Design of biomedical signal acquisition equipment with real-time constraints using Android platform

J. Yepes¹, J. Aguirre², S. Villa¹.
¹ARTICA, University of Antioquia, Medellin, Antioquia, Colombia
²Electronics Department / University of Antioquia, Medellin, Antioquia, Colombia

Abstract - Android is an open source platform which includes operating system core, application program interfaces (APIs) and middleware, originally designed for mobile devices. This platform has become quite popular, extending its use to several electronic devices. However, communications with other digital devices such as Microcontrollers, FPGA’s, Microprocessors, data acquisition systems and ASICs have been a bottleneck in the application development process. The main reason for this issue is that Android OS doesn't have support for real-time operations. This paper presents the development and implementation of a medical application using an Android-based platform for management and visualization of an Electrocardiogram (ECG) signal and a specialized ASIC for data acquisition tasks which involves time-critical management, converting the device communications in a delicate requirement for this development. This paper describes the strategy for real-time solutions on the communication process, and shows results awarded in final implementation.

Keywords: Android, Embedded System Design, Real Time Applications.

1 Introduction

The Android platform [1] is a complete set of integrated software tools. It consists of an adapted version of the Linux kernel, middleware, application framework and a set of specialized APIs (Application Programming Interfaces) to develop mobile applications. It was initially designed for using on mobile phones; however its use has spread to a lot of heterogeneous embedded systems [2].

Among the several advantages that Android offers for developing applications, we may quote the following: it is an open source, provides a free development platform for creating mobile applications. It has a large community of developers working on it and facilitates the administration of a wide range of peripherals such as sensor, displays, communication interfaces, and others. Nowadays, All those features make Android one of the most attractive platform for developers [3].

However, those facilities are useless if they have not been developing a set of utilities (Drivers and APIs) that allow design applications since a high level of abstraction.

In this paper, we describe the development of a prototype designed for remote health supervision oriented to capture and display electrocardiograms (ECG) signals. This prototype is a part of the System Integration of Medical Monitoring and Interoperability for Telecare (SIMMIT) [4]. One of the most important objectives of this prototype consists in to develop a real time application aimed to capturing bio-signals to be displayed on a system running Android OS (called host). A dedicated embedded system is responsible for the signal acquisition. Such a system consists of two parts: An ASIC which performs the capturing of the signals from the human body and, a MCU (microcontroller unit) which acts like a link between the ASIC and the host. The results of the implemented system sampling, processing and displaying of an ECG (Electrocardiogram) signal are also present.

This paper has the following structure: Chapter 2 presents related work, Chapter 3 describes the hardware architecture used in the implementation of the application, Chapter 4 presents a detailed description of the target application and solution strategy regarding software concerns, chapter 5 shows the results obtained and finally conclusions and future work in Chapter 6.

2 Related work

Real-time applications have become a necessity for some embedded and mobile systems. Android features facilitate the development tasks; however, it is known the Android platform doesn’t have reliable support for real time applications.

Some strategies have been explored, searching a solution for this trouble. Specifically, four solutions have been proposed in [5]. The first approach contemplates the replacement of the Linux operating system by one that offers real-time features and it considers the inclusion of a real-time Virtual Machine (VM). The second one respects the Android standard architecture by proposing the extension of Dalvik [6] as well as the substitution of the standard operating system by a real-time Linux-based operating system. The third one only substitutes the Linux operating system for a Linux real-time
version and real-time applications use the kernel directly. Finally, the fourth one suggests the addition of a real-time hypervisor that supports the parallel execution of the Android platform in one partition while the other partition is dedicated to the real-time applications.

Solutions previously mentioned have a complex background: They suggest delicate modifications to the virtual machine and/or kernel lawyer inside the Android’s standard architecture. This kind of modifications implicates a detailed knowledge about the operator system and inter-lawyer communication, thus, an implementation with real-time support could take a long development time.

Alternatively, in [7] native C code library has been used from developed applications with some time-restrictions. They report the time-execution of a native application has a significant improvement over a similar Java application (running over Dalvik Virtual Machine). However, the improved time doesn’t imply real-time support and the native approach cannot guarantee the timing requirements.

### 3 Android architecture

Android is a software platform, rather than just an OS, which has the potential to be utilized in a much wider range of devices. In practical terms, Android is an application framework on top of Linux, which facilitates its rapid deployment in many domains. [8]. Android’s framework is divided in lawyers, as it can be seen in Figure 1.

![Android Architecture](image)

**Figure 1.** Android Architecture.

Linux kernel gives support to low-level components, mainly hardware drivers are managed by this lawyer. Peripherals as cameras, printers, flash memories, Wi-Fi, displays, etc. Have to be directly controlled by the kernel.

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

Every Android application runs into its own process, with its own instance of the Dalvik. Dalvik is an optimized virtual machine (VM) for mobile devices and runs classes compiled by a Java language compiler that have been transformed for Android. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Android offers developers the ability to build applications in an easy way. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much more. At the top, Android offers Java programming language approach to development community with total access to framework.

### 4 Description of hardware platform

This work has involved three hardware platforms: A system based on TI’s DM3730 processor for multimedia application called Blizzard, a data acquisition system from Texas Instruments called ADS1298, specialized for medical applications, and a Freescale 8-bit microcontroller unit (MC9S08JM60) [9]. The block diagram of the complete ECG signals acquisition system is shown in Figure 2.

![Hardware Block Diagram](image)

**Figure 2.** Hardware block diagram.

#### 4.1 Data Acquisition Platform

The ADS1298 is a fully integrated analog front end (AFE) for patient monitoring. It belongs to family of integrated circuits manufactured by TI, which incorporates all the features that are required in medical applications such as electrocardiogram (ECG) and electroencephalogram (EEG). The ADS1298 has eight channels with simultaneously sampling and the possibility of using digital analog converters (ADCs) delta-sigma with 24-bit resolution by channel, with 32kSPS throughput capability. It also integrates programmable gain amplifiers (PGAs) for signal conditioning, internal reference voltage and an oscillator, all those inside a single integrated circuit. Figure 3 shows the block diagram of the ADS1298.

With its high integration degree, excellent benefits and exceptional performance, the ADS1298 allows the development of medical instrumentation systems by reducing the size, power consumption and decreasing development costs [10].
4.2 Microcontroller Unit

Due to unsupported real time operations presented by the Android platform, it became necessary to use an auxiliary subsystem in order to accomplish the real time requirements in communication tasks between the host and the data acquisition system. We used the Freescale Semiconductor's MC9S08JM60 Microcontroller Unit (MCU). It is member of the low-cost, high-performance HCS08 family of 8-bit MCUs, has a Von-Neumann architecture, Up to 60 KB of on-chip flash memory, 4KB of data memory, 24-MHz of internal bus frequency, two full duplex Serial Peripheral Interfaces (SPIs) communication ports, and other variety of modules. SPI communication speed can be established based on the MCU bus frequency and is configurable through control registers, to facilitate communication with a large number of devices.

4.3 Main Display System

Android platform is supported by Texas Instrument's DM3730 processor. The DM37x generation of high-performance, applications processors are based on the enhanced device architecture and are integrated on TI's advanced 45-nm process technology. This architecture is designed to provide best in class ARM and Graphics performance while delivering low power consumption. This balance of performance and power allows the device to support a huge variety of multimedia applications [11].

The DM3730 integrates a GPP (General Purpose Processor) ARM Cortex™-A8 @ 1GHz, a DSP (digital signal processor) TMS320C64x @800MHz plus a graphics accelerator 2D and 3D PowerVR SGX 530. The GPP controls all hardware resources using a generic operating system like Linux, Windows CE or, in this case, Android. The DSP acts as coprocessor of GPP. It also integrates various peripherals and interfaces to connect the different types of external devices.

5 Application description

In this paper we develop a prototype for biomedical monitoring. The main objective of this system is collect and transmit first-hand bio-signal information to a host for medical tracking and recording when a patient in emergency state within a medical assistance vehicle or when he's located in other place, far from a medical center.

The signals derived from monitoring equipment such as ECG, heart rate, respiratory rate, oxygen saturation and blood pressure should be integrated with patient’s record. This information will be placed in an appropriate way at the patient's electronic medical records using a standard format in order to send it to a remote location through a wireless network whenever the medical staff requires it. Figure 4 illustrates the system functionality and its environment.

Here we focus on ECG signal acquisition process, because this signal has the highest time variability, and therefore, it demands resources for processing tasks and high-bandwidth capability.

5.1 Synchronization Problem

Initially, the ADS1298 was directly connected to Blizzard platform through SPI ports in both devices, because it was thought that the system would operate properly. However, the acquired signal did not show the expected behavior. For example, for sinusoidal test waveform, we got a distorted version (Figure 5). Some strategies were implemented trying to fix the problem.

The issue was related with the fact that the OS in the Blizzard platform does not support the real-time demands of the application, necessary for the communication. Such a problem becomes more severe when the ASIC operates under continuous conversion mode, because the data acquisition times must be accurately respected, for the sake of ensuring...
correct signal sampling. Android operating system, cannot guarantee such conditions, as shown in Figure 6.

![Distorted sinusoidal signal](image1)

**Figure 5. Distorted sinusoidal signal.**

For testing, a GPIO pin toggles when the OS makes a sample request to the data acquisition system. As it can be seen in Figure 6, the samples are requested at different time periods. This phenomenon generates a distorted version of the acquired signal.

![Communication task is not periodically handled by the OS.](image2)

**Figure 6. Communication task is not periodically handled by the OS.**

### 5.2 Solution Strategy

In order to meet real time constraints imposed by the target application, and so solving synchronization problems mentioned above, it is necessary to find a mechanism to efficiently handle the sampling times required by the signal acquisition system, without leaving aside the many advantages which the Android platform has.

As mentioned in the section on related work, the alternatives currently available to use Android in real-time applications do not provide an optimal solution. Because of this, the paper presents an approach that seeks to separate the real time processing demand of the rest of the application. Thus using Android for what it does best: managing UI (graphical user interface) and cellular connectivity, and a subsystem that is responsible for managing the signal acquisition system.

That is why we use the MC9S08JM60 microcontroller, described in section 3.2. This microcontroller was used in order to overcome the synchronization drawback. It must fulfill the task of attaching the data acquisition system with the Android platform, seeking to meet the necessary requirements for the correct time sampling of the ECG signal and avoiding the overlapping/loss of data problems.

The MCU provide two serial interfaces (SPI), one to communicate with the data acquisition system and other to communicate with the Blizzard platform, the connection is illustrated in Figure 7. The SPI1 is configured as master mode with a frequency of 1Mbps. The SPI2 is configured as slave mode, for this reason the operation frequency is imposed by the master device (Blizzard platform) with a frequency of 500 Kbps.

Furthermore, it implements a First Input-First Output (FIFO) memory management. The FIFO implementation is very important because it decouples the data processes offered from the ADS1298 and demand from the Blizzard platform.

![Connection diagram among subsystems.](image3)

**Figure 7. Connection diagram among subsystems.**

In conclusion, we have 3 subsystems as it can be seen in figure 3: the subsystem based on android, the MCU to decouple the processes, and the data acquisition subsystem. The application developed on the Android platform is basically responsible for 3-function: receiving, storing, and displaying information from the data acquisition system.

The reception is performed via a SPI connection between Android-based system and a microcontroller, where the first acts as the master and the other one acts as slave. Also, there is a local repository, which collects all necessary information about the application users, both medical staff and patients. The MCU is responsible of management the data acquisition subsystem using a SPI protocol and implement FIFO policies.

### 6 Results

By deploying the application, various tests were performed. For the sake of evaluating the correct system operation, a sinusoidal wave was sensed and displayed, as shown on Figure 8. Notice that the previous exhibited distortions (Figure 5) have been fully corrected.
Thereafter samples were taken from an artificial signal from an ECG signal generator as shown on Figure 9. It can be seen that a continuous signal was obtained without any kind of distortion as required for proper system operation. Sampling of this artificial signal contributes to the system validation processes, since the displayed signal is precisely the expected.

The main test was conducted by connecting the System to a patient, as shown in Figure 10.

7 Conclusions

A real-time embedded system was designed for capturing, processing, storing and displaying an ECG signal. For this purpose, a platform based on Android OS, a conventional 8-bit microcontroller and data acquisition system were used.

Proposed strategy provides a wide running flexibility, mainly because the application acquires independency from the specific Android device that executes it. It’s important to remark the fact that it’s not required to make any modification on the Android’s standard architecture for adding real-time features. In other hand, adding new hardware to the system has any significant increase in system complexity, because extra hardware is quite simple, low cost, and totally transparent to the application.

DM3730 processor with Android OS provide an excellent solution for applications requiring joint user interface, connectivity and complex applications.

8 Acknowledgments

We would thank to microelectronics and control group researchers for the support in the development of this work. SIMMIT is being funded by COLCIENCIAS, the ICT Ministry of Colombia, and ARTICA (Research Center of Excellence in ICT) in the project “Design methodology of embedded systems with high reliability and performance focused on critical applications”.

9 References


