Monitoring and Controlling Embedded Device Using Android Framework

Mr. Dhruvilkumar Pankajbhai Patel
VLSI & Embedded Systems Design
Gujarat Technological University PG School
Ahmedabad, India
pateldhruvil123@gmail.com

Abstract— This paper presents additional support to embedded device and android phone. Today we are living in a smart world where controlling and monitoring is playing an important role in human life more tasks are carried out with the help of smart phones.. Computer control systems for controlling and monitoring need good processing power and Memory resources, standard networking options and enough Input /Output lines for connecting to Sensors, Gpio's, different bus's. The explosive advancement of mobile devices running on the Android operating system brought us minimal effort devices that satisfy every one of these requirements for mechanical control, missing just input/output lines.mobile phone will connect to any device without rooting just connect the device using simple USB cable and mobile is going into accessory mode and we access device which are connected with Smartphone.

Keywords*-Android; Embedded Board; AOA; USB Device Driver; VCAN; GPIO;*

*****

I. INTRODUCTION

Today people don't utilize their cell phone for talking as much as they use to, yet they are progressively utilizing them to Controlling and monitoring. Android is the first Open source programming toolkit for mobile environment. The quantity of portable running Google's Android has risen forcefully having 76% of the worldwide market in cell phones, It is clear why choosing Android as a development platform.

In this paper using AOA(Android Open Accessory) protocol make the communication between android device and external embedded boards.

The Android platform (from 3.1 onwards) has coordinated a specific protocol "Android Open Accessory", which permits external embedded hardware (accessory) to communicate with an Android-powered device (Android phone or tablet) in an special accessory mode over USB or Bluetooth interface. Android accessory are particularly intended to connect to Android-powered device over USB or Bluetooth and hold fast to a straightforward convention (Android accessory protocol) that permits them to identify Android-powered device that support accessory mode. With a dominant part of advanced mobile phones and tablets utilizing the Android today, an Android open extra standard gives an essential connection between phones, tablets and other Android powered device and embedded system which are pervasive in our day by day life.[1]

Accessories can be audio docking stations, activity machines, individual medicinal testing gadgets, climate stations,wearable gadgets or any on the other hand, any other external devices that add to the usefulness of Android. Accessories must also provide 500mA at 5V for charging power. Many previously released Android-powered devices are only capable of acting as a USB device and cannot initiate connections with external USB devices.

AOA (Android Open Accessory) support overcomes this limitation and allows you to build accessories that can interact with an assortment of Android-powered devices by allowing the accessory to initiate the connection. [1]

The Android Open Accessory (AOA) protocol allows simple USB communication with two bulk end points between two nodes, which are:

- An external hardware device with USB host capabilities and peripheral support known as an Accessory Development Kit (ADK) or Accessory Host, from now onwards.
- An Android gadget, which works in device mode.[2]

Beagle Board XM runs the Android accessory protocol (generic) and extra accessory specific programming. We utilize Linux 3.14 custom kernel for Beagle board and have developed and integrated the android accessory protocol and sample application. We have additionally built up application (.apk) for Android that runs on the phone or tablet that connects to the Beagle board XM application to control the peripherals on Beagle Board XM.

Functionally, Beagle Board XM is connected with Android powered (phone / tablet) device over USB, the Beagle Board operates USB in host mode (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. The beagle board runs the Android USB accessory protocol with the application, detects & enumerates the Android-powered device (phone/tablet) and starts the sample application (beagle board side) that waits for commands from the Android device (running sample accessory apk) to control/monitor the Beagle board peripherals.

II. Overview Of System

A. Android

Android stage since it has huge market and open source. Android is a product stack for cell phones that incorporates a working framework, middleware and key applications. The Android OS is in view of Linux. Android Applications are made in a Java-like dialect running on a virtual machine called "Dalvik" made by Google. The Android SDK gives the devices and APIs important to start creating applications on the Android stage utilizing the Java programming dialect. Embellishment mode is a highlight of Android OS since form 2.3.4 Gingerbread and 3.1 Honeycomb or more.

B Usb Host and Usb Accessory

There are two methods of usb: USB accessory and USB host. In USB accessory mode, the external USB equipment go about as the USB hosts. This gives Android powered gadgets that don't have host capacities the capacity to associate with USB hardware. Android USB accessory must be intended to work with Android-powered gadgets and must hold fast to the Android accessory communication protocol. In USB host mode, the Android-Powered device goes as the host. Examples of gadgets incorporate computerized cams, consoles, mice, and game controllers. USB gadgets that are intended for an extensive variety of utilizations and situations can in any case communicate with Android applications that can effectively speak with the gadget. Figure 1 shows the differences between the two Usb modes, Host and device. When the Android-powered device is in host mode(it has power) it acts as the USB host and powers the bus. When the Android-powered device is in USB accessory mode, (it takes power form host device) the connected USB hardware (an Android USB accessory in this case) acts as the host and powers the bus.[4]

C Android open accessory protocol working

Once the Android gadget is joined with the ADK host, the gadget will list with its unique arrangement, which may shift for every gadget. At first, the gadget may be joined in the camera mode (PTP backing) or media gadget (MTP mode) mode to the host. Presently, ADK instate to the gadget to transform into the accessory mode.[2]

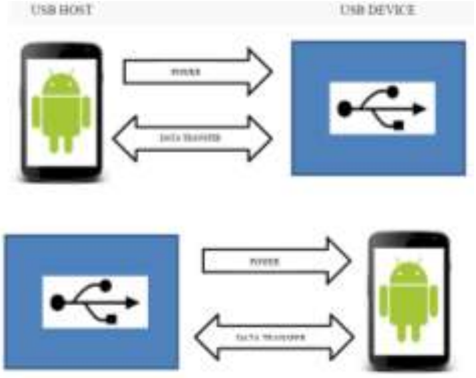


Figure 1 USB host and Accessory mode

This could be possible in the accompanying couple of ventures by sending different vendor (Google) particular control asks for through the control endpoint whose location is zero.an accessory should carry out the following steps:[2]

- Wait for and detect connected devices
- Determine the device 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

III. Proposed System

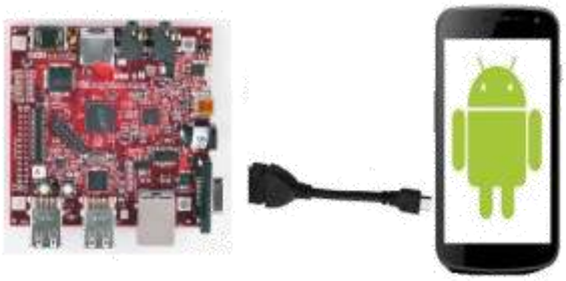


Figure 2 Block diagram of system

Figure 2 presents the proposed system architecture. Here Beagle board XM is connected to android device using a USB cable. Beagle board XM is in host mode and android devices in accessory mode.We can use simply with/without rooting any android Device.

A. Beagle board XM as A HOST:-

- Communication with android device.
- Control adk hardware like can bus and gpio.
-

As shown in figure , first of all driver for usb-aoa data transfer is made , that driver will install in linux system and using user space application we can access various file operation,such as open,read,write,close.

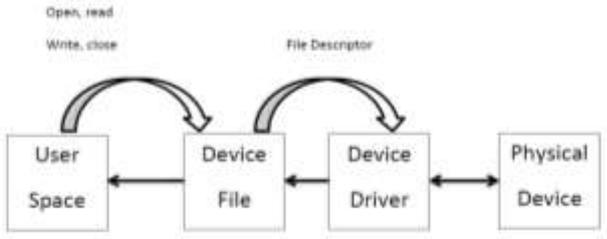


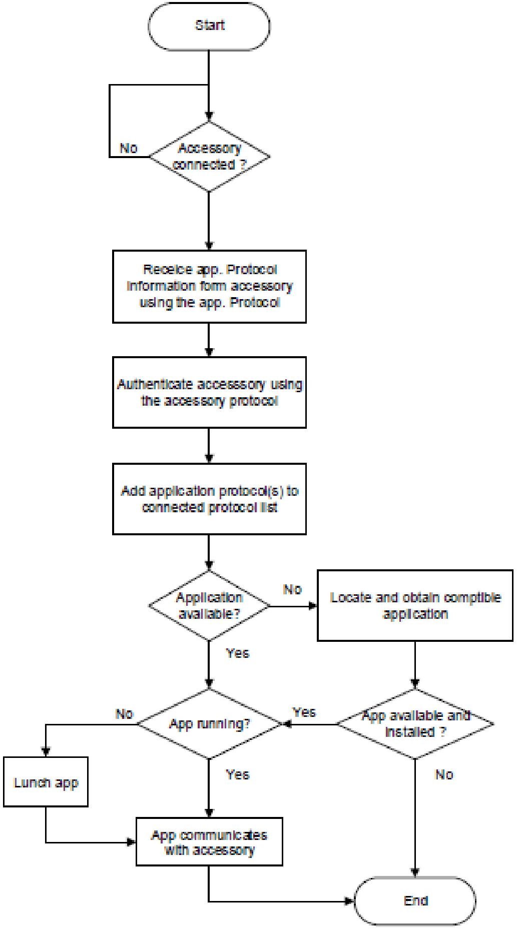
Figure:3 -Device communicates

B. *Android Device as Accessory:-*

On connection of Device with Embedded Module the following process is occurring:-

- Android Device is activated and application URL is available on the screen.
- The Device also handles communication with Beagleboard XM.
- GUI is given for control and access input /output data to Embedded Module.

As shown in Flow chart first it will check if any accessory is connected or not connected, if an accessory is connected then check application according to accessory is installed or not installed if not then go to the URL and download the application form net and launch the application.[6]


Start
Accessory connected ?
No
Receice app. Protocol information form accessory using the app. Protocol
Authenticate accesssory using the accessory protocol
Add application protocol(s) to connected protocol list
Application available?
No
Locate and obtain comptible application
Yes
App running?
No
Yes
App available and installed ?
Lunch app
Yes
No
App communicates with accessory
End

Flow chart of Android application

C. *Software Support*

Virtual CAN(VCAN) bus is used for communication. Can device driver and AOA userspace code is used combindially to transfer frames over the VCAN bus.By now PC as Embedded Device and Android mobile is used as components for communication in both way PC to Android phone and vice versa. After successfully communicated PC and Android Device, GPIO of Beagle Board XM is contolled by android application.

IV. Testing And Results

Device Driver is developed for providing ADK environment to Linux machine, that can write user space apps on top of this driver or customize the skeleton code for specific needs Device driver *adk-aoa-skeleton.ko* is loaded using insmod or modprobe. After that we can send manufacture, model and version strings to specific application. After successful reception of three strings mentioned above on device specific application is popped up. If the application is not installed on device that the URL for the same will poped up using that the app can be downloaded. Once app is initialised, you can communicate with the app using simple file operations like read and write from userspace with the help of the device file *(/dev/aoa-skel0)* created by this driver.[6]

A. *VCan bus interface*

CAN is a situated of open source CAN drivers and a systems administration stacks contributed by Volkswagen Research to the Linux kernel. Once known as Low Level CAN Framework. Customary CAN drivers for Linux are in light of the model of character Devices. Regularly they just permit sending to and accepting from the CAN controller.here for general purpose communication I use VCAN, which has given same functionality like can bus.

Like show in figure 4 the full can bus frame is divided In different part like first 1 bit is start of bit and after that 12 bit for identification for the can bus and 6 bit for the control bit and next 8 bytes for data.and next 16 bit for CRC check and next 2 bit for Acknowledgment and last 7 bit for EoF.

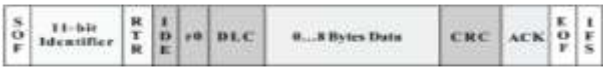


Figure 4 CAN Frames

For this I made Android application and user space code which transfer data using AOA protocol. So the data which we send from mobile application that show in the user space terminal as well as also shown in VCAN terminal. The same data is also transferred from terminal to Android Phone using the VCAN bus.

As shown in figure 5, the android application for can bus have 3 fields to set. We provide canId filed in which 1 bytes trasfer as canId and we set DLC as dropdown lists ,and we have 8 blocks for data of the frames.

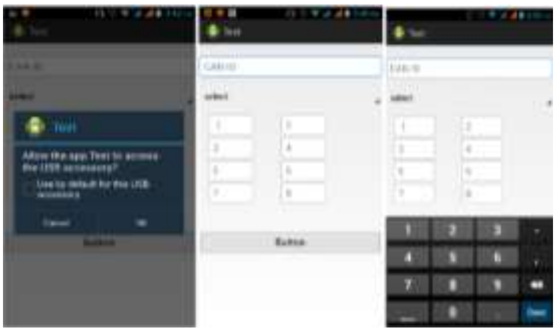


Figure 5 Android application for can bus

As shown in figure 6 screen short of the user space output is presented. It shows the data transfer from the Android app to the aoa-drive.

Figure 6 System side output

Figure 7 is screen short of the candump side output. In that data transfer from the Android app to the can bus is shown and we get the same frame in CAN bus.

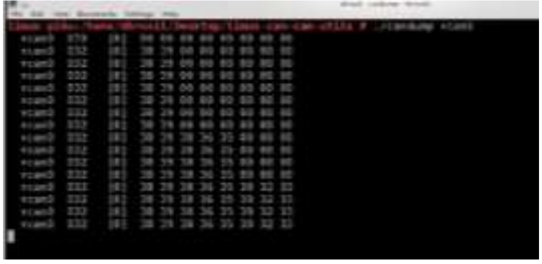


Figure 7 Can bus side output

B. *Gpio interface*

In this we made user space code and gpio driver, a driver which is installed on Beagle Board, we can on-off the led according to switch pressed from android phone.

As shown in figure 8 android application has 4 different button on android device for on-off Led on Beagle board side.

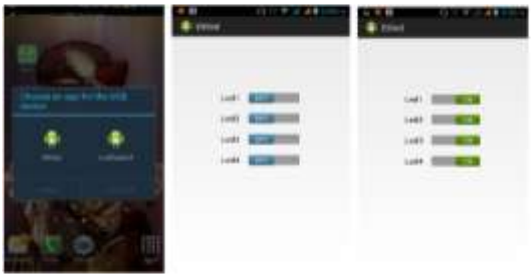


Figure 8Android application for Gpio

Figure 9 shows the terminal output at BeagleBoard XM side. According to the button on-off ,change in the status of LED on Beagle Board XM side is displayed on terminal.

Figure 9 Gpio access output

C. *Watchdog timer*

In that we can also manually set or reset software timer using the application, basically if we pressed any button the timer is set and if we again pressed the same switch then it will overwrite and after that time board will be reset.

V. CONCLUSION

As mentioned in this paper, a driver is developed that communicates between embedded device and android phone. And we can access different peripheral device or on-board GPIOs. Virtual CAN bus, GPIO and watchdog time is used for controlling and monitoring

VI. ACKNOWLEDGMENT

The author would like to express his sincere appreciation to his advisor Mr. Rajesh Sola and Mr. Aditya Kumar Sinha for enthusiastic guidance and advice throughout this research.

VII. REFERENCES

[1] “AOA protocol “:- available here https://source.Android .com /accessories/ protocol.html

[2] “ accessory “:- available here https://source.android.com /accessories /aoa.html

[3] Lipeng Fan, L. D. (2013). ―Design of Wireless Optical Access System using LED Based Android Mobile‖. IEEE

[4] Unboxing Android USBA hands on approach with real world examples, By Rajaram Regupathy, ISBN13: 978-1-4302-6208-4

[5] Rajesh Sola,Oct (2014). Android Open Accessory Protocol Turn Your Linux machine as ADK. Opensourse for you

[6] Holden, P. and Bolton, L. and Ganatra, N. and Adler, M. and Schubert, E.C. and Dorogusker, J.L. and Toprani, S. and Ananny, J. Connection to multiple accessories with multiple accessory-specific protocols, US Patent 8,554,924, 2013