

ドコモ スマートフォン

M1 Z-01K

アプリ作成ガイドライン

第 1.0 版

2018 年 2 月 9 日

株式会社 NTT ドコモ

目次

1章	はじめに	4
2章	Z-01K仕様	5
2.1.	表示モード	5
2.2.	端末の状態と利用可能な表示モード	6
3章	表示モード変更に対応する	7
3.1.	各表示モードの情報	7
3.2.	表示モード変更通知への対応	8
3.3.	2画面モードへの対応	9
3.4.	端末の開閉状態の変化への対応	10
4章	カメラアプリ作成時における注意事項	12
4.1.	カメラの切り替え	12
4.2.	クローズ状態でのインカメラ動作	12
4.3.	クローズ状態でのアウトカメラ動作	12
4.4.	スタンディング状態、オープン状態でのカメラ動作	13
4.5.	フラッシュライト使用時の注意	13

商標について

- Android、Android ロゴは Google Inc. の商標または登録商標です。
- その他、掲載されている会社名、製品名、サービス名は各社の商標または登録商標です。
- 本書ではコピーライト及び商標・登録商標表記はしていません。

改版履歴

版	項目	種別	内容
1.0	—		初版作成

1章 はじめに

本ドキュメントは、2017年冬モデルとして発売する、「ドコモスマートフォン M Z-01K」（以降、「Z-01K」と表記します）に向けてアプリケーションを提供するにあたっての注意事項を解説したものです。

このドキュメントは、Z-01Kの特有の振舞いを考慮したAndroidアプリケーション開発について解説しています。したがって、本ドキュメントはAndroidアプリケーション開発の基礎的な知識を有する読者を対象としています。



このドキュメントでは、2章にてZ-01Kの画面表示に関する仕様を解説します。3章以降では、アプリケーション開発時に特に注意すべき事柄について解説します。

2章 Z-01K 仕様

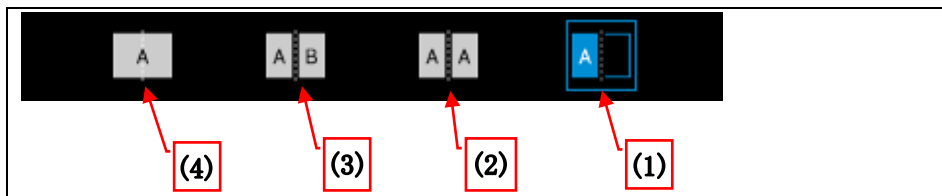
本章では、Z-01K 独自の拡張仕様のうち、画面表示に関する部分について、解説します。

2.1. 表示モード

Z-01K には、4つの表示モードがあります。各モードは **M** をタップすることで切り替えます。

モード切替できる場合	
モード切替できない場合 (M が淡色化されている)	

M をタップするとモード切替が表示されます。



(1) 通常モード

端末の片方の画面のみを使用し、もう片方の画面を使用しないモードです。

(2) ミラーモード

端末の両方の画面に同じ内容を表示するモードです。向かい合った状態で、同じ内容を見ることができます。

(3) 2画面モード

2つの画面を別画面として使用するモードです。この状態では、2つの異なるアプリを並べて同時に利用することができます。¹

(4) 大画面モード

2つの画面を一つの画面として使用するモードです。

¹ 2つの画面で、それぞれ別のアプリを表示および操作することが可能ですが、一部利用できないアプリがあります。

2.2. 端末の状態と利用可能な表示モード

Z-01K は、端末の開閉により、3つの状態を持ちます。各状態で利用できる表示モードが異なるため、注意が必要です。表示モードの切り替えは、**M**をタップします。

(1) クローズ状態(デフォルト)

端末を畳んだ状態です。この状態では通常のスマートフォンとして使用することができます。この状態では、**M**をタップして表示モードを変更することはできません。

■ 選択可能な表示モード

- 通常モード

(2) スタンディング状態

端末をL字に開いた状態です。この状態では、端末を立てて利用できます。この状態では、以下の2つのモードを選択することができます。

■ 選択可能なモード(**M**をタップした際の表示との対応)

- 通常モード
- ミラーモード



(3) オープン状態

端末を完全に開いた状態です。この状態ではスタンディング状態の表示に加え、2つの画面を1つの大画面として利用したり、2つの異なるアプリを同時に利用したりすることができます。

■ 選択可能なモード(**M**をタップした際の表示との対応)

- 通常モード
- ミラーモード
- 2画面モード
- 大画面モード



3章 表示モード変更に対応する

3.1. 各表示モードの情報

Z-01K の各表示モードは以下のような値になっています。

(1) 縦持ちの場合

表示モード	orientation	densityDpi	screen HeightDp	screen WidthDp	resolusion	isInMulti WindowMode
標準モード	PORTRAIT	480	640	360	1920x1080	false
ミラーモード	PORTRAIT	480	640	360	1920x1080	false
2画面モード	PORTRAIT	480	640	360	1920x1080	true
全画面モード	LANDSCAPE	520	590.8	664.6	1920x2160	false

※ densityDpi, screenHeightDp, screenWidthDp の各値はデフォルト時の値です。

※ 2画面モード時に Display.getRealSize で取得すると全画面モードと同じ幅になりますが、分割表示のため、アプリ表示の幅は、2160ではなく、1080となります。

※全画面モードでは、縦横比が逆転するため、LANDSCAPE になります。

(2) 横持ちの場合

表示モード	orientation	densityDpi	screen HeightDp	screen WidthDp	resolusion	isInMulti WindowMode
標準モード	LANDSCAPE	480	360	640	1080x1920	false
ミラーモード	LANDSCAPE	480	360	640	1080x1920	false
2画面モード	LANDSCAPE	480	360	640	1080x1920	true
全画面モード	PORTRAIT	520	664.6	590.8	2160x1920	false

※ densityDpi, screenHeightDp, screenWidthDp の各値はデフォルト時の値です。

※ 2画面モード時に Display.getRealSize で取得すると全画面モードと同じ高さになりますが、分割表示のため、アプリ表示の高さは、2160ではなく、1080となります。

※全画面モードでは、縦横比が逆転するため、PORTRAIT になります。

3.2. 表示モード変更通知への対応

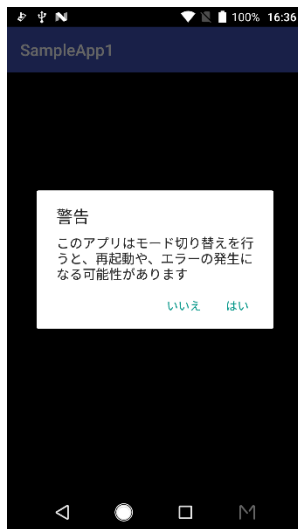
Z-01K では、表示モード変更にともない、画面サイズと画面の向き、画面密度が変更されます。表示モード変更の通知を受信するかどうかにより、Activity の更新動作が異なります。

(1) 表示モード変更の通知を受信する場合

AndroidManifest.xml でアクティビティに対し、configChanges のパラメータとして density, orientation, screenLayout, screenSize, smallestScreenSize を指定することで、通知を受け取ることができます。この設定により、Activity 内で onConfigurationChanged を実装することにより、表示モードの変更をアプリ内で検知することができます。

(2) 表示モード変更の通知を受信しない場合

AndroidManifest.xml でアクティビティに対し、configChanges を指定していない場合、もしくは(1)に記載のパラメータが指定されていない場合、表示モード変更時に Activity を再生成します。このため、表示モード変更前にアプリ実行画面上に以下のように警告ダイアログが表示します。警告に同意して、表示モードを変更すると Activity 内で保持しているデータが消失し、予期しない動作がおこる可能性があります。表示モード変更に関する通知を受け取らない場合、Activity が破棄されることを考慮し、Activity 外でのデータを保持する、表示モード変更時にアプリを停止するなどをご検討ください。



標準モード、2画面モード時の表示



全画面モード時の表示

なお、ミラーモードについては、通常モードと同様となるため、表示モードを通常モードとミラーモードとの間で変更する場合は上記に該当しません。

3.3. 2画面モードへの対応

2画面モードは、Android 7.0 (Nougat)でサポートされたマルチウインドウ機能に基づいて実現しています。このため、Android 開発者サイトのマルチウインドウサポートのページを参考に実装してください。

なお、表示モードを2画面モードにする場合、targetSdkVersionが23以下(Android 7.0以前)のアプリなどマルチウインドウに対応していないアプリでは、以下のトーストが表示されます。

2画面モードでは、アプリが正常動作しない恐れがあります。

また、2画面モードでは、ナビゲーションバーの表示されない側のアプリは、Pausedの状態となります。onPausedで処理を停止するとアプリの切り替えにより、動作が停止しますのでご注意ください。

詳しくは、Android 開発者サイトの下記のドキュメントをご参照ください。^{2 3}

² Home > Android > Nougat

Android 7.0 for Developers - Multi-window Support

https://developer.android.com/about/versions/nougat/android-7.0.html#multi-window_support

³ 開発 > API ガイド > User Interface

Multi-Window Support

<https://developer.android.com/guide/topics/ui/multi-window.html>

3.4. 端末の開閉状態の変化への対応

端末の開閉状態が変化したことを検知するために、Z-01Kには独自のセンサー(HallSensor)が内蔵されています。

このセンサーのイベント取得することで、端末の開閉状態（クローズ状態、スタンディング状態、オープン状態）が変化したことを検知できます。

このセンサーを利用するためには、リスナを作成し、センサータイプ 34 のセンサーのリスナとして登録します。

リスナの onSensorChanged メソッドの引数から以下の値が取得できます。

event.values[0]の値	値の意味
1.0	クローズ状態に変化した
2.0	スタンディング状態に変化した
3.0	オープン状態に変化した

これらの値により、端末がどの状態になったかを判断することができます。

コード例を下記に記載します。単純化のため、エラー処理等、説明に不要な部分を省いております。実装する際はエラー処理等を適切に実装してください。

```
private static final int SENSOR_TYPE_HALL = 34;
private static final int HALL_STATE_CLOSE = 1;
private static final int HALL_STATE_STANDING = 2;
private static final int HALL_STATE_OPEN = 3;

private SensorManager mSensorManager;
private Sensor mSensor;
private SensorEventListener mSensorListener = new SensorEventListener() {
    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == SENSOR_TYPE_HALL) {
            int value = (int)event.values[0];
            switch (value) {
                case HALL_STATE_CLOSE:
                    // 端末がクローズ状態になった場合の処理
                    break;
                case HALL_STATE_STANDING:
                    // 端末がスタンディング状態になった場合の処理
                    break;
                case HALL_STATE_OPEN:
                    // 端末がオープン状態になった場合の処理
                    break;
            }
        }
    }
};
<< 省略 >>
private void registerSensor() {
    private mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    private mSensor = mSensorManager.getDefaultSensor(SENSOR_TYPE_HALL);
    mSensorManager.registerListener(mSensorListener, mSensor, SensorManager.SENSOR_DELAY_NORMAL);
}
```

4章 カメラアプリ作成時における注意事項

4.1. カメラの切り替え

Z-01K に搭載されているカメラモジュールは1つですが、アプリを表示する画面を変更することで、インカメラとアウトカメラを切り替えて動作します。

インカメラ、アウトカメラを切り替えて利用できるのは、端末がクローズ状態(デフォルト)の場合のみです。

以降の節で詳細を説明します。

4.2. クローズ状態でのインカメラ動作

端末がクローズ状態の場合、アプリで Camera2 クラスのインスタンスを生成し、インカメラ (LENS_FACING_FRONT) として開始した際、以下のトーストが表示されます。

現在自分撮りモードです、メイン画面側にあるカメラで撮影します。

このモードでは、常にフラッシュライトが利用できません。

4.3. クローズ状態でのアウトカメラ動作

端末がクローズ状態の場合、アプリで Camera2 クラスのインスタンスを生成し、アウトカメラ (LENS_FACING_BACK) として開始すると、以下のトーストがメイン画面に表示され、アプリの表示面が裏側の画面に切り替わります。

まもなくメイン画面が消灯します、端末を反対側に向けて裏側にあるカメラで撮影します。

このモードでは、フラッシュライトを利用することができます。

4.4. スタンディング状態、オープン状態でのカメラ動作

端末がスタンディング状態やオープン状態でアプリから Camera2 クラスのインスタンスを生成し、開始するとインカメラ、アウトカメラの指定に関わらず、下記の内容のトーストが表示されます。

端末を閉じた状態でアプリを一度再起動してください。そのあと、このアプリのカメラ機能が正常に使えます。

この状態では、強制的にインカメラとして動作しており、フラッシュライトを利用することはできません。

4.5. フラッシュライト使用時の注意

端末がクローズ状態の時にアウトカメラを起動し、フラッシュライトをトーチ (FLASH_MODE_TORCH) としライトを点灯したまま端末をオープン状態にすると、目を傷める可能性がありますのでご注意ください。