Computing, Information Systems, Development Informatics & Allied Research Journal
ISBN 978-2257-44-7 (Print) ISSN 2167-1710 (online)
Vol. 5 No. 1. March, 2014 – www.cisdijournal.net

USING EMBEDDED TECHNOLOGY IN END-USER PROGRAMMING OF SMART SPACES WITH MOBILE DEVICES

Onukwugha, C. G.
Department of Computer Science,
University of Port Harcourt
Port Harcourt, Nigeria
onukwugha2000@yahoo.com

Asagba, P. O. (Ph.D)
Department of Computer Science,
University of Port Harcourt
Port Harcourt, Nigeria
pasagba@yahoo.com

ABSTRACT
A recent shift in computing paradigm from stand-alone microcomputers and mainframes towards entirely pervasive computing where billions of miniature, ubiquitous inter-connected computing elements weave themselves into the fabric of everyday life. Embedded systems run the computing devices hidden inside every object and appliance such as cell phones, toys, handheld PDAs, cameras, microwave ovens, cars, airplanes, etc. These numerous, easily accessible devices connected to each other and to network infrastructure exhibit context-awareness of an environment in order to optimize their operation in that environment. In this paper, we examined embedded systems in end-user programming of smart spaces with mobile devices. We designed and implemented a microcontroller-based system capable of monitoring and controlling the electronic appliances in a home from any location. We adopted a task-driven computing approach of the composition of the semantic web. The end user uses the functionality of the networked devices in the home as semantic web services to arbitrarily form his request which involves the typing of SMS through the user-friendly interface of a Java enabled mobile phone. An Arduino microcontroller for generating the timing and control signals programmed using Wiring language was used. The GSM wireless technology was used for transmission and reception of the data. Our work addresses the problem of energy wastage and domestic accidents by enabling end-users to easily use their mobile devices to monitor and instruct their home devices from any location over a wireless network.

Keywords: Embedded Technology, Smart Spaces, End-User Programming, Mobile Devices, Pervasive Networking

1. INTRODUCTION

A new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish into the background has emerged [1], a concept of a man who envisioned how computers are going to weave themselves into the fabric of everyday life until they are indistinguishable from it. Mark Weiser conceptualized a shift in computing paradigm from stand-alone microcomputers and mainframes towards entirely pervasive computing where billions of miniature, ubiquitous inter-connected devices will be spread worldwide. The result would be the proliferation of smart environments and wearable computers. Smart environment is a technological concept that, according to Mark Weiser is “a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network” [2]. Embedded technology is the driving force of pervasive networking and networked devices is the premise of pervasive computing. Pervasive networking is the ability to communicate and access the same types of services – anytime, anywhere.

This is regardless of location, type of network or type of device used to access the network. There is a convergence of technology, business needs and end user interest that is driving the development of networks to support pervasive communications, whether wireless or wireline, whether from home, business, hotel coffee shop or on the move [3].

Embedded Systems

Embedded systems generally have the following characteristics pointed out in the various definitions given:

- They are intelligent systems (regarded as smart or expert systems) built to perform its duty, completely or partially independent of human intervention.
- They can be standalone or interfaced to PC microcomputers.
- They are cost-effective and are specially designed to perform specific tasks in the most efficient way.
- They are enshrined into electromechanical systems, electronic, biomedical, domestic, industrial and commercial goods and interact with physical elements in our environment, viz. controlling and driving a motor, sensing temperature, etc.

Embedded systems often are real-time systems. A real-time system is defined as a system whose correctness depends on the timeliness of its response. Examples of such systems are flight control systems of an aircraft, sensor systems in nuclear reactors and power plants. Embedded systems run the computing devices hidden inside every object and appliance such as cell phones, toys, handheld PDAs, cameras, microwave ovens, cars, airplanes, etc. These numerous, easily accessible devices connected to each other and to network infrastructure exhibit context-awareness of an environment in order to optimize their operation in that environment. Wireless technology represents a rapidly emerging area of growth and importance for providing ubiquitous access to the network: WLANs based on the IEEE 802.11 standard are being implemented constantly in the houses and Broadband Wireless (BW) is also an emerging wireless technology which is competing with Digital Subscriber Line (DSL) [4]. RFID, Bluetooth and ZigBee are wireless technologies that also support pervasive networking. In this paper, we designed and implemented a microcontroller based system capable of monitoring and controlling the electronic appliances in a home from any location.

2. RELATED WORK

The increasing ubiquity of heterogeneous devices such as laptop computers, palms, mobiles, etc. shows that users prefer a ubiquitous access of a system rather than to be uncomfortably forced to go physically to the nearest control point [4]. One effort to develop smart homes is through the use of old, cheap and simple technologies like X-10 protocol to transfer data in the home network. This approach created low cost home automated systems considering the advantage that X-10 technology does not require additional wiring (X-10 technology makes use of existing house wiring). The major challenge of X-10 devices is noise on the wiring. Microcontroller based automatic gate [5] is an alternative to a manually controlled gate which is laborious, frustrating, costly, and energy consuming. There are many devices which are made with microcontroller some of which are GSM phones, PDAs, Sound systems, Pumping machines, Robots etc. The proposed system comprises of several components, the first is the sensors which detect any vehicle near it and send a signal to another set of components. The system developed can perform operations like opening the gate automatically, closing automatically, lock up totally when the car park is filled up and no vehicle can gain an entrance into the park.

[6] developed a GSM based control system for electrical appliances which made use of a GSM module, a controller (PIC16F877A) and relay driver (BufferULN2003). Nokia 6100 is used to send SMS message which is retrieved and stored in the SIM (subscriber Identity Modula) memory of the receiving GSM unit. The microcontroller extracts and processes the signal and eventually carries out the corresponding instructions. The shortcoming of this system, however, is that the user friendliness of the user interface is not established. The system also does not have facility for storing used messages that could be useful in recording user activity in the smart environment.

The convergence of home automation systems and mobile technology is demonstrated in another research work where GSM networks and PIC microcontroller (16F877A) were used to design a smart data logger system capable of monitoring automatically the punctuality of staff in educational institutions [7]. The various entities in the GSM network are connected one to another through signaling protocols over the reference points. The SPF200-USB fingerprint sensor has the USB-1.1 controller logic built into the sensor, it quickly captures the image of the fingerprint, analyses it and compares it to a previously registered fingerprint template.

Internet based home automation system has been proposed by Ali Ziya Alkar in his work on "An Internet Based Wireless Home Automation System for Multifunctional Devices." The work presents a low cost secure web-based, flexible, wireless solution where the home appliances of most types can be connected to a central node through a server [8]. The system is modeled with three different units; the PC unit, the manager node (master) and other nodes (slaves). The following programs run on the PC: web server and Internet web page, Database and its platform, user interface program. Through the Internet the database is accessed directly. The master node transmits the information directed by the computer to the nodes and also transmits the reply back to the computer. The master node is made up of an RS232 communication interface, PIC 16F877 microcontroller and an RF module for data transfer between other nodes in the system. The information that is transmitted and received is displayed on the LED indicators. The major lapse of this system is that manipulation and monitoring of the system will be disrupted during a power outage on the PC side.

In their paper, [9] proposed an SMS-based wireless control system for automating home appliances and security using GSM technology. The system comprises two subsystems – Appliance control subsystem, enables the user to control home appliances remotely whereas the security alert subsystem provides the remote security monitoring. Major components include a PC dedicated to the control of appliances and security, RS232 GSM modem and a cell phone. Remote controlled operation for home automation was equally researched by [4] with different network technologies for the optimization of the use of such systems. IP technology using UPnP package (where each device receives a valid IP address) directly connected to the Device and service descriptions defined by the UPnP Device Architecture. The GSM technology programmed using AT/AT+ commands which provides another security layer (modem will respond only to specific mobiles) and certain robustness was also used. Normally, the probability of accessing the GSM network will be higher than accessing the Internet [4].

There are a lot of smart systems which are targeting end users but the extent to which the end users can program them is the weakness we have observed in their design and patterning. This research follows the task-driven approach of the composition of the semantic web. This is because it is an approach that interactively and continuously captures the user's intent in a dynamic fashion. The functionality exposed by the networked devices in the home as semantic web services would be used by the end user to arbitrarily form his request which involves the typing of SMS through the user-friendly interface of a Java enabled mobile phone. An Arduino microcontroller (based on the AVR core architechture) for generating the timing and control signals programmed using Wiring language was used. The semantic descriptions of the services shield the user from the complexity of the underlying system. The GSM wireless technology is considered ideal for the transmission and reception of signals.

2.1 Task-Driven Computing Approach

The task-driven programming approach frees the user from the afore-mentioned low-level activities/operations. The key idea behind this approach is that the system should take over many low-level management activities so that users can interact with a computing system directly in terms of high-level tasks that they wish to accomplish. This new approach revolves around the following elements:

- System-maintained, globally-accessible abstract computing tasks and task states;
- New types of services that support automatic configuration;
- Automatic selection and configuration of service to accomplish computing tasks and re-establish computing contexts;
- Proactive guidance to help a user accomplish a task and compensate for resource limitations in an environment;
- Automatic management of dynamic service and resource conditions.

Making computing contexts globally accessible frees users from having to remember them mentally. Automatic collection of environment information frees users from having to locate accessible services in an environment by hand. Automatic task-based service discovery and configuration makes rapid configuration possible, even in the presence of large dynamic collections of devices and resources. There has been a lot of research on dynamically locating software services. This ranges from early efforts such as CORBA (Common Object Request Broker Architecture) and COM (Component Object Model) to the more recent service discovery frameworks such as Jini and universal plug and play (UP_nP). These mechanisms allow a client to dynamically locate and access software services. Many researchers have looked at the issue of software composition (sometimes called meta-programming) [10]. Formal languages have been developed to better enable specification of component interfaces crucial to component composition. Software component frameworks such as Javabeans and COM allow developers to create software systems out of pre-built binary components. Domain-specific component integration frameworks such as High Level Assembler (HLA), IBM Sanfrancisco, Enterprise, Javabeans etc. have been developed to support software component composition in specific domains. There also has been some HCI work on integrating multiple computing devices. Project Pebbles demonstrates that PDAs can collaborate with PCs using ad hoc mechanisms [11]. Task-driven computing takes such efforts a step forward by delivering a general framework that enables the collaboration among arbitrary computing devices

2.2 Architecture of a Task Computing Environment (TCE)

The general architecture of a task computing environment (TCE) is shown in Figure 1. It is made up of four layers namely: realization layer, service layer, middleware layer and presentation layer [12].

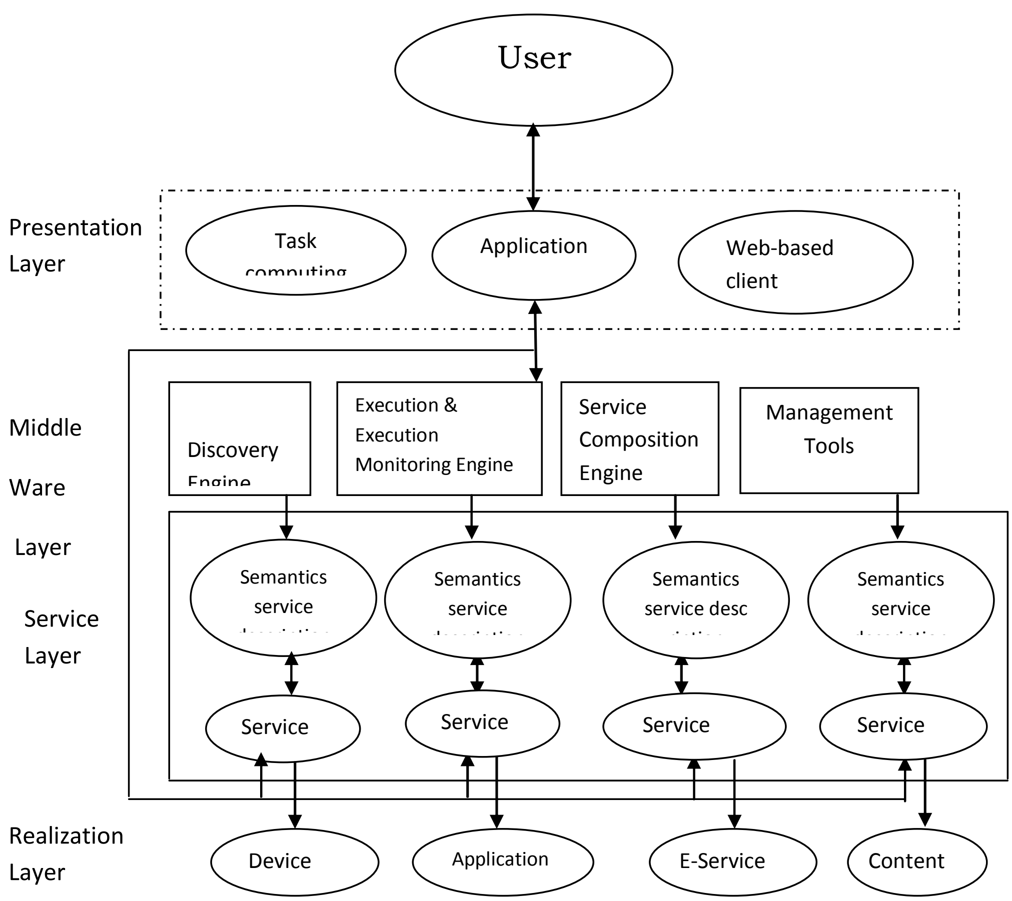

Fig.1: The general architecture of a task computing environment [12]

a) **Realization Layer:** The bottom most layer encompasses the universe of devices, applications, e-services and content, where all functionality available to the user originate.
b) **Service Layer:** These various sources of functionality are made computationally available as services, in the sense that, service interfaces are employed to access (execute) this functionality. Each service is associated with at least one semantic description, which sometimes may be created on-the-fly as services might be created dynamically. Services are the abstraction of functionality in the task computing universe, and semantic descriptions of these services are meant to shield the user from the complexity of the underlying sources of functionality and make it easy for the user to employ these sources in accomplishing desired tasks.
c) **Middleware Layer:** These goals are enabled by the middleware layer components that are in charge of discovering services, deciding how services can be composed, executing services and monitoring service execution, and finally enabling and facilitating a variety of management tasks, including the creation and publishing of semantically described services.
d) **Presentation Layer:** The most important aspect of task computing is the presentation layer, which uses the capabilities of the layers below in order to provide the user with a "Task" abstraction of the complexity of whatever lays underneath. Task computing researchers have developed a variety of clients for that purpose, such as voice, textual and graphical interfaces, (referred to as task computing clients TCC), and a web-based interface (utilizing a web browser). The presentation layer presents to the user an environment where functionality that can be transient and dynamically created can be assembled (real time) to perform users' tasks. Since the middleware layer components expose well-defined service API's, it is possible to create custom applications in the presentation layer in any development environments that can invoke web services.

Task computing defines a semantic service discovery mechanism (SSDM) which uses the underlying service discovery mechanisms offered by the middleware layer (e.g. UPnP SSDM) to discover semantically described services. For example, UPnP's discovery mechanism is used to find the UPnP devices on the subnet (not all of which are task computing enabled services), and for each UPnP device, the task computing client (TCC) invokes one specific UPnP action (getDescriptionURL) to determine if the UPnP device represents a Task Computing-enabled service [13]. If so, the TCC proceeds to download the semantic service description (SSD) from the UPnP device.

3. SYSTEM DESIGN

In order to ensure low component count and increased reliability, the design of our system is based on the arduino microcontroller. Arduino is not just a microcontroller but a single board microcomputer dedicated to embedded control applications. It is a software framework but built on AVR hardware architecture. Its complexity lies in the software. The proposed system is managed from the software side since managing it on the hardware side is very expensive. Figure 2 shows the design of the proposed system.

Computing, Information Systems, Development Informatics & Allied Research Journal
ISBN 978-2257-44-7 (Print) ISSN 2167-1710 (online)
Vol. 5 No. 1. March, 2014 – www.cisdijournal.net

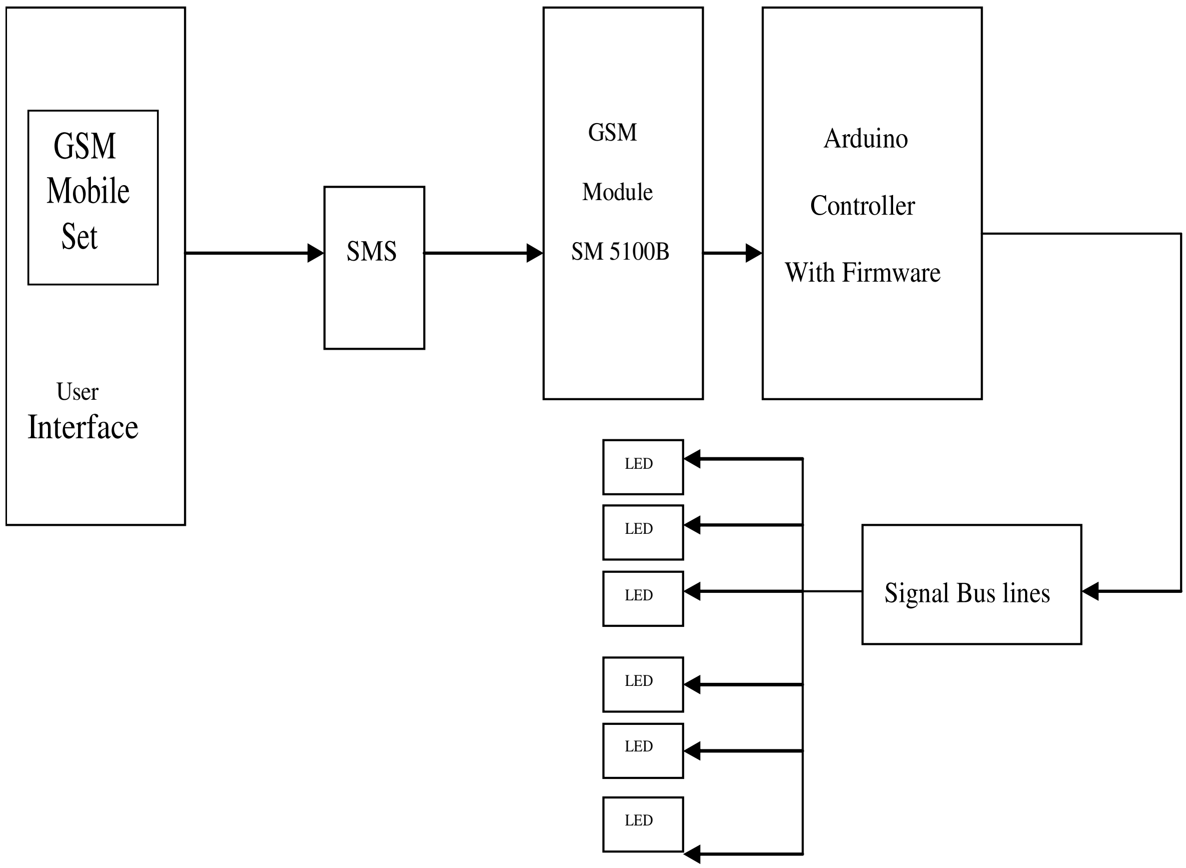

Fig. 2: Proposed system design

4. CASE DESIGN OF ELECTRONIC INTERFACE

The electronic interface is the hardware component design that illustrates the schematic features and interface of the entire system. A computer aided system engineering (CASE) tool - Simulink was used in designing the electronic interface. It clearly shows the manual switch equivalent of the message responses expected for each activation of the LED. The message code is also shown in the design as well as the corresponding actions expected from the system. In the design, a black LED illustrates a system LED that is OFF while the Green LED illustrates a system LED that is ON. Figure 3 clearly shows a CASE electronic interface design indicating where the LED is either switched ON or OFF. The design comprises four major subsystems: Switch Select, Transmission, Delay and SMS Module.

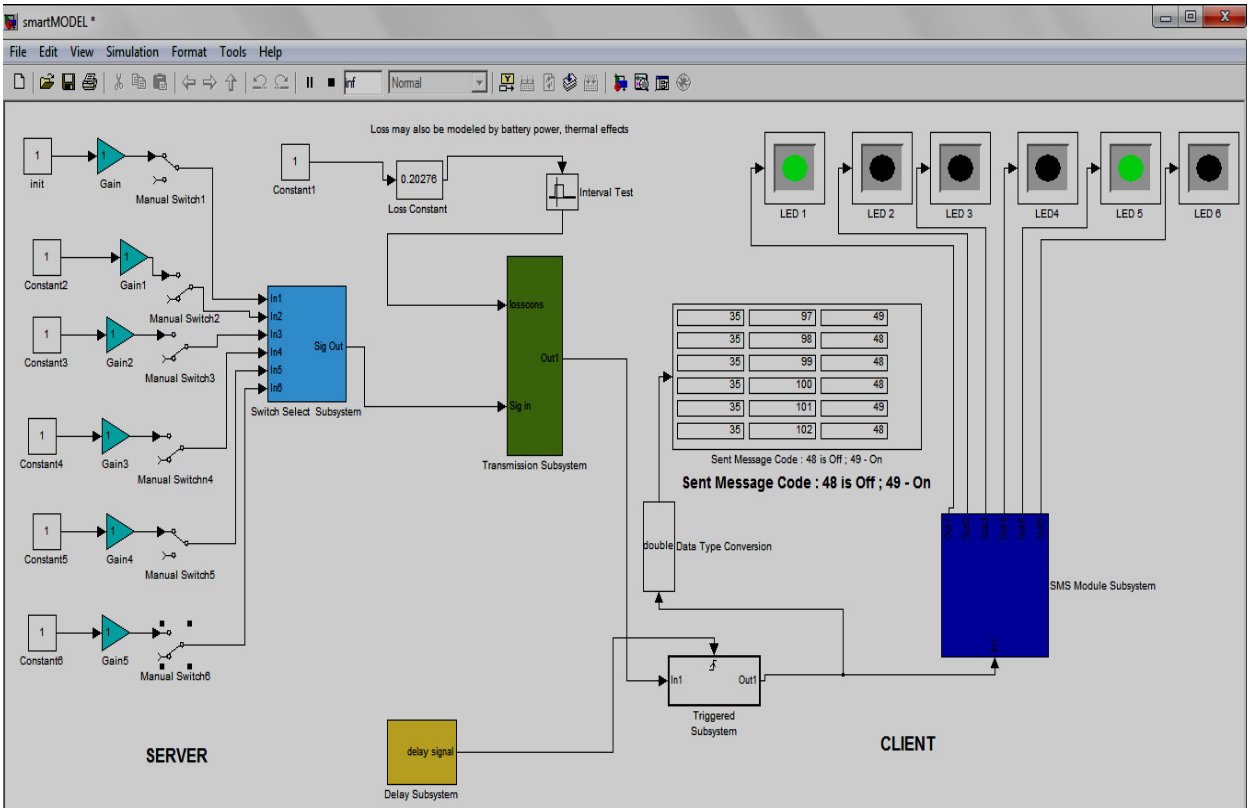

Fig. 3: A CASE electronic interface design

Figure 4 shows the Switch Select Subsystem. In this subsystem the hash codes for sending the control signal to the smart system are encoded. The hash codes are coded in an embedded Matlab function block concatenated using a matrix concatenated block and sent to the main model through a signal output port. Control signals are typically received from import blocks In1 to In6 which in turn receives its signals from manual switches (1-6) modeled as user selectable input.

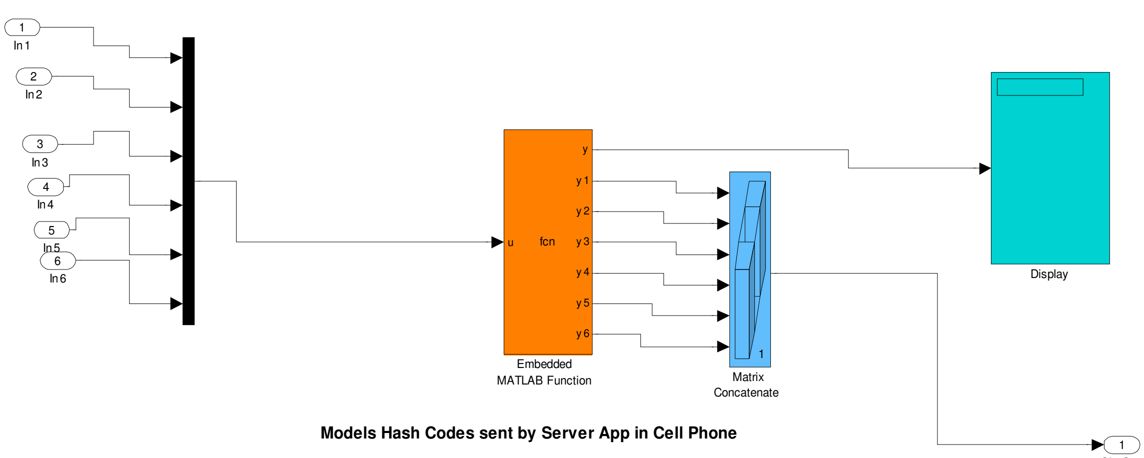

Fig. 4: Switch select subsystem

Figure 5 shows the transmission subsystem. In this subsystem the hash codes for sending the control signal to the smart system are encoded. The hash codes are coded in an embedded matlab function block concatenated using a matrix concatenated block and sent to the main model through a signal output port. Control signals are typically received from import blocks In1 to In6 which in turn receives its signals from manual switches (1-6) modeled as user selectable input. The work of this subsystem is to ensure that signals that are sent from the receiver are carried over to the trigger subsystem.

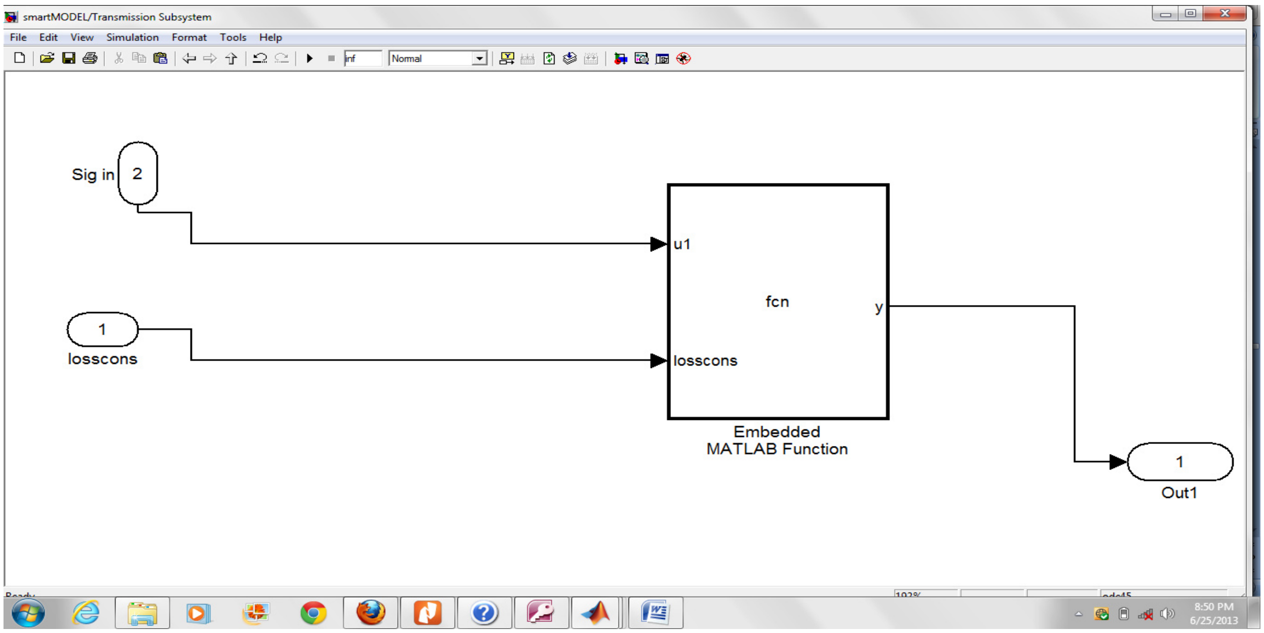

Fig. 5: Transmission subsystem

Figure 6 shows the delay subsystem. As the name implies, the delay subsystem models network delays. Here, messages that are sent are delayed 10 seconds before being mixed with the message signals.

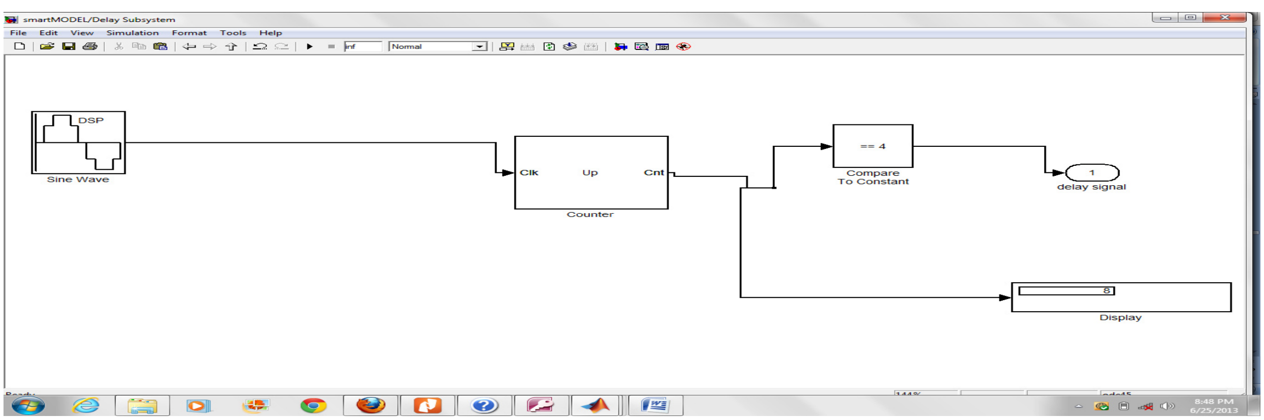

Fig. 6: Delay subsystem

Figure 7 shows the SMS Module Subsystem which receives the transmitted signal. This subsystem does the actual decoding of the transmitted messages (encoded hash codes) to their binary equivalent. This is handled by an embedded Matlab function block with the decoding logic embedded in it. A NOT logical operator block is used to carry out logic signal inversion of the binary code. The decimal conversion takes place at the transmission subsystem.

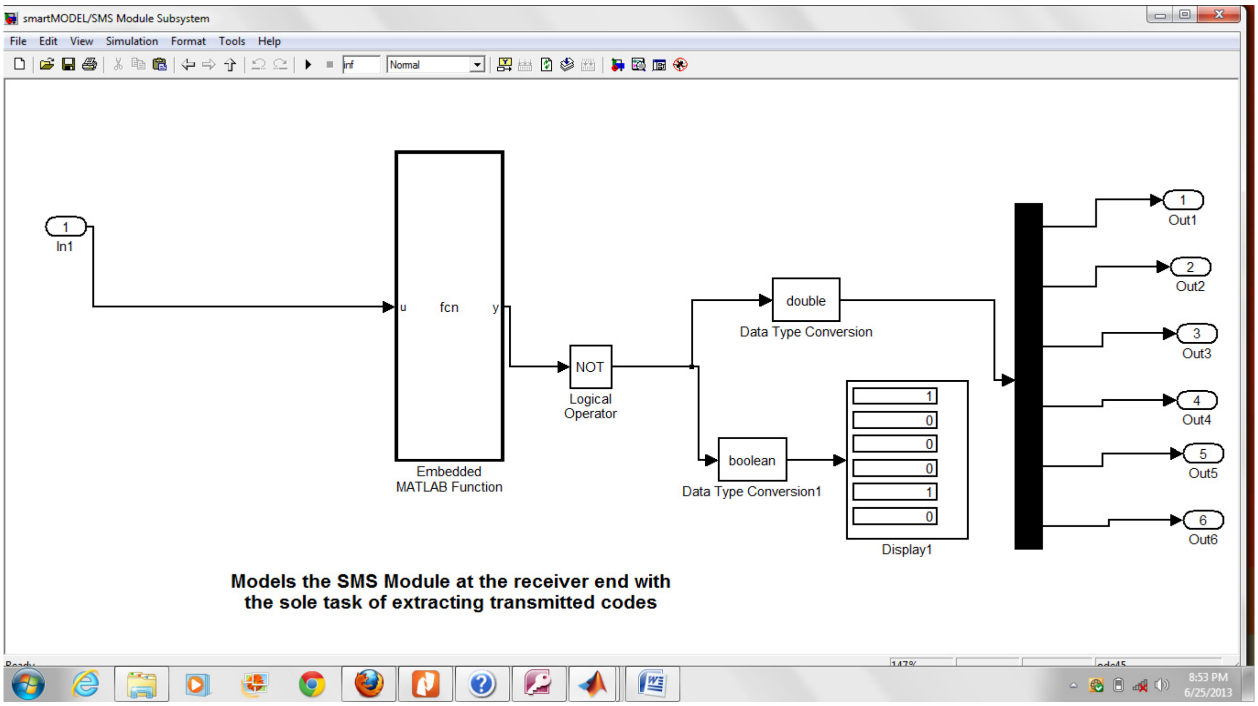

Fig. 7: SMS Module Subsystem

5. IMPLEMENTATION OF THE SYSTEM

The Programming languages chosen for the development of the system include Java for the mobile side and ARDUINO C/C++ for the microprocessor programming. Java is chosen as the programming technology for this work on the mobile because of its vast capability in mobile application development. ARDUINO C/C++ is a language closer to the machine and is ideal for semi-low level programming of processors. The 'jar' file format of the mobile application created using J2ME platform was distributed via Bluetooth for installation into the Nokia mobile phone (Java enabled) used for this research. Mobile devices interact with a MIDlet using their own software called Application Management Software (AMS), whether by emulators or real contact. Once installed, the user can instruct his smart home by typing simple code using the user friendly graphical user interface.

6. CONCLUSION

Embedded technology has made computing easy and enjoyable by providing real-time systems that support home automation. It is the technology that supports the proliferation of smart spaces which offer several benefits to the end-user such as mobile computing, task-driven programming, tracking of the homes and offices. These are the central ideas which this project is concerned with. This work aims at addressing the problem of energy wastage and domestic accidents (e.g. fire outbreak) by enabling end-users to easily use their mobile devices to monitor and instruct their home devices from any location over a wireless network. Mobile phones are considered ideal for the universal remote control of home devices since they are owned by virtually everyone.

REFERENCES

[1] Weiser, M. (1991), "The computer for the 21st Century" http://www.ubiq.com/hypertext/weiser/sciAmDraftz.html.
[2] Cook, D.; Das, S. (2005), Smart Environments: Technology, Protocols and Applications. Wiley-Interscience. ISBN 0-471-5448-5.
[3] Stong-Michas, J. L. (2005), http://www.ecmag.com/section/your-business/pervasive- networking.
[4] Delgado, A. R., Picking, R., Grout, V. (2006), Remote-Controlled Home Automation Systems with Different Network Technologies. Centre for Applied Internet Research (CAIR), University of Wales, NEWI, Wrexham, UK.
[5] Shoewu, O. & Olatinwo, O. (2013). Empirical Design and Implementation of a Microcontroller Based Automatic Gate. Afr J. of Comp & ICTs. Vol 6, No. 1. pp 21-32
[6] Oke, A. O., Emuoyibofarhe, J. O., Adetunji A. B. (2013), Development of a GSM based Control System for Electrical Appliances, International Journal of Engineering and Technology Volume 3 No. 4, April, 2013, IJET Publications UK.
[7] James, K. I. A., Sathik, M. J., Praveen, M. (2011), GSM Based Smart Data Logger System, International Journal of Scientific & Engineering Research, Volume 2, Issue 11, November-2011 1 ISSN 2229-5518 IJSER © 2011 http://www .ijser.org
[8] Ali, Z. A., Umit, B., (2005), An Internet Based Wireless Home Automation System for Multifunctional Devices, PScripts5.dll Version 5.2.2, Member IEEE.
[9] Khiyal, M. S. H., Khan, A. and Shehzadi, E. (2009), SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security. Issues in Informing Science and Information Technology, volume 6, page 887-894, Pakistan.
[10] Barbacci, M. R., Boubleday, D. L. and Weinstock, C. B., (1990). Application-Level Programming, Proceedings of the Tenth International Conference on Distributed Computing Systems.
[11] Meyers, B., Herb S. and Robert, G. (1998). Collaboration Using Multiple PDAs Connected to a PC, Proceedings of CSCW 98: ACM Conferences on Computer-Supported Cooperative Work. Nov. 1998.
[12] Kalofonos, N. D. and Reynolds, D. F. (2006), Task-Driven End-User Programming of Smart Spaces Using Mobile Devices, Nokia Research Center Cambridge, January 10, 2006.
[13] Song, Z., Labrou, Y. and Masuoka, R. (2004), Dynamic services recovery and management in task computing, pp. 310-318, mobiQuitous 2004, August 22-26, Boston, MA, 2004.