

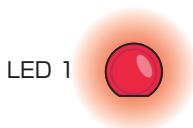
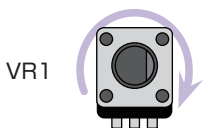
## PWM 制御

## 学習内容

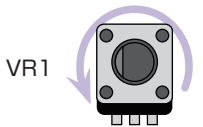
パルス信号の H レベルと L レベルの割合を変えることを PWM 制御といいます。PWM 制御は、LED の明るさ調節の他に DC モータの回転速度調節などに利用されます。ARM の PWM 制御を学習しましょう。

## 課題 10-1

以下のように、LED の明るさをボリュームで制御してみましょう。



VR1 を右に回すと、  
LED1 がだんだん明るくなる



VR1 を左に回すと、  
LED1 がだんだん暗くなる

## PWM 制御とは

パルス信号の H レベルと L レベルの割合を変えることを PWM 制御 (Pulse Width Modulation) といいます。PWM 制御は、LED の明るさ調節の他に DC モータの回転速度調節などに利用されます。

1 周期あたりの H レベルの割合をデューティ比と呼び、以下の式で求めることができます。

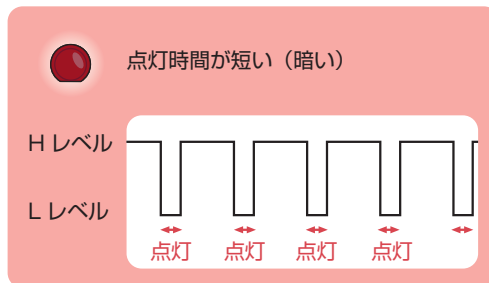
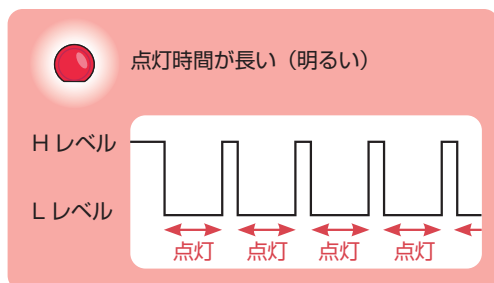
$$\text{デューティ比} = \frac{\text{H レベルの時間}}{\text{L レベルの時間} + \text{H レベルの時間}} \times 100$$

例えば H レベルの時間が 8 ミリ秒、L レベルの時間が 2 ミリ秒のパルス信号の場合、デューティ比は以下ようになります。

$$\text{デューティ比} = \frac{8 \text{ ミリ秒}}{2 \text{ ミリ秒} + 8 \text{ ミリ秒}} \times 100 = 80\%$$

## PWM 制御

LED は H レベルで消灯するため、デューティ比が低いと明るく、高いと暗く点灯します。



## TM4C123GH6PM の PWM 制御について

TM4C123GH6PM は 2 つの PWM モジュールを持っており、それぞれが 4 つの PWM ジェネレータブロックと 1 つのコントロールブロックを持つことにより、MOPWMO ~ MOPWM7, M1PWMO ~ M1PWM7 の計 16 個の PWM 信号を出すことができます。ただし、各モジュールの PWM 信号は端子を一部共有しているため、同時に使える PWM 信号の数はより少なくなります。

## TM4C123GH6PM のピンのデジタル機能

TM4C123GH6PM は、限られた数のピンで豊富なデジタル機能を提供するため、大半のピンに複数のデジタル機能が備わっています。

たとえば PDO ピンには、SSI3Cik, SSI1Cik, I2C3SCL, MOPWM6, M1PWMO, WT2CCPO の 6 つのデジタル機能が備わっており、使いたいデジタル機能を選択することができます (STEP 01 のピンアサインには、MOPWM6, M1PWMO, WT2CCPO の 3 つを抜粋して載せています)。

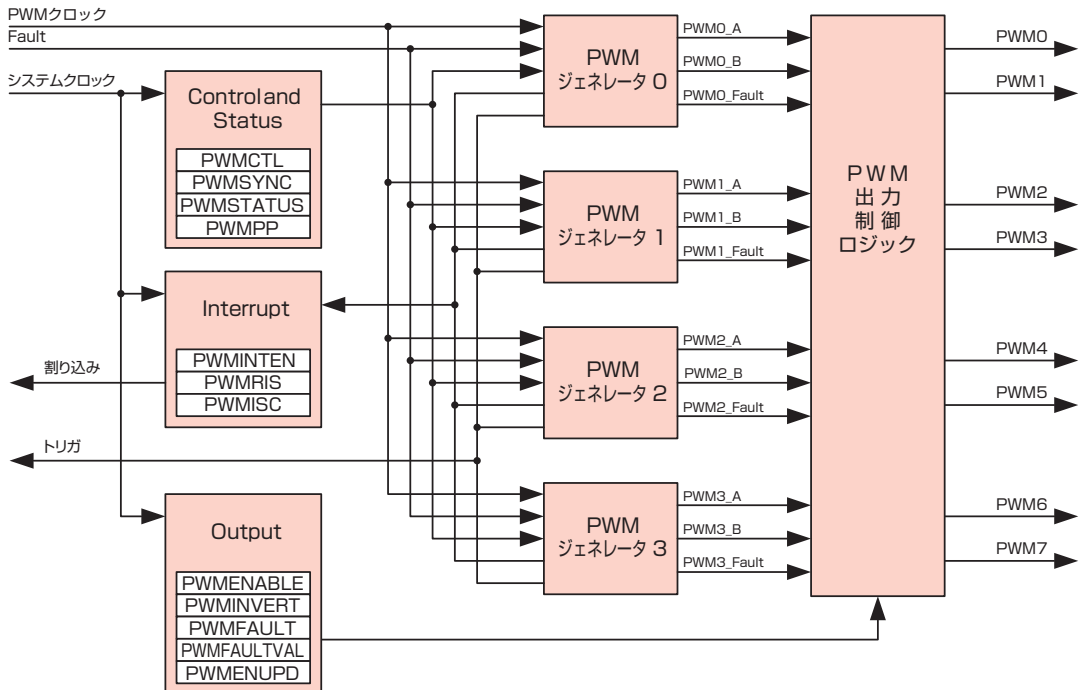
今回は、STEP 09 で LED1 に接続していた PDO ピンが PWM 機能(MOPWM6 および M1PWMO)を持っているので、MOPWM6 信号を用いることにします。LED1 を PDO 以外に接続している場合は、STEP 01 のピンアサインを参考に、用いる PWM 信号を設定してください (PWM 機能を持たないピンに接続している場合は、PWM 機能を持った適当なピンにつなぎ換えてください)。

なお、STEP 09 まで使った単純な端子の入出力は、これらのデジタル機能にかかわらず使うことができます。

また、アナログ入力もデジタル機能とは別のもので。

PWM 制御

PWM 機能のブロック図



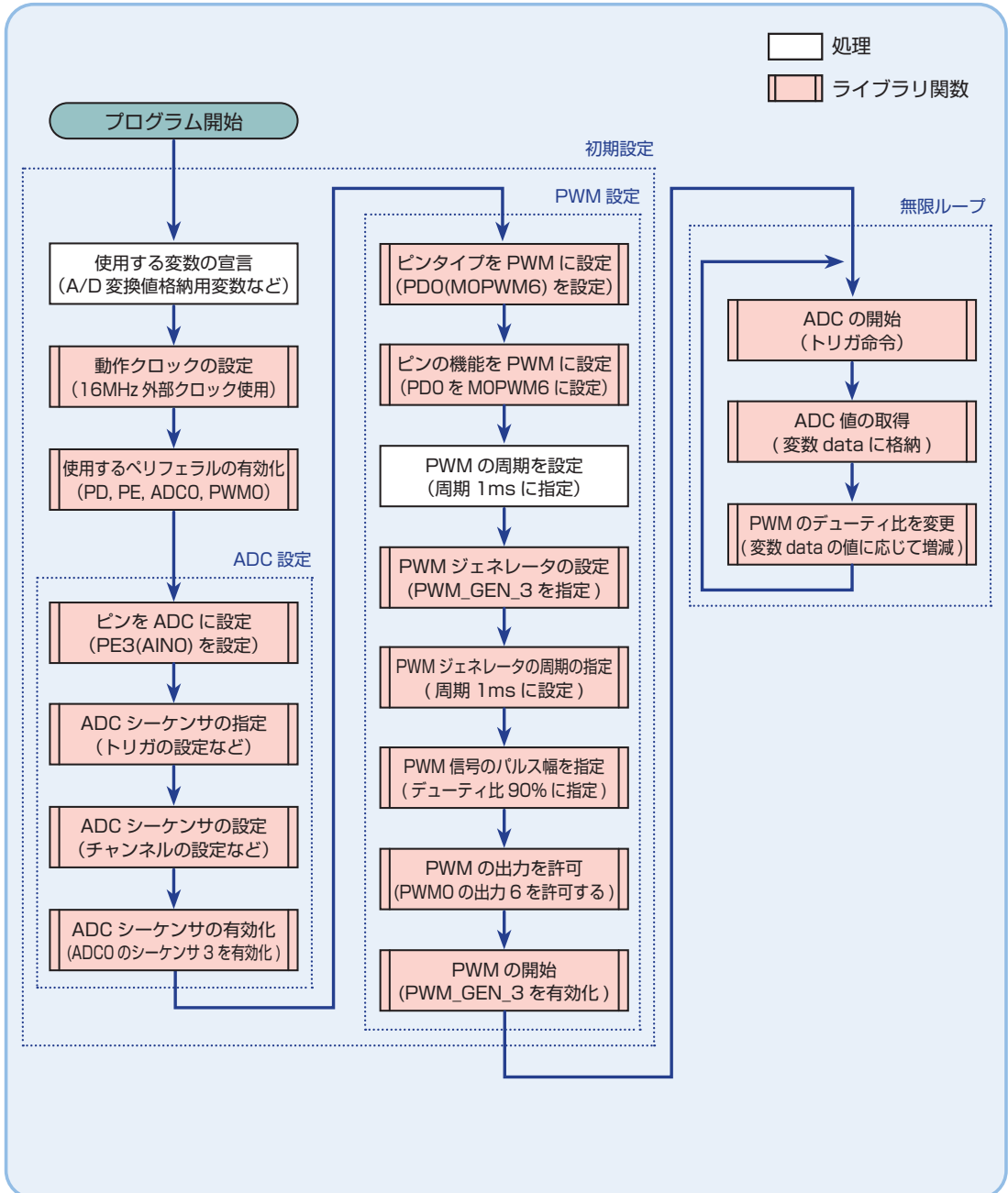
配線 10-1

「配線 09-1」と同じです。

## PWM 制御

### フローチャート 10-1

フローチャートは以下ようになります。



## PWM 制御

### インクルードファイル 10-1

STEP 10-1 で使用するインクルードファイルを解説します。

#### Tiva ヘッダファイル driverlib/pin\_map.h

```
#include "driverlib/pin_map.h"
```

Tiva のヘッダファイルで、ピンのデジタル機能を指定するためのマクロが宣言されています。次に示すライブラリ関数の設定例では、以下のマクロがここで宣言されています。

- GPIO\_PDO\_MOPWM6

#### Tiva ヘッダファイル driverlib/pwm.h

```
#include "driverlib/pwm.h"
```

Tiva のヘッダファイルで、PWM のためのマクロおよび関数が宣言されています。次に示すライブラリ関数の設定例では、以下のマクロおよび関数がここで宣言されています。

- PWM\_GEN\_0
- PWM\_GEN\_MODE\_UP\_DOWN
- PWM\_GEN\_MODE\_NO\_SYNC
- PWM\_OUT\_0
- PWM\_OUT\_0\_BIT
- PWMGenConfigure(ui32Base, ui32Gen, ui32Config)
- PWMGenPeriodSet(ui32Base, ui32Gen, ui32Period)
- PWMPulseWidthSet(ui32Base, ui32PWMOut, ui32Width)
- PWMOutputState(ui32Base, ui32PWMOutBits, bEnable)
- PWMGenEnable(ui32Base, ui32Gen)

## PWM 制御

## ライブラリ関数 10-1

STEP10 で使用するライブラリ関数を解説します。

関数の引数は ui32 ~ が「符号無し 32 ビット整数」です。ライブラリ関数は TivaWare で提供されています

## PWM 出力ピンの設定 GPIOPinTypePWM(ui32Port, ui8Pins)

指定したピンを PWM 機能に合うよう設定する。ただし、ピンのデジタル機能は多重化されているため、PWM を使うには別途 GPIOPinConfigure() でピンの機能を PWM に設定する必要がある。

```
GPIOPinTypePWM(GPIO_PORTD_BASE, GPIO_PIN_0);
```

## 設定項目 例

- GPIO\_PORTD\_BASE : ピンが属しているポートの設定値。 例はポート D。
- GPIO\_PIN\_0 : ピン番号の設定値。0 ~ 7 まで指定できる。 例は 0 番ピン。

## 多重化されたピンの機能設定 GPIOPinConfigure(ui32PinConfig)

ピンに備わっているデジタル機能の中から、使用する機能を指定する。

```
GPIOPinConfigure(GPIO_PD0_M0PWM6);
```

## 設定項目 例

- GPIO\_PD0\_M0PWM6 : ピンの機能を指定。 例は PDO ピンの機能を MOPWM6 に設定する。

## PWM ジェネレータの指定や諸設定 PWMGenConfigure(ui32Base, ui32Gen, ui32Config)

使用する PWM ジェネレータの指定や、PWM ジェネレータの設定を行う。

```
PWMGenConfigure(PWM0_BASE, PWM_GEN_3, PWM_GEN_MODE_UP_DOWN | PWM_GEN_MODE_NO_SYNC);
```

## 設定項目 例

- PWM0\_BASE : 使用する PWM モジュールを指定。
- PWM\_GEN\_3 : 使用する PWM ジェネレータを指定。  
PWM ジェネレータは 0 ~ 3 までである。  
1 つのジェネレータで 2 つのピンから PWM 信号を出力できる。
- PWM\_GEN\_MODE\_UP\_DOWN | PWM\_GEN\_MODE\_NO\_SYNC : PWM ジェネレータの設定。
  - ・ PWM\_GEN\_MODE\_UP\_DOWN : アップダウンカウントモードで使用する。
  - ・ PWM\_GEN\_MODE\_NO\_SYNC : 即時更新をしない。

## PWM 制御

### PWM ジェネレータの周期設定 PWMGenPeriodSet(ui32Base, ui32Gen, ui32Period)

PWM ジェネレータに周期を設定する。

```
PWMGenPeriodSet(PWM0_BASE, PWM_GEN_3, ulPeriod);
```

#### 設定項目 例

- PWM0\_BASE : 使用する PWM モジュールを指定。  
PWMGenConfigure で設定したものと同じにすること。
- PWM\_GEN\_3 : 使用する PWM ジェネレータを指定。  
PWMGenConfigure で設定したものと同じにすること。
- ulPeriod : PWM 信号の周期を設定する。設定に関してはタイマと同様。

### PWM 信号のパルス幅設定 PWMPulseWidthSet(ui32Base, ui32PWMOut, ui32Width)

PWM 信号のパルス幅 (ON 時間) を設定する。

```
PWMPulseWidthSet(PWM0_BASE, PWM_OUT_6, ulPeriod * 9 / 10);
```

#### 設定項目 例

- PWM0\_BASE : 使用する PWM モジュールを指定。  
PWMGenConfigure で設定したものと同じにすること。
- PWM\_OUT\_6 : 実際に PWM 信号を出力する PWM ピン番号。  
GPIOPinTypePWM で設定したものと同じにすること。
- ulPeriod \* 9 / 10 : パルス幅を指定する項目。  
PWMGenPeriodSet で設定した周期より短くなるよう設定する。  
例はデューティ比 90%

## PWM 制御

## ライブラリ関数 10-1

## PWM 信号の有効 / 無効 PWMOutputState(ui32Base, ui32PWMOutBits, bEnable)

PWM 信号の出力を有効 / 無効にする。

```
PWMOutputState(PWM0_BASE, PWM_OUT_6_BIT, true);
```

## 設定項目 例

- PWM0\_BASE : 使用する PWM モジュールを指定。  
PWMGenConfigure で設定したものと同じにすること。
- PWM\_OUT\_6\_BIT : 実際に PWM 信号を出力する PWM ピン番号。  
GPIOPinTypePWM で設定したものと同じにすること。
- true : PWM 信号の出力を有効 / 無効を指定する項目。  
true で有効, false で無効にする。

## PWM 信号の出力 PWMGenEnable(ui32Base, ui32Gen)

PWM ジェネレータを有効にする。この関数を実行することで PWM 信号が出力されるようになる。

```
PWMGenEnable(PWM0_BASE, PWM_GEN_3);
```

## 設定項目 例

- PWM0\_BASE : 使用する PWM モジュールを指定。  
PWMGenConfigure で設定したものと同じにすること。
- PWM\_GEN\_3 : 実際に PWM 信号を出力する PWM ピン番号。  
PWMGenConfigure で設定したものと同じにすること。



## PWM 制御

### コーディング 10-1

フローチャートを元に、ソースを記述してください。ソースが完成したら、実行して動作を確認しましょう。  
以下に解答例ソースを示します。解答例やサンプルソースを参考に、皆さんで工夫してみてください。

step10-1.c

CD-ROM の「サンプルソース」フォルダに、各ステップの c ファイルを収録しています

```

1 #include <stdint.h>
2 #include <stdbool.h>
3 #include "inc/hw_types.h"
4 #include "inc/hw_memmap.h"
5 #include "driverlib/gpio.h"
6 #include "driverlib/sysctl.h"
7 #include "driverlib/adc.h"
8 #include "driverlib/pin_map.h"
9 #include "driverlib/pwm.h"
10
11 // ボリューム入力による LED の PWM 制御
12 void main(void) {
13     // A/D 変換値格納用
14     uint32_t data;
15     // 周期設定用変数
16     unsigned long ulPeriod;
17     // 待ち時間用
18     volatile unsigned long Loop;
19     // 動作クロックの設定
20     SysCtlClockSet(SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN | SYSCTL_XTAL_16MHZ);
21
22     // 使用するペリフェラルの宣言
23     // : LED 用に I/O ポート D を, アナログ入力用にポート E を利用
24     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
25     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
26     // : A/D 変換モジュール 0 を利用
27     SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
28     // : PWM モジュール 0 を利用
29     SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);
30
31     // ADC の設定
32     // : PE3(AIN0) を ADC に設定
33     GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
34     // : 使用する ADC シーケンサの指定
35     ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);
36     // : ADC シーケンサの設定
37     ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_IE | ADC_CTL_END);
38     // : ADC シーケンサを有効にする
39     ADCSequenceEnable(ADC0_BASE, 3);

```

## PWM 制御

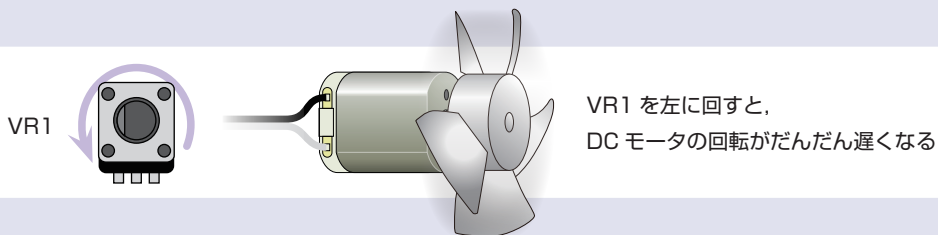
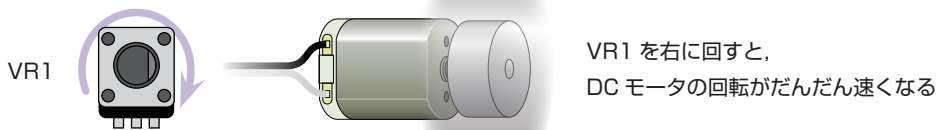
## コーディング 10-1

```
40
41 // PWM の設定
42 // : PD0(M0PWM6) を PWM に設定
43 GPIOPinTypePWM(GPIO_PORTD_BASE, GPIO_PIN_0);
44 // : PD0 を M0PWM6 に設定
45 GPIOPinConfigure(GPIO_PD0_M0PWM6);
46 // : PWM の周期を指定 周期: 1ms
47 ulPeriod = SysCtlClockGet() / 1000;
48 // : PWM ジェネレータの設定
49 PWMGenConfigure(PWM0_BASE, PWM_GEN_3, PWM_GEN_MODE_UP_DOWN | PWM_GEN_MODE_NO_SYNC);
50 // : PWM ジェネレータの周期を設定
51 PWMGenPeriodSet(PWM0_BASE, PWM_GEN_3, ulPeriod);
52 // : PWM 信号のパルス幅の指定
53 PWMPulseWidthSet(PWM0_BASE, PWM_OUT_6, ulPeriod * 9 / 10);
54 // : PWM0 の出力を許可
55 PWMOutputState(PWM0_BASE, PWM_OUT_6_BIT, true);
56 // : PWM 開始 (PWM ジェネレータの有効化)
57 PWMGenEnable(PWM0_BASE, PWM_GEN_3);
58
59 while (1) {
60     // A/D 変換開始
61     ADCProcessorTrigger(ADC0_BASE, 3);
62     // A/D 変換値の取得
63     ADCSequenceDataGet(ADC0_BASE, 3, &data);
64     // A/D 変換の値に応じてデューティ比を変更
65     PWMPulseWidthSet(PWM0_BASE, PWM_OUT_6, ulPeriod * (4096 - data) / 4097);
66 }
67 }
```

PWM 制御

課題 10-2

以下のように、DC モータの回転速度をボリュームで制御してみましょう。

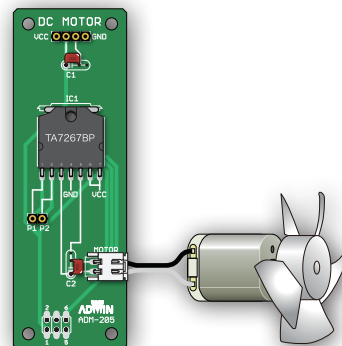


解答は、巻末の解答例集参照

配線 10-2

今回はマイコンボードとボリュームボード、DC モータボードを使用します。これらのボードはベースボード上で以下のように接続されています。DC モータボードの配線や制御方法について詳しくは、巻末に掲載しているエレモの取扱説明書をご覧ください。

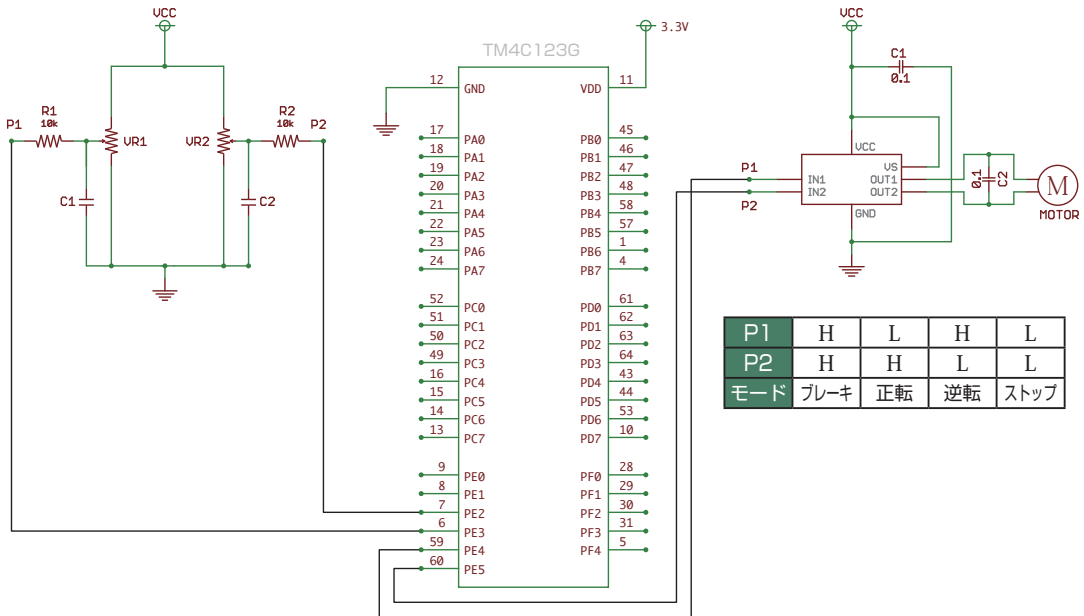
ボリュームボード		マイコンボード		DC モータボード
P1 - VR1	↔	3.09	PE3 (AIN0)	
P2 - VR2	↔	3.08	PE2 (AIN1)	
		1.05	PE4 (MOPWM4)	↔ P1 - IN1
		1.06	PE5 (MOPWM5)	↔ P2 - IN2
VCC	↔	1.01	+3.3V	
GND	↔	2.01	GND	↔ GND
		3.01	VBUS (+5V)	↔ VCC



マイコンボードと各ボードをベースボードから取り外し、ジャンパワイヤ（別売）で直接接続することも可能です。時間に余裕のある場合は、STEP 01 のピンアサインに注意しつつ、お好みのピンで課題を実現してみましょう。GND はマイコンボードの 2.01 ピン、3.02 ピンのどちらでもかまいません。

## PWM 制御

回路図



PWM 制御

課題 10-3

この課題は応用課題です。時間に余裕のある場合はチャレンジしてみてください。

STEP 01 のピンアサインを見てみると、マイコンボード上の LED も PWM 制御が可能であることが分かります。

ピン番号	GPIO	アナログ	ボード上の機能等	MPU ピン	デジタル機能 (抜粋)		
					4 注1	5 注1	6 注1
3.10	PF1	—	LED (赤)	29	—	M1PWM5	TOCCP1
4.01	PF2	—	LED (青)	30	MOFAULT0	M1PWM6	T1CCP0
4.02	PF3	—	LED (緑)	31	—	M1PWM7	T1CCP1

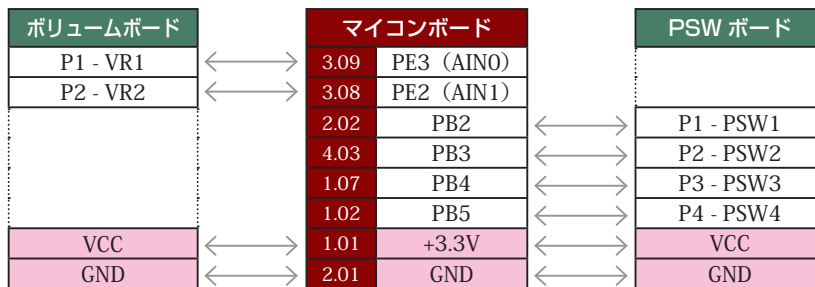
注1 : GPIOCTL レジスタ設定値

スイッチとボリュームを使い、マイコンボード上の LED を PWM 制御し点灯色を変化させてみましょう。スイッチとボリュームをどのように使うかは自由です。工夫してみてください。

解答は、巻末の解答例集参照

配線 10-3

今回はマイコンボードとボリュームボード、PSW ボードを使用します。これらのボードはベースボード上で以下のように接続されています。



マイコンボードと各ボードをベースボードから取り外し、ジャンプワイヤ (別売) で直接接続することも可能です。時間に余裕のある場合は、STEP 01 のピンアサインに注意しつつ、お好みのピンで課題を実現してみましょう。GND はマイコンボードの 2.01 ピン、3.02 ピンのどちらでもかまいません。