
**by Izaron**

## Vibration.cs methods overview

| | | |
|---|---|---|
| `void` | `Vibrate()` | Vibrate constantly for the short period of time |
| `void` | `Vibrate(long milliseconds)` | Vibrate constantly for the specified period of time |
| `void` | `Vibrate(long[] pattern, int repeat)` | Vibrate with a given pattern<br><br>Pass in an array of ints that are the durations for which to turn on or off the vibrator in milliseconds.<br><br>The first value indicates the number of milliseconds to wait before turning the vibrator on. The next value indicates the number of milliseconds for which to keep the vibrator on before turning it off.<br>Subsequent values alternate between durations in milliseconds to turn the vibrator off or to turn the vibrator on.<br><br>To cause the pattern to repeat, pass the index into the pattern array at which to start the repeat, or -1 to disable repeating |
| `bool` | `HasVibrator()` | Check whether the hardware has a vibrator. Returns true if the hardware has a vibrator, else false. |
| `void` | `Cancel()` | Turn the vibrator off |

# Step-by-step guide

The plugin “Vibration for Android” is written on Java with Android API use.

In the **Plugins/Android** folder there is **Vibration.jar**. It is library for vibration, created in Eclipse

**AndroidManifest.xml** in the same folder is necessary for .apk compilation with this library

There is a line in manifest

```
<uses-permission android:name="android.permission.VIBRATE"/>
```

It means that compiled .apk will ask for permission to use vibration for this application

In the **VibrationPlugin/Sources** folder there are two scripts which are a cover for Vibration.jar

At first create a script Example.cs

```
void Start()
{
    if (!Vibration.HasVibrator())
        Application.Quit();
}
```

Also set it on any object in a scene

If the device can't vibrate, we close application

```
void Start()
{
    Vibration.Vibrate();
}
```

After scene loading the device will begin to vibrate!

But also it is possible to vibrate long time. Change the Start()

```
void Start()
{
    Vibration.Vibrate(2000);
}
```

It means that vibration will proceed 2000 milliseconds (these are two seconds)

Also we can use patterns

```
void Start()
{
    // Start without a delay
```

```
        // Vibrate for 100 milliseconds
        // Sleep for 1000 milliseconds
        long[] pattern = { 0, 100, 1000 };

        // The '0' here means to repeat indefinitely
        // '-1' would play the vibration once
        Vibration.Vibrate(pattern, 0);
    }
```

And, at last, at any time it is possible to switch off vibration

```
    void OnMouseDown()
    {
        Vibration.Cancel();
    }
```

To use all opportunities of a plugin, write it

```
    void OnGUI()
    {
        if (GUI.Button(new Rect(10, 10, 250, 50), "Vibrate();"))
            Vibration.Vibrate();

        if (GUI.Button(new Rect(10, 90, 250, 50), "Vibrate(2000);"))
            Vibration.Vibrate(2000);

        if (GUI.Button(new Rect(10, 170, 250, 50), "Vibrate(4000);"))
            Vibration.Vibrate(4000);

        if (GUI.Button(new Rect(10, 250, 250, 50), "Vibrate(8000);"))
            Vibration.Vibrate(8000);

        if (GUI.Button(new Rect(10, 330, 250, 50), "Cancel();"))
            Vibration.Cancel();
    }
```

The good example of use of a plugin is in the **VibrationPlugin/Sources** folder