Accessing Device Sensor APIs in Advanced Mobile Platform

Many of the competitive smart phones devices do have their different specification, form factors and patents. The sensor features are available in most of the advanced smart phones now a day. However few of them are still not having all sensor support, which can be used on implementing and featuring with the web services and having third-party dependence on the nature of the case study. This section highlights the usage of embedded sensor APIs in Smartphone and accessing them by reading their data at the viewable UI component, i.e., on screen display over the application. Here are the best practice to impellent and steps on using mobile sensor APIs into the application:

a) *Sensor Management*

The sensor management package is a standard application that provides sensor management services to other applications on the device. This provides the sensor enabled mobile device user and client applications with an interface to list installed support packages and invoke them in limited ways.

b) *Sensor Support Packages*

Individual sensor support packages provide the features necessary to interface with a specific sensor or set of sensors.

c) *Sensor Availability*

For each sensor support package and permissions is required to be implemented. Any broadcast it receives must not be aborted.

d) *Sensor Communication*

Each sensor support package will handle all communication details with the actual sensor.

e) *Sensor Configuration*

Each sensor support package will implement an interface to view and change the individual sensor's configuration. This interface will handle all communication with the sensor and will provide an interface specific to the individual sensor.

f) *Sensor Sampling & Capture*

Each sensor support package will implement an interface to capture samples from supported sensors

g) *Sensor Capture Viewing*

Each sensor support package will implement an UI activity that will provide an interface to view a sample capture from the specific sensor.

h) *Sensor Identification*

A standard way of identifying sensor types and specific sensors (including communication information) is required: this ensures that new applications will be able to use the sensor support system and new sensors can be cleanly added to the system.

i) *Sensor Type Identification*

The sensor type identification scheme builds upon the mime-type support. All sensors are identified using a basic mime-type of *application/vnd.sensor.{sensor type}.{variant}*

j) *Sensor Selection and Communication Channel Identification*

The sensor selection and communication channel identification is built on the URI handling system. Sensors may use different communication channels to interact with the sensor embedded mobile device: the nominal default communication channel is Bluetooth.

In this research, we have developed a sample mobile application prototype which has been deployed successfully in advanced sensor embedded mobile phone to access or check the dynamic mobile sensing, including pervasive sensing, object localization, tracking, and wireless communication protocols.

Sensor API Reading Data View in Mobile Phone

We are also going to see here a sample application about how to access or play with the embedded sensor enabled APIs in BlackBerry10 device and retrieve data from it through the piece of code snippets which has been presented in the Appendix.

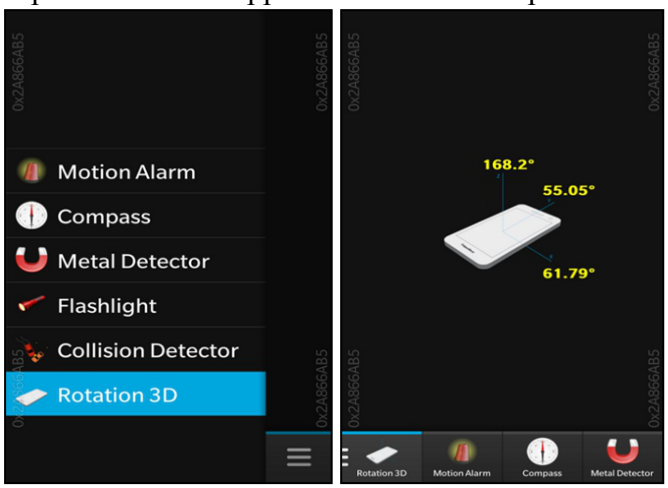

Figure: Sensor APIs Prototype UI View in Mobile Screen

The above UI screen shows the data reading of the various sensor types, such as, motion alarm, compass, metals detector, flashlight, collision detector and rotation. All these sensors APIs have been written and used in native programming language which does have support on top of the embedded sensor driver powered in the hardware of the smartphone. All of them are useful and embedded into the hardware on supporting various kind of application case study and have individual purpose and functionality, being used into the application based on the nature of the business model and application requirement.