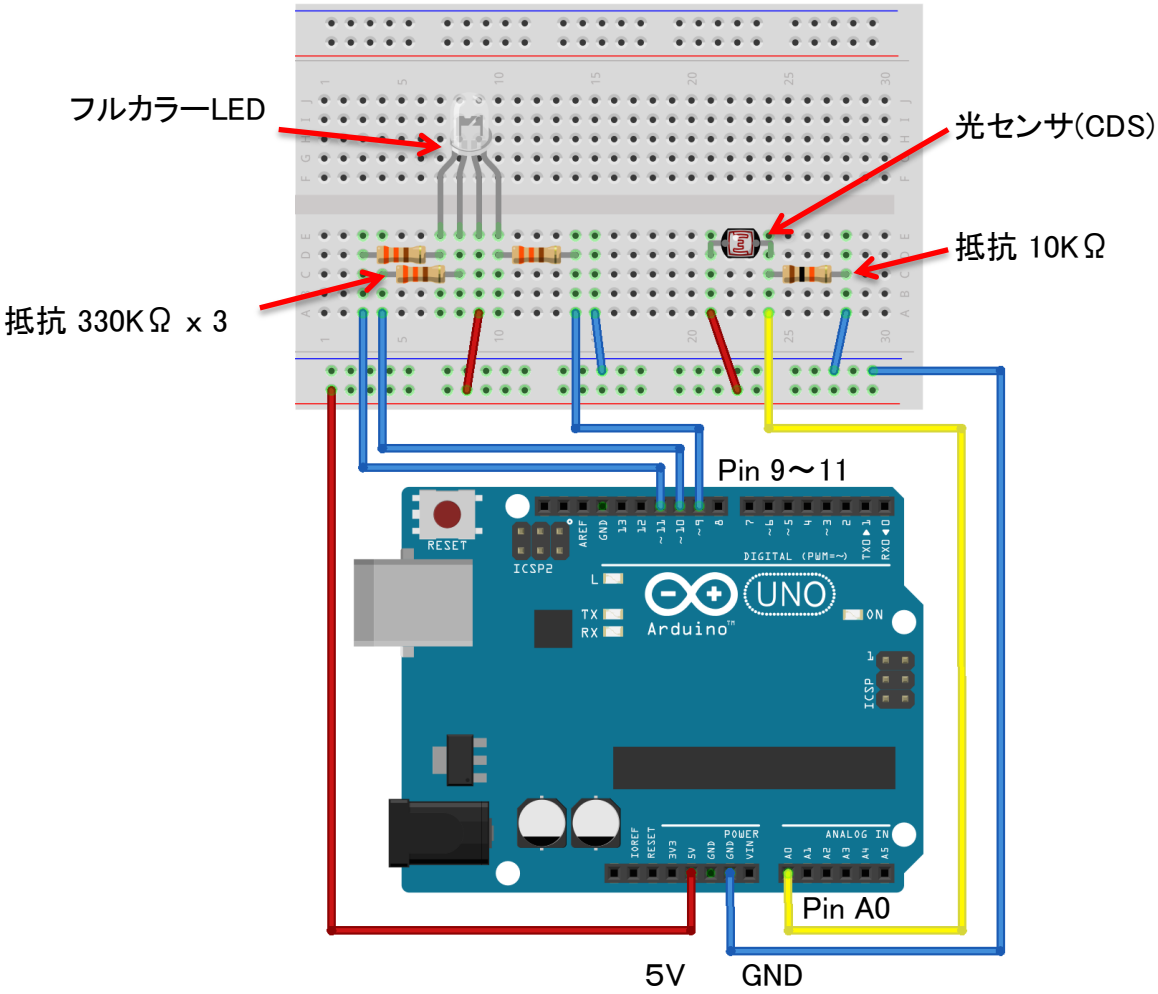
値の調整：150～700のsensorValueを0～180に割り振ってoutputValueへ

サーボの回転角度を指示

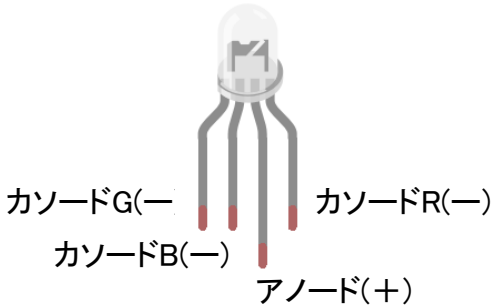
待機を100ミリ秒に増加

発展：フルカラーLEDで表示

光センサの入力量に応じてフルカラーLEDを発光させてみます。



フルカラーLED



光センサ(CDS)

※極性はありません

抵抗10KΩ

茶黒橙金

抵抗330Ω

橙橙茶金

フルカラーLED用にコード書き換え①

サンプル「AnalogInOutSerial」を読み込み以下の下線の箇所を書き換えます。

定数の宣言

```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
```

変数の宣言

```
const int analogOutPinR = 9; // Analog output pin that the LED is attached to
const int analogOutPinG = 11; // Analog output pin that the LED is attached to
const int analogOutPinB = 10; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
```

RGB三色分のピンを宣言する

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```

つづく

フルカラーLED用にコード書き換え②

つづき

メイン処理の記述

```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 150, 700, 0, 511);
  // change the analog out value:
   if(outputValue <= 255){
    analogWrite(analogOutPinR, 255);
    analogWrite(analogOutPinG, 255 - outputValue);
    analogWrite(analogOutPinB, outputValue);
   }
   else{
    outputValue = outputValue - 256;
    analogWrite(analogOutPinR, 255 - outputValue);
    analogWrite(analogOutPinG, outputValue);
    analogWrite(analogOutPinB, 255);
  }

 // print the results to the serial monitor:
   Serial.print("sensor = " );
   Serial.print(sensorValue);
   Serial.print("¥t output = ");
   Serial.println(outputValue);

   // wait 2 milliseconds before the next loop
   // for the analog-to-digital converter to settle
   // after the last reading:
   delay(2);
}
```

値の調整：150～700のsensorValueを0～511に割り振ってoutputValueへ

outputValueが以下かどうかで条件分岐

赤成分は消灯

緑成分は徐々に明るく

青成分は徐々に暗く

青～緑の色変化

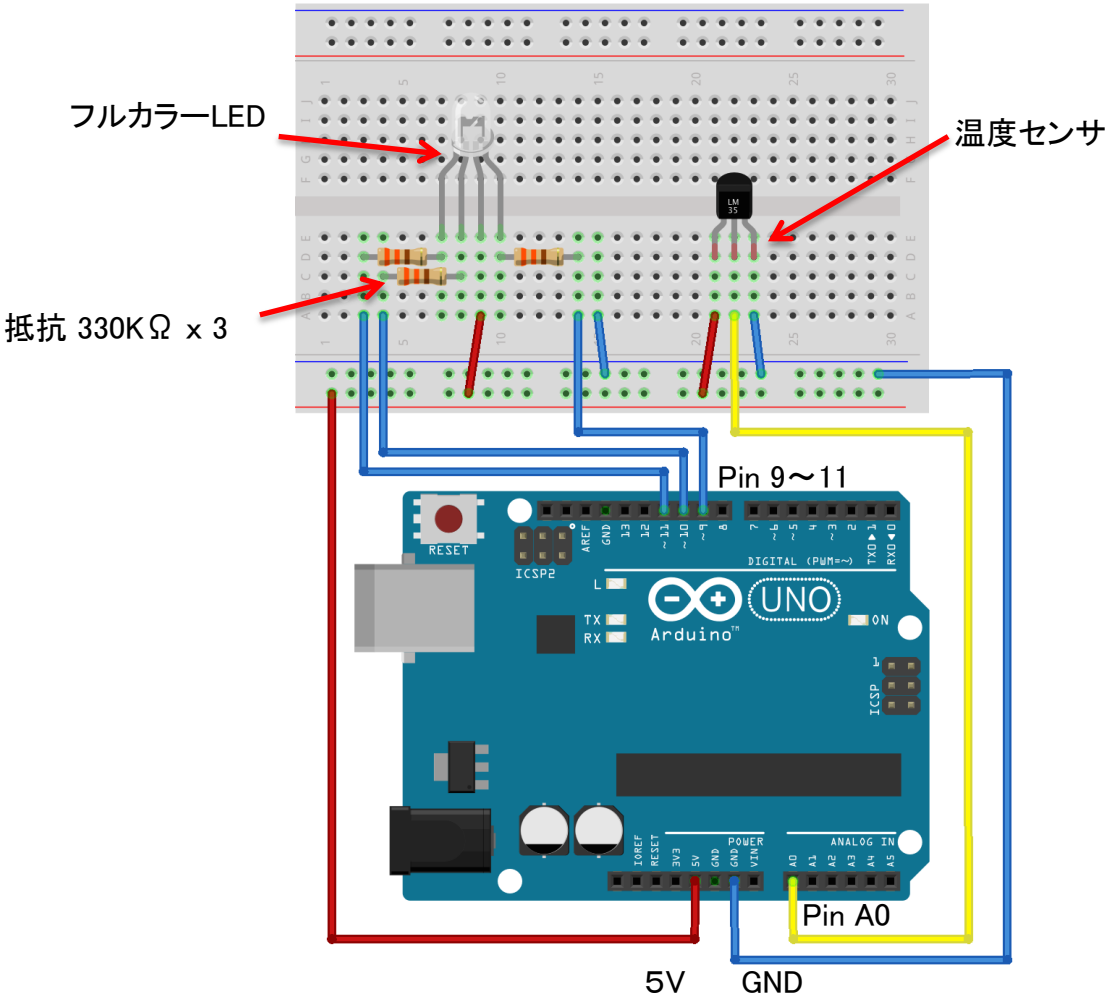
赤成分は徐々に明るく

緑成分は徐々に暗く

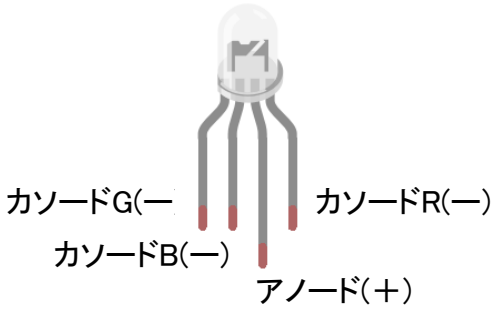
緑～赤の色変化

入力センサを変えてみよう

光センサ(CDS)を温度センサに変えてみましょう。



フルカラーLED



温度センサLM35

1　2　3

1:電源(5V)
2:信号
3:GND

抵抗330Ω

橙橙茶金

温度センサ用にコード書き換え①

温度センサ用にコードを一部書き換えます。

定数の宣言

```
// These constants won't change.  They're used to give names
// to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
```

変数の宣言

```
const int analogOutPinR = 9; // Analog output pin that the LED is attached to
const int analogOutPinG = 11; // Analog output pin that the LED is attached to
const int analogOutPinB = 10; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
```

※変更なし

初期化

```
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
```

つづく

温度センサ用にコード書き換え②

つづき

メイン処理の記述

```
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  sensorValue = sensorValue * 0.48;
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 25, 30, 0, 511);
  // change the analog out value:
   if(outputValue <= 255){
    analogWrite(analogOutPinR, 255);
    analogWrite(analogOutPinG, 255 - outputValue);
    analogWrite(analogOutPinB, outputValue);
   }
   else{
    outputValue = outputValue - 256;
    analogWrite(analogOutPinR, 255 - outputValue);
    analogWrite(analogOutPinG, outputValue);
    analogWrite(analogOutPinB, 255);
  }

 // print the results to the serial monitor:
   Serial.print("sensor = " );
   Serial.print(sensorValue);
   Serial.print("¥t output = ");
   Serial.println(outputValue);

   // wait 2 milliseconds before the next loop
   // for the analog-to-digital converter to settle
   // after the last reading:
   delay(2);
}
```

センサ読み取り値に係数0.48をかけて温度に変換

値の調整：25～30のsensorValueを0～511に割り振ってoutputValueへ