

Power of Android Wearable Technology

Saminath

*Team Lead Accenture

Abstract- Android is an open source mobile device operating system has rapidly recognized in electronics Industry. Android operating system runs in embedded products like smart phones, tablet, set-top box, Android TV, Auto-cars - Android Auto, and wrist watches - Android Wear. Android reaches consumer product beyond its envelop on Multimedia, Mobile Internet Devices, Digital video, Home Entertainment, Automotive, Medical, Networking, Instrumentation and Industrial control. Contemporary industrial product of Android Wearable device connected with human body, it monitors and sync with Mobile device. This technology squeezes cutting-edge features like voice activation, calendar, music, biometrics measurement, email and interacts with other handheld device. This paper presents the Technical Detail of Android wearable, Development and Debug techniques of Android Studio. Development system includes cross compilers, libraries, debuggers, pre-built file system images, and product-specific support libraries.

Index Terms- AVD-Android Virtual Device; ADB-Android Debug Bridge.

I. INTRODUCTION

Android Wear technology is tailored made specifically for Google devices like Watch, Glasses. It comes with a new simplified interface with collection of fitness features. Smart watch has advanced health tracker application, it coaches and remind about workouts, speed, distance, and time information. Biometric tools/sensor in watch integrated with fitness Application. It monitors Heart rate and update average Heart Beat information in regular basis.

The calendar application in the watch shows appointments for the day or upcoming week. Just open and see all your appointments for the day and dismiss it. Timer application is simple and useful for utilize your time bandwidth. Estimate you work, update-in timer, you will get notified by the Application. Google maps shows turn driving direction, sync with card label in phone address book application. Message Application will share the incoming message notification from phone. It shows message content, photo of the person.

You can swipe to reply for the fastidious message. Also you could list the history of the conversation. It supports Voice based message composition and sends it. The same way, you can check your emails and reply by text or voice based message formation. Voice call application is most deserved usage in Android wear device, Call notification is received from phone with pop-up indication. You can decline or attend by swipe the card from the watch. Google search application allows you to find the nearest location around you, with latest update on the

day. Android Wear supports on direct information from your phone. It catches Android phone near to you and gives quick responsive action.

Android Wear technology was engineered to work with wireless devices using Bluetooth. Android wear Google API Client provides high level abstract for Bluetooth communication. Three major API for communications: Nodes API (detects devices in the wearable network), Message API (allows sending messages between Nodes) and Data API (allows data synchronization between Nodes). The Wearable apps installed into wearable device, gives direct access to low-level hardware such as Sensors, Memories, and Audio systems.

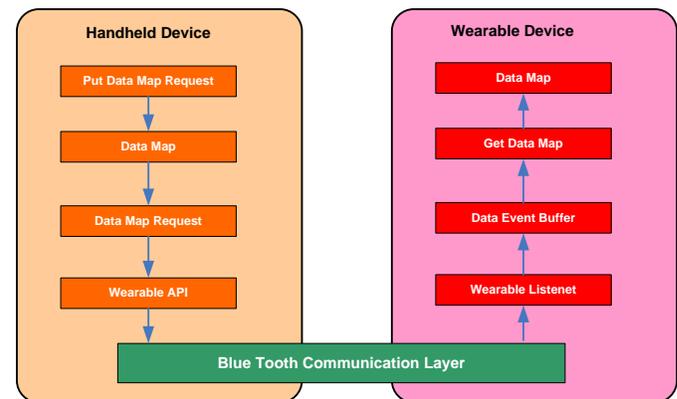


Fig. 1 Communication Layer between Handheld and Wearable Device

Figure-1 explains communication between the Android Handheld device and Wearable Device. Wearable API gets reference from connected mobile device. Data Layer API supports syncing data between handheld and wearable devices. Wearable Listener interface pulling the notifications from paired Android Smartphone's or tablets and display text using Message API. Message API is a one-way communication mechanism that's meant for 'fire-and-forget' tasks". This is just opposed to Wear's Data Layer API. The Google Play service will make this very easy. The communication between two apps over the Bluetooth link that pair two devices.

II. ANDROID WEARABLE COMPONENT

Android wear component allows user to design UI patterns with New Form factored Android Devices. Below are basic Android wearable component for custom development of Android watches.

A. Cards:

Basic UI design of Wearable component is Cards. It has information to users with a respectable look and feel for each UI apps. Card Frame class provides Layout design (Square, Circle) of watch, Size of the Card, Title information, Identification (ID) to each card. Figure-2 shows square watch Card component with Hello message Card Notification.

B. Notification:

Context Stream and Cue card are the two ways to share notification between handheld device and Wearable device when it's connected. The context stream is a vertical list of cards, each showing a useful or timely piece of information. Cards in the stream are more than simple notifications. Cards can also be dismissed by swiping left to right, removing them from the stream until the next time the app has useful information to display. Context Stream will display messages without user permission. Cue-card is Voice based notification (EXTRA_VOICE_REPLY), added in Remote Input. The Builder class of cue-card allows users by saying, "OK Google" or by selecting list of suggested voice commands, which can also be tapped with Text message. Stacking of Notifications are more than one message is received. The single notification to provide a summary of group messages which will be in stack. The stack of notifications appears as a single card, which users can expand to view the details from each notification separately.

C. Pages:

Pages provide additional content information instead of single package information in primary notification. Pages appear immediately to the right of the main notification card. It will be multiple cards for a single notification, a deeper glance with successive cards. It could be a next train departure card, a card showing the weather forecast, scheduled matches etc. Figure-3 and Figure-4 shows the page notification, you can navigate from Left to Right or Top to Bottom.

D. SYNC Data:

Google Play services provide several communication channels between handheld device and wearable device. GoogleApiClient is the Main entry point to Google play service, to integrate sync devices. GoogleApiClient_Builder class provides method that allows you to specify the Google APIs. Data Item is stored in a network, it replicated across all devices in the network. Two functions getData () and setData() will update Item value over the network. Limitation of the Data packet size is 100k.

MessageApi request/response message communication to handheld device. Intern use RPC mechanism for message synchronization. Device should register the Listeners like addListener() and removeListener(), sendMessage() will send the data in bytes format. Asset is another way of transferring the content in BLOB format. It sends a large size of Image and Video's in Bluetooth bandwidth. Wearable Listener Service pays attention of data layer events. It binds and unbinds with required services. On Peer Connectivity triggered when wearable device connected with handheld device. DataListener interface to identify state of the device, whether data items have been changed or deleted.

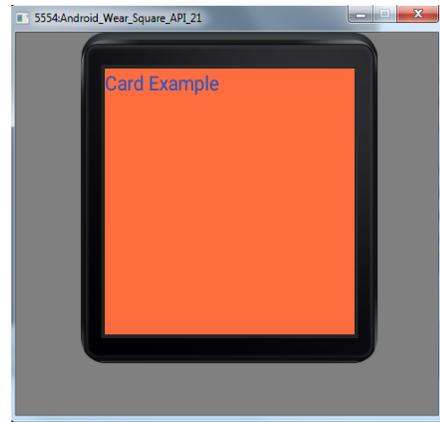


Fig:2 Card Notification

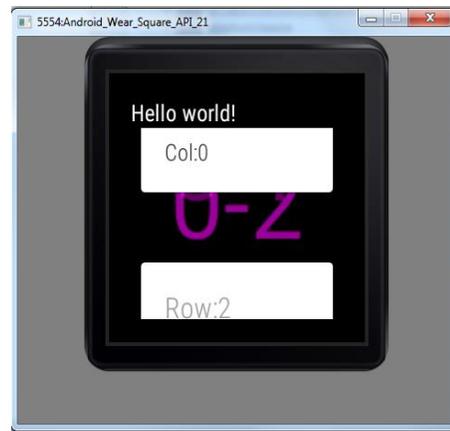


Fig:3 Page Vertical Navigation

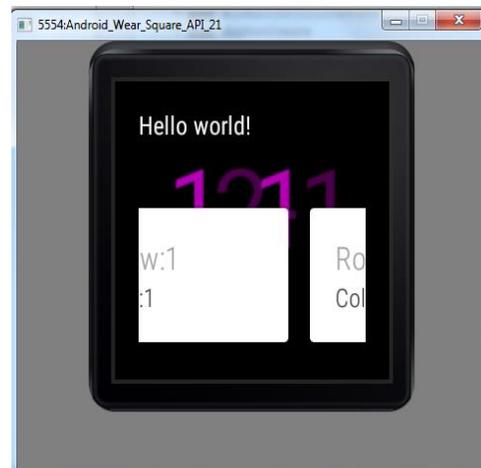


Fig:4 Page Horizontal Nav

III. ANDROID WEARABLE SPECIFICATION

Each Android wear device has different technical specifications. Android wear device make available accurate information at precise time. Various Android Wear partner has different hardware support. Device specification provides various technical feature of the Wearable watch device such as sensors, Battery, shape design. Table-1 explains minimal hardware specification requirement for Android Wear Device.

TABLE I. HARDWARE SPECIFICATION

Android Wear Hardware Functional Specification	
Display Screen	1.63-inch 320 x 320 Special LED configuration
Processor support	Qualcomm Snapdragon with 1.5Ghz
Memory	512MB RAM, 4GB storage
Sensors	Heart-rate monitor, 9-axis accelerometer, GPS sensor, compass, gyroscope
Android OS support	Android 5.0 and implementing API 21 onwards
Screen Face	Round/Square
Battery Support	400mAh, minimum of Complete Day usability
Wireless support	Bluetooth/WiFi
Ports	USB for Debug and Charging

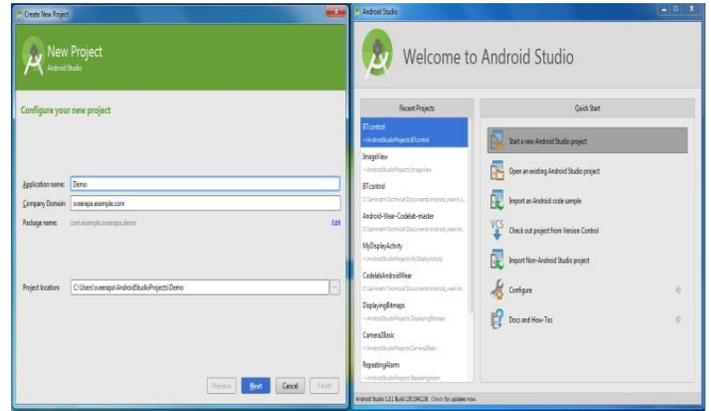


Fig:5 Create Android Wear Project

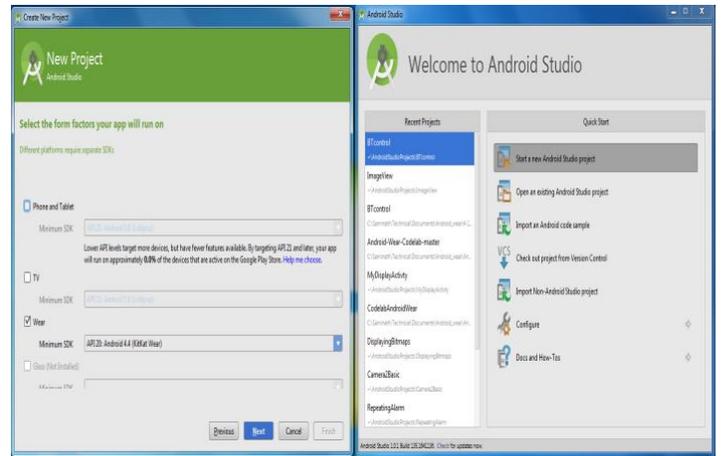


Fig:6 Android SDK Selection

IV. ANDROID BUILD SYSTEM

Android studio is the official IDE for Android wearable device. You can develop application for Android phones, tablets, Android TV, Auto, and Google Glass. To creates a new project with two modules, Mobile and Wear. Figure-5 and Figure-6 explains the creation of new project properties and selection of Platform SDK. New Project follows the Project Wizard instruction with Application Name. Select Wear and select API 20: Android 4.4 (KitKat Wear) under Minimum SDK. The Blank Wear Activity creates a different layout for each of the Android Wear screens: Round layout (Android Wear Round) and Rectangular (Android Wear Square) layout as shown Figure-7. You must create an Android Wear virtual device with the Android Virtual Device Manager, also known as AVD Manager. Launch the AVD Manager from Android Studio and click Create Virtual Device., select wear from the Run/Debug configuration drop-down menu and click the Play button. The activity shows up on the wearable simulator and prints out "Hello Square world!" as shown in Figure-8.

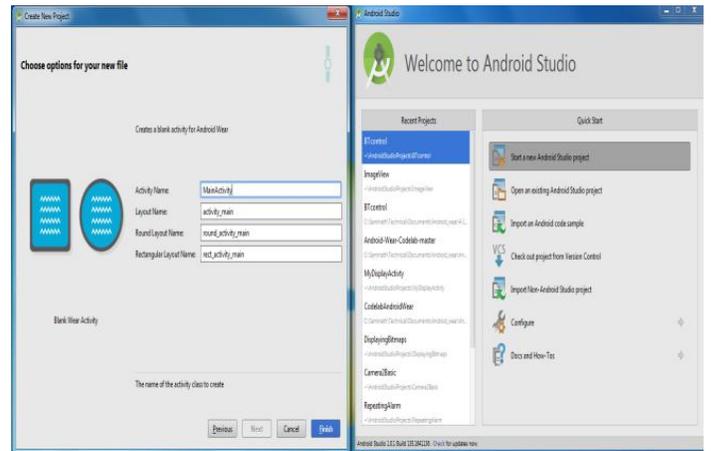


Fig:7 Selection of Android faces

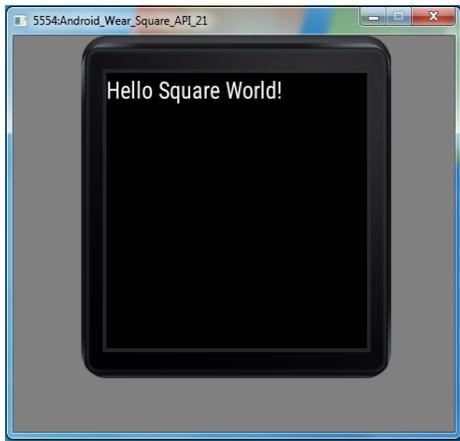


Fig:8 Simulator output

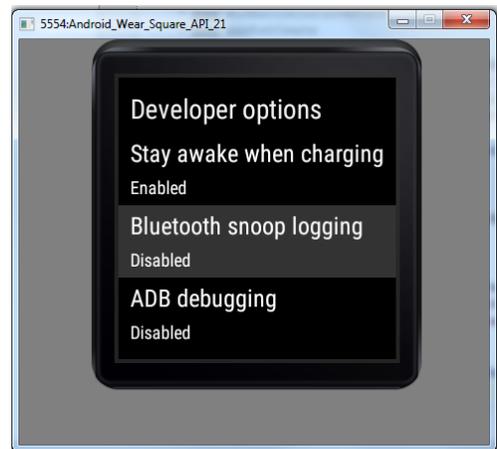


Fig:10 Debug Selection

V. ANDROID DEBUG SYSTEM

Android Studio enables to debug apps running on the emulator or on Android Wearable device. In Android Wearable device, debug system is disabled as default. It will be enabled by opening Setting and About menu entries. Tab the Build number by 7 times, you can see the Developer option enabled like in Figure-9. Figure-10 explains about the Open the Developer Options, Select the ADB Debugging and Debug over Bluetooth settings. Connect the Wear device with system, via USB. It will ask the RSA key enable, should be always selected. Figure-11 shows the screen info of RSA key configuration.

Another way for debug communication via adb over wireless Bluetooth, this is achieved by forwarding socket. Enable the Bluetooth Debug over Bluetooth in Wearable device. Attach Android phone to computer over USB cable. In Android phone run Android wear companion application, Open Settings, Enable Debugging over Bluetooth

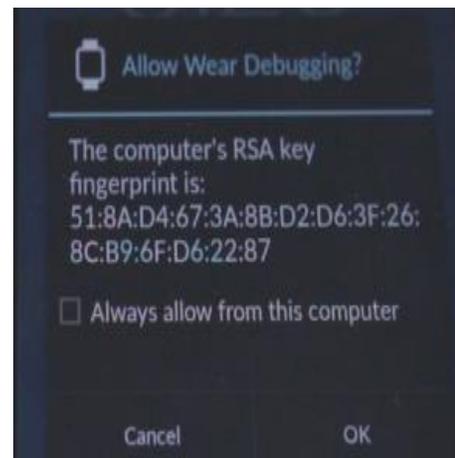


Fig:11 RSA Key configuration



Fig:9 Developer Mode selection

VI. CONCLUSION

Android Wear smartwatches allows to access and control right from their wrists. Your Identity will be reflected in your watch. It enlarges number of Android Application development and set their competency in android wear product. Personal Health segment (PHS) has scope on evolving quickly with wear devices. Your medical records can be shared to Centers of Disease Control (CDC), Health Insurance has quick estimate your individual fitness records and ease on policy Accountability. Another Major market on Home automation and security domain. You can monitor from wrist watch around you. Security IP cameras configured to watch, it allows the user to observe activity around a house or business right from a Wear device. The intercom system allows communication via a microphone and loud speaker between different places. This is open source Android project, cost of the Implementation is very low and effective for Target product development. As per the WDS (Wireless Device Strategies) research service says about the Automobile and controls. Connect2Car application has control and entertainment feature from wrist.

REFERENCES

- [1] <http://www.drdobbs.com/architecture-and-design/the-wearable-computing-revolution-begins/240169279>
- [2] <https://developer.android.com/design/wear/watchfaces.html>
- [3] <https://developer.android.com/training/wearables/apps/packaging.html>
- [4] <http://www.makeuseof.com/tag/what-operating-systems-do-wearable-devices-run-on/>
- [5] <http://www.androidpolice.com/2014/07/05/how-to-android-wear-enable-debugging-take-screenshots-unlock-the-bootloader-and-root-the-lg-g-watch/>
- [6] Android Wear Basic by “Pedro Lobito”
- [7] <http://www.wearabletechnology-news.com/news/2014/oct/01/deep-dive-getting-started-android-wear/>
- [8] <http://www.techradar.com/news/portable-devices/google-android-wear-what-you-need-to-know-1235025/3>
- [9] Android Wear Document Release 1.0 by “Micheal Hahn”
- [10] <http://observationdeck.io9.com/android-wear-review-the-os-itself-1637794213>

AUTHORS

First Author –Saminath, M.E, Accenture,Bangalore and s.x.veerapandian@accenture.com.