

Arduino を用いたマイコン制御の入門教育

檜本 弘* 平社 信人*

(2014年11月27日受理)

1. はじめに

平成25年度から機械工学科の4年を対象に「マイコン制御」なる科目を開設した。従来のコンピューターを計算の道具として利用するだけでなく、制御のためのコントローラーとしての活用方法の習得を目指したものである。マイコンについては以前からも実験実習の一部のテーマに組み込まれていたが、これらは短時間であり、後日学生に話を聞くとマイコンに「触れた・体験した」であっても「学習した・習得できた」とまではいえない様子であった。そこで、半期15回の必修科目であるが、

- ・マイコンで何ができるか
- ・そのためにはどのようなプログラムが必要かを全員に理解してもらえる内容を目指し、安価なセンサーや機器を使用しているが、興味を引くようなテーマの準備を行ってきた。今回は実際の講義で扱っている項目を中心に、その後の応用を含め、現状を報告する。

2. 機械工学科の情報処理教育の移り変わり

2.1. Fortran から C 言語へ

従来より機械工学科では情報処理として Fortran によるプログラミング教育を行ってきた。数値計算用途に限れば Fortran も十分実用であるとともに、過去に開発された膨大な計算用ライブラリが存在するという優位性がある。例えば線形計算用ソフトウェアライブラリである LAPACK[1] も初版は Fortran77 で実装され、その後 Fortran90 が用いられている。今では C 言語インターフェースも統合されているが、行列計算の場合には内部で行列転置が行われるため、オーバーヘッドが生じている。

このように Fortran の有用性もあるが、さすがに初心者の方が最初のプログラミング言語として学習するには古さを感じさせる書式も多い。また、過去の資産を活用するのでなければ Fortran にこだわる必要はなく、すでにパソコンから大型コンピューターまで広く使用されている C 言語に変更することになった。

2.2. カリキュラム変更と「マイコン制御」の新設

C 言語への変更は決定したが、講義内容の変更は学生の学年進行にともない平成23年から26年の4年間をかけて実施した。カリキュラム変更の過程を表1に示す。

表1 カリキュラム変更の学年進行

		H22年	H23年	H24年	H25年	H26年
2年	情報処理 I	Fortran	C 言語 入門	C 言語 入門	C 言語 入門	C 言語 入門
3年	情報処理 I	Fortran	Fortran	C 言語 入門続き	C 言語 入門続き	C 言語 入門続き
4年	設計 製図	Fortran 設計計算	Fortran 設計計算	Fortran 設計計算	C 言語 設計計算	C 言語 設計計算
5年	情報処理 II	Fortran 数値計算	C 言語 速習	C 言語 速習	C 言語 速習	C 言語 数値計算

組み込みマイコンのプログラミング言語として、アセンブラ以外では C 言語（最近では C++）のみ用意されていることが多い。C 言語への変更により、単に数値計算だけではなくマイコンを用いた制御用途の学習も可能になった。これは従来の Fortran では絶対に不可能である。そこで新たに4年生で「マイコン制御」なる半期科目を新設し、コンピューターをコントローラーとして活用するための講義内容とした。この科目は平成25年から開設したが、カリキュラムの変更の学年進行に合わせる形で、前年度においても計測工学の一部として講義を行った。

2.3. マイコンボードの選定

プログラミング教育では学生一人に1台の環境を提供し、複数の学生が1つの機器を使用する状況をつくらないことが重要と考えている。また、プログラミングは得手不得手が出やすいところがある。そこで、学生が自宅でも自習できるようにマイコンのボードは学生各自が購入し、それ以外に必要なものは学科で準備するようにした。マイコンボードを選定するにあたってはおもに次の点に絞って検討した。

*機械工学科

- (1) 学生各自が購入するため、低価格なものであること、おおよそ教科書1冊の価格と大差ないこと。
- (2) 特定の教科書は指定しないが、参考となる日本語の情報がWeb上に多くあり、初心者向けの書籍も刊行されていること。
- (3) 開発ためのソフトウェアが無償で提供されており、学生個人のパソコンにもインストールできること。
- (4) すでに多くの無償ライブラリが公開されており、短いコードでプログラムが記述できること。
- (5) ある程度の期間はマイコンボードの変更を行わずに済むよう販売実績の多いもの。

これらより、最終的にはArduinoを選択した。なお、Arduinoではプログラムのことを「スケッチ」と呼んでいるが一般的ではないため、本報告ではプログラムとしている。またプログラムの一部の意味でコードとも称している。

3. Arduino

3.1. Arduino とは

Arduino[2]は「コンピューターの初心者でも簡単に扱えるマイコンボード」を目指し、2005年にイタリアで開発プロジェクトがスタートした。現在でも安価で簡単に使用できるハードウェアとして販売されている。図1は現行品であるArduinoの上面と裏面である。

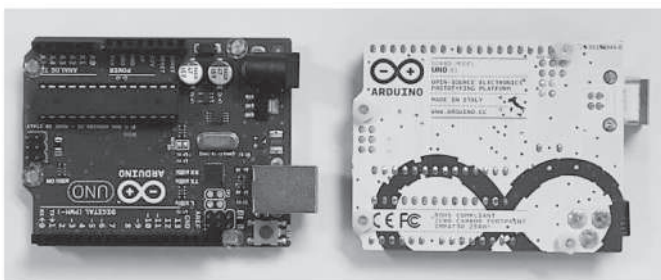


図1 マイコンボード Arduino UNO R3

ただし、ボードだけでなく、プログラムの言語や開発のための環境などソフトウェアを全て含んだ総称として「Arduino」と呼んでいる。

ところで、Arduinoの読み方であるが、このイタリア語をカタカナ表記すると「アルデュイーノ」に近いようである。

3.2. Arduino の特徴

授業で使用しているArduino UNO R3は米国Atmel社製のAVR ATmega 328Pという8bitのマイコンを搭載し

ている。今さら8bitでは時代遅れの感想を持つかもしれない。しかし、16さらには32bitと高性能のマイコンになるにつれ、その機能の理解と性能を引き出すための手順が複雑になり、初心者が最初の一步を踏み出すまでの敷居が高いものになる。入門用としてArduinoは初心者にとって最適であると判断している。

Arduino で出来ること・得意なこと

Arduinoは簡単にいろいろな機器を接続することが可能である。たとえば、LEDなどはボード本体の端子に直接接続でき、半田付けは不要である。

Arduinoのハードウェア、ソフトウェアのいろいろな情報がWeb上に公開されている。例えば、ボードの回路図やプリント基板の配線図も入手可能である。そのため、いろいろな周辺機器を使用するためのソフトウェアも作成・公開されており、例えば任天堂スーパーファミコンのゲームパッドもArduinoに接続して利用可能である。



図2 ゲームパッドを接続した Arduino

通常ArduinoはパソコンとUSB接続してプログラム開発から動作まで行うが、プログラムを書き込んだ後は乾電池などで電源を供給すれば単独で動作でき、いろいろな装置に組み込むことが可能になる。

Arduino に出来ないこと・不得意なこと

上述のようにArduinoに搭載しているのは8bitマイコンであり、処理能力は高くない。そのため複数の処理を同時に行ったり、多くの機器を接続して動作させるには無理がある。また、Arduinoは基板上に半田付けされたICがむき出しの状態であり、金属片がボードに触れるとショートする恐れがある。

4. 授業の進め方と題材

4.1. 学習環境と授業の進め方

繰り返しになるが、プログラムの学習において環境や

機器を一人ひとりに準備する事が絶対に必要と考えている。授業で使用する Arduino は学生に共同購入してもらっている。イタリア製のため、円安の影響で販売価格が上昇したが、現在の値段は 3,000 円弱である。(教科書は使用せず、Web 上においた資料等を用いて授業を進めているので、学生にこれ以外の費用負担はない。) 本体に接続するセンサーや機器類は学科の予算で購入し、授業で使用している。

授業は機械工学科 3 階の設計実習室で実施している。プログラムの開発環境は次節に示す通りであるが、必要なものは全て Arduino の Web サイト [2] からダウンロードでき、無料で使用できる。48 台の学生用パソコンには既にインストール済みであり、授業に限らず放課後も使用できる。また、ボード本体は学生が購入しているので、自宅のパソコンでも自習可能である。

毎回の講義前半は各項目の説明、後半はプログラム実習になるように工夫している。プログラミングにおいて初心者はプログラムが上手く動作したという「成功体験」を多く積むことが重要であり、内容を絞ったプログラムの作成を行えるように注意している。また、この授業は教員 2 名と支援センター員 1 名の計 3 名で行っており、進度の遅い学生が授業の進行に取り残されないようにサポートしている。

4.2. Arduino の開発環境

Arduino のプログラムの開発は Arduino と USB と接続し、コードの入力、コンパイルとボードへの書き込みまでを行う。これは Windows に限らず、Macintosh や Linux 上でも可能である。授業は Windows7 (64bit) のパソコンを用いている。



図 3 Arduino IDE を起動した直後

プログラムの入力には開発環境である Arduino IDE を起動する。図 1 は起動直後の様子である。Arduino はパソコンと USB 接続しているが、Windows 上ではシリアルポートして認識される。実際に接続されているシリアルポート番号と IDE のポート選択が一致していないとボードへの書き込みに失敗するので、確認が必要である。

Arduino IDE の画面からはプログラムの入力、コンパイルとボードへの書き込みだけでなく、ライブラリの使用、サンプルプログラムのオープンやシリアルモニター利用などの作業がソフトウェアの切り替えなしに実行することができる。

4.3. Arduino 言語

Arduino のプログラムには Arduino 言語で記述する。これは C/C++ 言語を基に変更・拡張した言語である。通常 C 言語でプログラムを作成する場合、最初に呼び出される関数として main 関数が必要となるが、Arduino では main 関数ではなく、次の二つの関数を記述することが最低限必要になる。

- (1) void setup()
- (2) void loop()

電源 ON もしくはリセットかかった後は (1), (2) の順番に実行される。

(1) の setup 関数はボードの初期設定を行う関数であり、最初に 1 回だけ実行される。プログラムの最初に記述し、省略することも名前を変更することもできない。

(2) の loop 関数は setup 関数に続いて実行され、処理は電源が切れるまで繰り返される。主に処理すべき内容を記述する。これも省略することも名前を変更することもできない。なお、実際にはプログラムをコンパイルする前に標準でライブラリを自動的に付加している。これは以下のファイルを見ると明らかである。

```
arduino-1.0.5\hardware\arduino\cores\arduino\main.cpp
```

main.cpp において、最初に呼び出される main 関数が存在しており、この中の setup() と loop() のみをプログラマーが作成していたことがわかる。リスト 1 に main.cpp の内容を示す。

```
#include <Arduino.h>

int main(void)
{
    init();
```

```
#if defined(USBCON)
  USBDevice.attach();
#endif

setup();

for (;;) {
  loop();
  if (serialEventRun) serialEventRun();
}

return 0;
}
```

この部分のみをプログラミングしている。

リスト 1 main.cpp の内容

以下、授業で扱う各題材について、羅列的ではあるが簡潔に説明する。

4.4. LED 点滅

C 言語の教科書では“Hello World!”と表示させるプログラムを最初のサンプルとして示されていることが多いが、マイコンでは接続した LED を点滅させるプログラムを例示している場合が多い。俗にいう「LED チカチカ」である。Arduino には自由に使用できる LED が一つだけ半田付けされている。この LED は 13 番ピンに接続されているため、一切のハードウェアを追加することなく点灯・消灯させる事が可能である。Arduino IDE やボード本体が正常に動作するかのテストのためにも購入後最初に動作させている。

実際にはサンプルプログラムの「Blink」を呼び出している。このプログラムで使用する関数は次の通りである。

- (1) pinMode 関数
- (2) digitalWrite 関数
- (3) delay 関数

pinMode 関数で入出力ポートの方向を設定し、digitalWrite 関数で入出力ポートに HIGH か LOW を書き込んでいる。さらに delay 関数で指定した時間、処理を停止させている。この間 LED は点灯もしくは消灯状態を維持している。

4.5. シリアル入出力と独自関数の活用

パソコンと Arduino の間でシリアル通信する機能が備わっている。ボード上で自由に利用でき、内部の状態を表示できるのは上述の LED 1 個のみである。シリアル通

信を用いれば、パソコンの画面上に多くのデータを表示させる事が可能である。特にプログラムが期待した動作をしない場合に実行状況をモニターできるので、プログラムの間違い箇所の発見が容易になる。

このプログラムで使用する関数は次の通りである。

- (1) Serial.begin 関数
- (2) Serial.print 関数, Serial.println 関数

Serial.begin 関数でシリアル通信の転送速度を設定し、Serial.print 関数もしくは Serial.println 関数でパソコンに送信する文字や常数、変数を設定している。なお、両者の違いは Serial.println 関数は改行とラインフィードのコードを自動的に追加するが、Serial.print 関数はそれがなく、改行されない。

Arduino で標準に準備されている関数は Arduino からパソコンへの通信であり、Arduino IDE のシリアルモニターやターミナルソフトを用いてパソコンの画面に出力している。しかし、逆方向つまりパソコンからの文字出力を Arduino が数値として受信する関数を用意されていない。そこで、このような機能があればいろいろ応用ができ有用と考え、独自に inpInt 関数を作成した。そのソースリストを以下に示す。この関数では先頭の負号(-)と数字以外の文字は無視するようになっている。この関数を用いれば、パソコンのキーボードを数値入力装置として使用することが可能である。但し、int 型の変数の使用範囲(-32768~32767)は考慮していないので、利用の際に注意が必要である。

```
void inpInt(int *x) {
  int inByte, n=0, m;
  boolean fst=true, neg=false;

  while(Serial.available() == 0);

  do {
    delay(3);
    inByte = Serial.read();

    if(fst && inByte == '-') {
      neg = true;
    }
    fst = false;

    if('0' <= inByte && inByte <= '9') {
      m = inByte - '0';
      n = n * 10 + m;
    }
  }
}
```

```

    }
}
while(Serial.available() > 0);

*x = (neg) ? -n : n ;
}

```

リスト2 inpInt 関数のソースリスト

4.6. キャラクタ液晶シールド

前節のシリアル通信を使用したパソコン画面への出力は手軽で機器の追加も不要と便利ではあるが、パソコンとUSB ケーブルで接続している必要がある。Arduino 単体で使用したい場合には利用できない。そこで、ボード本体の上部にキャラクタ液晶パネルと専用 IC からなる液晶シールドを接続して使用する。シールドとは Arduino 用の機能拡張用回路基板である。図 4 は授業で使用しているキャラクタ液晶シールド(16 文字×2 行)である。専用 IC は標準といえる日立製 HD44780 とコンパチブルである。

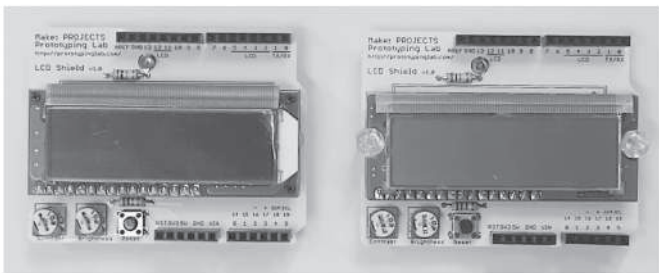


図4 キャラクタ液晶シールド 左が青色 右が緑色

HD44780 互換であるため、標準にインストールされている「LiquidCrystal ライブラリ」が使用できる。これにより、制御信号の意味を意識する必要がなく、モジュールの初期化など面倒な手順も1行記述するだけで済むなど短いコードを書くだけで利用可能である。

プログラムで使用するには、最初に LiquidCrystal オブジェクトのインスタンスの生成が必要になる。生成の際には引数で制御信号とデータバスの接続を指定すればよい。使用する主な関数は次の通りである。

- (1) インスタンス名.begin 関数
- (2) インスタンス名.clear 関数
- (3) インスタンス名.home 関数
- (4) インスタンス名.setCursor 関数
- (5) インスタンス名.print 関数

インスタンス生成後、最初に begin 関数を用いてライ

ブラリを初期化して、キャラクタ液晶パネルの文字数と行数の指定を行う必要がある。

4.7. 圧電ブザー

Arduino で簡単に音を鳴らせる部品に圧電ブザーがある。音を鳴らすにはスピーカーと連想するが、スピーカーを動作させるには Arduino からの出力では不可能であり、オーディオ用のオペアンプ等を用いた回路が必要になる。それに対して、圧電ブザーは Arduino のポートに直結でき、手軽に動かすことができる。図 5 に2種類の圧電ブザーを示す。



図5 圧電ブザー

Arduino では圧電ブザーを鳴らすために矩形波を出力する関数が用意されている。使用する関数は次の通りである。

- (1) tone 関数
- (2) noTone 関数

pinMode 関数で圧電ブザーを接続したピンを出力に設定した後、tone 関数で出力する周波数を指定する。また、出力時間も指定できる。この関数によりブザーから音がでる。tone 関数で出力時間を指定しなかった場合は音が鳴り続けるので noTone 関数で停止させる必要がある。

なお、tono 関数で出力時間を指定しても、それより前に再度 tone 関数を実行した場合には、その時点で次の音が出来られるので、delay 関数を併用して出力時間を確保することが必要になる。また、同時に複数の音を鳴らすことはできない。

4.8. 温度センサー

温度を測定するためのセンサーとしてナショナル・セミコンダクター社製 LM35DZ を使用している。摂氏温度(°C)に比例した電圧が出力でき、温度係数は使用範囲内で 10.0mV/°C である。これは温度が 1°C 上昇すると出力電圧も 0.01V 高くなる。0°C のとき出力は 0V に調整されているので、センサーを接続したアナログピンから AD 変換した電圧値を 100 倍すれば温度になり、扱いが楽である。図 6 に温度センサーの写真を示す。T0-92 パッケージなので、一見するとトランジスタのように見える。



図6 温度センサー LM35DZ

上述のようにこのセンサーは測定値に比例した電圧が出力される。このアナログ量を Arduino では内蔵の AD 変換機能を用いて取り込む。AD 変換で使用する関数は次の通りである。

- (1) analogReference 関数
- (2) analogRead 関数

Arduino ボードには6チャンネルの10bitAD変換機能を搭載している。analogReference 関数はアナログ入力に使用される基準電圧の設定であり、次の文字列のうちどれかを一つを指定する。

- DEFAULT : 5V の電源電圧を基準電圧 (省略時)
- INTERNAL : 1.1V の内蔵基準電圧
- EXTERNAL : AREF ピンに加えられる電圧 (最大5V) を基準電圧

analogRead 関数は指定したアナログピンから値を読み取るものである。0 から基準電圧までの入力電圧を0 から1023 までの数値に変換する。

温度センサーLM35DZ の動作温度範囲は0~100℃であり、温度係数からセンサーから出力される電圧の最大値は1Vとなる。analogReference 関数でDEFAULT と設定した場合、入力できる電圧の最大は5V となり、基準電圧の20%までしかLM35DZ では使用せず、精度的に非常に効率が悪い。INTERNAL と設定した場合、基準電圧は1.1V となり、約91%まで使用できている。測定温度は0.11℃刻みの測定となる。

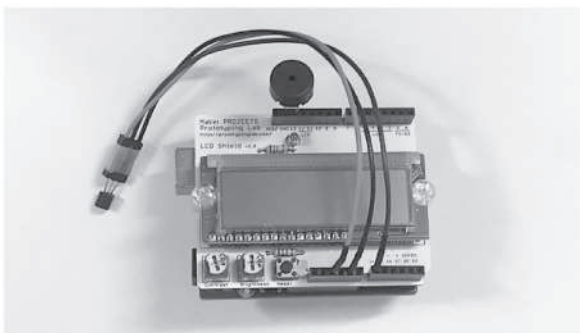


図7 気温警報装置

4.6~4.8 のキャラクタ液晶シールド、圧電ブザー、温度センサーを用いることにより、Arduino を気温警報装置として使用することができる。温度センサーで気温を測定して液晶に表示するとともに、設定した閾値を超えた時点で圧電ブザーから警告音を発生させることで実現している。図7にその構成を示す。

4.9. XBee による無線通信

4.5 のシリアル通信はパソコンと Arduino を USB で接続していたが、この通信を無線化する。無線化できればケーブルを引き回す必要がなくなり、より便利になる。実際には ZigBee 規格に準拠した Digi international 社製の XBee を使用した。この XBee を用いれば簡単に無線通信を実現可能である。パソコンと Arduino の間で通信するには Arduino 側に XBee シールド、パソコン側には XBee 本体をパソコンの USB に接続するためのモジュールが必要である。

XBee には動作モードが2つあり、「トランスペアレントモード」を使用することで、シリアル通信のケーブルを XBee と置き換えたように動作する。事前にこの設定を行った XBee モジュールを使用することで、プログラムには修正を加えることなく USB 接続から無線通信に置換が可能である。例えば上記の気温警報装置のプログラムに温度をシリアル通信する部分を追加すれば、パソコンから離れた位置にある温度もワイヤレスで収集することが可能になる。図8に専用シールド上にセットした XBee モジュールを示す。

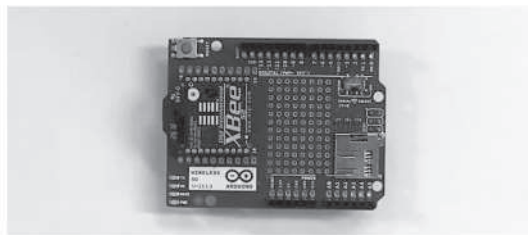


図8 XBee と専用シールド

4.10. DC モーターの回転

電子工作で動くものといえば、DC モーターが最も安価で手軽である。プラモデルでも使用されており、電池をつないでスイッチを入れればフル回転する。しかし、Arduino に DC モーターを直結しても電流不足から回転させる事はできない。そのため専用のモータードライバを搭載したシールドを使用する。授業ではガリレオ7社製の 4Motors シールド[3]を用いる。DC モーターの回転をより細かく制御するため販売元が無償で提供しているライブラリを使用する。このライブラリは Arduino IDE から使用できるようにインストールが必要である。

DC モーターの駆動で問題となるのは電源である。この電源は Arduino と共通にすることも別にすることもできるが、別電源（単3電池3本）を用意している。接続している DC モーターの定格電圧は3Vであり、これより高い電源を DC モーターにかけると故障の原因にもなる。必ず定格電圧以下になるように制限する必要がある。また、モーター起動時やストール時には大きな電流が流れるが、これがシールドの最大定格電流を超えないような DC モーターを選定する必要がある。図9に4Motors シールドを示す。

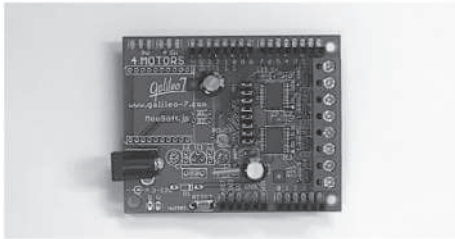


図9 4Motors シールド

5. 応用例 1 4M設計製図

4年機械工学科の「設計製図」の授業では2段の歯車減速機的设计計算, 3D-CADによるモデリングとFEMによる強度解析さらに3Dプリンターによる製作と発表会までを1年間で行っている。製作した歯車減速機の動作確認は従来手動で回転させていたが、これを(1)モーターによる連続回転(2)出力軸の回転数計測と減速比の確認を行う必要があった。高価な既成の装置を購入・利用するのではなく、Arduinoを用いたシステムの利用を検討した。

5.1. ステッピングモーターの回転制御

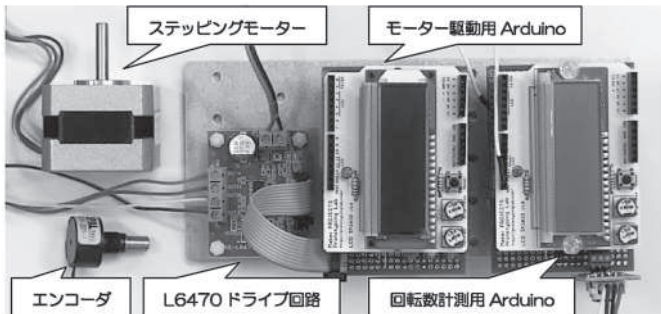


図10 ステッピングモーターとドライブ回路および回転数計測用回路

ステッピングモーターは軸が時計の秒針のように一定角度ずつ回転するモーターである。エンコーダ等がなく

ても回転を制御できる。このためにはDCモーターと同様に専用ICによる回路が必要になる。STMicroelectronics社製バイポーラスステップモータドライブICであるL6470を使用したドライブキットを使用した。図10にステップモーターとドライブ回路、さらに5.2~5.3で用いる回転数計測用回路とエンコーダを示す。

L6470はArduinoとSPIで通信し、2相バイポーラスステップモータの正逆転、加減速、ブレーキ、速度制御、位置制御に加えて脱調監視機能も内蔵している高機能なものである。モーター駆動用の電源電圧は8~45Vなので、別途駆動電流の大きい電源を準備する必要がある。

Arduino用シールドとして商品化されていないので、専用のライブラリも存在していない。詳細は割愛するが、公開されている技術情報[4]をもとにSPI通信で指令をL6470に送信することでモーターの加減速なども短いコードで実現可能である。

5.2. 回転数計測(1)

軸の回転数は取り付けられたロータリーエンコーダからのパルス信号を取り込み、これから算出する方法を用いている。計測方法は(1)一定時間のパルス数をカウントする方法(2)パルス信号のHIGHもしくはLOWである時間を測定する方法の2種類の方法で実現した。この節では(1)の方法、次節では(2)について簡単に説明する。

一定時間のパルス数をカウントする方法は文献[5]に紹介されたFreqライブラリを使用している。大まかな処理の流れは計測条件の設定、一定時間ごとのアップデートと周波数の随時読み出しを繰り返すことになる。このプログラムではArduino言語の範囲を超え、マイコンのレジスタの参照や書き込みが必要となる。ライブラリもC++で記述されている。残念ながら、ここまでになると授業の範囲を超えるので、ライブラリの使用のみに留める予定である。なお、調整を行ったプログラムで実測した結果、約1MHzまでは3桁程度の精度で周波数を測定することが可能であった。3,000円程度のマイコンでもこの範囲まで測定できるといえる応用が可能である。

5.3. 回転数計測(2)

上記(1)の方法は高い周波数の測定には向くが、低い場合には測定時間が長くなる問題がある。例えば、10Hz程度のパルスの周波数が有効桁数3桁を満たすよう測定するには100秒以上必要となる。これでは実用的ではない。そこで、ArduinoのpulseIn関数を使用し、パルスの長さをμ秒単位(実用上はその10倍程度)で求められるので、低い周波数でも短時間での測定が可能である。

6. 応用例 2 5 M卒業研究

最後に卒業研究における Arduino 活用の一例を示す。

6.1. ねじ緩み試験機におけるカウンター

JIS にはねじの緩み試験に関する規格がなく、企業においてはアメリカ航空規格 (NAS3350) に準拠した試験が行われている。この試験機の振動数 1750~1800cpm, 衝撃加振回数 30000cycles で、試験時間は約 17 分になる。加振回数を正確に数えるため Arduino を用いたカウンターを製作した。センサー部は反射タイプのフォトリフレクタを使用し、トランジスタのスイッチング回路を用いて赤外線反射強度の違いから矩形波を発生させている。センサーは試験機のスコッチヨーク機構の近くに固定している。

シールドは自作し、上述の回路の他に Start/Stop スイッチも配置している。液晶シールドにはカウンター数と試験開始からの経過時間も表示している。図 11 にセンサーとシールドを取り付けた Arduino を示す。Arduino には AC アダプターから 9V の電源を供給している。2 枚のシールドを重ねて使用している。キャラクタ液晶シールドは最上部に接続する必要があり、それを考慮して自作シールド上にスイッチ等の部品を配置している。

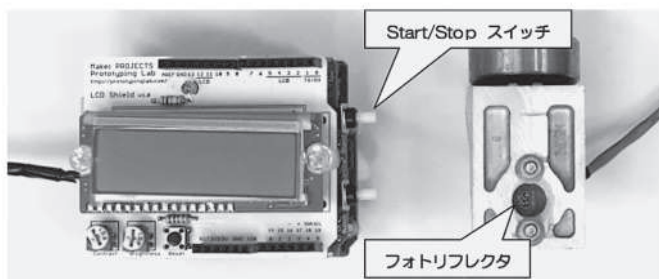


図 11 センサーと液晶+自作シールドを取り付けた Arduino

赤外線反射光は強弱の 2 値ではなく、ノイズを伴っているため、誤カウントも発生しやすい。しかし幸いにも振動数が 1750~1800cpm とほぼ一定でありため、センサーの不感時間を適当に設定することより、問題なくカウンターとして動作している。

この例のように、Arduino を試験機の一部として組み込むことは比較的容易であり、高価な装置の代替になり得ることが期待できる。また、プログラムの改良により、装置に組み込み後でも機能強化が可能である。

7. おわりに

「Arduino を用いたマイコン制御の入門教育」と題して、羅列的ではあるが、機械工学科 4 年の必修科目「マイコン制御」とその後の応用について報告した。

まだまだ内容的に不十分なのが、今までの感想である。例えば、Arduino に搭載されているマイコン (ATmega328P) の内部構造についてはほとんど触れていない。プログラムを作成するのにマイコンの内部を全て理解している必要はないが、より高機能や高速なプログラムの必要に迫られた時には避けて通れなくなる。

卒業研究などでも、これらの知識が活用されると期待して、題材の検討や変更も継続的に行っていきたいと考えている。

参考文献および URL

- [1] <http://www.netlib.org/lapack/>
- [2] <http://www.arduino.cc/>
- [3] <http://www.galileo-7.com/?pid=19411264>
- [4] <http://www.st.com/web/en/resource/technical/document/datasheet/CD00255075.pdf>
- [5] 鈴木 哲哉: 作って遊べる Arduino 互換機, ソシム, pp. 89-97, 2013

参考にした書籍

【Arduino 関係】

- (1) 河連 庸子, 山崎 文徳, 神原 健: Arduino スーパーナビゲーション しくみと応用テクニック, リックテレコム, 2012
- (2) Massimo Banzi, 船田 巧: Arduino をはじめよう 第 2 版, オライリージャパン, 2012
- (3) 小林 茂: Prototyping Lab — 「作りながら考える」ための Arduino 実践レシピ, オライリージャパン, 2010
- (4) 高橋 隆雄: たのしい電子工作 Arduino でガジェットを作ろう!, 秀和システム, 2013
- (5) 牧野 浩二: たのしくできる Arduino 電子工作, 東京電機大学出版局, 2012

【XBee 関係】

- (1) 濱原 和明, 佐藤 尚一ほか: 超お手軽無線モジュール XBee, CQ 出版, 2012
- (2) Robert Faludi, 小林 茂, 水原文: XBee で作るワイヤレスセンサーネットワーク, オライリージャパン, 2011

Introductory Education of the Microcomputer Control using Arduino

Hiroshi KASHIMOTO and Nobuto HIRAKOSO

The programming language used in information processing education from 2011 in mechanical engineering department was changed to C from Fortran. Not only the numerical computation but also the study of the control using the microcomputer was thereby enabled and a new subject has been initiated in 2013. The name of this 4th grader's subject is "microcomputer control" and is compulsory in half year. Each individual student is using Arduino purchased by oneself, is taking the lesson and doing a programming training. The aim of this subject is to have a student understand next.

- (1) what is made with a microcomputer ?
- (2) what kind of program is necessary for it ?

Inexpensive sensors and equipment were being used, were prepared subject to feel that there is interest to students. In this report, the contents of the lecture are reported including the later application around an item to treat by a lecture.