

## Remote Controlling of Home Appliances using Mobile Telephony

Rifat Shahriyar<sup>1</sup>, Enamul Hoque<sup>2</sup>, S.M. Sohan<sup>3</sup>, Iftekhar Naim<sup>4</sup>,  
Md. Mostafa Akbar<sup>5</sup> & Masud Karim Khan<sup>6</sup>

<sup>1,2,3,4,5</sup>Department of Computer Science & Engineering, Bangladesh University of  
Engineering & Technology, <sup>6</sup>Windows Mobile Division at Microsoft  
rifat@cse.buet.ac.bd, enamulhoque1@yahoo.com, sohan39@gmail.com  
chayanmse@yahoo.com, mostofa@cse.buet.ac.bd, masud.khan@microsoft.com

### Abstract

*In modern days, we must use various high-tech machineries and equipments to get our jobs done and make the life easier. These machineries should be controlled by the homeowner from any location as the homeowner might be away from home at workplace or traveling in a different place in the weekend. Thus a system of remote monitoring and controlling are very much necessary. Smart home is one of these types of system equipped with home appliances which we wish to control smartly from anywhere. Some products are commercially available which allow remote home appliance controlling through internet which is undoubtedly emerging. But it lacks the true sense of real mobility and security, making the remote home appliance controlling a limited term than it is supposed to be. In search of a true remote and adequately secure solution to be really effective and practicable, mobile telephony is better than any other solutions. Mobile phones have become almost an inseparable part of civil lives today. In this paper we introduce a new mechanism so that the ordinary services of the mobile phones can be leveraged to communicate with and control the home appliances and make our homes a really smart one.*

### 1. Introduction

Smart home is a home equipped with special facilities to enable occupants to control or program an array of automated home electronic devices. For example, a homeowner on vacation can arm a home security system, control temperature gauges, switch appliances on or off, control lighting, program a home theater or entertainment system, and perform many other tasks. Smart home became smarter if the controlling can be done from any remote place. Our main focus is to control the home appliances from remote place.

The motivations behind the goal to remote control of home appliances are simple. It's not always feasible to be physically near to the home still sometimes it's very important to control the appliances for many purposes. So the remote controlling takes the control of the home beyond the home and to the hands of the people. If a simple mobile phone takes the added responsibility to control the smart home then 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.

There exists a number of available media for remote communication. Internet is a good example of this type of remote communication. Internet places virtually no bounds on geographical placement and is thus considered "enough" remote by our definition. But the

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 requires resources like flawless internet connections and hosting servers, which may not always fit to the concept of remote controlling.

Another candidate solution to this remote communication 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 at a relatively low price and at a wide variety of places on the earth. On the other hand, the security is better achieved by the use of strict traffic control. We adhered to this method of remote controlling of home appliances because of its unparallel availability and modest security at the affordable price.

In this research we produce different feasible ways to leverage the mobile telephony using the existing services but redefining the trivial purposes they serve. So, We have investigated the different ways we could use the cell phones to go beyond making calls and sending SMS and devised some ways to implement the remote control, which is 'Remote' and can be used to control the home appliances of a smart home . There are two approaches for controlling home appliances. One is to make a custom build controller from scratch for controlling the home appliances using wired connection. The main problem is that the connections of this controller become clumsy as well as not reliable. We find out it's better to use the available home controllers than to devise one from scratch to aid this goal as there are standardized home controllers in the market and they offer wide coverage of controllable appliances. X10, Insteon, Z-Wave and ZigBee are the available candidates for the home controller manufacturers. We left the home appliances controlling part to the X10 and concentrated on the communication between the mobile phone and the X10 controller for remote controlling of the X10 controller. We choose X10 over others due to its wide availability. This paper is organized as follows. Section 2 and 3 describes the backgrounds and related works respectively. Section 4 provides a detailed description of the X10 technology we used for prototype. Section 5 describes the prototype of the application we developed and its advantages over the existing solutions. Section 6 and 7 provides descriptions of two possible Medias (Bluetooth and AT Command) of our application. Finally Section 8 and 9 depicts on the future expansion possibilities and references.

## **2. Backgrounds**

Mobile Telephony: In telecommunication, telephony encompasses 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 [1] offers services like voice and data transfer. Data transfer is done using SMS and some other enhanced data rate services like GPRS and EDGE. The latter two provides 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 in the controller we wish to design.

X10 Active Home Controller Pro: X10 controller [2] 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 standard interfaces. 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. The details of X10 technology is provided later.

Insteon: Insteon technology [16] is a dual-band mesh topology employing ac-power lines and a radio-frequency (RF) protocol to communicate with and automate home electronic devices and appliances, which normally work independently. It is a home automation networking technology invented by Smart Labs, Inc. Insteon was developed, based on the X10 model, for control and sensing applications in the home. Insteon is designed to enable simple devices to be networked together using the power line and/or radio frequency (RF). All Insteon devices are peers, meaning each device can transmit, receive, and repeat any message of the Insteon protocol, without requiring a master controller or complex routing software.

Z-Wave: Z-Wave [17] is a next-generation wireless ecosystem that lets all the home electronics talk to each other via remote control. It uses simple, reliable, low-power radio waves that easily travel through walls, floors and cabinets. Z-Wave functionality can be added to almost any electronic devices. One can control Z-Wave household remotely from a PC and the Internet from anywhere in the world. Z-Wave unifies all home electronics into an integrated wireless network, with no complicated programming and no new cables to run. Any Z-Wave enabled device can be effortlessly added to this network, and many non-Z-Wave devices can be made compatible by simply plugging them into a Z-Wave accessory module. In seconds, device gets joined to the network and can communicate wirelessly with other Z-Wave modules and controllers. Because Z-Wave operates on its own unique frequency, it won't interfere with other wireless equipment in home, like cordless telephones and Wi-Fi routers. Z-Wave technology is extremely affordable, giving a powerful home control at a fraction of the cost of conventional smart home technologies.

**ZigBee:** ZigBee [15] is built on an Institute of Electrical and Electronics Engineers (IEEE) global standard, 802.15.4, similar to the standards that govern Bluetooth and Wi-Fi. Open standards encourage innovation and competition, which bring down costs. But unlike Bluetooth and Wi-Fi networks, which require central hubs that distribute information to dispersed devices, ZigBee allows devices to form mesh networks meaning that all the devices within a network are equipped with radios and can communicate with each other, rather than just with the controller (the device that serves as central command for the network). Homeowners' first taste of ZigBee is likely to come in the form of adaptors into which lamps, stereos, and other appliances can be plugged. The adaptors are activated by wall-mounted wireless switches or handheld devices that mean one could have his whole house on one remote control. ZigBee is expected to be the key of smart home in future.

### 3. Related Works

Smart home is one of the recent fields in the context of computer science. The paper named as "Remote mobile control of home appliances" by F. Meija, M. Nikolova and P. Voorwinden depicts on the home controlling using WAP protocol. The architecture mentioned by them is much complex but it gives an initial idea about the remote home appliance controlling. Smart home studies sometimes affected by the concern about the possible harms to the humans' health. A great research was done by Toril Laberg, Directorate for Health and Social Affairs of the Delta Centre, Norway. He later publish in his paper named "Smart Home Technology: Technology supporting independent living - does it have an impact on health?" that there is no harm on humans' health by the technical setup required to support smart home technologies. Scott Davidoff, Min Kyung Lee, Charles Yiu, John Zimmerman, and Anind K. Dey in their journal named "Principles of Smart Home Control" describe the control that families want and suggest seven design principles that will help end-user programming systems deliver that control. Tatsuya Yamazaki in his journal "The Ubiquitous Home" suggests that automation should not become a goal of the smart home technologies. In this paper he represents a real-life test bed, called the Ubiquitous Home. In the Ubiquitous Home, a robot plays a role of interface for the residents. Three kinds of context-aware services have been implemented and a real-life living experiment was conducted. The experimental results were also reported. Recently some projects are organized for building the architecture of controlling home appliance using voice commands. VoiceXML is used for that purpose. A smart house system named NETVOX [18] based on the ZigBee standard is introduced recently. The system can use for home automation and industrial controls. It provides security, temperature, humidity, lighting, sensor, and multimedia control for comfort, convenience, and safety wirelessly. The system may be accessed and controlled over the telephone or over the Internet.

### 4. X10 Technology

This section presents a detail description of the X10 technology. The X10 protocol and different kinds of X10 modules are described thoroughly.

**What is X10:** X10 allows the control of appliances and lights throughout the home. Signals are sent through existing wiring in the home between modules to switch devices on. Appliance modules are simple adaptors which plugs into a socket and then one can plug the device to be controlled into the socket on the module. The X10 controller then switches on the module and it allows power to pass thorough to the attached device.

The user can control the modules in several ways. The X10 controller unit is attached to an electrical socket in the home and this passes messages to the modules. An alternative way of sending message to the controller is using a computer or micro-controller to pass the messages to the unit.

How Does It Work: An X10 controller is told to carry out an action and it sends a control message through the wiring in the home and the module with the correct address picks up the message. Each module has an address that is set by the user. The address consists of a letter between A and P known as the House Code and a unit number between 1 and 16 known as Device Code. This gives a possible 256 addresses allowing 256 different modules to be controlled and therefore up to 256 different devices can be controlled in a single home. The control message can be one of the several types depending on the action to be carried out. The main messages are On, Off, Dim or Brighten. Other messages such as ‘All Units Off’ can also be sent, to which all modules will respond. Control signals are sent as 1 millisecond bursts at a frequency of 120 KHz. Each burst is sent three times to ensure that it will not be cancelled by the 50Hz frequency of the AC current in the power lines. The appliance module used to control the appliances contains an electro-mechanical relay switch. When switched on this allows current to pass through to the attached device. The lamp module used to control lights contains a TRIAC switch that controls the amount of current flowing through the switch allowing the attached light to be dimmed. Because of the electronics required to allow dimming of the light, lamp modules can be used with lights up to 300W in power. If an appliance is attached to a lamp module, damage can occur to both the appliance and lamp module because of the increased current. That’s why appliance module is used to control other appliances. However a lamp can be attached to an appliance module [5].

X10 Protocols

House Codes and Device Codes: As we mentioned earlier, house codes and device codes range from A to P and 1 to 16 respectively although they do not follow a binary sequence. The encoding format of these codes can be found in [10]. For example to select the device (having code 1) of the house (having code A) we need to chose binary value 0110 and to turn it on we need to chose binary value 0010.

Standard Transmission: An X10 transmission from the PC to the controller interface typically refers to the communication of a House code and a Device code combination or the transmission of a Function code. The format of these transmissions is

**Table 1. X10 Standard Transmission Sequence**

Bytes	Type	PC	Controller Interface
2 bytes	Header: Code	Send	Receive
1 byte	Checksum	Receive	Send
1 byte	Acknowledgment	Send	Receive
1 byte	Interface ready to receive signal	Receive	Send

This format is typical of all transmissions between the PC and the controller interface with the difference being in the first transmission from the PC.

Header: Code: The 1st byte of Header: Code is as follows

**Table 2. X10 Header: Code (1st byte)**

7	6	5	4	3	2	1	0
Dim Amount					1	F/A	E/S

Where, Dim Amount is a value between 0 and 22 identifying the number of dims to be transmitted (22 is equivalent to 100%). Bit 2 is always set to '1' to ensure that the interface is able to maintain synchronization. F/A defines whether the following byte is a function (1) or address (0). E/S defines whether the following byte is an extended transmission (1) or a standard transmission (0).

The 2nd byte of Header: Code is as follows

**Table 3. X10 Header: Code (2nd byte)**

7	6	5	4	3	2	1	0
House Code				Device Code/Function			

Device code is used when address is sent and Function is used for sending the command of a specific action. Note the function only operates for devices addressed with the same House code.

Interface Checksum and PC Acknowledgement: When the interface receives a transmission from the PC, it will sum all the bytes and then return a byte checksum. If the checksum is correct, the PC should return a value of 0x00 to indicate that the transmission should take place. If however the checksum is incorrect, then the PC should again attempt to transmit the same Header: Code combination and await a new checksum.

Interface Ready to Receive: Once the X10 transmission has taken place the interface will send 0x55 to the PC to indicate that it is in a 'ready' state.

X10 Receptions: Whenever the interface begins to receive data from the power line it will automatically assert the serial ring (RI) signal to initiate the wake up procedure for the PC. Once the data reception is complete, the interface will begin to poll the PC to upload its data buffer (maximum 10 bytes). If the PC doesn't respond then the interface's data buffer will overrun and additional data will not be stored within the buffer.

Interface Poll Signal: In order to Poll the PC the interface will continually send 0x5A. This signal will be repeated once every second until the PC responds.

PC Response to the Poll Signal: To terminate the interface's polling and initiate the data transfer, the PC must send an acknowledgment of 0XcC to the interface's poll signal.

Dim or Bright: After a dim or bright code, the PC will expect the following byte to be the change in brightness level. An X10 module has 201 discrete brightness levels. Therefore this byte will be equivalent to a brightness change of  $(n/210)*100\%$ .

X10 Devices: A typical X10 automation system consists of a transceiver module and several receiver modules. The transceiver that can transmit command and receive status of a receiver mainly interfaces with PC or micro-controller through USB or Serial port and communicates with the receiver modules through the home power line. The receiver modules are our mail home appliance device like light, fan, heater, water sensor, security sensors etc. They receive commands from X10 transceivers and act according to the command.

X10 Transmitters are controlling devices comprised of simple plug-in-push button consoles, wall switches, programmable timers and controllers or even home PC or micro-controller. These transmitters send signal codes through the home's electrical wiring system with a push of a button or automatically to the X10 receivers waiting to hear their individual codes. There are several different X10 transceivers that can be used to interface with computer or micro-controller and control other receiver modules. CM15A and CM11A are amongst the mostly used transceiver.

Each X10 receiver module has a House code and a Device code that uniquely identifies it amongst the 256 different possible combinations. When a receiver module hears its user-settable code it responds by performing the corresponding command like Turning On, Off, Dimming or Brightening a light, Turning On or Off a fan, Coffee maker, Security sensors and many more things. They also listen for All On or All Off signals that enable the user to turn on or off all modules with the touch of just one button. A receiver may be a plug-in module, wired electrical outlet or an overhead light wall switch. The different types of receiver provided by X10 are Lamp modules, Appliance modules, Security sensors etc [6]. Short descriptions of them are given below.

*X10 Lamp Modules:* X10 Lamp modules provides us the opportunity to turn lamps on or off, dim or brighten lamp's light level from anywhere in the house with no additional wiring. The lamp modules can only support low power devices like lamps and it should not used to control high power devices like heater, refrigerators etc. The most common lamp modules are LM 465, LM15 and LM14A.

*X10 Appliance Modules:* X10 appliance modules are used to control high power home appliances like water heaters, air conditioners, dehumidifiers, stereos, televisions, some kitchen ranges and many more. Two different appliance modules are available rated at 15 amps and 20 amps respectively. The common appliance modules are AM486 and AM466. To control an appliance we just need to plug the 2 or 3 pin appliance into the appliance module and then plug the appliance module into any normal wall socket.

*X10 Security Sensors:* X10 products allow the user to ensure their home security by providing various kinds of security sensors such as EagleEye Motion Sensor, HawkEye Motion Sensor, and Wireless Motion Sensors etc. They not only provide security, they can be used for many other purposes. For example one can configure to turn on the light when someone enters the room using the motion sensors. Wireless motion sensors have a range of almost 100 ft.

## **5. Prototype Implementation**

We implemented a prototype of the discussed remote home appliance controller. Our implementation controls the home appliances of a smart home using mobile phones from anywhere. We also devised and implemented various ways so that a mobile phone can communicate with the computer or microcontroller. X10 is one of the widely used technologies in the world of Smart Home. We used X10 Active Home Pro to take care of the real appliance 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.

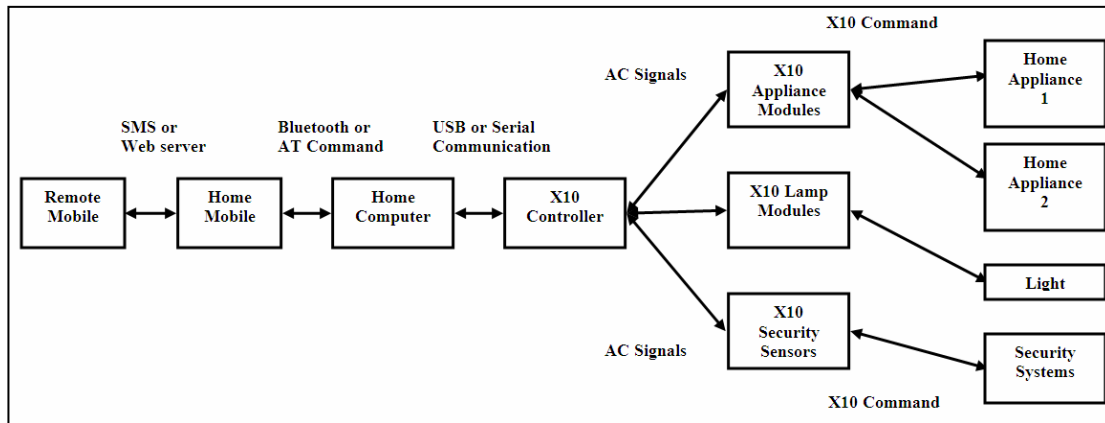
*Brief Comparison of Different Alternative Solutions:* Most of commercially available solution for home appliance controlling is web based. So we started out thinking with the web based solution to find out the pros and cons of the available solutions and to

come up with better solutions. The web based solution requires a web server hosting and internet connection at both the mobile phone and the computer resides at home. Providing these requirements are fulfilled, the home computer is responsible to receive the commands from the mobile phone and to forward the commands in appropriate format to the X10 controller. The main disadvantage of web based solution is its dependence over internet all the time. So this solution cannot be termed as 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 programs. 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 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 proposed the AT command based solution 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. 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.





**Figure 1. System Block Diagram**

*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 devices receive the message, but only the target device, that is specified in the message, responds. The constraint applied here is, X10 messages are only valid if there is no transformer 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. The advantages of our solution are described below

1. Our solution doesn't need internet which is not a secure and reliable media.
2. This is the first solution of its kind that uses mobile devices which are commonly used by everyone now days.
3. Use of mobile devices gives mobility to the solution and home appliance can be controlled from almost everywhere on earth.
4. Our solution uses standard third-party controller for home appliance controlling that makes our solution robust. Our solution controls the third-party controller and the third-party controller takes the responsibility of home appliance controlling. We proposed a number of communication mechanism by which a large number of controllers can be supported.
5. Our system is not tightly coupled to a particular vendor for the third-party controllers. So, we can have any third-party controller and still make use of the mobility and accessibility.
6. Although this is about home controlling, the underlying core of the technology can be used for controlling any system using mobile telephony.
7. This is a comparatively low cost, easy to maintain and the most accessible solution.

## **6. AT Command Based Solution**

Wherever AT commands 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 settings 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'. Otherwise it sends an error message.

SMS Related AT Commands: The sequence of commands and responses are given below [9]:

**Table 4. Sending SMS**

<i>PC</i>	<i>Mobile</i>
AT	OK
AT+CPIN?	+CPIN: <code>
AT+CMGF=1 [For Text Mode]	OK
AT+CSCA="<smc number>"	OK
AT+CMGS="<destination phone number>"	>
<SMS text>Ctrl-Z [must be terminated with Ctrl-Z (ASCII 26)]	OK

**Table 5. Receiving SMS**

PC	Mobile
AT	OK
AT+CPIN?	+CPIN: <code>
AT+CMGF=1 [For Text Mode]	OK
AT+CNMI=1, 2,0,0,0	OK

After this the 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.

#### Pros and Cons

##### Pros

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. From the programming point of view, all the efforts are rendered for the computer and not the mobile, which makes the total implementation a relatively simple task.
4. 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.

### Cons

1. AT command is somewhat device dependent. So, to permit devices from multiple vendors and multiple models, it may be necessary to alter the commands accordingly.
2. It is desired for a bidirectional communication that the communication can start from any of the two ends. But here the communication always starts with the command from the computer's end.

## **7. 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 devices, 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 entries, 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 services, 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 Bluetooth 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 devices 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 remote 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 device. 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 provides 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, Device 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 software (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 devices 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 `ServiceRecord` object 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 remember 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 <btsp://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 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 more information on Bluetooth see [12] [13].

#### Pros and Cons

##### Pros

1. Bluetooth offers a global standard for connecting a wide range of devices with different services. So, using a Bluetooth protocol at the bottom makes one application somewhat 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.

##### Cons

1. One of the major limitations of Bluetooth technology is the short range of operations. So the system is unable to cope with mobility.
2. Bluetooth is not fully developed and integrated into all the products involved. We have to still wait for it to get fully evolved.

3. In J2ME application data transfer by any custom built software requires human intervention which makes the Bluetooth solution infeasible for our case.

## 8. 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 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.

## 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] 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](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] SmartLabs Technology, INSTEON - Wireless Home Control Solutions for Lighting, Security, HVAC, and A/V Systems, <http://www.insteon.net/>
- [17] Z-Wave Alliance, Z-Wave - Wireless Home Control Solutions for Your Lighting, Security, Home Entertainment, Energy Management, <http://www.z-wave.com/>
- [18] NETVOX Smart House, <http://www.netvox.com.tw/Smart%20House/Index.html>



## Authors

**Rifat Shahriyar.** Rifat is a M.Sc. student at Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. He is also working as a lecturer in the same department. His research interests are Ubiquitous and Pervasive Computing, Wireless Sensor Networks and their applications in Ubiquitous and Pervasive Computing, Software Engineering and Programming Languages.

**Enamul Hoque.** Enamul is working as a lecturer at Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. His research interests are Wireless Sensor Networks and their applications. He is going to pursue PhD from University of Virginia in CS starting from Fall 2008.

**S.M. Sohan.** Sohan completes B.Sc. from Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. He is going to pursue M.Sc. with Erasmus Mundus scholarship in European Master in Informatics very soon.

**Iftekhar Naim.** Iftekhar completes B.Sc. from Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. He is going to pursue PhD from ECE department of Rochester University very soon.

**Md. Mostofa Akbar.** Mostofa completes PhD from University of Victoria, Canada. He is working as an associate professor at Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. His research interests are Multimedia Systems, Knapsack Problem, Distributed Systems, Computer Networks and VLSI (System on Chip, Network on Chip).

**Masud Karim Khan.** Masud completes M.Sc. from Department of Computer Science and Engineering of University of Texas at Arlington. He worked as a senior software engineer at Nokia from 1999 to 2007. Currently he is working at Windows Mobile Division at Microsoft.

