

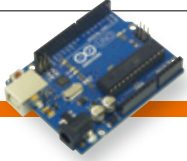
マイコン
活用シリーズ

AVR

Arduinoで 計る, 測る, 量る

見本

Various measuring techniques in Arduino



神崎康宏 著

計測したデータをLCDに表示, SDカードに記録,
無線/インターネットに送る方法を解説



CQ出版社

[第4章]

決められた入出力ポートだが逆に使いやすい

アナログ入出力もスケッチが用意されていて使い方は簡単

本章では、Arduinoのアナログ入力として、センサからの出力の代わりにボリュームを用いて、0Vから電源電圧まで変化する電圧を読み取り、この変化した電圧に対応した出力をLEDに加えてアナログ入出力のテストを行います。

続いて、アナログ入力の具体例として温度の測定を行います。そのため、正確に電圧を測定するための基準電圧の設定方法について確認し、半導体温度センサを実際に接続して温度を測定します。その後、応用例として湿度を測定するための乾湿球温度計を作ります。

4-1 アナログ入力とアナログ出力

4-1-1 マイコン・ボードとブレッドボードでテスト

Arduinoのアナログ入力ポートからは、0～数Vの連続的に変化するアナログ電圧を容易に読み取ることができます。読み取ったアナログ・データは、Arduinoに用意されているスケッチにて、加減乗除はもとよりもっと高度な関数の計算処理も行うことができます。そのため、いろいろなセンサを

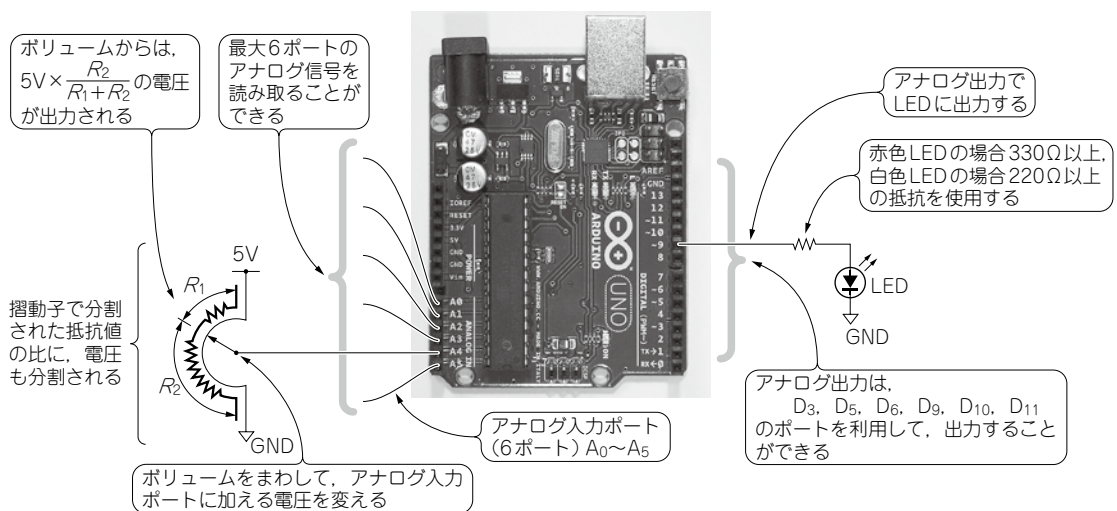


図4-1 アナログ入出力の確認

アナログ入力 `analogRead()` は
入力するポートを指定すると、結果
が関数の値として戻される

「アナログ・ポート0番」
から読み取った値を、
「アナログ出力ポート9番」
に書き出すという記述

アナログ出力 `analogWrite()` は
出力ポートと出力する値を指定する

```
ard040010 | Arduino 1.0
File Edit Sketch Tools Help

ard040010
void setup(){
}
void loop(){
  analogWrite(9,analogRead(0));
}
```

Setup() には、何も記述しない

analogRead(0) はアナログ・ポート0番のデータを読み込む

図4-2 アナログ入出力のスケッチ

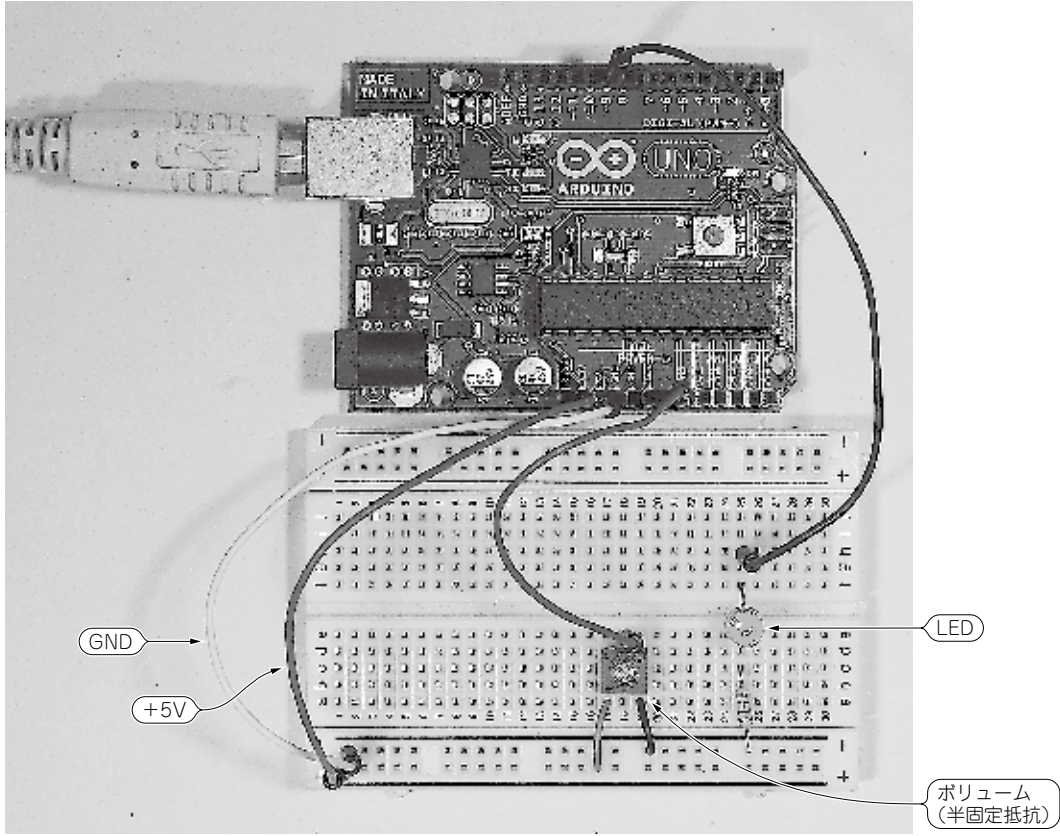


図4-3 アナログ入力、アナログ出力のテスト回路

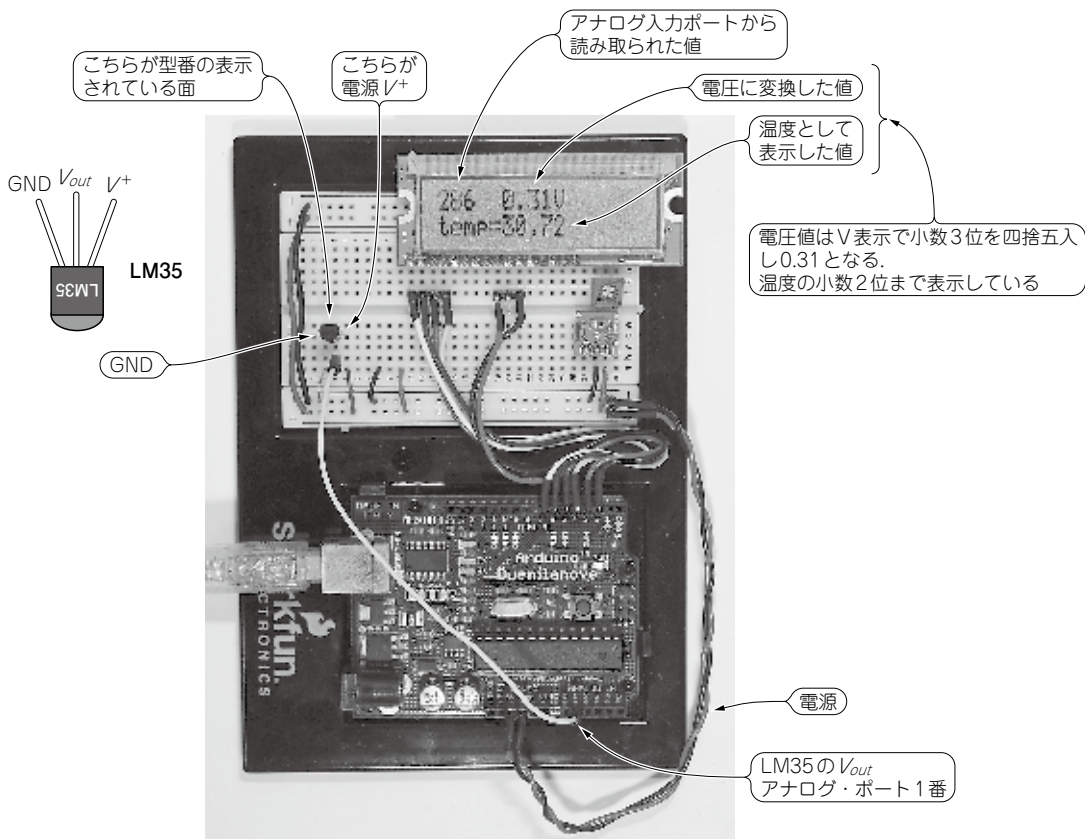


図5-11 LM35の測定結果をLCDに表示する配線

の内部基準電圧に変更するために、`setup`関数で`analogReference (INTERNAL);`の命令を記述してあります。

ここでは、基準電圧としてArduinoの内部の基準電圧を使用するので、温度は次の式で求まります。

$$(1100.0/1024) * \text{indata} * 0.1$$

後は、表示をクリアして、次に書き込むカーソルの位置を上段の左端にセットする`lcd.clear()`、変数、文字列を表示する`lcd.print()`、次の表示位置を指定する`lcd.setCursor()`の各関数を利用して目的を達成しています。このスケッチを利用して、LM35の測定結果をLCDへ表示したようすを図5-11に示します。

5-4 LM35DZを利用して湿度の測定結果をLCD表示する

第4章でLM35DZを二つ使用して湿度を測定した結果についても、結果をLCDに表示するように変更します。図5-12に変更したスケッチを示します。図5-13に測定結果を表示したようすを示します。また、図5-14にはセンサ部分のようすを示します。


```

ard050040 | Arduino 1.0
File Edit Sketch Tools Help
ard050040
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
float t1=0;
float t3=0;
float vref=1100;
int tset=0;
void setup()
{
  Serial.begin(9600); //
  analogReference(INTERNAL);
  lcd.begin(16,2);
}
void loop()
{
  t1=vref*analogRead(1)/1024/10; //
  t3=vref*analogRead(3)/1024/10; //
  float ps = 6.11*pow(10,(7.5*t3/(237.3+t3)));
  float pst = 6.11*pow(10,(7.5*t1/(237.3+t1)));
  float pt = pst - 0.0012*1013.25*(t3-t1)*(1+t1/610);
  float pts=pst-0.000662*1013.25*(t3-t1);
  int HM =(pt/ps)*100;
  int HM2 =(pts/ps)*100;
  lcd.clear();
  lcd.print("D=");
  lcd.print(t3);
  lcd.print(" W=");
  lcd.print(t1);
  lcd.setCursor(0,1);
  lcd.print(pt);
  lcd.print("/");
  lcd.print(ps);
  lcd.print(" ");
  lcd.print(HM);
  lcd.print("%");
  Serial.print(" t1 "); //
  Serial.print(t1); //
  Serial.print(" t3= "); //
  Serial.print(t3); //
  Serial.print(" ps ");
  Serial.print(ps);
  Serial.print(" pt ");
  Serial.print(pt);
  Serial.print(" HM ");
  Serial.print(HM);
  Serial.print(" pts ");
  Serial.print(pts);
  Serial.print(" HM2 ");
  Serial.print(HM2);
  Serial.println();
  delay(1000);
}

```

Annotations in the image:

- ヘッダ・ファイルを読み込む (Header file loading) - points to `#include <LiquidCrystal.h>`
- wet 湿球 (Wet bulb) - points to `t3`
- dry 乾球 (Dry bulb) - points to `t1`
- 湿度の計算 (Humidity calculation) - points to the calculation of `pt` and `pts`
- 相対湿度 (Relative humidity) - points to `HM` and `HM2`
- 2行目左端にカーソルをセット (Set cursor at the start of the second line) - points to `lcd.setCursor(0,1);`
- データ確認のためのもので、なくてもよい (Data confirmation only, not necessary) - points to the `Serial.print` statements

Additional annotations on the left side:

- 第4章参照 (Refer to Chapter 4) - points to the humidity calculation section
- LCDへの表示 (Display on LCD) - points to the `lcd.print` statements

図5-12 湿度を計算で求めるスケッチ

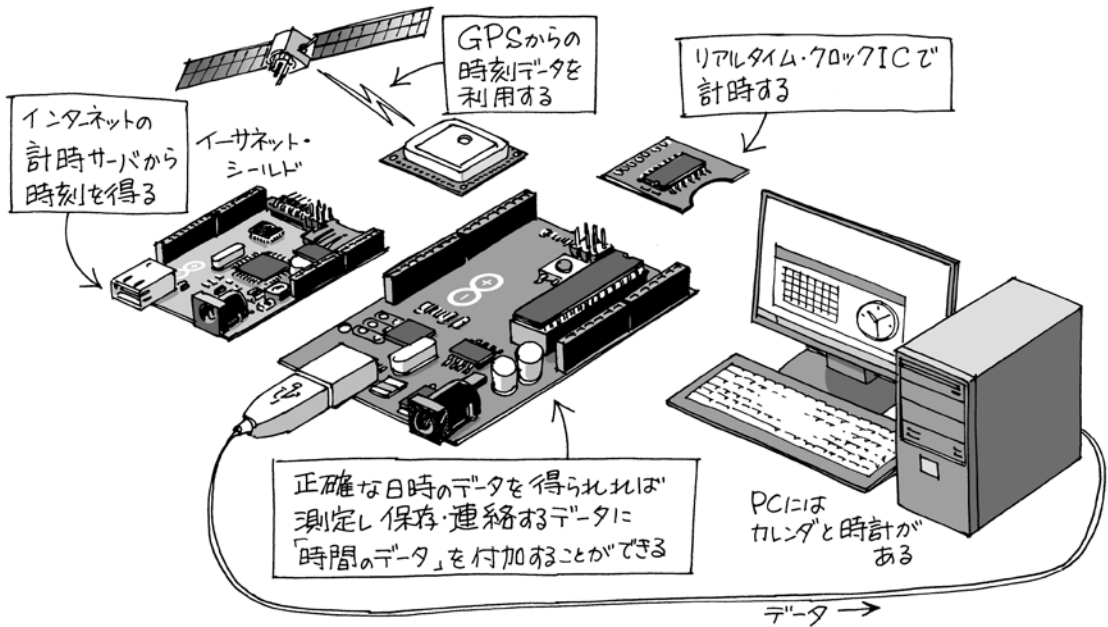


図6-7 PCで日付、時刻を得る方法

② GPS衛星から時刻を得る

GPS受信モジュールを利用し正確な時刻を得る。位置の検出を行うのでなければ衛星の補足の数も少なく済み、室内でも時刻データ得ることができる場合が多い

これらの方法が考えられます。ここでは、次に示すリアルタイム・クロック・モジュール RTC1307を使用することにします。

このモジュールはSparkfun製のモジュールで、DS1307というICを搭載しI²Cのインターフェースをもっています。バックアップ電源としてCR1225電池を搭載しているので、長期(9年以上)にわたって、Arduino側の電源と無関係に時刻を刻むことができます。この他にもI²CのインターフェースをもったArduino用のセンサも多くあるので、アナログ入力A₅、A₄の2本のピンで二つ以上のデバイスを接続できます。I/Oピンのあまり多くないマイコンでは助かります。

6-2-2 RTCモジュール (DS1307)

RTCモジュールは、図6-8(a)に示すように8ピンのDS1307と水晶振動子で構成されています。このICは5Vの動作電圧で、主な仕様は表6-1に示すようになります。電源の5VとGNDの電源端子、SDA、SCLのI²Cの端子と1Hz～32.768kHzのクロックを出力することもできるSQW端子が用意されています。

このRTCモジュール基板の端子の穴は2.54mmピッチのピン・ヘッダが利用できるもので、このモジュールにピン・ヘッダをはんだ付けし、ブレッドボードやユニバーサル基板で容易に利用できます。

動作電源が5Vなので5V電源のArduinoと直接接続することができます。このリアルタイム・クロック・モジュールは共立エレクトロニクスまたはスイッチサイエンスなどから入手することができます。

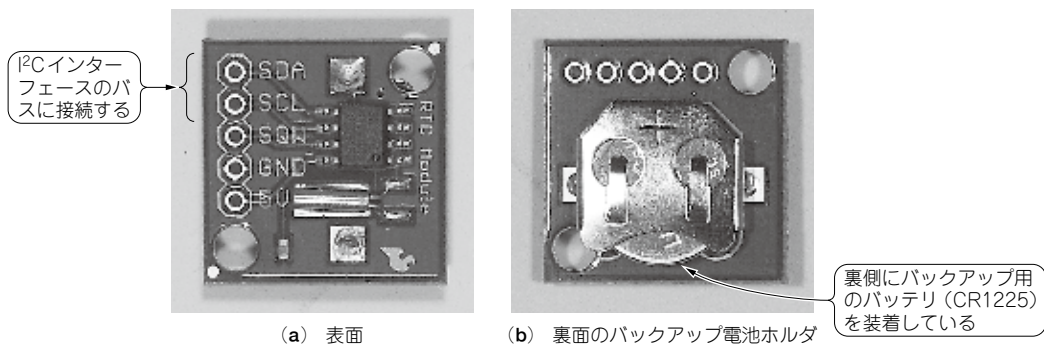


図6-8 DS1307と水晶振動子が実装されたRTCモジュール

表6-1 RTC1307の主な動作仕様

	min	max	
動作電源電圧 (V_{cc})	4.5V	~ 5.5V	動作時
Logic 1 (HIGH)	2.2V	~ $V_{cc} - 0.3V$	
Logic 0 (LOW)	-0.5V	~ +0.8V	
バッテリー電圧	2.0V	~ 3.5V	待機時
SCLクロック		100kHz	I ² Cクロック

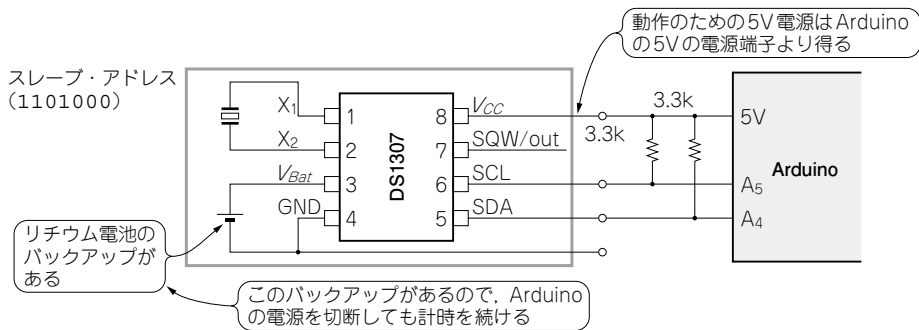


図6-9 ArduinoにDS1307を接続する

基板の背面には、図6-8(b)に示すようにCR1225のリチウム電池によるバックアップ電源も用意されています。

◆ DS1307の設置方法

DS1307の動作電源電圧は5Vです。そのため標準の5V動作のArduinoとDS1307との接続は図6-9に示すようになります。Arduinoのアナログ入力ポート5番(SCL)、4番(SDA)をDS1307のSCL端子、SDA端子に接続します。アナログ入力ポートですが、Wireライブラリを利用するときは、この二つのポートがデジタル信号でデータをやりとりするI²Cのバスのポートとして割り当てられています。

モジュールにピン・ヘッダをはんだ付けして、ブレッドボードにセットしてテストすることもできるようにします。使用するときは、仮配置でなくArduino用のユニバーサル基板を使用し、DS1307のリアルタイム・クロック・モジュールを取り付けるための5ピンのピン・ソケット、I²Cバスのプルアップ抵抗3.3kΩ 2本を基板に取り付け、はんだ付けします。

を行います。ArduinoでSPI通信を行う場合は図7-3に示すようにSPIインターフェースの各信号線にそれぞれデジタル・ポートの13, 12, 11, 10が割り当てられています。SSの10番ピン以外はそれぞれ専用のピンが割り当てられ、SPIインターフェースを利用するときはほかには利用できません。

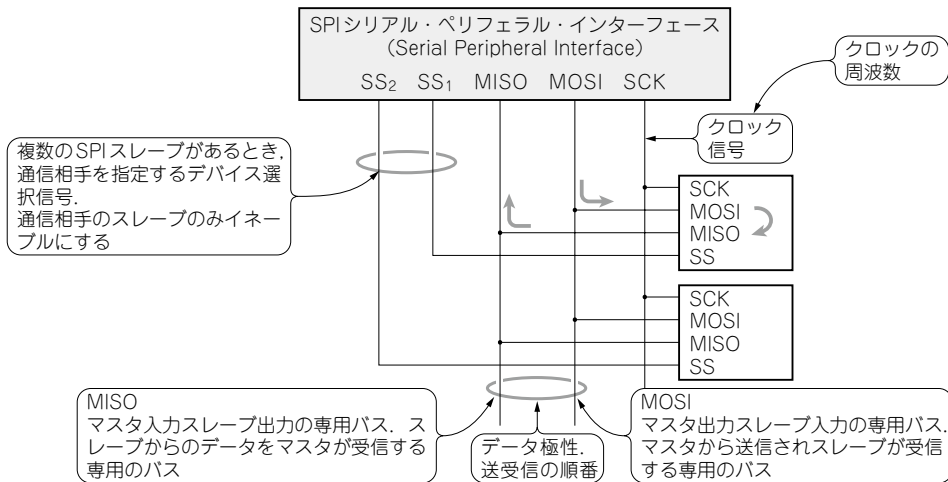


図7-2 SPI (シリアル・ペリフェラル・インターフェース) の通信手順

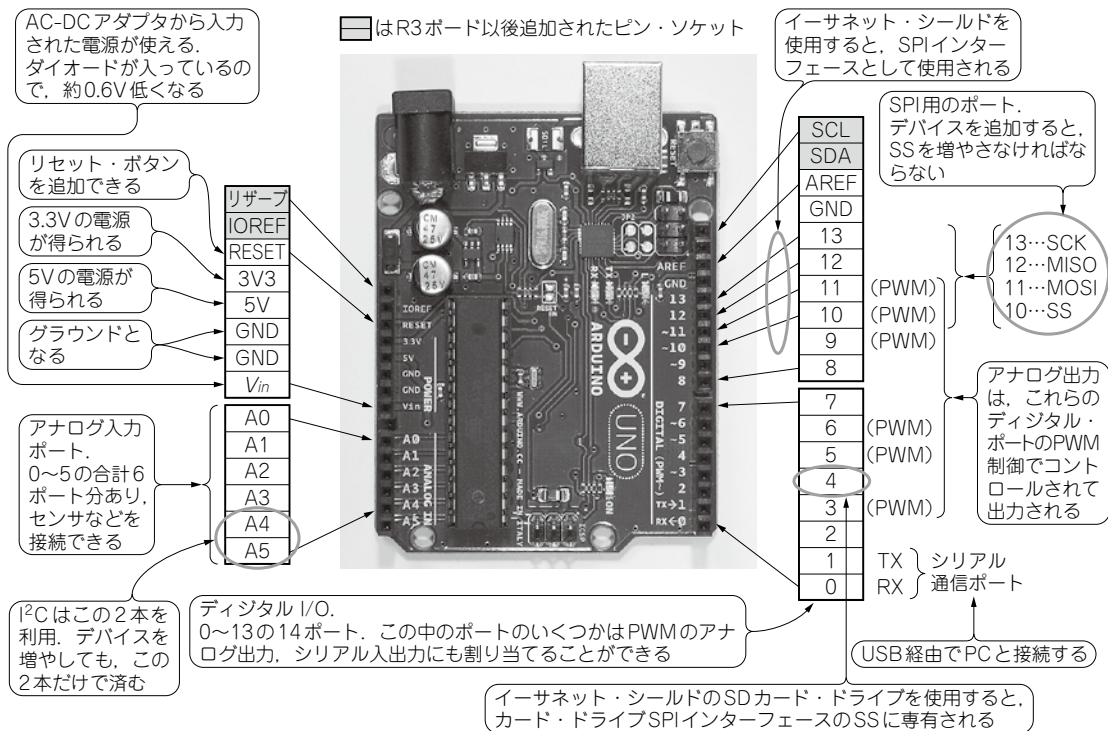


図7-3 ArduinoのSPIインターフェース

7-2 Arduino用の熱電対温度センサ (MAX6675 スイッチサイエンス)

150℃くらいまではLM35DZの半導体温度センサで測定できますが、オープン内部温度の測定や、てんぷらを揚げるときの油の温度、炎の温度の測定などを考えると200～800℃くらいの高温の温度測定が必要になります。このような高温の温度測定には熱電対がよく利用されます。

スイッチサイエンスのオリジナル商品として、このK型熱電対センサ・モジュール・キットが発売されています。このセンサ・モジュール・キットを利用すれば、オープンの内部やローストビーフの温度を測定できます。

7-2-1 K型熱電対センサ・モジュール・キット

スイッチサイエンスのK型熱電対センサ・モジュール・キットには、MAXIM社のMAX6675と基板、電源バイパス・コンデンサ、K型熱電対用コネクタ、熱電対センサがセットになっています。図7-4は、このキットを組み立ててArduinoのデジタル・ポートに差し込んだものです。

MAX6675は冷接点補償熱電対/デジタル・コンバータICで、熱電対からの信号電圧をデジタル変換してSPIのシリアル信号で送出します。Arduinoのデジタル・ポート13, 12, 11, 10でSPIの制御信号に接続します。この接続では電源の供給はデジタル・ポート9番から-電源、デジタル・ポート8番から+電源を供給しています。これはテスト時の接続が簡単になり便利ですが、電源は電源回路から供給し、あまり余裕のないデジタル・ポートは、ほかの入出力に使うほうがベターです。

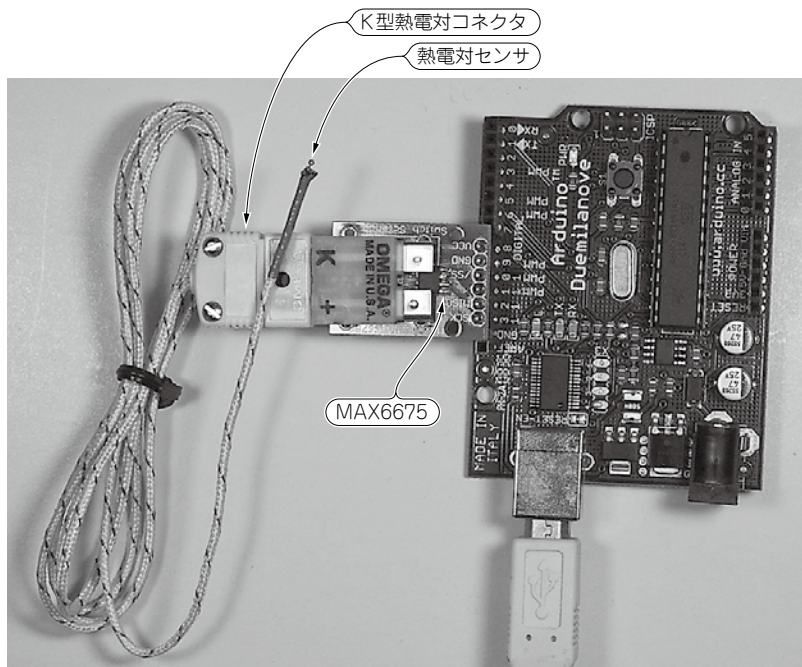


図7-4 K型熱電対センサ・モジュールを接続

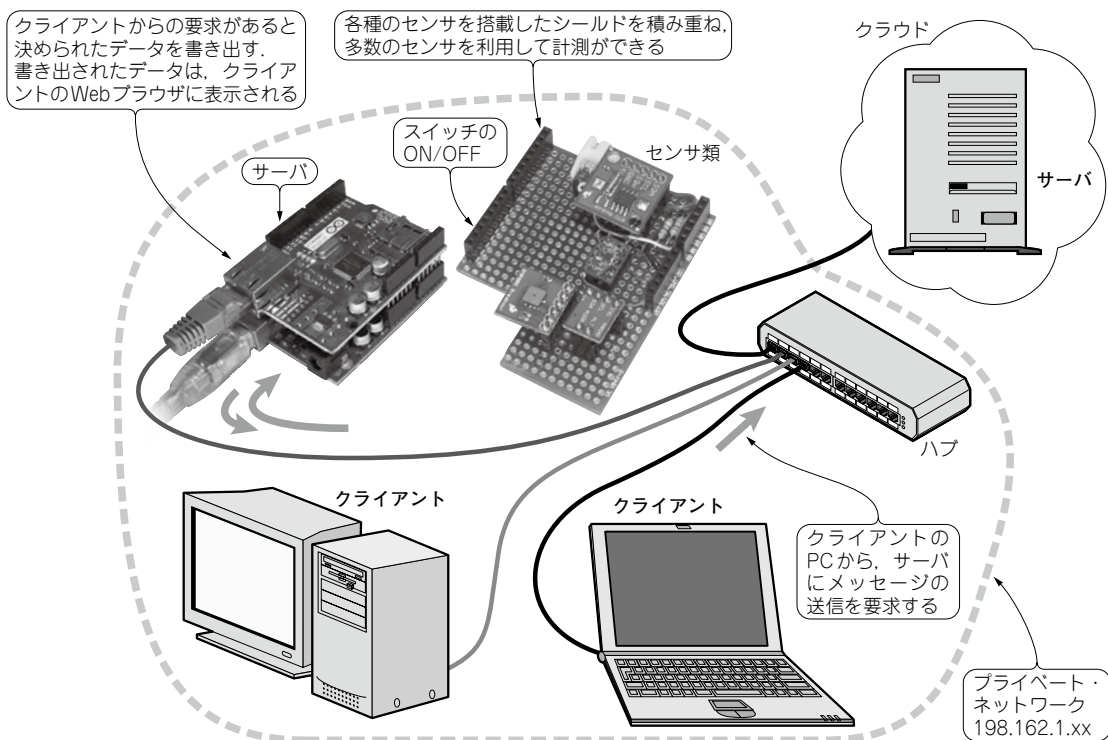


図9-2 イーサネット・シールドでArduinoがネットワークに参加できる

9-1 イーサネット・シールド

イーサネット・シールドはイーサネット制御チップとしてW5100を使用しています。このW5100はTCP/IPのプロトコル・スタックをハードウェア上に実装しています。そのため、OS側にプロトコル・スタックをもたなくてすみ、Arduinoのようなマイコンでも容易にイーサネットのネットワークに接続できるアプリケーションが実現できます。

◆ バージョンによって仕様が異なる

このイーサネット・シールドのボードは今まで何回か改訂がありました。初期のボードはSDカード・ドライブが搭載されていましたがサポート外で、そのうちSDカード・ドライブが外されて販売されていました。V5(図9-3)からはマイクロSDカード・ドライブが搭載され、SDライブラリも用意されデータの保存が容易になりました。

2011年8月現在、V6(図9-4)になり、SDカード・ドライブのインターフェースの電圧レベル変換がV5の抵抗による分圧から74LVC1G125のスリー・ステートのバッファになっています。また、V5のイーサネット・シールドでは、アナログ・ポートの0と1が10kΩで+5V電源にプルアップされています。V5のイーサネット・シールドを使用して0、1のアナログ入力ポートにセンサなどを接続するときは、この10kΩの影響を与えないセンサの出力であることを確認する必要があります。V6のイーサネット・シールドのボードでは、アナログ・ポートのプルアップはなくなりました。

[第10章]

無線対応で応用が広がる

XBeeでデータ収集

Arduinoにセンサを接続して図10-1に示すように、様々な場所の湿度や気温などをモニタできるセンサ・ネットワークを構築し、PCで離れた場所の情報をいつでも確認できるしくみを作ってみます。図10-2に示すXBeeを利用することで、Arduinoのセンサ・ステーションの無線化を図ります。これにより、有線ではつなぎにくい場所、例えば庭の気温、土中温度、湿度などの気象情報を無線でPCに送信できるようになり、利用範囲が広がります。

10-1 本章で使用するXBee

XBeeの無線モジュールは、ZigBeeのメッシュ・ネットには対応していないものです。1対1または1対Nで使用する場合は、この安価なXBeeで十分対応できます。

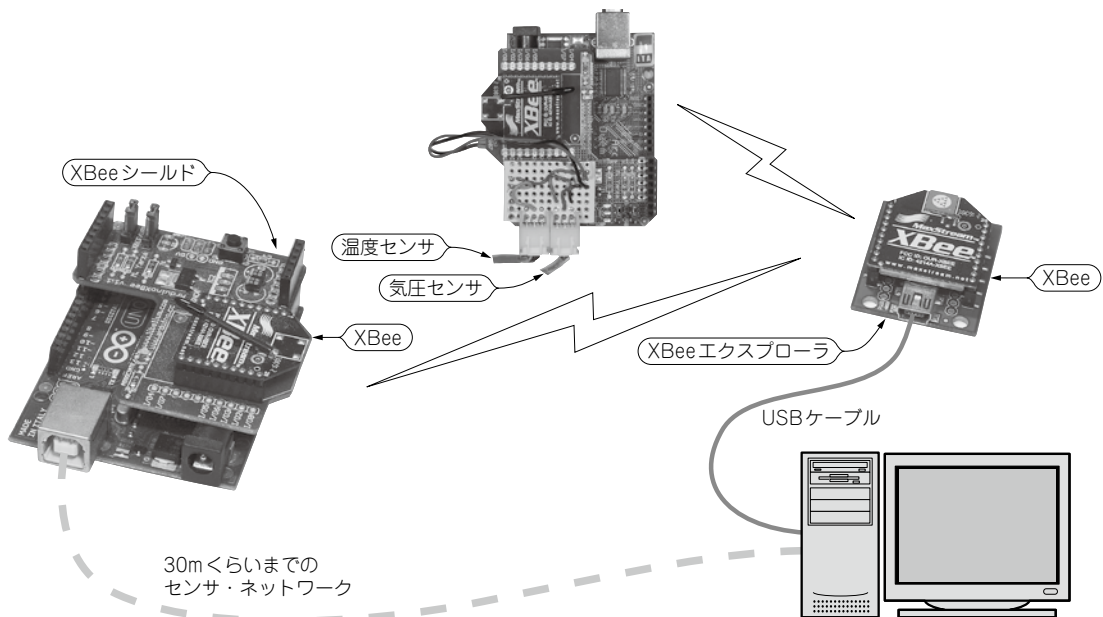


図10-1 ワイヤレス Private Area Network を XBee で作る

見本

ISBN978-4-7898-4219-8

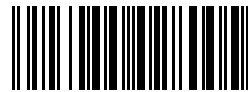
C3055 ¥2800E

CQ出版社

定価：本体2,800円（税別）



9784789842198



1923055028005

Arduinoで 計る,測る,量る

Various measuring techniques in Arduino

このPDFは, CQ出版社発売の「Arduinoで計る,測る,量る」の一部見本です.

内容・購入方法などにつきましては以下のホームページをご覧ください.

内容 <http://shop.cqpub.co.jp/hanbai//books/42/42191.htm>

購入方法 <http://www.cqpub.co.jp/hanbai/order/order.htm>