INTERNATIONAL JOURNAL OF COMPUTING, SCIENCE AND NEW TECHNOLOGIES (IJCSNT)
VOL. 5 NO. 1 AUGUST 2025

ENHANCED APPLIANCE CONTROL USING BLUETOOTH-BASED ANDROID APP

by
Obianyo O.R.[1], Ofodile Izuchukwu S.[2], Onyia T.C.[3]
1: Dept of Computer Engineering, Madonna University of Nigeria, Akpugo Campus, Enugu State, Nigeria.
2,3: Dept of Electronics & Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria.

ABSTRACT

This research work analyzes the design and implementation of a Bluetooth-based android app device control system for appliance controls in homes and industries. The study demonstrates the design of an automated system to assist and provide support in order to fulfill the needs of the physically disabled, sick and elderly ones in controlling electrical appliances in homes. The heart of the control system is the wireless Bluetooth technology which provides remote access from a smart phone to a Bluetooth module and then to a control and switching unit. The block diagram of the system was first designed, followed by the circuit diagram design using Proteus software. An algorithm (flowchart) was developed, and the system was programmed using two programming languages. One was MIT programming language, which is an aspect of Android programming language for the android APP development (for GUI phone usage), while the other one was the Arduino programming language for device controls using Arduino board. The range of appliance control is the Bluetooth range of 45 meters on a line of sight and not an SMS range, so there is nothing like network problem during operations. The system was later tested module by module, and all the control operations were working perfectly. The frequency of the Bluetooth is 2.5 GHz (type 2); and covers 45 meters on a line of sight and 10 meters on non-line of sight with wall penetration. This control system App thus, is of a low-cost design, installation ease, quick response time and a user-friendly interface in controlling home electrical appliances.

Keywords/Phases: Android, Arduino, Algorithm, Bluetooth Proteus, and Appliances.

I. INTRODUCTION

The attractiveness of controlling electrical devices through a phone has been increasing because of its high performance and availability. Connecting appliances through smartphone is useful for the elderly and physically disabled persons, who can access and control the appliances from where they are located and access them remotely without the help of others. Time is a precious thing; everybody wants to save time as much as they can (Greichen J.J., 2022).

Home automation systems are a technological means of intelligent monitoring, control, feedbacks and actions of home appliances according to the needs of the home occupants. Wireless medium such as ZigBee, Bluetooth, wireless Fidelity (Wi-Fi), Short Message Service (SMS), Android Application, Wireless Sensor Network (WSN), Radio frequency identification (RFID) and Software Defined Network just mention a few serves as a medium of communication between the appliances and the control unit and according to home automation are essential for non-invasive and non-intrusive implementation of the advanced automation system (Ehiabhili J.C. et. Al, 2018).

Android, a vivid operating system, has 76.24% usage worldwide, and 78.05% of Nigeria's total smartphone market share. Android has become the topmost used mobile gadget operating system (OS) on the market today. The Android smartphone has become the most popular and commonly used Operating System in our world, especially in Nigeria. This had made us base the control terminal of the home automated system on an android application as it has been shown that majority cannot do without their phone with them almost all the time.

Conventional home-controlled systems and its components are all wired to the same cable that connects them to the home control panel. The key problem with conventional home-controlled systems is that they require the mobility of the user to operate it, hence the need for automation.

II. LITERATURE SURVEY

Several research works have been done in the area of intelligent home/industrial automations using Arduino board and Bluetooth technology.
In the past, Radio Frequency Identification (RFID) was used to design and monitor the indoor activity of elderly people in a smart home by Al-Ali M. and Al-Rousan M. (2014). The RFID monitored and collected the movement activity of the elder person using RFID tags, information collected was used to take critical decisions about the health of the elderly person. But RFID cannot work where its radio waves do not cover.

Similarly, Sharon P. and Mahesh J. (2019) presented A Smart Home Model Using Android Application where the vital signs of the elderly are monitored. A communication module and sensors such as accelerometer, gyroscope, force and temperature were attached to the knee and walking analyzer to monitor and extract of a person's unique walking pattern, which is used to classify the walking pattern an individual whether walking anomaly and walking with a level of stability. The signals are taken by the sensors and information gathered from walking analyzer and knee monitor was used to design a smart sleep room where vital signs like body temperature, breathing patterns and cardiac are monitored these signs are sent through a smartphone and a local wireless network to monitor elderly persons' health. The processes involved in this design are too cumbersome and if sensors are not properly attached signal might fail and waking pattern will not be got.

Also, Poonam M. et Al. (2017) proposed "a smart home model using android application" the home model uses ZigBee module to communicate between the android phone and the smart home model. This is not an effective medium of communication, as an external ZigBee transceiver must be connected to the android phone. This leads to waste in power, use of many components as compare to Bluetooth that is part of an android phone already.

In the same vein, Nisar K. et al. (2016) proposed a smart home automation for the elderly using Wireless sensor network and android application, the system was divided into modules namely sensor, control and actuator. The sensor module served as the transmitting medium, the control module controlled the smart house, remote monitoring of other sub-modules and the actuator modules showed the response of all appliance connected in the actual system as illustration, the drawback of this system is that failure to the control modules is failure of the whole system.

Moreso, Kanma H. et al. (2020) presented a low-cost Bluetooth based home automation system using an Android phone was presented by. An Arduino Mega 2560-R3 board and relays were used to connect the home appliances as input/output ports of the board, a Bluetooth were used to establish wireless communication between them.

III. MATERIALS AND METHODS

3.1 Hardware and Software Requirements/Materials

3.1.1 Hardware Requirements:

- Arduino board
- Transformer
- Voltage Regulator
- HC-06 Bluetooth module
- Triacs
- Sockets
- Lamp
- Liquid Crystal Display (LCD)
- Android mobile phone with Bluetooth Device
- Push buttons

3.1.2 Software Requirements:

- Arduino Software package
- MIT App inventor 2 Software (for Android Application)

The wireless controlling technique utilized in this research is Bluetooth technology. It mainly consists of the following blocks:

- Android mobile phone with android app installed
- Bluetooth receiver unit
- Microcontroller
- LCD Display
- Relays
- Output devices

3.2 The Block Diagram of the System

The block diagram of the system is shown in figure 1. It consists of six units, namely: power unit, display unit, controller unit, switching unit, Bluetooth unit, and controlled device unit. These block units will be explained.

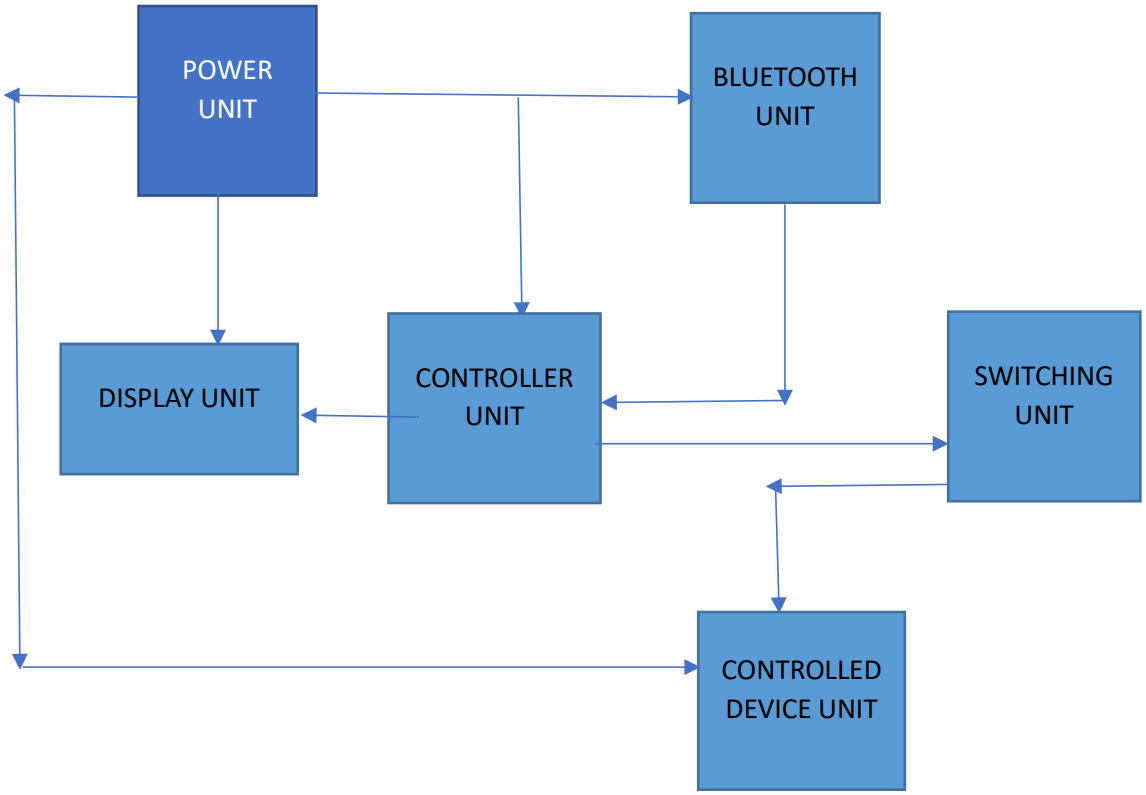

Figure 1: The Block Diagram of the System

3.2.1 Power Supply

The power supply unit supplies power to the whole circuit. It supplies both the 12V which drives the controlled device and the 5V needed by the display unit, microcontroller, and the display unit. This unit is made up of a transformer, a bridge rectifier, an electrolytic capacitor, and two regulators. A block diagram of the power supply unit is shown in figure 2.

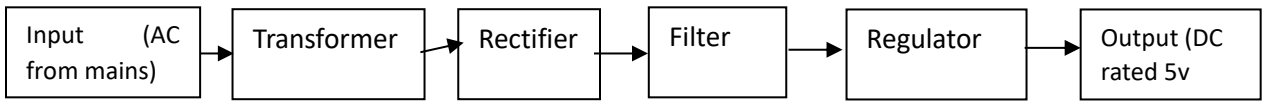

Figure 2: Block Diagram of the Power Supply Unit

3.2.2 Display Unit

The display unit consists of the Liquid Crystal Display (LCD). Different kinds of LCD exist, but the one to be used in this project is LM016L which is an alphanumeric LCD. This LCD is a 16 X 2 LCD and can display all alphabets as well as all numbers. It displays 16 characters up and down, i.e. in the upper and lower columns. Figure 3 shows a typical LM016L LCD.

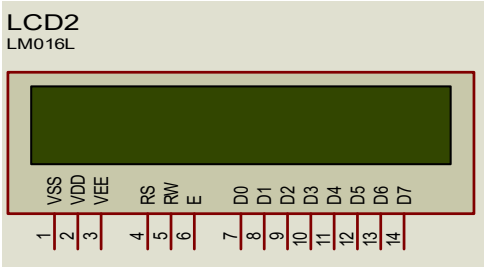

Figure 3: A Typical LM016L LCD

Main Control Pins: There are three control pins in LCD. They are; E = Enable, RW=read/Write, RS= Register Select.
The RS controls two important registers; the data register and the command register. When you are developing a program for LCD display;
If RS = 0, It means select command register
If RS = 1, it means select data register
If RW = 0, it writes data to the LCD
If RW = 1, it reads data from the LCD.
The E (enable pin) is used to latch information given to the data pins.
The D0, D1, D2,…D7 pins are the data pins. In LCD programing, the delay must be above 450µs because it takes a minimum of 450µs for latch to take place so use 1 second delay always.

Commands Used in LCD Programming
By default, there are commands that are inbuilt in every LCD device. These commands are listed in table 1. They are useful in LCD programming.

Table 1: Table Showing Commands Used in LCD Programming

CODES	COMMAND TO LCD
1h	Clear display screen
2h	Return home
4h	Shift cursor to the left(cursor not moving with character)
5h	Shift display to the right
6h	Shift cursor to the right
7h	Shift the display to the left
8h	Display off, Cursor off
Ah	Display On, Cursor On
Ch	Display On, Cursor Off
Eh	Display On, Cursor Blinking
10h	Shift cursor to the left (cursor moving with character)
14h	Shift cursor to the right (cursor moving with character)
18h	Shift the entire display to the left
1Ch	Shift the entire display to the right
80h	Force cursor to begin on first line
0C0h	Force cursor to begin on second line
38h	Select two lines and 5 X 7 matrix

LCD Subroutines
There are two important subroutines to note when writing an interfacing LCD program. One is the common write subroutine, and the other one is the data -write subroutine.

Command Write subroutine: This is a subroutine for writing command to the LCD. It is in this form:
Mov p1, A --- Put the content of the accumulator into port 1
Clr RS --------Make RS=0, so as to select command register

Clr RW -------Make RW=0, so as to write data onto LCD
Set b E -------Make E high, and delay it, then make it low so as to hold data.
Call delay
Clr E
Ret

Data write subroutine: This is a subroutine for writing data or character to the LCD display. Below are the codes:
Mov p1, A
Setb RS ------So as to select data register (RS=1)
Clr RW
Setb E
Call delay
Clr E
ret

3.2.3 Switching Unit

The Switching Unit of the project chiefly contains TRIACs. **Triac** can conduct whether the applied gate signal is positive or negative, making it ideal for AC systems. It is a three terminal, four-layer, bi-directional semiconductor device that controls AC power. The triac of maximum rating of 16 kw is available in the market.

Operation of a Triac

The TRIAC can be activated by applying a gate voltage higher than the break over voltage. Alternatively, it can be turned on by a 35-microsecond gate pulse. When the voltage is below the break over voltage, gate triggering is used.

There are four different modes of operations, they are-

1. When MT_2 and Gate being Positive with Respect to MT_1
   When this happens, current flows through the path P_1-N_1-P_2-N_2. Here, P_1-N_1 and P_2-N_2 are forward biased but N_1-P_2 is reverse biased. The triac is said to be operated in positively biased region. Positive gate with respect to MT_1 forward biases P_2-N_2 and breakdown occurs.
2. When MT_2 is Positive but Gate is Negative with Respect to MT_1
   The current flows through the path P_1-N_1-P_2-N_2. But P_2-N_3 is forward biased and current carriers injected into P_2 on the triac.
3. When MT_2 and Gate are Negative with Respect to MT_1
   Current flows through the path P_2-N_1-P_1-N_4. Two junctions P_2-N_1 and P_1-N_4 are forward biased but the junction N1-P1 is reverse biased. The triac is said to be in the negatively biased region.
4. When MT_2 is Negative but Gate is Positive with Respect to MT_1
   P_2-N_2 is forward biased at that condition. Current carriers are injected so the triac turns on. This mode of operation has a disadvantage that it should not be used for high (di/dt) circuits. Sensitivity of triggering in mode 2 and 3 is high and if marginal triggering capability is required, negative gate pulses should be used. Triggering in mode 1 is more sensitive than mode 2 and mode 3.

Characteristics of a Triac

The triac characteristics, shown in figure 4, is similar to SCR but it is applicable to both positive and negative triac voltages. The operation can be summarized as follows-

First Quadrant Operation of Triac:
Voltage at terminal MT_2 is positive with respect to terminal MT_1 and gate voltage is also positive with respect to first terminal.

Second Quadrant Operation of Triac:
Voltage at terminal 2 is positive with respect to terminal 1 and gate voltage is negative with respect to terminal 1.

Third Quadrant Operation of Triac:
Voltage of terminal 1 is positive with respect to terminal 2 and the gate voltage is negative.

Fourth Quadrant Operation of Triac:
Voltage of terminal 2 is negative with respect to terminal 1 and gate voltage is positive.

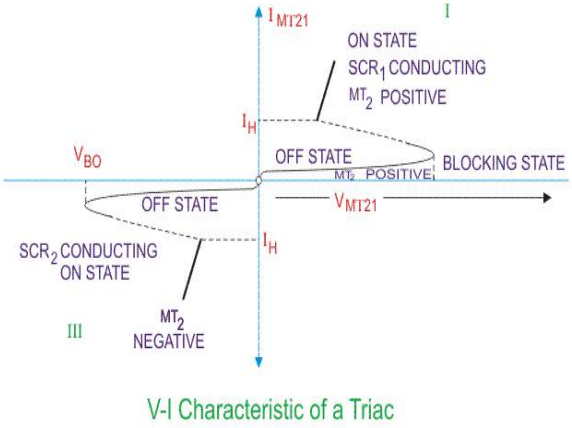

Figure 4: The V-I Characteristics of a Triac

When the TRIAC turns on, a heavy current flows through it, which can cause damage. To prevent this, a current limiting resistor should be used. Proper gate signals can control the firing angle of the device. Gate triggering circuits, such as a diac, can be used for this purpose, with gate pulses up to 35 microseconds.

3.2.4 Controller Unit

The controller unit contains the Arduino board. The type of Arduino Board is the Arduino Uno board. The layout diagram is shown in figure 5. It contains ATmega 328 microcontroller, which is the programmable heartbeat of the system. It also contains other features like digital pins, analog pins, serial ports, serial clock, serial data port, etc.

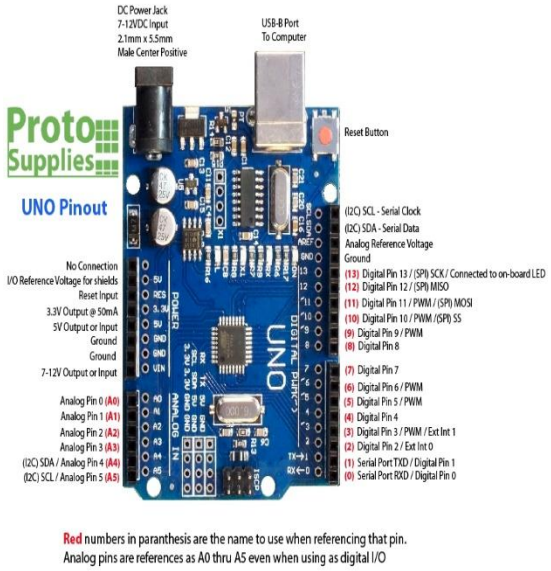

Figure 5: The Layout Diagram of Arduino Uno board

3.2.5 The Controlled Devices

The Controlled devices for this project are: electric bulb, fan, LCD, Buzzer, plus twin sockets. Some other devices can be connected to the sockets. Such devices can be refrigerator, air conditioner, etc.

3.3 Software Design and Implementation

Without the control program that is "burnt" into the microcontroller, the hardware design is as good as useless. In the development of a software program, the following steps were necessary:

- Design conception
- Planning
- Selection of tools
- Coding

Proper planning is required, if not, the software program is seldom successful. The first step in the development is to set out some form of blueprint based on the information gathered on the required system that is to be developed. As a guideline, one algorithm (flowchart) was developed, which represents the system's functionality.

3.3.1 The Algorithm (Flowchart)

The flowchart of the system which is the operational algorithm is shown in figure 6.

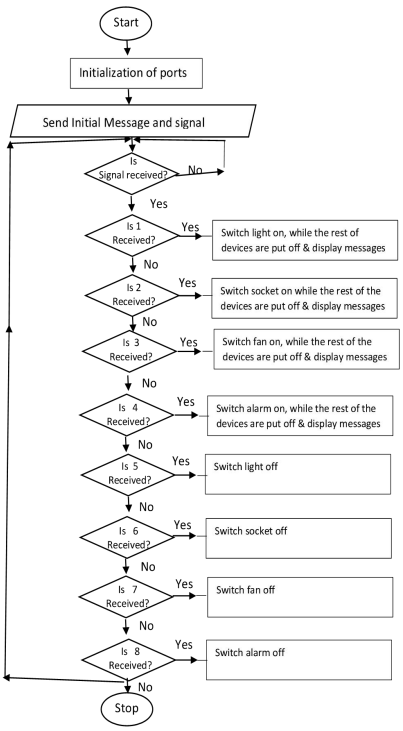

Figure 6: Operational Flowchart of the System

3.3.2 The Pseudocode

Step 1: Begin
Step 2: Initialization of ports
Step 3: Send signal and display initial messages
Step 4: If signal is not received, move to step 3, else move to step 5.
Step 5: If signal 1 is received, switch light on, while the rest of the devices are put off, and display messages; else move to step 6.
Step 6: If signal 2 is received, switch socket on, while the rest of the devices are put off, and display messages; else move to step 7.
Step 7: If signal 3 is received, switch fan on, while the rest of the devices are put off, and display message; else move to step 8.
Step 8:If signal 4 is received, switch alarm on, while the rest of the devices are put off, and display message; else move to step 9.
Step 9: If signal 5 is received, switch light off; else move to step 10.
Step 10: If signal 6 is received, switch socket off, else move to step 11.
Step 11: If signal 7 is received, switch fan off, else move to step 12.
Step 12: If signal 8 is received, switch alarm off, else move to step 13.
Step 13: Stop.

3.3.3 The Proteus IDE

This software was used to design the circuit diagram. Proteus IDE is a virtual system modeling (VSM) and circuit simulation application software. It has a virtual system studio, a free universal IDE for Proteus. Also, Proteus VSM can be used for advanced embedded simulation, offering-system level simulation based on the schematic circuit. It has a wide range of components in its database or library. Besides its own database component, more components can be designed as part of the library component if they are not part of the original software library. Such a library component includes Arduino and Bluetooth Library for Proteus, etc.

3.3.4 The MIT App inventor 2

The android developing platform MIT app inventor 2 is the latest version of an online app used in developing android applications. It contains blocks that are dragged and dropped at the desired blocks in the corresponding place. The developed Application can be installed on an Android phone/tablet with a Bluetooth module. This is shown in Appendix C.

3.3.5 Arduino Uno Environment and Platform

The Arduino IDE

The Arduino IDE is an open-source software that writes, compiles, and upload codes directly into the microcontroller. The version used in this paper is version 1.8.9. The Arduino IDE environment is used for writing the desired software code and for compiling, uploading code into the given Arduino board. Its environment supports both C and C++ language. It is also used for debugging, editing, compiling, and uploading code in its environment to physical hardware modules.

The Arduino IDE, shown in figure 7, is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including "File" (new, load save, etc.), "Edit" (font, copy, paste, etc.), "Sketch" (for compiling and programming), "Tools" (useful options for testing projects), and "Help". The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

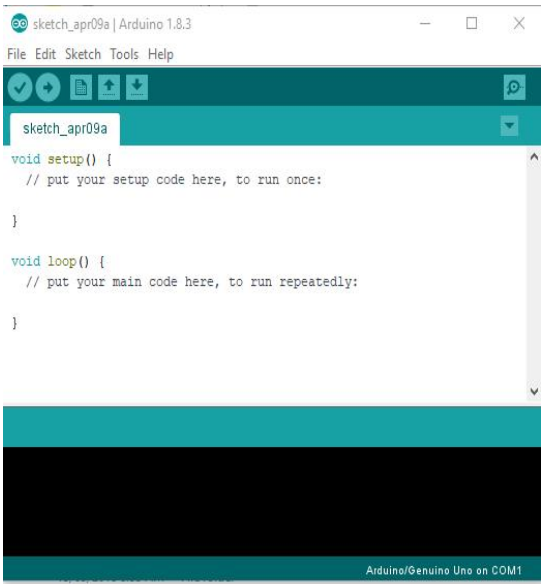

Figure 7: The Arduino IDE in its Default State

The Arduino development environment main program has two functions already defined. The setup and loop function. The setup function is where the code that should run once is written. Setup codes like defining the input and output pins, initializations etc. are written inside the setup function. The loop function should contain the codes that are meant to run continuously in the project. Examples of such codes could be continuous reading of sensor inputs to know when the state changes, displaying outputs in the selected display device etc.

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an "Arduino language." However, the Arduino is, in fact, programmed in C++, but it just used unique libraries for the device.

Steps in Programming the Arduino Board

Step 1:

First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer.

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5: Open your first project.

Once the software starts, you have two options −
Create a new project.
Open an existing project example.
To create a new project, select File → New
Step 6: Select your Arduino board.
To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.
Go to Tools → Board and select your board.
Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.
Step 7: Select your serial port.
Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.
Step 8: Upload the program to your board.
Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.
Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Steps For Burning Programs on the Arduino Board

1. Connect your Arduino using the USB cable. The square end of the USB cable connects to your Arduino and the flat end connects to a USB port on your computer.
2. Choose Tools→Board→Arduino Uno to find your board in the Arduino menu. You can also find all boards through this menu, such as the Arduino MEGA 2560 and Arduino Leonardo.
3. Choose the correct serial port for your board. You find a list of all the available serial ports by choosing Tools→Serial Port→ comX or /dev/tty.usbmodemXXXXX. X marks a sequentially or randomly assigned number. In Windows, if you have just connected your Arduino, the COM port will normally be the highest number, such as com 3 or com 15.

Many devices can be listed on the COM port list, and if you plug in multiple Arduinos, each one will be assigned a new number. On Mac OS X, the /dev/tty.usbmodem number will be randomly assigned and can vary in length, such as /dev/tty.usbmodem1421 or /dev/tty.usbmodem262471. Unless you have another Arduino connected, it should be the only one visible.

3.3.6 The Program
The Android App was developed using MIT (Android) Programming language. However, Arduino programming language was used to program the Arduino Board for device control. The circuit diagram of the developed system is shown in figure 8.

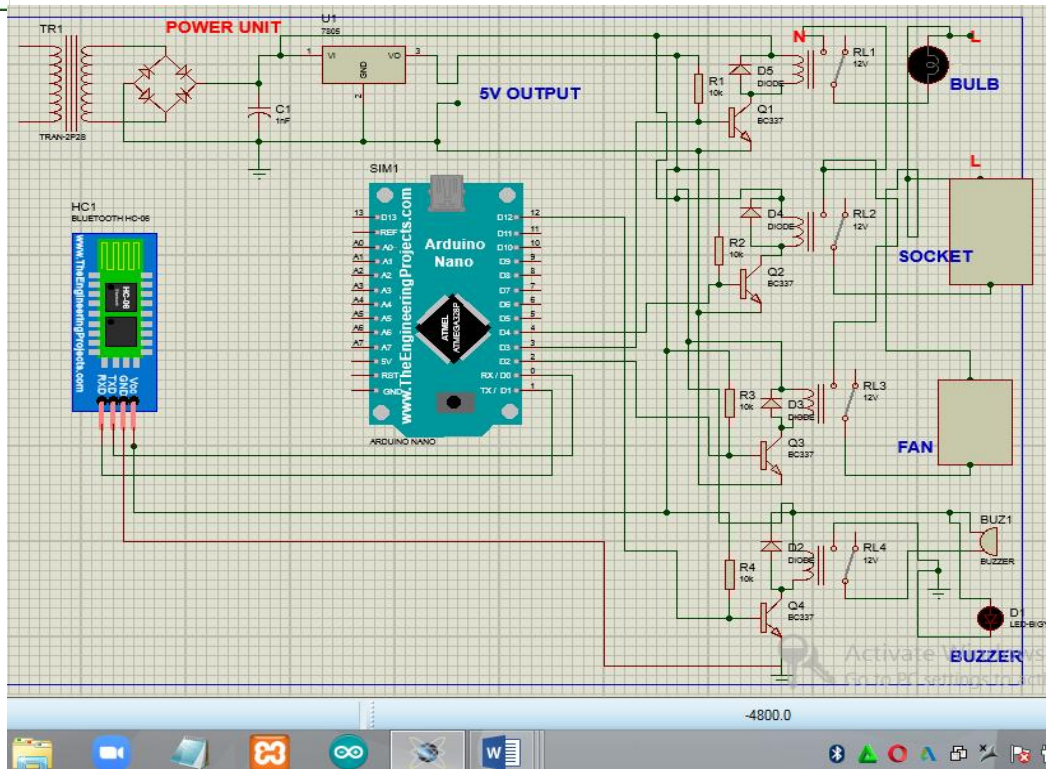

Figure 8: The Circuit Diagram of the Developed System

IV. RESULTS AND DISCUSSIONS

4.1 System Analysis

This research was implemented using the hardware and software approach. This was achieved by using a Bluetooth module (HC-06) to communicate between the hardware system and the smartphone as shown in the circuit diagram of Appendix A. The software part was based on an android app created using MIT App Inventor 2, which is an easy-to-use online platform used to develop the android application. The entire system is controlled by two devices, namely the microcontroller in Arduino board and an android phone in a wireless network known as Bluetooth.

4.2 Operational Architecture

The operational architecture composed of five major sections namely, microcontroller unit, notification unit, relay and triac unit, communication channel, and power supply unit as shown in figure 9. The input, which can either be a graphical user interface (GUI) command or a voice prompt (if included in the design) but not both, serves as input to the mobile smartphone through the Android app. The received command is transmitted using the Bluetooth, which is inbuilt in the mobile smartphone.

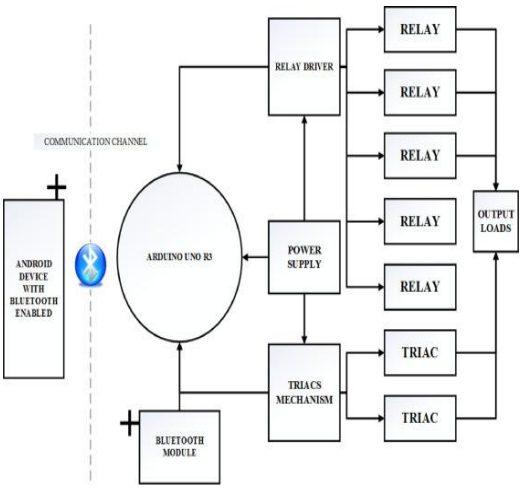

Figure 9: The System Operational Architecture

From the system's Bluetooth module, signal is communicated to the Arduino board. The microcontroller that is attached to the Arduino board controls the relay for the various

operations. The control operations include the switching of the bulb, fan, etc.

4.3 The Android App

The android App was developed on an android phone using Android Programming Language (MIT). The App is shown in figure 10.

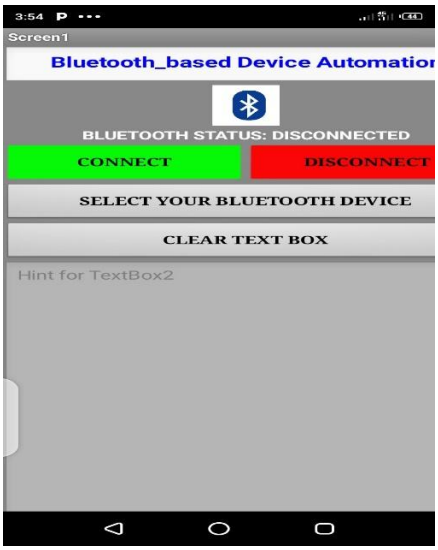

Figure 10: The Android App

4.4 Principle of Operation (How the System is Used)

1. Power the system
2. Put on your phone's bluetooth
3. Open the android app on your cell phone
4. Click on select your bluetooth device
5. Select hc06 (the bluetooth in hardware system)
6. Click on connect on the app
7. Then click on or off to control any of the devices of your choice
8. Click disconnect when you are done.

4.5 Result

When the above operations were carried out as in section 4.4, the system was observed to be working very well; all the devices were being controlled. The range of appliance control is the Bluetooth range of 45 meters on a line of sight and not an SMS range, so there is nothing like network problem during operations. The frequency of the Bluetooth is 2.5 GHz (type 2); and covers 45 meters on a line of sight and 10 meters on non-line of sight with wall penetration.

4.6 Discussions

This study demonstrates the design of an automated system to assist and provide support in order to fulfill the needs of the physically disabled and elderly ones in controlling electrical appliances in homes and industries. The heart of the control system is the wireless Bluetooth technology which provides remote access from a smart phone to a Bluetooth module and then to a control and switching unit. Bluetooth wireless connection enables the system to communicate with a Bluetooth application on an Android smart phone without cable. The user can easily touch on the phone screen to control the home appliances linked to the system. This method which requires minimal human effort is able to assist the disabled, sick and elderly people who have difficulties with locomotion. Thus, a control system has been designed which offers a low-cost design, installation ease, quick response time and a user-friendly interface in controlling home electrical appliances.

4.7 Testing

Testing is necessary in determining if the circuit meets the desired purpose for which it was constructed as well as for optimization. The sub-circuits of this system were tested module by module as it is being integrated. The implementation of both the hardware and software was done initially on a Bread board. To determine whether the different components are working fine, all the components were tested using a digital multimeter to check that they conform to their datasheet.

The tests carried out include a relay switching test, which is to ensure that the relay switches as expected that the desired signal went through the android application; timing test was also done since the android application was developed to include a time picker which can activate or de-activate the electrical appliances connected to the system. Other tests include:

- All the components were checked to see if they are okay before they were used in the circuitry.
- The power supply was properly checked to see if it falls within the tolerance

theoretically so as to avoid damage to the components.

- The components which were not functioning properly were changed as soon as possible to avoid other components being affected by the damage.
- Testing equipment was in proper range of the output measured at any point of the circuit, or component so as to avoid wrong readings.

V. SUMMARY AND CONCLUSION

5.1 Summary

The research work has been completed, with the initial design of the block diagram of the system. This was followed by the design of the circuit diagram using Proteus software (ISIS package). A program flowchart (algorithm) was developed, and the system was programmed using two programming languages. One was MIT programming language (an aspect of Android programming language) for the android APP development (for GUI phone usage), while the other one was the Arduino programming language for the device control using Arduino board. The operational range of the appliance control is the Bluetooth range of 45 meters on a line of sight and not an SMS range, so there is nothing like network problem during operations. The system was later tested module by module, and all the control operations were working perfectly. The frequency of the Bluetooth is 2.5 GHz (type 2); and covers 45 meters on a line of sight and 10 meters on a non-line of sight with wall penetrations.

5.2 Conclusion and Future Works

A home automation system using Bluetooth and an android application has been designed and implemented. That is to say that this home automation system uses an Android application and a Bluetooth technology in the design. Thus, the control system offers a low-cost design, ease of installation, quick response time and a user-friendly interface in controlling home electrical appliances. It is a highly reliable home automation system that can assist handicapped/old aged people, as well as a user-friendly device. Other features can be added to this system in the future, such as biometrics so that unauthorized persons cannot have access to the control system, and also a timing schedule can be developed for each appliance connected; this will effectively conserve energy.

REFERENCES

Ehiabhili J.C., Ezeh C.V., and Orji O.V. (2018), "Single Phase Microcontroller-Based Changeover Switch", International Journal of Electronics, Communications & Instrumentation Engineering Research and Development, Vol. 8(1), pp 7-16.

Greichen J.J. (2022), "Value Based Home Automation or Today's Market", IEEE Transactions on Consumer Electronics, vol. 38, no. 3, pp.34-38.

Jubadi W.M., Zulkifli N. (2019), "Programmable Infrared Accessory Light Switch",
Int. Conf. Intell. Adv. Syst., Kuala Lumpur, pp. 1130-1134

Kanma H., Wakabayashi N., and Kanazawa H.I. (2020), "Home Appliance Control System over Bluetooth with a Cellular Phone", IEEE Transactions on Consumer Electronics, vol. 49 (4), pp.1049-1053.

Nisar K., Ibrahim A.A.A., Wu L., Adamov A., and Deen M.J. (2016), "Smart Home for Elderly Living Using Wireless Sensor Networks and an Android Application", *Proceedings of the IEEE 10th International Conference on Application of Information and Communication Technologies (AICT).* IEEE Trans. Smart Grid, 3 (4) (2012), pp. 2262-2272, 10.1109/TSG.2012.2209130.

Ramlee R.A., Leong M.H., and Singh R.S.S. (2021)", Bluetooth Remote Home Automation System Using Android Application", International Journal of Engineering and Sciences, vol. 2. pp. 149-153.

R. Teymourzadeh (2018), "Smart GSM Based Home Automation System", Proceedings of the IEEE Conference on System (ICSPC2013), Kuala Lumpur, Malaysia (2018).
Sharon P., and Mahesh J. (2019), "Home Automation System (HAS) Using Android

Mobile Phone", International Journal for Electronics and Computer Science Engineering, Vol. 3 (1) pp. 1-10.

Shiu K. (2018), "Ubiquitous Smart Home System Using Android Application", International Journal of Computer Networks & Communications (IJCNC), Vol. 6, No.1.

