A Method for Universal System Interactions for Smart Phone

[1]Dr.C.Brintha Malar, [2]Dr.K.Siva Sankar

[1]Assistant Professor, [2]Associate Professor
[1]Department of Physics,
[1]Udaya School of Engineering, Vellamodi, India

***Abstract*: In this paper, we present a system architecture that allows users to interact with embedded systems acted in their proximity using Smart Phones. We have identified four models of interaction between a Smart Phone and the surrounding environment: universal remote control, dual connectivity, gateway connectivity, and peer-to-peer. Although each of these models has different characteristics, our architecture provides a unique framework for all of the models. Central to our architecture are the hybrid communication capabilities incorporated in the Smart Phones. These phones have the unique feature of incorporating short-range wireless connectivity (e.g., Bluetooth) and Internet connectivity (e.g., GPRS) in the same personal mobile device. This feature together with significant processing power and memory can turn a Smart Phone into the only mobile device that people will carry wherever they go.**

I. Introduction

Recent advances in technology make it feasible to incorporate significant processing power in almost every device that we encounter in our daily life. These embedded systems are heterogeneous, distributed everywhere in the surrounding environment, and capable of communicating through wired or wireless interfaces. Most of our daily interactions with the surrounding environment are still primitive and far from the ubiquitous computing vision. Our pockets and bags are still jammed with a bunch of keys for the doors we have to open/close daily (they did not change much since the Middle Ages), the car key or remote, access cards, credit cards, and money to pay for goods. Any of these forgotten at home can turn the day into a nightmare. If we travel, we also need maps and travel guides, coins to pay the parking in the city, and tickets to take the train or subway. In addition, we are always carrying our mobile phone, which for some mysterious reason is the least likely to be left at home. When we finally arrive home or at the hotel, we are "greeted" by several remote controls eager to test our intelligence. All these items are absolutely necessary for us to properly interact with our environment. The problem is that there are too many of them, they are sometimes heavy, and we will likely accumulate more and more of them as our life go on, requiring much larger pockets.

For this problem, the community does not lack innovative solutions that address some of its aspects (e.g., wireless micro servers, electronic payment methods and digital door keys). What is missing is a simple, universal solution, which end-users are likely to accept easily. Ideally, we would like to have a single device that acts as both personal server and personal assistant for remote interaction with embedded systems located in proximity of the user. This device should be programmable and support dynamic software extensions for interaction with newly encountered embedded systems (i.e., dynamically loading new interfaces.

We believe that Smart Phones are the devices that have the greatest chance of successfully becoming universal remote controls for people to interact with various devices from their surrounding environment; they will also replace all the different items we currently carry in our pockets. Smart Phone is an emerging mobile phone technology that supports Java program execution and provides both short range wireless connectivity (Bluetooth) and cellular network connectivity through which the Internet can be accessed.

In this paper, we present a system architecture that allows users to interact with embedded systems located in their proximity using a Smart Phone. We have identified four models of interaction between a Smart Phone and the surrounding environment: universal remote control, dual connectivity, gateway connectivity, and peer-to-peer. Central to our architecture are the hybrid communication capabilities incorporated in the Smart Phones which allow them to interact with the close-by environment through short-range wireless networking and with the rest of the world through the Internet over cellular links. This feature together with significant processing power and memory can turn a Smart Phone into the long awaited universal personal assistant that can make our daily life much simpler.

II. An Embedded System

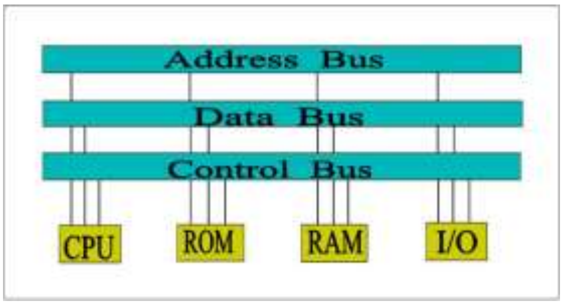
The embedded system is a combination of computer hardware, software and, perhaps, additional mechanical parts design to perform a specific function. A good example is an automatic washing machine or microwave oven .Such a system is in direct contrast to a personal computer, which not designed to do only a specific task. The PC aids you in drafting a letter, in computing at a faster rate in chatting with friends, and so on, but an embedded system is designed to do a specific task with in a given time frame, repeatedly, endlessly, with or without human interaction. A PC is made up of numerous embedded systems, such as a keyboard, hard drive etc. The function of a simple modem is to convert analogue signals to digital signals, and vice versa. This means it must have a certain amount of logic to perform that process in time and again endlessly.

It is important to note that all embedded systems do not have same hardware and software, which is why these systems perform varied tasks. It's even possible to have an embedded system that does not contain any processor and corresponding software to run through it. In such system, called hardwired systems, the hardware and software is replaced with integrated circuitry that performs a same function .However, a lot of flexibility is lost when applications are implemented this way. It is much easier to change the software code than to redevelop the hardware, for bringing about the small changes in application for which the system has been designed.

III. Embedded Hardware

All embedded systems need a microprocessor, and the kinds of microprocessors using them are quite varied. A list of some of the common microprocessor families are ZILOG Z8 family, INTEL 8051/80188/X86 family. An embedded system also needs memory for two purposes – to store its program, and to store its data. Embedded

systems store data and programs in different memories .This is simply because embedded system does not have an hard drive and the program must be stored in memory, even when the power is turned off .This special memory that remembers program even without power, is called ROM Very often this systems have a serial port I/O interfaces, or hard ware to interact with sensors



So an embedded system has a micro-processor or micro controller for processing information, memory for storing embedded software programs and data and I/O interfaces for external interfaces. The functional diagram is given above.

The processor uses the address bus to select a specific memory location within the memory sub system or a specific peripheral chip. The data base is used to transfer data between the processor and memory sub system or peripheral devices the control bus provides timing signals to synchronize the flow of data between the processor and memory sub system or peripheral devices.

IV. Embedded Software

C has become the language of choice for embedded programmers. The greatest strength of C is that it gives embedded programmers an extraordinary degree of direct hardware control without sacrificing the benefits of high level languages. Compilers and cross compilers are also available for almost every processor with C.

Any source code written in C or C++ or assembly must be converted into an executable image that can be loaded onto a ROM chip. The process of converting the source code representation of your embedded software into an executable image involves three distinct steps.

- First, each of the source files that make an embedded application must be compiled or assembled into distinct object files.
- Second, all of the object files that result first step must be linked into a final object file called the relocatable program.
- Finally physical memory address must be assigned to relocatable program.

The result of the third step is a file that contains an executable image that is ported on to the ROM chip. This ROM chip, along with the processor and other devices and interfaces, makes an embedded system run.

V. Smart Phones Technology

With more than a billion mobile phones being carried around by consumers of all ages, the mobile phone has become the most pervasive pocket-carried device. We are beginning to see the introduction of Smart Phones, such as a result of the convergence of mobile phones and PDA devices. Unlike traditional mobile phones, which have limited processing power and act merely as “dumb” conduits for passing voice or data between the cellular network and end users, Smart Phones combine significant computing power with memory, short-range wireless interfaces (e.g., Bluetooth), Internet connectivity (over GPRS), and various input-output components (e.g., high-resolution color touch screens, digital cameras, and MP3 players). Sony Ericsson P800/P900 runs Symbian OS [12], an operating system specifically designed for resource constrained devices such as mobile phones. Additionally, the phone has 16MB of internal memory and up to 128MB external flash memory. It runs a version of MontaVista Linux. Bluetooth is a low-cost, low-power standard for wireless connectivity. Today, we can find Bluetooth chips embedded in PCs, laptops, digital cameras, GPS devices, Smart Phones, and a whole range of other electronic devices. Bluetooth supports point-to-point and point-to-multipoint connections. We can actively connect a Bluetooth device to up to seven devices simultaneously.

VI. Smart Phone Interaction Models

A Smart Phone can be used to interact with the surrounding environment in different ways. We have identified four interaction models: universal remote control, dual connectivity, gateway connectivity, and peer-to-peer. With these models, a Smart Phone can be used to execute applications from as simple as remotely adjusting various controls of home appliances or opening smart locks to complex applications such as automatically booking a cab or ordering/paying in a restaurant using an ad hoc network of mobile phones to connect to the cashiers computer.

VII. Conclusions

In this paper, we have discussed for turning the Smart Phone into the device that people carry in their pockets wherever they go. The Smart Phone can be used as both personal server that stores or downloads data that its user needs and personal assistant for remote interaction with embedded systems located in the user's proximity. To achieve this vision, we have presented unified system architecture for different models of interaction between a Smart Phone and the surrounding environment. Central to this universal interaction architecture is the dual connectivity feature of Smart Phones, which allows them to interact with the close-by environment through short-range wireless networking and with the rest of the world through the Internet over cellular.

References

[1] Dr. K Siva Sankar, "A Method for Developing an Operating System for plug and play Bootstrap Loader for USB Drive", International Journal of Computer Science Issues, Vol. 8, No. 1, pp. 295-301, 2011. (Indexed, Impact Factor-0.24)

[2] Dr. K Siva Sankar, "A Method for Developing Bootstrap Loader and Configuring Network for Live USB Flash Drive", International Journal of Decision Making in Supply Chain and Logistics, Vol. 1, No. 2, pp. 221-232, 2010. (Scopus Database)

[3] Benjamin Speckmann, "The Symbian mobile platform," MS.Thesis, Depart. comp. science, Eastern Michigan Univ., Michigan, US, 2008.

[4] Dr. K. Siva Sankar, "A secure and reliable method to protect usb data" , APTIKOM Journal on Computer Science and Information Technologies, Vol. 4, No. 2, 2019, P : 57 – 60 (ISSN: 2528-2417)

[5] Dr. K. Siva Sankar, "Dynamic Boot Loader in USB Drive with Enhanced User Experience", Journal of Web Engineering, Accepted, 2019. (SCI Indexed)

[6] Benny Skogberg "Symbian Application Development," MS.Thesis, Depart. Comp. Science, Malmo Univ., Sweden, 2010.

[7] Chien-Wei Chang, Chun-Yu Lin, Chung-Ta King,"Implementation of JVM Tool Interface on Dalvik Virtual Machine",paper appears in (VLSI-DAT), 2010 International Symposium on Digital Object Identifier 26-29 April 2010. [4] Bimal Gadhavi & Khushbu Shah," Analysis of the Emerging Symbian Market ", Project Report Presented to SanJosé State University May 2010.

[8] Dr. K. Siva Sankar, "A Secure Boot Loader System for Loading an Operating System with Fingerprint Authentication", International Journal of Innovative Technology and Exploring Engineering, Vol.8, No.10, pp. 2368-2374, 2019. (Scopus Indexed)

[9] Stefan Brahler," Analysis of symbian architecture", Department of computer science, Karlsruher institute of technologies, Germany, june 2010

[10] Marakana,"SymbianBootcamp Training Course http://marakana.com/training/symbian/ symbian_bootcamp.html 18 Oct, 2012.

[11] James Steele, The Symbian Developer's Cookbook, New York : Pearson, 2011.

[12] http://www.developer.symbian.com, 23 Oct, 2012.

[13] Dr. K. Siva Sankar," Design of a Dynamic Boot Loader for loading an Operating System", Journal of Computer Science, 15(1),pp 190-196, 2019, DOI: 10.3844/jcssp.2019.190.196 (Scopus indexed- Impact factor 0.34), ISSN: 1552-6607