Design and Development of an Android Accessory

F. C. Serce
Information Systems Engineering, Atilim University, Kızılcaşar Mahallesi, 06836 Incek Golbasi, Ankara, Turkey
Email: cemileserce@gmail.com

A. Erpay
Arlentus Control, Ankara, Turkey
Email: ahmetr@gmail.com

A. Kara
Electrical and Electronics Engineering, Atilim University, Ankara, Turkey
Email: akara@atilim.edu.tr

B. Unal
Department of Architecture, Atilim University, Ankara, Turkey
Email: benbulentunal@gmail.com

D. Ozgubar
Industrial Product Design, Atilim University, Ankara, Turkey
Email: dogancanozgubar@gmail.com

A. Ozturk
Computer Engineering, Atilim University, Ankara, Turkey
Email: ozturkaburak@gmail.com

S. Sekerci
Software Engineering, Atilim University, Ankara, Turkey
Email: sekerci.asinan@student.atilim.edu.tr

***Abstract*—During Google I/O 2011, Google released the Android Open Accessory standard and launched Android Open Accessory Development Kit based on Arduino as a reference implementation for hardware manufacturers and as a starting point for building accessories for Android. This technology allows Android applications to communicate with real world devices such as keyboards, robots, exercise machines, cars and more. It has the potential to change the way we use and interact with our smartphones and tablets. In this study, an Android accessory has been designed, developed and manufactured. The paper covers the details of the development phases and experiences are shared at the end.**

***Index Terms*—android open accessory, embedded programming, communication networks**

I. Introduction

Android is the most widely used, fastest growing operating system for mobile devices. Android Open Accessory (AOA) is a new framework announced at Google I/O 2011 conference for further strengthening the Android devices by building accessories. The framework includes the protocol for developing Android accessories and a reference hardware and software implementation, called Accessory Development Kit (ADK). The protocol was based on the control of the communication between two USB devices. Android accessories can be audio docking stations, exercise machines, personal medical testing devices, weather stations, or any other external hardware device that adds to the functionality of Android. For example, imagine that your thermostat is AOA enabled. You could monitor the current temperature in your home and make adjustments to your settings anywhere you happen to have your device. Another example; imagine an AOA enabled stationary bike. An Android phone can be plugged in to the bike that initiates a game to assist to make the exercise process more fun and interactive. This was a real example from Google I/O 2011 event.

There are several projects working on creative solutions to build accessories for Android devices. IOIO

Manuscript received January 10, 2014; revised October 4, 2014.

doi: 10.12720/ijeee.3.5.390-395

is one of the successful one. It is a specialized hardware for Android. It is an input/output board for Android phones and tablets. There is no need for an external programmer. Everything connected to this board can be controlled from within Android apps [1]. The whole project, both hardware and software is open source. It manages connection with Android device using TCP socket and the Android Debug Bridge (ADB). Although ADB works well on all Android devices, it is not an ideal solution for security issues; enabling debugging mode opens the gates for all kinds of malware.

Several companies are already producing Android Open Accessory-compatible development board kits for a variety of purposes, including RT Corp, Microchip, DIY Drones, and more. However, there have been attempts to design application specific embedded devices to perform different functions using Android devices. In one of the recent works [2], an electronic device capable of communicating with an Android hand held device (or a PC) using USB has been designed. The device is appropriate for dual use of USB as it uses a PIC24FJ family microcontroller. The objective is to collect sensor data and pass them to remote locations, for further processing, via an Android device or a networked PC. In [3], use of Android's USB interface in controlling of industrial systems has been demonstrated. The design achieves lower latencies by using an USB chip before the microcontroller (16 bit old one). An interesting work on indoor pedestrian localization has been reported in [4] where a differential pressure-based MEMS anemometer is integrated to an Android smartphone by means of a dedicated 32 bits microcontroller. Here, computational power of PIC 32 has been used for signal processing, data fusion and position estimation processes.

This paper presents results of an Android based desktop accessory development work. The desktop accessory has been developed to keep track of missed calls, new SMS messages, weather condition, Internet connection and battery level of the Android device and displays the current state via LEDs and LCD panel on the accessory. It also allows user to send automatic messages to contacts linked to missed calls. The project includes the design and development of the board, software application, and industrial product design of the accessory considering the aesthetics, ergonomics, and usability principles. This study is a product of a multidisciplinary project having members from diverse disciplines such as Computer Engineering, Electric Electronics Engineering, Software Engineering and Industrial Design.

II. Android Accessory Development

A. Android Open Accessory

Android Open Accessory (AOA) [1] protocol defines how an accessory detects and sets up communication with an Android-powered device. According to the protocol, an accessory should carry out the following steps for establishing communication with the device:

- 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

Android Open Accessory support is included in Android 3.1 (API Level 12) and higher. It is also back ported to Android 2.3.4 (API Level 10) and higher using an add-on library.

There are two AOA protocol releases. In the second release (AOA 2.0) two new features are added: audio output (from the Android device to the accessory) and support for the accessory acting as one or more Human Interface Devices (HID) to the Android device. The Version 2.0 is available in Android 4.1 and it supports Bluetooth for communication.

B. Accessory Development Kit

Google launches Accessory Development Kit (ADK) as a reference implementation of AOA protocol to make the process of developing accessories easier. Each ADK release is provided with source code and hardware specifications. There are two versions available. The components of each release are summarized below [1], [2]:

- ADK 2011
  - A USB micro-controller board that is based on the Arduino Mega2560.
  - An Android Demo Shield (ADK shield) that affixes atop the ADK board implements the input and output points on the board
  - A library based on the Arduino USB Host Shield library provides the logic for the USB micro-controller board to act as a USB Host
  - An Arduino sketch, defines the firmware that runs on the ADK board and is written in C++
  - The Android accessory protocol library
  - Other third-party libraries such as CapSense library, I2C/TWI (Two-Wire Interface) library, Servo librar, Spi library, and Wire library.
- ADK 2012
  - A USB micro-controller board that is based on the Arduino Due Board
  - An ARM 32-bit Cortex M3 micro-processor
  - Separate USB connections for an Android device and computer connection for programming and debugging
  - Sensors for light, color, proximity, temperature, humidity, barometric pressure, and acceleration
  - Micro SD Card slot
  - Bluetooth support

C. Connecting over USB

AOA allows external USB hardware (an Android accessory) to interact with an Android-powered device in a special accessory mode. When an Android device is in accessory mode, the connected accessory acts as the USB host and the device acts in the USB client. USB host powers the bus and enumerates devices. The accessory needs to implement a simple handshake to establish a bi-directional connection with an application running on the Android device. After the low-level USB connection is

negotiated between the Android device and the accessory, control is handed over to an Android application. Any Android application can register to handle communication with any USB accessory [1], [5].

USB is an asymmetric protocol in that one participant acts as a USB Host and all other participants are USB Devices [1]. For example, a laptop acts as USB Host and a printer, mouse, webcam, etc., is the USB Device. The USB Host has two important functions. One is to be the bus master and control which device sends data at what times. The second is to provide power, since USB is a powered bus. Relatively few Android-powered devices are capable of acting as a USB Host and can initiate communication with external USB devices. Having USB Host capability puts extra cost to Android device. So many previously released Android-powered devices are only capable of acting as a USB device. AOA protocol overcomes this limitation and allows everyone to build accessories that can interact with any Android-powered devices by allowing the accessory to initiate the connection. The Android device acts as the USB Device and the accessory acts as the USB Host. As a bus master, the accessory is required to provide 500mA at 5V for charging power.

III. Development of an Android Desktop Accessory

In this project, a desktop accessory has been designed, developed and manufactured as a sample implementation of AOA. The accessory can be used by anyone at home or at office. It works like a docking station. When you are at home or at office, you plug your Android device into the accessory. By looking at the LEDs and LCD panel and by using the buttons on the accessory, you can track your missed calls, incoming messages, weather, battery level etc. without even touching the device. The functionality of each component is given below:

- One LED notifies new incoming SMS
- One LED notifies missed call
- One LED displays the Internet connection status
- One RGB LED displays the battery level using three different colors
- Four LEDs are used to display the current weather status (cloudy, sunny, rainy, snowy)
- The LCD panel is used to show the number of missed calls and number of new incoming SMS messages
- One button sends automatic SMS message for each missed call
- One button resets the accessory

The development of the accessory is divided into modules: software development, hardware development, and industrial design and manufacturing. Each module has a separate and distinct process and requires different domain knowledge. This is a multi-disciplinary work having members from the departments Information Systems Engineering, Computer Engineering, Software Engineering, Electrical and Electronics Engineering and Industrial Design. The study is granted by Atilim University as an undergraduate research project.

In the next sections, each module of the project is explained in detail.

A. Software Development

In this module, the requirement analysis, architectural design and development of the Android application and the communication protocol between the Android device and the accessory have been done. The architectural design of the system is summarized in Fig. 1.

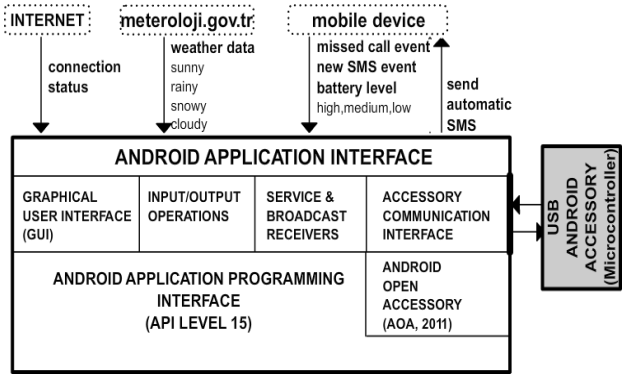


Figure 1. The architecture of android accessory

In this project, Android API Level 15 and Android Open Accessory Protocol Version 1.0 (2011) have been used. The software development module includes the following tasks:

- Development of the graphical user interface of the Android application,
- Creating input/output operations to establish USB communication and to read/write through USB port,
- Implementation of background services and broadcast receivers to listen certain events, and
- Development of the communication interface between Android application and the accessory.

When a device is connected to the accessory, it can already be in accessory mode, support accessory mode and is not in that mode, or does not support accessory mode. The Android application checks for these cases and responds accordingly. It checks the vendor and product ID of the device descriptor. A device in accessory mode has a vendor ID of 0x18D1 and a product ID of 0x2D00 or 0x2D01 [1]. Unfortunately, not all devices support accessory mode, and the only way to check its availability is to test it with the accessory. If the device is detected as being in accessory mode, the accessory tries to find the proper bulk endpoints and set up communication with the device. If there are endpoints found, they are appropriately set up for communication and notify that the device is properly set up to communicate with the USB accessory. During this handshaking process, the accessory sends its attributes such as model, manufacturer, and version to the Android-powered device. Then the device looks for its activities having matching configuration. Using AOA, the activity can define a filter for accessories. This filter is an xml file defining the attributes of the accessory. If the device finds a match, then it opens this activity automatically. If there are more than one activity, then it prompts user to select

one. For example, assume that the USB accessory has the following attributes:

- Model: M1
- Manufacturer=Atilim
- Version 1.0

Then, the corresponding filter file defined the Android application will be as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<usb-accessory model="M1" manufacturer="Atilim"
    version="1.0"/>
</resources>
```

Once the accessory is connected to the device, the user will be presented with a dialog asking what application should be opened, after the Android system has signaled that an accessory is available by issuing an Intent using the filter above. Moreover, the accessory can also specify a URL to present to the user if no application is found which knows how communicate with it. It may for example, be the URL for an application in Android Market that is specifically designed for use with the accessory.

According to the requirement analysis, the task is divided into two: one runs in foreground and one runs in background. Unless it is specified otherwise, most of the applications run in foreground. However, this causes problems because long-running operations will interfere with the responsiveness of your user interface. This annoys users, and can even cause system errors [6], [7]. To overcome this problem, Android platform supports executions of operations on separate threads in the background. In this project, majority of the operations requires event listening in the background such as:

- New incoming SMS event
- Missed call event
- Battery level update event
- Internet connection update event
- Weather condition update event: the weather condition is retrieved automatically from a website of a national meteorology web site.
- New signal from accessory event: This event occurs when the user presses the button on the accessory. Then the accessory sends a specific signal to the device to initiate message sending to contacts linked to missed calls.

The application listens for these events and notifies the application and the accessory if there is any update.

While the hardware module has been developing, the software team tests the software application with Arduino Mega ADK.

B. Hardware Development

The hardware requirements of the accessory are determined. The communication protocol between the accessory and the microcontroller has been programmed.

The circuit is built with the electronic component providing USB communication.

A firmware establishing communication with Android device is prepared based on Microchip's samples [8].

Android smartphone basically uses Android Open Accessory protocol in USB communicating with a host device. In the project, AOA Version 1.0 is used. The main component of the accessory was the microcontroller. In this work, a PIC24F series microcontroller is identified to be most appropriate microcontroller for USB host operation on the accessory side. Then, it performs USB communications with the smartphone, and management of all targeted functionalities on the electronic modules. There are several electronic modules in order to position electronic circuitry easily into the accessory (Fig. 2, Fig. 3). Electronic modules have necessary circuitry components for performing all functionalities listed previously.

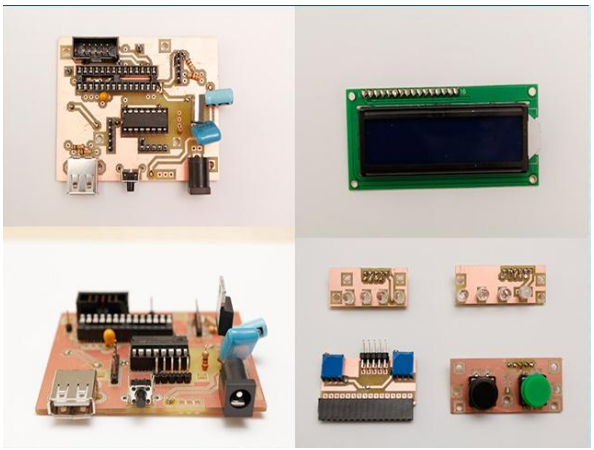


Figure 2. Hardware components of initial design

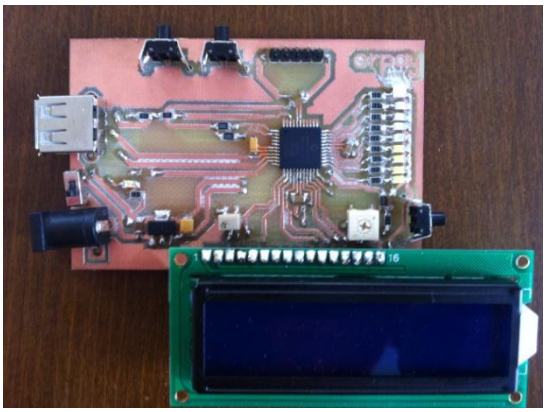


Figure 3. Surface mount device (SMD) design of the main board.

C. Industrial Design and Manufacturing

The project was a real experience of design, development and manufacturing of an industrial product prototype with a multidisciplinary team. Before starting the design process, the requirements and technical specifications were very critical for possible constraints to be applied in industrial design of the accessory. Moreover, aesthetics, ergonomics, functionality, and/or usability of the accessory all should be considered in the industrial design process. For example, posture of the accessory on the desktop (being static or dynamic as visually), ergonomic accuracy, physical stability when a smartphone is placed on it, legibility of the screen and the

buttons, the technical constraints like ventilation and manufacturing details and marketability are taken into consideration in the industrial design process.

Next, the first design is turned into three dimensional form by modeling it with 3DS Max, Rhino programs (Fig. 4). It is visualized with the use of VRay (light, camera, color and texture). The material to be used in the design was determined by considering texture, color and form.

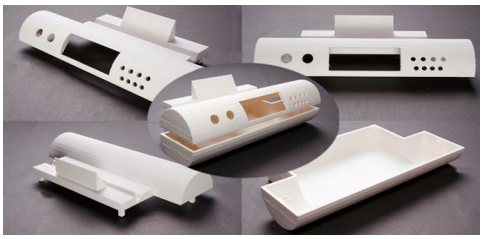


Figure 4. Sample visual design of the accessory

The prototype is manufactured in the laboratories of the department of Mechatronics Engineering at Atilim University. The manufacturing device is Stratasys, Fortus 360 mc and ABS plastic was used as the material. After manufacturing, the circuit and its peripherals are placed into the case to finalize the accessory. Placement of main electronic module in the first prototype is provided in Fig. 5. Then integration tests have been performed (Fig. 6). Samsung Galaxy S II was used as the testing Android device. After the components are installed inside the product, the shortcomings and necessary renewals are figured out. Consequently, the product is redesigned and the final model is produced.

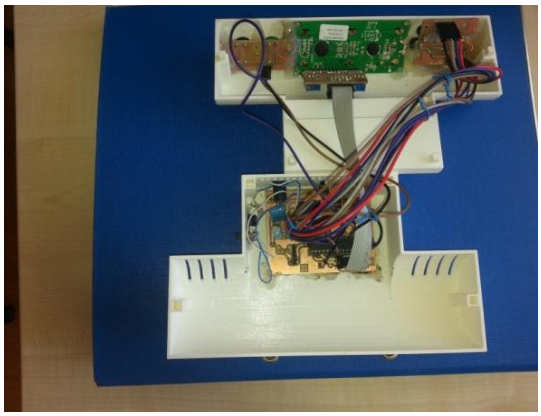


Figure 5. Placement of main electronic module in the first prototype

Figure 6. Sample images from testing of final design

IV. Conclusion

The Android Open Accessory (AOA) protocol and Android Development Kit (ADK) allow external USB hardware to interact with an Android-powered device in an accessory mode. When an Android-powered powered device is in accessory mode, the connected accessory acts as the USB host and Android-powered device acts as the device. This model has also the benefit that the phone/tablet can charge from the accessory. ADK is completely open meaning you are free to design what you like.

In this study, design, development and manufacturing of a desktop Android accessory has been experienced.

- Android Open Accessory support is included in Android 3.1 and later. However, this does not mean that accessory mode is supported in all Android devices running Android 3.1 or later. In this study, several android devices have been tested, however, only one of them was supporting the accessory mode. It seems that accessory mode can be configured at kernel level, and in some products, this mode is left in off mode.
- The other problem is that the only way to check the availability of the mode is to test it with the accessory. There are several applications to test the mode, however, they might result in some false positives. Therefore, getting access to compatible hardware is a little bit difficult.
- There are two releases of ADK and there are design differences between these releases. This results in problems while upgrading the application to the later version of ADK.
- In the last version, communication over Bluetooth has been given much attention

This is relatively new technology. This technology aims for open platform meaning that everyone can make their own accessory. If the design problems in ADK API and accessory mode configuration are solved, more accessories will be developed.

Acknowledgment

The authors would like to thank all other students contributing to the project, namely, Gozde Coban, Burak Kiremitci, Tugce Oktay and Ender Sancı. Moreover, they would like to thank the staff of Mechatronics Engineering Department, especially, Cahit Gurel, for providing the opportunity to manufacture the accessory.

This work was supported by Atilim University under the LAP (grant number: LAP-C-1213-02).

References

[1] (2011). Accessory development kit 2011 guide. Google Inc. [Online]. Available: http://developer.android.com/tools/adk/adk.html

[2] J. Ducloux, *et al.*, "An embedded USB dual-role system integrated for mobile devices," in *Proc. Argentine School of Micro-Nanoelectronics, Technology and Applications (EAMTA)*, 2012, pp. 66-72.

[3] A. Drumea, "Control of industrial systems using android-based devices," in *Proc. 36th International Spring Seminar on Electronics Technology (ISSE)*, 2013, pp. 405-408.

[4] G. Trehard, "Indoor pedestrian localisation solution based on anemometry sensor integration with a smartphone," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, 2012, pp. 1-9.

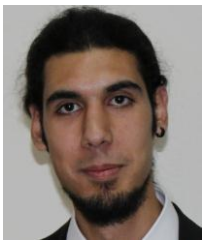
[5] R. Meier, *Professional Android 2 Application Development*, Wiley Publishing, Inc., 2010.
[6] M. Banzi, *Getting Started with Arduino*, O'Reilly Media, Dec. 2008
[7] V. Lee, H. Schneider, and R. Schell, *Mobile Applications: Architecture, Design, and Development*, Prentice Hall, 2004.
[8] (2011). Microchip's accessory framework for android. Microchip Microchip Technology Inc. Copyright (c). [Online]. Available: http://ww1.microchip.com

Dr. Fatma Cemile Serce has a BS and a MS in computer education and instructional technology. She has second MS and PhD in Information Systems. She worked as an Assistant Professor in the Department of Information Systems Engineering at Atilim University for six years. Her main research fields include mobile computing, cloud computing, software engineering. She is also one of the authors of textbook on C++. She is now studying cloud data management and analytics in University of Washington in USA.

Ahmet Erpay was born in Nigde. He received B.Sc. degree with First Class in Honors from Electrical and Electronics Engineering at Atilim University in 2013. He is currently a PhD student at Atilim University, and research engineer at Arlentus Control, Ankara. His research interests include embedded systems, signal processing, image processing and wireless communications.

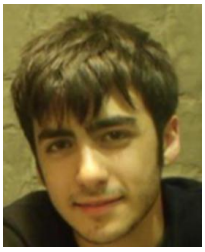
Dr. Ali Kara is an Associate Professor at Atilim University, and he was a EH consultant on defense and security projects, from 2006 to 2012, in BILGEM/TUBITAK. He was at Polytechnic University (Brooklyn, NY) between 1999-2000, and he has published extensively on radio communications, remote/virtual laboratories and technology enhanced learning, radar/electronic defense systems.

Bülent Ünal revieved the B.S.E.E. degree from Indurstrial Design, Faculty of Architecture, Middle East Technical University, Ankara, Turkey in 1993. He was the owner of a design company in Ankara between 1996-2010. He received M.S.E.E. degree from Atilim University, and he is currently working towards the Ph.D. degree at the Department of Architecture, Atilim University, Ankara and also instructor at the same university. His research interests include architectural design, industrial design, computer aided design and 3D visualization techniques.



Doğancan Özgubar received BS degree from the Department of Industrial Design at Atilim University in 2013. He is also studying Photography and Cinematography.

Ahmet Burak Öztürk received BS degree from the Department of Computer Engineering at Atilim University in 2013. He is interested in mobile application development.



Sinan Şekerci is fourth year student in the Department of Software Engineering. He is interested in Android and iOS mobile application development and programming languages. He received high achievement scholarship from Atilim University.