最終更新　2021.6.7

Arduino実習

奈良教育大学　薮 哲郎

# 1. Arduinoとは

　フィジカルコンピューティングという概念が2005年頃に提唱された。従来のコンピューターは入力として「キーボード、マウス」、出力として「ディスプレイ」しか使うことができなかった。フィジカルコンピューティング用のコンピュータは、電圧を入出力するための端子を持っており、センサやアクチュエータなどの電子回路を接続することができる。大きく分けると以下の2つがある。

|  |  |  |
| --- | --- | --- |
| 呼称 | マイコン | シングルボードコンピュータ |
| 代表的な製品 | Arduinomicro:bit | Raspberry Pi |
| 特徴 | * CPUの性能は低い
* OSはない
* マイコンの中で1個のプログラムが動作する
 | * パソコン1台と同等の能力を持つ
* OSを持つ。Raspberry Piの場合RaspbianというLinux系のOSが定番である
* 多数のプログラムが同時に動作する
 |

　2009年頃からフィジカルコンピューティング用のコンピューターが普及しはじめた。様々な製品があるが、マイコンとしてはArduino、シングルボードコンピュータとしてはRaspberry Piが最もポピュラーである。本実験ではArduinoを使う。

　Arduinoは定番製品であるUNOが3000円程度であり、互換品は2000円程度である。自作すれば500円程度でできる。ところが、Arduinoを含むキットが売っており、4000円弱である。今回使用するsmrazaのキットは2020.7の時点で3180円であった。Arduinoに加えて、ブレッドボード、ワイヤー、各種センサがついてこの値段であり、驚異的な価格である。2021.5の時点でArduinoのキットとしてはELEGOO, OSOYOO, Miuzei, Kumanというブランド名の製品が販売されており、smrazaのものは見あたらない。2020.7の時点でELEGOO, Kuman, smrazaの3つを購入しsmrazaを選択した。

# 2. IDE（統合開発環境）のインストール

　以下の手順で行う。

1. arduino-1.8.15-windows.zipをダウンロードする。
2. 「ダウンロード」フォルダから「ドキュメント」フォルダにコピーする。
3. 「右クリック」→「全て展開」で展開する（時間が非常にかかります）。

（注意！）

　IDEであるarduino.exeは「ドキュメント」の下に置くことを推奨する。dropboxやネットワークドライブに置くと、謎のコンパイルエラーが発生することがある。

## ドライバのインストール

　Windows 10においては、ドライバのインストールは不要である。接続すると、Microsoftのドライバがインストールされる。Windows 8.1では以下の手順が必要だと思われる。

1. 検索窓に「コントロールパネル」と入れてコントロールパネルを開く
2. 「システム」をクリック
3. 左上の「デバイスマネージャ」をクリック
4. Arduino互換機（smrazaと印刷されている）をWindowsパソコンに接続する
5. デバイスマネージャのポート（COMとLPT）の下に、USBシリアルデバイス（COM x）という項目が出現する。
6. 「右クリック」→「ドライバーの更新」→「ドライバーの検索方法は「コンピューターを参照して……」を選ぶ」→「参照」を選択する→「ドキュメント\arduino-1.8.13-windows\arduino-1.8.13\drivers」を選択する

## IDEの起動

1. arduino.exeをダブルクリックする。「WindowsによってPCが保護されました」と表示される。「詳細情報」→「実行」で実行する。
2. 初期設定では文字が小さいので「ファイル」→「環境設定」でエディタの文字の大きさを12→14 or 16に変更する。
3. 「ツール」→「ボード」でArduino UNOが選択されていることを確認する。
4. 「ツール」→「シリアルポート」でArduino UNOのポート番号を選択する

## 動作確認

　Arduino UNOの13番ポートにはボード上のオレンジ色のLEDが接続されており、HIGHにすると点灯、LOWにすると消灯する。以下のプログラムを入力し「→」のアイコンをクリックすると、オレンジ色のLEDが点滅する。

void setup() {

 // put your setup code here, to run once:

 pinMode(13, OUTPUT);

}

void loop() {

 // put your main code here, to run repeatedly:

 digitalWrite(13, HIGH);

 delay(500);

 digitalWrite(13, LOW);

 delay(500);

}

# 3. Arduinoの概要

<http://denki.nara-edu.ac.jp/~yabu/soft/arduino.html>

に備忘録がある。

## ハードウェア

　アナログ入力：A0～A5の6個

　デジタル入出力：0～13の14個（～印の端子はPWM出力可能）

　その他の端子：5 V, 3.3 V, Gnd

## ソフトウェア

　Arduino言語を用いる。Arduino言語はC言語とほぼ同じである。プログラムの構造は以下の通り。

void setup() {

 // put your setup code here, to run once:

 最初に1回だけ実行する命令を書く

}

void loop() {

 // put your main code here, to run repeatedly:

 このloopという関数が繰り返し呼ばれる。

}

## C言語の主な文法

コメント文 …… // はコメント文である。何を書いても無視される。

変数宣言 …… 変数を使う場合、宣言が必要である。変数はコンマで区切る。命令の末尾にセミコロン（;）を書く。

 int a, b;

 a = 10;

 b = 20;

if 文 …… 条件式は ( ) で囲む。「等しい」は「==」、「等しくない」は「!=」。ifが成立したときに実行する命令、成立しないときに実行する命令は { } で囲む。

 if (a == 10){

 b = 5;

 } else {

 b = 10;

 }

for 文 …… Basicは1から10まで変化させたいとき、For i = 1 to 10と書くが、Cではfor(i = 1; i <=10; i++){ } あるいは for(i = 1; i < 11; i++){ } と書く。「i++」は「i を1増やす」という意味であり、「i = i + 1」と同じである。

 for(i = 0; i < 10; i++){

 s = s + i;

 }

## Arduino言語特有の命令

 pinMode(13, OUTPUT); // pin 13 を出力用に設定

 pinMode(12, INPUT); // pin 12 を入力用に設定

 digitalWrite(13, HIGH); // pin 13 を HIGH に設定（5 V 出力）

 digitalWrite(13, LOW); // LOW に設定（0 V 出力）

 delay(500); // 500 ms何もせずに待機する

## グローバル変数

　loopからリターンし、再度呼ばれるとき、前回loopが呼ばれたときの変数の値は保証されない。グローバル変数（常に値が保持し続けられる変数）は以下のように使用する。

int a; // a はグローバル変数　setup() の手前で宣言する

void setup() {

 a = 0; // ここで初期化

 Serial.begin(9600); // シリアル通信を9600 bpsで行う

}

void loop() {

 a++;

 delay(1000);

 Serial.println(a); // シリアル通信線に出力

}

　上記のサンプルプログラムではsetup()で a = 0 を実行している。C言語では変数の宣言時に初期化することもできる。宣言の部分で int a = 0; としてもよい。

　上記のプログラムはシリアル通信線に出力している。これをパソコンで受けて表示するには、「ツール」→「シリアルモニタ」としてシリアルモニタのウィンドウを表示する。シリアルモニタウィンドウにおいて、bpsをプログラム中で宣言した9600に合わせる必要がある。これを怠ると正しく表示されない。

# 4. 課題

　以下の3つの課題を実行し、その動画を撮影してアップロードする。

　そのときに使ったプログラムをWordに貼り付け、提出する。プログラムリストは以下の設定で記述する。

・フォントはConsolas 10 pt

・行間固定値 13 pt

## Lチカ（第1課）



　1個のLEDを1秒間隔でチカチカさせる。動作チェックで使ったプログラムをそのまま使えば良い。LEDに直列に入れる電流制限抵抗は330 Ωを使う。電流制限抵抗は必ず入れること。出力端子をショートさせるとArduinoの出力端子のトランジスタが破壊される可能性がある。

## スイッチとLED（第5課）

　スイッチを押したらLEDが消える。スイッチと直列に必ず抵抗（図中 10 kΩの抵抗）を入れること。これを忘れて電源の5 Vをショートさせると、Arduinoに電力を供給しているパソコンのUSB端子がダメージを受ける可能性がある。LEDに直列に入れる電流制限抵抗も忘れない様にすること。

　サンプルプログラムは英語のコメントや変数宣言が書かれていて、難しそうに見えるが、以下と同一である。

void setup() {

 pinMode(2, INPUT);

 pinMode(13, OUTPUT);

}

void loop() {

 int a;

 a = digitalRead(2);

 if (a == LOW){

 digitalWrite(13, LOW);

 } else {

 digitalWrite(13, HIGH);

 }

}

　第5課のブレッドボードのイラストにはLEDは描かれていない。忘れているか省略されていると思われる。LEDの接続は第1課のLチカのままでよい。

## つまみ（第18課）

　サーボモータの端子は3個ある。3個の端子の意味は第17課のテキストの1ページ目に書いてある。以下の通り。

　赤 … 5 Vに接続

　茶 … Gndに接続

　橙 … Arduinoの9番ピンに接続

　サーボモータはブレッドボード用ワイヤーを利用して接続する。第17課のテキストp.4の写真を参考にすると良い。この写真では、サーボモータ側でない端子はArduinoに直接接続しているが、本実験をするときのサーボモータの接続は、第18課のテキストp.3 のイラストに従う。

#include <Servo.h> // サーボーモータを使うために Servo.h をインクルード

Servo myservo; // Servo クラスのオブジェクト myservo を作成

int potpin = 0; // ポテンショメータを接続するアナログ入力の端子番号 A0

int val;

void setup() {

 myservo.attach(9); // サーボモータ用出力は D9

}

void loop() {

 val = analogRead(potpin);

 val = map(val, 0, 1023, 0, 180); // 0-1023 の val を 0-180 に変換する

 myservo.write(val);

 delay(15);

}

以上