

ゼロからVivliostyleでレポートを書くまで

②カスタマイズ編

2022/10/16

時価総額5000兆円株式会社

kai ware

Twitter: @kai_ware4job

目次

目的	1
マイコンの基本構造	1
実験手順	1
実験結果	2
追記	4

1 目的

C 言語を用い、LED やサーボ・ステッピングモータを制御するプログラムを作成し、ワンチップマイコン実験装置を用い動作確認を行う。また、A/D コンバータなどのマイコン内部の周辺回路動作についても理解する。これらの実験を通し、マイコン制御の基礎と開発環境などについて学ぶ。

2 マイコンの基本構造

一般的なマイコンはCPU・ROM・RAM・各種周辺回路で構成され、各ブロックは、アドレスバス、データバス、制御バスに繋がっている。

2.1 CPU

CPU (Central Processing Unit) はマイコンの頭脳とも呼べる部分である。主に制御・演算を行う部位で、ROM に書き込まれている。プログラムコード（ソフトウェア）の命令通りに動作する。

2.2 記憶装置

レジスタは CPU の内部にある記憶領域で、アクセス速度は高速だが、保持できる容量は少ない。CPU は、ROM や RAM から読み出したデータや、演算結果をこのレジスタに一時的に保持する。レジスタのデータをメモリに書き込むには、アドレスを元にしてメモリにアクセスする。

ROM (Read Only Memory) は電源を切っても中身のデータは消えない「不揮発性」メモリで、プログラム（ソフトウェア）や固定データが記憶されている。エミュレーターでこのメモリにプログラムを書込む。

RAM (Random Access Memory) は電源を切ると記録している内容が消える「揮発性」メモリで、データの一時的な保存に使用する。ROM と比べて容量は少ない。

2.3 周辺回路

汎用 I/O はマイコン外部との信号を送受信するのに使用する。**シリアル**は UART, USB などの通信を行う。**ADC・DAC** はアナログ/デジタル変換機能を持つ。**タイマ**は時間を計測する機能を持つ。**割り込みコントローラ**は通常のプログラム実行時に、通常外の特例の処理を行うきっかけをつくる。例えばスイッチが押された、時間が経過した、ADC が完了した等である。**メモリコントローラ**は外付けの大容量メモリの制御を行う。

3 実験手順

3.1 課題 1 LED の点滅

LED ドットマトリクス of LED を点滅させるプログラムを作成し、動作確認を行った。点滅周期は 0.5 秒とした。

3.2 課題 2 SW 入力

プッシュ SW を押下時に LED を点灯させるプログラムを作成し、動作確認を行った。プッシュ SW は SW1 を使用し、**P8_0** に接続して使用した。

3.3 課題 3 サーボモータの制御

サーボモータを回転させるプログラムを作成し、動作確認を行った。プッシュ SW1 押下時は左方向、プッシュ SW2 押下時は右方向とした。また、SW1 と SW2 の同時押しで基点に戻ることにした。

SERVO MOTOR を **PA_2** に接続して使用した。プッシュ SW1 は **P8_0**、SW2 は **P8_1** に接続して使用した。また、プログラム転送後に **SERVO MOTOR** のスイッチを ON にした。

このサーボモータはデューティ比が 14 % で +60°、10 % で 0°、6 % で -60°、10 % で基点の位置に移動する。これらを Table 1 に示す。

Table 1 デューティ比と H/L 期間

	14 % での期間 [s]	10 % での期間 [s]	6 % での期間 [s]
H 期間	2.1	1.5	0.9
L 期間	12.9	13.5	14.1

4 実験結果

4.1 課題 1 LED の点滅

LED マトリクスの左下にある LED が点滅した。周期は 0.5 秒より若干長かった。

Figure 1 app1.c

```
#include<3048f.h>
int main(void)
{
    P4.DDR=0xff;
    PB.DDR=0xff;

    PB.DR.BYTE=0x7f;

    int i, j;

    while(1) {
        P4.DR.BYTE=0x01;
        for(i=0; i<500; i++) {
            for(j=0; j<20000; j++);
        }
        P4.DR.BYTE=0x00;

        for(i=0; i<500; i++) {
            for(j=0; j<20000; j++);
        }
    }
    return 0;
}
```

```
}
```

4.2 課題 2 SW 入力

プッシュ SW 押下時、LED マトリックスの右端 1 列を除いた全ての LED が点灯した。

Figure 2 app2.c

```
#include<3048f.h>
int main(void)
{
    P4.DDR=0xff;
    PB.DDR=0xff;

    PB.DR.BYTE=0x01;

    while(1) {
        if(P8.DR.BIT.B0 == 0) {
            P4.DR.BYTE=0xff;
        } else {
            P4.DR.BYTE=0x00;
        }
    }
    return 0;
}
```

4.3 課題 3 サーボモータの制御

プッシュ SW1 押下時に左方向、プッシュ SW2 押下時に右方向に回転した。同時押しで基点に戻った。

Figure 3 app3.c

```
#include<3048f.h>

/* 待ち時間を作り出す場合、コンペアマッチなどの内蔵タイマを使用する方法と
for 文などにより待ち時間を作成する方法がある。 */
void nop(int s) {
    int i, j;
    for(i=0; i < s; i++) {
        for(j=0; j<2000; j++);
    }
}
```

```
}

int main(void)
{
    PA.DDR=0xff;
    P8.DDR=0xfc;
    int high_time = 15;

    while(1) {
        if(P8.DR.BIT.B0 == 0 && P8.DR.BIT.B1 == 0) {
            high_time = 15;
        } else if(P8.DR.BIT.B0 == 0) {
            high_time = 21;
        } else if(P8.DR.BIT.B1 == 0) {
            high_time = 9;
        }
        PA.DR.BIT.B2=1;
        nop(high_time);
        PA.DR.BIT.B2=0;
        nop(150 - high_time);
    }
    return 0;
}
```

5 追記

VFM is developed in the GitHub repository^{*1}. Issues are managed on GitHub^{*2}. Footnotes can also be written inline^{*3}.

*1 VFM↔

*2 Issues↔

*3 This part is a footnote.↔