

徳島大学 大学解放実践センター 公開講座

センサのしくみを知ろう（基礎編）

徳島大学大学院社会産業理工学研究部総合技術センター  
徳島大学理工学部理工学科情報光システムコース  
技術専門職員 辻 明典

（桑折 範彦：徳島大学名誉教授）.

## センサのしくみを知ろう（基礎編）2

### 5月27日 温度をはかる

#### 前回の復讐

##### 1. arduinoNano の概要の復習

arduino の設定を前回行った。> ドライバー、ライブラリーのインストール

arduino1.8.2 をインストール（まだ、ボードはつながない）

USB コードをパソコンと arduino ボードを接続する

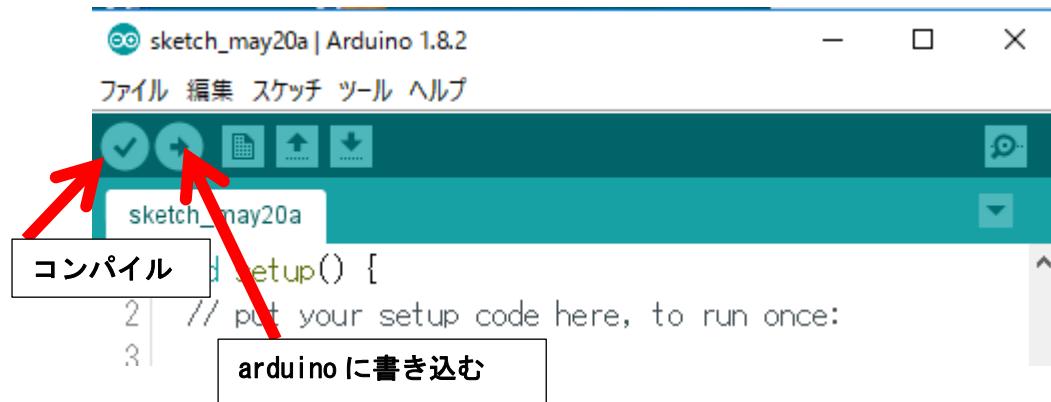
##### 2. 環境設定

注意点：arduino の型、シリアルポート番号など arduino1.8.2 を利用する

実行方法：スケッチ（プログラム）を読み込む、コンパイルする、arduino に書き込む（すぐに実行される）

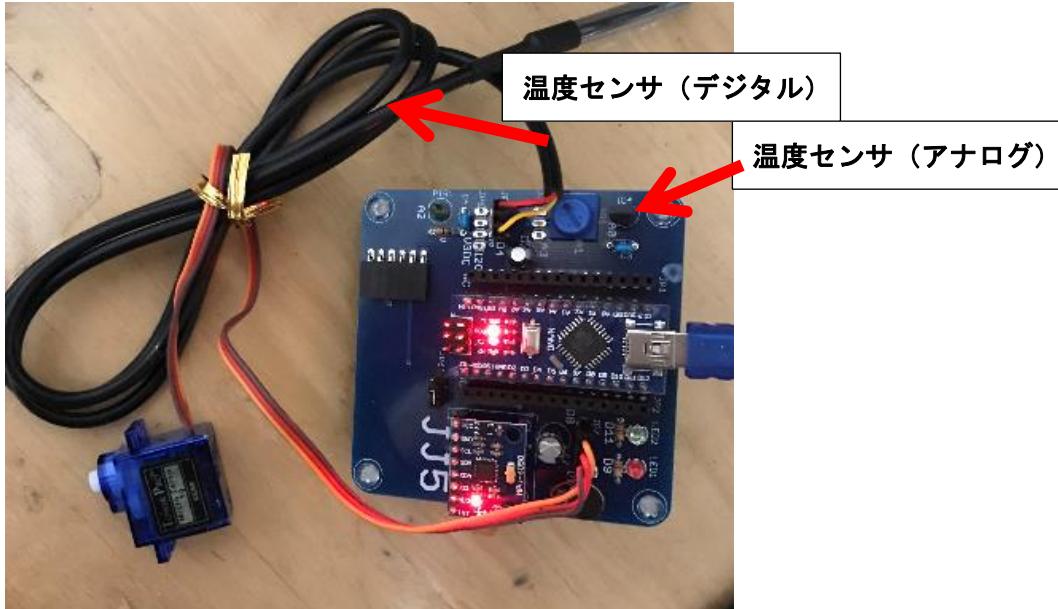
##### 3. テスト：簡単なスケッチの形式と解読 Example101

## ブレッドボードの使い方 LED の点滅をテストする Example 例題をやってみる。



## 本日の予定

1) 温度センサ（アナログ）LM61CIZ を利用 写真を参考にしてください。



。

2) 温度をはかるセンサ（デジタル）として DS18B20 を 1 線通信方式で利用  
それぞれを用いて室温を計測してみよう。どんな違いがあるか？

## 1) 温度センサ（アナログ）LM61CIZ

測定範囲 ; -30°C～100°C

次のスケッチ（プログラムのこと）をテストしてみよう。説明をする。

ライブラリの中から TEMP を読み込む 「温度センサ LM61CIZ の使い方」 参考

[www.geocities.jp/zattouka/GarageHouse/micon/Arduino/Temp/...](http://www.geocities.jp/zattouka/GarageHouse/micon/Arduino/Temp/)

```
// TEMP LM61CIZ
void setup() {
    Serial.begin(9600); // Arduino とシリアル通信の準備
}
void loop() {
    int ans, temp, tv; //変数の定義
    ans = analogRead(0); //アナログ 0 番からセンサ値を読み込む
    tv = map(ans, 0, 1023, 0, 5000); //センサ値 ans を電圧に変換する。元も範囲 0～1023 から新しい範囲 0 ～ 5000 へ
    temp = map(tv, 300, 1600, -30, 100); //電圧 tv を温度 temp (範囲-30, 100) に変換する。
    Serial.println(temp); //temp を arduinoIDE に送る。シリアルモニタ、シリアルプロッタ利用
    delay(1000); // 1000ms(1sec) 待って、繰り返す。
}
```

map 関数は 0～1023 の ans を 0～5000 の値に変換する。tv が 1023 なら 5000 (mV) となる

$tv = ans * (5000 / 1023)$

また、temp (温度) は tv が 300 なら  $-30^{\circ}\text{C}$  で 1600 なら  $100^{\circ}\text{C}$  と変換

される。線の勾配は  $(130^{\circ}\text{C}/(1600-300)) = (1/10)$

$\text{temp} = (1/10) * (\text{tv} - 300) - 30$  の式で表される。(右図参照)

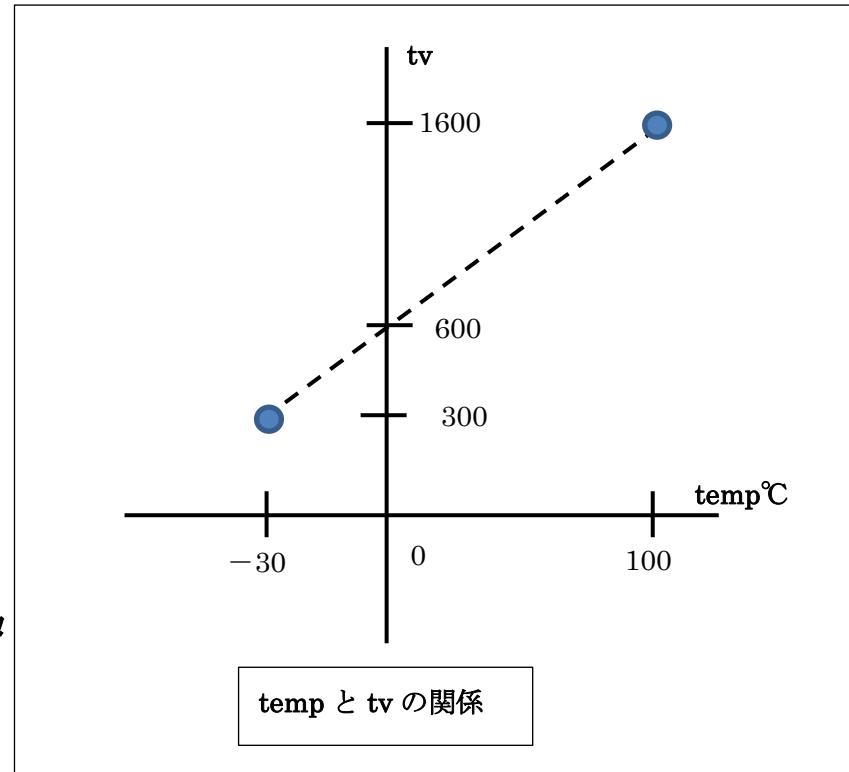
実際に室温を測定して温度計とどの程度違いがあるか？

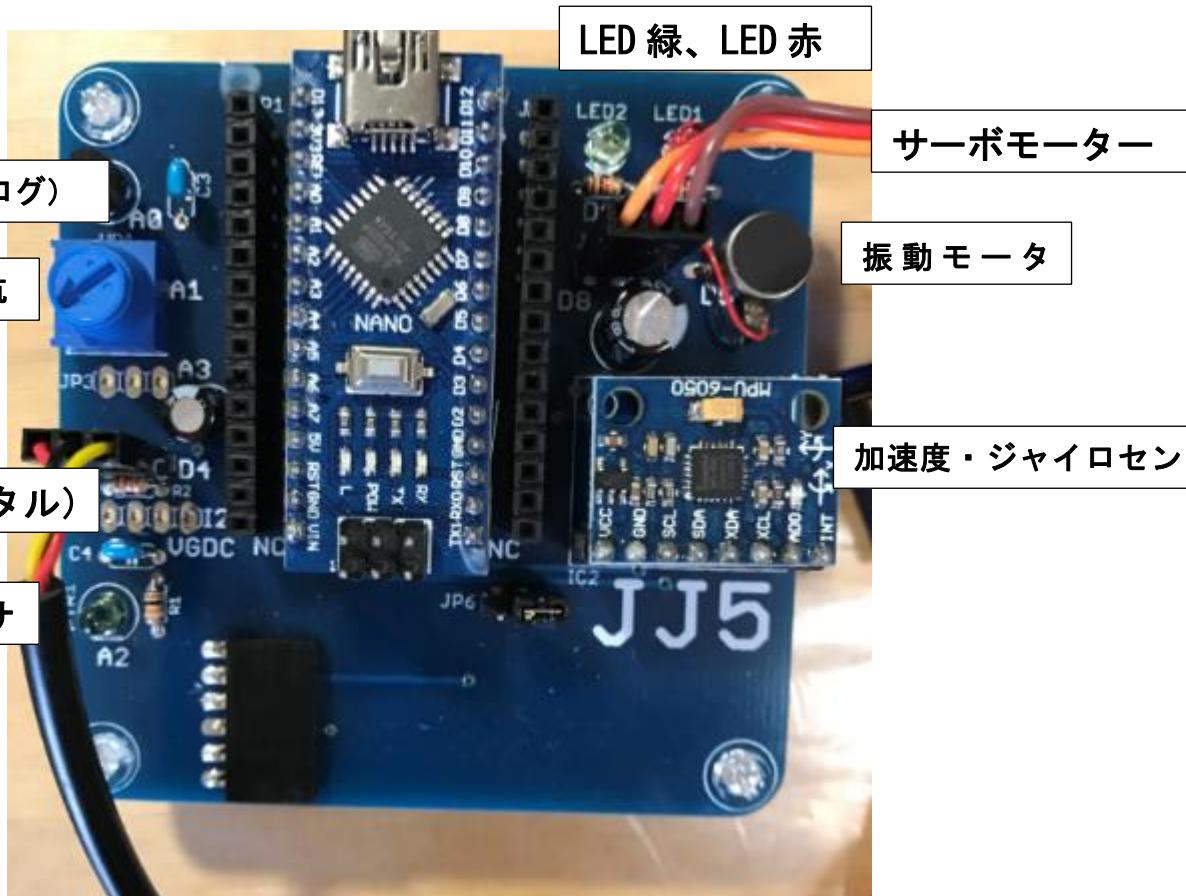
温度計も  $\pm 2^{\circ}\text{C}$  程度の誤差があるので、ほぼ良い値である。

センサ (アナログ) はトランジスタのよう形状で、指で  
触って温度を高くできる。シリアルモニタ、シリアルプロッタ  
で表示できる。

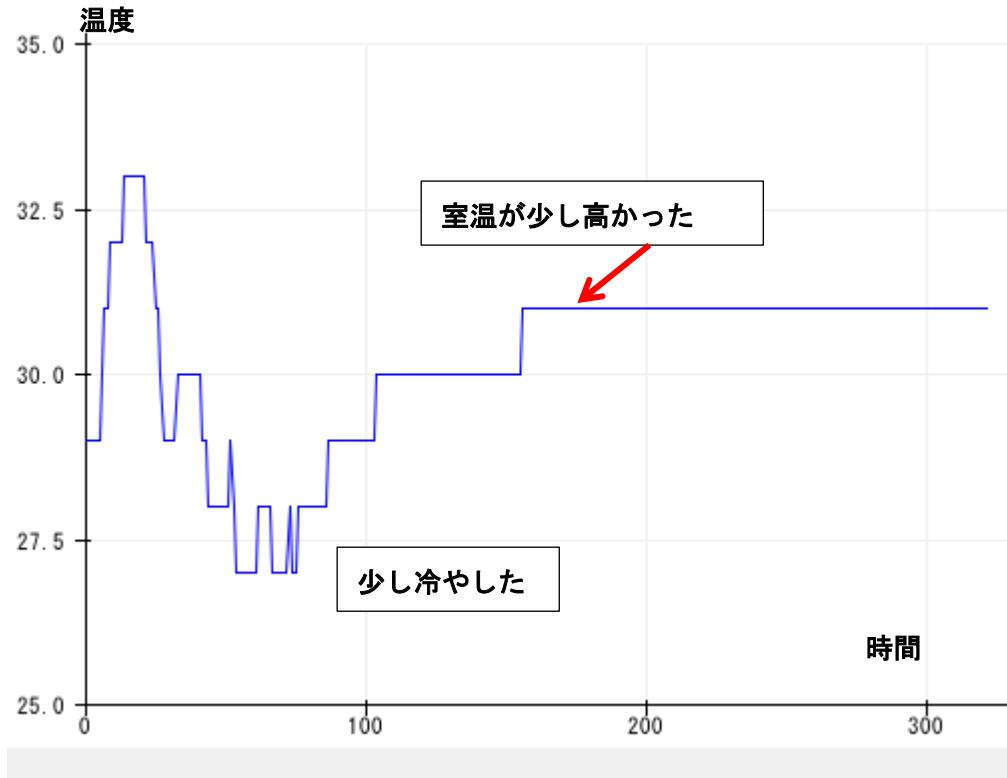
この参考「温度センサ LM61CIZ の使い方」の場合と、

ライブラリにある TEMP とで温度を決める方式の違いを考察せよ。





arduinoNano の基板 既にセンサを配置してある。



## 温度センサ（アナログ）を少し冷やしてみたときの様子

シリアルモニタ、シリアルプロッタで表示してみよう。

このような図を自分のレポート等に張り付ける場合は、MSWordで[挿入]で[スクリーンショット]する。

マウスで囲った部分が貼付く。

## 2) 温度をはかるセンサとして DS18B20 1線通信方式

スケッチを読んで動作を理解してみよう。決まり文句があり大まかな動作が分かる。

DS18B20 Dallas の読み取り ケーブルの付いたセンサの場合

DS18x20\_Dallas

```
1 // ds18b20, i2c
2 #include <DallasTemperature.h> // DS18B20
3
4 int DS_PIN = 4;
5 OneWire dsw(DS_PIN);
6 DallasTemperature ds(&dsw);
7
8 void setup(void) {
9   Serial.begin(9600);
10  ds.begin();
11 }
12
13 void loop(void) {
14   ds.requestTemperatures();
15   Serial.println(ds.getTempCByIndex(0));
16   delay(500);
17 }
```

DS18B20

DallasTemperature.h をインクルード

DS\_PIN を 4 とする

OneWire 接続 |

DallasTemperature. | このようにする

setup 関数

Serial.begin 9600 ポー シリアル通信速度

ds.begin

繰り返し loop

ds で温度を要求

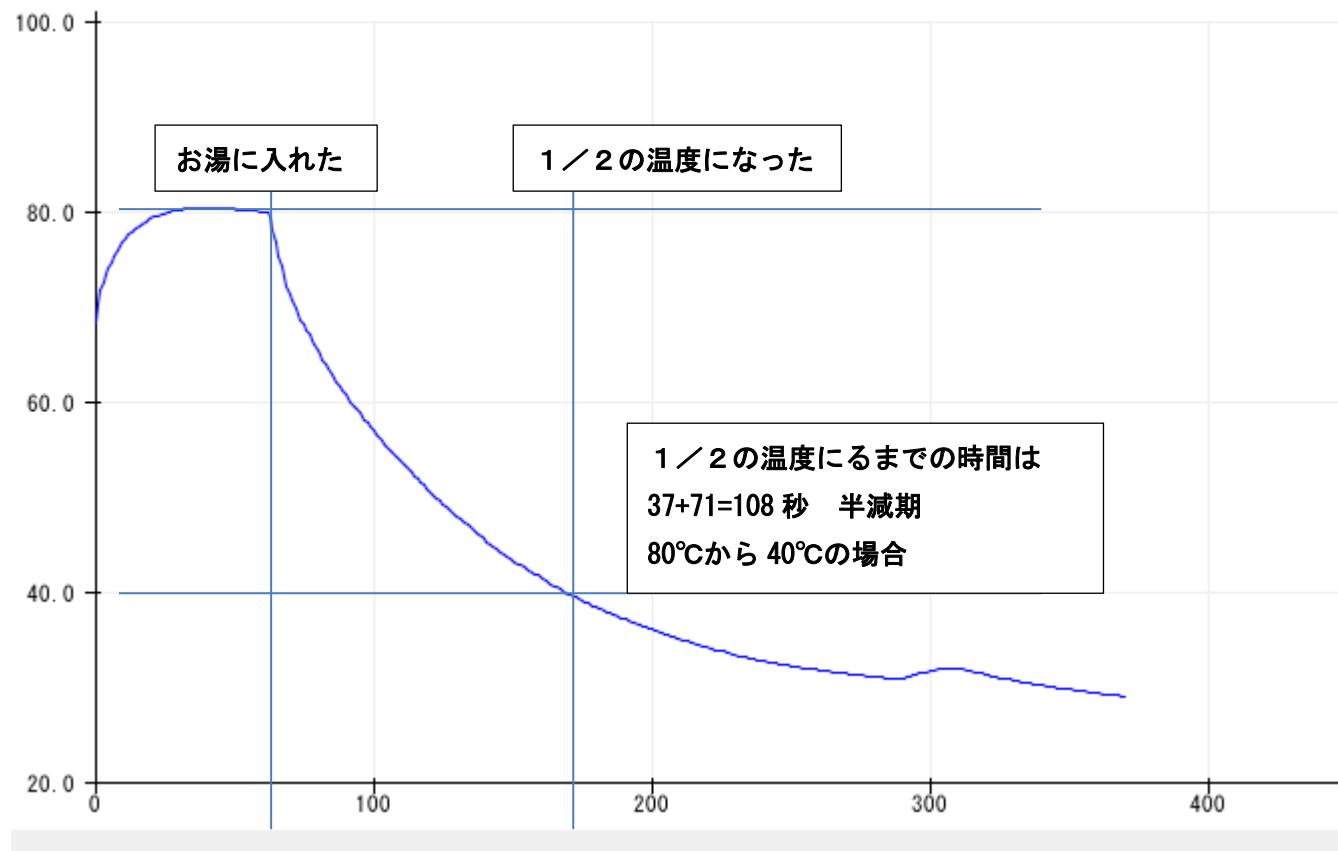
ds.getTempCByIndex(0) で値を読み Serial.print

500ms 待つ

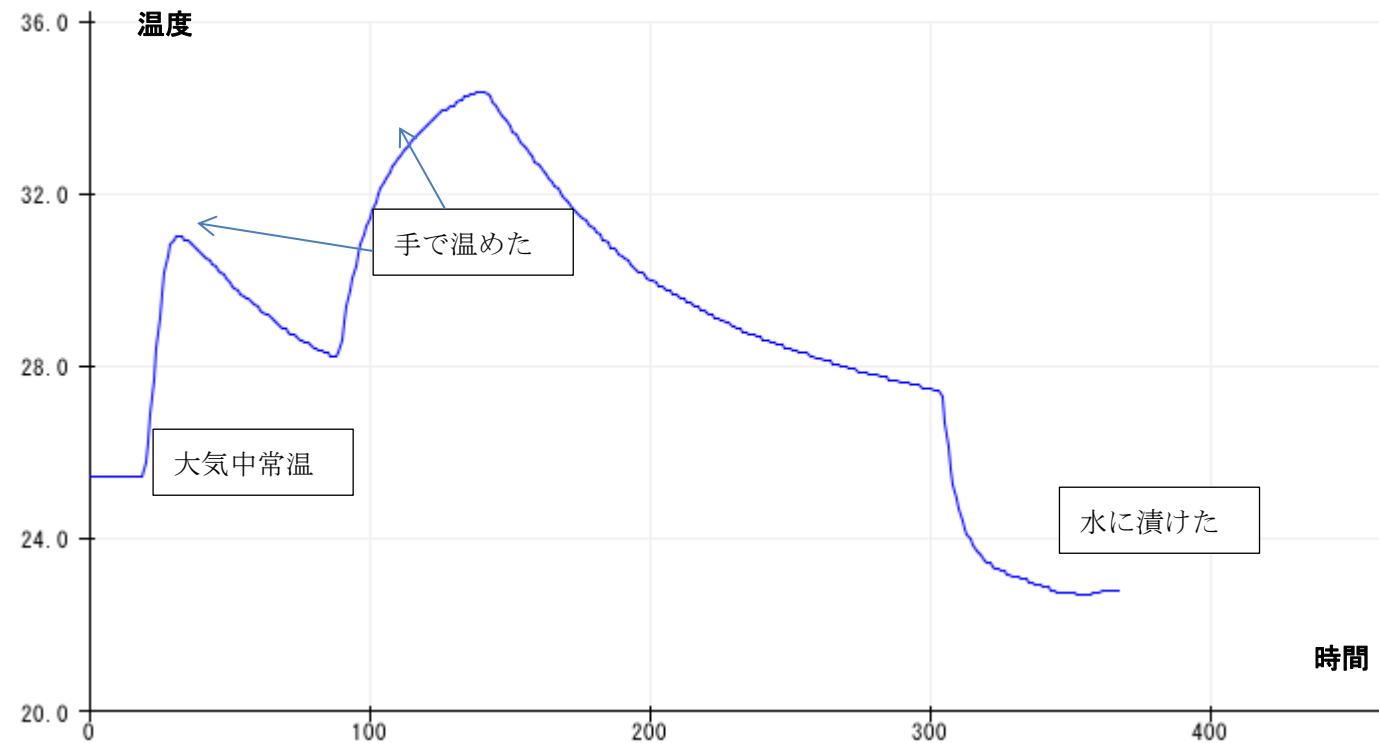
課題 1) LM61C1Z と比較してみよう。

課題 2) お湯をカップにいれて、温度計と温度センサ (DS18B20) (防水されている) をカップに  
入れる。シリアルプロッタで表示する。 温度が一定になったら、温度計とセンサを取り出して、  
温度の下がり具合をプロッタで測定する。どのようなグラフになるか？また、温度が 1/2 になる時間は  
どの程度か？半減期と呼ばれる。測定してみよう。

追記) 実は、放射線などの半減期があり、「使用済み核燃料の保管期間 フィンランドの最終処分場では 10~20 万年  
安全に保管する必要がある」とのこと。もちろん、非常に短いものもあるし、非常に長いものもある。



60°C～30°Cの場合は、もっと長くなる。常温に近づくため。



## 温度の変化の例

## 予備の課題) OneWire 通信による読み出し例

DS18x20\_Temperature を読み込む

```
// ds18b20, i2c
#include <OneWire.h>
OneWire  ds(4);  // on pin 4 (a 4.7K resistor is necessary)

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

void loop(void) {
  byte i;
  byte present = 0;
  byte type_s;
  byte data[12];
  byte addr[8];
  float celsius, fahrenheit;

  if ( !ds.search(addr)) {
    Serial.println("No more addresses.");
    Serial.println();
    ds.reset_search();
    delay(250);
    return;
  }
```

```

Serial.print("ROM =");
for( i = 0; i < 8; i++) {
    Serial.write(' ');
    Serial.print(addr[i], HEX);
}

if (OneWire::crc8(addr, 7) != addr[7]) {
    Serial.println("CRC is not valid!");
    return;
}

Serial.println();

// the first ROM byte indicates which chip
switch (addr[0]) {
    case 0x10:
        Serial.println(" Chip = DS18S20"); // or old DS1820
        type_s = 1;
        break;
    case 0x28:
        Serial.println(" Chip = DS18B20");
        type_s = 0;
        break;
    case 0x22:
        Serial.println(" Chip = DS1822");
        type_s = 0;
}

```

```

break;

default:
  Serial.println("Device is not a DS18x20 family device.");
  return;

}

ds.reset();
ds.select(addr);
ds.write(0x44, 1);      // start conversion, with parasite power on at the end

delay(1000);      // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE);      // Read Scratchpad

Serial.print("  Data = ");
Serial.print(present, HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) {          // we need 9 bytes
  data[i] = ds.read();
  Serial.print(data[i], HEX);
  Serial.print(" ");
}
Serial.print(" CRC=");
Serial.print(OneWire::crc8(data, 8), HEX);

```

```

Serial.println();
// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
    raw = raw << 3; // 9 bit resolution default
    if (data[7] == 0x10) {
        // "count remain" gives full 12 bit resolution
        raw = (raw & 0xFFFF) + 12 - data[6];
    }
} else {
    byte cfg = (data[4] & 0x60);
    // at lower res, the low bits are undefined, so let's zero them
    if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
    else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
    else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
    // default is 12 bit resolution, 750 ms conversion time
}
celsius = (float)raw / 16.0;
fahrenheit = celsius * 1.8 + 32.0;
Serial.print("  Temperature = ");
Serial.print(celsius);
Serial.print(" Celsius, ");
Serial.print(fahrenheit);
Serial.println(" Fahrenheit");

```

}

## 上のスケッチを実行したときのシリアルモニタの出力例

```
ROM = 28 FF A8 8 8A 16 3 F5
Chip = DS18B20
Data = 1 90 1 4B 46 7F FF C 10 33  CRC=33
Temperature = 25.00 Celsius, 77.00 Fahrenheit
No more addresses.
```

```
ROM = 28 FF A8 8 8A 16 3 F5
Chip = DS18B20
Data = 1 90 1 4B 46 7F FF C 10 33  CRC=33
Temperature = 25.00 Celsius, 77.00 Fahrenheit
No more addresses.
```

温度は手で温めると上がっていく。

## センサのしくみを知ろう（基礎編）3

### 6月3日 明るさをはかる

照度とはある光源によって照らされている明るさの度合いで、ルックス（lx）の単位で表される。

照度計で測定する。光源の明るさを表すのは、輝度で、単位は  $\text{cd}/\text{m}^2$ （カンデ毎平方メートル）である。

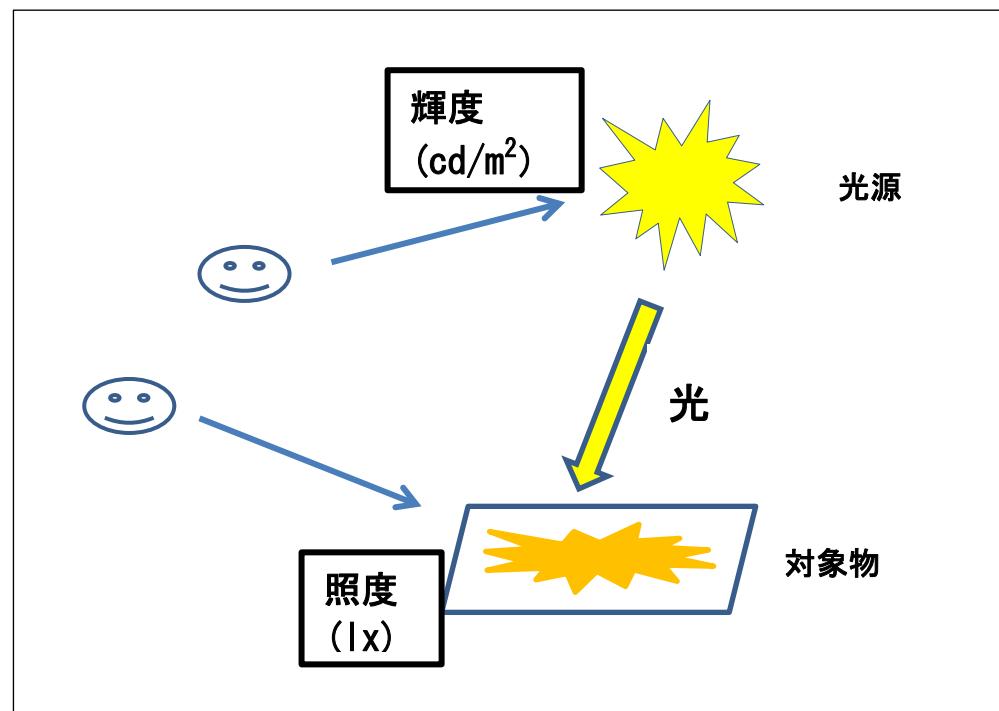
測定には輝度計を用いる。照度は光源からの距離により変化するが、輝度は光源そのものによって

決まるものである。

光の明るさに関する物理量は次の通りである。

- a) 光度  $\text{cd}$ （カンデラ） 光源の明るさ
- b) 輝度  $\text{cd}/\text{m}^2$  人間の感ずる明るさ
- c) 光束  $\text{lm}$ （ルーメン） ある方向に放射された光
- d) 照度  $\text{lx}$ （ルックス） 投射された面の明るさ

ここでは d) の照度について測定し、特性を理解しよう。



## 第1回目に arduino の基礎を学んだ

1) LED の点滅 スケッチ例→ 01. Basics→ Blink <<資料1回目 9 ページ参照

2) LED\_RedGreen を読み込む <<赤と緑の LED を点滅

赤と緑を互い違いに点滅するように変更してみよう

赤点灯→赤消灯→赤点灯→赤消灯→赤点灯→赤消灯

青消灯→青点灯→青消灯→青点灯→青消灯→青点灯

- ・ 変わった点灯消灯パターンを 1 つ考えてみよう・

### スケッチ赤と青の LED の点滅

```
//LED_RedGreen

pinMode(LED_GREEN_PIN, OUTPUT); //LED (赤、緑)は D9 と D11 に接続されている
pinMode(LED_RED_PIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_RED_PIN, HIGH);
  digitalWrite(LED_GREEN_PIN, HIGH);
  delay(1000);
```

```
digitalWrite(LED_RED_PIN, LOW);  
digitalWrite(LED_GREEN_PIN, LOW);  
delay(1000);  
}
```

### 3) 半固定抵抗 (10kΩ) A1 (analogRead)

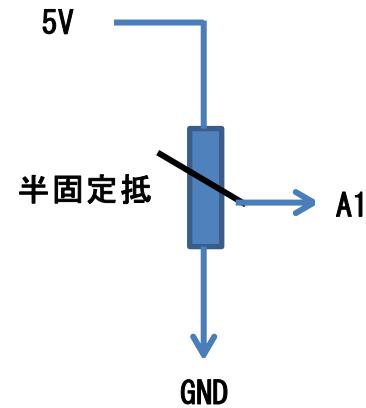
半固定抵抗は抵抗の値を変更できる素子で可変抵抗とも呼ばれる。

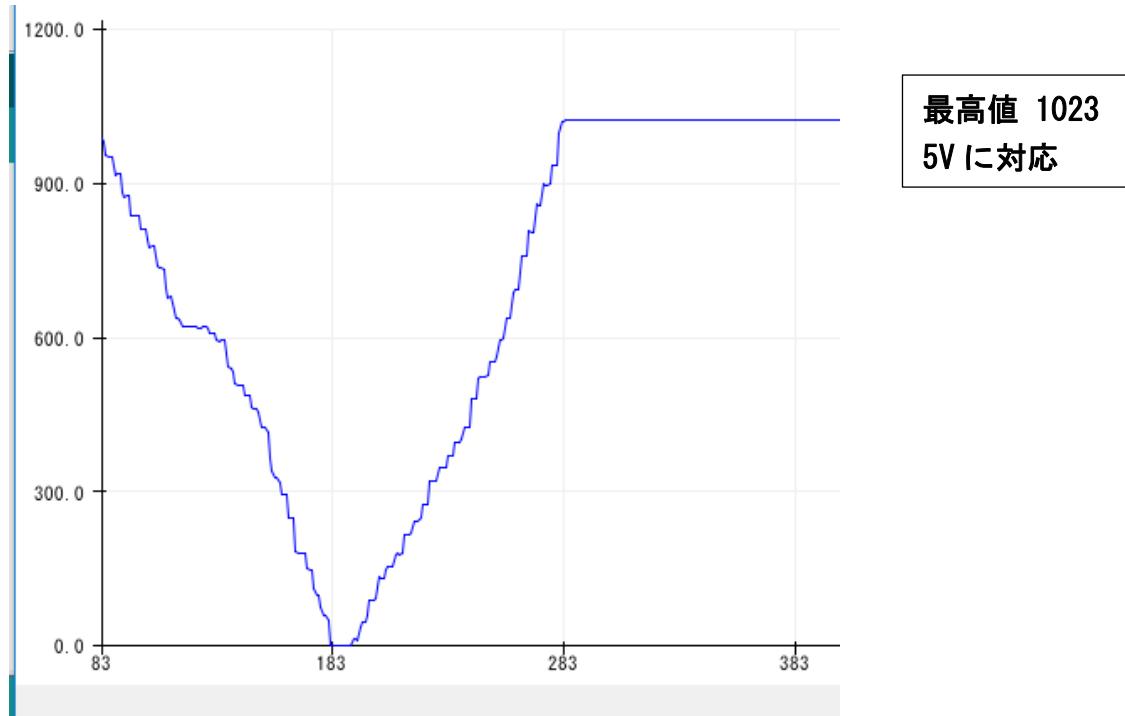
これをテストするには、ライブラリにある `VOL` を読み込む。

半固定抵抗の値を少しづつ変化させて、値の変化をシリアルプロットしてみよ。

最大値と最小値を調べよ。シリアルプリントおよびシリアルプロッタで表示してみよ。

半固定抵抗の読み出しが A1 ポートから





半固定抵抗の値をプロットしたもの

```
//VOL
// potentional meter (A1)
int PT_PIN = A1;

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

```
void loop() {  
    int pt = analogRead(PT_PIN); // 0 - 1023 5V が 1023 の値に対応  
    Serial.println(pt);  
    delay(500); //少しゆっくり表示  
}
```

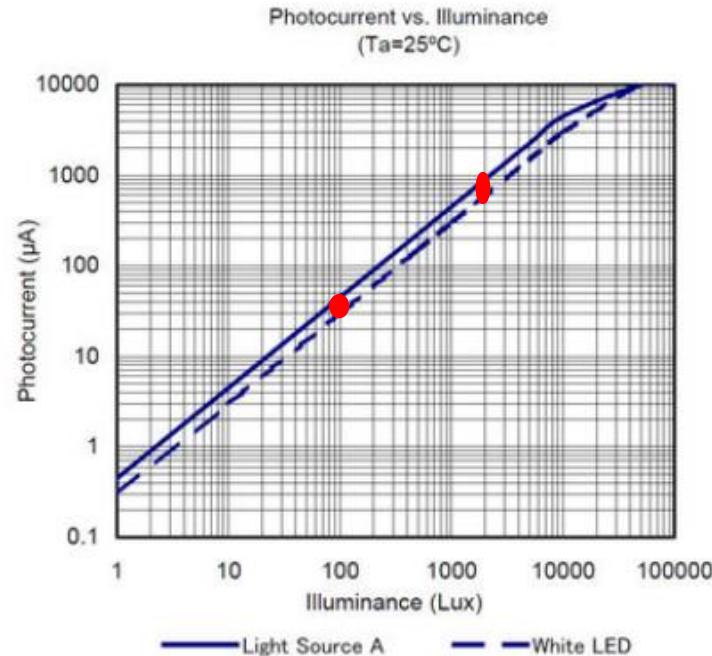
#### 4) 照度センサ (NJL7502L) A2 (analogRead)

照度センサ (NJL7502L) は別名フォトトランジスタとも呼ばれ、光に感じるトランジスタである。

白色 LED の 100Lux の明るさに対して約  $35 \mu\text{A}$  の光電流が得られる。

照度と光電流の関係は <http://www.eleki-jack.com/akifan/kanzaki/asens030010.jpg> から

次のグラフに表されている。



線の違いは光源の違い。点線は白色 LED。330Ωの負荷抵抗で光電流を電圧に変換すると、  
 $330\Omega \times 33\mu\text{A} = 10890\mu\text{V} = 0.01089\text{V}$   
 図から 2000Lux の時  $1000\mu\text{A}$  があるので、出力電圧は  $0.33\text{V}$  となる。

照度センサは光が当たると内部の抵抗が下がって、流れる電流が増える素子である。

```
//Lux NJL7502L
// light sensor (analog)
int PT_PIN = A2;

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

```

void loop() {
    int sensorValue = analogRead(PT_PIN); // センサ値の読み込み
    float vo = sensorValue * 5.0 / 1024; // 電圧 Vin に変換
    float lux = 222.0 * vo; // 照度 lux に変換
    Serial.println(lux);
    delay(100); // waits for the servo to get there
}

```

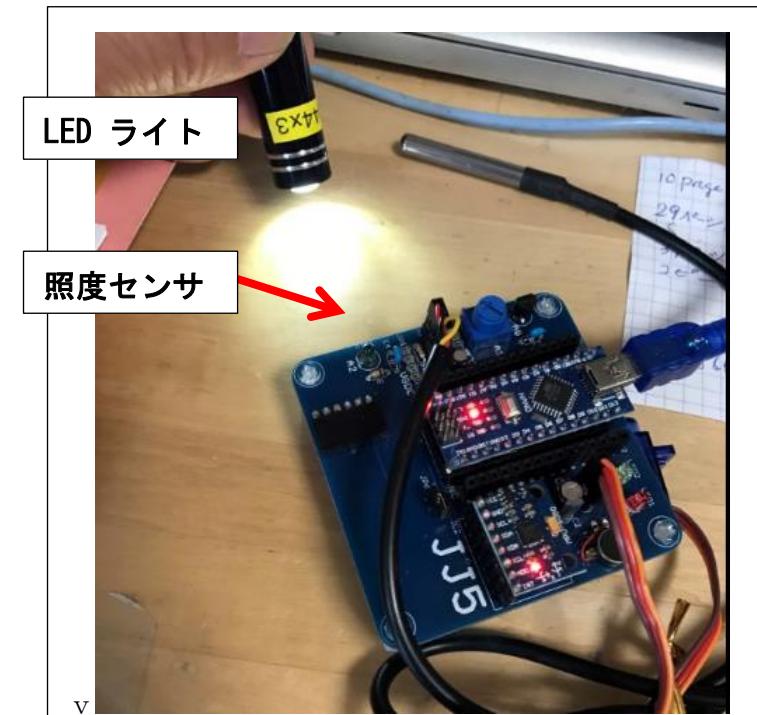
課題 1 使用している照度センサ (NJL7502L) は光の特性に敏感である。  
LED ライトを用いて、照度センサの感度の範囲を大体測定してみよ。

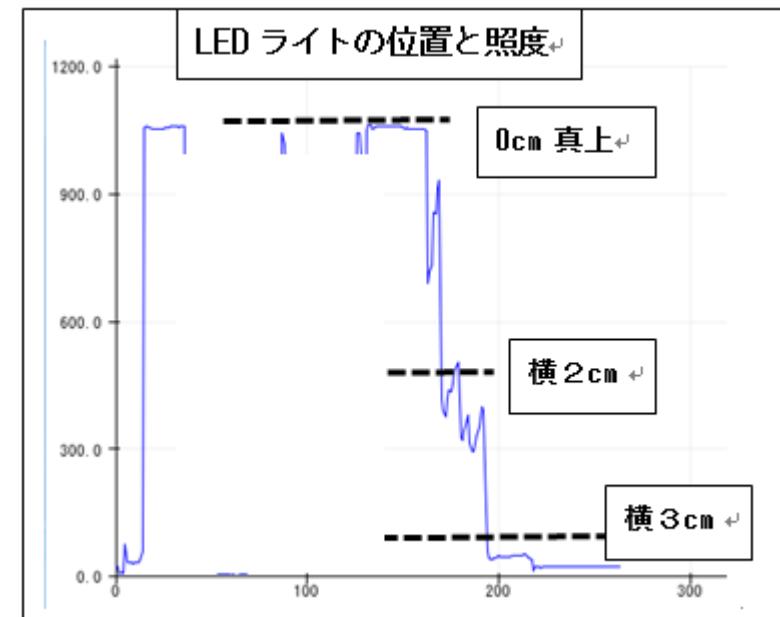
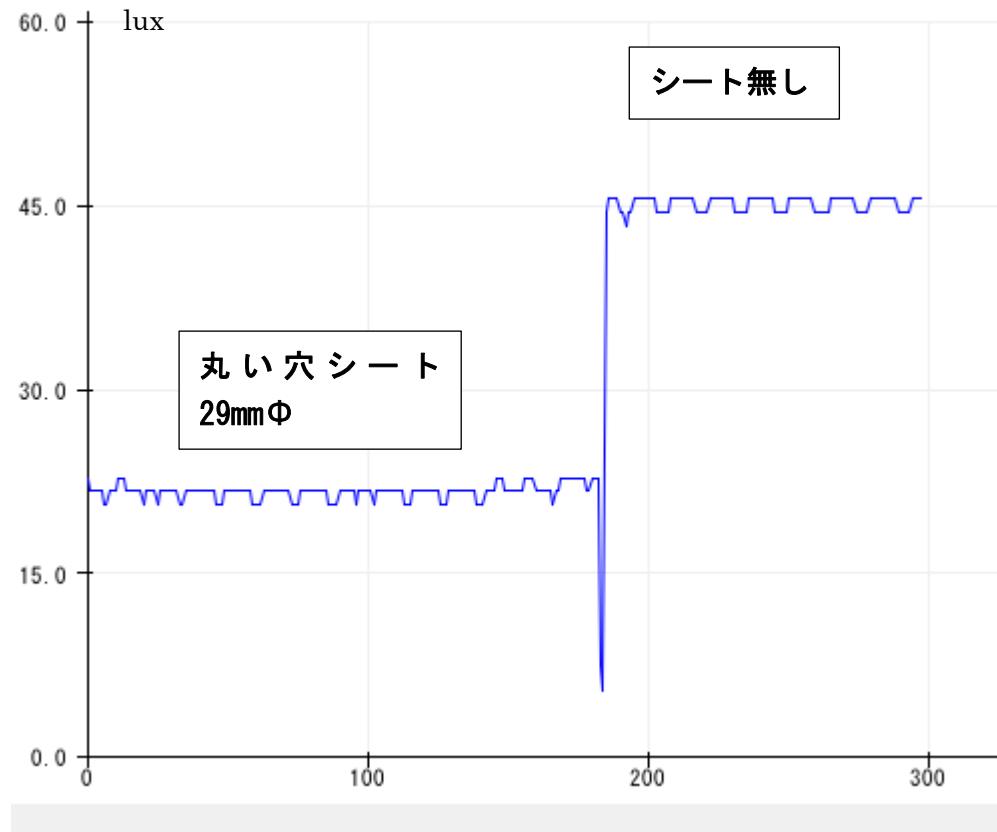
課題 2 机の上の照度（明るさ）は、新聞を読むとき、大体いくら以上必要か。

課題 3 直径 20mm、30mm などの円形スリットを用いて、照度センサの感度の範囲を調べよ。

### 計算式

sensorValue をセンサから読み取った値とすると  
 $vo = \text{sensorValue} \times 5.0 / 102$  ボルト  
 照度 lux =  $222.0 \times vo$  ルックス





## 照度センサの測定例

LED ライトによる感度テスト

LED ライト 高さ 10cm、横ずれ 5cm 程度で ほとんどセンサの値が 0 となる

## 関数を用いた照度センサの読み取り例

```
// light sensor (analog)
int PT_PIN = A2;

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

void loop() {
  Serial.println(read_lux());
  delay(100); // waits for the servo to get there
}

float read_lux() { //センサの読み取り関数
  int sensorValue = analogRead(PT_PIN); // センサ値の読み込み
  float vo = sensorValue * 5.0 / 1024; // 電圧 Vin に変換
  float lux = 222.0 * vo; // 照度 lux に変換

  return lux;
}
```

## 次週以降

④ 6/10 サーボモーターと振動モーター

振動モーター(analogWrite), サーボモーター(SG90) (Servo)

LED (analogWrite) との組み合わせ

⑤ 6/17 動きを知るセンサー

加速度・ジャイロセンサー(MPU6050)(I2C)

⑥ 6/24 まとめ

センサとアクチュエータの組み合わせ, 討論