Advanced Sheet（発展）

ここでは発展的な内容をあつかっています。より深いところまで学びたい人は挑戦してみましょう。

型キャスト

小数をあつかえる double 型の変数を学びました。では，整数しか扱えない int 型の変数と同じ式で一緒に使うとどうなるでしょうか。

```
int main() {
    int a = 3;
    int b = 2;
    double c = a / b;
    printf("%lf", c);
}
```

int 型の a = 2 と int 型の b = 3 を定義し，double 型の c に c = a / b として代入します。c は $\frac{3}{2}=1.5$ となるように思えますが，実は小数が切り捨てられて 1 となってしまいます。なぜなら，int 型同士の割り算 a / b の結果は int 型になるという決まりがあるからです。これを防ぐには変数の前に(double)のように書く，**型キャスト**という操作をします。例のように書くと，a,b は割り算の前に double 型に変換され，c には 1.5 が代入されます。

```
int main() {
    int a = 3;
    int b = 2;
    double c = (double)a / (double)b;
    printf("%lf", c);
}
```

for 文を使ったグラデーション点灯

実践課題 問 4 では数段階の明るさを切り替えることで LED のグラデーション点灯をしていました。ここでは，for 文を使って明るさを 100 段階ほど切り替えてみましょう。

発展課題 1

for 文を使い LED を 100 段階のグラデーションで点灯させてみよう。

ヒント: for 文で i を 1~100 まで増やす。しかし PwmOut は 0~1 でしか出力できないから毎回 100 で割る。

(解答例は最後のページ)

scanf 関数

printf 関数を使うと Tera Term に文字や値を表示することができました。scanf 関数を使うと，逆に Tera Term から文字や値を入力することができます。右の例では，Tera Term から整数を入力して Enter キーを押すと，入力した数字を表示することができます。また，scanf 関数では変数の前に&をつけて書くことに注意しましょう。

```
int a;
scanf("%d", &a);
printf("%d", a);
```

発展課題 2

scanf 関数を使って Tera Term から LED の明るさを変えられるようにしてみよう。

ヒント: 明るさの調節は PwmOut と 0~1 の数を使う。

関数の引数と戻り値 (これは次回の授業でも取り扱います)

授業で関数を習いましたが、実は関数には引数と戻り値というものがあります。さっそく下の例を見てみましょう。

```
1  #include "mbed.h"
2
3  int sample(int arg);
4
5  int main(){
6      int num = 0;
7      num = sample(4);
8      printf("num = %d", num);
9  }
10
11 int sample(int arg){
12     int re;
13     re = arg + 6;
14     return re;
15 }
16
```

sample という関数を宣言しています。引数とは関数のカッコ内の値のことで、関数に値をわたすことができます。関数のカッコ内には引数の型と、それを代入する変数名を記述する必要があります。それではプログラムを順に追っていきましょう。main 文で引数を 4 に sample 関数を呼び出します。sample 関数は受け取った値を int 型の arg に代入します。さてここからは sample 関数の動作です。引数を受け取った sample 関数は、それに 6 を足して re という変数に代入しています。最後の return は関数の呼び出し元に値を返します。この場合戻り値は re ということになります。そして動作は main 文に戻り、num に戻り値である re の値が代入されます。なので printf で表示される数字は 10 ということになります。

「int sample(int arg)」の先頭の int はその関数の戻り値の型と同じでなければなりません。戻り値がない場合は void と書きましょう。「void」というのは「空(から)」という意味です。関数に引数がない場合も(void)と書くか、省略して()と書いても動作します。

引数は「,」で区切ることでいくつでも指定できますが、戻り値として return できるのは 1 つだけなので注意しましょう。

発展課題 3

モーターの回転速度と回転方向がだんだん変化するプログラムを作ってみよう。(モーターの代わりに LED でも可)

・motorA という関数を作ること。引数はモーターの速さで、-100 から 100 までの int 型整数。マイナスとプラスでは回転方向が逆になる。戻り値は無し(void)。

・つまり main 文では motorA 関数を 200 回呼び出すことになる。

ヒント：motorA 関数内では受け取った-100～100 の値を PWM で使える範囲に計算しなければならない。int 型(整数)と double 型(小数)の違いに注意！

発展課題 1 解答例

```
#include "mbed.h"

PwmOut led(A0);

int main() {
    while(1){
        for(int i = 1; i <= 100; i++){
            led = (double)i / (double)100;
            // double型に型キャストすることが大切！
            wait(0.01);
        }
    }
}
```

発展課題 2 解答例

```
#include "mbed.h"

PwmOut led(A0);

int main() {
    while(1){
        double a;
        scanf("%lf", &a);
        printf("%lf\r\n", a);
        // 入力した数字を表示すると分かりやすい
        led = a;
    }
}
```

発展課題 3 解答例

```
#include "mbed.h"

PwmOut MotorA1(D6);//モータードライバーのAin1を設定
PwmOut MotorA2(D9);//モータードライバーのAin2を設定
void motorA(int speed);//モーター制御関数speed:-100~100の値を入力できる。負の数で逆回転
int main()
{
    while(1) {
        for(int i=0; i<200; i++) {
            motorA(i-100);
            wait(0.1);
        }
    }
}
void motorA(int speed)//モーターを回す関数
{
    //入力値が-100～100を超えないためのバグ対策
    if(speed>100) speed=100;
    else if(speed<-100) speed=-100;
    //モーター回転
    if(speed>=0) {
        MotorA1=(double)speed/100;
        MotorA2=0;
    } else if(speed<0) {
        MotorA1=0;
        MotorA2=(double)-speed/100;
    }
    printf("speed= %d %%¥n",speed); //出力をprintf
}
```