第2章 ライブラリマネージャに登録！世界と共有！

Arduino IDEで使える自作ライブラリの作り方

山崎 雅夫 Masawo Yamazaki

Arduinoで開発を進めていくと，複数のスケッチ(Arduinoでのプログラムのこと)で同じような処理を繰り返し記述することになり，コードが複雑になることがあります．

このような場合に役立つのがライブラリの機能です．作成したライブラリは，他のプロジェクトでも簡単に利用できるようになります．

ここでは，Arduino IDE向けの自作ライブラリの作成方法と，ライブラリマネージャへの登録方法を解説します．

ライブラリの基本構成

Arduinoライブラリは，**図1**の構成で作成します．この例ではMyLibraryというライブラリ名にしていますが，これは必要に応じて名前を設定できます．

それぞれのファイルは，次の役割をもっています．

- MyLibrary.h：ライブラリのヘッダ・ファイル．ライブラリのインターフェース(関数やクラスの宣言)を定義する
- MyLibrary.cpp：ライブラリのソース・ファイル．ヘッダ・ファイルで宣言した関数の具体的な実装を記述する
- library.properties：ライブラリのメタ情報．ライブラリのバージョン，作者，説明などのメタ・データを定義する
- keywords.txt：ライブラリの関数名情報(オプション)．Arduino IDEのコード・エディタでの関数名の色付けのために使用する

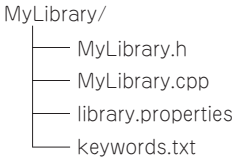

図1　Arduinoライブラリの構成

最小のライブラリを作ってみる

Arduinoのライブラリは，C++のクラスをベースにして作成するのが一般的です．例として，LEDの点滅を制御するシンプルなライブラリSimpleBlinkを作ってみます．

● ステップ1：ライブラリ・フォルダの作成

ライブラリを格納するフォルダSimpleBlinkを任意の場所に作成します．

● ステップ2：ヘッダ・ファイル(.h)の作成

ライブラリのインターフェースを定義します．SimpleBlinkフォルダ内にSimpleBlink.hというファイル名で作成します(**リスト1**)．

リスト1　ヘッダ・ファイル…ライブラリのインターフェースを定義する

SimpleBlink.h

```
// SimpleBlink.h
#ifndef SimpleBlink_h
                          // ヘッダファイルの多重インクルード防止
#define SimpleBlink_h

// Arduinoの基本関数を使用するためのヘッダ
#include "Arduino.h"

// SimpleBlinkというクラスを定義
class SimpleBlink {
 // クラスの外部からアクセス可能なメンバー(関数や変数)を定義
 public:
  SimpleBlink(int pin);
                    // クラスオブジェクト生成のコンストラクタ
  void blink(int delayMs); // LEDを点滅させる関数
  void on();             // LEDをONにする関数
  void off();            // LEDをOFFにする関数

 //クラスの内部からのみアクセス可能なメンバーを定義
 private:
  int _ledPin; // LEDが接続されているピン番号を
                                          保持する変数
};

#endif // ヘッダファイルの多重インクルード防止
```