



技術科教育におけるステッピングモーター教材の開発

メタデータ	言語: Japanese 出版者: 公開日: 2021-04-07 キーワード (Ja): キーワード (En): 作成者: 秋葉, 治克 メールアドレス: 所属:
URL	https://doi.org/10.32150/00006993

技術科教育におけるステッピングモーター教材の開発

秋 葉 治 克

北海道教育大学旭川校電気研究室

Development of the Stepping Motor Teaching Materials in Technology Education

AKIBA Haruyoshi

Electric Laboratory, Asahikawa Campus, Hokkaido University of Education, Asahikawa 070-8621

概 要

In this paper, the stepping motor teaching materials which united the electric region and the information domain for the student of the department of technology were developed. Software used the Stepper library attached to Arduino IDE. In the electric region, the unipolar and bipolar stepping motor driver circuits for controlling by Arduino was manufactured. In the information domain, the programming education using serial communication, an analog input, a photosensor and an arm robot was developed. Thereby, in both sides of hardware and software, I think that students can understand the feature of a stepping motor.

1. まえがき

ステッピングモーターは中学校技術・家庭科の教科書¹⁾でも取り上げられており、利用機器の例と特徴および構造について簡単に説明がなされている。ステッピングモータ^{2), 3)}は、一般にステッパまたはパルスモーターと呼ばれているが、このモーターの最大の特徴はパルスによってデジタル的に制御できるという点にある。これはフィードバックなしで決められた角度で回転し、高精度で停止することができる。また、停止時においても他のモーターとは異なり、大きな静止トルクを持っている。

本研究では、技術科の学生を対象として電気領域と情報領域を融合した、Arduinoで制御するステッピングモーター教材を開発した。Arduinoでステッピングモーターを回す事例は数多くある^{4), 5), 6), 7)}。しかし、これらの事例ではAllegro社のステッピングモータードライバー A4988を搭載したモジュールを使用しており、その内部の回路動作を理解することは難しい。そこで筆者は、ユニポーラおよびバイポーラステッピングモータードライバーを自作することにより電気回路づくりを学び、ソフトウェアとして Arduino IDEを使用することによりプログラミングを学ぶことを目的とし、授業例としてシリアル通信、アナロ

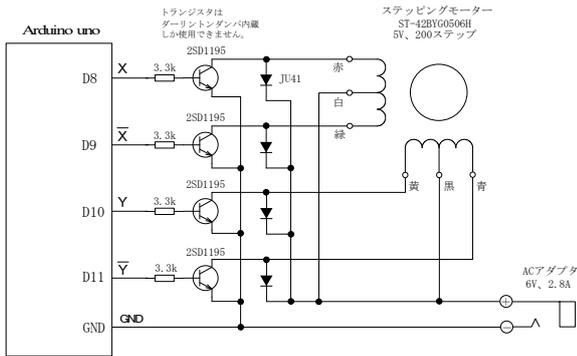


図1 ユニポーラステッピングモータードライバの回路図

グ入力，光センサー，アームロボを使ったプログラミング学習の導入を試みた。

2. ステッピングモータードライバの製作

図1および図2にユニポーラステッピングモータードライバの回路図および外観を示す。

製作に使用した部品および道具はダーリントントランジスタ2SD1195（5 A，100 V），ファストリカバリダイオード1JU41（600 V，1 A），カーボン抵抗（3.3kΩ，1/4W），2.1mm標準DCジャック（4 A，基板取付用），分割ロングピンソケット（2.54mmピッチ，42 P，3 A），ターミナルブロック（3ピン，青，縦，小，15 A/300 VAC），片面ガラスコンポジットユニバーサル基板めっき仕上Cタイプ（72×47mm，秋月電子通商），電線AWG24（梅澤無線），潤フロETFTE電線（CSMW用0.32mm），10mm金属製スペーサ，φ3×5mmなべ小ねじ，卓上ボール盤，はんだごて，ハンダ，はんだこて台，ニッパ（ミニ），ラジオペンチ（ミニ），ドライバ（+）である。

製作手順は以下の①～⑩に従う。

- ① 卓上ボール盤を使って，基板にDCジャックを取り付けるための穴（φ3mm）を3か所あける。
- ② DCジャックを穴に挿入しホットグールボンドで接着する。
- ③ ニッパを使って分割ロングピンソケットを5極にカットし，基板にはんだ付けする。

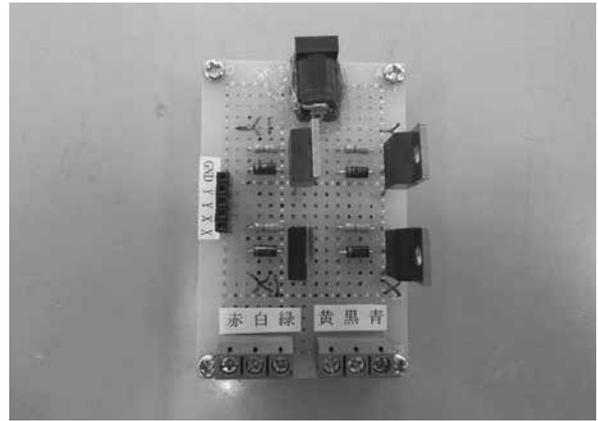


図2 ユニポーラステッピングモータードライバの外観

- ④ ターミナルブロック2個を基板にはんだ付けする。
- ⑤ トランジスタ4個を基板にはんだ付けする。
- ⑥ カーボン抵抗4個を基板にはんだ付けする。
- ⑦ ダイオード4個を基板にはんだ付けする。
- ⑧ はんだごてを使って，基板裏面に配線する。分割ロングピンソケットに至る電線は潤フロETFTE電線を使用し，その他はAWG24を使用する。
- ⑨ 基板に金属製スペーサ4個をなべ小ねじ4個でねじ止めする。
- ⑩ プリンターを使ってラベルシールに「赤白緑黄黒青」と「GND/Y Y/X X」と印刷した後，基板に貼る。

励磁方式として2相励磁方式⁸⁾を採用した。この方式は2相を同時に励磁し，消費電力は大きい，大きなトルクが得られる。図3にデジタルオシロスコープを使用して観測した，Arduinoから出力される2相励磁信号波形X（D8），/X（D9），Y（D10），/Y（D11）を示す。振幅5V，周波数25Hz，デューティー比50%である。Xと/X，Yと/Yはそれぞれ論理反転で，XとYの位相差は90°である。なお，2相励磁停止中はXとYは5V一定に，/Xと/Yは0V一定になる。

3. ソフトウェア

インターネットブラウザを立ち上げ，下記にア

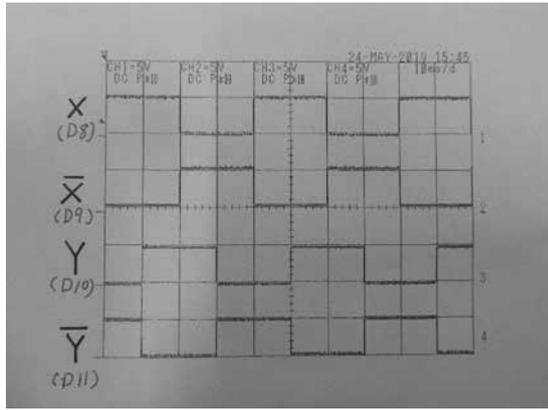


図3 2相励磁信号波形

アクセスしてフリーソフトであるArduino IDEをダウンロードする。

<http://arduino.cc/en/Main/Software>

Arduino IDEにはユニポーラおよびバイポーラステッピングモーターをコントロールするためのライブラリ⁹⁾が用意されている。

```
#include <Stepper.h>
```

ステッピングモーターをコントロールしようとする場合は、この命令により最初にStepperライブラリをインクルードする必要がある。

```
Stepper(steps, pin1, pin2, pin3, pin4)
```

この命令はArduinoに接続するステッピングモーターのステップ数 (steps) と接続ピン (pin1, pin2, pin3, pin4) を割り当てる。

```
setSpeed(rpms)
```

この命令はステッピングモーターの速さを毎分の回転数 (rpms) で設定する。この命令を実行

してもステッピングモーターは回転しない。

```
step(steps)
```

この命令はsetSpeed命令で設定した速さで、指定したステップ数 (steps) だけステッピングモーターを回転させる。逆回転させるにはステップ数を負の値にする。この命令を実行するとステッピングモーターは回転する。そして、止まるまで次の命令を実行せずに待ち続ける。

4. ステッピングモーター

図4および図5に今回使用したユニポーラステッピングモーター ST-42BYG0506HとST-42BYG020の外観を示す。

ST-42BYG0506Hは相数4, 基本ステップ角 $1.8^\circ \pm 5\%$, 1回転ステップ数200, コイル抵抗 $10 \pm 10\% \Omega/\text{相}$, 入力定格電圧5V, 定格電流0.5A/相, 静止トルク0.83kgf・cm, 重量0.22kgである。使用に際しての注意は、感電するので手で軸を回してはならない。なお、このままでは回転のようすがわからないので、ホットグールボンドを使ってステッピングモーターの回転軸にレゴブロック (Technic Brick 1×8) を、ステッピングモーターのボディにレゴブロック (4-Stud Cross Axle) を接着した。

ST-42BYG020は相数4, 基本ステップ角 $1.8^\circ \pm 5\%$, 1回転ステップ数200, コイル抵抗 $20 \pm 10\% \Omega/\text{相}$, 入力定格電圧12V, 定格電流0.6A/相,



図4 ST-42BYG0506H

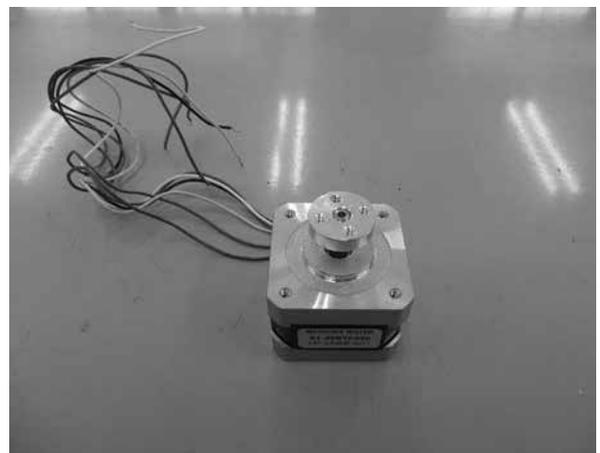


図5 ST-42BYG020

静止トルク1.6kgf・cm, 重量0.21kgである。使用に際しての注意は、感電するので手で軸を回してはならないのと、ハイパワー（高出力）仕様のため、DC12Vで使うと20分程度で触れられなくなるほど熱くなる。連続使用を想定する場合は、9V程度で使うことが推奨されている。なお、ステッピングモーターの回転軸には物を載せて回転できるように5mmモーターシャフト用ハブ（秋月電子通商）を取り付けた。

5. 授業例

5-1. 動かしてみる

まず、接続を行う。図6に示すように、Arduinoのデジタル8番（D8）、9番（D9）、10番（D10）、11番（D11）、GNDピン（GND）とステッピングモータードライバーのX、/X、Y、/Y、GNDをブレッドボード・ジャンパーワイヤー（オスーオス）で接続する。図7に示すように、ステッピングモーター ST-42BYG0506Hのコイルをステッピングモータードライバーのターミナルブロックにねじ止める。図8に示すように、ステッピングモータードライバーのDCジャックにスイッチングACアダプター（NP12-1S0523, 5V, 2.3A, 100~120V, 秋月電子通商）を接続し、その電源プラグをコンセントに差し込む。図9に示すように、ArduinoとパソコンをUSBケーブルで接続する。

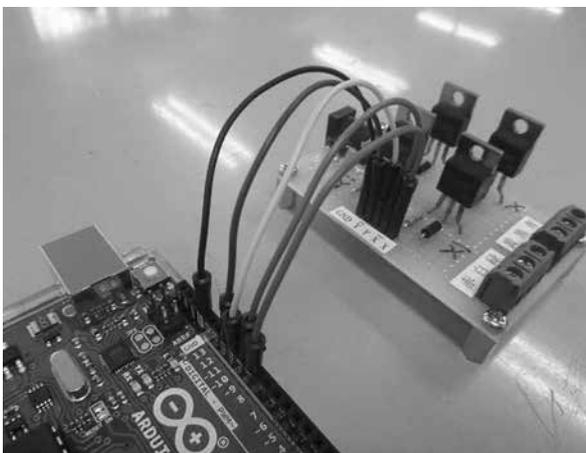


図6 Arduinoとドライバーの接続

次に、Arduino IDEを立ち上げてスケッチを作成する。このステッピングモーターのステップ数は200である。つまり、200ステップすると1回転する。図10にステップ数を200一定にし、set Speed (rpms) 命令で回転の速さを変化させた場合の回転のようすを観察するスケッチ1を示す。スケッチ1の1行目はStepperライブラリをインクルードする。2行目はStepper型の変数をstepperとして宣言し、Arduinoに接続するステッピングモーターのステップ数を200、接続ピンとして8、9、10、11番ピンを割り当てる。スケッチ1の4行目でrpmsを1とするとステッピングモーターは1分かけて1回転する。同様に2とすると30秒かけて1回転、3とすると20秒かけて1回転、4とすると15秒かけて1回転、6とすると10秒かけて1回転、12とすると5秒かけて1回転、30とすると2秒かけて1回転、60とすると1秒かけて1回転する。5行目はステップ数を200に設定する。

図11に回転の速さを30一定にし、step (steps) 命令でステップ数を変化させた場合の回転のようすを観察するスケッチ2を示す。スケッチ2の5行目でstepsを200とすると1回転、400とすると2回転、600とすると3回転、100とすると1/2回転、50とすると1/4回転する。逆回転させるには、それぞれ-200, -400, -600, -100, -50とすればよい。

図12に正転・逆転を繰り返すスケッチ3を示

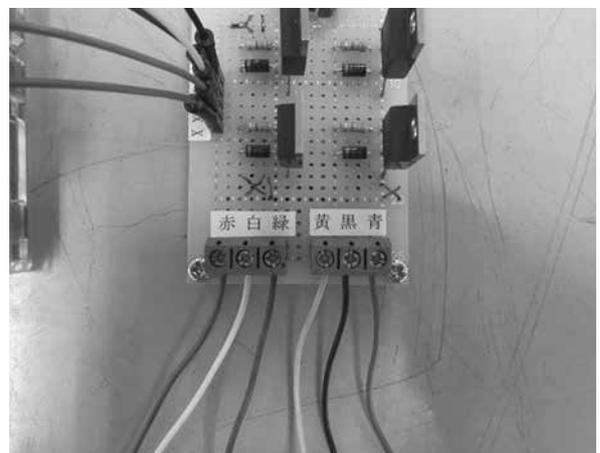


図7 ドライバーとステッピングモーターの接続

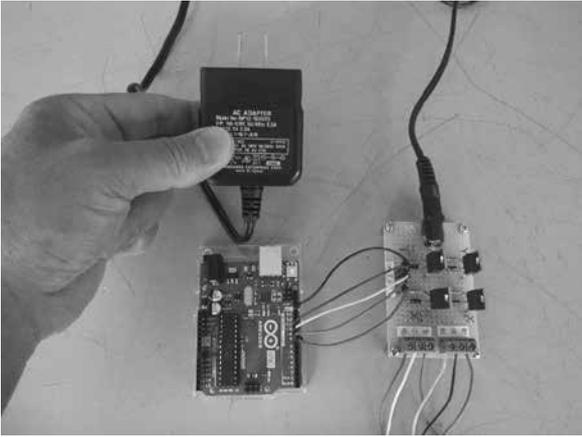


図8 ドライバーとACアダプターの接続

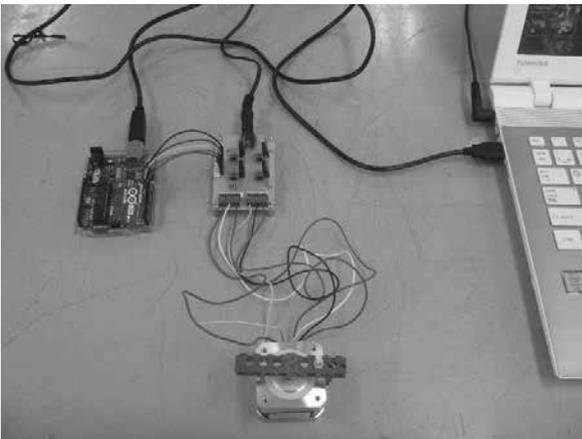


図9 Arduinoとパソコンの接続

す。void setup ()関数はスケッチの実行開始時、すなわち電源を入れた直後またはボードをリセットした後に一度だけ呼び出される。void loop ()関数はvoid setup ()関数が呼ばれた後に、繰り返し呼び出される。このため正転・逆転を繰り返すことになる。8行目と10行目のdelay (1000)はプログラムを1000ms、つまり1秒だけ止める命令である。

5-2. シリアル通信機能を使う

図13にシリアルモニタから f を送信するとステッピングモーターを1回転、bを送信すると逆1回転、hを送信すると1/2回転、qを送信すると1/4回転、1を送信すると1ステップ回転、-を送信すると逆1ステップ回転するスケッチ4を示す。スケッチ4の3行目は文字型の変数nを宣言する。

```
#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);

void setup() {
  stepper.setSpeed(1);
  stepper.step(200);
}

void loop() {
}
```

図10 スケッチ1

(ステップ数を一定、回転の速さを可変)

```
#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);

void setup() {
  stepper.setSpeed(30);
  stepper.step(200);
}

void loop() {
}
```

図11 スケッチ2

(ステップ数を可変、回転の速さを一定)

```
#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);

void setup() {
  stepper.setSpeed(30);
}

void loop() {
  stepper.step(200);
  delay(1000);
  stepper.step(-200);
  delay(1000);
}
```

図12 スケッチ3 (正転・逆転を繰り返す)

6行目のSerial.begin (9600) はシリアル通信のデータ転送レートを9600bps (ビット/秒) に設定する。9行目のSerial.read () は受信データを読み込み変数nに代入する。10行目以降はif文

```

#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);
char n;

void setup() {
  stepper.setSpeed(10);
  Serial.begin(9600);
}

void loop() {
  n=Serial.read();
  if(n=='f') {
    Serial.println("FORWARD");
    stepper.step(200);
  }
  if(n=='b') {
    Serial.println("BACKWARD");
    stepper.step(-200);
  }
  if(n=='h') {
    Serial.println("1/2");
    stepper.step(100);
  }
  if(n=='q') {
    Serial.println("1/4");
    stepper.step(50);
  }
  if(n=='1') {
    Serial.println("1");
    stepper.step(1);
  }
  if(n=='-') {
    Serial.println("-1");
    stepper.step(-1);
  }
}

```

図13 スケッチ4 (シリアル通信)

を使って受信した文字を判別し、Serial.println (データ) でデータの末尾にキャリッジリターンとニューラインを付けてメッセージをシリアルモニタに送信した後、指定したステップ数でステッピングモーターを回転させる。

図14に示すように「ツール」→「シリアルモニタ」の順にクリックしてシリアルモニタを起動する。図15に示すように入力エリアに f, b, h, q, 1, - のいずれかの小文字を一つずつキー入力して送信ボタンをクリックし、パソコンからArduinoに送信すると指定したステップ数でス



図14 シリアルモニタの起動

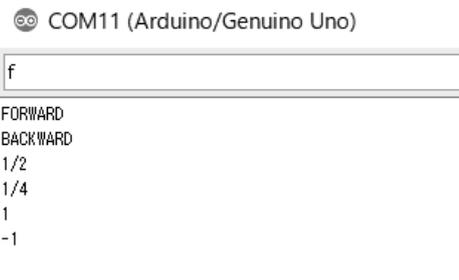


図15 シリアルモニタ画面

テッピングモーターが回転する。逆にArduinoからパソコンに f を受信するとFORWARD, b を受信するとBACKWARD, h を受信すると1/2, q を受信すると1/4, 1 を受信すると1, - を受信すると-1 というメッセージが送信される。

5-3. アナログ入力を使う

アナログピンの入力値をもとにステッピングモーターを回転させる。これまでの回路に図16に示す回路を追加する。半固定抵抗器 (10kΩ) をブレットボード (6 穴版, EIC-3901, 秋月電子通商) 上に配置し、ジャンパーワイヤーを使ってArduinoの5V, アナログピン (A0), GNDに接続する。

図17にスケッチ5を示す。スケッチ5の3~4行目は整数型の変数valおよびpreviousを宣言する。9行目のanalogRead (0) はアナログピン (A0) から半固定抵抗器の出力電圧0V~5Vの電圧を読み取り、0~1023の数値に変換した後、変数valに代入する。10行目はval-previousを計算してステップ数とし、ステッピングモーターを

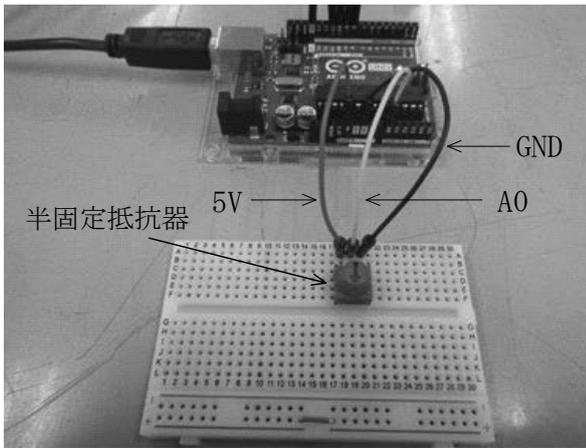


図16 追加した回路 (アナログ入力)

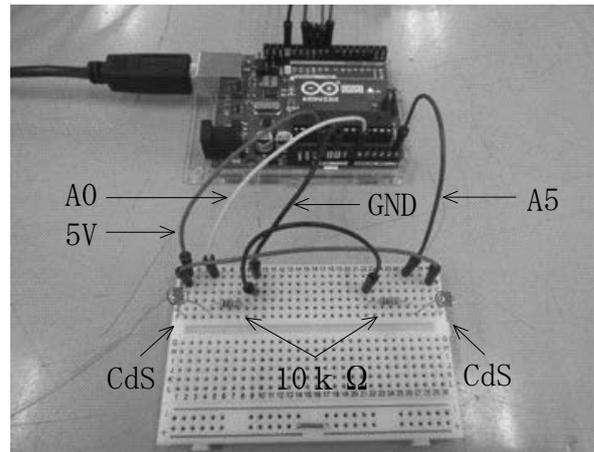


図18 追加した回路 (光センサー)

```
#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);
int val;
int previous = 0;

void setup() {
  stepper.setSpeed(30);
}

void loop() {
  val = analogRead(0);
  stepper.step(val - previous);
  previous = val;
}
```

図17 スケッチ5 (アナログ入力)

回転させる。11行目は変数valの値を以前のアナログピンの入力値として変数previousに代入して保存する。これにより、半固定抵抗器のつまみの回転に追従してステッピングモーターが回転する。

5-4. 光センサーを使う

光センサーを2個使用して、左側の光センサーに手をかざすとステッピングモーターが1/2回転、右側の光センサーに手をかざすと逆1/2回転させる。図16の回路を取り除き、図18に示す回路を追加する。光センサー (CdS)とカーボン抵抗(10kΩ, 1/4W) をブレッドボード上に配置し直列に接続したものを左右2組作る。左側の光センサーとカーボン抵抗の接続部をジャンパーワイヤーを使ってArduinoのアナログピン (A 0) に、右側

の光センサーとカーボン抵抗の接続部をアナログピン (A 5) に接続する。光センサーのカーボン抵抗と接続されていない側は2組ともArduinoの5 Vに、カーボン抵抗の光センサーと接続されていない側は2組ともGNDに接続する。

図19にスケッチ6を示す。スケッチ6の3~4行目はアナログピンの番号0および5を整数型の変数left_cdsおよびright_cdsとして宣言する。5~7行目は整数型の変数nL, nRおよび文字型の変数frgを宣言する。11行目は変数frgに空白を代入し初期化する。14~15行目は左側の光センサーから読み取った値を変数nLに、右側の光センサーから読み取った値を変数nRに代入する。16~19行目は変数nLおよびnRの値にメッセージを付けてシリアルモニタに送信する。20~29行目はif文を使って左右どちらの光センサーに手をかざしたかを判別し、ステッピングモーターを1/2回転または逆1/2回転させる。その際、変数frgにはステッピングモーターが現在1/2回転し終えた状態にある場合は文字列Lが、逆1/2回転し終えた状態にある場合は文字列Rが代入されており、これをif文を使って判別することによりgoto out命令でif文から抜け出し、重複して同方向に回転しないようにしている。

「ツール」→「シリアルモニタ」の順にクリックしてシリアルモニタを起動する。図20に示すように、光センサーから読み取った値そのものが時々刻々と表示される。左側の光センサーの値が

```

#include <Stepper.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);
int left_cds =0;
int right_cds =5;
int nL;
int nR;
char frg;

void setup() {
  stepper.setSpeed(30);
  Serial.begin(9600);
  frg=' ';
}

void loop() {
  nL=analogRead(left_cds);
  nR=analogRead(right_cds);
  Serial.print("left=");
  Serial.print(nL);
  Serial.print("  right=");
  Serial.println(nR);
  if(nL < 900 ) {
    if(frg == 'L') { goto out;}
    stepper.step(100);
    frg='L';
  }
  if(nR < 900) {
    if(frg == 'R') { goto out;}
    stepper.step(-100);
    frg='R';
  }
out:
  delay(200);
}

```

図19 スケッチ 6 (光センサー)

left, 右側の光センサーの値がrightである。例えば、左側の光センサーに手をかざして数値の違いを見てみると、明るいとき数値が大きくなり、暗いとき小さくなることを確認できる。これにより、光センサーの値がある値(閾値)よりも小さくなった場合に手がかざされたかと判別するわけであるが、スケッチの20行目と25行目では900を使っている。この値は周囲の環境(室内照明灯のON/OFF, 外から入る太陽光など)により変化するものであり、シリアルモニタに表示された値を見てそのつど決めなければならない。

COM11 (Arduino/Genuino Uno)

```

left=401  right=937
left=408  right=938
left=416  right=937
left=401  right=938
left=418  right=937
left=437  right=936
left=400  right=937
left=425  right=937
left=413  right=937
left=404  right=937
left=404  right=937
left=406  right=937
left=424  right=937
left=399  right=937
left=410  right=938
left=418  right=937

```

図20 シリアルモニタ画面

5-5. アームロボを使う

ステッピングモーターとサーボモーターを使って物を運び、物を容器に入れるアームロボ教材を製作した。製作手順は以下の①~⑧に従う。

- ① 図21に示すように、アーテックのサーボモーター3個とブロックを使用してアームロボ¹⁰⁾を組み立てる。
- ② 図22に示すように、支柱と接続するためにアームロボにブロック(基本四角, 白)を差し込む。
- ③ 図23に示すように、支柱を組み立てる。支柱1および支柱2はアームロボを取り付けるための支柱である。ブロック(ハーフA, 薄グレー)2個の上にブロック(基本四角, 薄グレー)を3個差し込み、それを2列にした。支柱3はアームロボが運ぶ物を載せるための支柱である。ブロック(基本四角, 薄グレー)を4個差し込んだ。土台はブロック(18ベース)を2個連結した。
- ④ 図24に示すように、アームロボを支柱に差し込む。サーボモーター servo5はアームを上下に振る。servo6はアーム先端を握る放す。servo7はアームを左右に振る。
- ⑤ 図25に示すように、厚さ2mmのダンボール紙を直径190mmの円形にカットした後、高さを1/2にカットした紙コップ4個を回転角度90°間隔で配置し、ホットグルボンドで接着する。

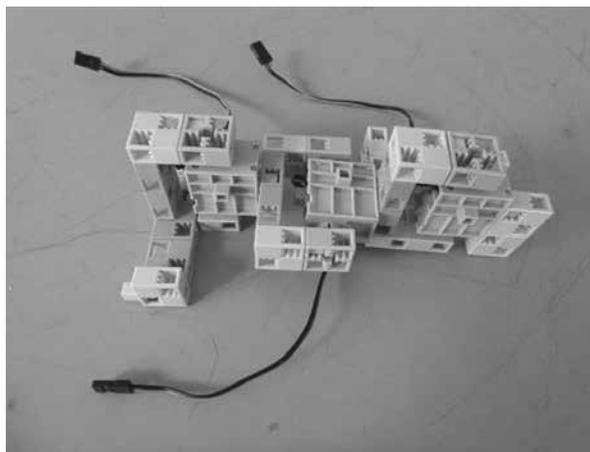


図21 アームロボを組み立てる

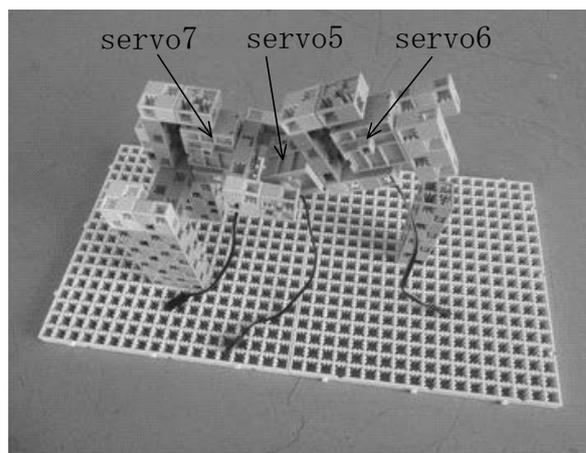


図24 アームロボを支柱に取り付ける

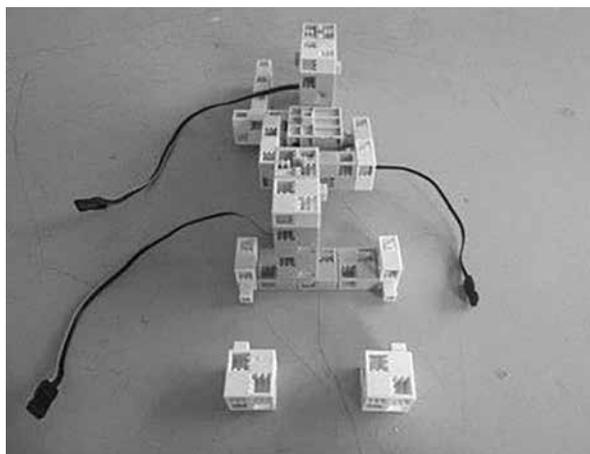


図22 基本四角を差し込む

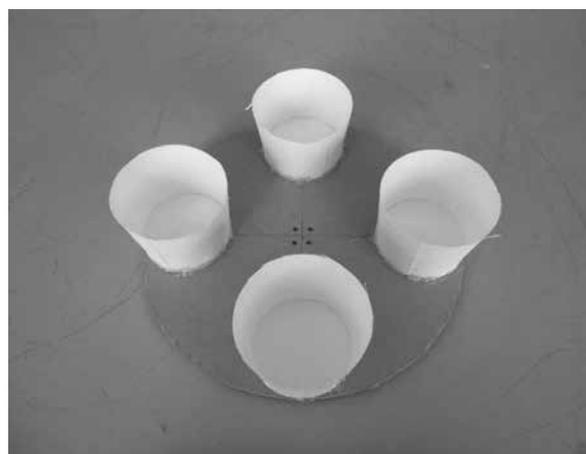


図25 ダンボール紙に紙コップを接着する

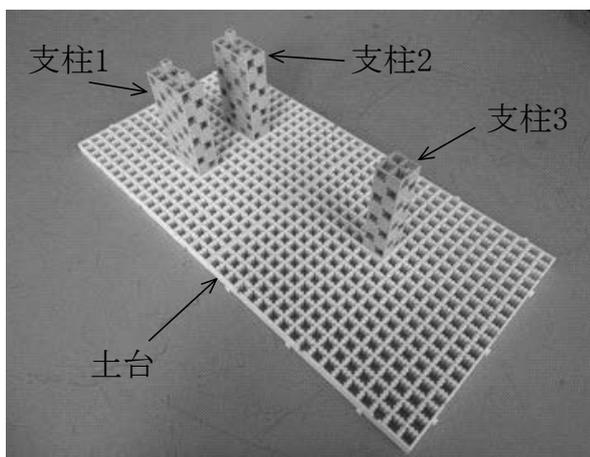


図23 支柱を組み立てる



図26 ステッピングモーターにねじ止めする

⑥ 図26に示すように、⑤をユニポーラステッピングモーター ST-42BYG020の回転軸に取り付けた5mmモーターシャフト用ハブにねじ止めする。

⑦ 図27に示すように、Arduinoのデジタル

8番, 9番, 10番, 11番, GNDピンとステッピングモータードライバーのX, /X, Y, /Y, GNDをジャンパーワイヤーで接続する。Arduinoのデジタル5番, 6番, 7番ピンとサーボモーター servo5, servo6, servo7

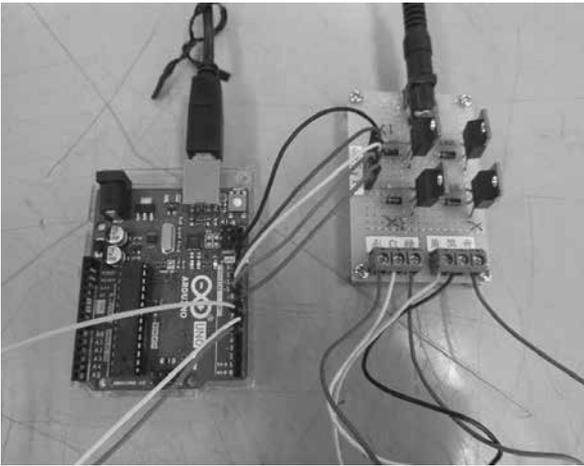


図27 Arduinoとドライバーおよびサーボモーターの接続

の各SIG（制御信号）をジャンパーワイヤーで接続する。ステッピングモーターのコイルをステッピングモータードライバーのターミナルブロックにねじ止めする。ステッピングモータードライバーのDCジャックにスイッチングACアダプター（9 V, 2.0A, 100~120V, 秋月電子通商）を接続し、その電源プラグをコンセントに差し込む。ArduinoとパソコンをUSBケーブルで接続する。

- ⑧ 図28に示すように、ブレットボード上でサーボモーター servo5, servo6, servo7の各+V（電源）およびGNDを電池ボックス（単3×4本、スイッチ付、XHコネクタ付）の+極および-極にジャンパーワイヤーで接続する。なお、電池ボックスはXHコネクタに両端ロングピンヘッダ（2P）を差し込んでブレットボードに接続した。

以上で、組み立て・配線作業は完成である。

図29にアームロボを使ってブロック（基本四角）を選び、4個の紙コップにブロックを3個ずつ入れるスケッチ7を示す。スケッチ7の2行目はServoライブラリをインクルードする。4~6行目はServo型の変数をservo5, servo6, servo7として宣言する。7行目は整数型の変数tを宣言し500を代入しておく。10~12行目のattach（pin）はサーボ変数servo5, servo6, servo7をArduinoのデジタル5番, 6番, 7番ピンに割り当てる。



図28 サーボモーターと電池ボックスの接続

13行目以降において、write（angle）はサーボモーターの角度を0°~180°の範囲で設定し、シャフトをその方向に向ける。この命令を実行すると、//の右側にコメントしてある通りにアームロボは動作する。up, downはアームを上下に振らせる。left, rightはアームを左右に振らせる。grip, releaseはアームを握る放つ。なお、rotationはステッピングモーターを1/4回転させる。delay（t）はプログラムを500ms、つまり0.5秒だけ止める。19行のfor文は、4個の紙コップにブロックを3個ずつ入れるために波カッコで囲まれた命令を12回繰り返し実行する。

図30に位置決めを完了したアームロボ教材の全体の外観を示す。位置決めおよび操作の仕方は以下の①~⑦に従う。

- ① 土台のブロック（18ベース）を粘着テープ（セロテープや両面テープなど）で机の上に固定する。
- ② ステッピングモータードライバーのDCジャックからACアダプターを外して回転しないようにする。
- ③ プログラムを実行する。アームロボはブロックを選び、紙コップに入れる動作を12回繰り返すので、この間に物を載せる支柱（図23の支柱3）と紙コップ（つまり、ステッピングモーター）の位置を調整する。
- ④ 位置決めができれば支柱を移動し、ステッピングモーターを粘着テープで机の上に固定す

```

#include <Stepper.h>
#include <Servo.h>
Stepper stepper = Stepper(200, 8, 9, 10, 11);
Servo servo5;
Servo servo6;
Servo servo7;
int t=500;

void setup() {
  stepper.setSpeed(30);
  servo5.attach(5);
  servo6.attach(6);
  servo7.attach(7);
  servo5.write(45);           // up
  delay(t);
  servo6.write(60);          // release
  delay(t);
  servo7.write(100);         // left
  delay(2000);
  for(int i=1; i<=12; i++) {
    servo5.write(85);        // down
    delay(t);
    servo6.write(130);       // grip
    delay(t);
    servo5.write(45);        // up
    delay(t);
    servo7.write(170);       // right
    delay(t);
    servo5.write(80);        // down
    delay(t);
    servo6.write(60);        // release
    delay(t);
    servo5.write(45);        // up
    delay(t);
    servo7.write(100);       // left
    delay(t);
    stepper.step(50);        // rotation
  }
  servo5.write(45);          // up
  delay(t);
}

void loop() {
}

```

図29 スケッチ7 (アームロボ)

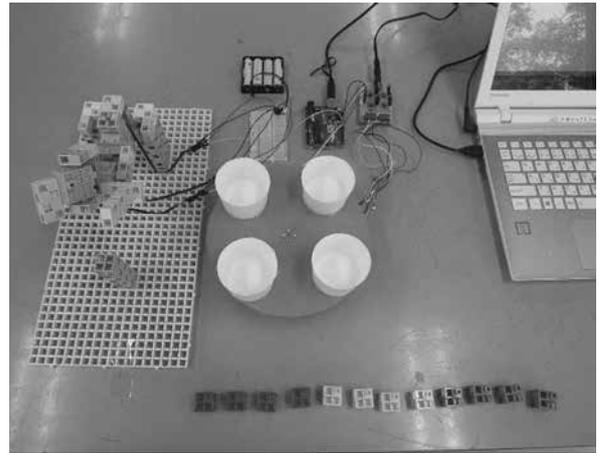


図30 アームロボ教材の全体の外観

ロックを持ち上げる。ブロックを左へ運ぶ。ブロックを下げる。ブロックを放し紙コップに入れる。アームを上げる。アームを右に移動し最初の状態にする。ステッピングモーターで紙コップを回転させる。以上の動作を12回繰り返すと図40に示すように、4個の紙コップそれぞれに赤、黄、緑のブロックが1個ずつ入れられ終了となる。

6. 発展

ステッピングモーターにはユニポーラステッピングモーターとバイポーラステッピングモーターの2種類がある¹¹⁾。前者は高速トルクが大きく、後者は低速トルクが大きい特長を持っている。

図41および図42にバイポーラステッピングモータードライバーの回路図および外観を示す。

る。

- ⑤ ステッピングモータードライバーのDCジャックにACアダプターを接続する。
- ⑥ アームロボに運ばせるブロック(基本四角, 赤4個, 黄4個, 緑4個)を用意する。
- ⑦ ArduinoのRESETボタンを押してアームロボを動作させ、物を載せる支柱の上でブロックを手渡し運ばせる。

図31～図39にアームロボ教材の動作のようすを示す。人の手によりブロックを支柱の上に載せる。上からアームが下りて来てブロックを握る。ブ

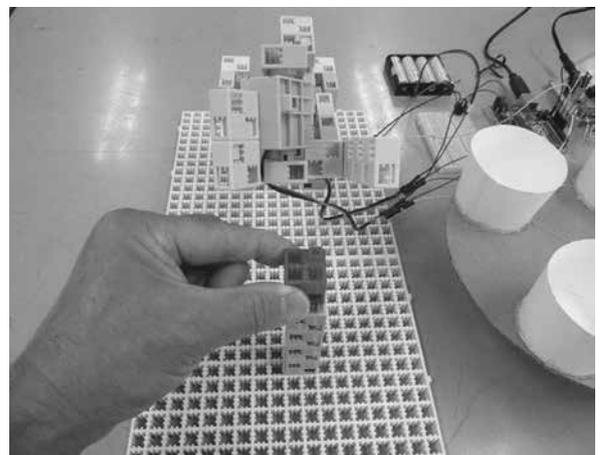


図31 ブロックを支柱に載せる

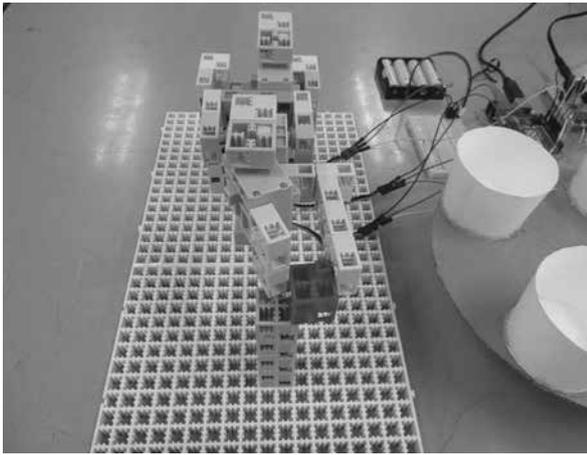


図32 ブロックを握る

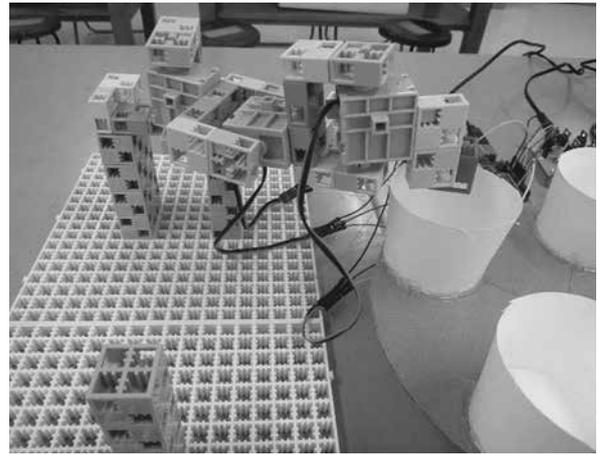


図35 ブロックを下げる

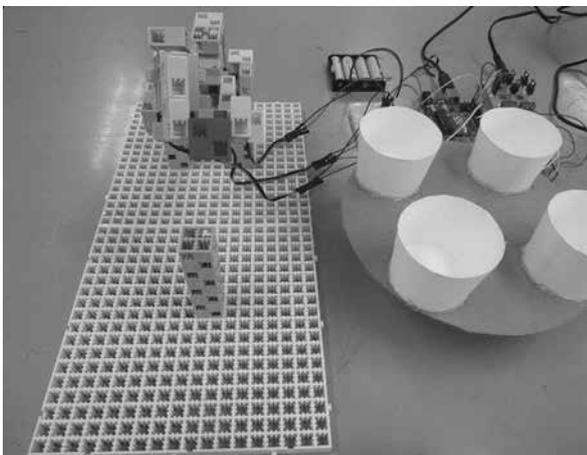


図33 ブロックを持ち上げる

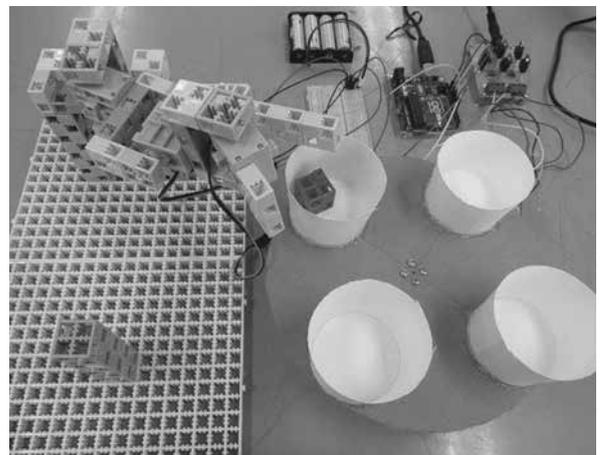


図36 ブロックを放し紙コップに入れる

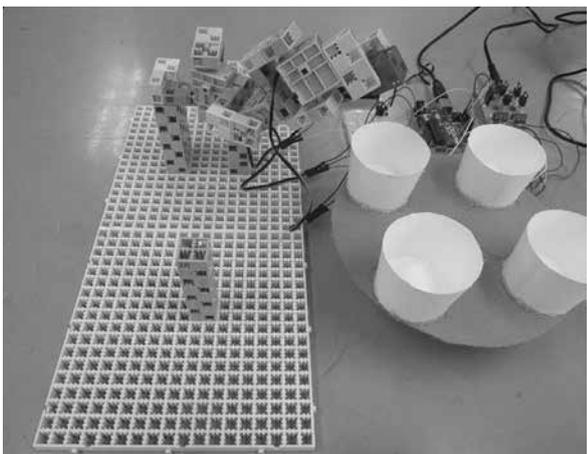


図34 ブロックを左へ運ぶ

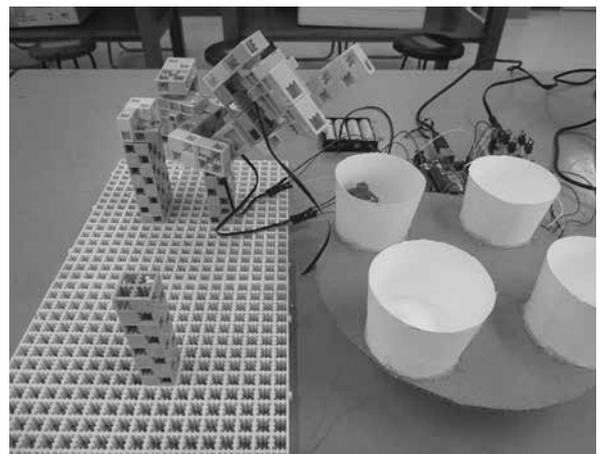


図37 アームを上げる

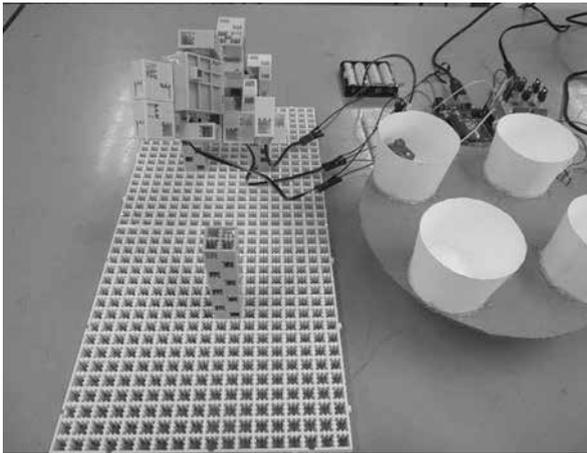


図38 アームを右に移動する

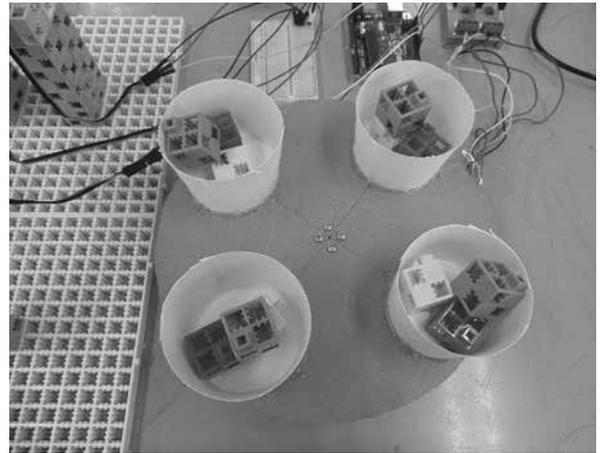


図40 12回繰り返し終了

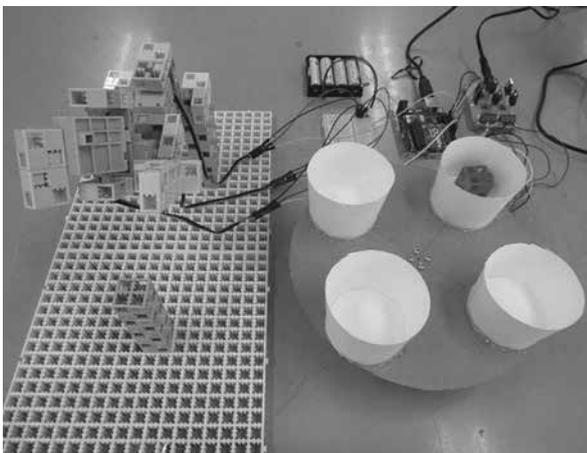


図39 紙コップが回転する

Arduinoへの接続の仕方とスケッチはユニポーラの場合と同一である。

大学生または院生を対象とした授業では、ステッピングモータードライバの製作が容易なユニポーラの方が導入しやすい。大学生なら90分で製作できる。しかし、バイポーラの方は部品点数が2倍であり製作時間が長くなるのにもない配線ミスも多くなることが予想される。そこで、授業では回路図と実物を見せながら口頭で紹介することとどめ、特に電気回路について豊富な知識と上達した技能を有する受講生だけに製作させること

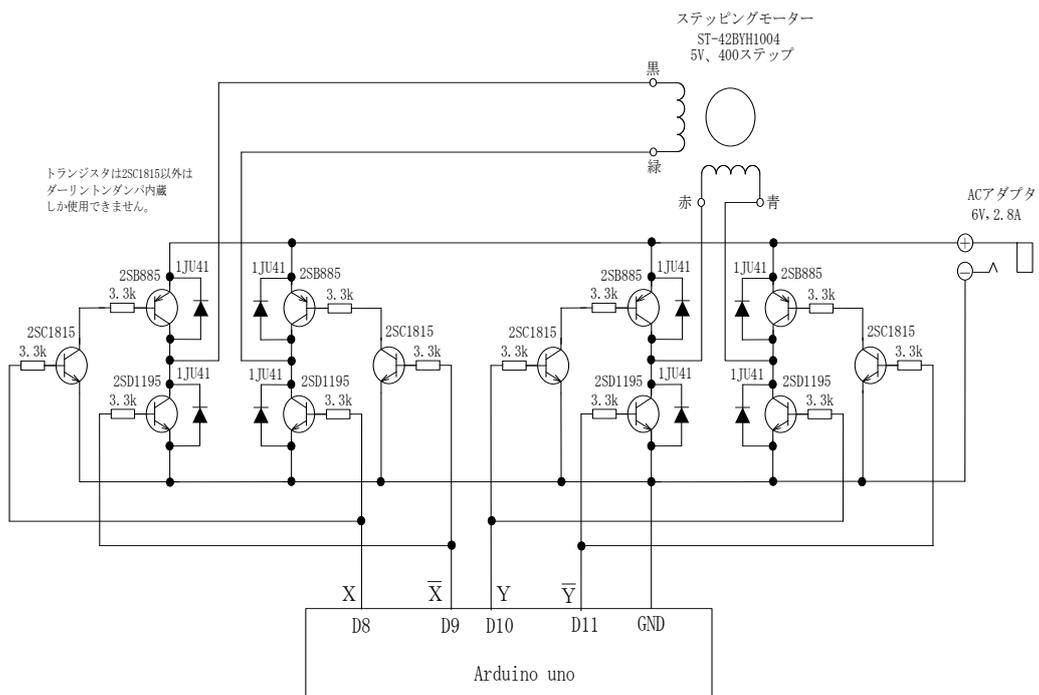


図41 バイポーラステッピングモータードライバの回路図

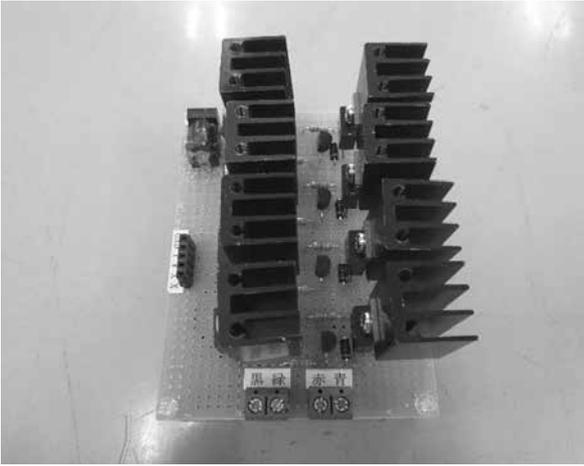


図42 バイポーラステッピングモータードライバの外観

を想定している。

7. おわりに

筆者は、技術科の学生を対象として電気領域と情報領域を融合した、Arduinoで制御するプログラミング教材を開発し、2019年に院生を対象に授業実践を行った。授業全体の構成は第1章 LED, 第2章 RGB LED, 第3章 ブザー, 第4章 圧電サウンダ, 第5章 光センサー, 第6章 温度センサー, 第7章 超音波距離センサー, 第8章 人体検出赤外線センサーと外部割込み, 第9章 ボイスメモと外部割込み, 第10章 ソリッドステートリレーと外部割込み, 第11章 DCモーターの制御, 第12章 サーボモーターの制御, 第13章 ステッピングモーターの制御, 第14章 SDカードシールド, 第15章 XBeeである。本論文は第13章をまとめたものである。今後は難易度を調整の上、学部学生を対象に授業実践を行う。

引用文献

- 1) 文部科学省検定教科書, 技術・家庭 [技術分野], 開隆堂, 106, (2016)
- 2) ステッピングモーターの基本,
<https://www.marubun.co.jp/service/mps/a7ijkd>

000001156.html (2020年8月12日参照)

- 3) ステッピングモーターの基本的な構造と動作原理,
<https://www.moonsindustries.com/jp/article/techschooljp-stepmotor-00006-step-motor-basics> (2020年8月12日参照)
- 4) 臨床工学技士の為の電子工作,
<https://electronic.tousekice.com/> (2020年8月12日参照)
- 5) Arduino UNO + A4988でステッピングモーターを回す,
<https://blog.goo.ne.jp/snagai4500/e/ca8bbfca4c6c7f1c772d129ea5ea4e65> (2020年8月12日参照)
- 6) ArduinoとA4988でステッピングモーターを制御する方法,
<https://iot.keicode.com/arduino/arduino-stepper-motor-a4988.php> (2020年8月12日参照)
- 7) arduinoでステッピングモーターを回す実験,
<http://miha.jugem.cc/?eid=94> (2020年8月12日参照)
- 8) ステッピングモーターの駆動運転,
<https://www.indexpro.co.jp/article/detail/2/8> (2020年8月15日参照)
- 9) Arduino日本語リファレンス,
<http://www.musashinodenpa.com/arduino/ref/> (2020年8月15日参照)
- 10) Studuino ブロックプログラミング環境 入門ガイド 後編, 35-53, (2018),
https://www.artec-kk.co.jp/studuino/ja/studuino_v2.php (2020年8月20日参照)
- 11) ユニポーラ駆動とバイポーラ駆動,
<http://www.mechatroidea.com/sekkei-seizu/acutuater/s027-stepping-u-b.html> (2020年8月28日参照)

(旭川校准教授)