

Arduino と XBee, SQL Server による 自動無線計測システムの実験

兵庫県立西宮香風高等学校主幹教諭
松本 吉生

1. Arduino とシールド

Arduino(アルドゥイーノ)は Atmel 社が製造する RISC ベースの 8 ビットマイクロコントローラ Atmel AVR を使ったマイコンボードの一種で、デジタルとアナログの I/O ピンを利用したセンサ実験を手軽に行うことができる。この Arduino の I/O ピン配置に適合した「シールド」と呼ばれる様々な拡張ボードが販売されており、これを搭載することで Arduino の機能を拡張することができる。一般的に基板状の電子部品に付け足す形状の小型の基盤部品を「ドーターボード」と呼ぶが、Arduino と「シールド」もこれと類似の関係になる。

「シールド」には回路を自作するための「バニラシールド」やプロトタイプ作成用の「プロトシールド」といった部品素材に近いものから、DC モーターを制御できる「モータードライブシールド」、SD カードやマイクロ SD カードスロットを持つ「SD カードシールド」、RJ45 ジャックを有してイーサネットに接続できる「イーサネットシールド」など完成された機能を持つものがある。

2. XBee

XBee はディジインターナショナル社が開発、販売する無線通信モジュールで、組み込み機器の開発などに用いられる。対応する無線規格としては、Wi-Fi や独自プロトコルの DigiMesh、そして

ZigBee 規格のものなどがあり、物理的にはアンテナ形状の異なるものや、電波出力の大きさの異なるものなどいくつかの種類がある。

ZigBee は Bluetooth と同種の技術で家電向けに策定された無線通信規格で、Bluetooth よりも低速だが省電力で低コストの通信が可能であり、物理層のインターフェースには IEEE 802.15.4 が使われる。

前述のように XBee は ZigBee 規格のものだけではないが、Arduino などマイコンボードの無線化によく用いられるのは ZigBee 対応の XBee か、または ZigBee 準拠ではないが IEEE 802.15.4 で通信を行う Series 1 である。この実験では Series 1 の XBee を使っている。

XBee を利用するには、通信上の設定をするためにアダプターを介して USB でコンピュータに接続する必要がある。この実験では、スイッチサイエンス社の XBee USB アダプターを使用した。

3. X-CTU による XBee の設定

XBee のセットアップには、XBee を USB アダプターでコンピュータに接続し、ディジインターナショナル社が提供する X-CTU というアプリケーションを利用する。X-CTU はディジインターナショナル社の Web ページからダウンロードできる。

通信をするために 2 台の XBee を用意する。XBee を互いに通信させるには、出荷時に決められ



Fig.1 XBee Series 1 モジュール



Fig.2 Xbee を Xbee USB アダプターに取り付けた



Fig.3 XBeeの裏面に印刷された個別ID

た個別IDを互いに登録し、ユーザーが自由に決める共通のPAN IDを設定しなければならない。

XBeeの個別IDはX-CTUで接続するとわかるが、本体裏側にも印刷されている。XBeeをアダプターに接続すると裏側は見えなくなるので、設定の前に手元に控えておくとうい。

XBeeをUSBアダプターに挿してUSBケーブルでコンピュータに接続し、X-CTUを起動するとXBeeの設定ができる。X-CTUのPC Settingsメニューで接続されているXBeeが確認でき、通信に使われるシリアルポートの番号がわかる。Test / QueryボタンでXBeeとの通信テストができ、個別IDを取得することができる。

次にModem ConfigurationメニューでXBeeの設定値を読み出し、Networking & Securityの項目であらかじめ決めた共通のPAN IDと、相手XBeeの個別IDを登録する。自分の個別IDはSerial Number HighとSerial Number Lowで上位と下位に分かれて表示されている。相手XBeeの個別IDはDestination Address HighとDestination Address Lowに登録する。なお設定中にXBeeをコネクタか



Fig.4 X-CTUでXBeeの設定をする

ら抜き差しする際には、ピンが折れないように注意して取り扱う。

4. Arduino XBee シールド

XBeeをArduinoで使うにはシールドが必要となる。XBeeを使うためのシールドも様々なものがあり、XBeeシールドやワイヤレスシールドなどと呼ばれる。この実験では、スイッチサイエンス社のワイヤレスプロトシールドを使用する。



Fig.5 ワイヤレスプロトシールドとXBee

5. 通信テスト

通信テストのために簡単なスケッチを作りArduinoに書き込む。このテスト用スケッチは、1秒ごとに数字をカウントアップし、シリアル通信で数字を送信する。

```
int i = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.print(i);
  Serial.print("\n");
  delay(1000);
  i = i + 1;
}
```

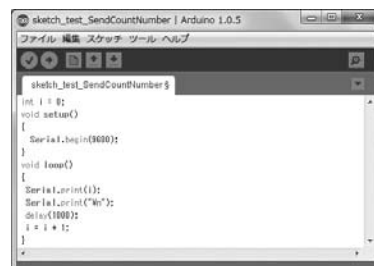


Fig.6 XBee 通信テスト用のスケッチ

Arduino にテスト用スケッチを書き込んだらコンピュータから取り外し、設定できた2台のXBeeのうち一方をワイヤレスプロトシールドに載せてArduinoに取り付ける。もう一方のXBeeはXBee USBアダプターに取り付けてコンピュータに接続する。コンピュータでX-CTUを起動してXBeeとの接続を確認し、Terminalメニューを開いておく。

Arduinoを5VのUSB電源に接続するとスケッチが起動し、XBeeから無線通信で数字のデータを送り出す。送られたデータはもう一方のXBeeが受け取り、USBアダプターからコンピュータに送信され、X-CTUで受け取ることが確認できる。

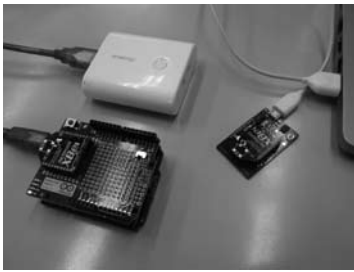


Fig.7 通信テスト中のArduinoとXBee



Fig.8 X-CTUで数字データを受け取る

6. 温度センサの値を送信するスケッチ

通信テストが成功したら、ArduinoからXBeeをシールドごと取り外し、コンピュータに接続してArduinoのスケッチを温度センサの値を読み取りシリアル通信で送信するものに書き換える。

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

void loop()
{
  Serial.print(analogRead(A0),DEC);
  Serial.print("\n");
  delay(1000);
}
```



Fig.9 LM35の出力をA0で受け取るスケッチ

スケッチを書き換えたら再びXBeeシールドを取り付け、温度センサLM35を接続し、Arduinoを外部電源に接続して動かす。コンピュータでX-CTUを起動し、XBeeで受け取るシリアルデータをモニターする。LM35を指でつまんで温めるなどすると、アナログ出力データが変化することがわかる。

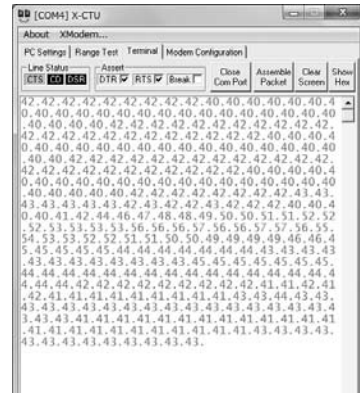


Fig.10 X-CTUで受け取った温度データ

7. シリアルデータをSQL Serverに記録する

次に受け取ったシリアルデータをSQL Serverに記録するアプリケーションを作成する。Visual StudioでC#によるWindowsアプリケーションを作り、シリアルポートからデータを受け取るイベントでデータをSQL Serverに送るコードを書く。

```
using System.IO.Ports; // 名前空間の追加

namespace GetArduinoData_NA_toFIX4
{
  public partial class Form1 : Form
  {
    SerialPort mySerialPort
    = new SerialPort("COM4");
    System.Data.SqlClient.SqlCommand sCom;
```

```

public Form1()
{
    InitializeComponent();
    mySerialPort.BaudRate = 9600;

    mySerialPort.Open();
    System.Data.SqlClient.SqlConnection scn
    = new System.Data.SqlClient.SqlConnection
    ("Data Source=OWNER-PC¥¥SQLSERVER;
    Initial Catalog=test;Integrated Security=True");
    scn.Open();

    sCom = new System.Data.SqlClient.
    SqlCommand("INSERT INTO
    LM35DATA(data) VALUES(@data)", scn);

    mySerialPort.DataReceived
    += mySerialPort_DataReceived;
}

void mySerialPort_DataReceived
(object sender, SerialDataReceivedEventArgs e)
{
    sCom.Parameters.Clear();
    sCom.Parameters.Add("@data", SqlDbType.
    Int).Value = Convert.ToInt32(mySerialPort.
    ReadLine());
    sCom.ExecuteNonQuery();
}
}

```

C#によるWindowsアプリケーションを起動すると、Windowsフォーム自体は空白のフォームが表示されるだけだが、受け取ったデータをSQL Serverに記録している。データはSQL Server Management Studioで確認できる。

8. まとめ

ディジインターナショナル社のZigBeeあるいはIEEE 802.15.4 インタフェースのXBeeは、無線出力が1mWで最大90mの屋外見通しレンジのもの

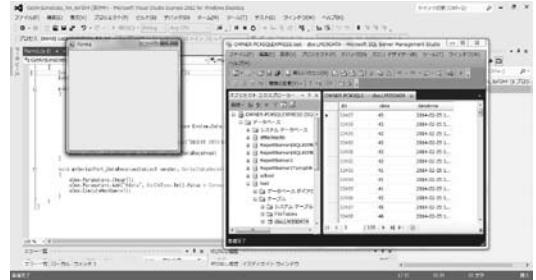


Fig.12 テストビルド画面

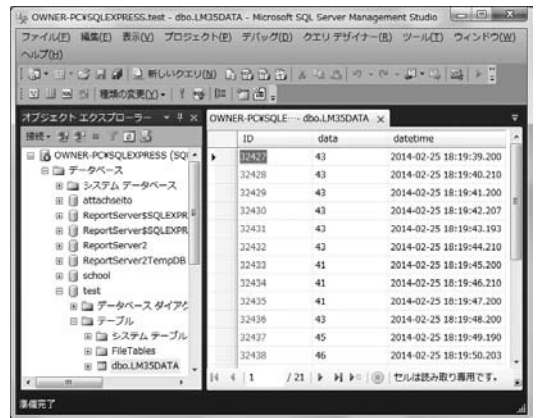


Fig.13 SQL Serverに記録されたデータ

から出力10mWで最大750mの屋外見通しレンジのものなどがある。Arduinoで使えるセンサも温度センサ以外に光センサ、気圧センサ、距離センサ、加速度センサなど様々なものがある。XBeeで無線通信を行うことにより、離れたところの計測が可能になり、SQL Serverに記録すれば計測データの管理や分析もしやすくなる。

参考URL

- 1) 「スイッチサイエンス社 - XBee をはじめてみよう」
http://mag.switch-science.com/2012/07/20/startup_xbee_s1/
 (アクセス日: 2014年2月25日)
- 2) 「ディジインターナショナル株式会社」
<http://www.digi-intl.co.jp/index.html>
 (アクセス日: 2014年2月25日)
- 3) 「IT用語辞典 e-Words - ZigBee」
<http://e-words.jp/w/ZigBee.html>
 (アクセス日: 2014年2月25日)
- 4) 「IT用語辞典 バイナリ - ZigBee」
<http://www.sophia-it.com/content/ZigBee>
 (アクセス日: 2014年2月25日)



Fig.11 Visual StudioによるC#プログラミング