Controlling Remote System using Mobile Telephony

Rifat Shahriyar
Bangladesh University of Engineering and Technology, Bangladesh
rifat@cse.buet.ac.bd

Enamul Hoque
Bangladesh University of Engineering and Technology, Bangladesh
enamulhoque1@yahoo.com

Iftekhair Naim
Bangladesh University of Engineering and Technology, Bangladesh
chayancse@yahoo.com

S M Sohan
Bangladesh University of Engineering and Technology, Bangladesh
sohan39@gmail.com

Mostofa Akbar
Bangladesh University of Engineering and Technology, Bangladesh
mostofa@cse.buet.ac.bd

ABSTRACT

In modern days, we have to be in touch with various high-tech machineries and equipments to get our jobs done and make our lives easier. Too often these machineries serve very important purposes and sometimes require continuous monitoring. But it's not always feasible to be physically near to the system. So, to be in touch with this sort of important systems by not being physically close, we need some sort of remote solution. The remote solution should unleash the restrictions due to physical distance yet provides enough reliability even from distance. Some products are commercially available which allow remote systems controlling through internet which is undoubtedly emerging; yet, it lacks the true sense of real mobility and security, making the remote system controlling a limited term than it is supposed to be. In search of true a remote and adequately secure solution to be really effective and practicable, which can be a better choice to mobile telephony? Mobile phones have become almost an inseparable part of civil lives today. In this paper we introduce the mechanism so that the ordinary services of the mobile phones can be leveraged to communicate with and control the remote systems.

Categories and Subject Descriptors

J.7 [**Computer Applications**]: Computers in Other Systems – *command and control, consumer products, process control.*

Terms

Design

Keywords

Remote System, Mobile Telephony, X10 Active Home Controller Pro, Java 2 Micro Edition (J2ME).

1. INTRODUCTION

Definitions of a Remote System - We consider the 'Sys-tem' to be a concept that encapsulates a bunch of machines and instruments grouped to perform some tasks. An example of such a system is a cell phone tower or a manufacturing house or even a home or anything that is equipped with electrically or electronically controlled devices. We define 'Remote' to be a place anywhere on the earth. So, conceptually remote controlling a system is global remote controlling that is usable for any sort of control involving machines and instruments.

Motivation Behind Controlling Remote System - The motivations behind the goal to control such remote systems are simple. It's not always feasible to be physically near to the system setup and still sometimes it's very important to control it. For an example, a smart remote controller may enable us to track the surveillance from anywhere. So, it takes the control of the home beyond the home and to the hands of the people. And if a simple mobile phone takes the added responsibility to control the remote system, the control is reachable from almost everywhere people travels and lives on earth. This sort of high end technology is supposed to facilitate the different life easing utilities to a new age and bringing things out of the box to as near as one's palm.

Available Media

Internet - Of all the available media, Internet is a good example of the remote communication. Internet places virtually no bounds on geographical placement and is thus considered 'enough' remote by our definition. But internet is a place crowded with various types of traffics, often hostile to each other. Security vulnerability is the most striking alert point of the internet. Whenever a web based application goes live, a lot of efforts have to take place before it can be said to be secured, if at all. When we say remote control, we want to make sure no malicious party ever gains control and abolishes everything. Also to use web, it re-quires resources like flawless internet connections and hosting servers, which may not always fit to the concept of re-mote controlling systems.

Mobile Telephony - Another candidate solution to this problem is the use of mobile telephony. Mobile telephony offers a wide range of communication services like voice and data transfer through SMS and other enhanced data transfer protocols like GPRS, EDGE [1] at a relatively low price and at a wide variety of places on earth. In the mean time, the security is better achieved by the use of strict traffic control. We adhered to this method for our approach to the remote system controlling because of its unparallel availability and modest security at the affordable price.

Scope - The main goal of this project is to find out the feasible ways to leverage the mobile telephony using the existing services but redefining the trivial purposes they serve. So, we investigated the different ways we could use the cell phones to go beyond making calls and sending SMS for taking care of personal interests and devised some ways to implement the remote control, which is 'Remote' and conceptually can be used at any System. Conceptually it is a generic solution for controlling remote systems. To make it a reality and also to demonstrate its power we started with the simple task of controlling a home remotely using

MOBILWARE 2008, February 13-15, Innsbruck, Austria

DOI 10.4108/ICST.MOBILWARE2008.2732

mobile telephony. A home is also a System in the context that there is a setup of various electrical and electronic machines and instruments. Although the televisions and air conditioners come with some sort of small range remote controllers, we look beyond the small range and take it to any range indeed. So, although the water motor or the infrared security system doesn't come with such a remote controller, it's perfectly possible to leverage the technology of mobile telephony to extend it. And also we don't need to carry one remote controller for each of the devices we wish to control; rather we can do this just by using our mobile phone, which we generally carry anyway. Even more, we don't have to subscribe to any other services like internet or whatsoever beyond the normal services we get from our traditional mobile phones, which makes it very handy indeed. It's better to leverage the available home controllers than to devise one from scratch to aid this goal. Because there are standardized home controllers in the market and they offer wide coverage of controllable appliances. X10 is one of the best candidates in the home controller manufacturers [2]. We left the home appliances controlling part to the X10 and concentrated on the communication between the mobile phone and the X10 controller.

This paper is organized as follows. Section 2 de-scribes the preliminaries. Section 3 is prototype demonstration of the project and Section 4 and 5 gives the description of the various solutions we devised .The future expansions possibilities are in section 6. Finally, section 7 gives the conclusion.

2. PRELIMINARIES

This section introduces the mostly used terms and provides basic backgrounds of the participants in the various solutions to the remote system controlling problem which we devised and implemented.

Mobile Telephony - In telecommunication, telephony en-compasses the general use of equipment to provide voice communication over distances, specifically by connecting telephones to each other. The term mobile telephony is derived from original telephony to denote the communication that facilitates mobility using wireless technology. Mobile telephony offers services like voice and data transfer. Data transfer is done using SMS and also some other enhanced data rate services like GPRS and EDGE [1] pro-vides internet access facilities to the mobile phones. Short Message Service (SMS) is a telecommunications protocol that allows the sending of short (160 characters or less) text messages. It is available on most digital mobile phones and some personal digital assistants with onboard wireless telecommunications. Devices such as computer and microcontroller which can connect to mobile phones and PDAs through protocols such as Bluetooth and AT command can also sometimes send SMS messages using the mobile phones. The usability of this type of SMS can be used to leverage for the controller we wish to design.

X10 Active Home Controller Pro - X10 controller comes as a package that has one controller module and other appliance modules categorized by their classes like lights, fans and so on. The controller connects to a computer using USB connectivity. It can be instructed using the provided software from computer and also from the remote controller that comes with the package. An appliance specific module or generic module is plugged in between the controller and the appliances. The controller directly impacts the modules and not the appliances attached with the modules. So, the devices are completely unaware of the presence of the X10 home controlling and this is why X10 doesn't limit its operations to some specific vendors. X10 controller uses the power line to send and receive commands to the modules. This signal is passed using a bandwidth that doesn't interfere with the existing power connections. To control a specific appliance of many so connected, X10 uses an addressing mechanism to detect the desired one. This addressing is set in the appliance modules prior to connecting it and can be changed at anytime. Whenever X10 controller has to send some commands, it broadcasts the command to the power line. The command contains the address of the device that is intended to control. So, the module that has an address matching with the address in the command, responds immediately. This way, it handles a specific request issued by the controller to control an appliance. [2]

Java Specification Request (JSR) - Java Specification Requests (JSR) are the actual descriptions of proposed and final specifications for the Java platform. At any one time there are numerous JSRs moving through the review and approval process. JSR - 82 one of the JSRs and used for programming Bluetooth in java [3].

Java 2 Micro Edition (J2ME) - Java Platform, Micro Edition (Java ME) is the most ubiquitous application platform for mobile devices across the globe. It provides a robust, flexible environment for applications running on a broad range of other embedded devices, such as mobile phones, PDAs, TV set-top boxes, and printers. The Java ME plat-form includes flexible user interfaces, a robust security model, a broad range of built-in network protocols, and extensive support for networked and offline applications that can be downloaded dynamically. Applications based on Java ME software are portable across a wide range of de-vices, yet leveraging each device's native capabilities.

Java Native Interface (JNI) - The Java Native Interface (JNI) is the native programming interface for Java that is part of the JDK. By writing programs using the JNI, a code becomes completely portable across all platforms. The JNI allows Java code that runs within a Java Virtual Machine (VM) to operate with applications and libraries written in other languages, such as C, C++, and assembly. Programmers use the JNI to write native methods to handle those situations when an application cannot be written entirely in the Java programming language [4].

3. DEMONSTRATING PROTOTYPE

We implemented a prototype of the discussed remote system controlling project. Our implementation controls the home appliances of a home using mobile phones from anywhere. We also implemented various ways to communicate between the mobile phone and the computer. X10 Active Home Pro is used to take care of the real control mechanism once fed enough input to it. Using this set of devices, we concentrated on the communication aspect and came up with various solutions.

Brief Comparison of Different Alternative Solutions - The web based solution requires a web server hosting and internet connection at both the mobile phone and the home computer. Given that this specification is met, the home computer is responsible to forward the commands after formatting to the X10

controller [5] [6]. We do not stress one to adhere to this solution because the limiting factor of availability of internet is the obstacle to be called it truly remote. The FBUS based solution [7] incurs the extra coding complexity and also clumsy connection to the home mobile. The setup requires the home mobile to be stationed at home and connected to the computer. The controlling is done by the remote mobile. The unavailability of FBUS protocol details and strong dependency on Nokia mobile phones, it gets progressively harder to make robust pro-grams. Bluetooth based solution is a very strong candidate to be judged the best among all. Bluetooth offers full bidirectional communication between the computer and the mobile. And also, a standard client-server based programming model eases the task. The controlling can be done from both the home and remote mobiles and also the home mobile is somewhat mobile in the sense that Bluetooth [8] offers a small range of mobility. Also Bluetooth works for any mobile phone with Bluetooth API irrespective of the vendor. The only restricting factor with this solution is, it requires human intervention for custom made applications in the mobile for sending and receiving data to and from outside resources. Provided that, this step is avoidable, it seems very logical to make use of this solution. We pro-posed the AT command based solution [9] for its simplicity and ease of use. It doesn't incur any extra cost and it's available with all the mobile phones. The connection has no restrictions for the home mobile as it can be connected using any of the ways available like Bluetooth, Infrared and data cables. It also alleviates the necessity of any human intervention. The programming is concerned only with the computer and nothing about the home mobile, which makes the task simpler.

Description of the Prototype - The prototype implementation involves two mobile phones, one computer and X10 Active Home Pro system as hardware components. The software facilitating the whole communication is developed using programming languages Java Standard Edition (J2SE) and Micro Edition (J2ME) and also a C program is used from the Java using JNI (Java Native Interface) [4].The components are described below:

Remote Mobile - The remote mobile is free to operate in either of the two ways. The preferred one is for the Java enabled mobile phones. We developed a J2ME application for the remote mobile. The application lets one to assign human friendly set of names instead of the default naming of the X10 modules [10]. The user is free to use either the assigned names or the X10 specific names using a simple graphical user interface from their mobiles. For the rest of the mobiles that doesn't come with a support for Java the user is asked to send a Home Control Message to control the home appliances. The remote mobile is totally vendor independent and the capability of sending and receiving SMS is the only necessary requirement for it.

Home Mobile - The home mobile is connected to the computer using a standard data cable. The computer communicates with the mobile using AT command.

Home Computer - The computer in our prototype uses the AT command protocol to determine if a new Home Control Message is received in the home mobile. The prototype software is developed in windows based platform but can easily be migrated to other OS like Unix/Linux and so on. Upon recognizing and authenticating a home control message the computer makes a look up to the mapping between the human readable names and the X10 specific codes. Once the machine key corresponding to the X10 device is formed the message is sent to the specific USB port which connects to the X10 controller.

X10 Home Controller Package - X10 is a communications protocol, similar to network protocols such as TCP/IP. However, X10 works across home power lines. Like a broadcast network, every command is sent through every wire in the house; it's up to each individual device to decide whether it needs to respond to a particular command. With X10, a wide variety of devices can talk to each other and also with the computer. CM15A, a 2-way PC interface is at the other end of the communication hop from the computer. Based on the sent Home Control Message, we send the corresponding X10 code to CM 15A. Also we send the address of the target device. CM15A then broadcasts this message through the power line. All the attached X10 de-vices receive the message, but only the target device, that is specified in the message responds.

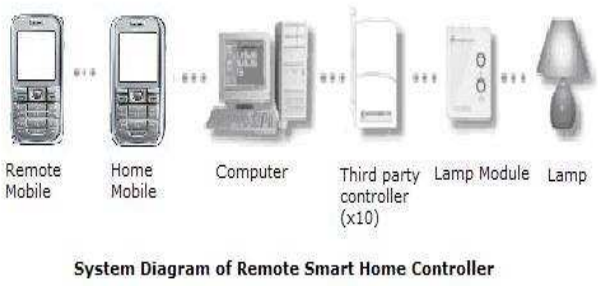

Figure 1. Prototype System Diagram.

The constraint applied here is, X10 messages are only valid if there is no trans-former between the controller and the receiving module. A Lamp Module LM465 and an Appliance Module AM486 is used in our prototype to control the lights and other generic appliances like hitters and so on. But this is not a restrictive feature, as addition of other modules is just only a matter of plugging into the home power line. The controller software module wraps up the lower level of the communication of the computer with the CM15A through USB. The software module presents an interface which enables the client of this module to transfer the raw data without worrying about the details about the low-level protocol implementation. Once this message is sent, the connected CM15A receives the data and broadcasts it over the power line using a designated frequency. The lamp module (also other such modules) is pre-tuned to act upon receiving these control signals. CM15A receives these acknowledgment signals from power line and forwards it to PC through USB. Finally, the wrapper program confirms the successful completion of the issued command. For any queries on X10 see [11].

4. AT COMMAND BASED SOLUTION

AT commands [9] provide the computer with the most flexible way to control and explore the services and resources of a mobile. AT commands enable one to send and receive SMS from the computer and also it lets the computer to browse the mobiles

resources like memory and phone book and so on. AT Stands for 'Attention' command. GSM mobile phones are equipped with built in GSM Modems which responds to the commands issued as an SMS by the connected computer. AT commands create a logical bidirectional communication between computer and mobile phone.

Format of AT Commands - The commands are similar to any other commands we use in various terminals. The commands conform to a well defined syntax. To start with, it is not case sensitive. The common prefix is AT for all the commands. Multiple commands can be specified in a single line starting with only one 'AT' as the common prefix. The symbols < > and [ ] denotes compulsory and optional set-tings values respectively. When the computer issues a command, the effect retains its value until it's changed explicitly. The connection test is performed by first sending the command 'AT' and if the mobile is correctly connected, it notifies the computer through the message 'OK'. Other-wise it sends an error message.

SMS Related AT Commands - The sequence of commands and responses are given below:

Sending SMS

PC: AT

Mobile: OK

PC: AT+CPIN?

Mobile: +CPIN: <code>

PC: AT+CMGF=1 [For Text Mode]

Mobile: OK

PC: AT+CSCA="<smsc number>"

Mobile: OK

PC: AT+CMGS="<destination phone number>"

Mobile: >

PC: <SMS text>Ctrl-Z [SMS text must be terminated with Ctrl-Z (ASCII 26)]

Mobile: OK

Receiving SMS

PC: AT

Mobile: OK

PC: AT+CPIN?

Mobile: +CPIN: <code>

PC: AT+CMGF=1 [For Text Mode]

Mobile: OK

PC: AT+CNMI=1, 2,0,0,0

Mobile: OK

After this mobile will wait for any incoming SMS and upon receiving directing it towards the PC.

Mobile Connectivity - The mobile can be connected using standard data cable or Bluetooth for communication using AT Commands. However, after connecting, it normally binds the mobile with one of the virtual COM ports. So, the communication virtually becomes serial data communication behind the scene irrespective of the physical connection used. The commands presented above can be tested in the computer HyperTerminal to verify the sequence of commands and their syntax. Only if the commands are verified, then it is suggested to port to programming codes.

Advantages - The advantages of using the AT Command based solution are give below:

1. The communication is solely implemented using SMS protocol which is available at most of the places.

2. It doesn't rely on the internet and web servers, which cuts down the overhead and the cost of the communication.

3. There is no binding about the physical connection link that is used between the computers and mobile, which makes it more versatile and interoperable with data cables, Bluetooth and Infrared.

5. BLUETOOTH BASED SOLUTION

Bluetooth wireless technology [8] stands in the way of traditional short-range wired communications technology connecting portable and/or fixed devices while maintaining high levels of security. It obsoletes wires between your workstation, mouse, laptop computer, music head-phones etc. The key features of Bluetooth technology are robustness, low power yet low cost. The Bluetooth specification sets the standard for a wide range of devices to connect and communicate with each other. A fundamental strength of Bluetooth wireless technology is the ability to simultaneously handle both data and voice transmissions, making it most feasible for handheld devices.

Understanding the Bluetooth Communication: Bluetooth devices must have the ability to discover nearby Bluetooth devices. When a new Bluetooth device is discovered, a service discovery may be initiated in order to determine which services the device is offering. The Bluetooth Specification refers to the device discovery operation as inquiry. During the inquiry process the inquiring Bluetooth device will receive the Bluetooth address and clock from nearby discoverable devices. The inquiring device then identifies the other devices by their Bluetooth address and is also able to synchronize the frequency hopping with discovered de-vices, using their Bluetooth address and clock. Devices make themselves discoverable by entering the inquiry scan mode. Discoverable devices make use of an Inquiry Access Code (IAC) for device discovery. The General Inquiry Access Code (GIAC) and the Limited Inquiry Access Code (LIAC) are the types of Inquiry Access Code. The GIAC is used when a device is general discoverable implying it will be discoverable for an undefined period of time. The LIAC is used when a device will be discoverable for only a limited period of time. Different Bluetooth devices offer different sets of services. Hence, a Bluetooth device needs to do a service discovery on a remote device in order to obtain information about available services. Service searches can be of a general nature by polling a device for all available services, but can also be narrowed down to find just a single service. The service discovery process uses the Service Discovery

Protocol (SDP). A SDP client must issue SDP requests to a SDP server to retrieve information from the server's service records. Bluetooth devices keep information about their Bluetooth services in a Service Discovery Database (SDDB). The SDDB contains service record en-tries, where each service record contains attributes describing a particular service. Each service has its own entry in the SDDB. Remote devices can retrieve service records during service discovery and possesses all information required to use the services described. The Universally Unique Identifier (UUID) is used for identifying services, protocols and profiles. A UUID is a 128-bit identifier that is guaranteed to be unique across all time and space. A range of UUID values has been pre-allocated for often-used ser-vices, protocols and profiles and is listed in the Bluetooth Assigned Numbers document on the Bluetooth Membership website. Generic Access Profile (GAP) The basis for all profiles in the Bluetooth system. The GAP defines basic Bluetooth functionality like setting up L2CAP links, handling security modes and discoverable modes and primitive data transfers. That's why it falls into our area of interest. Authentication is performed using bonding and pairing .Bonding is the procedure of a Bluetooth device authenticating another Bluetooth device and is dependent on a shared authentication key. If the devices do not share an authentication key, a new key must be created before the bonding process can complete. Generation of the authentication key is called pairing. The pairing process involves generation of an initialization key and an authentication key, followed by mutual authentication. The initialization key is based on user input, a random number and the Blue-tooth address of one of the devices. The user input is referred to as a Personal Identification Number (PIN) or passkey and may be up to 128-bits long. The passkey is the shared secret between the two devices. The authentication key is based on random numbers and Bluetooth addresses from both devices. The initialization key is used for encryption when exchanging data to create the authentication key, and is thereafter discarded. When the pairing process is completed, the device authenticates each other. Both de-vices share the same authentication key, often called a combination key since both devices have contributed to the creation of the key. When two devices have completed the pairing process they may store the authentication key for future use. The devices are then paired and may authenticate each other through the bonding process without the use of a passkey. Devices will stay paired until one device requests a new pairing process, or the authentication key is deleted on either of the devices. Authorization is the process of giving a remote Bluetooth device permission to access a particular service. In order to be authorized the re-mote device must first be authenticated through the bonding process. Access may then be granted on a temporary or a permanent basis. The trust attribute is related to authorization, linking authorization permissions to a particular de-vice. A trusted device may connect to a Bluetooth service, and the authorization process will complete successfully without user interaction.

Mobile Connectivity: We need to build a mobile Bluetooth Client application and Computer Bluetooth server application in order to establish a bidirectional communication between the two. We choose JAVA for the development of both client and server, as J2ME is the most used language platform for mobile applications development. Java pro-vides a Bluetooth Application Programmer Interface (API) which is known as JSR-82. It is also known as JABWT (Java APIs for Bluetooth Wireless Technology).The basic concepts of any Bluetooth application (Java or otherwise) consist of the Stack Initialization, Device Discovery, De-vice Management, Service Discovery and Communication.

Stack Initialization: At the very beginning it is required to initialize the Bluetooth stack. The stack is the piece of soft-ware (or firmware) that controls the Bluetooth device. Stack initialization can consist of a number of things, but its main purpose is to get the Bluetooth device ready to start wireless communication. Every vendor handles stack initialization differently. The basic functions for stack initialization are getLocalDevice (), setDiscoverable (), getDiscoveryAgent ().

Device Management: LocalDevice and RemoteDevice are the two main classes in the Java Bluetooth Specification that allow performing Device Management. These classes give the ability to query statistical information about the local Bluetooth device (LocalDevice) and information on the devices in the remote area (RemoteDevice). The static method LocalDevice.getLocalDevice () returns an instantiated LocalDevice object to use. In order to get the unique address of the Bluetooth radio, one needs to call getBluetoothAddress () on the local device object. The Bluetooth address serves the same purpose of the MAC address on the network card of computer. Every Bluetooth device has a unique address. If a device wants other Bluetooth devices in the area to find it, then it needs to call the setDiscoverable () method in LocalDevice object. In a nutshell, that's about all it takes to perform Device Management with the Java Bluetooth Specification APIs.

Device Discovery: Bluetooth device has no idea of what other Bluetooth devices are in the area. Perhaps there are laptops, desktops, printers, mobile phones, or PDAs in the area. In order to find out, Bluetooth device will use the Device Discovery classes that are provided into the Java Bluetooth API in order to see what's out there. The two classes responsible for Bluetooth device to discover remote Bluetooth devices in the area are DiscoveryAgent and DiscoveryListener. After getting a LocalDevice object by getLocalDevice () function, it needs just to instantiate a DiscoveryAgent by calling LocalDevice.getDiscoveryAgent (). First, the object must implement the DiscoveryListener interface. This interface works like any listener, so it'll notify when an event happens. In this case, it will notify when Bluetooth devices are in the area. In order to start the discovery process, it needs to call the startInquiry () method on DiscoveryAgent. This method is non-blocking, so one is free to do other things while you wait for other Bluetooth devices to be found. When a Bluetooth device is found, the JVM will call the deviceDiscovered () method of the class that implemented the DiscoveryListener interface. This method will pass a RemoteDevice object that represents the device discovered by the inquiry.

Service Discovery: After finding the other Bluetooth de-vices it would be necessary to see what services that those devices offer. One can never be sure what services a RemoteDevice may offer. Service Discovery allows finding out what they are. Service Discovery is just like Device Discovery in the sense that it use the DiscoveryAgent to do the"discovering." The searchServices () method of the DiscoveryAgent class allows to search for services on a RemoteDevice. When services are found, the servicesDiscovered () will be called by the JVM if the object implemented the DiscoveryListener interface. This callback

method also passes in a ServiceRecordobject that pertains to the service for which one searched. With a ServiceRecord in hand, one can do plenty of things, but it would be most likely to connect to the RemoteDevice where this ServiceRecord originated.

Service Registration: Before a Bluetooth client device can use the Service Discovery on a Bluetooth server device, the Bluetooth server needs to register its services internally in the Service Discovery database (SDDB). That process is called Service Registration. In a peer-to-peer application, such as a file transfer or chat application, one should re-member that any device can act as the client or the server, so one needs to incorporate that functionality (both client and server) into the code in order to handle both scenarios of Service Discovery (i.e., the client) and Service Registration (i.e., the server). Here's a scenario of what's involved to get the service registered and stored in the SDDB. First, Call Connector.open () and cast the resulting Connection to a StreamConnectionNotifier. Connector.open () creates a new ServiceRecord and sets some attributes. Second, Use the LocalDevice object and the StreamConnectionNotifier to obtain the ServiceRecord that was created by the system. Third, Add or modify the attributes in the ServiceRecord (optional).Fourth, Use the StreamConnectionNotifier and call acceptAndOpen () and wait for Bluetooth clients to discover this service and connect. The system creates a service record in the SDDB. Fifth, wait until a client connects. Sixth, Then the server is ready to exit, call close () on the StreamConnectionNotifier. The system removes the service record from the SDDB. StreamConnectionNotifier and Connector both come from the javax.microedition.io package of the J2ME platform. That's all one needs to do Service Registration in Bluetooth.

Communication: Bluetooth is a communication protocol. The Java Bluetooth API gives three ways to send and receive data. In this document we cover one of them named RFCOMM. RFCOMM is the protocol layer that the Serial Port Profile uses in order to communicate, but these two items are almost always used synonymously. For server side, the communication needs a unique UIUD.The URL is <btspp: //localhost :>+< UIUD>+<name=appName> where appName is the name of the application chosen by the user. A StreamConnectionNotifier and StreamConnection are needed. We need to cast the Connector.open (URL) to the instance of StreamConnectionNotifier. The acceptAndOpen () method of StreamConnectionNotifier returns an instance of StreamConnection which is needed for the communication. The StreamConnection contains openDataInputStream () and openDataOutputStream () functions which are used to open the streams. The readUTF () and writeUTF () functions of the StreamConnection are used for reading and writing. For client side, the URL is obtained to connect to the device from the ServiceRecord object that one gets from Service Discovery. The getConnectionURL ( ) method of the Service Record returns the URL. The open (URL) method of Connector returns an instance of StreamConnection which is needed for the communication. The Stream-Connection contains openDataInputStream () and open-DataOutputStream () functions which are used to open the streams. The readUTF () and writeUTF () functions of the StreamConnection are used for reading and writing. For any queries on Bluetooth programming see [12] [13].

Advantages: The advantages of using the Bluetooth based solution are given below:

1. Bluetooth offers a global standard for connecting a wide range of devices with different services. So, using a Blue-tooth protocol at the bottom makes one application some-what versatile in terms of interoperability.

2. Bluetooth is available at most of the handheld devices like cell phones, music players and cameras all conforming to the defined standard. So, to use the Bluetooth with the computer, one need not think about overheads like internet and web servers.

3. The technology is very easy to use. The devices generally come with built-in software support for Bluetooth operations and these are most commonly used applications for handheld devices.

4. The most desirable property for a communication protocol is the ability to measure the security support it provides. And Bluetooth restricts the malicious attacks by using the 128-bit long shared keys and once securely started, maintains it until the otherwise stated.

6. FUTURE EXPANSIONS

Looking at the devised solution, one keen reader would readily point that, the computer is a bit of overhead. Computer is a device with enormous capability and our application fails to make most of it. We tried to replace the computer with a micro-controller. A micro-controller is a small IC with a micro-processor, memory and IO support, which is comparatively very cheap and requires fewer resources. Micro-controllers come with built-in support for USB and Bluetooth connectivity with traditional serial ports and also programming support for these ports. Micro-controllers come with custom SDKs and can be used to serve our goal with reduced efforts. We studied theoretically with micro-controllers and it seemed possible to be implemented in reality. Given enough time and wise thinking, one can proceed with this application and move the computer part to the micro-controller. This modification will highly add value to this project by expanding its usability to a great extent. This can create a whole new dimension of remote controlling it take the control as near as to one's palm for always and everywhere. Moreover the wireless networking protocols such as IEEE 802.11 [14] and ZigBee [15] can be used for the wireless controlling. By deploying wireless sensors inside the home smart home can be made smarter. Recent works on smart home is governed by the concepts of wireless sensors. Remote systems can be controlled in a more flexible ways by integrating wireless sensors to the system.

7. CONCLUSION

We implemented a prototype of the remote system control-ling. Although the implementation is just a home controller using mobile telephony, it may be a guide to do more than just this. Conceptually we provided the solution to send and receive data between a mobile phone and a computer which can be the most important part of the remote system con-trolling. Once there is a way to control devices from the computer, all we need to do is an application that bridges between our application and the controller application. This way, it can be stated that, this solution is usable with a wide range of remote controlling systems with a bit of tuning to the particular context where it is applied. Our solution is somewhat unique in the sense that, commercial products

facilitating such a concept come up with internet based solution. And it's evident that, there are scopes to improve these existing solutions and also it may be a breakthrough innovation. We stepped in with this view that, a mobile telephony based solution leveraging the power of the real mobility can create real scopes. So, we presented different ways to implement this solution with minimal and available resources using mobile phones for these ports. Micro-controllers come with custom SDKs and can be leveraged to serve our target goal with reduced efforts. We studied theoretically with micro-controllers and it seemed possible to be implemented in reality. Given enough time and wise thinking, one can proceed with this project and move the computer part to the micro-controller. This modification will highly add value to this project by expanding its usability to a great extent. The usability of the remote system controlling concept can be highly diversified beyond that of a home controlling. For an example, one can use it to monitor and control security systems. This can create a whole new dimension to the security systems used worldwide as it take the control as near as to one's palm for always and everywhere.

8. ACKNOWLEDGMENTS

This work is performed as a part of undergraduate thesis in the Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology (BUET), [16]. Our special thanks to BUET for providing such a good environment of research activities in the field of Computer Science and Engineering.

9. REFERENCES

[1] GSM Association, Mobile Telephony Services Description, http://www.gsmworld.com.

[2] X10.com, X10 Active Home Controller Pro Resources, http://www.x10.com.

[3] Sun Microsystems, JSR - Java Specification Request, http://jcp.org/en/jsr/overview.

[4] Sun Microsystems, JNI - Java Native Interface, http://java.sun.com/j2se/1.4.2/docs/guide/jni/.

[5] Kevin Boone, Using X10 for home automation, http://www.kevinboone.com/x10.html.

[6] Laser Business Systems Ltd, Home Automation Website – Laser Business Systems, http://www.laser.com.

[7] Wayne Peacock, Embedtronics - Nokia FBUS Protocol made simple, http://www.embedtronics.com/nokia/fbus.html.

[8] Bluetooth SIG, Bluetooth Technology, http://www.bluetooth.com.

[9] AT Commands Reference, http://nds1.nokia.com/phones/files/guides/Nokia_AThelp.pdf

[10] Dan Suther, X10 Protocol by Dan Suther, http://www.linuxha.com/athome/common/protocol.txt.

[11] X10.com, X10 Community Forums, http://www.x10community.com/forums/.

[12] Benhui.net, Java Bluetooth Programming help center, http://www.benhui.net/modules.php?name=Bluetooth.

[13] Dan Harkey, Shan Appajodu and Mike Larkin, Wireless Java Programming for Enterprise Applications.

[14] Institute of Electrical and Electronics Engineers, (IEEE), IEEE 802.11, http://www.ieee802.org/11/.

[15] ZigBee Alliance, ZigBee, http://www.zigbee.org.

[16] Rifat Shahriyar, Enamul Hoque, Iftekhar Naim and S M Sohan, Controlling Remote System using Mobile Telephony, Undergraduate Thesis, Department of Computer Science and Engineering, BUET, 2007.