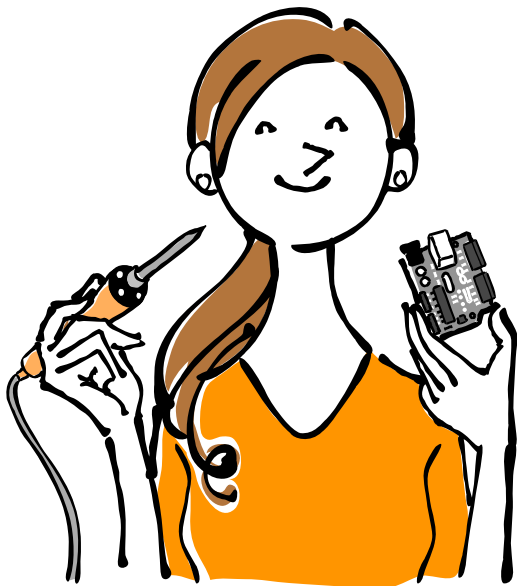


第Ⅲ部

ステップ アップ編



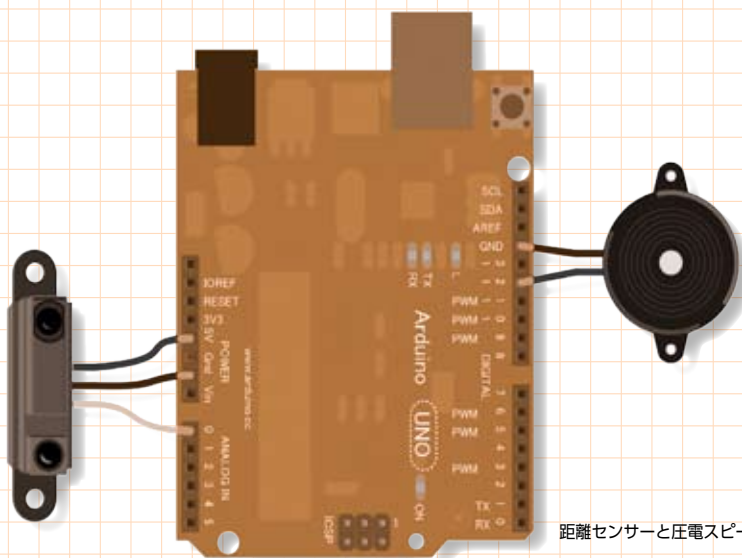
前の基礎編において、基本的なアナログ入出力とデジタル入出力について理解できたのではないのでしょうか。いよいよステップアップ編では、さらにArduino活用のレベルアップについて紹介していきます。

第6章では、これまで覚えたアナログやデジタル、それにシリアル通信を理解したことで、より高度な電子部品まで使いこなせるように紹介していきます。よく使うであろう温度センサーや光センサー、加速度センサー、距離センサー、それに出力系のLCD（液晶ディスプレイ）の使い方を紹介しましょう。これによって、入出力の基本的電子部品の組み合わせにいろいろと挑戦できるようになります。

最終章の第7章では、知っておくと便利な機能群を紹介します。ここでは、タイマー機能やタブ機能、不揮発メモリー（EEPROM）、割り込み機能、それにシリアル通信機能などについてまとめました。また、インターネットを利用した情報収集の方法などについても紹介します。

第 6 章

高度な入力出力部品を 使ってみよう



距離センサーと圧電スピーカーを使った応用例

ここでは、身の回りで便利に使える入力系のセンサー類やLCD（液晶ディスプレイ）などの出力系の電子部品を紹介していきます。Arduino上で電子部品をどう接続し、つぎにスケッチを作成し、すぐに動かすためにはどうするかを説明していきます。

すでに第4章と第5章でも紹介したように、電子部品をArduinoにつないで動かすには、まず部品がアナログかデジタルのどちらで扱うかがポイントとなります。つぎに、スケッチの作成で、どのような変換式を使うかが重要となります。この2つのポイントをしっかりと押さえることで、ここで紹介する電子部品を簡単に使いこなせるようになってきます。

それでは、基本的な仕組みを理解し、その使い方を覚えていきましょう。



6.1

温度センサー(アナログ)を使ってみよう

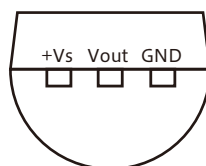
最初に扱いが簡単なアナログ電子部品の温度センサーを説明しましょう。温度センサーは、安価なものであれば、1個が数十円程度で、しかも簡単に温度測定ができます。ただし、安価なものは、測定値の精度を求めることはできません。精度を求めるのであれば、使う状況や、まわりの環境に合わせた高価なデジタルセンサーなどを使う必要があります。しかもArduinoからの熱が伝わらないように、ケーブルで離したりする必要があります。

ここでは、安価で、簡単にArduino上で温度測定ができるアナログの温度センサー「LM61BIZ」の使い方を紹介しましょう。

(1) つないでみる

温度センサー「LM61BIZ」は、外界の温度変化によって抵抗値が変わるアナログセンサーで、その抵抗値の特性をもって、摂氏温度(単位:°C)温度や華氏温度(単位:°F)温度への変換計算を行います。この「LM61BIZ」は、3本のピンがあり、両端ピンが電源電圧(+Vs)とグラウンド(GND)で、中央のピンが出力電圧(Vout)となっています。

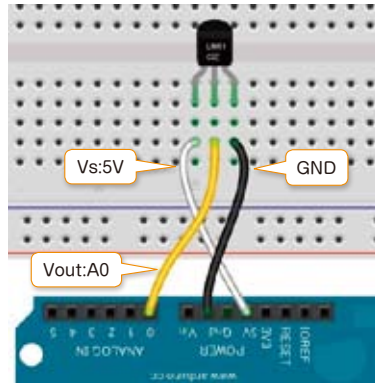
図6.1 温度センサー(LM61BIZ)と配線ピン(下から見上げた図)



それでは、図6.2のように、この温度センサーをArduinoに接続してみましょう。

ここでは、温度値が出力される「Vout」ピンは、アナログ入力ポートのA0番ピンに接続します（もちろん他のアナログ入力ポートのピンでも構いません）。

図6.2 温度センサー(LM61BIZ)の接続①



(2)スケッチを作成する

つぎにスケッチを作成していきますが、その前に、この温度センサー（LM61BIZ）の出力の仕様を確認していきましょう。仕様によると、この温度センサーの出力電圧（Vout）と摂氏温度（T）との関係式は、つぎのようになっています。つまり、これが変換式となります。

$$T=(Vout*Vs/1024-600)/10$$

ここで、Vsは、電源電圧（mV）となります。この場合5V（=5000mV）であるとする、つぎのようになります。

$$T=(Vout*5000/1024-600)/10$$

この関係式を使って、ここでは温度の値を、摂氏と華氏との2つの値で出してみましょう。先に、「LM61BIZ」によって出力された値を、上述した変換式で摂氏温度を求めます。つぎに、摂氏温度（T）から華氏温度（F）への以下の変換式を使ってスケッチを作成してみます。

$$F=T*9/5+32$$

それでは、スケッチ6.1にサンプルスケッチを紹介します。

スケッチ 6.1 温度センサー(LM61BIZ)を使った例

```

void setup() {
  Serial.begin(9600);
}
void loop() {
  int val = analogRead(A0);
  float cel = (float)val*500.0/1024.0-60.0; // 温度(摂氏 単位℃)の計算
  Serial.print ( "Celsius = ");
  Serial.print ( cel );
  Serial.print ( " / Fahrenheit =");
  Serial.println ( (cel * 9)/ 5 + 32 ); // 温度(華氏 単位℉)の計算
  delay(1000);
}

```

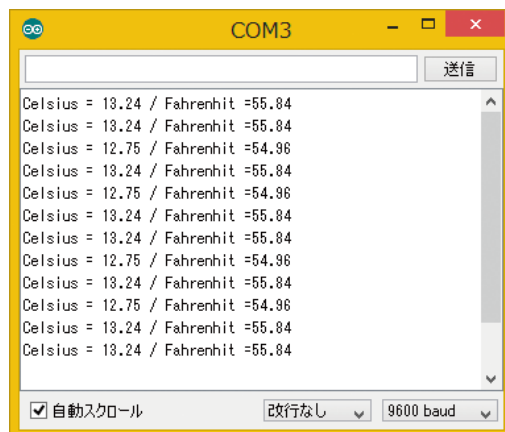
この場合も、出力温度はシリアルモニタを使って表示させています。このスケッチで特に注目すべき点は、出力されたセンサー値 (val) を摂氏温度に変換する式と、さらに摂氏温度を華氏温度に変換する式が使われているところです。

難しいところはありませんが、センサー値を読み込む「analogRead」が整数となるため、変換式ではそのセンサー値を実数に置き換えるための型変換キャストの「(float)」を付けていることにも注意してください。

(3)動かしてみる

それでは、「LM61BIZ」とArduinoとの接続配線を行い、PC上で上記のスケッチをコンパイルして、Arduinoに書き込んで実行してみてください。なお、温度の値はシリアルモニタを開いて見ることができます(図6.3)。

図6.3 温度センサー値のシリアルモニタ画面

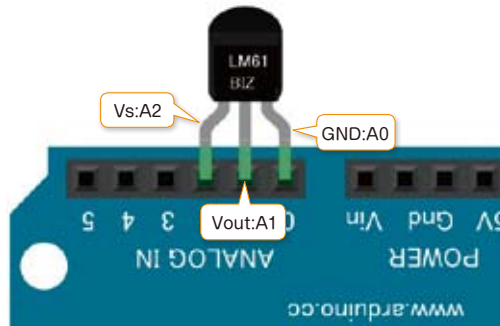


(4) デジタル入力ピンの電源とGNDのテクニック

ここまでは、電源電圧5VピンとGNDピンとして、Arduinoの標準ピンを利用しました。ここでは、2つのデジタルピンを用い、ピンの値をHIGH (5V) とLOW (0V) に設定して、電圧5VピンとGNDピンに切り替える特殊な使い方（テクニック）を紹介しましょう。また、アナログピンの「A0」から「A5」までは、デジタルピンの「D14」から「D19」までに相当することも利用します。

それでは温度センサーの3本のピンを、そのまま「A0」、「A1」、「A2」ピンに接続してみてください。具体的な接続方法は、図6.4のとおりです。

図6.4 温度センサー(LM61BIZ)の接続②



この場合のスケッチは、スケッチ6.2のようになります。

スケッチ6.2 連続したアナログピンで温度センサーを利用する例

```
void setup ()
{
  pinMode(A0, OUTPUT); // A0(LM61BIZ - GND)
  digitalWrite(A0, LOW);
  pinMode(A2, OUTPUT); // A2(LM61BIZ - VSS+)
  digitalWrite(A2, HIGH);
  Serial.begin(9600);
}

int getTemp(void) // 温度センサ読み取りおよび変換式関数
{
  int mV = analogRead(A1) * 4.88;
  return (mV - 600);
}

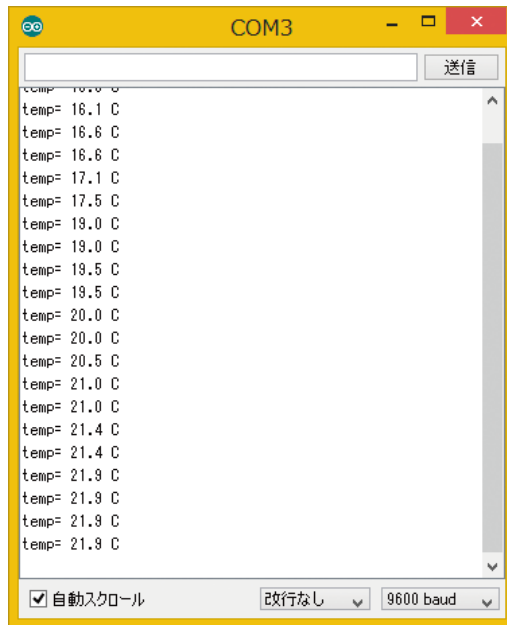
void loop()
{
  int temp = getTemp();
  char body[20];
  sprintf(body, "temp= %d.%d C", temp/10, temp%10); // 文字列連結関数
  Serial.println(body);
  delay(1000); // 待機時間
}
```

この中の「setup」関数の最初の4行で、「A0」ピンをデジタルピンとして宣言し、GNDを意味する「LOW」を設定し、つぎに「A1」ピンを同じくデジタルピンとして宣言し、電圧5Vを意味する「HIGH」を設定しています。温度センサーの真ん中のピンは、アナログ入力の「A1」ピンとして、「getTemp」関数の中で値を取り出しています。

ところで、この「getTemp」関数では、先のスケッチ6.1で算出した温度値よりも10倍大きい値で取り出しています。その理由は、次に紹介する「sprintf」関数を使った文字列処理で利用するためです。ここでは、出力表示する文字列(配列)「body」を使い、この「sprintf」関数によって温度値を代入し、次の「Serial.println(body);」でシリアルモニタ画面に結果を表示させています。

結果は、図6.5のように表示されます。

図6.5 温度センサー値を出力したシリアルモニタ画面



ここで紹介したデジタル出力の「HIGH/LOW」を電源電圧(5V)とGNDに利用する方法を学びましたが、この方法が必ずしもすべてのセンサーに使えるわけではないことに注意してください。特にデジタルセンサーでは、Arduinoの電圧が切り替わる際に不安定になり、値を取得できない場合があります。

(5) ポイントを理解する

この温度センサー「LM61BIZ」は、アナログセンサーであるため、「analogRead」関数を使って値を読み込みました。しかし、取得されたセンサー値は温度そのものではなく、変換式が必要となります。

アナログセンサーには、このように変換式が必要なものが多く、しかも簡単な式で表せるとは限りません。利用する温度センサーの特性や変換式を確認しながら、利用するようにしてください。

それからここでは、アナログ入力ポートを使って、電源電圧ピンとGNDピンとして使うことや、「printf」関数を使って、整数を文字列に組み込む処理を学びました。この2つのテクニックは、いろいろな場面で利用できますので覚えるようにしてください。

ここで「printf」関数定義を紹介しておきます。

「printf」関数の定義

```
int printf(pr, fm, x0, x1, ...xn)
```

ここで、pr は出力する文字列

fm は出力する文字列フォーマット

x0, x1...xn は文字列フォーマットに出てくる変数

例えば、先の例でのfmは、"temp= %d,%d C"となっています。この中に「%d」が2回出てきますが、この2つに変数、「temp/10」（tempを10で割った商）と「temp%10」（tempを10で割った余り）との2つが「%d」に割り当てられます。

例えば、「temp」の値が「123」の場合には、「temp/10」が「12」、「temp%10」が「3」となり、出力する文字列「body」には、「temp = 12.3 C」が入ります。

その他「%d」（整数）以外では、「%c」（1文字）、「%s」（文字列）、「%x」（16進数整数）、などがあります（注意：Arduinoでは、実数表記の「%f」が使えないために、上記のような処理を行いました）。

以下「printf」関数を使ったサンプルスケッチを掲載しておきます。

スケッチ 6.3 printf関数を使った文字列処理の例

```
char pr[30];
void setup() // printf関数のサンプル出力テスト
{ Serial.begin(9600);
  printf(pr,"%c,%s,%5x',%2d.%2d",'A',"BCDE",2013,15,21);
  Serial.print(pr); }
void loop() {}
```

ここでの結果は、以下のようになります。

```
A,BCDE,' 7dd',15.21
```

上記のスケッチを、いろいろと変更してみて、表示結果を確認してみてください。