

総合課題（MusicPlayer.java）

これまでに学んだ知識と経験を活かし、以下の仕様を満たすミュージックプレイヤーを作成しなさい。

◆使用電子部品

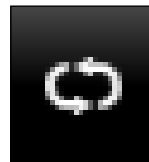
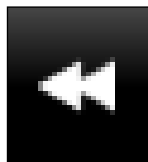
- ・ LED（1・2・3・4）
- ・ SWITCH（1・2・3・4）
- ・ 圧電ブザー
- ・ OLED（有機 EL ディスプレイ）

◆仕様

【基本情報】

- ・ ミュージックプレイヤーの曲数は5曲とし、[mp3]フォルダ内のファイルをプレイリストに追加する。
- ・ プレイヤー機能は以下の4つとし、各機能が視覚的に操作できるよう OLED に画像を表示する。

- 前の曲を再生
- 再生/一時停止の切り替え
- 次の曲を再生
- リピート ON/OFF の切り替え



【LED】

- ・ SW1 が押された場合は LED1 を点灯し、SW4 が離された場合は LED4 を消灯するといったように、各スイッチの ON/OFF と連動させて、押されているかどうか視覚的に確認できるようにする。

【SWITCH】

プレイヤー機能の4つと連動させる。

- ・ SW1 → 前の曲を再生する。リピートが ON の場合に1曲目を再生していたら5曲目を再生する。
- ・ SW2 → 曲が再生中の場合は一時停止し、一時停止中は再生に切り替える。
- ・ SW3 → 次の曲を再生する。リピートが ON の場合に5曲目を再生していたら1曲目を再生する。
- ・ SW4 → リピート機能が ON の場合は OFF に、OFF の場合は ON に切り替える。

【圧電ブザー】

- ・ 各スイッチが押された場合に操作音（周波数：1000Hz）を0.1秒間鳴らす。

【OLED】

一行目に課題名を表示し、
二行目に作成者（名前）を
表示する



常時表示

1

常時表示

再生中は一時停止画像を、
一時停止中は再生画像を
表示する

リピート機能が ON の
時のみ表示する

◆参考手順

【定義&準備処理】

- ① GPIO・SPI 通信&OLED・MP3Player ライブラリを使用する為に必要なインポート文を記述する。
- ② クラス、及び main メソッドを定義し main メソッド内に以降⑩までの処理を記述する。

```
public class MusicPlayer {  
    // 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, IOException{  
        System.out.println("プログラム開始");  
    }  
}
```

- ③ 以下の定数を定義する。
- ・ デジタル入出力用信号 (HIGH・LOW・ON・OFF)
 - ・ 使用ピン番号 (LED1~4・SW1~4・BUZZER)
- ④ GPIO 初期化処理を記述する。
- ⑤ LED1~4・SW1~4・BUZZER の入出力設定を記述する。
- ⑥ ディスプレイの初期化設定を記述する。
- ⑦ ディスプレイの書式設定、及びタイトル・作成者の表示設定を行う。

```
// ディスプレイの書式設定  
disp.getGraphics().setFont(new Font("Serif", Font.PLAIN, 12));  
disp.getGraphics().setColor(Color.WHITE);  
// 文字を設定 (文字列, X 座標, Y 座標)  
disp.getGraphics().drawString("タイトル", 0, 10);  
disp.getGraphics().drawString("作成者 ", 0, 30);
```

- ⑧ 使用する 5 種類の画像のオブジェクトを作成する (以下例)
- ```
Image playImg = ImageIO.read(MusicPlayer.class.getResourceAsStream("./image/play.png"));
```
- ⑨ 画像の表示を設定し、文字と画像の初期描画を行う。

```
// 画像を設定
disp.getGraphics().drawImage(〜 省略 〜, 0, 48, null);
disp.getGraphics().drawImage(playImg, 32, 48, null);
disp.getGraphics().drawImage(〜 省略 〜, 64, 48, null);
disp.displayImage(); //ディスプレイ描画
```

- ⑩ MP3Player のオブジェクトを作成し、5 種類の曲を追加する。
- ```
// 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"));
```
- ⑪ スイッチを一度しか反応させない為に、各 SW の状態を保持する変数 (sw1Data ~ sw4Data) と、前回の各 SW の状態を保持する変数 (sw1DataOld ~ sw4DataOld) を定義する。

【メイン処理】

- ⑫ プログラムを終了させない為に無限ループを記述し、内部に以降⑭までの処理を記述する。
- ⑬ 各スイッチが押されているかどうか状態を読み取り、変数（sw1Data ~ sw4Data）に保持する。
- ⑭ SW1 に割り当てられた機能の処理を記述する。

// 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(省略){  
    if(sw1Data == ON){  
        ~ 省略 ~; //LED 点灯  
  
        // 一時停止または停止している場合  
        if(~ 省略 ~ || player.isStopped()){  
            disp.getGraphics().clearRect(32, 48, 16, 16); // 再生画像クリア  
            disp.getGraphics().drawImage(~ 省略 ~, 32, 48, null); //一時停止画像設定  
        }  
        ~ 省略 ~; //前の曲へ  
  
        // 操作音を鳴らす  
        Gpio.pwmSetClock(192000 / 省略);  
        Thread.sleep(~ 省略 ~);  
        Gpio.pwmSetClock(~ 省略 ~); //消音  
    }else{  
        ~ 省略 ~; //LED 消灯  
    }  
}
```

- ⑮ 上記と同様に SW2 に割り当てられた機能の処理を記述する。(以下 SW が ON になった時の処理例)

省略; //LED 点灯

disp.getGraphics().clearRect(~ 省略 ~); // 再生・一時停止画像クリア

```
// 一時停止または停止している場合  
if(~ 省略 ~ || player.isStopped()){  
    ~ 省略 ~; //再生  
    disp.getGraphics().drawImage(~ 省略 ~); //一時停止画像設定  
}else{  
    ~ 省略 ~; //一時停止  
    disp.getGraphics().drawImage(~ 省略 ~); //再生画像設定  
}  
  
// 操作音を鳴らす（以下省略）
```

- ⑯ 上記と同様に SW3 に割り当てられた機能の処理を記述する。(以下 SW が ON になった時の処理例)

```
～ 省略 ～; //LED 点灯
```

```
// 一時停止または停止している場合
```

```
if(～ 省略 ～ || player.isStopped()){  
    disp.getGraphics().clearRect(32, 48, 16, 16); // 再生画像クリア  
    disp.getGraphics().drawImage(～ 省略 ～); //一時停止画像設定  
}  
～ 省略 ～; //次の曲へ
```

```
// 操作音を鳴らす（以下省略）
```

- ⑰ 上記と同様に SW4 に割り当てられた機能の処理を記述する。(以下 SW が ON になった時の処理例)

```
～ 省略 ～; //LED 点灯
```

```
// 繰り返しがオンの場合
```

```
if(～ 省略 ～){  
    ～ 省略 ～; //繰り返しオフ  
    disp.getGraphics().clearRect(96, 48, 16, 16); //繰り返し画像クリア  
}else{  
    ～ 省略 ～; //繰り返しオン  
    disp.getGraphics().drawImage(～ 省略 ～, 96, 48, null); //繰り返し画像設定  
}
```

```
// 操作音を鳴らす（以下省略）
```

- ⑱ 各 SW 機能の処理で設定した画像をディスプレイに描画する。
⑲ 次回ループの為に現在の各 SW の状態を保存する。
⑳ チャタリング防止の為に 100 ミリ秒待機する。

以上、完成してチェックを受けた人は「自分の好きな曲を追加する」「曲の再生中はフルカラーLED がグラデーションで光り続ける」といった追加機能を考えて実装してみましょう。

Raspberry Pi 環境終了

今回の総合課題を持って Raspberry Pi 環境での演習は終了します。

第 10 回から新たな環境での演習となる為、それ以降 Raspberry Pi 課題のチェックは受け付けません。次回の第 9 回は講義を進めない調整回としますので、これまで作成できていない課題がある場合は必ず作成し、忘れずに講師からチェックを受けましょう。

※チェックを受けなかった場合、課題点が付かなくなり、結果成績点も下がります。