CONTROLLING & MONITORING THE STATUS OF HOME APPLIANCES THROUGH RF REMOTE SYSTEM

Sukhwinder Singh
Department of Electronics and Communication Engineering PEC University of Technology Sector-12, Chandigarh Chandigarh, (India)

ABSTRACT

This paper is about RF controlled home automation which is aimed at streamlining daily life and making it more comfortable. Using such system, there is no need to get up to reach the switchboard to switch ON/OFF home appliances. In proposed model, Controlling of the switches is by sending radio frequency waves using data acquisition and processing system (LabVIEW).This project involves sending data through LabVIEW via RF transmitter and at the receiver end, it is processed by microprocessors to control the relays and then, appliances accordingly. This proposed technology can go a long way in making human life more comfortable and streamlined.

Keywords: Automation, Atmega16, Baud Rate, Data Acquisition, Labview, RF Transmitter, RF Receiver, VISA.

I. INTRODUCTION

Development projects based on home automation started on a large scale around the world in 1990s but the absence of a standard for the networking of home appliances appeared as the major roadblock .This was removed in 1992 with the development of 'Consumer Electronic Bus' (CEBus) by the Electronic Industries Association of America. The CEBus standard includes specifications for a layered network architecture based on the Open Systems Interconnection model, with network layer protocols for the Physical, Data link, Network, and Application layers. Until 1993, the home automation networks developed employed guided or wired media for interconnection of appliances. However Fujieda felt that for achieving complete marketability, home networks should be easily installed not only to newly built houses but also to existing houses. So it would be desirable to build up networks without any extra wiring. Therefore he advocated the use of wireless media for home networking and called the network as wireless home networks. For the wireless home network he proposed the use of 400 MHz specified low power (SLP) band. Fujieda developed a low power and small size RF section of SLP band and communication protocol and demonstrated the proposed wireless network to be viable. Using the prototype he also implemented a couple of application systems, a maintenance system for instantaneous gas water heaters and a health promotion system with chronic disease prevention. In 2003 Hiroshi Kanma and others observed that although the rapid spread of Internet at home may provide a convenient way of implementing a home network and its control, however there were certain hindrances to be removed to make home automation common. To solve all these problems Kanma proposed the use of Bluetooth as communication medium and a cellular phone as the terminal equipment. A communication adapter was attached to the home appliances in order to provide a Bluetooth communication functionality which eliminated the need of purchasing new appliances for the home network. Further developments in this direction have been done on various Bluetooth kits/boards produced by Man n Tel ltd, Korea. In 2005, Tajika articulated that the home network technology was focused primarily on how data and access protocols on the Internet can be utilized in the home network by converting them into in-house protocols through a home gateway. However he felt that apart from control only, other novel services can be provided to home through the Internet resulting in better user experience without any loss of flexibility. The system proposed by him composed of networked home appliances such as refrigerators, microwave ovens, air conditioners, and washing machines, Bluetooth access point and home terminal in a home. Bluetooth units were embedded within all the appliances and these were connected to the Internet. Home terminal was connected to the home network and the Internet and it provided a well-designed GUI to the user through touch panel and voice recognition and it also worked as a gateway between the home network and the Internet service provider. In 2006 Mario Kolberg and Evan H. Magill addressed the control of complex networked appliances. A standard computer interface was most often used to configure and remotely control these appliances. However they argued that this is unsuitable for the target audience which is often inexperienced with the use of computers. Therefore they proposed an auto-enabled pen and paper as a suitable alternative as users are highly familiar with pen and paper and they will find it suitable for control. In the proposed system Bluetooth and mobile telecommunication network is used to transfer data to a service provider where it is processed and sent to the user's home. In this paper, a system controlling the electrical devices using radio frequencies is proposed. The user just types in a specific character to monitor the device even while sitting at some distance from it. I have tried to keep the interface user friendly. LabVIEW has been used for data acquisition.

II. HARDWARE IMPEMENTATION

The following modules are used in proposed system:

1. Radio Frequency Module

An RF Module (Radio Frequency Module) is a (usually) small electronic circuit used to transmit and/or receive radio signals on one of a number of carrier frequencies. RF Modules are widely used in electronic design owing to the difficulty of designing radio circuitry. RF module comprises of RF Transmitter and RF Receiver.

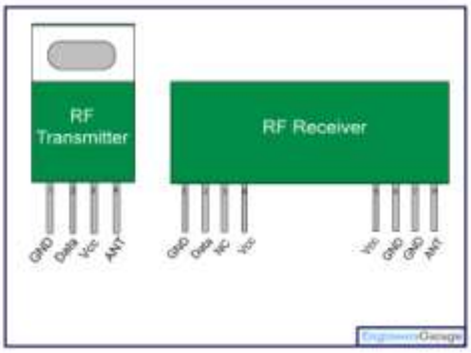

Fig.1: Pin diagram of RF Module

RF Transmitter: An RF transmitter module is a small PCB sub-assembly capable of transmitting a radio wave and modulating that wave to carry data. RF Receiver: The transmitted signals are received by the receiver module placed away from the source of transmission. In the receiver side, the RF Receiver receives the modulated signal through the antenna, performs all kinds of processing, filtering, demodulation, etc. and gives out a serial data. This serial data is then converted to a TTL level logic data, which is the same data that the user has input.

1. Antenna

An antenna (or aerial) is an electrical device which converts electric power into radio waves and vice versa. It is usually used with a radio transmitter or radio receiver. In transmission, a radio transmitter supplies an oscillating radio frequency electric current to the antenna's terminals, and the antenna radiates the energy from the current as electromagnetic waves (radio waves). In reception, an antenna intercepts some of the power of an electromagnetic wave in order to produce a tiny voltage at its terminals that is applied to a receiver to be amplified.

2. SPDT Relays (sc5-s-dc6v relay):

The actuation of appliances is performed via them. Since they are designed to run on 230 VAC, they are ideal for our application.

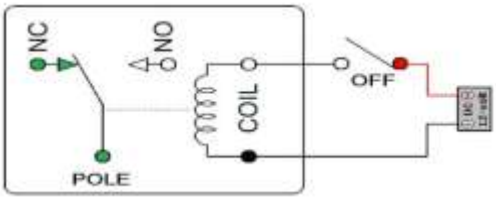

Fig. 2: Internal Structure of a SPDT relay

3. ULN2803 actuation IC

To power on the relay we need 60mA of current, whereas the maximum sourcing capacity of the controller is 2mA hence some external actuation is required. Also it is worthwhile to note that even switching transistors are not able to source as much current hence one may use a power transistor or an opto-isolator, we chose the latter as it is small in size and cheaper. The power rating of this IC is 12V and can source 100mA+ from each.

4. USB to Serial convertor

LabVIEW communicates with external world using serial port (UART) and since modern day laptops and desktops do not have DB9 (serial port) connector and ATmega16 does not support USB protocol hence a USB to serial convertor is required. I have used PL2303 USB Bridge to perform the said operation.

Fig. 3: USB to Serial convertor

6. ATmega16 microcontroller**:** The UART signals from labVIEW must be processed by some microcontroller, I have used ATMEL'S ATmega16 for this purpose, it has an on chip UART engine hence combined with a USB to serial convertor it is ideal for the job. Besides it has 16KB of on chip flash which is more than sufficient to hold the required code. The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM; Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run. The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

III. SOFTWARE IMPLEMENTATION

We have used LabVIEW for sending data to the RF transmitter module via USB to TTL convertor. This signal is then processed by Atmega16, whose code has been written using AVR Extreme Burner.

1. LabVIEW

LabVIEW is interfaced with hardware using an interface called VISA.

2. VISA

VISA is a standard I/O language for instrumentation programming. VISA by itself does not provide instrumentation programming capability. VISA is a high-level API that calls into lower level drivers. VISA is capable of controlling VXI, GPIB, or Serial instruments and makes the appropriate driver calls depending on the type of instrument being used.

VISA Advantages

One of VISA's advantages is that it uses many of the same operations to communicate with instruments regardless of the interface type. For example, the VISA command to write an ASCII string to a message-based instrument is the same whether the instrument is Serial, GPIB, or VXI. Thus, VISA provides interface independence. This can make it easy to switch interfaces and also gives users who must program instruments for different interfaces a single language they can learn. VISA is also designed so that programs written using VISA function calls are easily portable from one platform to another. To ensure this, VISA strictly defines its own data types such that issues like the size, of an integer variable from one platform to another should not affect a VISA program. The VISA function calls and their associated parameters are uniform across all platforms; software can be ported to other platforms and then recompiled. In other words, a C program using VISA can be ported to other platforms supporting C. A LabVIEW program can be ported to other platforms supporting LabVIEW. At first, the abundance of new VISA data types might seem like a burden, but LabVIEW makes them virtually transparent when using VISA.

Another advantage of VISA is that it is an object-oriented language which will easily adapt to new instrumentation interfaces as they are developed in the future. The interface independence that is built into the language will also make it even easier for programmers to move to new interfaces with VISA in the future. VISA's greatest advantage, perhaps, is that it is an easy language to learn and use. Its object-oriented structure makes the language and its operations intuitive to learn. This is due in part to the fact that VISA provides interface independence by using the same operations for different interfaces and also by the fact VISA presents a very simple and easy-to-use API. VISA provides the most frequently used functionality for instrumentation programming in a very compact command set.

3.1 VISA Programming

VISA is an object-oriented language. The most important objects in the VISA language are known as resources. In object-oriented terminology, the functions that can be used with an object are known as operations. In addition to the operations that can be used with an object, the object has variables associated with it that contain information related to the object. In VISA, these variables are known as attributes. There is a default resource manager at the top of the VISA hierarchy that can search for available resources or open sessions to them. Resources can be GPIB, serial, message based VXI, or register-based VXI. The most common operations for message based instruments are reads and writes. The most common operations for register based instruments are In's and Out's. In addition, resources have a variety of properties associated with them known as attributes. These can be read or modified with Attribute Nodes. The Default Resource Manager is at the highest level of VISA operations. Communication must be established with the Resource Manager at the beginning of any VISA program. This immediately brings up two terms that need to be defined: resource and session. Resource - An instrument or controller. Session - A connection, or link, to the VISA Default Resource Manager or a resource. The reason the VISA Default Resource Manager is so important is because one of its operations is to open sessions to other resources. Sessions must be opened to the instruments the application will communicate with. Therefore communication with the Default Resource Manager must be established first within an application.

The VISA Default Resource Manager also has another operation that it can carry out. That operation is to search for available resources in the system. Sessions can then be opened to any of these resources. The VISA Find Resources VI that carries out the operation of searching for available resources in the system. This VI is a common starting point for a VISA program. It can be used to determine if all of the necessary resources for the application to run are available. The only necessary input to the VISA Find Resources VI is a string called the search expression. This determines what types of resources the Find Resources VI will return. The important return values of the VI are the return count (which simply reports the number of resources that were found) and the find list. The find list is an array of strings.

3.2 Error Handling With Visa

Error handling with VISA VIs is similar to error handling with other I/O VIs in LabVIEW. Each of the VISA VIs contain Error Input and Error Output terminals that are used to pass error clusters from one VI to another in a diagram. The error cluster contains a Boolean flag indicating whether an error has occurred, a numeric VISA error code, and a string containing the location of the VI where the error occurred. If an error occurs, subsequent VIs will not try to execute and will simply pass on the error cluster. VISA error codes are 32-bit integers that are usually referred to in hexadecimal format. The LabVIEW error cluster displays the code in decimal. VISA also provides an operation that will take a VISA error code and produce the error message string corresponding to the code as an output. The VI will check the VISA code in the error cluster that was passed to it and output the text description of the code from the status description terminal. Even if the input error cluster did have the error value set, a status string description indicating success or a warning will be produced. This VI allows a text description of the status after a VISA VI executes to be obtained and used in a program.

3.3 Visa Attributes

VISA resources have a variety of attributes whose values can be read or set in a program. In a LabVIEW program, these attributes are handled programmatically in the same way that the properties of front panel controls and indicators are handled. Attribute nodes are used to read or set the values of VISA attributes.

Initialization of RF Module: To communicate with external hardware a serial port must be opened, the said serial resource is opened using visa serial function.

3. Parameters of Serial communication

3.1 Baud Rate: No of signals transferred per second is known as baud rate, commonly used baud rates are 1200, 2400, 4800, 9600, 19200 and so forth. For our project we have configured the serial port at 2400bps.

3.2 No. of data bits: This variable specifies the number of data bits transmitted per data packet, since we are transmitting ASCII characters hence we are using 8 bits per data packet.

3.3 Start bit: Since the serial communication is asynchronous hence we require a start bit, in case of USART start bit is active low.

3.4 Stop bits: The bit(s) indicate the end of data reception, they are active high. In our project we use a single stop bit.

3.5 COM PORT: This specifies the visa resource name i.e. the virtual serial port that we are using for communication with the external hardware.

4. Labview Programming

We have placed the whole circuit in a flat sequence structure with different frames.

After configuring the serial port, While loop checks the status of the 'Main Switch'. If it is OFF, control will be forwarded to the 'VISA Write' function, which sends the character 'X' to the port, outside the while loop. Then, this session is closed through 'VISA Close'.

If the 'Main Switch' is ON, frames of sequence structure will be executed. These frames monitor the switching operation of electrical devices. Some character is sent using VISA write function according to the input.

Frame 1: In our project the first fame of the sequence structure has been used to operate a bulb, named 'Bulb-1'. This frame reads the input of the 'Bulb-1' switch and operates accordingly. When the input is True, character 'A' is sent to the configured port. For False input, character 'a' is sent.

Frame 2: The second frame has been used for 'Fan'. True input from the 'Fan' switch results in character 'B' being sent while in the other case, 'b' is sent.

Frame 3: This frame operates 'Bulb-2' switch. It sends the character 'C' for True input while 'c' for False input.

Followed by VISA Write function, a function for providing delay of 1 second has been used. Before sending new data bits, we need to ensure that the previously sent data bits have been transmitted. For this purpose, we delay the sending process of new data bits by an appropriate time gap. In case we don't use this function, previous data may get overwritten by the new one leading to the transmission of incorrect data.

Finally when 'Main Switch' is turned OFF, While loop stops executing and character 'X' is sent. Further, VISA Close function is used to close the LabVIEW session. This action allows the port used for transmission to be used for other purposes as well.

5. AVR Studio

AVR Studio is used to write the code to the ATmega 16 microcontroller. Code is written in C programming.

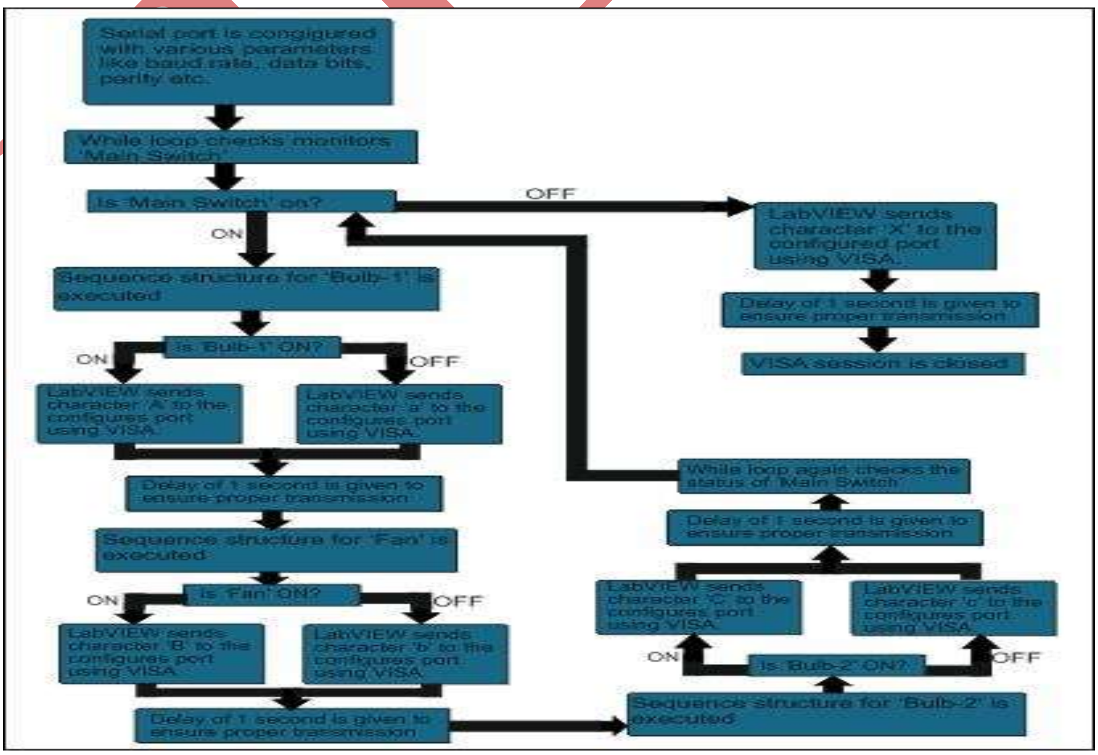

Fig. 4: Algorithm of LabVIEW code

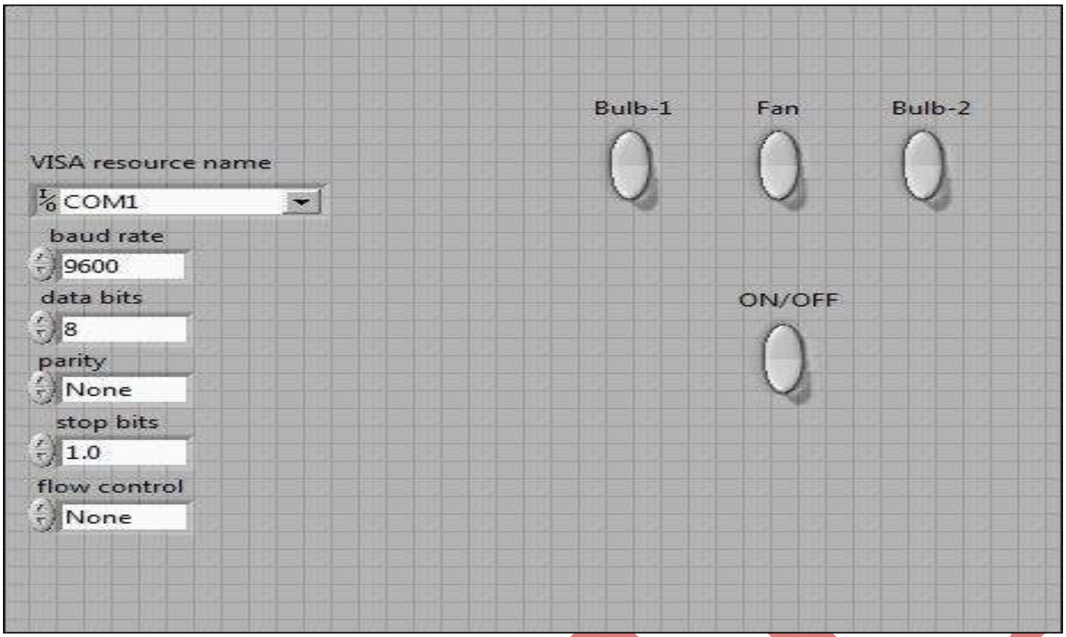

Fig. 5: Front Panel

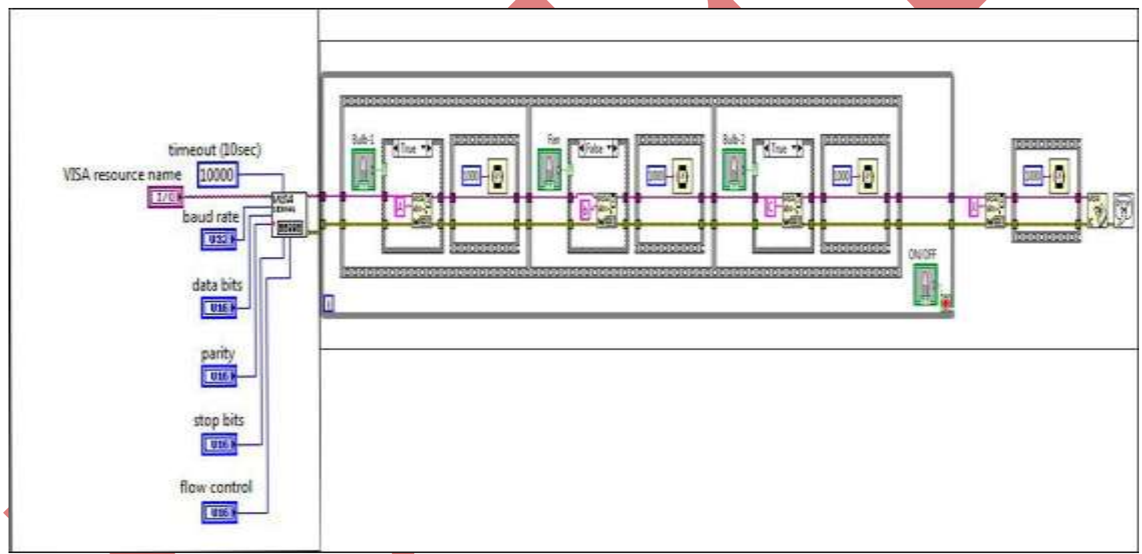

Fig. 6: Block Diagram

IV. DISCUSSIONS AND CONCLUSIONS

We were successful in achieving the following major tasks required in this project: 1. Manipulating the received data in the LabVIEW, to such a format that we wanted to send to the port of the computer which we were using as the output port. 2. Using the output data from the computer to drive various home devices, via the help of a microcontroller (Atmega 16). Advantages of this system of home automation: 1. One can control his home appliances using laptop even while sitting away from the switchboard. 2. More reliable and a secure method as interference by presence of other signals will not be there as in case of Bluetooth. 3. If required the system can also be interfaced with the security system of the house.

4. If commercialized, the system would be great success.

Disadvantages of this system: 1. User will have to be connected to his/her computer always to use this system to keep the device continuously ON. 2. The system can't be used for wide range as the range of RF module is up to 7-8 meters only. 3. The system is feasible only for the houses or offices under construction, as the system can be wired in the initial stages of the construction. In already built houses or offices complete adjustments have to be made in the wiring, which is not feasible.

V. FUTURE SCOPE

Increased range of RF module will be a great boon to this technology. This technique can also be used in home security. For example if someone tries to break in a house, the security system will sense the presence of the unwanted person and will immediately switch on the lights. This can be used in automation of industries to great extent. Mechanical switches used in houses will be completely replaced by the automatic ones and will be controlled by the laptops. By interfacing security cameras and mechanical actuators that control door locks, the user will be able to allow only certain people to enter his house by unlocking the doors. Apart from the appliances in our project the system module can be used to control nearly all home appliances.

REFERENCES

[1] Langhammer N., Kays R.; "Smart Grid" ,IEEE Transactions Dec. 2012, Volume 3, Issue 4, pp 2252 – 2261.

[2] Zhao Jun ,He Lin Bo; "Electrical and Control Engineering (ICECE), 2011",16-18 Sept. 2011,pp 2861 - 2864 (Conference Publications).

[3] Rao B.S., Prasad S.D.V., Mohan R.M.; "Power, Control and Embedded Systems (ICPCES), 2010", IEEE conference paper, 29/11/2010-1/12/2010; pp 1 – 4.

[4] Al-Ali, A.R. Rousan, M.A. &Mohandes M.; "Information and Communication Technologies from Theory to Applications, 2004", IEEE communication magazine, 19-23 April 2004, pp 237 – 238.

[5] Mario Kolberg& Evan H. Magill; "Using Pen and Paper to control Networked Appliances", IEEE Communications Magazine, November 2006, pp. 148-154.

[6] H. Nakakita, K. Yamaguchi, M. Hashimoto, T. Saito, M. Sakurai; "A Study on Secure Wireless Networks Consisting of Home Appliances", IEEE Transactions on Consumer Electronics, Vol. 49, No. 2, May 2003.