

情報工学実験のためのH8マイコン学習システム用プログラムの開発

三上 剛*・阿部 司**・吉村 斎***

Development of H8 Microcomputer Programs for Computer Science Laboratory

Tsuyoshi MIKAMI, Tsukasa ABE and Hitoshi YOSHIMURA

Abstract

This paper describes the computer programs we have developed for the purpose of introducing the H8 microcomputer into the computer science laboratory. Students have learned so far the architecture of z80 microcomputer and its assembly language. As seen from the educational view, these techniques are very important for students to understand the fundamentals, but otherwise those have become classic. We have developed the computer programs for our students to understand the new microcomputer readily.

1. はじめに

情報工学科3学年の実験では、マイクロコンピュータによる各種制御用プログラミングの学習を目的として、従来、市販のZ80マイコン学習キットを用いてきた。しかし、Z80マイコンは既に古典的なアーキテクチャであり、また、本学習ボードは導入後10年を経過し、故障などが相次いでいる。以上の点を考慮し、平成15年度後期から、ルネサステクノロジー社のH8マイコンボードを新たに導入した。H8マイコンはタイマユニット、シリアル通信インターフェース、A/D、D/A変換器などがワンチップに組み込まれているマイコンであり、Z80に比べ多機能である反面、学生実験で用いるには少々複雑で理解し辛いという面を併せ持つ。また、前期の実験において、既にZ80マイコンを実験として用いているため、Z80マイコンに関する知識を生かして、効率的に理解できるようにすることが望ましい。

本稿では、H8マイコンボードの学習用プログラムを開発したので報告する。また、実験を行った学生の理解度について検討する。

2. 実験環境

2-1 実験用マイコンボードのハードウェア構成

今回、新たに導入したマイコンボードは、秋月電子通商のH8/3069F-LANマイコンボード、LCDユニット、I/Oボードである。H8/3069F-LANマイコンボードは、CPUとしてH8/300H（20MHz）を搭載しており、シリアル通信用インターフェース、A/D、D/A変換器、タイマユニットなどがワンチップに組み込まれている。メモリは、フラッシュROM（512kB）とRAM（16kB）が搭載されており、フラッシュROMは100回まで書き換え可能である。LCDユニットは、キャラクタジェネレータを内蔵したオックス電子製のSC1602BS*Bを用いており、プログラムの作成が容易に行えるようになっている。I/Oボードは、2つのLEDとディップスイッチが搭載されており、H8マイコンから制御することが可能である（図1参照）。

2-3 プログラムの開発環境

プログラムは、Windows上で動作するUNIX環境（Cygwin）で開発を行う。コンパイラとコンバータは、それぞれ、h8300-hms-gccとh8300-hms-objcopyを用いる。作成された実行ファイルは、シリアルポートを経由して、H8マイコンのRAMへアップロードし、モニタプログラムを経由して実行される。

* 助手 情報工学科

** 助教授 情報工学科

*** 教授 情報工学科

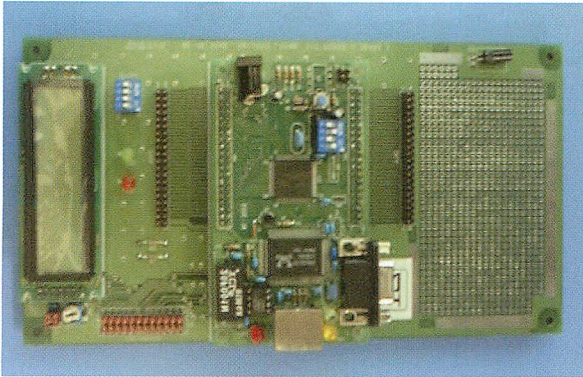


図1 実験用マイコンボード

3. 実験の計画

第3学年の情報工学実験では、1クラス40人の学生を4班に分けて、各々別テーマでローテーションを組んで行う形式となっている。実験は全12テーマで、そのうち、H8マイコン実習として割り当てられているのは6テーマである。学生は、今回の実験で初めてH8マイコンを扱うため、ハードウェアの概要、および、開発環境に関する概要について2週にわたって講義することにした。

表1 H8マイコンの実験計画

週	実験テーマ
第1週	H8マイコンの概要(1) ハードウェアの概要
第2週	H8マイコンの概要(2) C言語、開発環境
第3週	タイマによるLEDの制御
第4週	キーボードとディップスイッチの入力制御
第5週	LCDの制御
第6週	コンピュータ間通信

4. H8マイコン実験の概要

4-1 C言語の復習

マイコンのプログラムをC言語で開発するに当り、通常のプログラミングでは使用頻度の低い特殊な命令を用いる。第2週目において、C言語の復習を兼ねて解説を行った。

(1) 共用体

共用体は、ほとんどのテキストで解説されているが、構造体に比べ使用頻度は低いため、復習を兼ねて説明を行った。H8マイコンのプログラミングにおいて、共用体は、同一アドレスに対して異なった単位（バイト、ビット）でア

クセスするために用いる。

(2) ビット単位のアクセス

構造体を用いて、ビット単位のアクセスを可能とするには、各変数に対して“:1”を付与することで可能となる。以下にポート1を定義している共用体の一例を示す。

```
union un_port1 {
    unsigned char BYTE;
    struct {
        unsigned char B7:1;
        unsigned char B6:1;
        unsigned char B5:1;
        unsigned char B4:1;
        unsigned char B3:1;
        unsigned char B2:1;
        unsigned char B1:1;
        unsigned char B0:1;
    } BIT;
};
```

これは、BIT という構造体の中に、B0～B7 までの変数が定義されており、一番最初に定義されているB0はLSBを指し、B7はMSBを意味する。また、全体が共用体であるため、バイト単位のアクセス（BYTE）と、ビット単位のアクセス（BIT.B0～BIT.B7）を使い分けることが出来る。

(3) 修飾子 volatile

修飾子 volatile は、組み込み系のプログラムを作成する際に重要となる概念であるが、市販されているテキストにはほとんど解説されていない。volatile が付与された変数は、“揮発性”があるとみなし、メモリからアクセスした直後でのみ有効になるという意味である。これは、同一メモリの値を、プログラム以外の外部機器（ディップスイッチなど）が値を書き換えた場合、プログラムが過去にアクセスした値は無効となる。したがって、変数の値を参照する際は、常にメモリからアクセスする必要がある。これは、コンパイラによるプログラムの最適化を避けることで可能となり、修飾子 volatile を付与することで実現できる。

4-2 ハードウェアタイマによるLEDの制御

4-2-1 概要

ハードウェアタイマは、前期の実験で行った

Z80-CTC と基本的には同じ概念である。実験で行ったのは、2秒間隔で二つのLEDを交互に点灯させるプログラムを作成することであり、正確に2秒を刻むために、ITU（インテグレートド・タイマ・ユニット）の設定方法について説明した。H8/3069Fでは、16ビットのITUが3チャンネル、8ビットのITUが3チャンネルある。各々を結合させて用いることもでき、また、外部パルスのカウンタとしても用いることが出来る。今回は、ITUの基本的な原理を理解するために、16ビットのITUを1チャンネル（ITU0）のみ用いることとした。

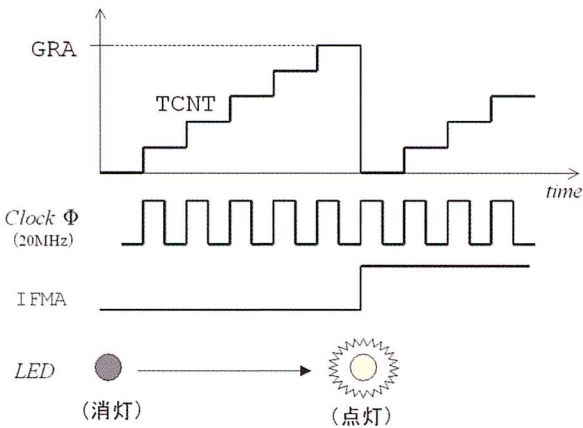


図2 ハードウェアタイマによるLEDの点灯制御

タイマの原理は、内部クロックパルス数をカウントしていき、予め設定されたカウンタの上限値に達すると0へ戻り、再びカウントを始めるという仕組みになっている（コンペアマッチ）。カウンタの値がクリアされたときフラグが立つので、フラグの値を常に監視し、1となったときに所望のプログラムを実行させる。これにより、一定周期の処理を行うことが出来る。タイマの初期設定、動作などは、すべてレジスタに値を代入することで実現できる。クロックパルスをカウントした値を保存しておくレジスタを、タイマカウンタ（TCNT）と呼び、カウンタの最大値（カウンタと比較する値）を記憶しておくレジスタを、ジェネラルレジスタ（GR）と呼ぶ。ジェネラルレジスタは、AとBの2種類あり、各チャンネル毎に用意されている。また、A、Bどちらを使用してもよい。

また、長時間のタイマを作成するために、内部クロックパルスの周波数をあらかじめ低く設定することが出来る（プリスケイリング）。

4-2-2 実験プログラム

実験で配布したのは0.8秒間隔でLED1とLED2を交互に点灯させるプログラムである。このプログラムにおいて、IMFA0はチャンネル0のコンペアマッチフラグを表し、この値が0の間は何も処理を行わず、1となったときにLEDを点灯させる。点灯した後は、フラグを0に戻し、再び1になったときにLEDを点灯させるというアルゴリズムになっている。

プログラムを以下に示す。

```
#include "../include/m3067f.h"

int main() {
    int c = 50;

    P4DDR = 0xff;
    ITU0.TCR.BYTE = 0x23;
    ITU0.GRA = 500;
    ITU.TSTR.BIT.STR0 = 1;

    P4DR.BYTE = 0x80;
    while(1) {
        while(!ITU.TISRA.BIT.IMFA0);
        c--;
        if(c==0) {
            P4DR.BIT.B6 = ~P4DR.BIT.B6;
            P4DR.BIT.B7 = ~P4DR.BIT.B7;
            c = 50;
        }
        ITU.TISRA.BIT.IMFA0 = 0;
    }
    return 0;
}
```

実験では3つの課題を学生に与えた。

- 1. サンプルプログラムを参考にして、2秒間隔でLED1とLED2を交互に点灯させるプログラムを作成せよ。
- 2. チャンネル1、ジェネラルレジスタBを用いて、(1)と同じ動作をするプログラムを作成せよ。
- 3. 以下の関数を作成せよ。

関数：void wait(int msec)

機能：msec で指定した時間だけウェイトをかける。ただし msec の単位はミリ秒とする。

課題1の実行例を図3に示す。

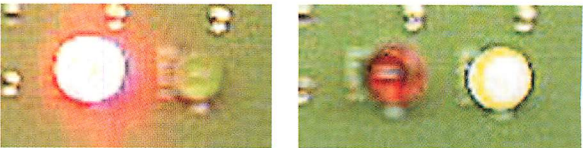


図3 課題1の実行例

4-2-3. 学生の理解度

ITU は、前期の実験で既に扱ったことのある z80-CTC と類似している部分が多い。そのため、ほとんどの学生は、理解しやすかったと答えており、また、LED の制御方法に関しても、実際にプログラムを作成し、動作させることにより、理解が深まったという学生が多かったので良好であるといえる。

4-3. キーボードとディップスイッチの入力制御

4-3-1. 概要

この実験では、キーボードおよびディップスイッチを用いて LED を制御するプログラムの作成を目的とする。キーボードからの入力を H8 側で受付ける関数として getchar, また、H8 から TeraTerm 端末上に文字を出力する関数として、putchar, printf をアセンブリ言語で作成した。putchar, printf は、シリアルポートを通じて PC ヘデータが送信される仕組みとなっている。具体的なアルゴリズムを理解するためには、シリアル通信に関する知識が必要となるため、今回の実験では説明を割愛した。

ディップスイッチからの入力は、スイッチ ON のとき 0, スイッチ OFF のとき 1 の値がデータレジスタ (DR) に書き込まれる。ただし、H8/3069F における入力モードは、ポート 2, 4, 5 のみ有効となることに注意する。スイッチ入力の際、ポートの各端子をプルアップさせることが必要となるが、H8マイコンではハードウェア的にプルアップさせる必要はなく、プルアップコントロールレジスタ (PCR) を用いることでソフトウェア的に制御可能である。

4-3-2. 実験プログラム

学生には、キーボード入力プログラムとディップスイッチ入力プログラムを作成した。プログラム(1)は、TeraTerm 上でキーボードから文字を入力すると、入力した値をそのまま TeraTerm 上に表示するというプログラムとなっている。プログラム(2)は、ディップスイッチの ON/OFF の状態がポート 5 のデータレジスタに格納されているため、その値を読み出して printf で出力させるプログラムである。

(1) キーボード入出力プログラム

```
#include "../include/m3067f.h"
#include "../common/stdio.h"
int main() {
    unsigned char  c;

    printf("press any key\n");
    while(1) {
        c = getchar();
        putchar(c);
    }
}
```

(2) ディップスイッチ入力プログラム

```
#include "../include/m3067f.h"
#include "../common/stdio.h"
int main() {
    char sw[4];
    P5DDR=0xf0;
    P5PCR.BYTE=0xff;    // PULL-UP
    while(1) {
        sw[0] = P5DR.BIT.B0;
        sw[1] = P5DR.BIT.B1;
        sw[2] = P5DR.BIT.B2;
        sw[3] = P5DR.BIT.B3;
        printf("%d%d%d%d\n",sw[0],sw[1],
[2],sw[3]);
    }
}
```

実験では、以下に示す 4 つの課題を学生に与えた。

1. キーボードからタイマの時間を入力することにより、任意の時間間隔で LED を点灯するプログラムを作成せよ。
2. ディップスイッチ sw1, sw2 が押されると、それに対応する LED1, LED2 が点灯するプログラムを作成せよ。
3. スペースキーを押してから、タイムをカウントしていき、1 秒毎に経過時間を出力するプログラムを作成せよ (2 分経過するまでカウントし、スペースキー以外は無効とせよ。)
4. キーボードから入力した 0 ~ 3 の値を、2 つの LED を使って 2 進表現するプログラムを作成せよ。0 ~ 3 以外のキーが入力された場合は無効とせよ。

4-3-3. 学生の理解度

この実験では、ほとんどの学生が課題のプログラムを完成させたが、一部の学生は、課題 3 のプ

プログラムを完成させることが出来なかった。すべての学生がプログラムを完成させることが出来るように、今後、課題の内容を検討していきたい。

4-4. LCD の制御

4-4-1. 概要

この実験では、液晶ディスプレイ (LCD) に文字を表示するプログラムを作成することを目的とする。今回の実験で用いる LCD は、制御ピンとデータバスで合計14本あり、H8マイコンのポート4と接続されている。LCDの初期設定は、以下の手順で行う必要がある。

- ① 15[ms] ウェイト
- ② 0x23を P4DR に書き込む (イニシャライズ)
- ③ 4.1[ms] ウェイト
- ④ 0x23を P4DR に書き込む (2 回目)
- ⑤ 0.1[ms] ウェイト
- ⑥ 0x23を P4DR に書き込む (IF データ長 8 ビット)
- ⑦ 0.04[ms] ウェイト
- ⑧ 0x22を P4DR に書き込む (IF データ長 4 ビット)
- ⑨ 0.04[ms] ウェイト
- ⑩ 0x28を P4DR に書き込む (FS)
- ⑪ ウェイト
- ⑫ 0x0e を P4DR に書き込む (ON/OFF コントロール)
- ⑬ ウェイト
- ⑭ 0x06を P4DR に書き込む (EM セット)
- ⑮ ウェイト
- ⑯ 0x01を P4DR に書き込む (表示クリア)
- ⑰ ウェイト

基本的な制御方法は、データモード (RS = 1) とした後で文字 (ASCII コード) を 4 ビットずつ P4DR に書き込むことである。使用できる文字は、数字とアルファベット、半角カタカナ、記号となっている。カーソルの位置を変更させるには、コマンドモード (RS = 0) とした後にデータを書き込む。

ポート4のDRは下位4ビットのみがデータバスとなっているので、データを書き込むときは、上位、下位の4ビットずつ2回書き込まなければならない。また、文字を表示するのに要する時間は、CPUクロックに比較してかなり遅いので、1つの値をレジスタに書き込んだ後、次の値を書

き込むまでに一定時間のウェイトが必要となる。

4-4-2. 実験プログラム

実験プログラムは、初期設定の後、"Hello H8"の文字列をLCDに表示させるものを作成した。RS = 1とした後、文字コードをLCDへ書き込み、その後、RS = 0とする。このような処理をすべての変数に対して行うことで、LCD上に表示することが可能となる。

```
int main(){
    int i;
    unsigned char s1[17];

    P4DDR=0xff;
    timer_init();
    lcd_init();
    P4DR.BIT.B4=0; // コマンドモード (RS=0)
    lcd_out4(0x01); // ディスプレイクリア
    wait(2);        // 1.64[ms]
    P4DR.BIT.B4=0; // コマンドモード (RS=0)
    lcd_out4(0x02); // カーソルをホームへ移動
    wait(2);        // 1.64[ms]
    s1[0] = 'H'; // LCDへ表示する文字列の設定
    s1[1] = 'e';
    s1[2] = 'l';
    s1[3] = 'l';
    s1[4] = 'o';
    s1[5] = ' ';
    s1[6] = 'H';
    s1[7] = '8';
    s1[8] = '\0';

    i=0;
    while(s1[i]){
        P4DR.BIT.B4=1; // RS=1
        lcd_out4(s1[i]); // LCDへ文字列を出力
        P4DR.BIT.B4=0; // RS=0
        i=i+1;
    }
    return 0;
}
```

学生には以下の課題を与えた。

1. キーボードから入力した文字をLCDに出力するプログラムを作成せよ。ただし、16文字を超えた場合は、自動的に2行目に改行して出力するプログラムを作成せよ。2行目が16文字に達したときにプログラムを終了せよ。

2. 以下の関数 lcd_char を作成せよ。

関数: void lcd_char(char s[], int row);

使用例: lcd_char("Hello H8", 1);

動作説明: LCDの1行目に "Hello H8" という文字が出力される。

3. 以下の関数 `lcd_int` を作成せよ。ただし、`x` は 4 桁の整数とする。

関数: `void lcd_int(int x, int row);`

使用例: `lcd_char(1234, 2);`

動作説明: LCD の 2 行目に 1234 という数値が出力される

課題 1 の実行例を図 4 に示す。



図 4 課題 1 の実行例

4-4-3. 学生の理解度

LCD の制御は、LED、スイッチ等の制御に比べて、少々複雑になっており、出来るだけ理解しやすいプログラムを作成したが、学生の理解度はそれほど高くなかった。特に、4 ビット入力モードに設定したときの制御方法に関して理解が十分でない学生が多く見られた。個々の関数の具体的な処理に関して、時間的な都合で説明を割愛したことが原因であると思われる。この点に関しては今後改善したい。

4-5. コンピュータ間通信

4-5-1. 概要

この実験では、調歩同期方式によるシリアル通信の原理について理解することを目的とする。調歩同期方式では、始めにデータの送信開始の合図として 0 が送信される（スタートビット）。文字データは、2 進数の下位ビットから順番に送信され、文字データの最後にエラーチェックのためのパリティビットが付与される。1 文字分の送信が終わると、最初の状態へ戻すためにストップビットを付与する。ストップビットは、1 ビット、または 2 ビットに設定可能であり、パリティは偶数パリティ、奇数パリティがそれぞれ設定可能であ

る。また、データ長を 8 ビットまたは 7 ビットに設定することも可能である。パルスの時間幅は、通信速度によって定まる。

H8 マイコンで、通信パラメータの設定を行うためには、SCI（シリアル通信インターフェース）を使用することになる。SCI は 3 チャンネルあるが、実験で用いる I/O ボードでは、チャンネル 1 がシリアルポートと接続されている。チャンネル 2 は、外部機器接続用コネクタ（CN2A-37）と繋がっている。

4-5-2. 実験プログラム

今回は、チャンネル 2 を使用してデータを送信するプログラムを実行し、CN2A-37 ピンからオシロスコープを使用して波形を記録することにした。受信に関しては実験を行わなかった。プログラムでは、分周率 $\Phi/8$ 、ストップビット 2、偶数パリティ、通信速度 9600 [bps] に設定している。

```
#include "../include/m3067f.h"

void main() {
    int count = 0;
    unsigned char c = '*';

    SCI1.SCR.BYTE = 0;
    SCI1.SMR.BYTE = 0;
    SCI1.BRR = 15;
    SCI1.SCR.BYTE = 0x30;
    SCI1.SMR.BYTE = 0x24;
    SCI1.SSR.BYTE;
    SCI1.SSR.BYTE = 0x80;
    while(++count < 10) {
        while(SCI1.SSR.BIT.TDRE == 0);
        SCI1.TDR = c;
        SCI1.SSR.BIT.TDRE = 0;
    }
}
```

学生に与えた課題は、以下のとおりである。

1. 送信データを 0x07, 0x28, 0x5a に変更し、波形を記録せよ。通信パラメータはキャラクタ長 8 ビット、パリティなし、ストップビット 2、ボーレートは 38400 とする。
2. ボーレートを 1200, 2400 に変更し、波形を記録せよ。通信パラメータはキャラクタ長 8 ビット、パリティなし、ストップビット 2、ホストに送信するデータは 0x3a とする。

課題 1 で観測した波形の一例を図 5 に示す。

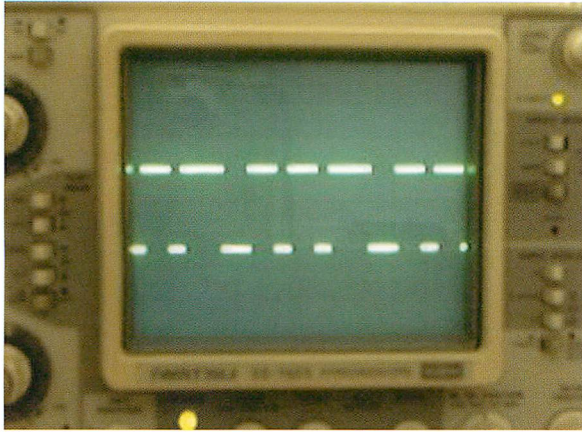


図 5. 観測波形の出力例

4-5-3. 学生の理解度

調歩同期方式の原理に関しては学生の理解度は高かった。SCI の原理と制御方法に関しては、時間の都合上、割愛せざるを得なかった。したがって、学生自身が通信プログラムを作成するまでには至らず、通信方式の原理を理解するのみに留まった。この点に関しては、今後検討していきたい。

6. おわりに

情報工学実験のための H8マイコン学習用プログラムを開発し、その目的と概要を説明した。また、実験報告書と実験時の様子から、学生の理解度について検討した。今後、アンケート等による調査を行うことで、学生の希望をできるだけ取り入れたテーマへの改訂、および実験計画の改善等を検討していきたい。

参考文献

1. ルネサステクノロジー(株)：H8/3067F ハードウェアマニュアル
2. 梅田：H8マイコン学習用テストプログラムの開発，平成14年度情報工学科卒業研究
3. 白土：H8ビギナーズガイド，東京電機大学出版（2001）
4. 堀：H8マイコン入門，東京電機大学出版（2003）
5. 特集・H8マイコン活用テクニック，トランジスタ技術2002年3月号，CQ 出版
(平成15年11月28日受理)

