

AVR入門4

到達目標

マイコンでステッピングモータが制御できる。

今回の共通コード

以下に示すプログラムは以降省略する。

ポートBをマイコンからステッピングモータの制御ボードへの出力用、ポートDをワーク6にて使用するボタンの入力用のポートと設定する。

```
INIT:
    LDI R16, 0b11111111
    OUT DDRB, R16
    LDI R16, 0b00000000
    OUT DDRD, R16
```

空白時間(c++でいうとdelay)を実現するためのプログラム。今回は、1秒遅らせる**T1S**と、R17レジスタで指定したmsを遅らせる**CUSTOM_MS**を使用した

```
CUSTOM_MS:
_T1MS:
    RCALL T1MS
    SUBI R17, 1
    BRNE _T1MS
    RET

T100US:
    LDI R21, 249
_TUS:
    NOP
    NOP
    NOP
    NOP
    NOP
    SUBI R21, 1
    BRNE _TUS
    NOP
    RET

T1MS:
    LDI R22, 10
```

```
_T:
    RCALL T100US
    SUBI R22, 1
    BRNE _T
    RET

T10MS:
    LDI R22, 100
_T100US:
    RCALL T100US
    SUBI R22, 1
    BRNE _T100US
    RET

T1S:
    LDI R23, 100
_T10MS:
    RCALL T10MS
    SUBI R23, 1
    BRNE _T10MS
    RET
```

実験

ワーク1

ステッピングモーターとそれに繋がれた制御出力ボードをマイコンと接続し、1相励磁方式を用いて、1ステップ1秒でステッピングモータを正回転させる

制御ボードにあるx, x_bar, y, y_barのポートの順にhighを流し込んだ。

```
ONE:
    LDI R16, 0b00000001
    OUT PORTB, R16
    RCALL T1S

    LDI R16, 0b00000010
    OUT PORTB, R16
    RCALL T1S

    LDI R16, 0b00000100
    OUT PORTB, R16
    RCALL T1S

    LDI R16, 0b00001000
    OUT PORTB, R16
    RCALL T1S

    RJMP ONE
```

ワーク2

2相励磁方式で回転させる（その他の条件はワーク1と同じである）

2つの連続したポートを同時にhighにすることで2相励磁方式を実現した。ただし動作としては、目視では特にこれといった変化がなかったため、事前に説明を受けていた速度が上がるというのはどういうことかな？と思った。

SECOND:

```
LDI R16, 0b00001001
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00000011
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00000110
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00001100
OUT PORTB, R16
RCALL T1S

RJMP SECOND
```

ワーク3

反回転させる(その他の条件はワーク1と同じである)

ポートをhighにする順番を逆にした。

MIRROR:

```
LDI R16, 0b00001000
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00000100
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00000010
OUT PORTB, R16
RCALL T1S

LDI R16, 0b00000001
OUT PORTB, R16
RCALL T1S
```

```
RJMP MIRROR
```

ワーク4

ワーク1のプログラムを変更して、1ステップ0.1秒で動作を観察する

事前に用意していたサブルーチン(レポートのはじめを参照)を使って0.1秒遅らせる用のサブルーチンに置き換えた。

```
ONE_100MS:
    LDI R17, 100
    LDI R16, 0b00000001
    OUT PORTB, R16
    RCALL CUSTOM_MS

    LDI R16, 0b00000010
    OUT PORTB, R16
    RCALL CUSTOM_MS

    LDI R16, 0b00000100
    OUT PORTB, R16
    RCALL CUSTOM_MS

    LDI R16, 0b00001000
    OUT PORTB, R16
    RCALL CUSTOM_MS

    RJMP ONE_100MS
```

動作としては、ワーク1よりも回転速度が上がった。

ワーク5

ワーク2のプログラムを変更し、1ステップ0.1秒で動作を観察する

ワーク4と同様である。

```
SECOND_100MS:
    LDI R17, 100
    LDI R16, 0b00001001
    OUT PORTB, R16
    RCALL CUSTOM_MS

    LDI R16, 0b00000011
    OUT PORTB, R16
    RCALL CUSTOM_MS

    LDI R16, 0b00000110
```

```
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00001100
OUT PORTB, R16
RCALL CUSTOM_MS

RJMP SECOND_100MS
```

ワーク6

通常は正回転を続け、ボタンを押している間反回転するプログラムを作成し、観察する

ポートDでボタンのインプットを受け取ってそれをもとに条件分岐をした。

```
BUTTON_CTR:
IN R18, PIND
DEC R18
BRNE MIRROR_100MS

ONE_100MS:
IN R18, PIND

DEC R18
BRNE MIRROR_100MS

LDI R17, 100
LDI R16, 0b00000001
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00000010
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00000100
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00001000
OUT PORTB, R16
RCALL CUSTOM_MS

RJMP BUTTON_CTR

MIRROR_100MS:
LDI R17, 100
LDI R16, 0b00001000
OUT PORTB, R16
RCALL CUSTOM_MS
```

```
LDI R16, 0b00000100
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00000010
OUT PORTB, R16
RCALL CUSTOM_MS

LDI R16, 0b00000001
OUT PORTB, R16
RCALL CUSTOM_MS

RJMP BUTTON_CTR
```

全体の感想・考察

ステッピングモーターを、制御ボードを通した使い方は概ね理解した。しかし、どのように1ステップ7.5度を実現しているのか疑問に思った。スライドの図解では90度に回転することしかできないのでは? と思ったが簡易的な解説がされているのかと勝手ながら予想した。