

# プログラミング入門教育を教わった後の Arduino の活用効果について

土肥 紳一

東京電機大学 情報環境学部

dohi@chiba.dendai.ac.jp

**概要**：プログラミング入門教育は、一般的に受講者が所有するノートPCや大学が設置している情報端末などを活用して授業が行われる。この場合、標準出力はディスプレイ、標準入力はキーボードとなるが、何か物足りない。プログラミング入門教育で学習した知識を使って、気温を測ろうとしても、暗くなったら自動的に照明を点けようとしても、できない事に気が付く。この問題の本質を解決する物として、Arduino(アルドゥイーノ)が存在する。この活用によって、プログラミング入門教育で学習した知識を基に、発光ダイオード(LED)への出力、スイッチや光センサーからの入力等を容易に行えるようになり、プログラミングに対する興味を一層喚起できることが期待できる。本論文では、2014年1月に実施したエクステンションの中で、「コンピュータプログラミング A を教わった後のハードウェア入門」を開講し Arduino を活用した。その取組みと活用効果を述べる。

## 1 はじめに

情報環境学部のプログラミング入門教育は、Java 言語を使い、テキストエディタとコマンドプロンプトを使って、白紙の状態からプログラムを入力し、コンパイル、実行を行うスクラッチプログラミングに拘っている。その科目名は、「コンピュータプログラミング A」(以下「コンプロ A」と略)である。毎年、300名近い受講者がおり、5クラスに分割している。主な内容は、手続き型の入門を学ぶ。授業の前半では、変数、分岐、繰返し、配列、プログラム分割等の基本を学習する。後半ではプログラム分割に重点を置き、引数、返却値を活用したメソッドを多く扱う流れになっている。「コンプロ A」の目的は十分に達成できているが、習得した知識を使って、気温を測ろうとしても、暗くなったら自動的に照明を点灯しようとしても、ハードウェアの制御はできない事に気が付く。

## 2 Arduino の活用

Arduino は、イタリアで開発されたマイクロコンピュータである[1]。大学での利用も始まりつつある[2]。Arduino Uno がキットとして販売されており、価格は4千円程度である。学生諸君のお小遣いでも十分に手に入れることができる。気の利いた部品が一通り揃っており、さらに Arduino の統合環境は、受講者が所有するノート PC で実行できる。統合環境は Processing に似ており、Processing の文法は Java に似ている。「コンプロ A」を教わった後の受講者は、手続き型の基本を理解しているため、Arduino の統合環境の利用は

容易である。受講者の様子を図 1 に示す。



図 1 受講者の様子

## 3 エクステンションでの実施

情報環境学部では、エクステンションと言う制度を設けている。教員がテーマを公開し、希望するテーマに学生諸君が自主的に参加する。例年、授業の補講、基礎、応用・実践、資格試験関連、就職関連のテーマが取り上げられる。開催期間は、授業が終了した後の1月から3月であり、単位の認定は無い。筆者は「コンピュータプログラミング A を教わった後のハードウェア入門」を担当した[3,4]。スケジュールと実施内容は、表 1 に示す。開講時間は、14:30~16:20(途中10分の休憩)である。受講者は20名を想定し、Arduino Uno キットの他に、テキストとして「Arduino をはじめよう 第2版」を20セット準備した[5]。その他の特記事項として、半田付けはしない事、ハードウェ

アを初めて体験する人の参加を奨励した。応募者は、ちょうど20名であった。SAやTAは居ない。

表 1 スケジュール

日程	内容
1月20日	統合環境 Arduino のインストール、LED の表示
1月21日	スイッチを使った LED の表示
1月22日	光センサーを使った LED の表示

## 4 初日の実施内容

### 4.1 統合環境のインストール

初日は、本テーマの実施について主旨説明を行い、テキスト、Arduino UNO キットの配布を行った。取り扱う部品が小さいので、紛失しないように注意を喚起した。その後、Arduino の統合環境をダウンロードし、受講者が所有しているノート PC にインストールした。受講者のノート PC は大半が Windows であったが、MacOS の受講者も数名いた。統合環境は、これら2つの OS に対応しており、特に問題は発生しなかった。

### 4.2 LED を 1 秒毎に点滅と消灯

最初に体験したプログラムを例 1 に示す。このプログラムは、13 番のデジタルポートに接続した LED を 1 秒間点灯した後、1 秒間消灯し、これを無限に繰り返すものである。使用したパーツは、LED が 1 個である。極性に注意しながら、ArduinoUNO の 13 番のデジタルポートと GND 端子に直接接続する。

```
const int LED = 13;

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

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

#### 例 1 LED が 1 秒毎に点滅するプログラム

プログラムは int 型の変数 LED をグローバル変数として定義し、13 を代入している。この変数は const が指定されているため、値を変更できない。

setup 関数の定義は、プログラムの実行開始時に 1 度だけ実行される。pinMode 関数を使って、使用するデジタルポートを出力に設定する。

loop 関数の定義は、setup 関数の実行後、無限に繰り返される。digitalWrite 関数は、指定したデジタルポートに HIGH または LOW を出力する。デジタルポートが HIGH になった時に LED を点灯し、LOW になった時に LED が消灯する。プログラムの実行結果を、物理的に光の点灯と消灯で示すことができる。Arduino は最大で毎秒 1600 万行の処理能力がある。LED の点灯直後に消灯すると、点滅が速すぎて目視できない。そこで delay 関数で点灯後と消灯後に 1000ms の遅延を入れる。

この短いプログラムには、「コンプロ A」で教わった知識、すなわち int 型、変数、変数のスコープ、メソッド(Arduino では関数)、引数、ブロック等々、多くの知識が活用されている。また、プログラム入力に関する作法は重要である。「コンプロ A」の授業では、ブロック内は一段字下げすることを徹底的に指導している。可読性の良いプログラムを記述する事は大変重要な事であり、Arduino においても同様である。

### 4.3 初日のアンケート調査結果

初日のアンケート調査結果を表 2 に示す。半田付けの経験は 68.4%であった。高校以前に全員が体験しているものと考えていたが、予想に反する結果となった。電子工作の経験者は 31.6%、Arduino を知っていた受講者は 10.5%であった。「コンプロ A」で教わった知識は、全員が役立ったと回答し、本テーマの効果が窺える。

表 2 アンケート調査結果 (初日)

質問項目	はい(%)
半田付けを行った事がありますか。	68.4
電子工作を行った事がありますか。	31.6
Arduino を知っていましたか。	10.5
Arduino のセットアップは理解できましたか。	100.0
Arduino とパソコンの関係は理解できましたか。	100.0
コンピュータプログラミング A の知識は、役立ちましたか。	100.0
LED の電極 (アノード、カソード) は、理解できましたか。	100.0

#### (受講者の感想)

- Android のアプリ開発をしていますが、ハードウェアのことについてはあまり触れたことがなかったので楽しいです。
- Arduino については、知ってはいましたが手は出していなかったもので、いい経験になっています。3日間で終わってしまうのは少し残念ですが、機会があれば自分でも購入して弄ってみたいと思いました。
- コンピュータで動かすことで少し実感がわきました。
- そもそもハードウェアってなんですか？

- ・まだ、ついていけそうです。
- ・難しいかと思っていたが、今日のところは意外と難しくなかった。
- ・面白そうな講義なので受けてよかったと思っている

感想の中には、Android のアプリを開発している受講者がおり、ハードウェアに触れることができ、興味が喚起されていることが窺えた。また、Arduino の存在は知っていたものの、体験する機会が無かったとの指摘があり、良い体験の機会となっている事が窺えた。

#### 4.4 初日の受講者の様子

初日は、統合環境のインストールと LED の表示が主な内容であり、ハードウェアを扱う上で難しいものは無かった。LED は、極性のある部品である。キットの中には、赤、青、緑の 3 色が用意されている。非常に小さな部品であるが、想像以上に明るく輝く事に興味深々であった。極性(アノードとカソード)の見分け方は、リード線の長い方がアノード(陽極)、短い方がカソード(陰極)になる。別な見分け方として、発光ダイオードのカソード側は少し欠けており、また内部の電極が大きい方がカソードである。「コンプロ A」では気が付かない、LED 固有の特徴を受講者は体験できた。

### 5 二日目の実施内容

#### 5.1 スイッチを使った LED の点灯と消灯

二日目は、スイッチを押すと LED が点灯し、スイッチを離すと LED が消灯する例から始めた。使用したパーツは、LED が 1 個、10KΩ の抵抗が 1 個、スイッチが 1 個、ジャンパー線が 3 本である。LED の接続は、4.2 節と同じである。その他のパーツはブレッドボード上で接続する。実体配線図は省略した。プログラムは例 2 に示す。

```
const int LED = 13;
const int BUTTON = 7;
int value = 0;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

void loop(){
  value = digitalRead(BUTTON);
  if(value == HIGH){
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

例 2 スイッチを使った LED の点灯と消灯

スイッチは 7 番のデジタルポートへ接続し、その状態を digitalRead 関数で読み取り、int 型のローカル変数 value へ代入する。if 文を使って value の値が HIGH に等しければ 13 番のデジタルポートへ HIGH を出力し、value の値が LOW に等しければデジタルポートへ LOW を出力する。

「コンプロ A」の授業では、実行時のプログラムへの入力にはキーボードを活用する。例えば気温を入力し真冬日を判定したり、if 文を複数組み合わせることによって点数に応じた成績評価を判定する例を取り上げている。Arduino ではスイッチの状態を if 文で判定できる事、デジタルポートからの入力は HIGH もしくは LOW の 2 つしかない事を体験した。

#### 5.2 トグルスイッチ機能の実装

次の例はスイッチを 1 回押すと LED が点灯し、もう 1 回押すと消灯する、すなわちトグルスイッチの機能を実装する。使用したパーツは、5.1 節と同じである。この例には「コンプロ A」では体験できない、重要な問題がたくさん隠されている。プログラムは例 3 に示す。

```
const int LED = 13;
const int BUTTON = 7;
int value = 0;
int status = 0;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

void loop(){
  value = digitalRead(BUTTON);
  if(value == HIGH){
    status = 1 - status;
  }
  if(status == 1){
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

例 3 トグルスイッチ機能の実装

スイッチの状態を digitalRead 関数で検出する仕組みは例 2 と同じである。トグルスイッチの機能を実装するために int 型の変数 status をグローバル変数として宣言し、初期値は 0 を代入しておく。変数 status は、スイッチが 1 回押されると 1 になり、もう一度押されると 0 になる。この仕組みは status = 1 - status;で行っている。例 3 は「コンプロ A」を教わった後の考え方として、非常に

素直なプログラムである。

受講者は上手く行くと信じて Arduino で実行するが、不安定な動作を体験する。主な原因は、Arduino が最大で毎秒 1600 万行の処理能力があり、スイッチを押している間に、value が何度も HIGH になり、この結果 if 文が何度も実行されるためである。このイメージを図 2 の ● で示した。横軸が時間を示しており、スイッチが押されたことによって波形が立ち上がり、この間、if 文が何度も実行されてしまう。

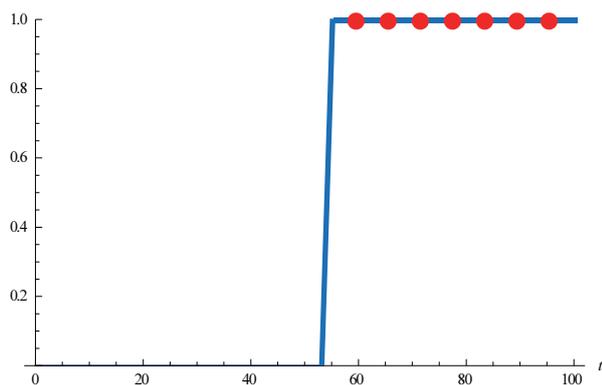


図 2 スイッチを押した時のイメージ

### 5.3 トグルスイッチの実装方法の改善

5.2 節の問題の解決策は、スイッチが押された瞬間を検出する事である。int 型の変数 old\_value をグローバル変数として定義し、初期値は 0 を代入しておく。この変数は、value よりも 1 つ前の値を保存している。プログラムを例 4 に示す。

```
const int LED = 13;
const int BUTTON = 7;
int value = 0;
int old_value = 0;
int status = 0;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

void loop(){
  value = digitalRead(BUTTON);
  if((old_value == LOW) && (value == HIGH)){
    status = 1 - status;
  }
  old_value = value;
  if(status == 1){
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

例 4 トグルスイッチの実装方法の改善

if 文で old\_value が LOW に等しく、かつ value が HIGH に等しい事を if 文の条件式に記述することによって、スイッチが押された瞬間を検出できる。この時、status の値を変更する。「コンプロ A」の授業では、点数に応じた成績評価を行うプログラムの中で、論理 AND 演算子 && を活用している。Arduino の例ではスイッチが押された瞬間を検出するために、&& を活用できる事を体験した。以上の対策により、以前よりは改善されるものの、時々不安定な動作が発生する事を、さらに体験する。

### 5.4 バウンスの対策

5.3 節の主な原因は、スイッチが押された瞬間に、スイッチ内部の接点が短時間に振動し、ON と OFF を繰り返す現象、すなわちバウンスが発生するためである。バウンスの発生によって old\_value が LOW に等しく、かつ value が HIGH に等しい瞬間を if 文が何度も検出し、不安定な動作をする。このイメージを図 3 の ● で示した。

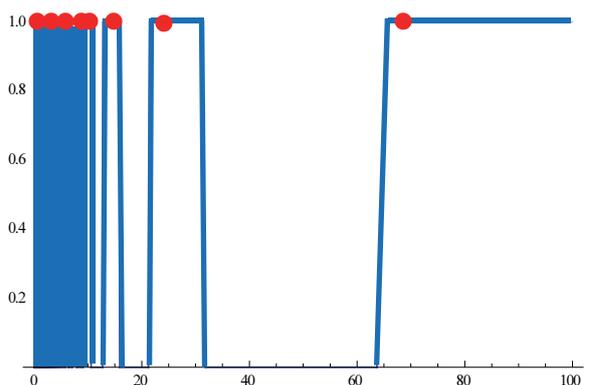


図 3 バウンスのイメージ

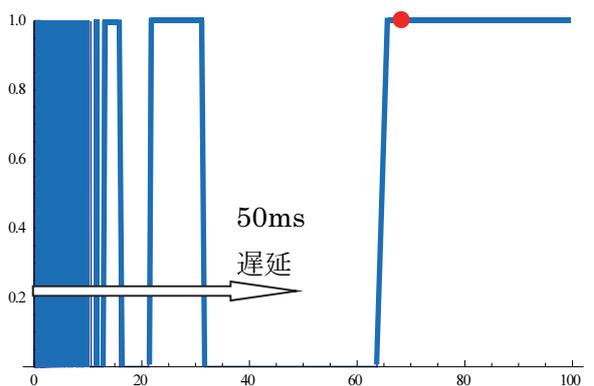


図 4 遅延によるバウンス対策のイメージ

バウンスの対策は、スイッチが押された瞬間を検出した時に、バウンスが治まるまで遅延を入れる事である。delay 関数を使って 50ms 程度の遅延を入れ、やっと安定した動作になる。改善のイメージを図 4 の ● で示した。例 5 は、loop

関数内の if 文のみを示した。

```
if((old_value == LOW) && (value == HIGH)){
  status = 1 - status;
  delay(50);
}
```

#### 例 5 バウンシングの対策

バウンシングはハードウェア固有の問題であり、「コンプロ A」の授業では出て来ない。しかし、スイッチが押された瞬間の検出やバウンシングの対応策は、ソフトウェアで行える事が興味深い。

### 5.5 二日目のアンケート調査結果

二日目のアンケート調査結果を表 3 に示す。抵抗のカラーコードの読み方が理解できない人、バウンシングの意味と対策について、理解できない人が若干発生した。

表 3 アンケート調査結果（二日目）

質問項目	はい(%)
const の意味は、理解できましたか。	100.0
変数のスコープは、理解できましたか。	100.0
ブレッドボードの仕組みは、理解できましたか。	100.0
抵抗のカラーコードの読み方は、理解できましたか。	94.7
ジャンパは、理解できましたか。	100.0
モーメンタリ型プッシュボタンスイッチ(タクトスイッチ)は、理解できましたか。	100.0
if 文でスイッチの状態を判断する仕組みは、理解できましたか。	100.0
バウンシングの意味は、理解できましたか。	94.7
バウンシングの対策に delay 関数を使う方法は、理解できましたか。	89.5

#### (受講者の感想)

- ・どうにか、ついて行けてます。
- ・バウンシングというものを初めて知った。面白い現象だと思う。
- ・思うように動かなかったので難しかったが、今日の内容はとても面白かった。
- ・少し難しかった
- ・少し難易度は上がってきましたが、スイッチのバウンシングは制御プログラムを作るうえで気を付けないといけないということは理解しました。
- ・抵抗のカラーコードの読み方を初めて知りました。あと、delay が 10~50 ミリ秒とありましたが、何度か試行してみて、50 よりも大きい値が必要だと感じました。

バウンシングの存在を初めて知ったとの感想が多かった。また、delay 関数の引数の値を大きくしないと正しく動作しないことが指摘されているが、バウンシング以外に接触不良も考えられる。

### 5.6 二日目の受講者の様子

ブレッドボードを使った配線は、半田付けを行わなくて良い事は便利である。一方、接続の不具合による接触不良が発生した。特に、スイッチの不具合が目立った。具体的にはブレッドボードに差し込んでも、浮き上がってくる事が多く、扱う部品が小さいため、受講者の力加減や器用さが求められた。ジャンパーに使用する配線は単線を使っているが、ブレッドボードに抜き差ししているうちに単線が折れ、ブレッドボードに刺さったままになる事態も生じた。接触不良に関する原因の究明は、手間取った。

ブレッドボード上の配線は、電気的な接続が正しければ、色々な実体配線が存在する。この点は、同じ実行結果を得るためのプログラムは何通りかの記述方法がある事と同じである。自由度があることに、戸惑う者がいた。抵抗はカラーコードによって抵抗値が表現されている。初めてカラーコードの読み方を知る者がいた。

書画カメラのズーム機能は、小さな部品を拡大して示すことができ、大変重宝した。またテキストの上に、ブレッドボード、ArduinoUNO を乗せ、テキストの実体配線図との関係を説明する工夫を行った。書画カメラ上の様子を図 5 に示す。

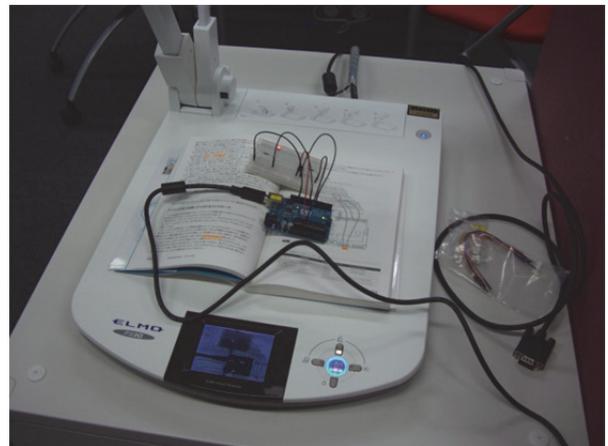


図 5 書画カメラの様子

## 6 三日目の実施内容

### 6.1 LED の明るさを徐々に変更

三日目は、LED の明るさを徐々に変更する、フェードインとフェードアウトを行う例から開始した。使用したパーツは、LED が 1 個、270Ω の抵抗が 1 個、ジャンパー線が 2 本である。これらをブレッドボード上で接続する。実体配線図は省略した。Arduino には、パルス幅変調(PWM Pulse

Width Modulation)の機能を利用できるデジタルポートがあり、analogWrite 関数を利用する事によって利用できる。フェードインとフェードアウトを行うプログラムを例 6 に示す。

```
const int LED = 9;

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

void loop(){
  for(int i = 0; i < 255; i++){
    analogWrite(LED, i);
    delay(10);
  }
  for(int i = 0; i < 255; i++){
    analogWrite(LED, 256 - (i + 1));
    delay(10);
  }
}
```

#### 例 6 LED のフェードインフェードアウト

このプログラムは、PWM を利用できる 9 番のデジタルポートに接続した LED を 0 から 255 段階の数字で、明るさを変更する。loop 関数の定義の中では、2 組の for 文を使って、0,1,2...254 と 255,254...,2,1 の整数を作り出し、その整数を analogWrite 関数の引数に指定する。delay 関数では 10ms の遅延を入れ、フェードインとフェードアウトが徐々に変化することを目視できるようにしている。このような機能は、受講者のノート PC がスリープ状態に入った時に、LED が同様の表示を行う機種がある。偶然、多くの受講者の機種がこの機能を搭載しており、興味を喚起した。

「コンプロ A」の授業でも繰り返しは for 文を重点的に学習する。例えば 1 から 10 まで 1 ずつ増加するプログラムや、逆に 10 から 1 まで 1 ずつ減少するプログラムである。for 文の記述に関して特に拘っている事は、10 回の繰り返しに対して、制御変数 i の初期値は必ず 0 にする事である。これによって for 文の記述は for(int i = 0; i < 10; i++)に統一でき、繰り返しの回数を変更する場合は、10 を他の数字や変数に置き換えるだけで済む。例 6 では 2 つ目の for 文で 255,254...,2,1 の数を生成しているが、for(int i = 0; i < 255; i++)の部分は同じである。数が順に減る場合も、制御変数 i が増える中で減る仕組みを記述し、プログラムの可読性を重視している。もちろん、実行速度を競う場合は、他の記述方法を検討する必要がある。

## 6.2 CdS による LED の点灯時間の変更

Arduino にはアナログ入力のポートが装備されている。このポートを活用する事によって、LED の点灯を制御することができる。接続するセンサーは硫化カドミウムセル(CdS)を利用する。CdS は明るくなると抵抗値が減少し、暗くなると抵抗値が上昇する。この事を利用すると室内の明るさに応じて、LED の輝度を変化できる。CdS 以外に使用したパーツは、LED が 1 個、10KΩの抵抗が 1 個、ジャンパー線が 3 本である。これらをブレッドボード上で接続する。実体配線図は省略した。CdS による LED の明るさを変更するプログラムを例 7 に示す。

```
const int LED = 13;
const int CDS = 0;
int value = 0;

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

void loop(){
  value = analogRead(CDS);
  digitalWrite(LED, HIGH);
  delay(value);
  digitalWrite(LED, LOW);
  delay(value);
}
```

#### 例 7 明るさで LED の点滅時間を変更

このプログラムは、0 番のアナログポートに接続した CdS の抵抗値を analogRead 関数によって 0 から 1023 の整数として取り出す。取り出した整数は int 型の変数 value に代入し、この変数を delay 関数の遅延時間として利用する。

受講者は CdS を使うことによって、明るさを 0 から 1023 の整数に変換できる事を初めて知る。キーボードからの入力とは異なり、リアルタイムに次から次へと数値を得る事を体験した。

## 6.3 CdS による光スイッチの実現

CdS から入力された数値を analogWrite 関数の引数に利用すると、光スイッチを実現できる。13 番のデジタルポートは 128 以上の整数になると HIGH になり LED が点灯し、128 を下回ると LOW になり LED が消灯する。このプログラムを例 8 に示した。なお value の値は 0 から 1023 の整数になり、0 から 255 の整数に変換するために 4 で割っている。loop 関数以外は、例 7 と同じため省略した。使用したパーツは、6.2 節と同じであ

る。同じハードウェアを利用しているが、プログラムを書き換えるだけで光スイッチを実現できた。

```
void loop(){
  value = analogRead(CDS);
  analogWrite(LED, value / 4);
  delay(10);
}
```

例 8 光スイッチの実現

## 6.4 コンソールへの表示

Arduino は、ノート PC と USB ケーブルで接続し、Serial.begin 関数で通信速度を設定する事によってアナログポートの状態をモニタできる。このプログラムを例 9 に示した。

```
const int CDS = 0;
int value = 0;

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

void loop(){
  value = analogRead(CDS);
  Serial.println(value);
  delay(100);
}
```

例 9 シリアルモニタへの表示

6.3 節のハードウェアを活用し、CdS から入力された数値を Serial.println 関数の引数に指定し、Arduino の統合環境でシリアルモニタを起動すると、明るさの変化によって数値が変わる様子を観測できるようになる。この様子を図 6 に示す。

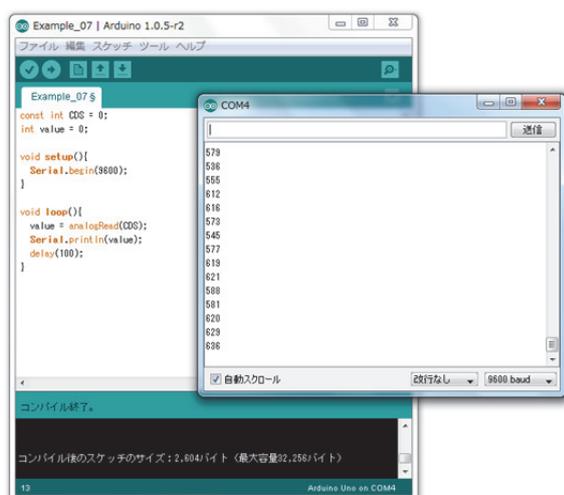


図 6 シリアルモニターの様子

## 6.5 三日目のアンケート調査結果

三日目のアンケート調査結果を表 4 に示す。アナログの入出力、シリアル通信について若干理解できない人が出た。また、MOSFET は、説明だけに止めた為、88.9%の理解に止まった。

表 4 アンケート調査結果 (三日目)

質問項目	はい(%)
パルス幅変調(PWM)は、理解できましたか。	100.0
LED のフェードインとフェードアウトの仕組みは、理解できましたか。	100.0
光センサ(CdS セル)の働きは理解できましたか。	100.0
アナログ出力(analogWrite)は理解できましたか。	94.4
アナログ入力(analogRead)は理解できましたか。	94.4
シリアル通信は、理解できましたか。	94.4
MOSFET の働きは、理解できましたか。	88.9
今後 Arduino を購入し、ハードウェアの工作を行いますか。	50.0

### (受講者の感想)

- ・3回の講義でしたが、新しく学ぶことは多かったです。シリアル通信の知識を使って、今後色々と遊べそうです。
- ・いつか購入して弄ってみたいと思いました。
- ・このエクステンションプログラムで色々なことを学ぶことができた。エクステンションだけでなく普通の授業として、プログラミングとハードの関係の授業を開講してほしいと思った。
- ・このエクステンションを通してハードウェアとプログラミングの関係性が実感できた。
- ・シリアルモニタがとても好み
- ・ハードウェアと GUI を連結させた、簡単なゲームみたいなものを作成するような授業であれば、もっと面白かったと思います。
- ・ハードウェアの仕組みがわかり、ハードウェア特有の問題があることが初めて知り、勉強になりました。
- ・楽しかったです!!
- ・今回のプログラムで Arduino のことについて少し学ぶことができたのと、ハードウェアに対する興味が一段と高まったのでよかったです。今後も Arduino について、もっと学びたいと思いました。
- ・前よりも、プログラミングに対する自信も知識もつけることができました。
- ・面白い講義だった。受講してよかったと思っている。ハードウェアの知識も少しは身についたと思う

普段の授業でも、プログラミングとハードウェアの関係について授業を行って欲しいとの要望があった。Arduino を購入し、ハードウェアの工作を行うと回答した割合が 50%になった事は、本テーマの効果を反映していると考えられる。

## 6.6 受講者の様子

Arduino でプログラムが起動すると、時々刻々送られてくる数値データが、手で CdS を隠すこと

によって変化する事に、受講者は興味深々であった。「コンプロ A」では、プログラムへの入力にはキーボードを使うが、リアルタイムにデータが入力され、時々刻々変化する事は受講者にとって新鮮な様子であった。大きな電力が必要な装置を動かす場合は、USB で供給される電流では不足する。この場合は別に電源を準備し、これを制御するために MOSFET(metal-oxide-semiconductor field-effect transistor)を使う。大きな電力を制御しながら物を動かす仕組みは説明のみとした。また CdS は、素子を手で少し隠しただけでは光を受光するため、ラインマーカのキャップを使って素子を隠すと、顕著に反応が現れた。その他、ハードウェアの特性に合わせて光スイッチの感度の微調整は、プログラムで行える事を体験できた。

## 7 受講者のモチベーションの分析

「コンプロ A」の授業では、受講者のモチベーションの向上を目的に SIEM(School of Information Environment : ジーム)を実践している[6]。これを活用すると、受講者のモチベーションを測定できる。モチベーションの測定は、初日と三日目に行った。授業毎のアンケート調査項目の中に、重要度「ハードウェアを学習することは重要だと思いますか」、期待度「もっとハードウェアの知識や技術を高めたいと思いますか」の設問項目を設け、5 段階のリッカート尺度で回答を求め、その積の平均で 1~25 に数値化する。

前期(初日)のモチベーションは 21.3、後期(三日目)は 20.3 であった。残念ながら前期から後期にかけて 1.0 低下した。

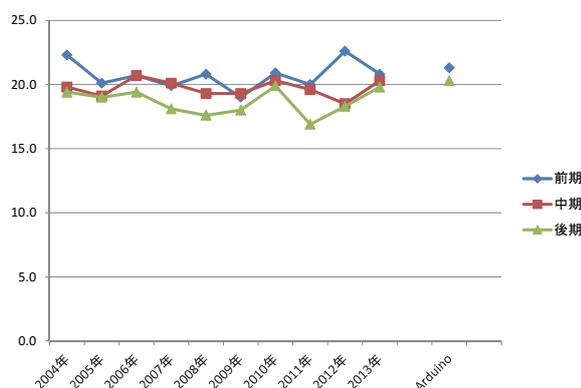


図 7 モチベーションの様子

「コンプロ A」では、授業の前期、中期、後期の 3 回に分けてモチベーションを測定している。2004 年から 2013 年までのモチベーション推移と

今回実施したテーマのモチベーションの関係を図 7 に示した。グラフ上のラベルが Arduino になっているものが、今回測定した結果である。単年度での開催のため、グラフ上では点として表示される。「コンプロ A」と異なる点は、エクステンションの実施形態が集中講義形式である事、また前期から後期まで期間が短い事である。「コンプロ A」と比較すると、前期、後期共に高い値であることが分かった。エクステンションに参加する受講者の潜在的なモチベーションの高さが窺える。

## 8 おわりに

プログラミング入門教育を教わった後の Arduino の活用効果について述べた。「コンプロ A」を教わった知識を活用する上で、Arduino の極めてシンプルな活用が効果的である事が分かった。if 文、for 文、変数、変数のスコープ等の知識がハードウェアを活用する上でどのように役立つのかを、体験を通じて学ぶことができた。さらに「コンプロ A」では体験できないハードウェア固有の問題であるバウンシングに遭遇し、その解決策はソフトウェアで行える事を学んだ。

受講者は、今後、各自の専門に応じてハードウェアに関する専門科目を学ぶ事になる。「コンプロ A」との係わりを理解し、興味が深まることを期待したい。2014 年度も同様にエクステンションで開催する予定であり、LED アレイ等を新しく導入する計画である。今後は、このような取り組みを継続しながら、プログラミング入門教育に対する一層のモチベーションの向上を目指したい。

## 参考文献

- [1]Arduino をはじめようキット,  
<http://www.switch-science.com/catalog/181/>
- [2]Arduino を用いたタイルプログラミング教材の開発, 中川洋, 中西通雄, コンピュータ利用教育学会, PC カンファレンス講演論文集, p172-p175, 2014
- [3]コンピュータプログラミング A を教わった後のハードウェア入門, エクステンション,  
<http://dohi.chiba.dendai.ac.jp/~dohi/ext-arduino>
- [4]エクステンションでのコンピュータプログラミング A を教わった後のハードウェア入門, 土肥紳一, コンピュータ利用教育学会, PC カンファレンス講演論文集, p248-p251, 2014
- [5]Arduino をはじめよう 第 2 版, オライリー
- [6]SIEM を導入したプログラミング入門教育におけるペンタブレットの活用, 土肥紳一, 宮川 治, 今野紀子, 大学 ICT 推進協議会, 年次大会講演会講演論文集, p399-p406, 2011