

# CAS765 Programming Assignment 2 – Data collector

**Due date: Oct 28th, 2013**

In this programming assignment, you will develop Java codes on Android for collecting data from the mobile phone's accelerometer, gyro, compass, and WiFi interface, and store the data on the external storage (e.g., SD card) on the devices. Similar to Project 1, sample codes have been provided with some coding blocks left out. You are expected to test your program on a real device for demonstration.

A screen shot of the application running on an Android device can be found in Figure 1(a). As can be seen in Figure 1(a), the user can choose among 4 types of sensors: accelerometer, compass, Gyro and WiFi using checkboxes. A checkbox is disabled if the sensor is not present on the device and thus cannot be used to acquire sensing data. For simplicity, the first three sensors share the same sampling rate, which can be specified using a slide bar. The textbox below the slide bar displays the sampling rate selected in Hz. WiFi scan is performed as fast as possible, namely, when the probe responses are received, the next probe starts. Upon selecting interested sensors and clicking "Start", the sensor readings will be displayed as shown in Figure 1(b). In the text box under WiFi, information from a list of APs can be viewed by scrolling down. SSID, MAC address, capability information, signal strength (in dBm), and channel are displayed.

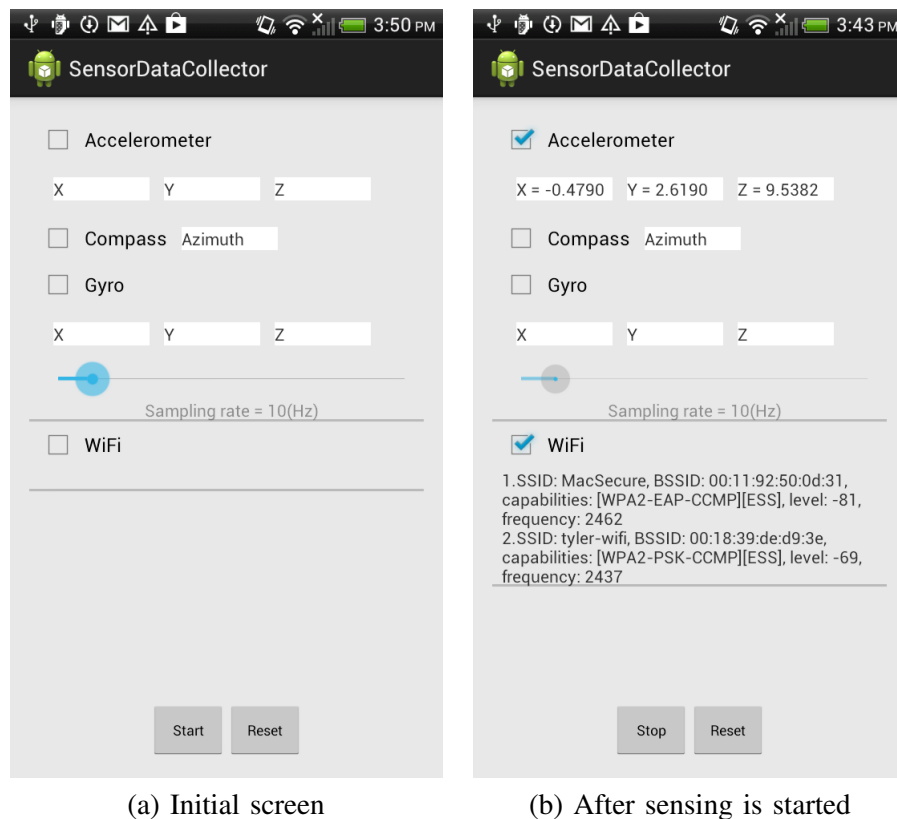


Fig. 1. Screen shot of the execution on an Android device

**Step 1: Basic setup:** Follow the instructions provided during the lecture to set up your ADT. Download the project code from the course website, and import to your own work place. When connecting the device to the

IDE for debugging, enable “USB debugging” and select the USB connection type to “Charge only” (as opposed to “Disk drive”). To access the files stored on the device, on the other hand, you will need to set the USB connection type as “Disk drive”).

**Step 2: Set up permissions:** Modify the `AndroidManifest.xml` file to allow accessing and changing `WIFI_STATE`, read and write from/to external storage.

**Step 3: Handle sensor readings:** Modify the `startSensor` method to register listeners for accelerometer, gyro, and compass sensors. The sampling rate is specified by the variable `mSensorSamplingRate`, which can be changed by moving the slide bar. Next, modify the `onSensorChanged` method to display the readings on the respective text boxes and store the readings in the respective file. You can utilize the `SensorReading` class. Note that each time the application is started, a new directory will be created on the external storage. Sensor readings will be written to separate files, namely, `acc`, `gyro`, `compass` and `wifi` under the directory.

To this end, you should be able to select the type of sensors for measurement, start, stop, display and store the sensor readings.

**Step 4: Handling WiFi scan:** Unlike sensors, to obtain WiFi scans, we need to register a broadcast receiver `WifiReceiver` with intent `WifiManager.SCAN_RESULTS_AVAILABLE_ACTION`. Upon availability of the scan results, the `onReceiver` method will be triggered. Therefore, you will need to modify the `startWiFi` method for creation and registration of `WifiReceiver`. Additionally, you need to modify the `onReceive` method in the `WifiReceiver` class to display the readings on the UI and write to the storage. A list variable `wifiList` is provided to store the scan results from the `WifiManager` in the method.

**Step 5: Stop sensors and WiFi scan:** When the user presses “Stop”, the sensor listener and `WifiReceiver` should be unregistered. Additionally, the stream writers should be flushed so that data in the memory will be written to the external storage. You need to accomplish this by modifying the `stopAll` method.

Please put your codes in a `tar.gz` file and submit via email to the instructor. Though three students can work in a group due to the limited number of devices, **individual student** needs to develop his/her own programs and uses the shared device for testing. During the class time, each student will be asked to give a small demonstration. Please name your application differently and install it on the phone.