

MP3 (MPEG Audio Layer-3) とは

MP3 とは、音声データを圧縮する方式及びそれから作られる音声ファイルフォーマットの一つ。ファイルの拡張子は.mp3 となる。Raspberry Pi は HDMI や AudioJack から音声出力を行う事ができる。今課題では **JACo MP3Player ライブラリ** を使用して音楽を再生する。

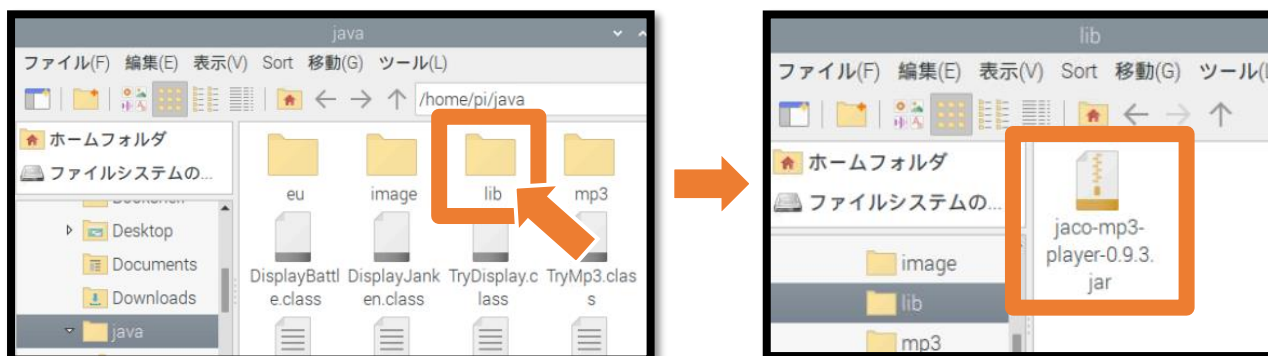


ライブラリ (library) とは

直訳すれば図書館という意味だが、IT 業界・プログラミング用語でいうライブラリとは他のプログラムからも利用できるように **プログラムを纏めて部品化したもの**を指す。Java でライブラリを使用したい場合、そのライブラリの JAR ファイルを配置して import を行う。

確認と準備

今回使用する JACo MP3Player ライブラリの JAR ファイル (jaco-mp3-player-0.9.3.jar) は 1 回目の環境構築で /home/pi/java 内の **[lib]** フォルダ内に配置しているので存在を確認しておく。



もし **削除してしまっていた場合は** LAN ケーブルを挿入し、以下のサイトからダウンロードすること。
[JACo MP3 Player \(java mp3 player \) \(sourceforge.net\)](http://sourceforge.net/projects/jacomp3player/)



練習（TryMp3.java）

以下のサンプルプログラムを記述して実行し、SW1 を押したときに MP3 ファイル（8bit18.mp3）が再生され、SW2 を押したときに一時停止されるか確認しなさい。

```
/*
 * TryMp3. java
 * Date   : 2022/01/01
 * Author : IE1A 99 K.Murakami
 */
```

```
// GPIO ピンを利用するために必要なクラスを読み込む
import com.pi4j.wiringpi.Gpio;
import com.pi4j.wiringpi.GpioUtil;
// MP3 ファイルを利用するために必要なクラスを読み込む
import jaco.mp3.player.MP3Player;
import java.io.File;
```

```
public class TryMp3 {
    // Thread.sleep メソッドで発生する割り込み例外を throws する
    public static void main (String[] args) throws InterruptedException {
```

```
        System.out.println("プログラム開始");
```

```
        // デジタル出力信号を定数化
```

```
        final int HIGH = 1;
```

```
        final int LOW = 0;
```

```
        // デジタル入力信号を定数化
```

```
        final int ON = 0;
```

```
        final int OFF = 1;
```

```
        // LED のピン番号を宣言
```

```
        final int LED1 = 15;
```

```
        final int LED2 = 16;
```

```
        // SW のピン番号を宣言
```

```
        final int SW1 = 7;
```

```
        final int SW2 = 0;
```



ライブラリを使用する為に必要な import 文

```
// GPIO を初期化
Gpio.wiringPiSetup();
System.out.println("GPIO 初期化完了");

// 各 LED を出力に設定
Gpio.pinMode(LED1, Gpio.OUTPUT);
Gpio.pinMode(LED2, Gpio.OUTPUT);
// 各 SW を入力に設定
Gpio.pinMode(SW1, Gpio.INPUT);
Gpio.pinMode(SW2, Gpio.INPUT);
System.out.println("GPIO 入出力設定完了");

// Mp3Player オブジェクトを作成
MP3Player player = new MP3Player();
// player に曲を追加 (パス/ファイル名. 拡張子)
player.addToPlaylist(new File("./mp3/8bit18.mp3"));

// 各 SW の状態を保持する変数
int sw1Data, sw2Data;
// 前回の各 SW の状態を保持する変数 (初期値は OFF)
int sw1DataOld = OFF;
int sw2DataOld = OFF;

// プログラムを終了させない為に無限ループする
while(true) {
    // 現在の各 SW の状態を読み取る
    sw1Data = Gpio.digitalRead(SW1);
    sw2Data = Gpio.digitalRead(SW2);

    // 現在の SW の状態が、前回の SW の状態と違う (変化があった) 場合
    if(sw1Data != sw1DataOld) {
        if(sw1Data == ON) {
            Gpio.digitalWrite(LED1, HIGH); //LED 点灯
            player.play(); //再生
        } else {
            Gpio.digitalWrite(LED1, LOW); //LED 消灯
        }
    }
}
```

addToPlaylist メソッドを使用し、
player オブジェクトに曲を追加する

オブジェクト.メソッドで
様々な操作が可能

```
// 現在の SW の状態が、前回の SW の状態と違う（変化があった）場合
if(sw2Data != sw2DataOld) {
    if(sw2Data == ON) {
        Gpio.digitalWrite(LED2, HIGH); //LED 点灯
        player.pause(); //一時停止
    }else{
        Gpio.digitalWrite(LED2, LOW); //LED 消灯
    }
}

// 次回の為に現在の各 SW の状態を保存する
sw1DataOld = sw1Data;
sw2DataOld = sw2Data;

// 100 ミリ秒待機する
Thread.sleep(100);
}
}
}
```

リファレンスとは

直訳すると【参照する】という意味で、今回のようにライブラリやクラスを使用する場合に参考となるサンプルプログラムやメソッドの仕様などを一覧で記載している公式的な Web ページを指す。プログラマーはリファレンスを見ながらコードを書きあげていくので、見慣れておく必要がある。

【JACo MP3Player リファレンス】

[MP3Player \(sourceforge.net\)](http://sourceforge.net)

Method Summary	
void	addMP3PlayerListener (MP3PlayerListener listener) Adds a MP3PlayerListener to the player.
void	addToPlayList (java.io.File mp3) Appends the specified mp3 (as File object) to the end of the play list.
void	addToURLList (java.net.URL mp3) Appends the specified mp3 (as URL object) to the end of the play list.
java.util.List<java.io.File>	getPlayList () Returns the current play list.
boolean	isPaused () Determines whether this player is paused.

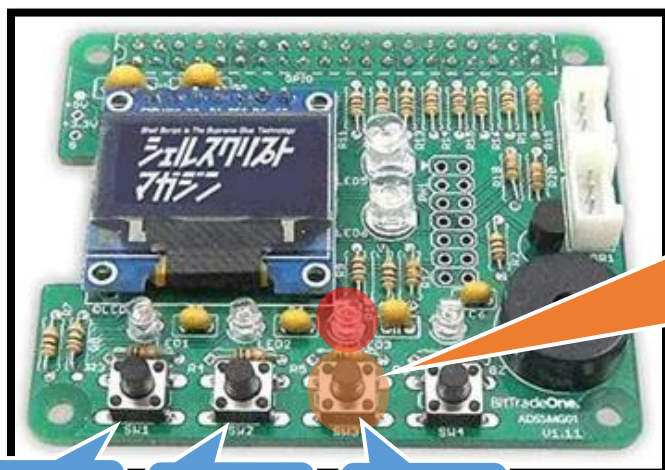
メソッド名をクリックすると、そのメソッドに必要な引数・処理の内容・戻り値などの詳細が記載されている箇所に飛んで確認できる

課題 1 (Mp3Next.java)

前述の練習を**別名保存**し、以下の条件で複数の曲を再生できるプログラムを追記しなさい。

必要な処理 (メソッド) をリファレンスから探して活用すること。

- ・ 曲数は mp3 フォルダ内の 5 曲 (8bit18・ethnic32・healing08・neorock73・orchestra24) とする。
- ・ SW1 → 練習のまま。押している間は LED1 を点灯し、曲を再生する。離すと消灯する。
- ・ SW2 → 練習のまま。押している間は LED2 を点灯し、曲を一時停止する。離すと消灯する。
- ・ SW3 → 押している間は LED3 を点灯し、**次の曲を再生する**。離すと消灯する。



曲の再生中に押すと次の曲が再生され、一時停止中に押すと今の曲が再生されるプレイリストの最後の曲 (5 曲目) を再生中に押した場合は何も起こらない

再生

一時停止

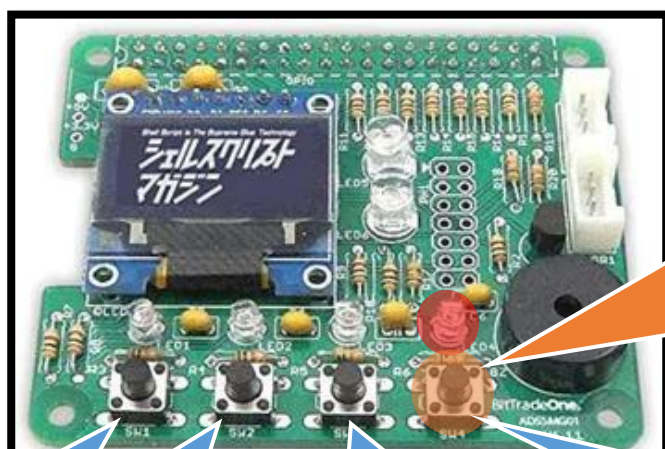
次の曲へ

課題 2 (Mp3Repeat.java)

前述の課題を**別名保存**し、以下の条件で複数の曲を繰り返し再生できるプログラムを追記しなさい。

- ・ SW4 → 押している間は LED4 を点灯し、**リピートの ON / OFF を切り替える**。離すと消灯する。

※現在リピート機能が ON (true) なのか、OFF (false) なのか状態を判定する処理が必要となる。



リピート機能は実行時にデフォルトで OFF になっている
リピート機能が ON の状態で次の曲へ進むと、最後の曲 (5 曲目) から最初の曲 (1 曲目) に移る

再生

一時停止

次の曲へ

リピート ON / OFF