

プログラムを作ろう！

~mbedを使ったプログラミング~
テーマは「光」

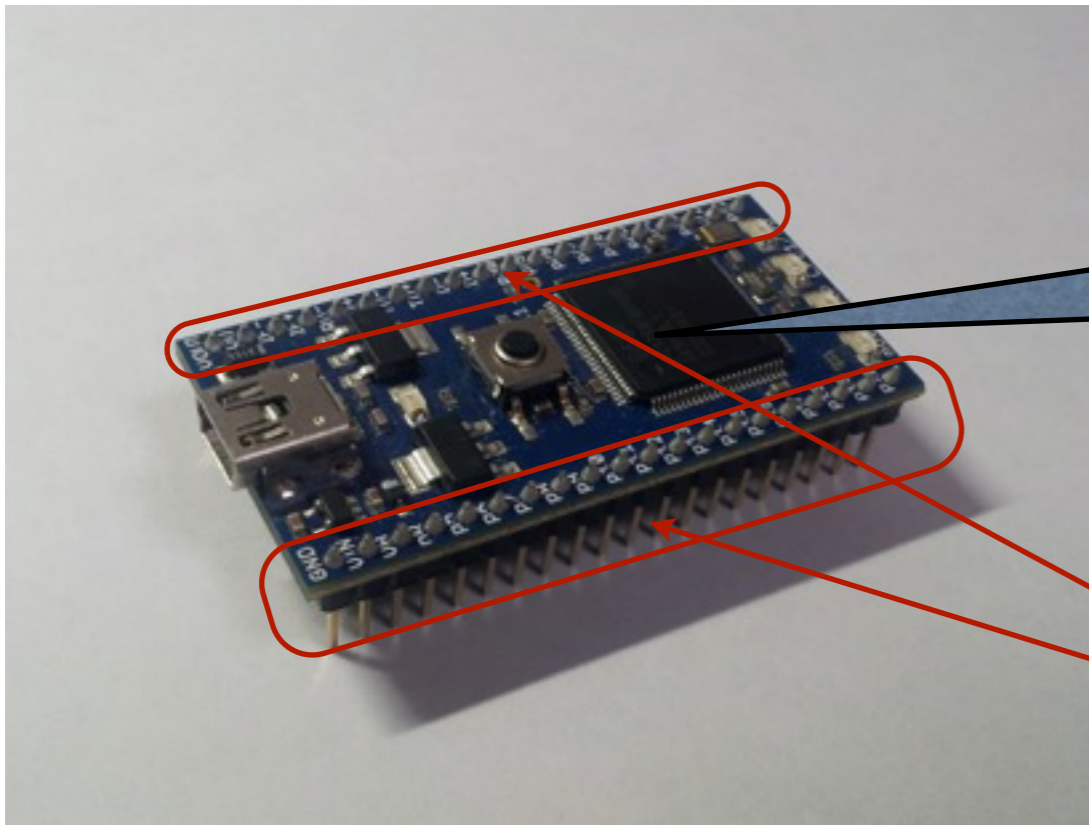
今日の内容

- 9:00~ デモンストレーション、作成例
- 9:15~ プログラミング体験 (LED、センサーなど)
- 10:00~ 休憩
- 10:15~ プログラミング体験 (パソコンへの表示や入力)
- 11:00~ 休憩
- 11:15~ アンケート記入、交流会

さっそくはじめよう！

プログラムって何？

プログラムは、パソコンの頭脳を動かす命令

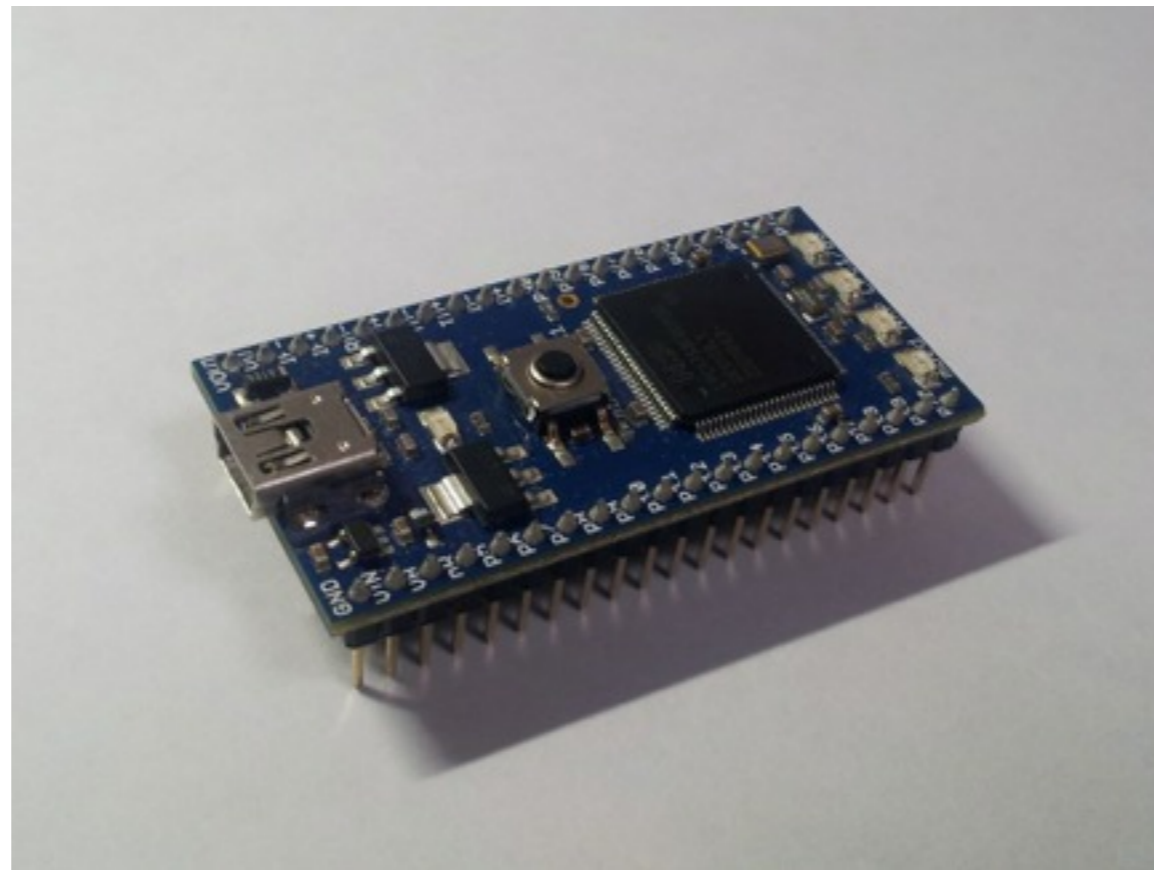


ここが「脳」
CPU

人間でいう神経
ここにセンサーをつ
ける

mbedって何？

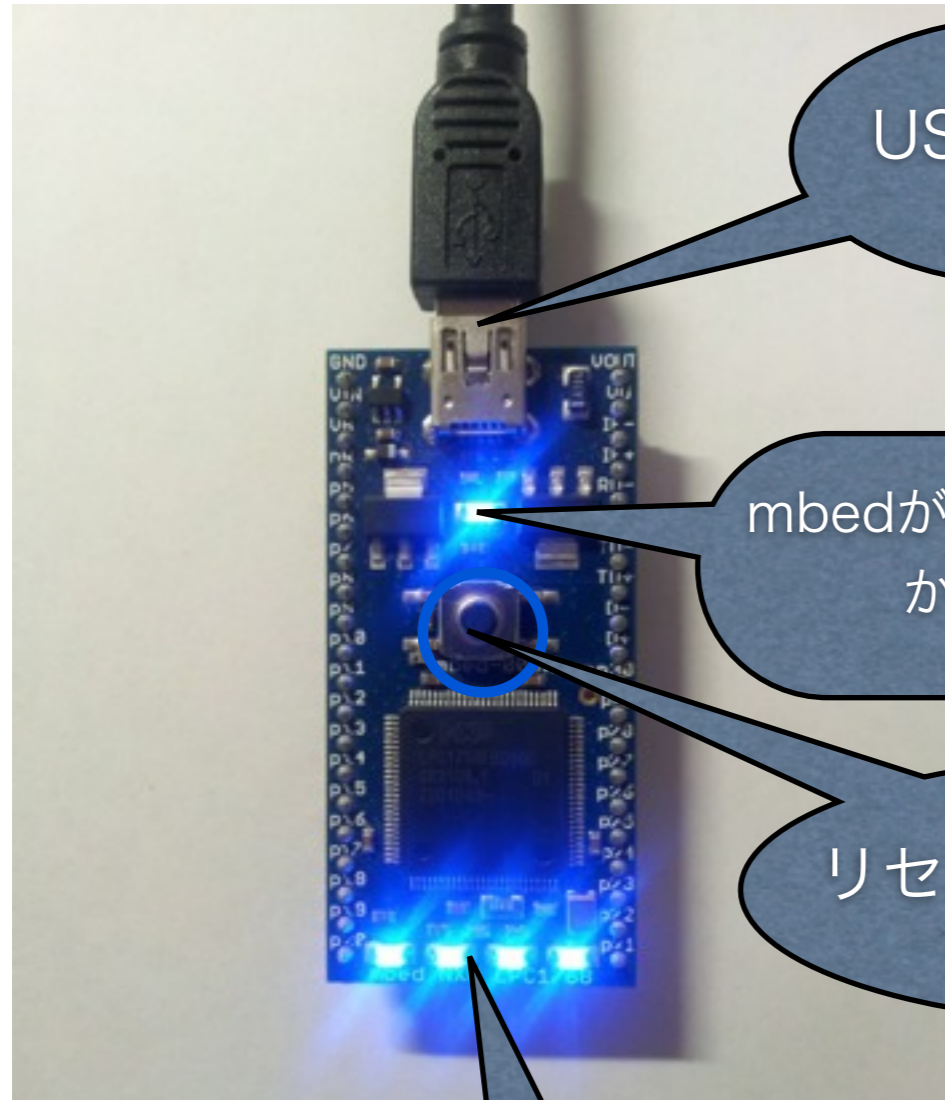
C++言語でプログラミングが出来るマイコン



出来ることって？

計算、ディスプレイ表示、センサーの値取得
...etc

mbedの各部説明



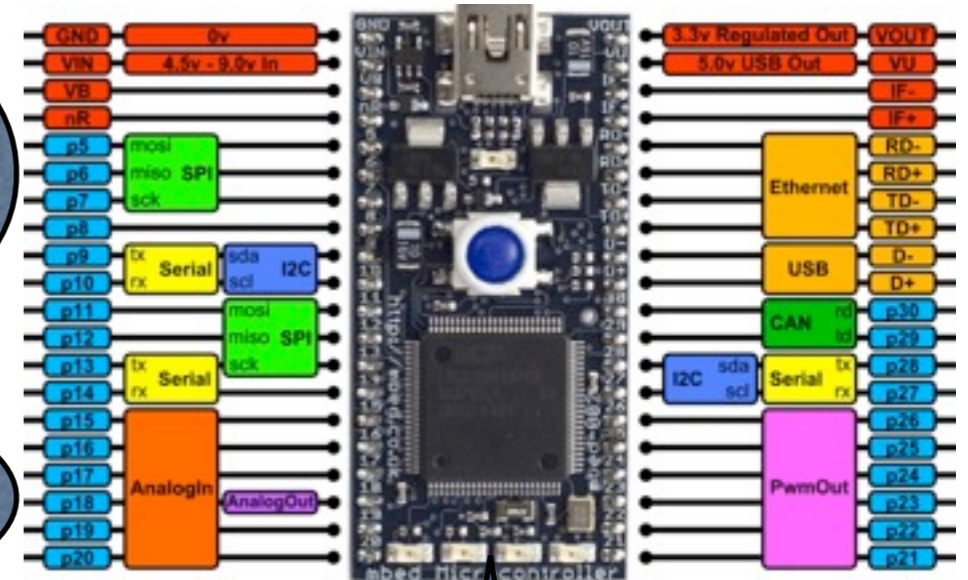
USBポート

mbedがきちんと動作しているかが分かります。

リセットボタンです。

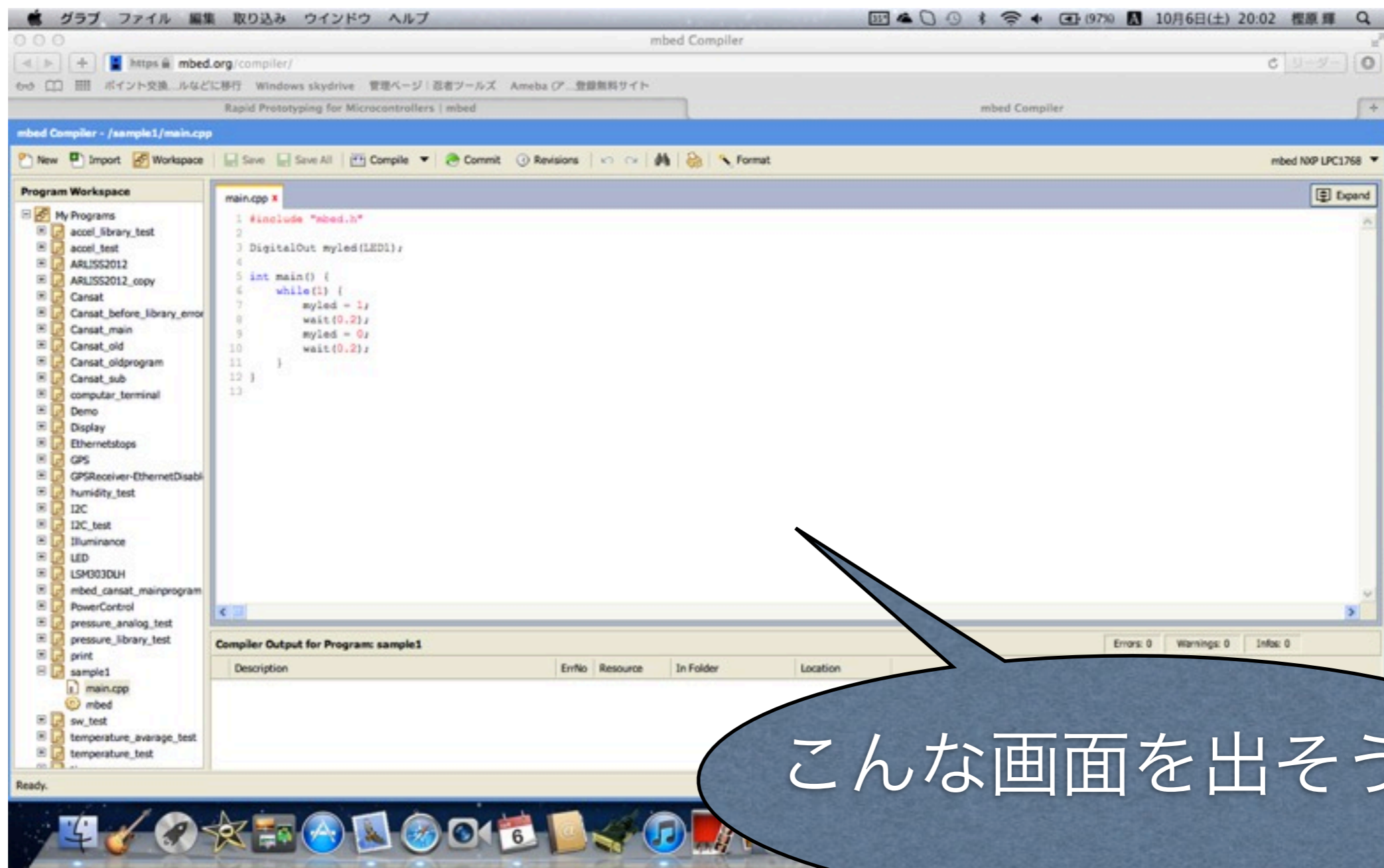
左からLED 1、LED 2、LED 3、LED 4
プログラムで光らせます。

どのピンがどの役目をするのかは、
このカードを見れば分かります。



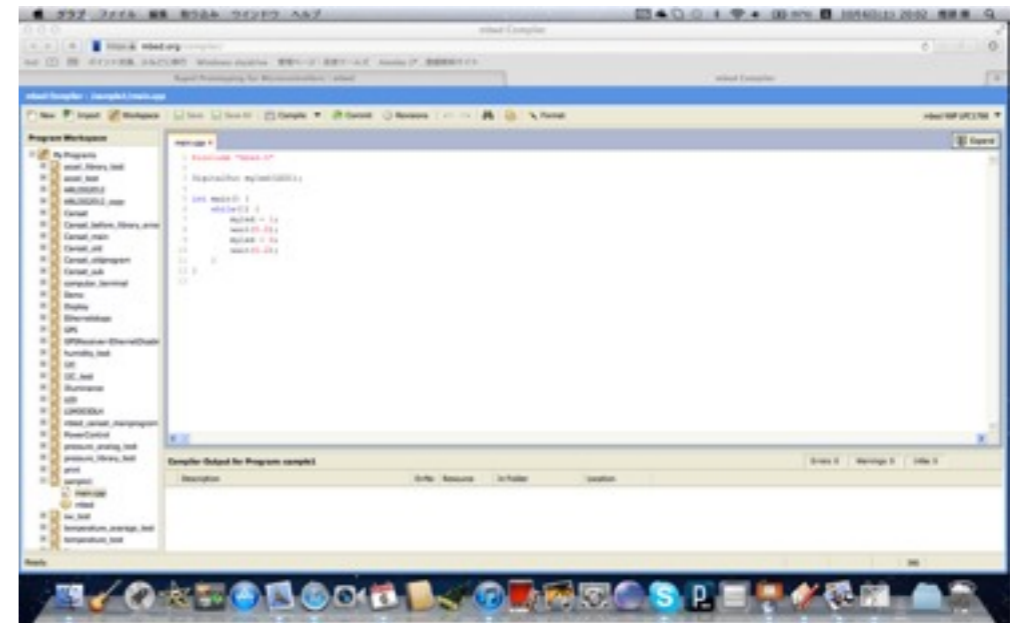
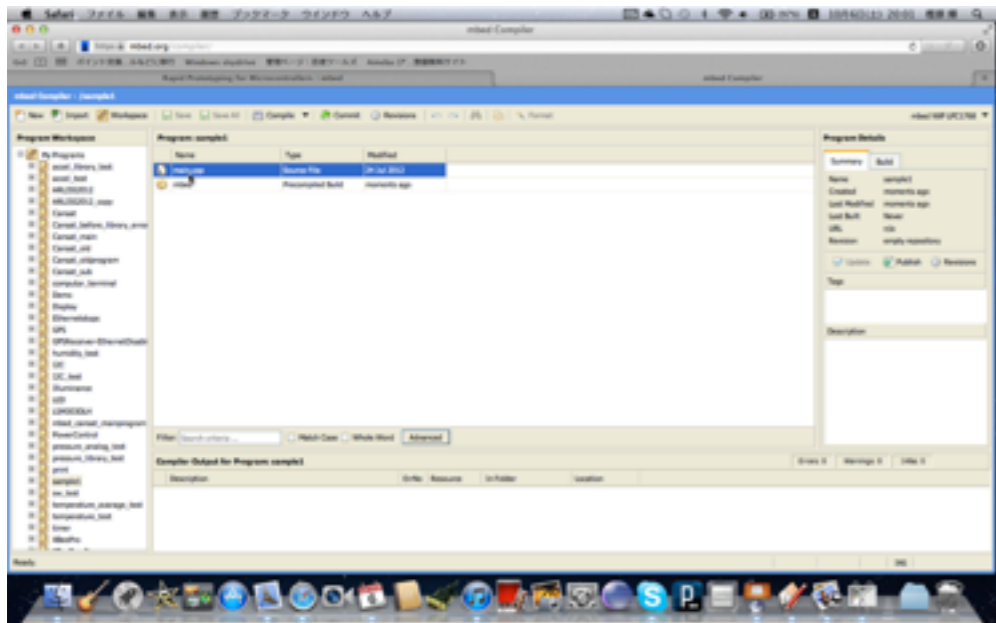
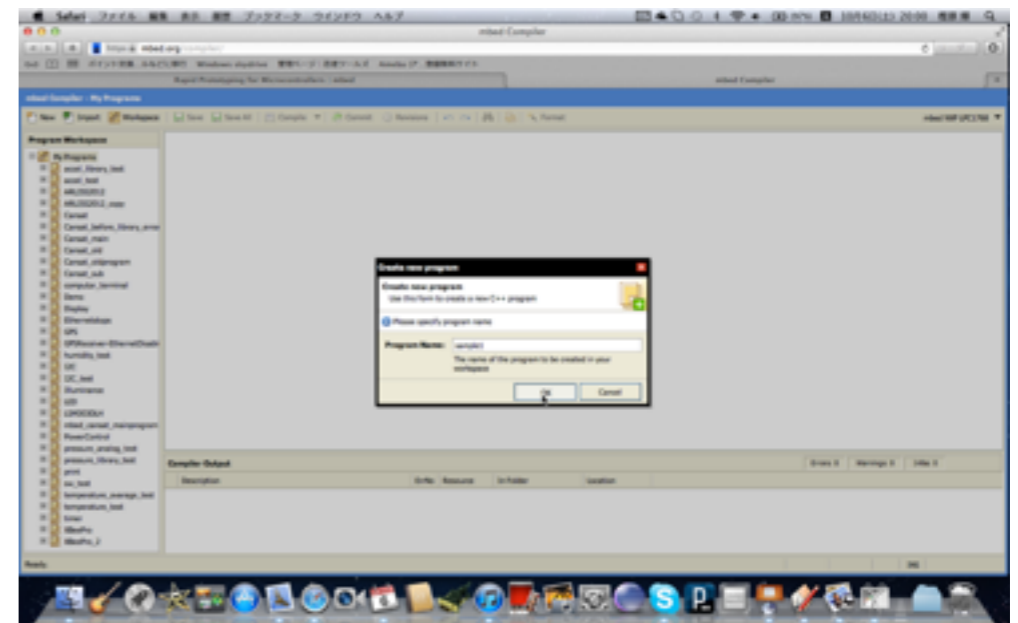
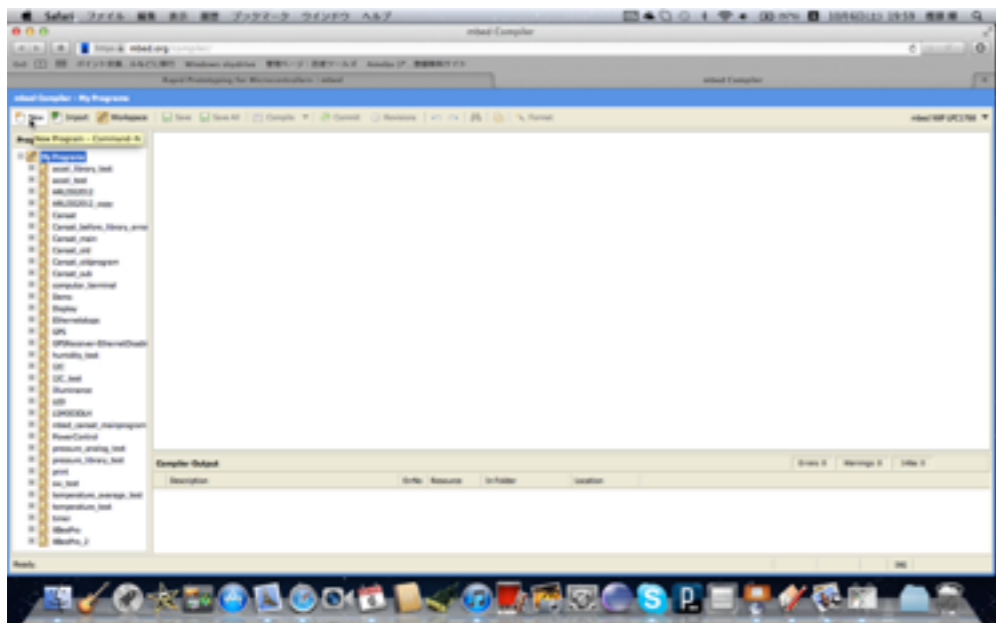
プログラミングをしよう！

mbedはプログラミングはインターネットで行う



こんな画面を出そう！





画面の説明

mbedの基本的なプログラムを使いますと宣言します。

ここでどのような仕組みで値を読み取るか、出力するのかを指定します。

上で定義したことを利用して、プログラムを作ります。コンピュータがどのような動きをするのかを書きます。

The screenshot shows the mbed Compiler interface with the following code in `main.cpp`:

```
1 #include "mbed.h"
2
3 DigitalOut myled(LED1)
4
5 int main() {
6     while(1) {
7         myled = 1;
8         wait(0.2);
9         myled = 0;
10        wait(0.2);
11    }
12 }
```

The interface includes a 'Program Workspace' on the left with a tree view of projects, a 'Compiler Output' window at the bottom, and a system tray at the very bottom.

LEDを光らせよう！

```
1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4
5 int main() {
6     while(1) {
7         myled = 1;
8         wait(0.2);
9         myled = 0;
10        wait(0.2);
11    }
12 }
```

← mbedの基本ライブラリを使うと宣言します。

← LED 1 にmyledという名前をつけます。

← int main()の{}の中に主 (main)となるプログラムを書きます。

← while()の{}の中には繰り返し動作させたいプログラムを書きます。もしもwhile(true)またはwhile(1)と書いたのであれば、while()の{}の中は繰り返し実行されます。また、while(false)もしくはwhile(0)と書けば、while()の{}の中は繰り返し実行されません。

← myledに0を代入します。もしも1を代入すればmyledに3. 3Vの電流が流れます。もしも0を代入すれば電流は流れません。

← waitは文字通り「待つ」という動作をします。()の中には、待つ時間を秒で記入します。この場合は0. 2秒待つプログラムです。

いろんな色のLEDを光らせよう!

☆LEDの基本

LEDは直流電源で光る電球です。アノード (+) とカソード (-) が決まっています、きちんと配線しないと光りません。

☆アノード、カソードの見分け方!



カソードは足が短く、
また、大きいLEDでは、電球の部分が平です。

アノードは足が長い

いろんな色のLEDを光らせよう!

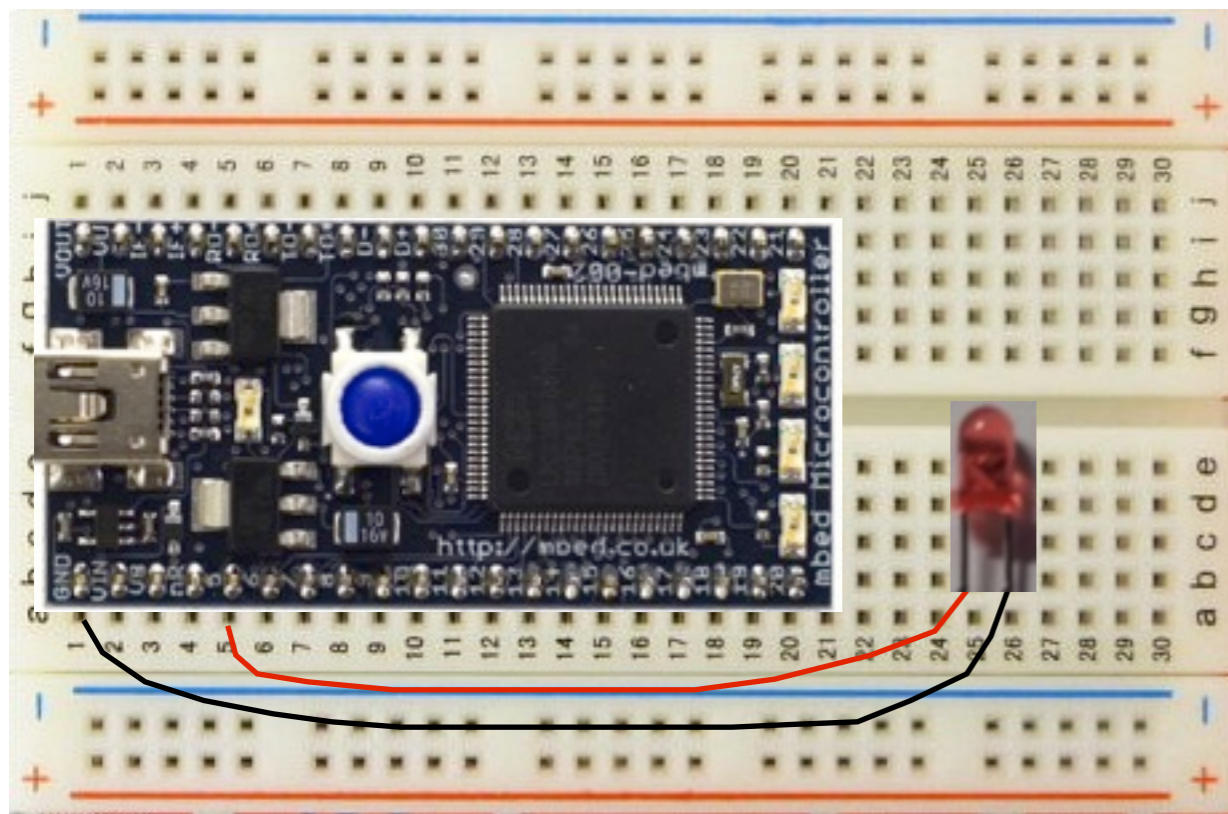
☆配線するときのマナー

1. ーに配線する線は黒、＋に配線する線は赤にする。
2. 電源は配線してから入れる。
3. 出来れば配線した後、誰かに確認してもらおう

いろんな色のLEDを光らせよう!

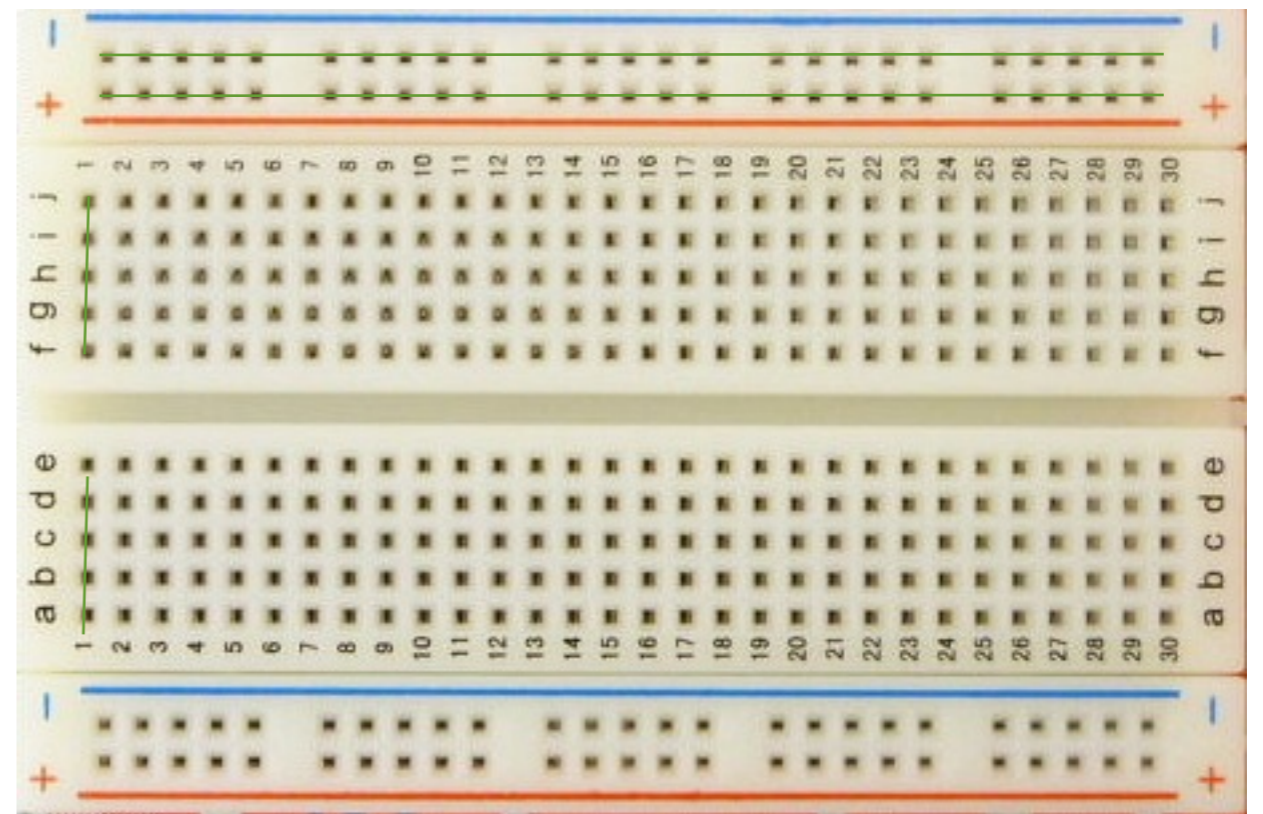
☆配線しよう!

以下の図のように配線します。



☆ブレッドボードについて

+とーのラインは横1列が、アルファベットと数字で記されている列は縦につながっている。



配線したLEDを光らせるには？

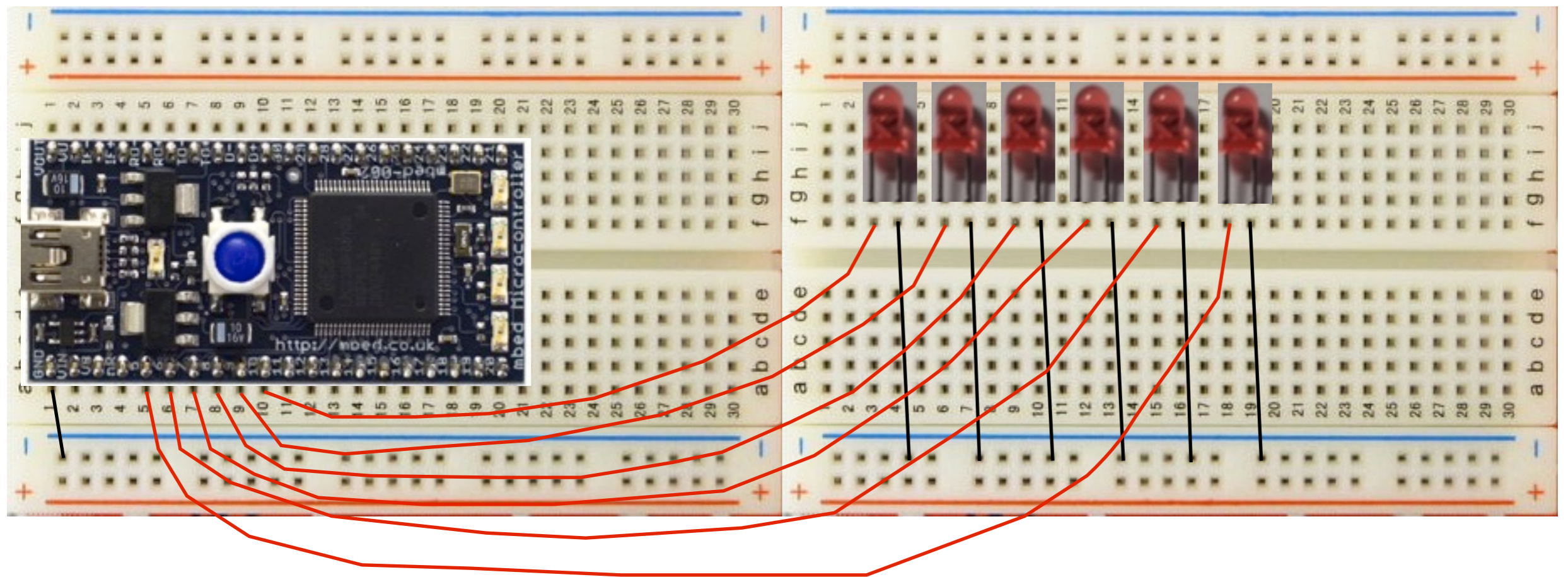
配線したLEDを光らせるには、次のようなプログラムを書きます。

```
1 #include "mbed.h"
2
3 DigitalOut myled(p5);
4
5 int main()
6 {
7     while(1) {
8         myled = 1;
9         wait(0.2);
10        myled = 0;
11        wait(0.2);
12    }
13 }
```

一つ前に作ったプログラムのLED 1 をp5に変えるだけ！
これで、5番のピン（p5）をmyledとするという意味

このプログラムを動かすと、配線したLEDが
0.2秒ごとに点滅するはずですよ

じゃあ、LEDを増やすと・・・



LEDを増やしたプログラム

```
1 #include "mbed.h"
2
3 DigitalOut myled1(p5);
4 DigitalOut myled2(p6);
5 DigitalOut myled3(p7);
6 DigitalOut myled4(p8);
7 DigitalOut myled5(p9);
8 DigitalOut myled6(p10);
9
10 int main() {
11     while(1) {
12         myled1 = 1;
13         wait(0.2);
14         myled2 = 1;
15         wait(0.2);
16         myled3 = 1;
17         wait(0.2);
18         myled4 = 1;
19         wait(0.2);
20         myled5 = 1;
21         wait(0.2);
22         myled6 = 1;
23         wait(0.2);
24         myled6 = 0;
25         wait(0.2);
26         myled5 = 0;
27         wait(0.2);
28         myled4 = 0;
29         wait(0.2);
30         myled3 = 0;
31         wait(0.2);
32         myled2 = 0;
33         wait(0.2);
34         myled1 = 0;
35         wait(0.2);
36     }
37 }
```

LEDに名前を付けます。

どのLEDを点灯させるかを書きます。

LEDを増やしたプログラム（簡単）

```
1 #include "mbed.h"
2
3 DigitalOut myleds[] = {p5, p6, p7, p8, p9, p10};
4
5 int main()
6 {
7     while(true) {
8         for(int i = 0; i < 6; i++) {
9             myleds[i] = 1;
10            wait(0.2);
11        }
12        for(int i = 5; i > -1; i--) {
13            myleds[i] = 0;
14            wait(0.2);
15        }
16    }
17 }
```

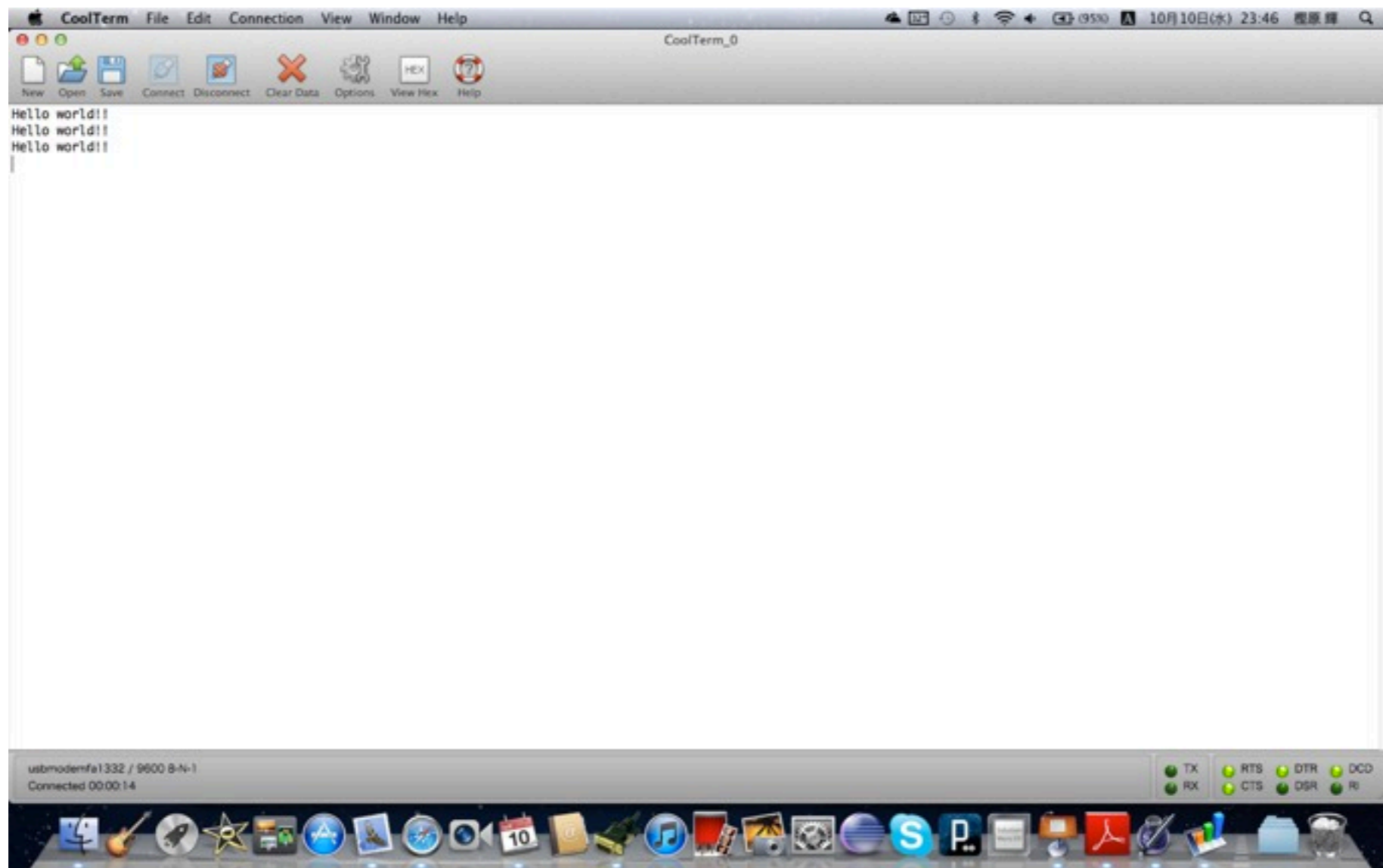
左から

myled[0],myled[1],myled[2],
myled[3]...と名前を付けています。

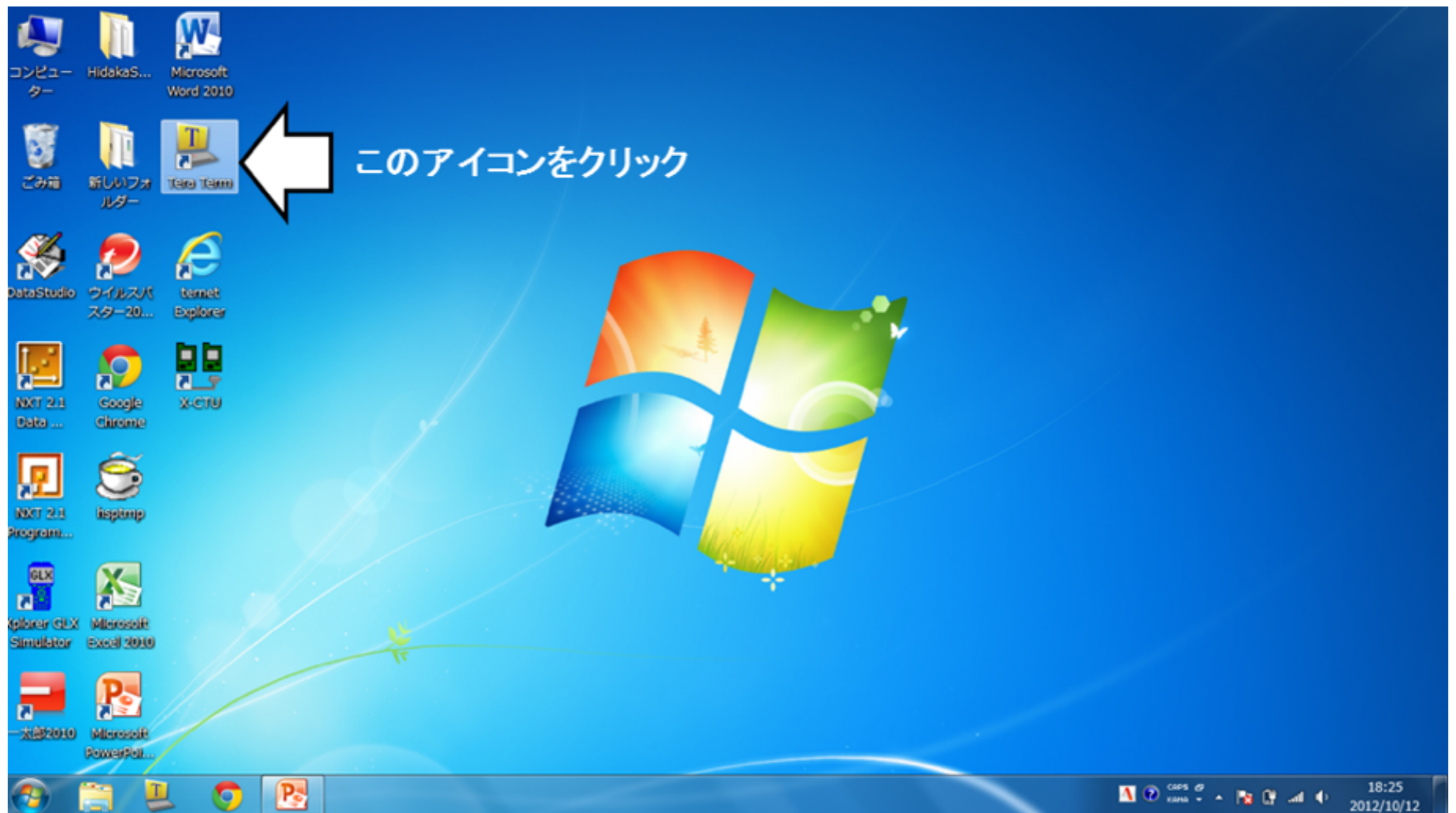
for()は繰り返し処理するプログラムに条件を付けることが出来ます。その条件を()の中に入ります。このプログラムだとint i = 0;の部分で、まずiという文字をおいて、それに0を代入しています。また、次のi < 6でiは6までの数字と範囲を決めています。i++ではiを1ずつ増やすという意味があります。つまりこのfor(int i = 0; i < 6; i++)には、{}の中のプログラムを6回繰り返したら、この中のプログラムを終了し、次のプログラムに移りなさいという意味があります。一言で言えば、{}の中のプログラムを6回繰り返し、終了して次の{}へ移行なさいというプログラムです。

パソコンに表示する

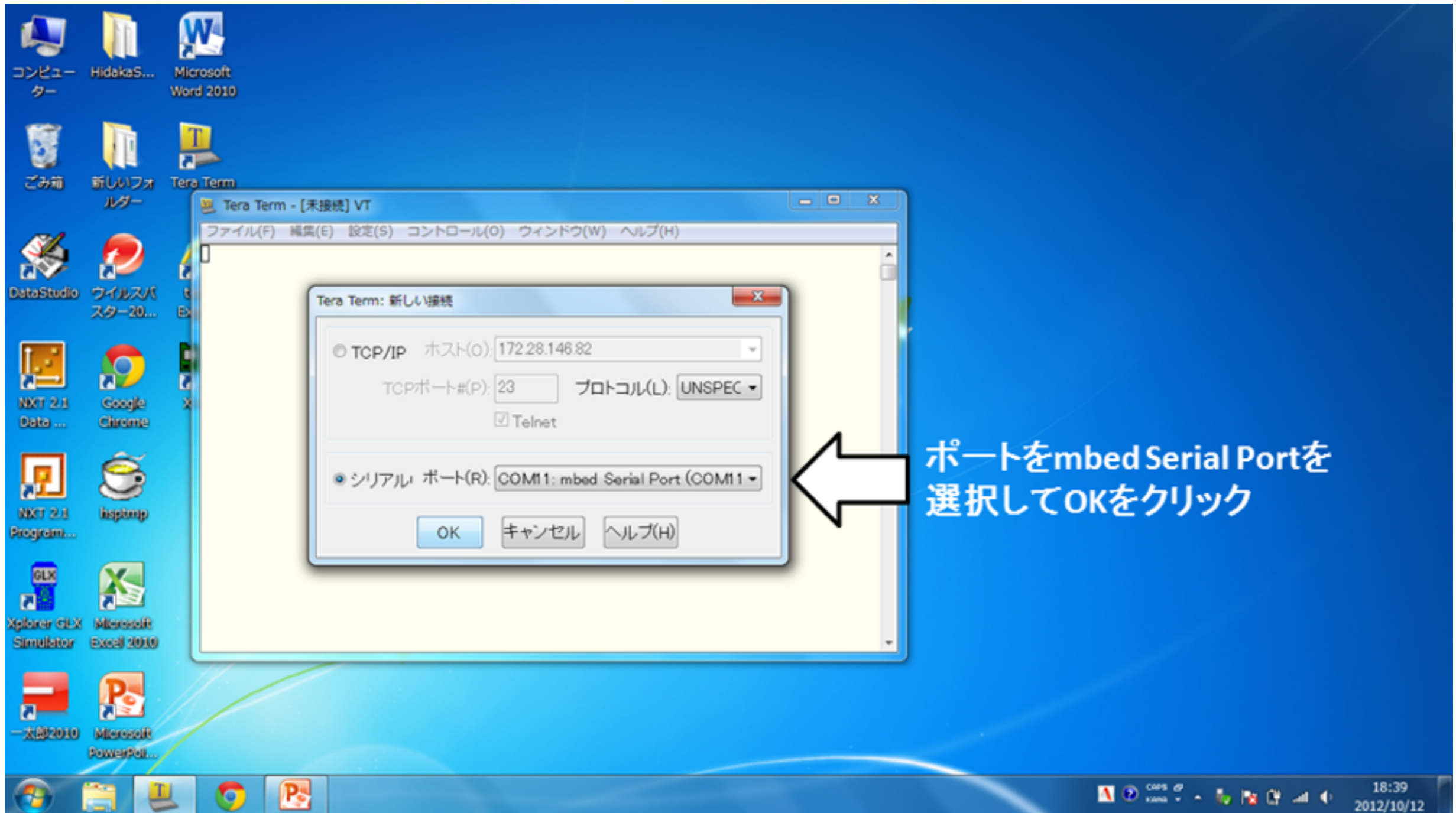
mbedのプログラムから、パソコンのディスプレイへ文字を表示させます。



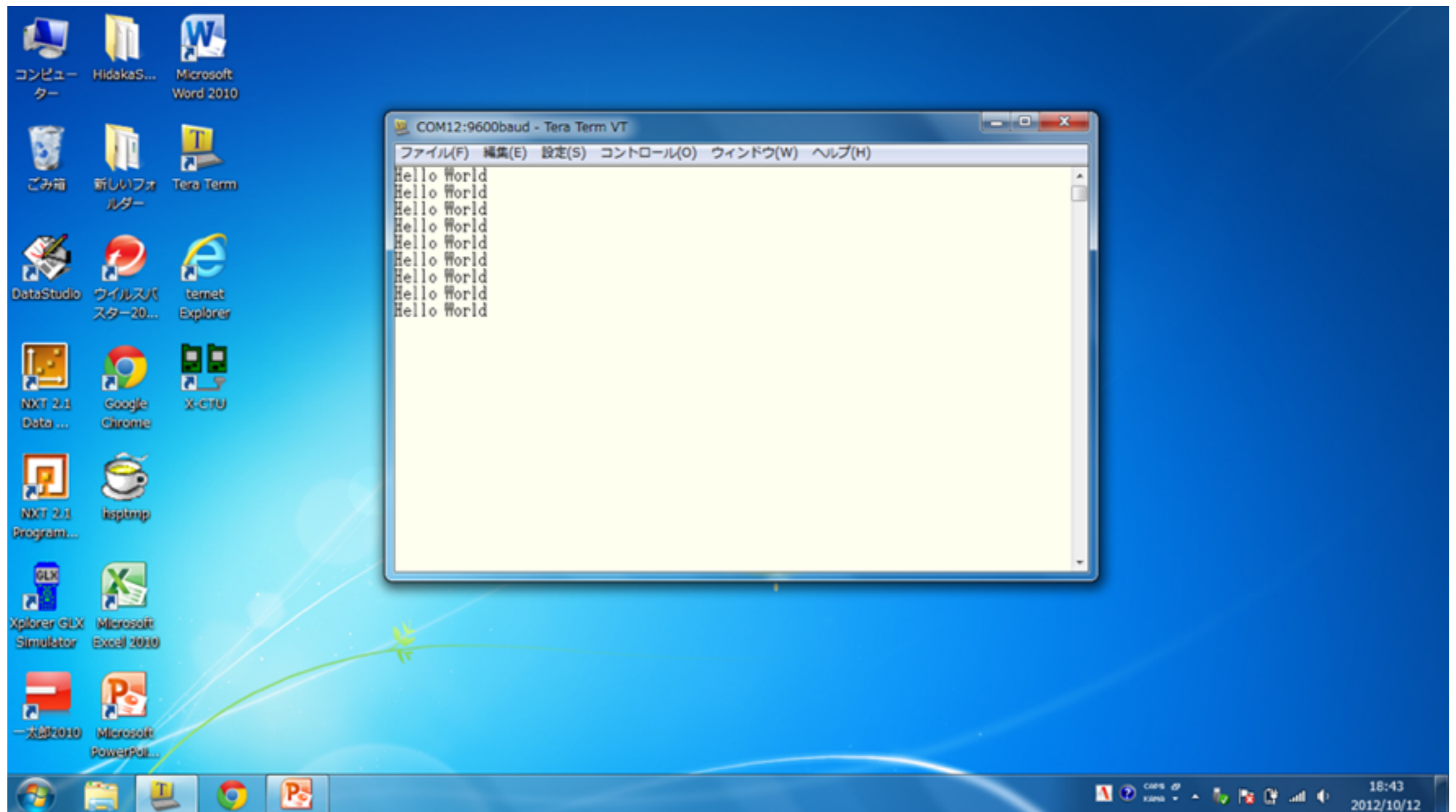
文字を表示させるには



文字を表示させるには



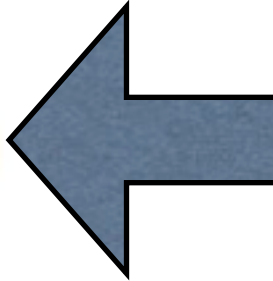
文字を表示させるには



プログラム

たったこれだけ！

```
1 #include "mbed.h"
2
3 int main()
4 {
5     printf("Hello world!!\n");
6 }
```

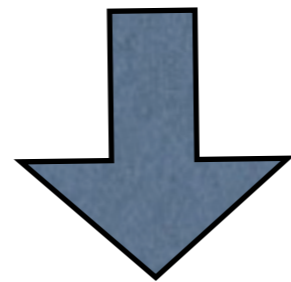


printfとは文字などを書き出す（出力する）時に使う命令です。()の中には” ”でくくった文字列を入れます。
\nは改行コードと言われるもので、改行をさせる命令みたいなものです。

照度を測ってみよう！

☆照度とは？

照度とは、光がある面を照らすとき、この面の単位面積が単位時間に受ける光束。単位はルクス (lx)



難しいので・・・

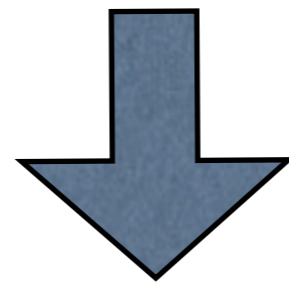
簡単に言うと、明るさ。単位のルクス (lx) は覚えておいてください！！

照度センサーを配線してみよう

☆はじめに、照度センサーについて

プログラミング、配線をする上で、センサーがどのような仕組みで動くのかを知ることが大切！（ある程度で良い）

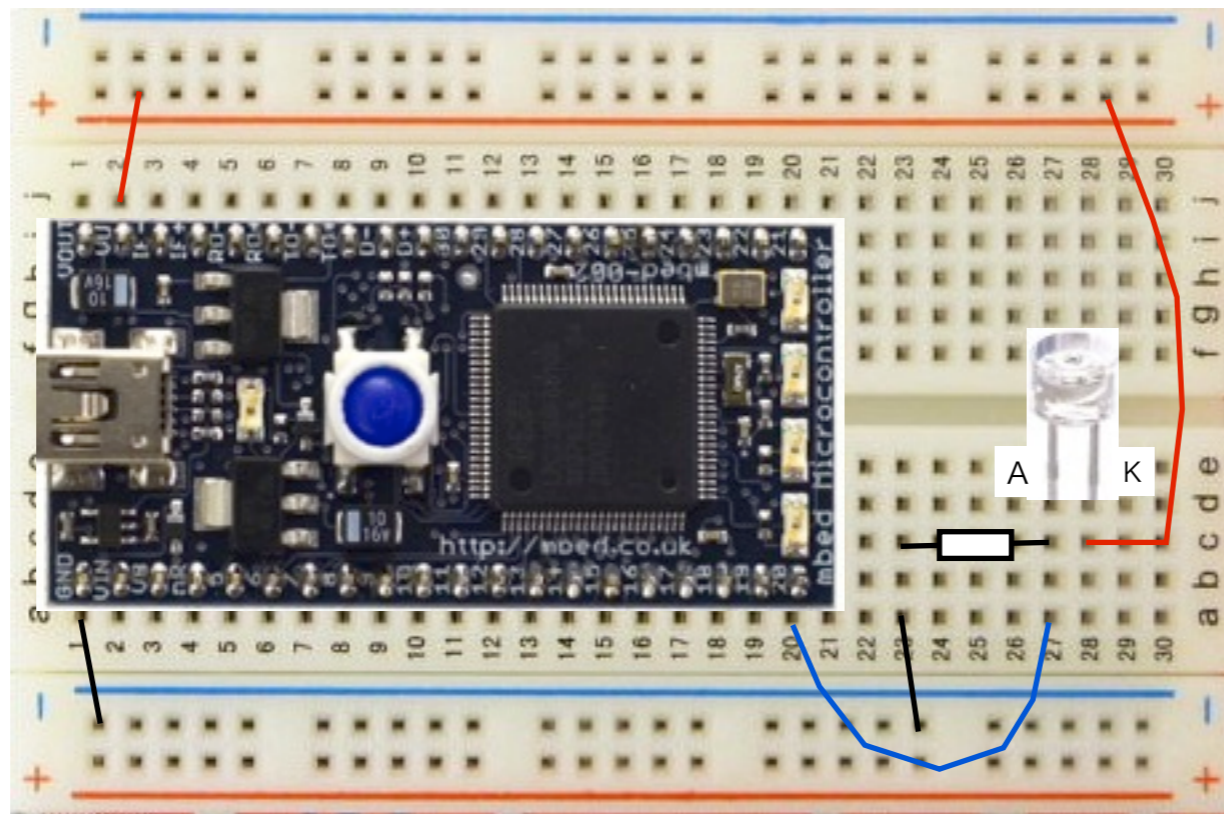
今回使う照度センサーはダイオードを使用しており、発行ダイオードの逆の動作をして数値を読み取っている。



つまり！

発光ダイオード（LED）と配線は基本的には同じ！

照度センサーを配線してみよう



プログラミングをしよう！

```
1 #include "mbed.h"
2
3 AnalogIn illu(p20);
4
5 int main() {
6     while (true) {
7         float lx = illu * 3.3 / 3 * 1000;
8         printf("%5.2f\r\n", lx);
9         wait(1.0);
10    }
11 }
```

← AnalogInとは、電圧をそのまま忠実に読み取る方式です。

float lx = でlxという浮動小数点の数（つまり小数点の数）の文字に右の式の値を代入しています。
printf("%5.2f\r\n, lx")の%5.2fのfは浮動小数点型の数を入れるということを示しており、%5.2というのは整数の桁数が最大5桁、小数点以下の桁数が最大2桁ということを示しています。また\r\nはwindowsでの改行コードです。%5.2fに代入されるのはlxの値です。

挑戦しよう！

今まで習ってきたことでmbedの入出力の基本が学べたはずです！

余裕のある人は次の課題にチャレンジしてみてください！

(プログラミングの仕方には何通りかあります。答えが一つとは限りません！)

課題 1

LEDを使ったプログラムです。配線図、プログラムの形は問いません。自由にLEDを4つ以上使ったイルミネーションなどを制作してください！

課題 2

照度センサーの値と比例して、LEDの点灯する
個数が変わるようなプログラムを作ってください
！（これは、プレゼンで少し説明します）

Challenge2_easy
Challenge2_Simple

課題 3

隣の人と話し合って2人で一つ大作を作ってみましょう！今日中に出来なくてもかまいません。どんなものを作りたいかを書き出してください！

(LEDだけでなく、モータや各種センサー、無線装置は自由に使えるとします！)

さいごに

もしも、高度なプログラミングに興味を持って、もっとやりたいと思った人は、平日の放課後物理講義室に来れば続きをすることが出来ます！質問も受け付けているので、気軽にきてください！

もっと高度なプログラムを自分で学びたいときは、次のページに書かれてあるリンク先で質問をすれば、senio networks.incの方が、親切に教えてくれます！

リンク

mbed website : <http://mbed.org>

senio networks.inc 掲示板 : <http://www.senionetworks.jp/bbs/index.php>

サンプルプログラム : <http://mbed.org/users/AkiraK/code/>

