

マイコン基礎及び演習

13EC060 武澤 裕介

2016 年 1 月 27 日

概要

AVR マイコンによる LED 及びモータの同時制御方法について述べ、考察し改善点を指摘する。

1 設計仕様

設計仕様は以下である。

- 停止状態：LED が 5 秒毎に点滅する。
- 高速状態：スイッチを押すと、LED は 0.5 秒毎に点滅する。モータは高速に回転する。
- 低速状態：5 秒後に LED は 2 秒毎に点滅するようになり、モータは低速に回転する。 さらにスイッチを押すと、停止状態になる。

2 使用機器

まず、今回使用する部品を表 1 に示す。

表 1 使用機器

使用機器名	メーカー	型番	使用個数
AVR マイコン	atmel	attiny2313	1
プログラマ	atmel	mk-	1
FET	東芝	2SK2231	1
モータ	マブチモータ	FA-130RA	1
ダイオード	パンジット	1N4007	1
基盤用タクトスイッチ	記載なし	汎用品	1
0.1[Ω]5[W] 抵抗	記載なし	汎用品	1
330[Ω] 抵抗	記載なし	汎用品	1
10[k Ω] 抵抗	記載なし	汎用品	2
22[k Ω] 抵抗	記載なし	汎用品	1
LED	記載なし	汎用品	1
電池ボックス (4.5[V])	記載なし	汎用品	1
電機ボックス (3.0[V])	記載なし	汎用品	1

3 接続図

今回使用した回路の回路図を図 1 に示す。

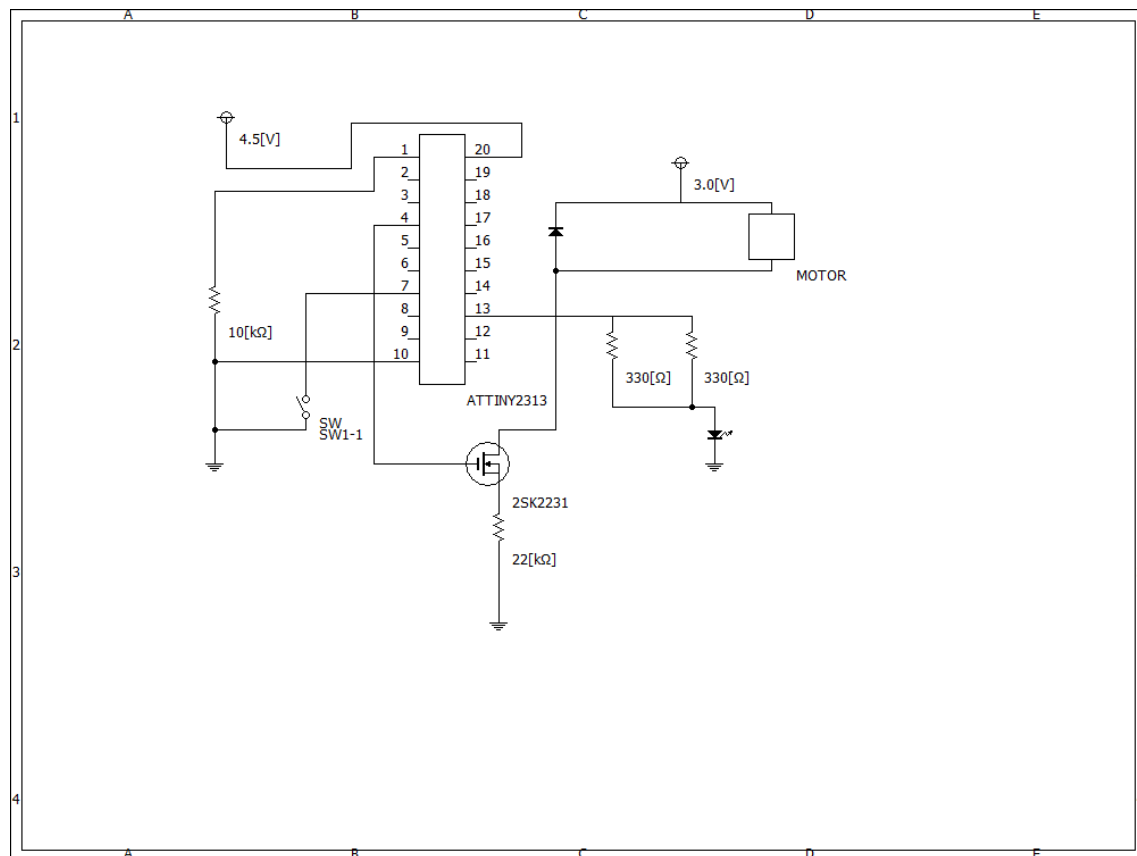


図 1 回路図

4 プログラムの説明

今回使用したプログラムについて説明する。今回 LED を PORTB で制御し、また PORTD の 2 ピンをモータの制御、4 ピンをスイッチからの電圧読み取りに使用している。また今回停止状態、高速状態、低速状態の遷移を制御する変数を st として定義している。図 2 のような状態の遷移となる。また st に 1 が代入された時にモータと LED を制御しその後 st に 2 を代入し 5 秒数えている。また変数 counter は 5 秒数えるための変数である。main 関数にて led 関数と mortor 関数を while 文にて実行した PORTD、PORTB の初期状態を設定している。led 関数は LED を制御、mortor 関数はモータを制御また、停止状態、高速状態、低速状態の LED の点滅時間を waitStop、waitSm、waitFast、waitSlow 関数で制御する。

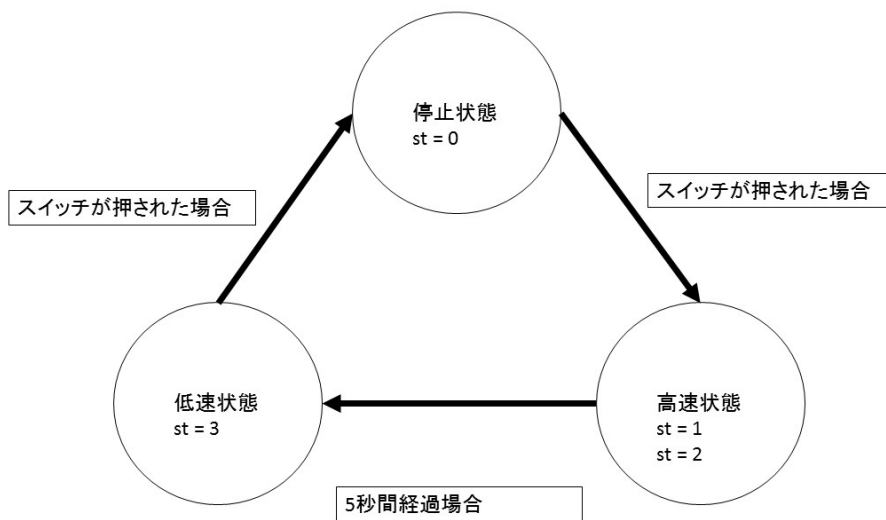


図 2 状態遷移図

5 考察

今回作成したコードでは長押ししなければ状態が遷移しないという事態が起きていたので改良する必要がある。また回路設計時に LED に過大な電流が流れるように、マイコンにも負荷がかかるように回路を組んでしまっていた。今後 FPGA ボードなどの高額な機械を扱う場合に対して注意する必要があるとされる。また一番苦労した点としてはハードウェアプログラミング特有のプログラムの言語仕様であった。実際 PORT 指定に関しても正直なところ最初は理解が及んでいなかった。また c 言語を今回利用したが int 型の違いや、return 文を main に対して使用しないのは驚いた。最後に今回の課題を通して基本的なマイコンの扱いまた回路の設計に対する技術が向上したと考えられる。また自分は ardino のみしかマイコン関係を触れたことがなかったので AVR マイコンに触れられたことは大変貴重な経験になったと思う。

6 付録 (プログラム)

```

#include <avr/io.h>

int8_t st = 0;
int8_t counter = 0;
void led(void);
void motor(void);
void waitSm(void);
void waitStop(void);
void waitFast(void);
void waitSlow(void);

```

```

int main(void){
    DDRB = 0xff;
    PORTB = 0xff;
    DDRD = 0xff;
    PORTD = 0;
    while(1){
        led();
        mortor();
    }
}

void led(void){
    switch(st){
        case 0:
            PORTB = 0xff;
            waitStop();
            PORTB = 0;
            waitStop();
            st = 0;
            while(PIND & (1<<4)){
                st = 1;
            }
            break;
        case 1:
            PORTB = 0xff;
            waitFast();
            PORTB = 0;
            waitFast();
            counter++;
            if (counter > 4)
                st++;
            break;
        case 2:
        case 3:
            PORTB = 0xff;
            waitSlow();
            PORTB = 0;
            waitSlow();
            break;
    }
}

void mortor(void){
    switch(st){
        case 0:
            break;
        case 1:
            PORTD = 0x02;
            break;
        case 2:
            if(PIND & (1<<4))
                st=0;
            counter=0;
            PORTD = 0;
            break;
    }
}

```

```

void waitSm(void){
    volatile int32_t i;
    for(i=0;i<1000;i++){
    }
}

void waitStop(void){
    volatile int32_t i;
    for(i=0;i<150000;i++){
    }
}

void waitFast(void){
    volatile int32_t i;
    for(i=0;i<15000;i++){
    }
}

void waitSlow(void){
    volatile int32_t i;
    for(i=0;i<30;i++){
        PORTD = 0xff;
        waitSm();
        PORTD = 0;
        waitSm();
        i++;
    }
}

```