Implementation of Data Acquisition using Android USB Framework: A Survey

Yagnesh M. Shinde[1] Rajesh Sola[2]

[1,2]Department of Computer Engineering

[1]GTU PG School, Gujarat Technological University, Ahmedabad, India [2]Advance Computer Training School, CDAC, Pune, India.

Abstract— This paper presents a survey of communication of embedded system to the android device. One of the main principles of the Smartphone and tablet operating systems is simplicity and accessibility. Embedded system does not have that kind of features. By connecting these two types of devices user accessibility can be simplified. The goal of this paper is to create communication and provide the user interface to the embedded system with some the new facility which is not in the embedded system. Connection is established through the standard USB framework in both the side. Application is installed on the android device through users are presented with the reading of the data acquired from the model. Embedded system is connected through the USB with Android device. Android device works on the both accessory mode and host mode. This paper is focused on the methods of wired data transfer and possibilities derived from them.

Key words: Android; Embbded System;Data Acquiction; USB Communication;LibUSB; USB Driver Skeleton.

I. INTRODUCTION

Peripheral support of typical android gadget so adding more peripherals like temperature sensors, light sensor, gas pressure, fluid flow, force , voltage etc. Peripherals are called as Embedded systems or Embedded device. They can communicate that embedded system to the Open Sources Android Operating system. Here for the instance they use Android phone or tablet which has Android USB Host capabilities. Android device is connected to this supported USB device capabilities embedded hardware. Here Alternatives for USB is there like Bluetooth, WI-Fi, Near Field communication (NFC) and GPRS-GSM (3G/4G). Android device provide rich user interface for the embedded hardware controls and get the result using data analysis graph. In this survey Embedded system is Android Accessory Development Kit (ADK) or Accessory Host which is used Android open Accessory Protocol for communication. Google Standard and APIs used for USB communication to accessories for Android device. Embedded system hardware with USB host capabilities and capable of supplying a power of 5V@500mA to the android device. Simple USB commutations with two bulk end points. A Critical spec still unresolved is the possibility of acquiring eternal information such as data from the other sensor.

In addition to the Embedded Host, one more type of USB host is present on some peripheral hardware. The block diagram of this accessory is illustrated in Figure 1.It's a USB port that can flip between host or peripheral status on-the-fly—what the USB- Implementers Forum refers to as On-The-Go (OTG) functionality[5]. While the full specification allows for a port to be a USB host or USB peripheral through this capability, in practice OTG ports actually lock down the functionality to being a host or client, based upon the execution of protocols, for chips that are designed for OTG support (note: OTG support is not mandatory. Intent behind that particular selection is to prove that it is possible and inexpensive to design such a system. Both the design platforms are in wide use and have open source code can easily modify to a particular use.

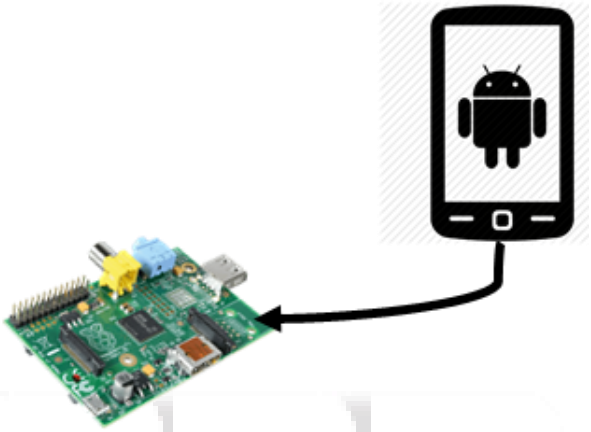

Fig. 1: Peripheral with Android Open Accessory Interface

A. Embedded System:

Embedded system [1] is application –oriented special computer system which is scalable on both the software and hardware. Embedded system control many devices in common use today. Android is the operating system that powers all android devices. Just like the other open source operating system power laptops and desktop computers. Android is also an open source technology based on the Linux open source platform. Accessory development kit (ADK) [7] is a physical accessory that can be attached to Android device. The communication is established between the ADK an android mobile using USB framework. These particular devices perform specific action .For USB accessories to be supported on particular device and it must be support to the accessory-mode. Android API used for communication and data transfer between devices and external peripherals.

B. Android:

Android platform since it has huge market and open source. Android [4] is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android OS is based on Linux. Android Applications are made in a Java-like language running on a virtual machine called 'Dalvik' created by Google. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. Accessory mode is a feature of Android OS since version 2.3.4 Gingerbread and 3.1 Honeycomb and above.

C. Android Open Accesroy Protocol:

Android Open Accessory [3] support allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special accessory mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts in the USB accessory role. Android Open Accessory Protocol allows detecting Android-powered devices that support accessory mode. Accessory mode is ultimately dependent on the device's hardware and not all devices support accessory mode.

Android Open Accessory support is included in Android 3.1 (API Level 12) and higher, and supported through an Add-On Library in Android 2.3.4 (API Level 10) and higher. Android 4.1 and higher has support for audio output over a USB connection or Bluetooth An Android USB accessory must adhere to Android Accessory Protocol [3], which defines how an accessory detects and sets up communication with an Android-powered device. In general, an accessory should carry out the following steps:

- Wait for and detect connected devices
- Determine the device's accessory mode support
- Attempt to start the device in accessory mode if needed
- Establish communication with the device if it supports the Android accessory protocol.

D. LibUSB:

LibUSB is a C library that gives applications easy access to USB devices on many different operating systems. It is an open source library that allows you to communicate with USB devices from user space [6].

LibUSB has all transfer supported like control, bulk, interrupt, isochronous. It has two transfer interfaces. Synchronous transfer is simple kind of transfer. Asynchronous transfer is more complicated but more powerful. LibUSB [7] provide Thread safe although the asynchronous interface means that we usually won't need to thread. It is lightweight with API compatible with Libusb-0.1[6] through the Libusb-compat-0.1 translation layer and transfers at a logical level.

At a logical level, USB transfers typically happen in two parts. For example, when reading data from an endpoint: A request for data is sent to the device sometime later, the incoming data is received by the host or when writing data to an endpoint: The data is sent to the device sometime later, the host receives acknowledgement from the device that the data has been transferred.

E. USB Driver Skeleton:

Once adk-aoa-skeleton module[21] is loaded and connected device got probed we can use appropriate sysfs opration form userspace for initalization of device in accessory mode. For the sake of easy sysfs operations are added to his module to initialize using simple shell commands without the need of c programm to use ioctl or any cross compolation.also we can pass the module parameter init_on_probe=1 to initialize device in AOA mode using default identity strings without the need for sysfs operations for initial testing [20].

F. Data Acquiction:

Data Acquisition [19] is the process of sampling that measure real world physical phenomenon or property and converting the resulting samples into digital numeric values that can be manipulated by a computer system .The component of Data Acquisition are sensors that covert physical parameters to electrical signals, Signal conditioning for converting the analog signals into the digital signals and Last need is controlled software system programs. DAQ hardware is connected to ports parallel, serial, USB, etc in the Embedded System. DAQ have device drives to the DAQ hardware to work with the PC or Mobile. A device driver performs low-level register writers and reads on the hardware.

II. LITERATURE SURVEY

As per our Survey currently there various systems uses different techniques to communication of embedded system. An application running on the Android platform initiates control of the accessory. This broadens the scope of the different scenarios in which Android platforms can be utilized.

Implementation of a system design using the Android Open Accessory Initiative will have advantages in a broad spectrum of industry sectors [16].

- In the medical sphere, it will enable data acquired from patients on their current physical condition (heart rate, blood glucose level, body temperature, blood pressure, etc.)To be transferred to a tablet PC for subsequent analysis [14].
- Likewise within a sporting environment it will be possible to download workout data (calories burnt, distance run/cycled/rowed, heart rate, etc.) from gym equipment to a Smartphone. The acquired data could then be compared with data from earlier workout sessions [13].
- For the household appliance market it could have notable benefits, as through it engineers could upload software upgrades to white goods, or conversely download diagnostic information during maintenance checks [12].
- In home/building automation systems [11], it adds a new dimension to how interfacing with thermostats, sprinkler systems, security alarms and audio-visual systems is conducted so that more intuitive Smartphone- or tablet-based applications with vibrant screens and touch capabilities can be used to configure the equipment rather than the clunky user interfaces currently found on these items.

III. METHODOLOGY

This survey is structured according to the key approaches on the Implementation of Embedded system communication in various fields. While buttons, LED(s) and characters LCD(s) were sufficient of the past, future embedded system will be more intelligent, configurable and connected its requiring the new approach to the user interface. There are better and more cost effective ways to provide the embedded system with an easily programmable and controller user interface. These approaches have wired and wireless with embedded

system engineers can use Android enabled devices in design.

Wired transport usually relies on a serial transport using common bus in form of a cable. Wireless transport relies on using public frequencies of GSM broadband connection. Most popular wireless connections are Wi-Fi, Bluetooth, Near Field Communication (NFC), Radio Frequency Identification (RFID) and ZigBee. Comparisons between Wired and Wireless in shown in Table 1.

Characteristic	**USB**	**Bluetooth**	**802.11(Wi-Fi)**
Security	Maximum	Minimum	Reduced
Bandwidth	480 Mb/s(2.0)	2.1 Mb/s (2.0)	802. 11 b Mb/s 802.11 g 54 Mb/s 802.11 n 600 Mb/s
Cost	Low	Relatively Low	High
Range	5 m 30m- 5 hubs	10 m	As Access Point:20m Wi-Fi Direct-200m

Table 1: Connectivity Comparisons

A. Android USB Host Mode:

One other possibility for Android devices with a USB host port is to the use Android USB Host Mode [3]. This is different than the traditional operating system approach of loading a device driver and writing an application to communicate with the device drive. With USB Host Mode, an Android application accesses the USB bus and directly Communicates with a USB device. There is no device driver loaded at the protected operating system level. Show the Fig.2 Android device in Host mode and the embedded device in acts as accessory mode. The Android device may provide power to the USB device.

B. Android USB Accessory Mode:

The underlying Linux kernel on Android provides USB host supports. USB accessory mode allows users to Connect USB host hardware specifically designed for Android-powered devices. The accessories must adhere to the Android Accessory Protocol. When an Android-powered device is in USB accessory mode, the attached Android USB accessory [3] acts as the host to provide power to the USB bus and enumerates connected devices. Android 3.1 (API level 12) supports USB accessory mode.

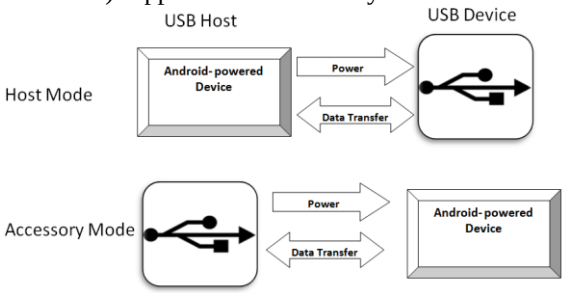

Fig. 2: Host mode and Accessory mode

C. Communicating With A Device:

Communication with a USB device [3] can be either synchronous or asynchronous. Fig 3 illustrates the flow of Communication in both the side. First when user connect USB devices discover connected USB device by using intent filter and ask user permission to connect to the USB device. In either case crate a new thread on which to carry out all data transmission to properly set up communication with a device to obtain the appropriate USB Interface and USB Endpoint of that want to communicate on and send request on these endpoints with an USB DeviceConnection. Check an USB Device object's attributes such as product ID, vendor ID or device class [18] to figure out whether communicate with the device. Supply the data to transmit on the endpoint with the balkTransfer() or controlTransfer() method. To send data asynchronously the usbRequest class to initialize and queue as asynchronous request then wait for the result with requestWait().When done communicating with a device close the USB Interface and USB Device Connection.

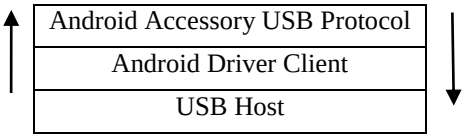

Fig. 3: Communication Flow

- Send Control request 51(0x33) to get the protocol version its returns 1 for AOA 1.0 and 2 for AOA 2.0.
- Send identity strings through control request 52(0x34) essentially one should send manufacturer, model, version to communicate embedded system to Android device.
- Start in accessory mode through control request 53(0x35) for launch an application matched by essential identity strings or start without application for audio/hid support.
- Send control required 0x58 for audio support with value=1 for 2 channel 16 bit PCM @ 444100 KHz.
- Embedded device acts as HID event source its uses four request code, 54 for Register HID,55 Unregister HID,56 Set HID report, 57 Send HID Event.

IV. CONCLUSIONS AND POSSIBLE FUTURE WORK

A lot of the current and future embedded systems can be customized and adapted to be controlled or monitored form the Smartphone's and tablets. This allows users an easy and well structured interaction with the embedded devices in industry, infrastructure and home. System consists of embedded device program and mobile application. Here connections with the Data Acquisition system with embedded system maintain. Connection can be realized by direct wired connection so it will be fast endpoint transfer mode.

Some future improvements can be made to the function and performance of the Data Acquisition system. Wi-Fi can be used if long distance controls or monitor is required. With the few Upgrades the whole system can become modular so it can be used with different sensor communication method. Embedded data acquisition can easily be adapted to work with other mobile platforms.

REFERENCES

[1] Diab, M.O.; Marak, R.A.; Dichari, M.; Moslem, B., "The smartphone accessory heart rate monitor," *Computer Medical Applications (ICCMA), 2013 International Conference on* , vol., no., pp.1,5, 20-22 Jan.2013 doi: 10.1109/ICCMA.2013.6506185
[2] De Dominicis, C.M.; Depari, A.; Flammini, A.; Rinaldi, S.; Sisinni, E.; Vezzoli, A., "Acquisition and elaboration of cardiac signal in android Smartphone devices," Sensors Applications Symposium (SAS), 2014 IEEE , vol., no., pp.83,88, 18-20 Feb. 2014 doi: 10.1109/SAS.2014.6798922.
[3] http://developer.android.com/guide/topics/connectivity/usb/index.html(Sep. 2014)
[4] http://www.computer-solutions.co.uk/info/Embedded_tutorials/usb_tutorial.htm(Sep. 2014)
[5] http://en.wikipedia.org/wiki/USB_On-The-Go (Oct. 2014)
[6] http://libusb.sourceforge.net/api-1.0/ (Oct. 2014)
[7] Deepali Javale,Mohd. Mohsin,Shreerang Nandanwar,MayurShingate,"Home Automation and Security System Using Android ADK, International Journal of Electronics Communication and Computer Technology (IJECCT) Volume 3 Issue 2 (March 2013)
[8] Zhan Wei Siew; Chen How Wong; Shee Eng Tan; Hou Pin Yoong; Teo, K.T.K., "Design and development of a tablet based real time wireless data logger," Global High Tech Congress on Electronics (GHTCE), 2012 IEEE , vol., no., pp.111,116, 18-20 Nov. 2012
[9] Mohamad O. Diab, Reem Abou Marak/ Brome, Mustapha Dichari, and Bassam Moslem,"The Smartphone accessory heart rate monitor", Computer Medical Applications (ICCMA), 2013 International Conference on , vol., no., pp.1,5, 20-22 Jan. 2013
[10] M. Kozlovszky, Bartalis L., B. Jókai, J. Ferenczi, P. Bogdanov, Zs. Meixner, L. Németh, K. Karóczkai, "Personal health monitoring with Android based mobile devices", MIPRO 2013, May 20-24, 2013, Opatija, Croatia.
[11] Rajeev Piyare "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone", International Journal of Internet of Things 2013, 2(1): 5-11
[12] Bruno Vunderl, Martin Žagar and Danko Basch, Remote control of model vehicles using Android mobile devices, ", MIPRO 2013, May 20-24, 2013, Opatija, Croatia.
[13] Nicolae, M.; Lucaci, L.; Moise, I., "Embedding Android devices in automation systems," Design and Technology in Electronic Packaging (SIITME), 2013 IEEE 19th International Symposium for , vol., no., pp.215,218, 24-27 Oct. 2013.
[14] Drumea, A., "Control of industrial systems using Android-based devices," Electronics Technology (ISSE), 2013 36th International Spring Seminar on , vol., no., pp.405,408, 8-12 May 2013.
[15] LIN Zhigui, MENG Dejun, ZHONG Qingqing,Design and Implementation of OTG communication system terminal Based on USB.
[16] "Extending the Scope of Android Platforms via USB connectivity," By Gordon Lunn, FTDI Chip ,www.eecatalog.com/usb
[17] Unboxing Android USBA hands on approach with real world examples , By Rajaram Regupathy, ISBN13: 978-1-4302-6208-4.
[18] Leveraging the Android Accessory Protocol, Slides of Gary Bisson Adeneo Embedded, Android Builders Summit 2013.
[19] Shajahan, A.H.; Anand, A., "Data acquisition and control using Arduino-Android platform: Smart plug," Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on , vol., no.,pp.241,244, 10-12 April 2013 doi: 10.1109/ICEETS.2013.6533389
[20] "TURN YOUR LINUX COMPUTER INTO A HUGE ANDROID USB ACCESSORY" , www.android.serverbox.ch/?p=262 (Oct 2014)
[21] Greg Kroah-Hartman,"How to Write a Linux USB Device Driver" *Linux Journal*.(Nov 2014)

