

Mathematica で Arduino を利用する（2015 年度）

はじめに

情報工学部では全学科で Arduino を導入して、フィジカルコンピューティングに関連した教育を行っている。電子情報工学科ではいくつかの実験演習を通じて Arduino を取り入れてきている。

ここでは、電子情報工学実験 I の Mathematica のテーマで Arduino を利用した実習をすることを考えた。シリアルポートを利用することにより、相互にデータのやりとりが可能であるので、アイデア次第でさまざまなことを実現できる。ここではいくつかの例を通じて Mathematica で Arduino を利用する方法について述べる。

準備

1 年後期の「計算機システム II」で Arduino を利用したが、復習をすることにする。詳しくは「計算機システム II のホームページ」を参照してほしい。

1. Arduino を USB ケーブルで PC に接続し、必要なソフトがインストールされているか確認する。演習室では準備されているはずである。もしインストールされていない場合は、スタッフに依頼をおこなう。

参考までに以下はインストール関係の情報である。Arduino のホームページで最新の情報を元に、ソフトとドライバをインストールする。

- (ア) ソフトは `arduino-1.0.1-windows.zip` をインストールする。`arduino.exe` により次のような画面が立ち上がる。
- (イ) さらにドライバは、「デバイスドライバ」から「ドライバーソフトウェアの更新...」をして、先ほど展開したフォルダの中の“drivers”を指定してインストールをおこなう。

```
sketch_jul23a | Arduino 1.0.1
ファイル 編集 スケッチ ツール ヘルプ
sketch_jul23a
int ledPin = 13;

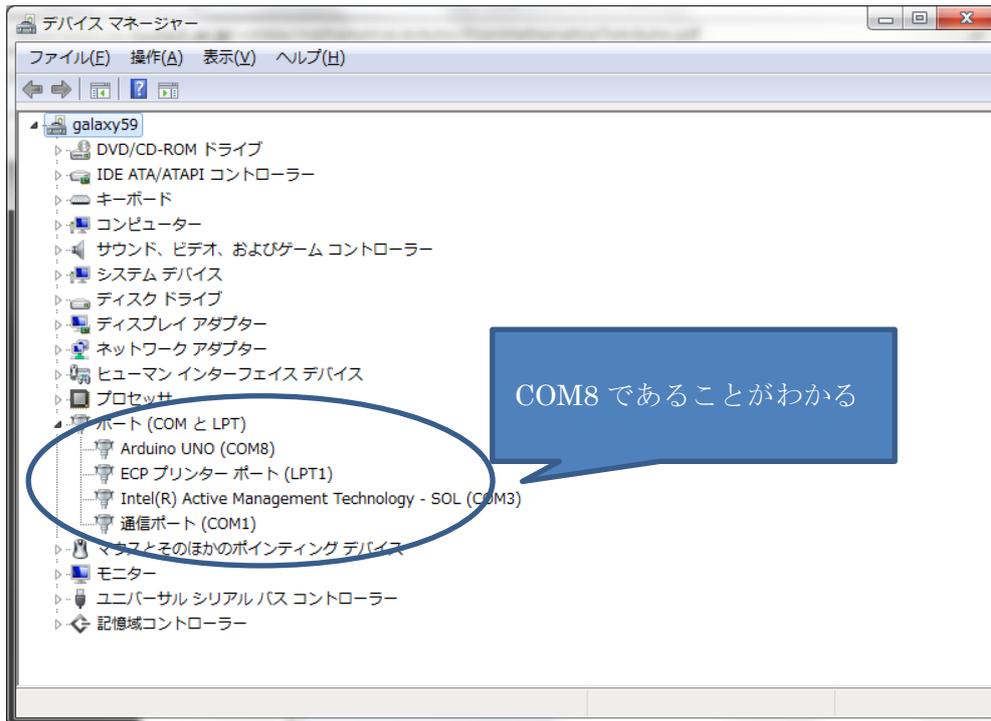
void setup()
{
  pinMode(ledPin,OUTPUT);
}

void loop()
{
  digitalWrite(ledPin,HIGH);
  delay(1000);
  digitalWrite(ledPin,LOW);
  delay(1000);
}

1 Arduino Uno on COM3
```

LED を点滅させるスケッチ（プログラム）の例





2. 試しに、「計算機システム II のホームページ」にある以下の例をコンパイルしアップロードして実行させてみよう。一秒おきに LED が点滅するはずである。初期設定で Tools→Board を確認すると“Arduino Duemilianove or Nano w/ Atmega328”になっていて、今回使用した Arduino と型が異なっていた。そのため、“Arduino UNO”にチェックしコンパイル・アップロードし直す。この例では COM ポートは COM8 であるが、実際にはそれぞれの PC により異なるので注意する。

```
int ledPin = 13;

void setup()
{
    pinMode(ledPin,OUTPUT);
}

void loop()
{
    digitalWrite(ledPin,HIGH);
    delay(1000);
    digitalWrite(ledPin,LOW);
    delay(1000);
}
```

Mathematica から Arduino を利用する

概要

Arduino ではシリアルポートを扱ういくつかの関数が用意されている。また Mathematica では.NET (ドットネット) を利用することにより、シリアルポートに簡単にアクセスすることができる。したがって、これらを組み合わせて Mathematica から Arduino を利用することが可能となる。

例題 1 : Arduino の LED を Mathematica からオンオフする

Arduino 側のプログラムは以下の通りである。



```

/*
  Mathematica から Serial port を通じて'A'が送られると
  LED を点灯して、その他は LED を消灯する。
*/

int inByte = 0;
int ledPin = 13;

void setup() {
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}

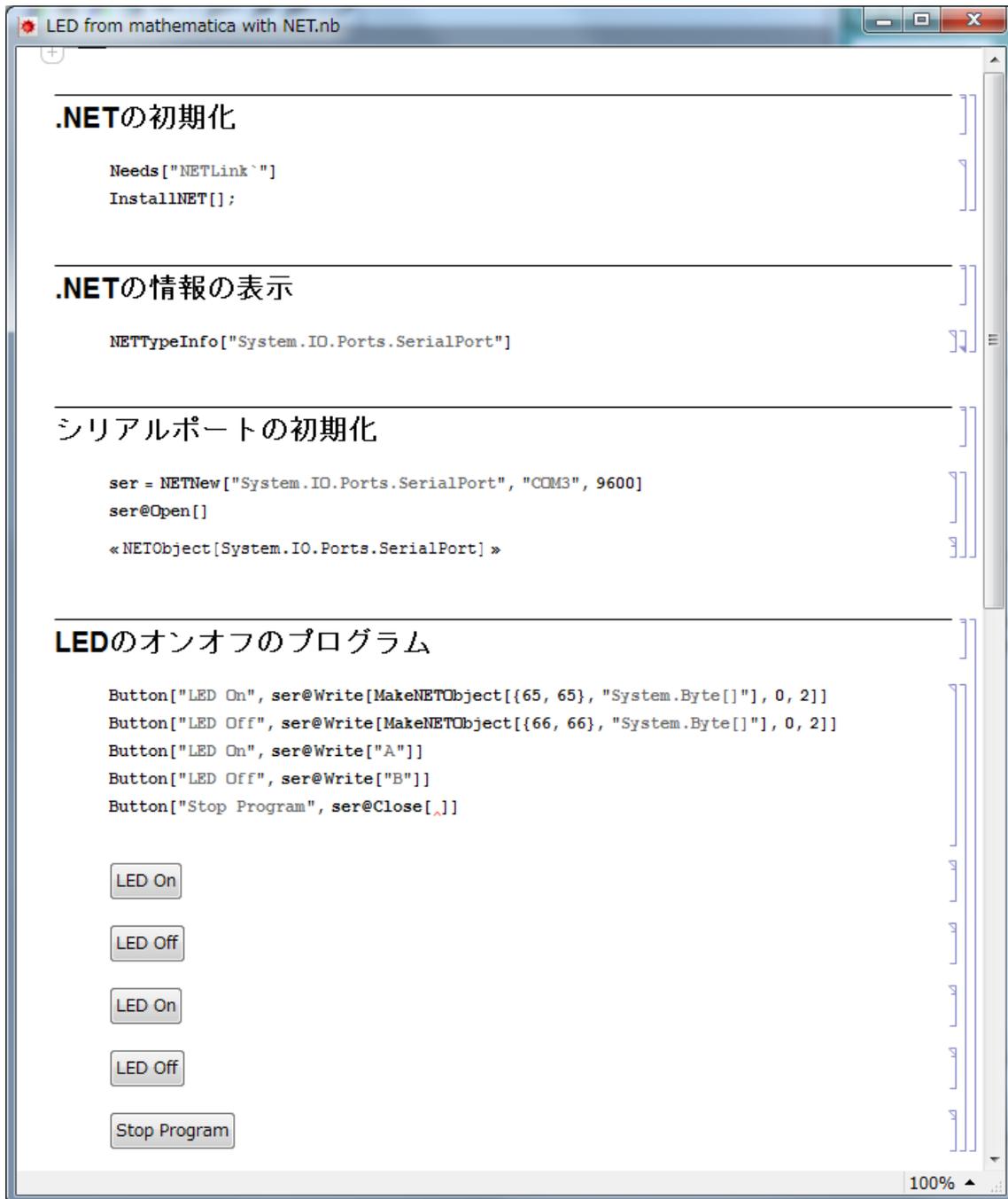
void loop() {
  if (Serial.available() > 0) {
    inByte = Serial.read();
    // シリアルから一文字よみとる
    if(inByte=='A'){ // A の時には LED を点灯する
      digitalWrite(ledPin,HIGH);
    }else{
      digitalWrite(ledPin,LOW);
    }
  }
}

```



(マイコンボードに書き込む) のボタンを押して、このプログラムをあらかじめコンパイルして Arduino にアップロードしておく。このプログラムでは `setup()` で、LED の点灯の準備をしておき、またシリアルポートの通信速度を 9600 ビット毎秒に設定している。また `loop()` では、シリアルポートが有効になったときに、一文字を読み取り、これが 'A' であったときに、LED を点灯し、それ以外の文字であったときには消灯するようにしている。

Mathematica 側のプログラム



まず以下で.NETの初期化を行う。

```
Needs["NETLink`"]
InstallNET[];
```

次に必要があれば、NETTypeInfo["System.IO.Ports.SerialPort"]を使って、情報を得ることができる。

次に、シリアルポートの初期化を行う。ここでは下記のように行って COM3 を指定している。PC によりこの番号は変わるので、確認してから変更して欲しい。

```
ser = NETNew["System.IO.Ports.SerialPort", "COM3", 9600]
ser@Open[]
```

ここで、`ser@Open[]` は `ser[Open[]]` と同じである。

さらに、ボタンを押すと、シリアルポートを通じて 'A' (ASCII コード、十進数で 65) を送るプログラムがある。

```
Button["LED On", ser@Write[MakeNETObject[{65, 65}, "System.Byte[]"], 0, 2]]
Button["LED On", ser@Write["A"]]
```

ここでは、どちらでも実現できている。しかし 127 を超える数字を送りたいときには上の方法を使う必要がある。

最後に、`ser@Close[]` を使うことにより、シリアルポートを開放する。これを行わないと、Arduino が別なプログラムを使って情報のやりとりができない。

動作確認

1. USB ケーブルで PC と Arduino を接続する。
2. Arduino にあらかじめプログラムをコンパイルしてアップロードしておく。
3. Mathematica のプログラムを準備する。
4. Mathematica のプログラムを実行し、ボタンを押すと、LED のオンオフができる。

トラブルシューティング

1. Arduino IDE において。「シリアルポート「COM3」は、ほかのアプリケーションが使用中です。シリアルポートを使っている可能性のあるアプリケーションを終了してみてください。」というエラーが表示されるときには Mathematica で `ser@Close[]` をしてシリアルポートを開放する。
2. Mathematica で `ser@Open[]` をしたとき、「ポート 'COM3' は存在しません。」というエラーが出るときには、次の方法を上から順に試してみてください。
 - Mathematica のカーネルを終了して再起動する。
具体的には Mathematica のメニューの「評価」→「カーネルの終了」→「ローカ

ル」で、終了する。そのままプログラムで Shift+Enter により実行するとカーネルが再起動する。

- Arduino を USB ケーブルから外して、リセットする。
- ログオフする。
- PC の再起動を行う。

3. 何故か、動かない。

Arduino と Mathematica の双方のプログラムがきちんと協調して動く必要があります。Mathematica で一行ずつプログラムを動かして、シリアルポート経由でデータを送ったり、読んだりして、Arduino のプログラムがきちんと動いているかどうかを確認すると、分かりやすいかも知れません。



何故か、動かない



例題 2 : Arduino のアナログ 1 の出力を Mathematica で読む

Arduino のプログラムは以下の通りである。このプログラムは参考文献で紹介されたものである。establishContact()で最初のコンタクトをおこなっている。つまり、Mathematica から文字が送られてくるまで 100 m 秒ごとに 'A'を送り続けている。文字が送られると、loop()になる。loop()では、シリアルポートが有効である間、アナログ 1 の出力値を 0—255 に変換して、それをシリアルポートを通じて Mathematica 側に送っている。

```

/*
  arduino_mathematica_example

  This code is adapted from
  http://arduino.cc/en/Tutorial/SerialCallResponse

  When started, the Arduino sends an ASCII A on the serial port until
  it receives a signal from the computer. It then reads Analog 1,
  sends a single byte on the serial port and waits for another signal
  from the computer.

  Test it with a potentiometer on A1.
*/

int sensor = 0;
int inByte = 0;

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

void loop() {
  if (Serial.available() > 0) {
    inByte = Serial.read();
    // divide sensor value by 4 to return a single byte 0-255
    sensor = analogRead(A1)/4;
    delay(15);
    Serial.write(sensor);
  }
}

void establishContact() {
  while (Serial.available() <= 0) {
    Serial.print('A');
    delay(100);
  }
}

```

Mathematica でのプログラム

```
slider from arduino with NET.nb *

.NETの初期化

Needs["NETLink`"]
InstallNET[];

シリアルポートの初期化

ser = NETNew["System.IO.Ports.SerialPort", "COM3", 9600]
ser@Open[]

単独でデータを取り込む

ser@Write["B"]
ser@ReadChar[]

79

取り込んだデータをスライダを使って動的に表示する

Slider[Dynamic[Refresh[ser@Write["B"];
ser@ReadChar[], UpdateInterval -> 0.1]], {0, 255}]

終了のときにはCloseした方がいい

ser@Close[.]

100%
```

最初の.NET とシリアルポートの初期化は例題 1 と同じなので説明を省略する。

Mathematica 側からは `ser@Write["B"]` を使って、文字をおくり、`ser@ReadChar[]` を使って、データを読み出す。この例では 79 という値を得ている。

Mathematica では `Dynamic[]` や `Slider[]` を使って、リアルタイムにデータを取り込んで、表示を変えることができる。

```
Slider[Dynamic[Refresh[ser@Write["B"];
ser@ReadChar[], UpdateInterval -> 0.1]], {0, 255}]
```

実験終了時には `ser@Close[]` を使って、シリアルポートを閉じた方がよい。

考えられる例

Mathematica は数式やデータ処理が得意であるので、Arduino から得られたデータを処理して、分かりやすく表示することが考えられる。また Mathematica では `Dynami[]` を利用して、インタラクティブな操作ができるので、これを利用して、Arduino のインターフェース部分にすることが考えられる。実際にオシロスコープを Mathematica を利用して作っている例がある。その他にも単純な例はいくらでもあるので、アイデア次第でいろいろと実現することができるだろう。

まとめ

ここでは Mathematica と Arduino を .NET を使って、シリアルポートからデータのやりとりをすることにより、Mathematica から Arduino を利用する方法について、例題を使って説明をした。Arduino を使ったことがあるのであれば、割と簡単なのでぜひ試していただきたい。

参考となるリンク先

[1] <https://leo11.cse.kyutech.ac.jp/wordpress/>

「計算機システム II ホームページ」。ここを参照すると Arduino の最初の使い方についてわかる。

[2] <http://arduino.cc/en/Guide/Windows>

「Arduino のホームページ」ここから `arduino-1.0.1-windows.zip` をダウンロードして、ソフトとドライバを整備することができる。

[3] <http://arduino.cc/playground/Interfacing/Mathematica>

「Arduino で Mathematica とリンクする方法のページ」SerialIO という Wolfram が作った、ライブラリについても触れられているが、もっと簡単に .NET を利用してリンクする方法が書かれている。この文書もこの情報を元に準備している。

[4] <http://library.wolfram.com/infocenter/MathSource/5726/>

Wolfram のサイトにある `serial port` を扱えるようにしたライブラリの紹介記事。このやり方も実現できることを確認したが、若干準備が大変なので、.NET を使った方法について述べた。

[5]

<http://williamjturkel.net/2011/12/25/connecting-arduino-to-mathematica-on-mac-os-x-with-serialio/>

Wolfram のライブラリを使って、実際に Arduino と交信する例を記述している。ただし、Mathematica は MacOSX 上である。この文書では、例題 2 についてはこの例を引用している。

[6] <http://www.youtube.com/watch?v=REXS0tS8BFI>

Mathematica と Arduino を使ってオシロスコープを実現している例が紹介されている。



それだけでは何なので

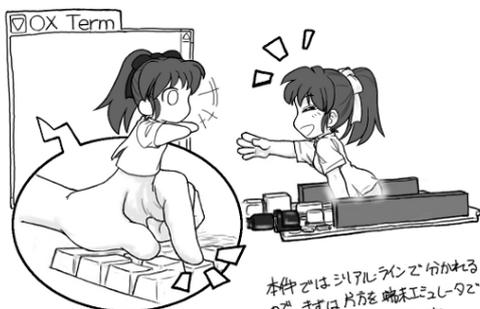


泣き落としでも何でも、相手と間違えなければ「使える」よらしい。
(使える相手はそう居ないだろうけど)

基本は「切り分け」



怪しい同志を組合せてもワケワカヌ。
小分けにして「動確」を増やして行く。

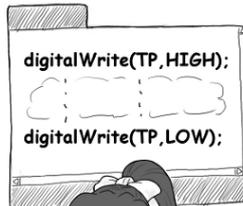
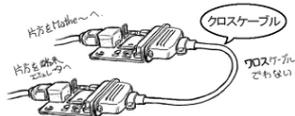


本件ではシリアルラインで分かれるので、まずは片方を端末エミュレータで相手のフリをして確認する。



ところで、
実機相手のリズレベルデバイスがほまたないそうだな
せいたく言うたらいかんいうことかな

Mathe~相手は、
COMポートを2つ作ってクロスと繋ぐ。



便利が無いならないで、
空いたポートを使って
check pointで off-on-on-offさせ、ロジックを監視、通信を確認する。
所謂 Heart Beat とか?

裏はワタシ、
すごいんです。



一つのツールで万全はないので
都度適当なのを揃える。
手札は多い方がいい。
つまり、加減はね。





さあっ!!

がんばれ

K.I.T.ie,

精進せよ!

一般人など

僅か 0.001秒で

ドン引きだっ!!