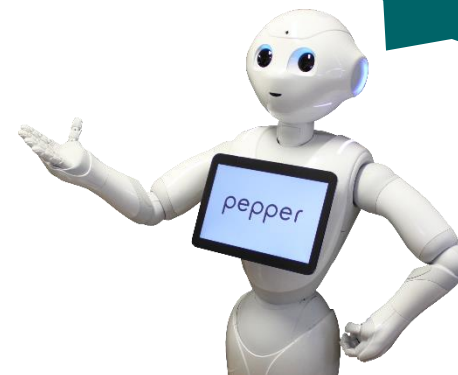
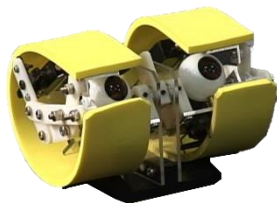


# ほぼなんでもつくれるようになるための 一日でわかるArduino入門

神奈川工科大学 創造工学部  
ホームエレクトロニクス開発学科  
山崎 洋一  
yamazaki@he.kanagawa-it.ac.jp

“つくりたいもの”を  
つくるための  
組み込み開発



# 3つの機能 例：ボールを蹴る

## <人の場合>



目で  
ボールをとらえ



頭で判断し



筋肉を動かす

知覚

センサ

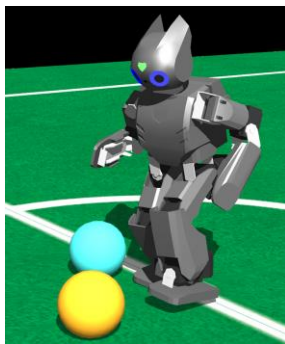
判断

コンピュータ

行動

アクチュエータ

## <ロボットの場合>



カメラで  
ボールをとらえ

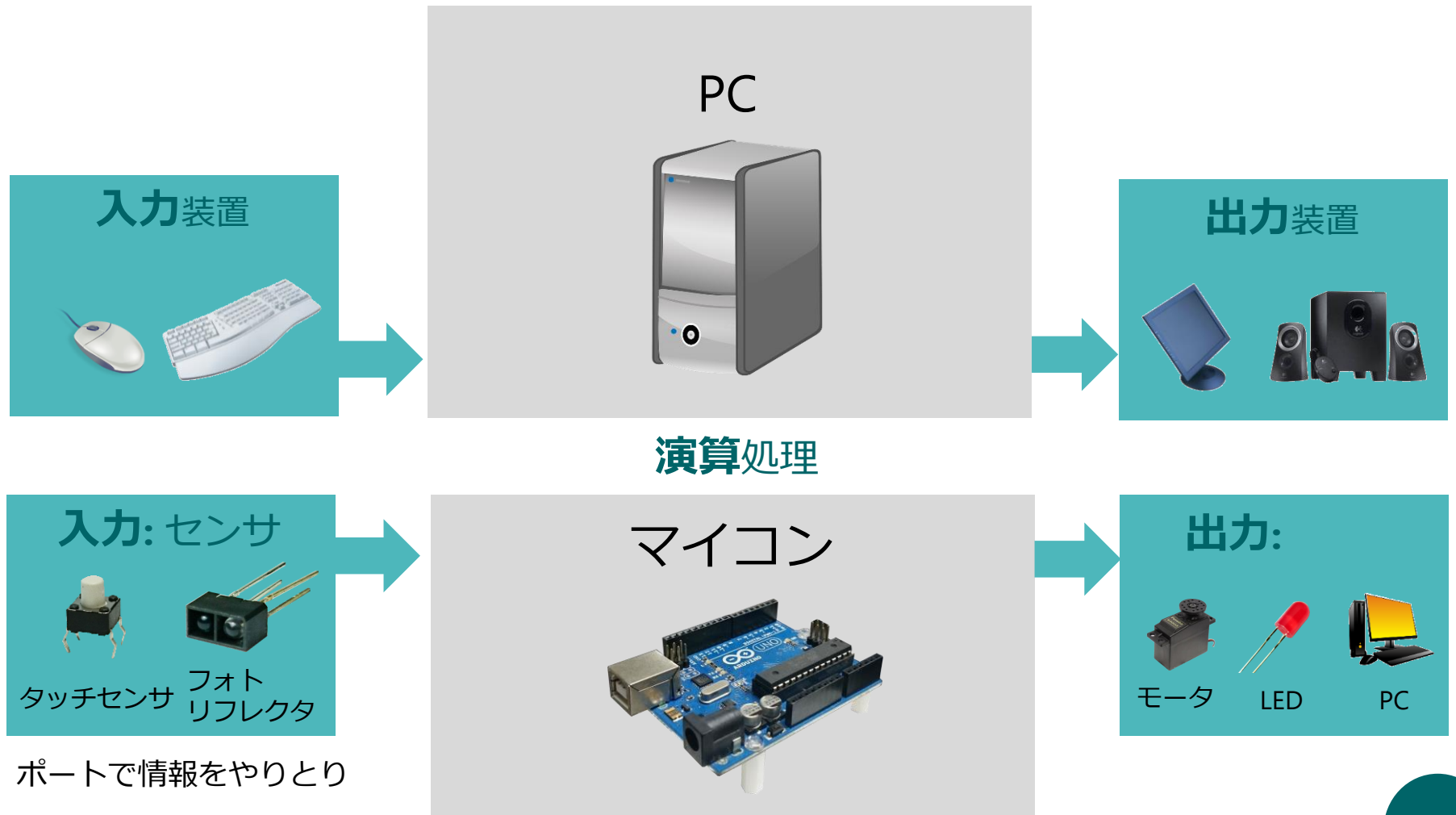


コンピュータで  
判断し



モータを  
動かす

# PCとマイコン



# 「マイコンができる」って何ができればいいの？ マイコンの必須4機能

## I/Oポート

- ・ LED点灯 , ON/OFFの検知・制御など

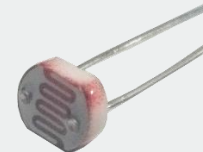
### 1. Lチカ



## A/D変換

- ・ センサ入力の取得

### 2. CdSセンサ値のAD変換



## PWM/タイマ

- ・ サーボモータ制御, 割り込み など

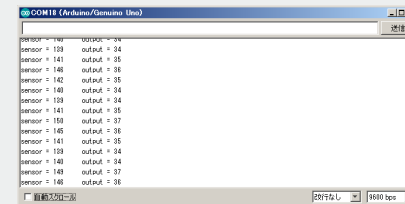
### 3. センサ値に応じた制御



## シリアル通信

- ・ PCとの通信など

### 4. PC上でのセンサ値の取得



# 「マイコンができる」って何ができればいいの？ マイコンの必須4機能

Turbo!

## I/Oポート

- ・ LED点灯 , ON/OFFの検知・制御など

### 1. Lチカ



## A/D変換

- ・ センサ入力の取得

### 2. フотリフレクタのAD変換



## PWM/タイマ

- ・ サーボモータ制御, 割り込み など

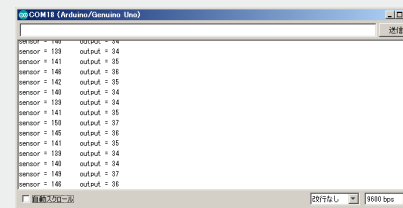
### 3. センサ値に応じた制御



## シリアル通信

- ・ PCとの通信など

### 4. PC上でのセンサ値の取得



# Arduinoとは？

- ・マイコンに通信モジュール，入出力ポートを備えてボードにしたもの
- ・C++ライクなArduino言語で開発できる統合開発環境がある
- ・Arduino Unoなど，多数のバージョンがある

Technical specs [Arduino Uno R3]



Atmel社 AVRマイコン  
ATmega328



Arduino LLC / Arduino SRL  
入出力ポート付きマイコンボード  
Arduino Uno R3

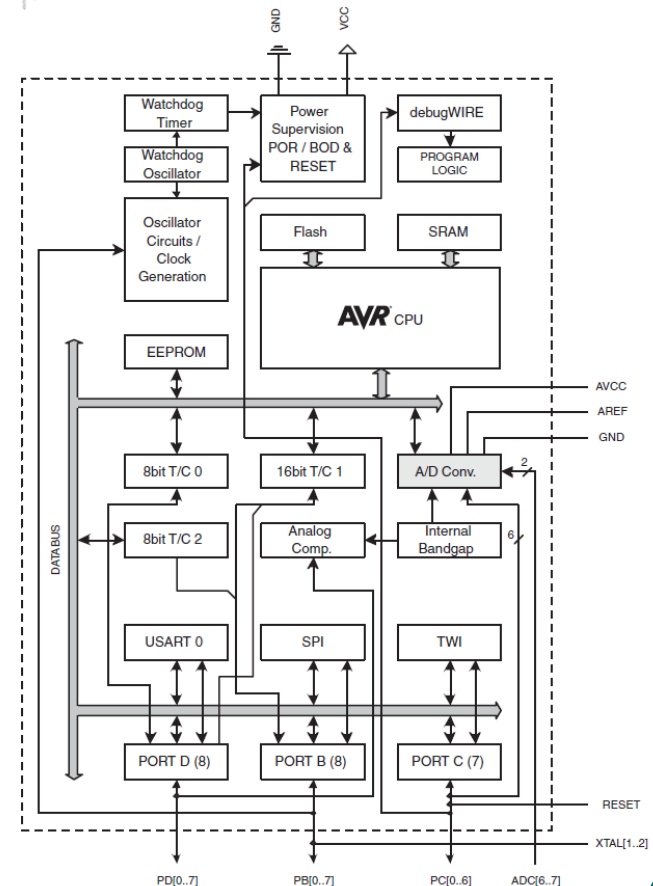
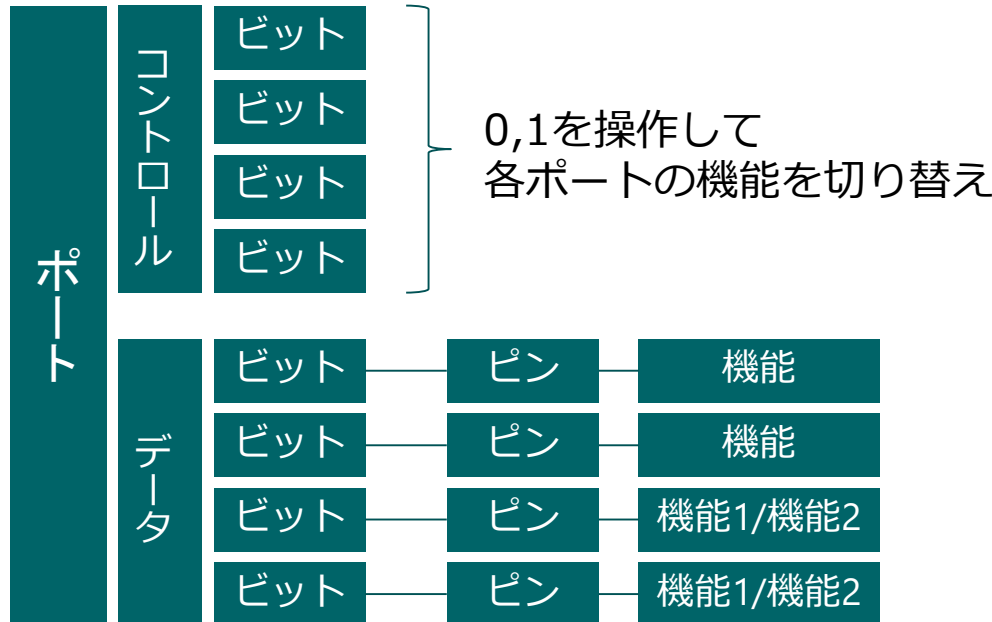


Microcontroller	<a href="#">ATmega328P</a>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

<https://www.arduino.cc/en/Main/ArduinoBoardUno> より

# AVR ATmega328

- マイコンの機能はポートに割り当てられる
- 各ポート, ピンは複数の機能を持つ



※Arduino言語では専用の関数が用意されているので  
ビットコントロールは不要

# Arduino Uno R3の主な機能

## 電源コネクタ

- ・ 外部から電源供給時に使用
- ・ 7V~12V推奨 (6~20V許容)

## 電源 (3.3V, 5V, GND)

- ・ センサ・モータ接続に使用

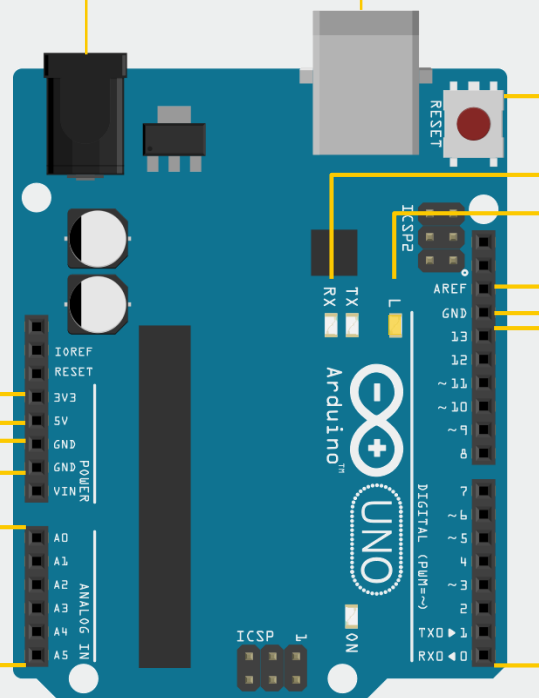
3.3V

5V

GND

## ANALOG入力ピン(A0-A5)

- ・ センサ入力等をAD変換



## USB Bコネクタ

(書き込み/PC通信/電力供給[5V])

## リセットスイッチ

通信確認用LED

動作確認用LED (13ピンと接続)

アナログ入力用基準電圧

GND

## DIGITALピン(0-13)

- ・ ~印はPMW機能あり (3, 6, 7, 9, 10, 11の6ピン)
- ・ 0, 1はシリアルに使用可

fritzing



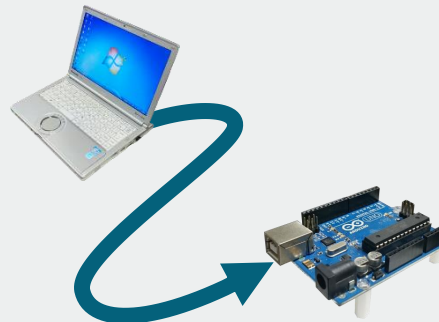
# Arduino開発の手順

## PC上でスケッチ作成

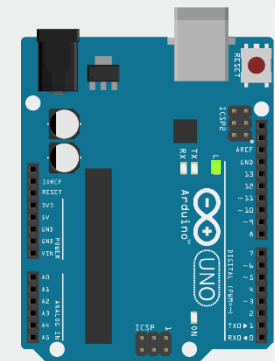


Arduino IDE

## Arduinoに書き込み



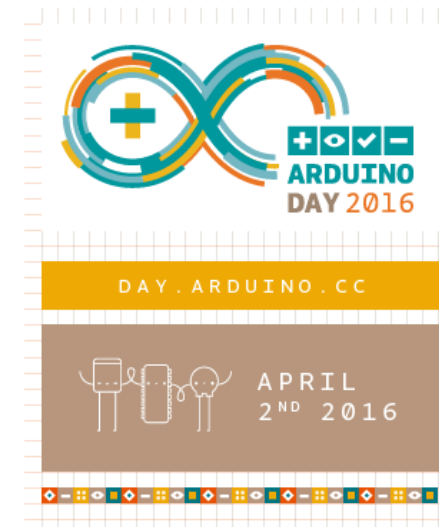
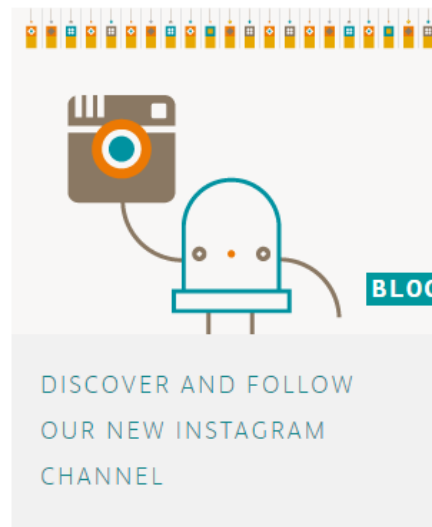
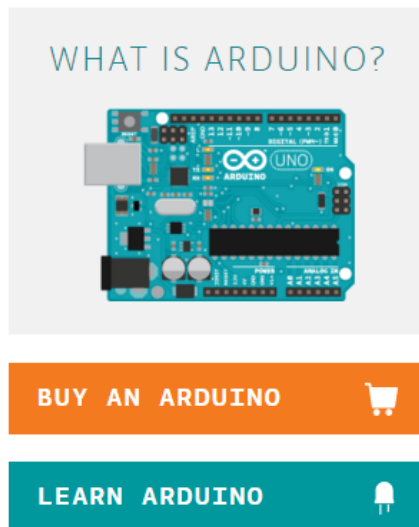
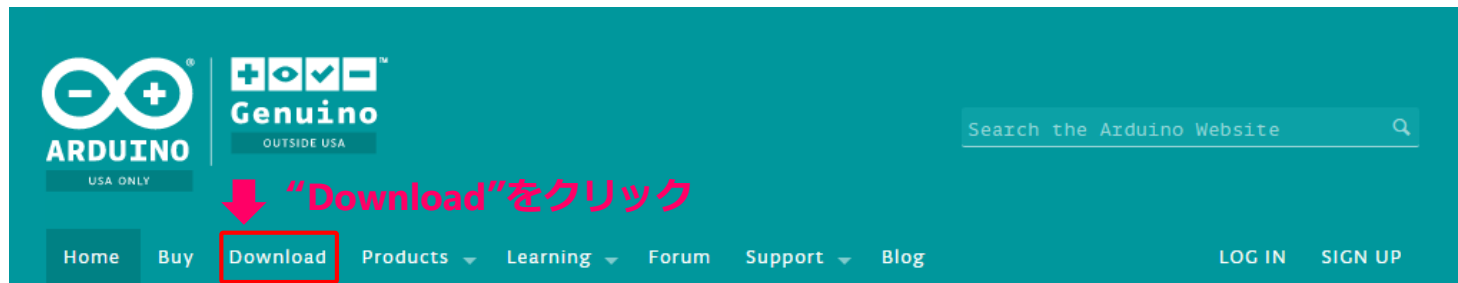
## 動作確認



fritzing

# Arduino IDE 開発環境のダウンロード①

- 公式サイト <https://www.arduino.cc/>



# Arduino開発環境のダウンロード②

- OSにあわせてWindows Installerをダウンロードしインストール

[下記はOSがWindowsの例]



Download the Arduino Software

**ARDUINO 1.6.8**  
 The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions.

**Windows Installer**  
 Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits  
 Linux 64 bits

Release Notes  
 Source Code  
 Checksums

ARDUINO SOFTWARE  
**HOURLY BUILDS**

Download a preview of the incoming release with the most updated features and bugfixes.

Windows  
 Mac OS X (Mac OS X Lion or later)  
 Linux 32 bit, Linux 64 bit

LAST UPDATE  
 31 March 2016 22:13:3 GMT

ARDUINO 1.0.6 / 1.5.x / 1.6.x  
**PREVIOUS RELEASES**

Download the previous version of the current release, the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version.

All the Arduino 00xx versions are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.



Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **6,778,993** TIMES. IMPRESSIVE! THIS IDE IS NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS. HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING IT TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEIT. YOU CAN HELP ACCELERATE THE DEVELOPMENT OF THE ARDUINO IDE BY CONTRIBUTING TOWARDS THE EFFORT OF MAKING IT BETTER.

\$3 \$5 \$10 \$25 \$50 OTHER

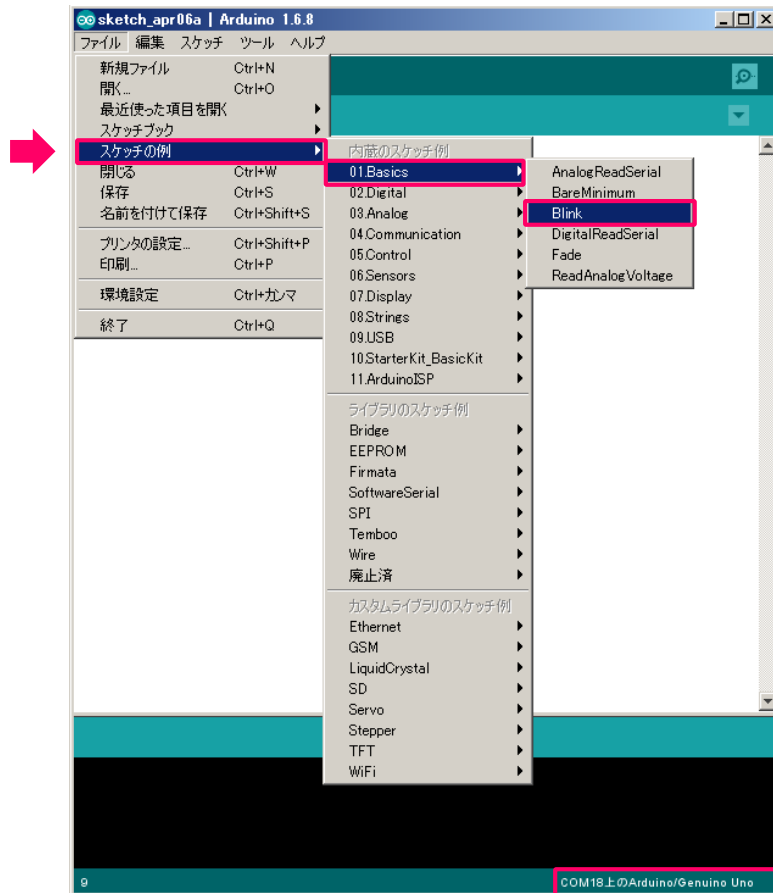
**クリック**

JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

# 1. Lチカ（LEDの点灯）

## ① Arduino IDEからスケッチを開く

### ②スケッチの作成（サンプルの利用）



### ①COMポート番号の確認（COM18）

#### STEP① COMポート番号の確認

- Arduinoを接続しArduino IDEを立ち上げ、右下のCOMポート番号を確認する

#### STEP② スケッチの作成（サンプルの利用）

- [ファイル]→[スケッチの例]にあるサンプルを利用する
- [ファイル]  
→[スケッチの例]  
→[01.Basics]  
→[Blink]  
を開く

# 1. Lチカ（LEDの点灯）

## ② サンプルスケッチの書込み

### ③書き込み



COMポート番号

### 各機能

- ✓ コンパイル
- ➡ ボードに書込む（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

# 1. Lチカ（LEDの点灯）

## ③ サンプルスケッチの動作



Blink

```

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
  
```

これだけわかればサンプルが読める！重要単語：

function	関数
initialize	初期化する
run	動作する, ~を実行する
voltage	電圧
second	秒

// リセット時またはボード電源を入れたときに、一度だけsetup関数が動作する

// デジタルピン13を出力として初期化

// loop関数は何度も繰り返し動作しつづける

// LEDをONにする(電圧レベルはHIGH)

// 1秒待つ

// 電圧をLOWにしてLEDをOFFにする

// 1秒待つ

# 1. Lチカ（LEDの点灯）

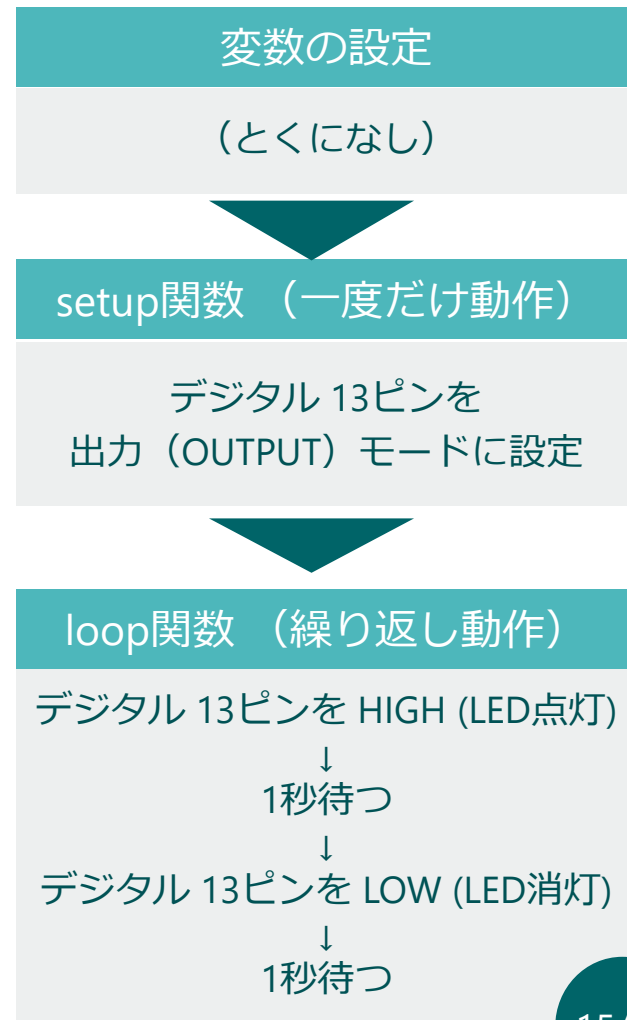
## ③ サンプルスケッチの動作



```

// リセット時またはボード電源を入れたときに、一度だけsetup関数が動作する
void setup() {
  // デジタルピン13を出力として初期化.
  pinMode(13, OUTPUT);
}

// loop関数は何度も繰り返し動作しつづける
void loop() {
  digitalWrite(13, HIGH); // LEDをONにする（電圧レベルはHIGH）
  delay(1000);           // 1秒待つ
  digitalWrite(13, LOW); // 電圧をLOWにしてLEDをOFFにする
  delay(1000);           // 1秒待つ
}
  
```

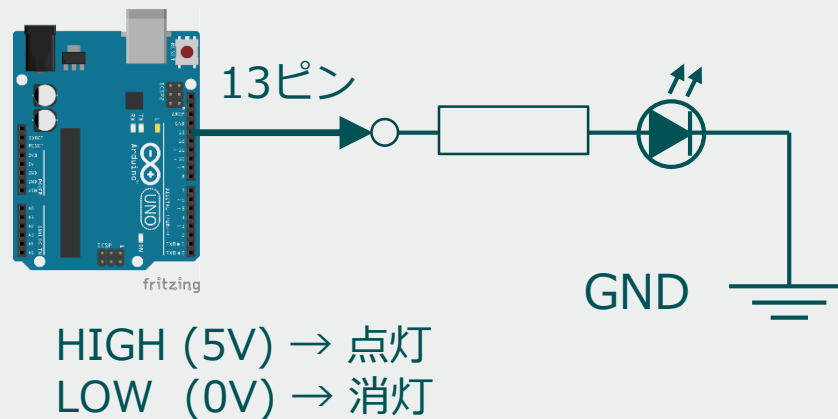


## 1. Lチカ (LEDの点灯)

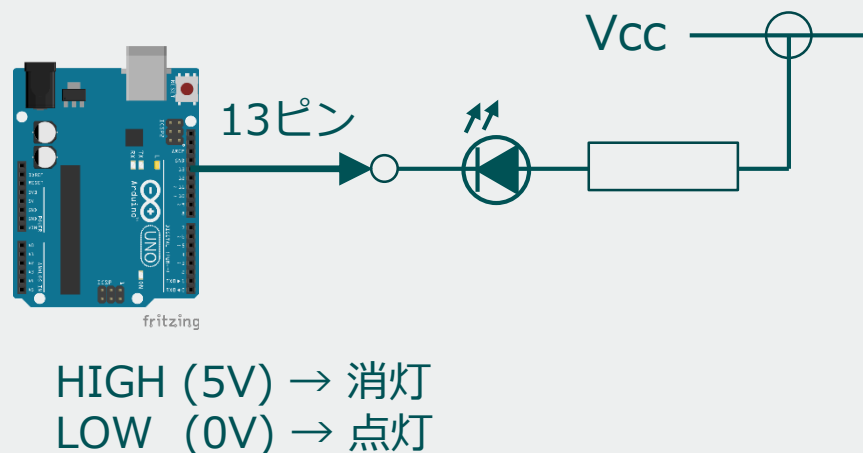
## ④ 吐き出しと吸い込み

- 各ポートの電流容量に合わせて選択
- 多数のLEDを制御するときなどは吸い込みを使用

## 吐き出し



## 吸い込み



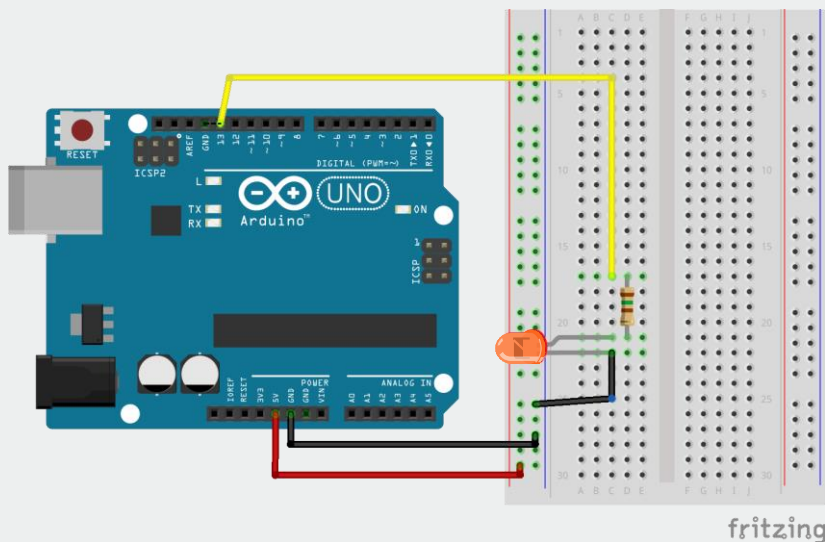


## 1. Lチカ (LEDの点灯)

## ④ 吐き出しと吸い込み (接続例)

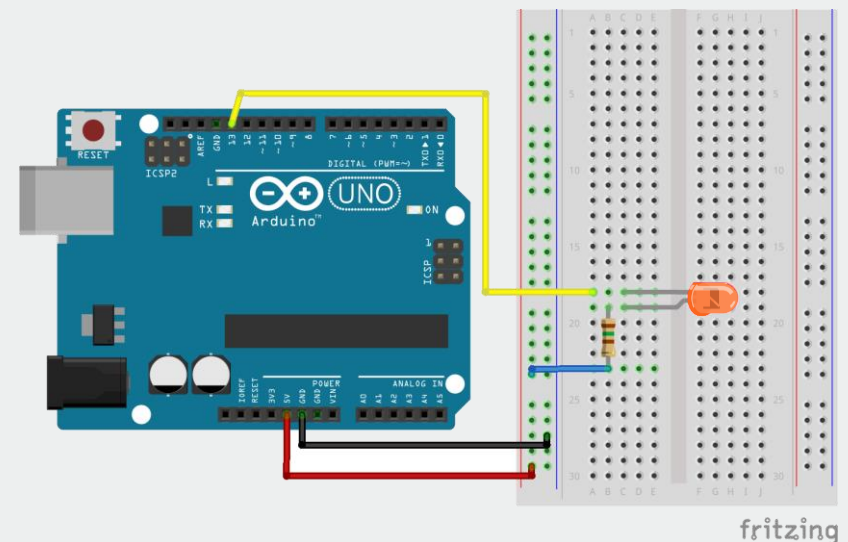
課題 1: Arduino Unoに搭載されている動作確認用LEDは13ピンに接続されている。このLEDの接続方式は吐き出しか吸い込みのどちらか。13ピンにそれぞれの方式でLEDを接続したものと比較して考えよ。

## 吐き出し接続の例



13ピンに接続  
 HIGH → 点灯  
 LOW → 消灯

## 吸い込み接続の例



13ピンに接続  
 HIGH → 消灯  
 LOW → 点灯

# 「マイコンができる」って何ができればいいの？ マイコンの必須4機能

Turbo!

## I/Oポート

- ・ LED点灯 , ON/OFFの検知・制御など

### 1. Lチカ



## A/D変換

- ・ センサ入力の取得

### 2. フォトリフレクタのAD変換



## PWM/タイマ

- ・ サーボモータ制御, 割り込み など

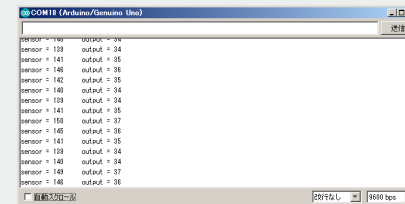
### 3. センサ値に応じた制御



## シリアル通信

- ・ PCとの通信など

### 4. PC上でのセンサ値の取得



#### STEP① サンプルスケッチを開く

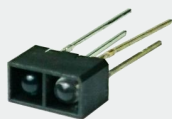
- [ファイル]  
→[スケッチの例]  
→[03.Analog]  
→[AnalogInOutSerial]  
を開く

#### STEP② 名前をつけて保存する

- [ファイル]  
→[名前を付けて保存]  
から適当名前をつけて保存する

#### スケッチの流れ

##### センサ入力



アナログ電圧  
[ 0~5V ]



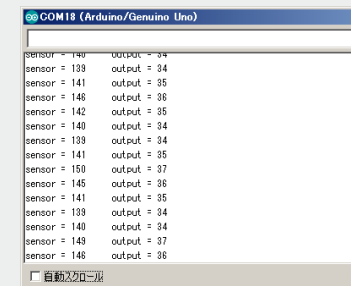
AD変換 (10bit)  
[ 0~1023 ]



シリアル通信  
(9600bps)



##### 出力① PC表示



##### 出力② LED/サーボモータ



PWM信号  
[ 0~255で指定 ]

## 赤外線センサ

①何を検知？ ⇒ 物体の有無・距離・色

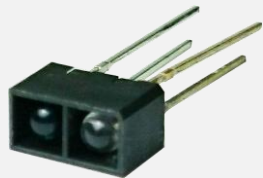
②どんなしくみ？

⇒ 発光部：赤外線LED (IR-LED)

受光部：フォトトランジスタ

フォトダイオード

例)



フォトリフレクタ



PSDセンサ

※PSD: Position Sensitive Device

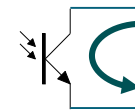
## フォトリフレクタのしくみ

赤外線LED

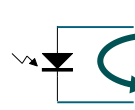


赤外線

フォト  
トランジスタ

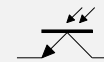


フォト  
ダイオード



LED

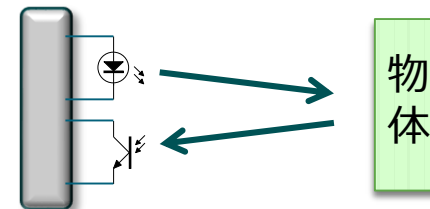
決まった方向に  
電流を流すと発光する



フォトトランジスタ  
赤外線光を受けると  
決まった方向に電流が流れる

発光素子と受光素子をペアで用いる

## 検知のしくみ



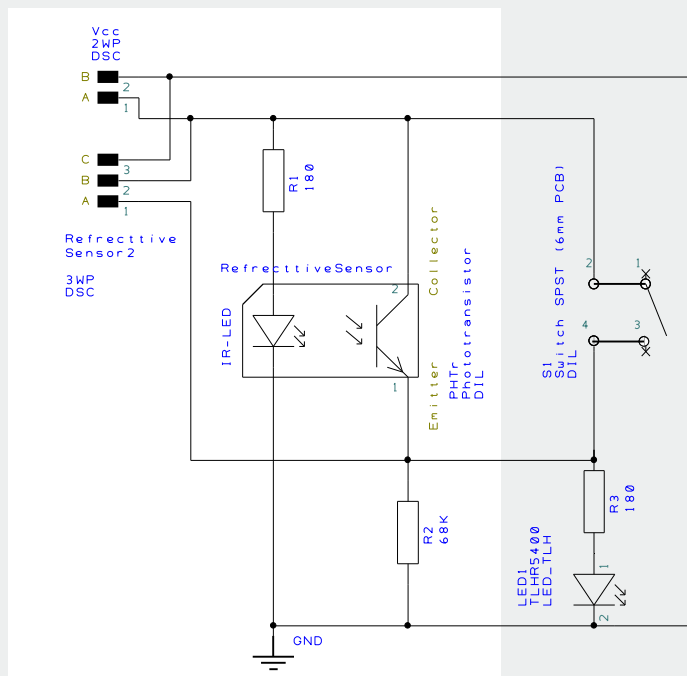
反射光の強さを利用して検知

## 2. PC上でのCdSセンサ値取得

### ③ フォトリフレクタの回路と接続例

課題 2：明るさを変化させセンサ値，出力値，LEDの明るさの関係を調査せよ

#### 参考回路例



#### 接続例





#### Blink

```
// ピンの設定:
const int analogInPin = A0; // A0ピンをアナログ入力
const int analogOutPin = 9; // 9ピンをアナログ出力 (LED)
int sensorValue = 0;        // センサ値用変数
int outputValue = 0;        // PWM (アナログ) 出力用変数

//セットアップ
void setup() {
  // シリアルポートをボーレート 9600 bpsで初期化:
  Serial.begin(9600);
}
```

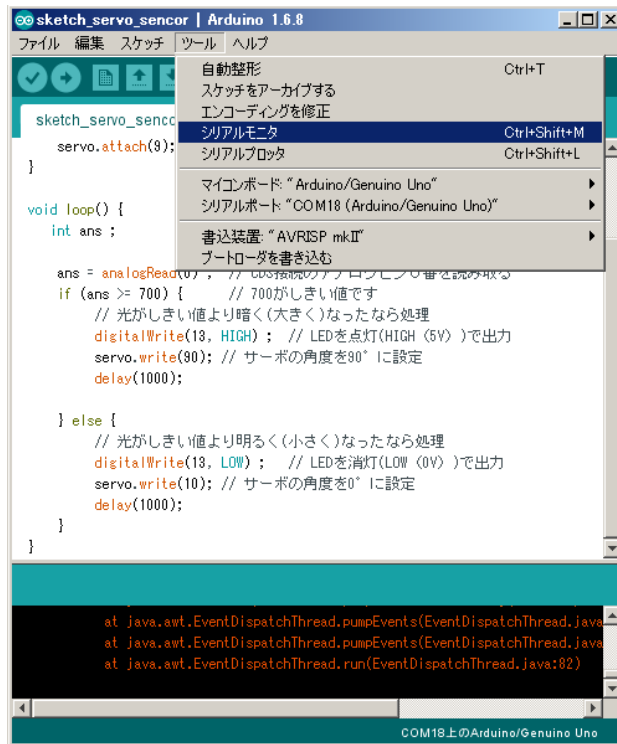
```
void loop() {
  // アナログセンサ値読込 (analogRead関数)
  sensorValue = analogRead(analogInPin);
  // 範囲を指定しマッピング ([0-1023]→[0-255])
  outputValue = map(sensorValue, 0, 1023, 0, 255);

  // アナログ電圧を出力 (analogWrite関数 ([0-255] を指定) )
  analogWrite(analogOutPin, outputValue);

  // シリアルモニタに数値を出力
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("¥t output = ");
  Serial.println(outputValue);

  // AD変換の処理のため2m秒待機
  delay(2);
}
```

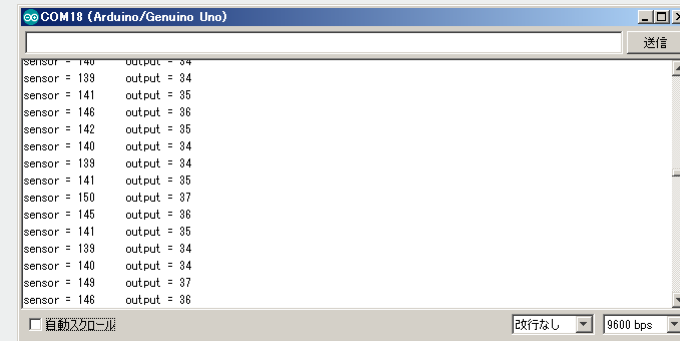
#### リアルタイムでデータを可視化するツール



注) シリアルプロッタでは、Serial.println関数で数値のみ(複数グラフの場合は“, ”で区切る)を送信している時にグラフを表示できる。試したい場合は文字列部分を消去するか、[スケッチの例] → [01.Basics] → [ReadAnalogVoltage]を試すとよい。

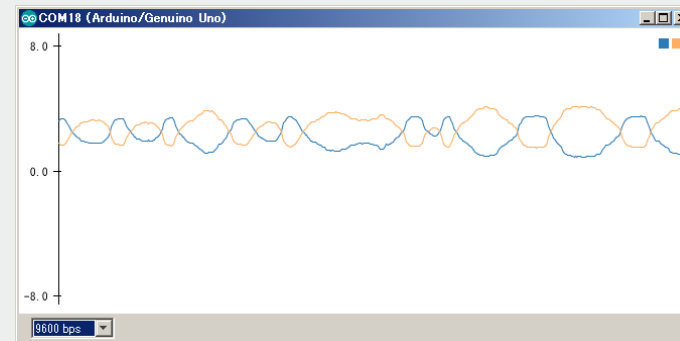
#### シリアルモニタ

- [ツール]→[シリアルモニタ]



#### シリアルプロッタ

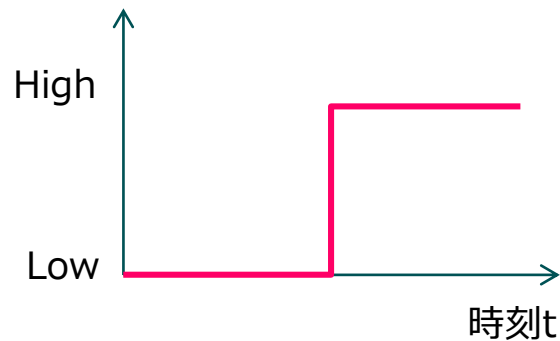
- [ツール]→[シリアルプロッタ]



### ⑥ PWM制御 (Pulse Width Modulation : パルス幅変調)

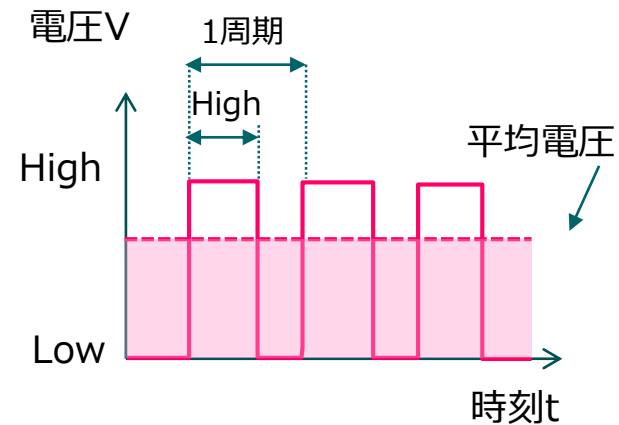
- 2値の電圧から連続値をつくるには？

電圧V



マイコンのポートは  
2値 (High, Low) の電圧しか  
出力できない

電圧V



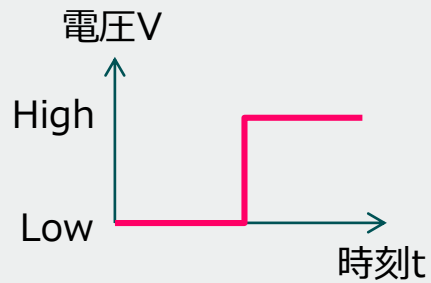
HighとLowを高速で切り替えると、平  
均電圧が出力されているように見える

一定周期のパルス列のHighとLowの幅を変えて見かけ上の電圧を変化させる



⑥ PWM制御 (Pulse Width Modulation : パルス幅変調)

2 値制御 (ON/OFF制御)

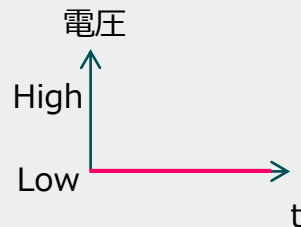
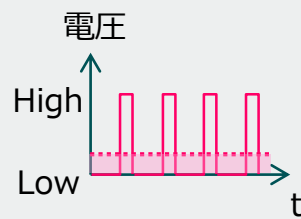
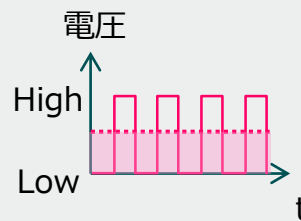
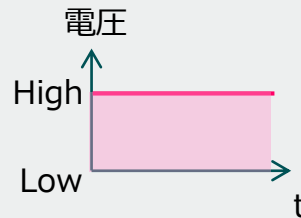


LEDの  
点灯・消灯



DCモータの  
ON・OFF

PWM制御



LEDの明るさ制御



サーボモータの位置制御



DCモータの速度制御

## CdSセル (フォトレジスタ)

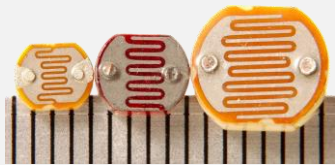
①何を検知? ⇒ 明るさ

②どんなしくみ?

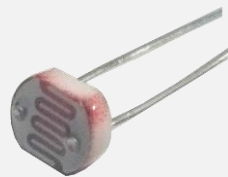
⇒ 光の強さに応じて抵抗値が低下する  
硫化カドミウム (CdS) セルを利用

- ・ 明所では抵抗値が低い
- ・ 暗所では抵抗値が高い

例)

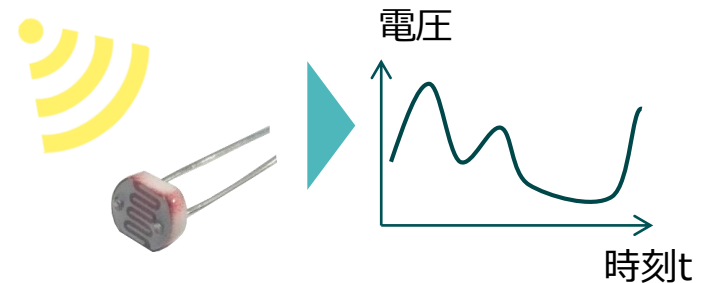


さまざまなCdSセル



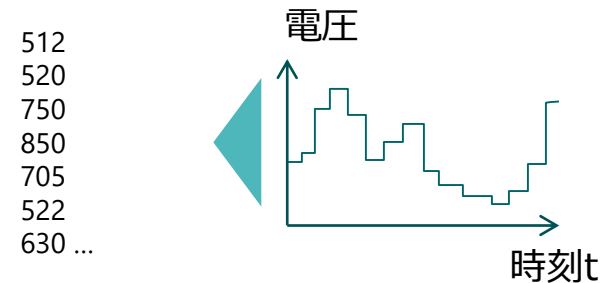
CdSセル GL5516

## CdSセルのセンサ利用 (AD変換)



明るさによってアナログ電圧値が変化

## アナログ入力ポートでAD変換 (10bit)



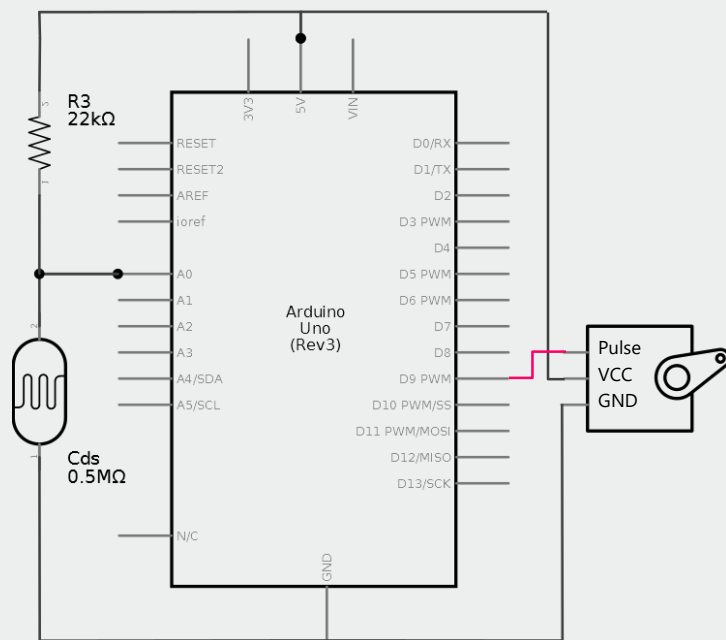
一定時間ごとのデジタルデータ (数値) として出力

## 2. PC上でのCdSセンサ値取得

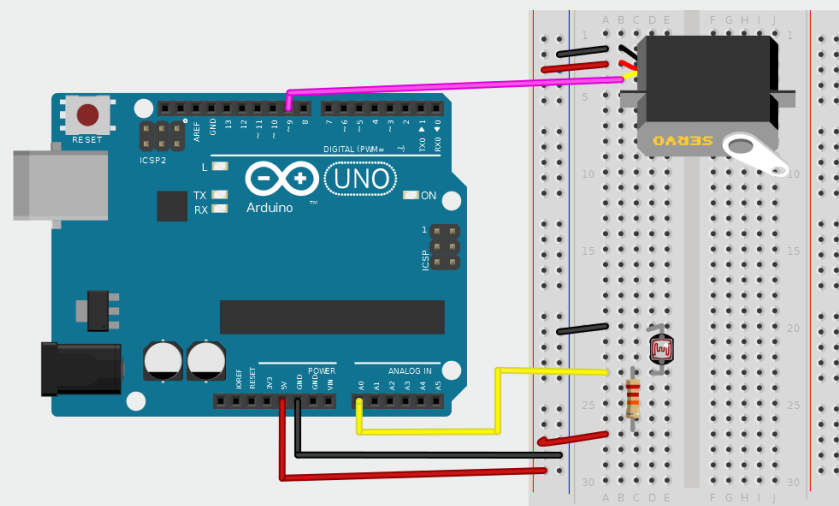
### ⑦ CdSセル(フォトレジスタ)の回路と接続例② サーボモータ

課題3: 明るさを変化させセンサ値, 出力値, サーボの位置の関係を調査せよ

#### 参考回路例



#### 接続例



### 3. 自動照明の製作

課題4: 暗くなると点灯し, 明るくなると消える自動照明を製作せよ。

下記の2通りを考えること。

- ① ある一定の暗さになると, LEDが点灯する照明
- ② 暗さに応じて, しだいにLEDが明るくなる照明

### 3. 自動照明の製作

## ヒント：サンプルスケッチ AnalogInOutSerial



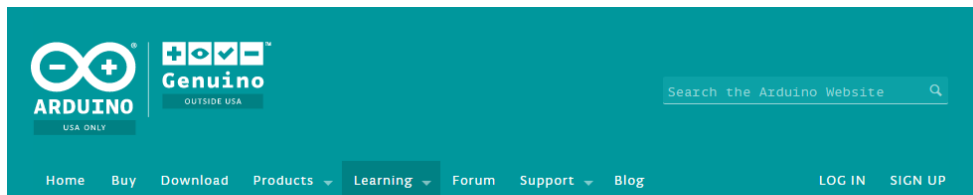
#### Blink

```
// ピンの設定:
const int analogInPin = A0; // A0ピンをアナログ入力
const int analogOutPin = 9; // 9ピンをアナログ出力 (LED)
int sensorValue = 0; // センサ値用変数
int outputValue = 0; // PWM (アナログ) 出力用変数

//セットアップ
void setup() {
  // シリアルポートをビットレート 9600 bpsで初期化:
  Serial.begin(9600);
}
```

```
void loop() {
  // アナログセンサ値読込:
  sensorValue = analogRead(analogInPin);
  // 範囲を指定しマッピング:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // マッピングした値を出力: ここがLEDのPWM制御部
  analogWrite(analogOutPin, outputValue); ←
  // print the results to the serial monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  // AD変換の処理のため2m秒待機
  delay(2);
}
```

## 英語マニュアルの攻略の仕方① 少しずつ読んでみよう



Arduinoのサンプルプログラムは  
公式サイトで解説されている

TUTORIALS

## Tutorials

**BUILT-IN EXAMPLES**

Built-in Examples are sketches included in the Arduino Software (IDE), to open them click on the toolbar menu: File > Examples. These simple programs demonstrate all basic Arduino commands. They span from a Sketch Bare Minimum to Digital and Analog I/O, to the use of Sensors and Displays.

**EXAMPLES FROM LIBRARIES**

The Arduino Software (IDE) can be extended through the use of libraries, just like most programming platforms, to provide extra functionality to your sketches. These tutorials walk you through the Examples of a number of libraries that come installed with the IDE, to open them click on the toolbar menu: File > Examples.

**TUTORIALS ON ARDUINO PROJECT HUB**

Arduino Project Hub is our official tutorial platform powered by hackster.io. Get inspired by a variety of tutorials, getting started guides, showcases and pro tips. Contribute projects and ideas, comment on the tutorials you are curious about, and 'Respect' the ones you like the most.

## Tutorials

<https://www.arduino.cc/en/Tutorial/HomePage>

[Learning] → [Built-In Examples]

※回路もあわせて紹介されており独学できるようになっています。一般に技術文書はシンプルな英語で書かれています。この資料のサンプルプログラムのコメントの英語を理解するだけでも、読める部分が多くあります。技術英語のトレーニングを兼ねて気長にチャレンジしてください！

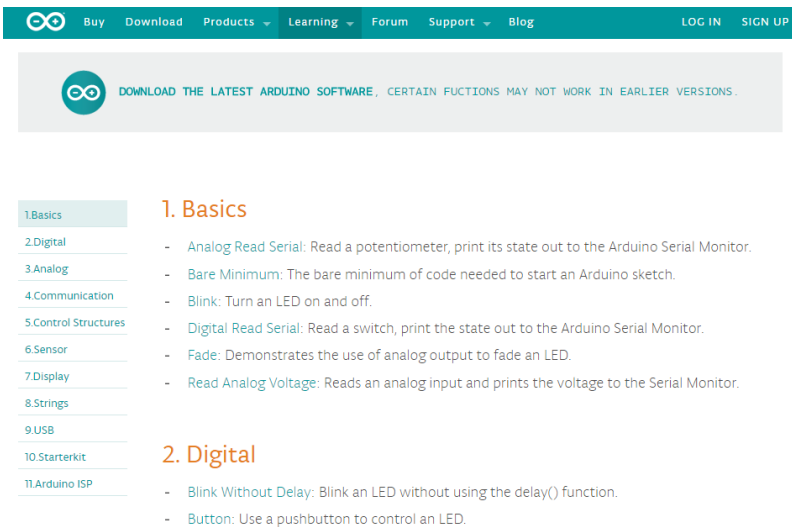
## 英語マニュアルの攻略の仕方② Chromeの翻訳機能

英語を読むのが難しいときは...

ウェブブラウザChromeの翻訳機能を試してみる

(日本語が不自然なこともありますが、単語を理解する助けになる)

“”をクリック



Buy Download Products Learning Forum Support Blog LOG IN SIGN UP

DOWNLOAD THE LATEST ARDUINO SOFTWARE. CERTAIN FUCTIONS MAY NOT WORK IN EARLIER VERSIONS.

- 1. Basics
- 2. Digital
- 3. Analog
- 4. Communication
- 5. Control Structures
- 6. Sensor
- 7. Display
- 8. Strings
- 9. USB
- 10. Starterkit
- 11. Arduino ISP

### 1. Basics

- Analog Read Serial: Read a potentiometer, print its state out to the Arduino Serial Monitor.
- Bare Minimum: The bare minimum of code needed to start an Arduino sketch.
- Blink: Turn an LED on and off.
- Digital Read Serial: Read a switch, print the state out to the Arduino Serial Monitor.
- Fade: Demonstrates the use of analog output to fade an LED.
- Read Analog Voltage: Reads an analog input and prints the voltage to the Serial Monitor.

### 2. Digital

- Blink Without Delay: Blink an LED without using the delay() function.
- Button: Use a pushbutton to control an LED.



購入 ダウンロード プロダクト 学習 フォーラム サポート ブログ ログイン サインアップ

最新のARDUINOのソフトウェアをダウンロードして、特定の面数は、以前のバージョンでは動作しない場合があります。

- 1. Basics
- 2. Digital
- 3. Analog
- 4. Communication
- 5. Control構造
- 6. Sensor
- 7. Display
- 8. Strings
- 9. USB
- 10. Starterkit
- 11. Arduino ISP

### 1. 基本

- アナログ・リードのシリアル: Arduinoのシリアルモニターに出、その状態を印刷し、ポテンショメータをお読みください。
- 最低限: Arduinoのスケッチを開始するために必要なコードの最低限。
- 点滅: オンのLEDと電源を切ります。
- シリアル・デジタル・リード: スイッチを読むArduinoのシリアルモニターへのアウト状態の一覧を表示します。
- フェード: LEDをフェードするアナログ出力の使用方法を示します。
- アナログ電圧を読む: アナログ入力を読み取り、シリアルモニターに電圧を出力します。

### 2. デジタル

- 遅滞なく点滅: デレイ()関数を使用せずにLEDを点滅させます。
- ボタン: LEDを制御するためにプッシュボタンを使用します。

## 英語マニュアルの攻略の仕方③ 日本語マニュアルを探す

それでも英語を読むのが難しいときは...

日本語のサイトを探す



Google

arduino サンプルプログラム

すべて 動画 画像 ショッピング ニュース もっと見る ▼ 検索ツール

約 67,700 件 (0.36 秒)

**Arduino サンプルプログラム解説 - GarretLab - FC2**  
[garretlab.web.fc2.com/arduino/examples/](http://garretlab.web.fc2.com/arduino/examples/) ▼  
Arduinoソフトウェア(arduino 1.0.3)についてくる**サンプルプログラム**の解説です。もとのプログラムにもコメントが詳しく書いてありますが、それにはあまりこだわらずに書いていきます。できる限りプログラミングの初心者の方にもわかるようにしていくつもりです。  
[AnalogReadSerial - Debounce - 02.Digital - DigitalInputPullup](#)

**Arduinoを用意して簡単なプログラミングをはじめよう | Think IT ...**  
<https://thinkit.co.jp/story/2013/02/12/3960> ▼  
2013/02/12 - 前回はミニ四駆の紹介と、FULLERで作製したミニ四駆の紹介で終わりましたが、今回は実際にモノを動かしていきます。まずは、**Arduino**に触れるところからはじめます。**Arduino**キットや工具を準備しようまずは今回必要となる**Arduino**の ...





# 英語マニュアルの攻略の仕方③ 日本語マニュアルを探す



## Arduinoで遊ぶページ

Arduinoで遊んだ結果を残すページです。

<a href="#">トップ</a>	<a href="#">Arduino</a>	<a href="#">Arduinoリファレンス</a>	<a href="#">Arduinoガイド</a>	<a href="#">Intel Edison</a>	<a href="#">連絡先</a>	<a href="#">Amazon Store</a>
---------------------	-------------------------	-------------------------------	----------------------------	------------------------------	---------------------	------------------------------

Arduino

Arduino 概要

製作

実験室

サンプルプログラム解説

01. Basics

02. Digital

03. Analog

04. Communication

05. Control

06. Sensors

07. Display

08. Strings

09. USB

10. StarterKit

逆引き Arduino

参考サイト

内部構造

C/C++言語

### サンプルプログラム 解説

[トップ](#) > [Arduino](#) > サンプルプログラム解説

Arduinoソフトウェア(arduino 1.0.3)についてくるサンプルプログラムの解説です。もとのプログラムにもコメントが詳しく書いてありますが、それにはあまりこだわらずに書いていきます。できる限りプログラミングの初心者の方にもわかるようにしていくつもりです。

サンプルプログラムは、Arduinoソフトウェアを起動して、File > Examples(ファイル > スケッチの例) からソースを選択します。

<b>01. Basics</b>						
<a href="#">AnalogReadSerial</a>	<a href="#">BareMinimum</a>	<a href="#">Blink</a>	<a href="#">DigitalReadSerial</a>	<a href="#">Fade</a>	<a href="#">ReadAnalogVoltage</a>	
<b>02. Digital</b>						
<a href="#">BlinkWithoutDelay</a>	<a href="#">Button</a>	<a href="#">Debounce</a>	<a href="#">DigitalInputPullup</a>	<a href="#">StateChangeDetection</a>	<a href="#">toneKeyboard</a>	<a href="#">toneMelody</a>
<a href="#">toneMultiple</a>	<a href="#">tonePitchFollower</a>					
<b>03. Analog</b>						
<a href="#">AnalogInOutSerial</a>	<a href="#">AnalogInput</a>	<a href="#">AnalogWriteMega</a>	<a href="#">Calibration</a>	<a href="#">Fading</a>	<a href="#">Smoothing</a>	
<b>04. Communication</b>						
<a href="#">ASCIITable</a>	<a href="#">Dimmer</a>	<a href="#">Graph</a>	<a href="#">MIDI</a>	<a href="#">MultiSerialMega</a>	<a href="#">PhysicalPixel</a>	<a href="#">ReadASCIIStrng</a>
<a href="#">SerialCallResponse</a>	<a href="#">SerialCallResponseASCII</a>	<a href="#">VirtualColorMixer</a>				
<b>05. Control</b>						
<a href="#">Arrays</a>	<a href="#">ForLoopIteration</a>	<a href="#">IfStatementConditional</a>	<a href="#">switchCase</a>	<a href="#">switchCase2</a>	<a href="#">WhileStatementConditional</a>	
<b>06. Sensors</b>						
<a href="#">ADXL3xx</a>	<a href="#">Knock</a>	<a href="#">Memsic2125</a>	<a href="#">Ping</a>			
<b>07. Display</b>						
<a href="#">barGraph</a>	<a href="#">RowColumnScanning</a>					
<b>08. Strings</b>						
<a href="#">CharacterAnalysis</a>	<a href="#">StringAdditionOperator</a>	<a href="#">StringAppendOperator</a>	<a href="#">StringCaseChanges</a>	<a href="#">StringCharacters</a>	<a href="#">StringComparisonOperators</a>	<a href="#">StringConstructors</a>
<a href="#">StringIndexOf</a>	<a href="#">StringLength</a>	<a href="#">StringLengthTrim</a>	<a href="#">StringReplace</a>	<a href="#">StringStartsWithEndsWith</a>	<a href="#">StringSubstring</a>	<a href="#">StringToInt</a>
<b>09. USB</b>						
<a href="#">KeyboardLogout</a>	<a href="#">KeyboardMessage</a>	<a href="#">KeyboardReprogram</a>	<a href="#">KeyboardSerial</a>	<a href="#">KeyboardAndMouseControl</a>	<a href="#">ButtonMouseControl</a>	<a href="#">JoystickMouseControl</a>
<b>10. StarterKit</b>						
<a href="#">SpaceshipInterface</a>	<a href="#">LoveOMeter</a>	<a href="#">ColorMixingLamp</a>	<a href="#">ServoMoodIndicator</a>	<a href="#">LightTheremin</a>	<a href="#">Keyboard</a>	<a href="#">DigitalHourglass</a>
<a href="#">MotorizedPinwheel</a>	<a href="#">MotorizedPinwheel</a>	<a href="#">CrystalBall</a>	<a href="#">KnockLock</a>	<a href="#">TouchSensorLamp</a>	<a href="#">TweakTheArduinoLogo</a>	<a href="#">HackingButtons</a>

http://garretlab.web.fc2.com/arduino/examples/

## 電子回路の資料をつくるツール Fritzing

- 本資料の回路, ブレッドボード図はFritzingで作成している
- 電子回路設計ツール Fritzing <http://fritzing.org/>
- オープンソース, 製作した画像はCC-BY-SAライセンス

