User Manual for using Gesture Monitor App with a Wristband Sensor

Gesture Monitor App has been designed to log data from up to 3 Invensense devices. It is recommended to collect data from at least 2 devices so that you can later reconstruct hand-gesture motion in the 3D visualization application. For hand-gesture motions, we recommend using devices at following on-body locations:

- **Wrist**: Use a wristband that has Invensense MPU-9150/MPU-9250 embedded in it.
- **Upper Arm**: Use an armband similar to what is used to hold Ipod Nano or a smartphone that people use while running/jogging. Place an Invensense device in this armband in an orientation that is the same as the orientation of the wristband.
- **Chest (Optional)**: Use a chestband to attach an Invensense device close to the center of the chest in the same orientation as the orientation of the device in the wristband.

**Setting Up Wristband to work with the phone**
Before the wristband sensor can work with Gesture Monitor App, we need to pair the sensor with the phone using Bluetooth. The following lists the steps involved:

- Locate the button on the wristband sensor to turn it on. The button is hidden inside the wristband and can be located as shown in the picture below. Slide the button to switch on the sensor.
• To make sure that the sensor is on, flip the wristband and notice if you can see a light glowing as shown here

![Light indicating that the sensor is on](image)

• Next, open the Bluetooth settings on the phone. Search for a new device and click on it to pair with it. Upon successful pairing, the wristband sensor will show up as ‘BlueRadios’ in the list of available devices under Bluetooth settings.

**Charging the wristband and the Correct Way to wear it**

The wristband sensor can be charged using a micro-USB cable that is used for charging Android phones. The sensor runs for 4 hours when fully charged. The image below shows a micro-USB cable connected to the wristband. Make sure that the sensor is turned off while charging. The image below also shows the correct way of wearing the wristband. The charging point should be towards the left when worn correctly on the right-hand.

![Connect microUSB cable to charge here](image)
**Setting up Armband and the Chestband for logging**

The Invensense devices in the armband and the chestband should be paired the same way as described earlier (Bluetooth pairing). One important thing to keep in mind is the orientation of these devices in the armband and the chestband. When the arms are hanging straight and loose and the chest of the user is straight, the Invensense devices should align with the direction of the gravity with microUSB charging point facing down towards the ground. The visible battery should face outwards away from the body. This way, all the devices will have same orientation and the data from these devices can later be used to reconstruct hand-gesture motions.

**NOTE:** You must explicitly note down the Bluetooth ID of each device you paired. You will need this information later to identify which device is associated with the wrist, arm and the chest.

---

**Logging with Gesture Monitor App**

**Setting Up Gesture Monitor:** Launch the Gesture Monitor application. You can find the settings in a drop-down menu towards the right corner of the application. Open the settings and it should show a screen as shown towards the left. Ensure that the ‘IMU (Lower Arm) Enable IMU sensor’ option is checked under Logging options as shown in the
image. Next, click on the ‘IMU (Lower arm)’ option from the devices option.

Upon clicking the option, you will see a screen as shown on the left. Select the device you paired recently for the wristband. Click on the device.

Similarly, you should enable logging for armband (denoted by Upper arm) and for the chestband and select appropriate devices for each.
Connecting/Disconnecting from the Sensor
Towards the top of the application, there is a button with Bluetooth icon. It gives you two options: ‘Start Logging’ that connects the application to all the sensors you enabled and ‘Pause Logging’ that disconnects the application from the sensor. If the sensor connects successfully, the application shows the message ‘Connected’ towards the bottom of the application. Else it may show messages like ‘Reconnecting…’. When the sensor's battery is dead, the application shows ‘Reconnecting...’ message.
**Reporting Events or Ground Truth**

The Gesture Monitor application allows a user to report the ground truth i.e. when a person smoked a cigarette or did some activity similar to smoking like eating, drinking, etc. These ground truth reports help us in improving the smoking detection algorithm. For example, if a user clicks on the ‘Smoking’ button, the application shows the options as shown in the first figure below. A user can select the appropriate response. Please, advise the user to report at least the start and the end of each smoking episode or any other gesture activity you want to visualization such as eating or drinking.

![Report Event](image)

**Viewing and Correcting Reports**

Sometimes a user may mistakenly submit a report. In such scenarios, a user can correct the mistakes in the report by by updating or deleting it. The reports can be seen in the ‘Reports’ tab of the application and can be corrected by clicking on the appropriate report. The screenshot on the left shows sample reports submitted by a user.
Configuring Buttons in UI

Our application by default shows buttons for smoking, eating, drinking and other gestures. But if you want to configure these buttons, you may do so by updating the Android source code of the application. This can be done in the resource file res/values/arrays.xml. The following shows the current values in the resource file.

```xml
<resources>
  <string-array name="events">
    <item>Smoking</item>
    <item>Eating</item>
    <item>Drinking</item>
    <item>Other</item>
  </string-array>
  <string-array name="smoking_event">
    <item>Before smoking</item>
    <item>During smoking</item>
    <item>After smoking</item>
  </string-array>
  <string-array name="eating_event">
    <item>Before eating</item>
    <item>During eating</item>
    <item>After eating</item>
  </string-array>
  <string-array name="drinking_event">
    <item>Before drinking</item>
    <item>During drinking</item>
    <item>After drinking</item>
  </string-array>
  <string-array name="other_event">
    <item>Before</item>
    <item>During</item>
    <item>After</item>
  </string-array>
</resources>

If you want to remove say 'Smoking' and in stead want to create a button for some activity X where let's say X='Push-ups', then, you should take following steps:

1. Replace 'Smoking' with X i.e. 'Push-ups' in the "events" array.
2. Create a string-array with name="X_event" where X="push-ups" in this case.
3. Our app will first create buttons for all the events in the "events" array and will create options from the "X_event" array for each "X" observed in the "events"
4. You may create as many buttons as you want by adding more values in "events" array and appropriate arrays for the options for added events.
5. Now, compile the Android project and install it on the Android phone. You will see your app with the buttons configured in the resource file.
**Downloading Data from the phone to your computer**

Gesture Monitor application stored the collected data locally on the phone in a database. In order to download it to your computer, you will be first required to export the database to csv files. You can do so by clicking ‘Export DB’ button given on the top of the application.

The exported data is stored locally on the phone in the location ‘Downloads/mcrowd’ directory in the form of csv files. Invensense data files follow the naming convention InvensenseXY.csv where X indicates the IMU device number. Device number 1 is used for the armband whereas 1 is used for the wristband and 3 is used for the chestband. Y is the timestamp when the files were first created i.e. the time when the first sensor data was obtained for this file. You will see numerous files for each device if logged for sufficiently long period. You will also see Reports<timestamp>.csv file containing collected reports. Here the timestamp is the time when the reports were exported from the database. A new instance of Reports file is generated every time ‘Export DB’ button is clicked on the app.

To download the data, you can use Android tools like ddms that can be executed from the command line. You can use ddms to explore files on the phone connected to the computer via USB. Depending on the device, you may find 'mcrowd' directory in a location such as /storage/emulated/0/Downloads/. Use ‘Save’ button in the ddms file browser program to save the files to your computer.

**Deleting Data on Phone**

You can also delete all the data on your phone using ‘Delete data’ button given on the top of the application along with the ‘Export DB’ button. You should be careful with the delete operation, as the data once deleted cannot be recovered.