

最初のスケッチ

(Arduino 公式サイトの First Sketch ページ <http://arduino.cc/en/Tutorial/Sketch> を日本語に訳したものです。)

Arduino 入門 (Getting started guide (Windows, MacOS X, Linux)) では、LED を点滅させるスケッチをアップロードしました。ここでは、スケッチのそれぞれの部分がどのような働きをしているかを学びます。

スケッチ

Arduino で使うプログラムを「スケッチ」と呼びます。Arduino ボードにアップロードされ、Arduino を動かす一連の記号(コード)の集まりです。

コメント

Blink スケッチの最初の集合はコメントです:

```
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

/*
 * 基本的な Arduino の使用例。
 * LED を1秒間点けて、その後、1秒間消して、それを繰り返す。
 * 13番ピンを使用。Arduino ボードにも依存するが、
 * 備え付けの LED や、LED を一つだけ用意するだけで済むレジスタが
 * 備わっている。
 * http://www.arduino.cc/en/Tutorial/Blink
 */
```

/* と */ の間にある全ては、Arduino がスケッチを実行する時には無視されます(それぞれの行の最初にある*はコメントをカワイく見せる為にあるだけで、必須ではありません。)。これは人間がコードを読む為にあります: プログラムが何をするのか、どのように動くのか、どうしてこのように書かれているのかなどを説明します。スケッチにコメントを書くことと、コードを変更した時にコメントを最新にすることは良い練習になります。コメントは他の人達がコードを理解するのと修正変更をすることを助けます。

コメントのスタイルには、この他に短い一行コメントもあります。

```
int ledPin = 13;           // LED connected to digital pin 13
```

"LED connected to digital pin 13"がコメントです。

変数

変数とは一つの情報を格納する場所です。変数は名前と型と値を持っています。例えば、前述の Blink スケッチの行は `ledPin` という名前で、`int` 型で、初期値が `13` の変数を宣言しています。これは Arduino のどのピンに LED が接続されているかを示すのに使われています。コードに `ledPin` という名前が現れる度に、その値が読み出されます。このケースでは、プログラムを書く人がわざわざ `ledPin` 変数を作らずに、代わりにピン番号を特定する必要がある所すべてに単純に `13` を書く方法も選択できたでしょう。変数を使う利点は LED を別のピンに変更するのが簡単になることです: 変数の初期値を指定している一行を編集するだけで済みます。

しかし、スケッチが実行している間にしばしば変数の値が変わることがあります。例えば、入力から読み出した値を変数に格納することもできます。詳細は「変数チュートリアル (Variables tutorial) <http://arduino.cc/en/Tutorial/Variables>」をご覧ください。

関数

関数 (他ではプロシージャやサブルーチンとして知られている) は、スケッチの他の場所から使うことができる、名前の付けられたコードの断片です。例えば、これは Blink の例の `setup()` 関数の定義です。

```
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
```

最初の行は、"`setup`" という名前のような、関数に関する情報を規定しています。名前の前と後ろの文字列は、戻り値の型とパラメータを指定しています: これらは後で説明します。{ と } の間のコードは関数の本体 (body) と呼ばれています: 関数が何をするかを表しています。

既に定義済みの関数 (貴方のスケッチの中にある関数と、Arduino 言語の一部となっている関数) を呼び出すことができます。例えば、`pinMode(ledPin, OUTPUT);` の行では、2つのパラメータ: `ledPin` と `OUTPUT` を渡して、`pinMode()` 関数を呼び出しています。これらのパラメータは、`pinMode()` 関数でどのピンとモードをセットするかを決めるのに使われます。

`pinMode()`, `digitalWrite()`, and `delay()`

関数 `pinMode()` は、ピンを入力か出力に設定します。使うには、設定するピンの番号と `INPUT` または `OUTPUT` の定数を関数に渡します。入力に設定されたときは、ピンはプッシュボタンのような

センサーの状態を検出できます;これは後のチュートリアルで話題にします? (訳注1) 出力では、LED のような作動装置を制御できます。

訳注1:現在のホームページでは、該当箇所を参照できません。まだ書かれていないのかと思います。

関数 `digitalWrite()` は、ピンに値を出力します。例えば、行:

```
digitalWrite(ledPin, HIGH);
```

は、`ledPin`(13番ピン)を `HIGH`、または 5 ボルトに設定します。ピンを `LOW` にするとグランド、または 0 ボルトに設定します。

関数 `delay()` は、次の行へ行く前に、`Arduino` をミリ秒単位の数値で指定された時間だけ待たせます。1秒は 1000 ミリ秒なので、行:

```
delay(1000);
```

は、1秒間の遅れを発生させます。

setup() and loop()

すべての `Arduino` のスケッチには、`setup()` と `loop()` の2つの特別な関数があります。関数 `setup()` は、スケッチが開始するときに一度だけ呼び出されます。ピンモードの設定やライブラリの初期化のような初期化の仕事には丁度良い場所です。関数 `loop()` は、何度も呼び出され、殆どのスケッチの心臓部となります。例え貴方が必要無くても、スケッチには両方の関数を含める必要があります。

練習

1. LED を 100 ミリ秒点燈して、1000 ミリ秒消すようにコードを変更しなさい。
2. LED をスケッチの開始時に点燈して、そのまま点燈し続けるようにコードを変更しなさい。

参照

- * `setup()` <http://arduino.cc/en/Reference/Setup>
 - * `loop()` <http://arduino.cc/en/Reference/Loop>
 - * `pinMode()` <http://arduino.cc/en/Reference/PinMode>
 - * `digitalWrite()` <http://arduino.cc/en/Reference/DigitalWrite>
 - * `delay()` <http://arduino.cc/en/Reference/Delay>
-

ご注意:

*この翻訳は `arduino` 公式ページを独自に翻訳したものです。`Arduino` プロジェクトとは関係ありません。

*このページのライセンスは、`arduino` 公式ページの `Creative Commons Attribution-ShareAlike 3.0 License`. <http://creativecommons.org/licenses/by-sa/3.0/> に基づいています。