

支援センター主催 平成19年度教育研究支援センター技術研修会報告
-----「コントロールマイコンPICを使ったアクチュエータ制御入門」-----
教育研究支援センター 中尾三徳

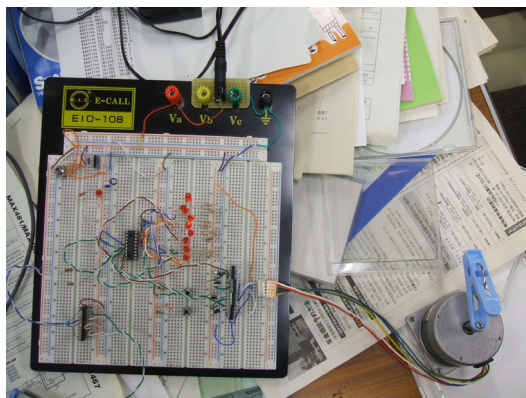
概要

支援センターで開催した技術研修会も以下の様に、今回で8回目を迎えました。

教育研究支援センター技術研修会 開催一覧

- 1回 「初めてのホームページ作成講座」 中尾三徳 2002. 3. 18～3. 22
- 2回 「マシニングセンター使用方法」 神田尚弘 2002. 7. 22～7. 26
- 3回 「走査型電子顕微鏡・X線マイクロアナライザの操作方法」 西 彰矩
2003. 9. 2～9. 2
- 4回 「4サイクルエンジンの理論と分解組み立て」 仲井正明 2004. 8. 5～8.
- 5回 「ネットワーク環境におけるセキュリティーのあり方」 日下孝二 2004. 12.
- 6回 「マイクロコントローラPICの活用」 徳方孝行 2005. 8. 3～8. 5
- 7回 「ワイヤー放電加工機による物作り教育への利用と応用」 川村純司
2006. 3. 8～3. 1
- 8回 今回「コントロールマイコンPICを使ったアクチュエータ制御入門」 中尾三徳
2007. 11. 27～11. 2

第6回技術研修で行ったPICの制御はアセンブラ言語で行いました。今回は、FREWAREのHITEC-C言語を使った実験を行います。アセンブラ言語からC言語に替わったことで、皆様もよりPICの制御が分かりやすくなったのではないかと思います。また、それぞれの周辺電子素子も規格表やマニュアル等をそのまま添付しましたので、この実験が終われば、各人で自力でPICを使った電子回路を構築できると思います。自由な発想でブレッドボード上に回路を構築していただければ幸いです。



コントロールマイコンPICを使ったアクチュエータ制御入門

教育研究支援センター 第1班 中尾三徳

1. まえがき

この技術研修会も以下の様に、今回で8回目を迎えました。(´▽`)

教育研究支援センター技術研修会 開催一覧

- 1回 「初めてのホームページ作成講座」 中尾三徳 2002.3.18～3.22
- 2回 「マシニングセンター使用方法」 神田尚弘 2002.7.22～7.26
- 3回 「走査型電子顕微鏡・X線マイクロアナライザの操作方法」 西 彰矩 2003.9.24～9.2
- 4回 「4サイクルエンジンの理論と分解組み立て」 仲井正明 2004.8.5～8.
- 5回 「ネットワーク環境におけるセキュリティのあり方」 日下孝二 2004.12.
- 6回 「マイクロコントローラPICの活用」 徳方孝行 2005.8.3～8.5
- 7回 「ワイヤ放電加工機による物作り教育への利用と応用」 川村純司 2006.3.8～3.1
- 8回今 「コントロールマイコンPICを使ったアクチュエータ制御入門」 中尾三徳 2007.11.27～11.2

第6回技術研修で行ったPICの制御はアセンブラ言語で行いました。今回は、FREEWAREのHITEC-C言語を使った実験を行います。アセンブラ言語からC言語に替わったことで、皆様もよりPICの制御が分かりやすくなったのではないかと思います。また、それぞれの周辺電子部品も規格表やマニュアル等をそのまま添付していますので、この実験が終われば、各人で自力でPICを使った電子回路を構築できると思います。自由な発想でブレッドボード上に回路を構築していただければ幸いです。

PICのプログラムのコンパイル、焼き付け用に3台のPCサーバを用意しておりますのでサーバのマイクロソフトファイル共有フォルダーにCで作ったプログラムをコピーしていただければ、各サーバで処理が行えます。

サーバ名は

- ①HPC09
- ②HPC15
- ③HPC21

で、それぞれに共有フォルダをフォルダー名

「技術研修」

という名前で作成しています。

マイネットからコンピュータの検索でサーバ名を指定していただけたらフォルダーが出てくると思います。読み書き自由になっていますので、コピーして、コピーしたサーバのところに外向きコンパイル、PIC書き込みを行ってください。

2. PIC書き込みライター

PICの書き込みライターはAKIのPICライターを3つのサーバともに同じものを使用しています。

ライターボード 秋月 AKI-PICプログラマ Ver4 キット秋月電子通商オリジナルのプログラマキットです。MPLABで実際にPICに書き込むことができるプログラムデータ（ヘキサファイル）をHITEC-Cで生成します。

ブレッドボードの周辺の電源コネクタは赤が+、青が-となるように配線しています。

ACアダプターから供給される12V、1.0Aの電力をそれぞれ、周辺配線に送っています。

4. 電子素子の説明

今回使用する電子素子は、抵抗、コンデンサ、レギュレータ、PIC、スイッチ、発振子（セラロック）LED、DCモータドライバIC、ダーリントトランジスタ、DCモータ、ステッピングモータ、整流ダイオードとなっています。

資料はP-09の「使用電子部品説明」を参照ください

5. PICマイコンとHITEC-C

MPLAB Ver7.62では統合環境のインストールと一緒にHITEC-Cの開発環境もインストールされます。以下に参考文書を添付します。

資料はP-12の「C言語によるPICプログラミング」を参照ください

6. 実習

基本的に1日1テーマと考えています。ハードの回路は前回作ったものを取り外し違うところだけ新たに作成するという方法で実験を行っていきます。

6.1. LEDを使ったSWによるエレベータシミュレータ

実験概要は、LEDを1列に8個配置し、一番手前が1階、1番向うが3階、真中が2階というようにする。そして、プッシュスイッチを2つ用意し、通常では、LEDランプは1階で点灯している。2階にあたるSWを押すと、2階までLEDが点灯を順順に行き、2階に達したら、また1階に戻ってくる。同じように3階を押すと3階まで順順に点灯して行き、3階に達したら、1階に戻ってきて、1階で待機する。

6.1.1. 使用素子

ブレッドボード、LED×9、PSW×3、sw、PIC16F84A、R(330Ω)×8、レギュレータ、コンデンサ、セラロック20M

6.1.2. 考え方と回路図

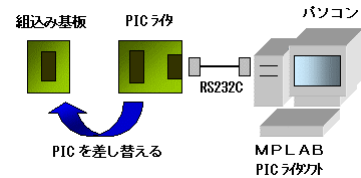
メインルーチンでは、1階にあたるLEDのみを点灯させて、while文で繰り返し、この繰り返しの中にSW1、2に当たるポートの変化があれば、それに対応したサブプログラム（2階もしくは3階に順順に点灯するプログラム）に動作が渡されれば良い。

6.1.3. C言語プログラム

P-25の「プログラムリスト01」を参照ください

開発環境の全体像は、以下のような感じになります。

3つのサーバ開発用パソコンのシリアルポートとPICライター基板を接続して、PICライターソフトでPICにプログラムをダウンロードします。プログラムを書き込んだPICはライター基板から取り外して、ターゲットとなる基板に付けて使用します。

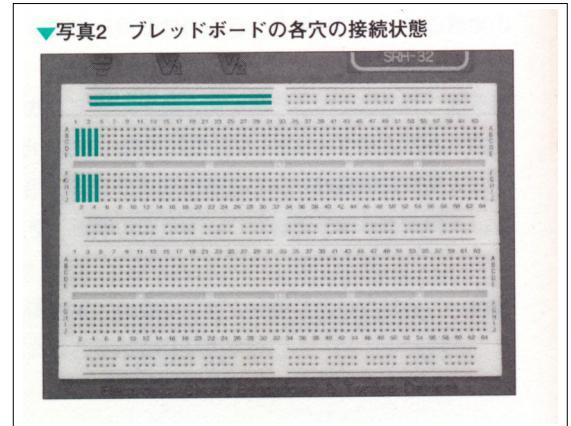


細かい操作については質問していただければとおもいます。

資料はP-06の「AKI-PICライター」を参照ください

3. ブレッドボード

少し大きめの4列のものを使用します。ブレッドボードの裏の配線の様子は以下の図のようになっています。赤と青の線の様につながっています。



線で消された穴の箇所が全部、裏でつながっています。

使用するケーブルは、10BASETケーブルの芯線を使用します。これは適当な長さでケーブルを切りラジオペンチで芯線をケーブルカバーから引き出して使用します。(少し力が必要です)

6.2. DCモータの回転制御

4つのプッシュSWを押すそれぞれ押したびにSWに対応するDCモータドライバのモードが0から1から0というように替わり、それに対応してDCモータの回転の様子が変わるプログラム。SWの状態はLEDで確認できるようにする。

6.2.1. 使用素子

ブレッドボード、DCモータ、DCモータドライバIC、LED×10、PSW×3、PIC16F84A、R330Ω PSW×2、TSW、R10K×2、R100×2、コンデンサ

6.2.2. DCモータ原理

資料はP-19の「DCモータを回す」を参照ください

6.2.3. 考え方フローチャートと回路図

6.2.4. C言語プログラム

P-27の「プログラムリスト02」を参照ください

6.3. STEPPINGモータの回転制御

ステッピングモータの回転の励磁方法を3つ変えて、動作させるプログラムを作成する。プッシュSWを3つ使い、

- ①で1相励磁
- ②で2相励磁
- ③で1-2相励磁

で回転するようにする。うまくできれば、逆回転するSWも考えてみる。

6.3.1. 使用素子

ブレッドボード、STEPPING MOTOR、LED×10、PSW×3、PIC16F84A、R330Ω×8ダイオード、ダーリントトランジスタ

6.3.2. ステッピングモータ原理

資料はP-22の「ステッピングモータの原理」を参照ください

6.3.3. 考え方と回路図

6.3.4. C言語プログラム

P-29の「プログラムリスト03」を参照ください

7. 応用実習

今回は素子用意していませんが、やってみようと思われる方がおられれば幸いです。

7.1. 7セグメントLEDのダイレクト制御表示

数字表示用7セグメントLEDを使って、専用のドライバーICを使い、高速にセグメント表示を切り替え、消費電力を少なくする。

7.2. 赤外線LEDとホトダイオードによる通信実験

赤外線LEDから信号を送信し、赤外ホトダイオードでその信号を受信するシステムを作成する。

```
//
// Delay Object Extern Define Verup version
// Hitec-C Sample program list02 Version 2007.10.2
// DC モータの回転制御デモ(LIST02)
//
#include "pic.h"
#include "delay.h"
//
int dcfoward(void),dcreverse(void),dcbreak(void);
//
int main(void)
{
//
TRISB=0x00;
TRISA=0xff;
//
while(1)
{
// スイッチ検出
DelayMs(10);
if(PORTA==0x01) dcfoward();
DelayMs(10);
if(PORTA==0x02) dcreverse();
DelayMs(10);
if(PORTA==0x03) dcbreak();
//
}
}
///// subroutine
/////
int dcfoward(void)
{
PORTB=0x01;
DelayMs(10);
return(1);
}
int dcreverse(void)
{
PORTB=0x02;
DelayMs(10);
return(1);
}
int dcbreak(void)
{
PORTB=0x03;
DelayMs(10);
return(1);
}
}
```

```
//
// Delay Object Extern Define
// Hitec-C Sample program list01 Version 2007.10.2
// エレベータシーケンスデモ(LIST01)
//
#include "pic.h"
#include "delay.h"
//
int f2ndfloor(void), f3ndfloor(void);
//
int main(void)
{
TRISB=0x00;
TRISA=0xff;
while(1)
{
PORTB=0x01;
if (PORTA==0x02) f2ndfloor (); // 3階
SW
if (PORTA==0x01) f3ndfloor (); // 2階
SW
}
}
///// subroutine
/////
int f2ndfloor(void)
{
PORTB=0x02;
DelayMs(250);
DelayMs(250);
PORTB=0x04;
DelayMs(250);
DelayMs(250);
PORTB=0x08;
DelayMs(250);
DelayMs(250);
PORTB=0x04;
DelayMs(250);
DelayMs(250);
PORTB=0x02;
DelayMs(250);
DelayMs(250);
return(1);
}
int f3ndfloor(void)
{
PORTB=0x02;
DelayMs(250);
DelayMs(250);
PORTB=0x04;
DelayMs(250);
DelayMs(250);
PORTB=0x08;
DelayMs(250);
DelayMs(250);
PORTB=0x10;
DelayMs(250);
DelayMs(250);
PORTB=0x20;
DelayMs(250);
DelayMs(250);
PORTB=0x40;
DelayMs(250);
DelayMs(250);
PORTB=0x80;
DelayMs(250);
DelayMs(250);
return(1);
}
}
```

```
//
// Delay Object Extern Define Verup Version
// Hitec-C Sample program list03 Version 2007.11.29
// STEPPING モータの回転制御デモ (LIST03)
//
#include "pic.h"
#include "delay.h"
//
int
step1phase(void), step2phase(void), step12phase(void);
//
int main(void)
{
int STEPMODE=0x00;
//
TRISB=0x00;
TRISA=0xff;
//
while(1)
{
// スイッチ検出 入力ラッチ モータ制
御
DelayMs(10);
if (PORTA==0x01) step1phase (); // 1
相励磁
DelayMs(10);
if (PORTA==0x02) step2phase (); // 2
相励磁
DelayMs(10);
if (PORTA==0x03) step12phase (); // 3
相励磁
}
}
///// subroutine
/////
int step1phase(void)
{
PORTB=0x00000001;
DelayMs(250);
PORTB=0x00000010;
DelayMs(250);
PORTB=0x00000100;
DelayMs(250);
PORTB=0x00001000;
DelayMs(250);
return(1);
}
int step2phase(void)
{
PORTB=0x00000011;
DelayMs(250);
PORTB=0x00000110;
DelayMs(250);
PORTB=0x00001100;
DelayMs(250);
PORTB=0x00001100;
DelayMs(250);
PORTB=0x00001001;
DelayMs(250);
return(1);
}
int step12phase(void)
{
PORTB=0x00000001;
DelayMs(250);
PORTB=0x00000011;
DelayMs(250);
PORTB=0x00000110;
DelayMs(250);
PORTB=0x00001100;
DelayMs(250);
PORTB=0x00001001;
DelayMs(250);
return(1);
}
}
```

このとき、通信プロトコルは232C用のものを使う。