

# Scilab による Arduino の制御

花本 和也\*, 木下 健人, 大野 修一  
(広島大学)

## Control of Arduino by Scilab

Kazuya Hanamoto, Kento Kinoshita and Shuichi Ohno (hiroshima University)

### 1 はじめに

近年, 制御教育や電子工作において, Arduino と呼ばれるワンボードマイコンボードを用いることが主流となっている. Arduino は初心者でも簡単に取り組むことができる. さらに MATLAB® と呼ばれる数値解析ソフトウェアと Arduino を連携させて, 複雑な演算も簡単にできるようになっており, MATLAB はスクリプト言語として実装されているため, 低水準の操作を意識する必要がなく, 人間にとってわかりやすい言語となっている. しかし, MATLAB は有料であるためソフトの購入が必要である. そこで, 本稿では, フリーソフトの中でも MATLAB によく似た Scilab と呼ばれる数値解析ソフトウェアを利用して Arduino の制御を行う方法を紹介する.

### 2 Scilab による Arduino の制御方法

Scilab には Arduino を制御するためのツールボックス `arduino_toolbox [1]` が公開されており, 以下の手順で利用できる.

1. `arduino_toolbox` をダウンロードして回答し, Scilab のインストールフォルダ中の `contrib` フォルダに移動する.
2. ツールボックスに含まれる `toolbox_arduino_v3.ino` というファイルを Arduino software によって書き込む.

上記の手順により, コマンドライン上で Arduino 用の関数を使用することができる.

### 3 デジタル入出力

デジタル入出力の利用例を 2 つ紹介する. Arduino 上の回路図は図 1 に示す.

#### 3.1 LED の点滅

デジタル出力の例として, LED を点滅させるプログラムを図 2 に示す. このプログラムの意味を上から順に解説する. 1 行目は Arduino を接続したシリアルポートの初期設定を行うための文で, この関数に与えられるパラメータは左から順に, ボード番号, Arduino が接続された COM ポート番号, シリアル通信の通信速度 [baud] である. これは Scilab で Arduino を動かす際に必須の文である. 2 行目からはプログラム本体で, LED を点滅させる動作を while ループと呼ばれる繰り返し構造で実現している. ここで, while 文の条件式の中に `cmd_digital_in()` という Scilab の Arduino

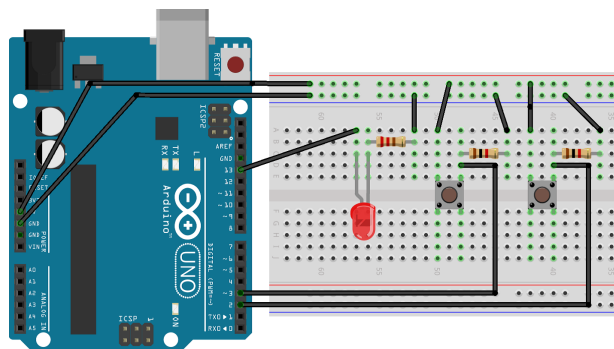


図 1 デジタル入出力の回路図

```
1 ok=open_serial(1,4,115200)
2 while cmd_digital_in(1,2)==0
3   cmd_digital_out(1,13,1);
4   sleep(1000);
5   cmd_digital_out(1,13,0);
6   sleep(1000);
7 end
8 close_serial(1)
```

図 2 LED を点灯させるプログラム

```
1 ok=open_serial(1,4,115200)
2 while cmd_digital_in(1,2)==0
3   if cmd_digital_in(1,3)==1
4     cmd_digital_out(1,13,1);
5   else
6     cmd_digital_out(1,13,0);
7   end
8 end
9 close_serial(1)
```

図 3 タスクスイッチによる LED の点灯のプログラム

用 toolbox に含まれる関数があるが, これは Arduino のピン番号を指定してそのピンへのデジタル入力を読み込む関数である. Arduino の 2 番ピンにはタスクスイッチを接続しており, この条件式の意味はタスクスイッチが押されるまで while ループを繰り返すことである. while ループの中に `cmd_digital_out()` という関数があるが, これは 2 行目と同様 Scilab の Arduino 用 toolbox にある関数で, 指定したピンへデジタル値 0 または 1 を送ることができる. LED は 13 番ピンに接続されているため, while ループの中身は 1 秒おきに LED を点滅させるというプログラムになっている. 最後の行に `close_serial(1)` とあるがこれは Arduino と接続されたシリアルポート間の通信を切断するための文で 1 行目同様必須の文である.

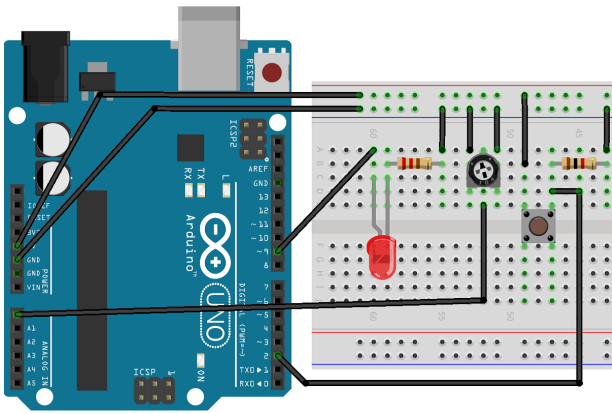


図 4 アナログ入出力の回路図

```

1 ok=open_serial(1,4,115200)
2 lightness=0;
3 hoge=0;
4 while cmd_digital_in(1,2)==0
5   if lightness == 0 then
6     hoge = 1;
7   elseif lightness == 255 then
8     hoge = -1;
9   end
10  cmd_analog_out(1,9,lightness);
11  lightness = lightness + hoge;
12  sleep(10);
13 end
14 close_serial(1)

```

図 5 LED を点灯させるプログラム

```

1 ok=open_serial(1,4,115200)
2 while cmd_digital_in(1,2)==0
3   lightness=cmd_analog_in(1,0);
4   cmd_analog_out(1,9,lightness*255/1023);
5 end
6 close_serial(1)

```

図 6 LED の明るさを調整するプログラム

### 3.2 タスクスイッチによる LED の点灯

デジタル入力の例として、タクトスイッチを押している間 LED を点灯させるプログラムを 3 に示す。このプログラムの意味を上から順に解説する。まず 1 行目でシリアルポートの初期設定を行う。2 行目からはプログラム本体の内容であり while ループを行う。while ループの中身は if 文を使って 3 番ピンへのデジタル入力が 1 ならば LED を点灯、そうでないならば LED を消灯するというプログラムになっている。最後の行で通信を切断する。

## 4 アナログ入出力

アナログ入出力の利用例を 2 つ紹介する。Arduino 上の回路図は図 4 に示す。

### 4.1 LED の点滅

アナログ出力の例として、LED を点滅させるプログラムを図 5 に示す。ここでの点滅というのは、LED がだんだん明るくなりだんだん暗くなることの点滅を意味する。このプログラムの意味を上から順に解説する。まず 1 行

目でシリアルポートの初期設定を行う。2 行目と 3 行目に lightness, hoge, という変数を宣言しており、lightness は LED の明るさの度合いを決める値が入り、また hoge は lightness の値を順繰りに変化させるために 1 または -1 の値が入る。4 行目からは while ループを行う。while ループの中身は if 文を使って lightness の値が 0、つまり LED の明るさが最小のとき変数 hoge に 1 を代入し、lightness の値が 255、つまり LED の明るさが最大のとき変数 hoge に -1 を代入する。その後 Scilab の Arduino 用 toolbox にある、指定したピンへ 0~255 までのデジタル値を 0~5 までのアナログ値として出力する関数 cmd\_analog\_out() を使用して、lightness の値を LED に出力し、lightness の値をインクリメントあるいはデクリメントするプログラムになっている。ここで注意するのは Arduino は PWM という信号方式を用いて擬似的なアナログ信号を出力として生成しており、その PWM 信号の発生させるのに 8bit のタイマを使用するため、分解能 8bit の 0~255 までの合計 256 段階で入力する必要がある。最後の行で通信を切断する。

### 4.2 可変抵抗を用いた LED の明るさの調整

アナログ入力の例として、可変抵抗の抵抗値を変化させることにより LED の明るさを調整するプログラムを図 6 に示す。このプログラムの意味を上から順に解説する。まず 1 行目でシリアルポートの初期設定を行う。2 行目からは while ループを行う。while ループの中身で Scilab の Arduino 用 toolbox にある、指定したピンへのアナログ信号を読み込む関数 cmd\_analog\_in() を用いて 0 番ピンに接続されている可変抵抗にかかっている 0~5[V] の電圧を 0~1023 のデジタル値に変換し、その値を変数 lightness に代入する。ここで注意するのは Arduino にある A/D コンバータは分解能 10bit であり、0~1023 までの合計 1024 段階で表現される。その後、lightness の値がとりうる範囲は 0~1023 であるため、0~255 までの範囲に変換した値を LED に接続されている 9 番ピンに入力し、その値に応じた 0~5[V] の電圧を出力する。最後の行で通信を切断する。

## 5 おわりに

本稿では、Arduino をフリーソフトである Scilab を使用して、Scilab のコマンドライン上で Arduino を制御する方法とその利用例について報告した。本稿の目的は Scilab でも Arduino を制御することができるというところに焦点を置いていたため、誰でも理解できるように簡単な利用例を紹介したが、これらを応用すればモータの回転数やヒーターの温度調整等、複雑な制御も実現可能である。

### 謝辞

本研究は JSPS 科研費 JP16H02921 の助成を受けたものです。

### 参考文献

- [1] <https://www.scilab.org/>