A SPEECH ACTIVATED CONTROL SYSTEM FOR INFRARED APPLIANCES.

A PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE BSC(HONS) IN COMPUTER SCIENCE.

BY:

ODEMWINGIE OGHOGHO GEORGE 16/1490

PAPKA DZIRAM YUSUFU 15/2148

TO:

THE DEPARTMENT OF COMPUTER SCIENCE.

SCHOOL OF COMPUTING AND ENGINEERING SCIENCES.

BABCOCK UNIVERSITY.

ILISAN-REMO.

OGUN STATE.

NIGERIA.

APRIL, 2019.

DECLARATION

We declare that this project **“Speech activated control system for infrared appliances”** was carried out by the undersigned individuals of the Department of Computer Science, School of Computing and Engineering Sciences, Babcock University.

Oghogho Odemwingie George 16/1490

……………………………………………….

Dziram Papka Yusufu 15/2148

……………………………………………….

CERTIFICATION

I certify that this project was carried out by ODEMWINGIE OGHOGHO GEORGE, PAPKA DZIRAM YUSUFU under my supervision in the department of Computer Science and School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Odemwingie Oghogho George 16/1490

Papka Dziram Yusufu 15/2148

…………………………………… ………………………………..

Ajaegbu C. PhD, *MCPN* Date

Project Supervisor

…………………………………… ………………………………..

External Supervisor Date

…………………………………… ………………………………..

Prof. Adekunle Y. A. Date

Head of Computer Science Department

DEDICATION

To our parents for continuing nurture and support as well as our teachers for guidance that has made this possible.

ACKNOWLEDGEMENT

We want to appreciate our Family and friends for the unending love and support they showed us throughout the course of this project especially during the difficult times.

Our Gratitude goes to our supervisor Dr. Ajaegbu C. for taking his time in putting us through the right part and ensuring that everything we did in this project was up to standard and for the best in other to achieve success.

We also want to appreciate our lecturers, the school of computing and Engineering Sciences and Babcock university as a whole for giving us the platform to learn and get standard education, for impacting in us knowledge both spiritually and academically. We are are very grateful.

ABSTRACT

Infrared control is a technology that has been around for over half a century and involves the use of infrared signals, naked to the human eye, to transmit control code to appliances. It has applications in several areas of the world to control heavy or small machinery and appliances.

Speech recognition refers to the process by which language patterns can be inferred and interpreted from a given audio signal. This process is used in a wide variety of applications in modern day – from complex IoT systems to simple dictation applications. The popularity of IoT over the last two decades has made operating numerous devices from a central location easier and more seamless. However earlier infrared controlled appliances cannot benefit from this technology as they do not possess an interface to connect to the network which is necessary for IoT.

The major objective of this project is to provide support for earlier infrared appliances so they can benefit from current speech recognition technology. This is done by implementing a central control system which can receive and process commands in the form of speech inputs as well as transmit infrared signals for device control.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Home automation today incorporates the latest technology ranging from IOT to speech recognition to simple motion sensors. However, a majority of these systems require the devices to exist on the same wireless network. Older model devices do not possess the components necessary to connect to and communicate with these networks. Thus, users of older model devices cannot benefit from the ease of using these new technology.

Speech control is a field in Artificial Intelligence which although appears to be a new field to the world is not new to the world of Computer Science. It has been in existence since the 1950's which was first seen in a system "Audrey" the very first known speech recognition system. This system was built in Bells Laboratories in 1952 by Davis, Biddulph, and Balashek. Audrey had the ability to recognize numbers from zero to nine and could only understand this coming from a familiar voice (Pinola, 2011).

The evolution of technology in Artificial Intelligence creates opportunity for even greater control over these devices. Functionalities such as speech recognition and Natural Language Processing have made it possible for users to interface with systems by just speaking to it (Pinola, 2011).

Speech control has since advanced from the methods used for voice control and synthesis since the 1950's. now in our daily lives we see applications of voice recognition and control from our soft copy dictionaries to the iPhone Siri.

In recent years, Speech control has been implemented in IoT devices. IoT devices have been popularized over the last few years from application in complex systems such as cars to the simpler ones like blenders. These systems ensure control of these devices can be controlled even when human agents aren't available (Brun, 2017).

IoT has applications across broad spectrum of industries, most especially smart homes. Home appliances such as the TV, AC, and decoder can now be controlled remotely.

1.2 PROBLEM STATEMENT

Infrared technology is still prominent in modern day although an argument can be made of its lack of importance given the availability of more secure and less hazardous means of control in modern day. Nevertheless, it still possess applications in modern day control systems.

Over the last two decades, the use of IoT has become more prominent in mainstream. It is now implemented in a lot of homes, offices and schools in both developing and developed countries. This popularity is brought about by the development of Artidficial Intelligence technology such as speech and voice recognition, automation, and so on which enable signals to be sent, received and processed through a plethora of media (Pinola, 2011).

With the evolution of speech recognition technology and the impending paradigm shift from infrared control, what then happens to the older infrared signal controlled appliances without the ability to take advantage of this evolving technology?

Older model infrared appliances cannot benefit from modern day speech control as they do not possess an interface to receive and process speech input and do not have the ability to connect to a local area or wireless network to fully benefit from this technology.

1.3 AIM & OBJECTIVES

This project aims to design and implement a voice controlled system for IR devices.

The specific objectives include:

i. To design a system for integration with already existing devices

ii. To implement the designed

iii. To create a mobile interface to control the database

iv. To integrate a natural language processing system with the control system

1.4 METHODOLOGY

The prototype model is the ideal choice for our study process as a working build of the project with limited functionalities can be implemented quickly and the workings of the project can be analyzed. It provides concise documentation on the various aspects of the system. This methodology ensures that a working system is made readily available, albeit not with all the initial functionalities speedily. A list of processes to be considered comprise:

- A study of similar projects as well as their limitations
- Steps taken to counteract these limitations
- Requirement determination
- Implementation of an error free system

Programming languages to be used in the implementation of our study include:

- Arduino
- JAVA
- MIT App Inventor

Hardware Components to be used in the implementation of out study include but are not limited to:

- Arduino Nano
- Bluetooth Module (HC-05)
- IR Transmitter

1.5 SCOPE OF THE STUDY

The voice controlled automated system takes the Babcock University classroom as its case study. This is because the classroom has devices that can be integrated with the system. The classroom also provides an avenue to test and demonstrate the system’s capabilities.

1.6 SIGNIFICANCE OF THE STUDY

This study aims to provide an interface by which various existing Infrared devices and systems can be controlled using current voice processing technology.

Infrared devices are used in various environments such as homes, hospitals, schools etc. these devices control a spectrum of appliances ranging from heavy assembly line machinery to the common television in homes. These devices however do not have a centralized system from which they are controlled. This has necessitated the need for the infrared control system:

IN THE HOMES: The average modern home is surrounded with a lot of infrared devices. Having multiple remote controls in a home with toddlers might be dangerous as they can choke on them. In addition, having multiple devices with similar remote poses a level of redundancy which our system aims to solve.

IN THE CLASSROOM: Modern classrooms use various IR devices like projectors, smart-boards, etc. The voice controlled IR system enables a lecturer control these devices that may not be located in close proximity to their lecture stage

1.7 LIMITATIONS

- Certain Infrared devices require the infrared signal to be encoded by the proprietary encryption and thus cannot be controlled by the raw signal.
- The system is microcontroller based and thus must be handled carefully to prevent shifting of the components.
- Not all infrared device types and manufacturers are currently supported by the system.

1.8 OUTCOME

The expected result of this project is to construct a physical control device as well as an android application. The android application shall possess the ability to receive speech input from users and send corresponding control code associated to the particular input to the physical control device through Bluetooth. The control device shall possess the ability to receive control data from the android device through a Bluetooth module, convert it to a transmittable format and send infrared signals to control infrared devices through the infrared emitter.

1.9 VOICE ACTIVATED CONTROL SYSTEM TERMINOLOGIES

- **Speech Recognition**: This refers to the means by with the device analyses sound to differentiate between human speech and noise.

- **Infrared Signal**: This is a pulse of infrared light rays carrying a particular instruction
- which controls what the device should do.
- **Effectors**: These refer to the output element as well as other related devices that are used to perform the functions of the system after processing input.
- **Sensor**: This is the device which receives/detects the inputs to be fed into the system for processing.
- **Controller**: This refers to the control unit of the system. It is in charge of processingthe input signal and outputting the relevant data.

1.10 ORGANIZATION OF PROJECT

CHAPTER ONE: INTRODUCTION

This chapter comprises the background of the study, aims, objectives, problem statement, significance of the study as well as a brief review of the methodology to be employed.

CHAPTER TWO: LITERATURE REVIEW

A catalogue of related projects are examined to determine potential limitations as well as proposing solutions to counteract these limitations.

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

Discussion of the various tools such as programming languages and hardware components to be used in the implementation of the study. Also, a visual depiction of the workings of the system is presented.

CHAPTER FOUR: IMPLEMENTATION

System implementation techniques and processes will be discussed in this chapter in relation to output. Testing of the system is also done.

CHAPTER FIVE: SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Results, inference from methodology and recommendations will be presented here.

CHAPTER TWO

LITERATURE REVIEW

2.1. INTRODUCTION

In our daily activities, we get to use infrared appliances and gadgets that are being controlled by remote controls. With the advancements in speech recognition technology and home automation, compatibility between older model IR devices and Newer technologies should be of paramount concern in a society bolstering ease of access to services.

In the classroom environment, there are various IR devices such as Air Conditioners, Smart Boards, Projectors. These devices have dedicated remote controls, this Implies having to deal with several remote controls and constant replacement and usage of batteries. Thus, a central control system is needed to combat this limitation. This project aims at creating a single Voice controlled remote for multiple infrared devices with the use of mobile phones.

Internet of Things is a network of interconnected devices that communicate and share data over the internet and can be controlled with minimum human intervention. IOT is a concept that although isn't new, but just recently gained popularity in the 2010s. the terminology internet of things was created in 1999 by Kevin Ashton. Ashton's idea which was part of the beginning of IOT was focused on a technology based on Radio Frequency Identification (RFID) for connecting devices together but the modern IOT has since changed since it uses the internet and IP networking to connect devices. Even as the world advances, technologies that do not have the ease or capability of connecting and bring controlled over the internet exist and this brings us to the discussion of infrared controlled devices (Lueth, 2014).

Infrared is derived from a Latin word "infra" that means "below". It is electromagnetic radiation with longer wavelengths. Due to it having longer wavelength than all the colours in the electromagnetic spectrum, it has a lower frequency hence "below". Infrared was first discovered in the early 19th century by Sir William Hershel. Hershel being aware of newtons discovery that sunlight could be separated into chromatic components via refraction through a glass prism. He conducted an experiment to measure the temperatures of the various colours of the spectrum and noticed that the highest temperatures came from colours beyond the red portion of the spectrum. This experiment lead to the discovery of infrared light (Rowan-Robinson, 2013).

Infrared is a section of invisible light found on the electromagnetic spectrum, it is a form of radiant heat called infrared energy. The electromagnetic is divided into three segments by wavelengths measured in microns.

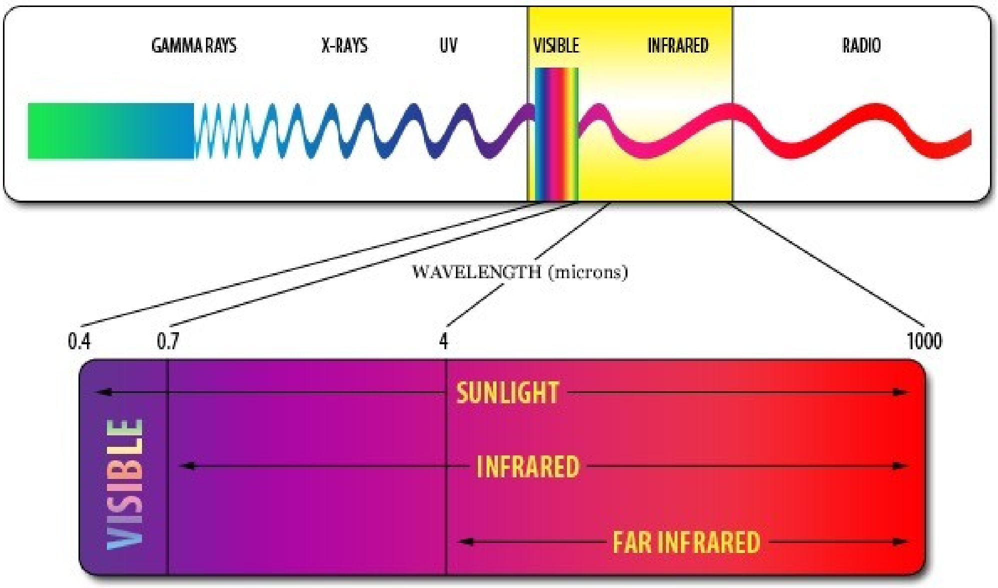


Figure 2.1: Electromagnetic Spectrum
(Source : BiosmartSolutions, 2012)

Infrared radiation is emitted or absorbed by molecules when they change their rotational vibrational movements. Infrared radiation starts at the colour red from 700 nanometres to 1 millimetre (Liew, 2001).

Infrared light is not visible to the mare human eyes but can be seen through special cameras that translate light into colour

Light comparison[9]			
Name	**Wavelength**	**Frequency (Hz)**	**Photon energy (eV)**
Gamma ray	less than 0.01 nm	more than 30 EHz	more than 124 keV
X-ray	0.01 nm – 10 nm	30 EHz – 30 PHz	124 keV – 124 eV
Ultraviolet	10 nm – 400 nm	30 PHz – 790 THz	124 eV – 3.3 eV
Visible	400 nm–700 nm	790 THz – 430 THz	3.3 eV – 1.7 eV
Infrared	700 nm – 1 mm	430 THz – 300 GHz	1.7 eV – 1.24 meV
Microwave	1 mm – 1 meter	300 GHz – 300 MHz	1.24 meV – 1.24 µeV
Radio	1 meter – 100,000 km	300 MHz – 3 Hz	1.24 µeV – 12.4 feV

Table 2.1: Light Comparison

(Source : Haynes, 2011)

Objects emit infrared radiation across a spectrum of wavelengths but only a limited region of spectrum is of interest because sensors collect radiation within a specified bandwidth. Thermal infrared radiation also has a maximum emission wavelength which is inversely proportional to the ablsolute temperature of the object. Infrared bands are subdivided into five sections which are;

- Near-infrared: this has a wavelength of 0.75 to 1.4 micrometer, a temperature of 3591-1797celcius and a frequency of 886 to 1653TeraHertz
- Short-wavelength infrared: this has a wavelength of 1.4 to 3 micrometer, a temperature of 1797-693 and a frequency of 100 to 214 Terahertz
- Mid-wavelength infrared: this has a wavelength of 3 to 8 micrometer, a temperature of 693-89 and a frequency of 37 to 100 terahertz.
- Long-wavelength infrared: this has a wavelength of 8-15 micrometer, a temperature of 89-80 and a frequency of 20-37 terahertz.
- Far-wavelength infrared: this has a wavelength of 15-1000 micrometer, a temperature of -80.15 - -270.15 and a frequency of 0.3-20 terahertz.

2.2. EVOLUTION OF THE REMOTE CONTROL

The invention of the remote control was in the 1950s. Prior to this invention, appliances where operated directly from buttons on the body of the device.

The Zenith Lazy Bones Remote control was the very first television remote control created by zenith. it was a wired remote connected to the Television and had a switch to turn on or off the television, together with a few other buttons. This invention triggered other companies to start creating remotes. This remote control was preceded by the Flash Matic remote (Farhi, 2007).

The Flash matic Remote After the Lazybones remote control Zenith created the first wireless remote called the Flash Matic that looked like a hand Gun., it had basic features such as a switch on and off button, a volume regulator and to change channels. It used a beam of light flashed towards a photo electric cell on the television, this had limitations because light from other random sources could activate controls.

The Zenith space command Remote was the third remote invented. This remote used sound waves as opposed to light. It debuted in 1956 and had more buttons and it had a mechanical controller that clicked and struck a bar when a button was pushed hence it was called the cliker. This was the first remote that enabled users zoom and adjust the color of their Television (Gertner, 2007).

The Modern Remote came to play in the 80's. The growth of cable television and how common they became led to an up in the innovation of remotes as well. Remotes began to use infrared technology and in the 80s they became very common.

The first work done on building a universal remote was in the mid 1980s by Steve Wozniak. He thought having multiple remotes to control the several devices he had was unnecessary so he decided to come up with the universal remote which was successful. In 1987 the CL 9 CORE was released but limitations to this invention was a big deal for the average customer because it had 256 different codes for different devices to be controlled (Wozniak, 2006).

2.3. HISTORY OF SPEECH RECOGNITION

Speech recognition was implemented first in 1952 when Bell laboratories invented 'Audrey' which could only comprehend numbers. During the 1950s to 1960s era, the technology was further advanced with the invention of the 'shoebox' in 1962 which could understand 16 words in english. Voice recognition was then enhanced to comprehend 4 vowels and 9 consonants (Pinola, 2011).

In the 1970s, the US government contributed heavily to the development of speech recognition systems and from 1971 t0 1976, it funded the DARPA SUR(Speech Understanding Research) program which led to the birth of ‘Harpy’, a system with the ability to comprehend 1011 words (Pinola, 2011).

The 1980s entailed a major breakthrough as the hidden Markov model, a model which used statistics to predict the probability of a word, was developed. It did not depend on speech patterns or fixed templates. Speech recognition systems in this era however possessed a vital flaw – breaks had to be taken between each spoken word (Pinola, 2011).

Due to the introduction of faster microprocessors in the 1990s, speech software became more feasible. Dragon, in 1990, released ‘Dragon Dictate’ which was the world’s first speech recognition software for consumers. This was further improved in 1997 in ‘Dragon NaturallySpeaking’ which could recognize 100 words in a minute (Pinola, 2011).

By the start of the 2000s, speech recognition technology development had hit a wall until Google came along with an application called ‘Google Voice Search’ which employed data centres to perform the large amount of analysis needed for matching the speech inputs with user queries. In 2010, Google released a personalized recognition on android devices to develop an enhanced speech model comprising over 230 billion English words. Eventually current personal assistants such as Siri, Alexa and the Google assistant were invented which rely on cloud computing as well (Pinola, 2011).

2.4 REVIEW OF RELATED WORKS

Keval Pandaya, 2015, constructed a Voice controlled home automation system in a bid to create a cheaper automation system for appliances at home. It makes use of an Arduino Microcontroller, a Bluetooth module, a relay an android application and the google speech recognition. The circuit comprises an Arduino Mega 2560 microcontroller (Board1), for comparing the stored string with the input string received through Bluetooth to give output to digital pin 6 of Board 1 to control the relay. Bluetooth module HC-05 transmits and also receives data serially via Board1 that can be read by the Microcontroller. A relay is used here to control only one appliance the Arduino Mega 2560 has 54 digital pins, each of which can be used to control an appliance. The MCU is programmed to compare the relative strings.

Arduino IDE is used for programming the MCU. For developing Android app, MIT App Inventor, a web tool used to easily build android applications, is used.

The Android app uses Google's speech-recognition function and transmits the text to the MCU via the connected Bluetooth (HC-05).

Aravind Sasikumar, 2015, Constructed a Voice Controlled Smart Home. The aim of the project was to build a perfect companion for someone to be at home. It is a computer based system that can accept voice as well as direct commands and process them. The system has a control unit, a transceiver unit, an audio processing part and the software unit. The control unit used for the embedded section is an ARM based LPC2148 Microcontroller. All the devices on the network must have a switching circuit to toggle between an OFF and ON state. The status of each device is displayed on an LCD. The voice commands are captured and processed in MATLAB after the voice recognition method used identifies speech patterns. The captured speech is compared with the previously trained words in the database and recognizes the speech using appropriate algorithm. After the commands are successfully identified, the control characters are sent wirelesslyto the Microcontroller, which will in turn activate the corresponding relay. As a result, home appliances could be turned on or off depending on the voice command given ZigBee transmitter.

The status of each device is monitored by using toggle circuit. The control part will turn on or turn off the device according to the control characters.

Microcontroller lab, 2015, implemented this project aimed at creating a system and application that would be easy to use for people of any age. All components/units in the system include

- Arduino Uno: provides the much digital pin to interface with LCD, Bluetooth module, and relay module at the same time.
- HC-05 Bluetooth Module: HC-05 is used to communicate with the mobile phone.
- 4 Channel Relay Modules: The module used in this project is HL-54S. It switches on and off using a 5v logical signal from Arduino. It can bear up to 250VAC and 10A. These modules have 4 channels so 4 AC devices can be controlled at a time.
- 16×2 LCD: LCD is used to display a list of commands which can be entered then it asks to give any command and show the status of the command which is entered. the is used 16×2 LCD because it is easy to interface with Arduino and very cheap in price. A 10k potentiometer is employed to control the contrast of display
- AC bulbs with holders to represent appliances.

Predefined commands are spoken to an AMR_Voice application. The application sends the command to Bluetooth which is then received by Arduino and perform the described task. The Arduino displays the status on LCD and write on the serial monitor. Each command has its unique operations, which can be changed according to your ease, defined in the program.

Sonali Sen, 2015, Envisioned the Design of an intelligent Voice Controlled Home Automation System. The project aimed at creating a very easy to operate system that required minimum training and take advantage of the android platform because of its vast and versatile uses. The system components include an Android based phone, Bluetooth module, Arduino Uno, Relay boards. The voice recognizer which is an in built feature of Android phones is used to build an application which the user can operate to automate the appliances in the house .

The Bluetooth module allows the transmission and receiving of signals. It receives the text from the Android phone and transmits it to the serial port of the Arduino Uno.

The Arduino Uno is a Microcontroller board based on the ATmega328p The Arduino circuit acts as an interface between the software part and the hardware part of the project. The Bluetooth module transmits the text to the Arduino Uno serial port. The text is matched against the predefined texts to switch the appliances on or off. The appliance name and associated command for on/off are stored in the android application as predefined commands. When the matching text is detected the corresponding pin number is given a high or low output signal to switch the appliance on and off.

A relay is an electromagnetic switch n this project the relay circuit is used to turn the appliances on/off. The high/low signal is sent from the microcontroller. When a low voltage is given, it is turned off and when a high voltage is given, it is turned on.

The Microcontroller, Bluetooth module and relay circuit have to be attached with the switch board. After the android application is launched on the smartphone, Through the application the commands instruct the Microcontroller to switch on/off an appliance. After getting the instruction through the Bluetooth module the Microcontroller gives the signal to the relay board. The application first searches for the Bluetooth device. If it is available then it launches the voice recognizer. It reads the voice and converts the audio signal into a string. It produces a value for each appliance which will be given to the Microcontroller device. The Microcontroller uses the port in serial mode. After reading the data it decodes the input value and sends a signal to the parallel port through which the relay circuit will be activated.

B.V.S Sai Chaitanya, 2016, Implemented an IOT based Smart Device using CC32000. The objective of this project was to create a smart IR device to be able to control Basic home Gadgets from a central point using Android and IOS applications. The Smart IR device incorporates a CC3200 Wi-Fi module and has an IR transmitter Led and an IR receiver Module. The IR receiver Leds are put a specific pattern to cover a complete 360degree angle and the receiver receives the remote codes which are IR bit patterns from all different home appliances remotes and stores them in the CC3200 Wi-Fi module which will be controlled through android or IOS applications. The buttons on the remote have equivalent IR bit patterns that vary for different remotes.

Parameshachari B. D., 2013, proposed A Study on Smart Home Control System through Speech for an automation system where users can use voice commands to control their electrical appliances, such as light, fan, television, heater, etc. The project aimed to help make life for individuals, such as the elderly or physically disabled, easier as they do not need to be in physical interaction with the appliance to control it. This system utilizes Digital Signal Processor to process and voice commands and accordingly control the required appliances.

The output of the DSP will be sent through XBee transceivers, to the control part where a microcontroller will select the required device depending on the input received. The system has three parts:

- Audio processing part
- Transmission part
- Control part

The speech commands are received and captured in the DSP according to the voice recognition method used and after successful identification of the command, control signals are sent wirelessly through the Xbee transceivers to the microcontroller which in turn activates the corresponding appliance.

Below is the block flow diagram of the proposed project:

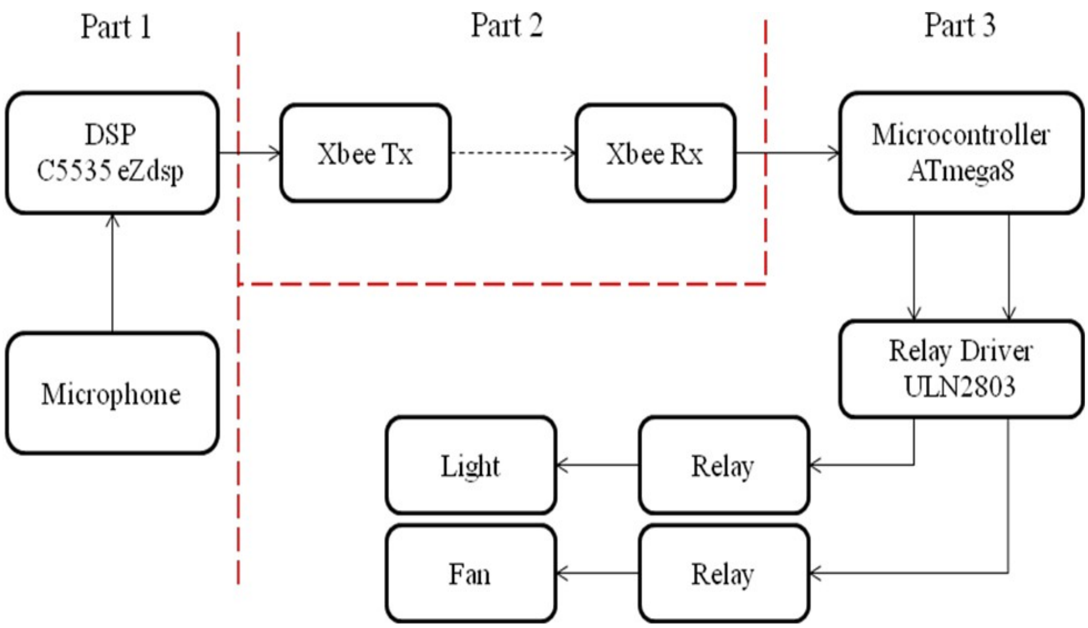


Figure 2.2: Block Diagram Of The Smart Home Control System
(Source : Parameshachari, B. D., 2013)

The flow diagram for the voice recognition program is given below:

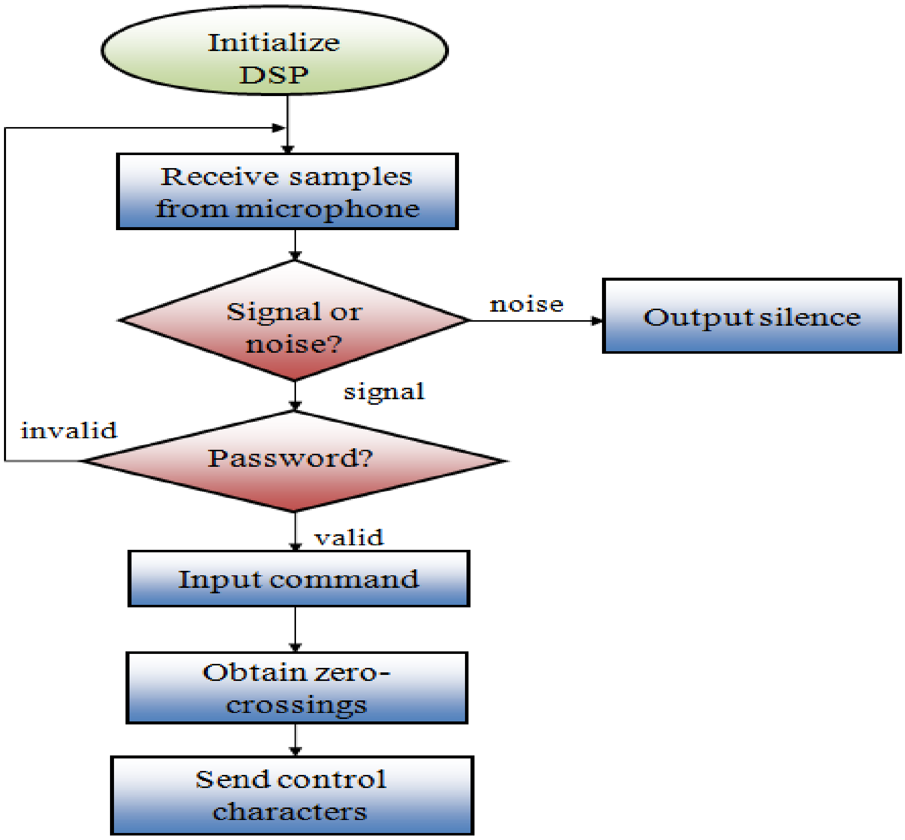


Figure 2.3: Flow Diagram For Voice Recognition Program

(Source : Parameshachari, B. D., 2013)

Awadalla Taifour Ali, 2017, built a Voice Recognition Based Smart Home Control System capable of being activated through voice commands. The voice commands were trained on the Easy VR2.0 voice recognition unit using the software named “Easy VR Commander” which gives an option of entering any custom voice command. The module is trained to recognize any command typed. There were two microcontrollers in this project which comprise: an ATNega328 IC which has an RF chip connected to it working as the transmitter to transmit signals and an Atmega16 IC which has an RF chip connected to it working as the receiver. The flow chart illustrating the sequence of functions of the Voice Recognition Based Smart Home Control System is given below:

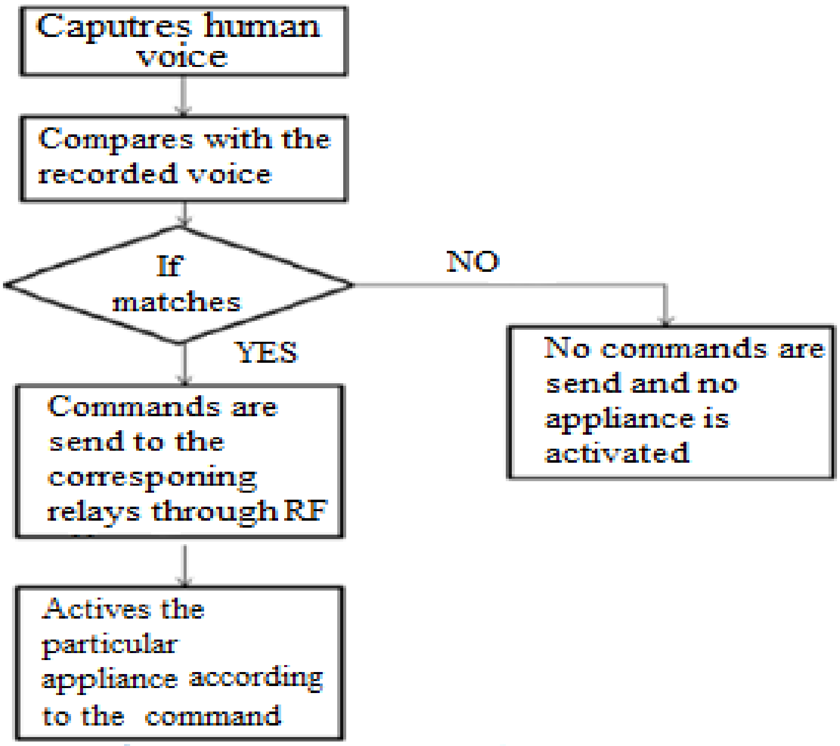


Figure 2.4: Flow Diagram Of The Voice Recognition Based Home Smart Control System

(Source : Awadalla, T. A., 2017)

Khaled Mahmud, 2015, imagined to build a Voice Activated Electronic Devices Control System For Home Appliances which would receive voice commands and then behave accordingly to control home appliances. A software was implemented to recognize every voice command of every person. From the voice command, a corresponding control signal is generated to be passed through the parallel port to the control circuit to control the designated appliance.

The voice command is the initiator in the system for all further controls and operations. A particular voice commands corresponds to a predefined control action. The figure below is a block diagram depicting the system‘s workings:

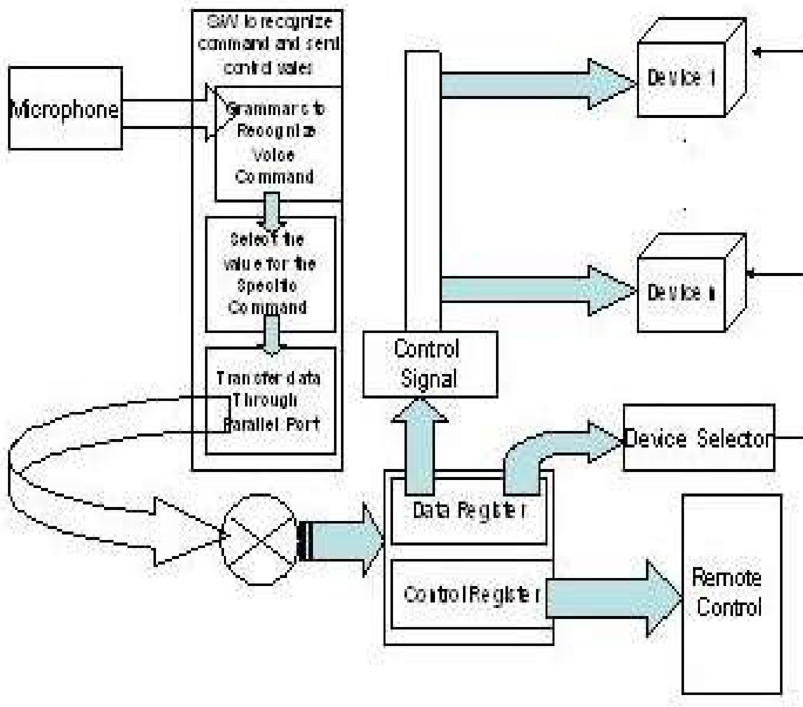


Figure 2.5: Block Diagram Of Voice Activated Electronic Device Control System
(Source : Khaled M., 2015)

The system is divided into two major functions: Detection of the voice and Control of the device. The first part receives voice commands in the computer through a microphone while the second part receives the converted command and controls the appliance accordingly. C# was used to recognize voice commands as well as passing data to port.

Cheng-Chi Tai, 2004, envisioned The Information Home Appliance Control System-- A Bluetooth Universal Type Remote Controller that will replace all remote controllers from various products, resolve the integration problem and provide an integrated control interface for various home appliances. Using the functions and characteristics of Bluetooth radio and voice communication, he developed a universal type remote control system combining wireless phones and home appliances inside a living household. The system was divided into a transmitting end and a receiving end. The transmitting end includes: keyboard, MCU and Bluetooth module. The receiving end can comprise multiple systems, including TVs, phones, air conditioners, sound and stereo systems, electrically controlled doors. The Bluetooth module was inserted into a regular interior phone set, replacing general traditional interior wireless communication function. Since the voice adopts PCM encoding and is transmitted with digital mode, the voice is very clear and its signal is stable and undisturbed. The Block flow diagram of the system is shown below:

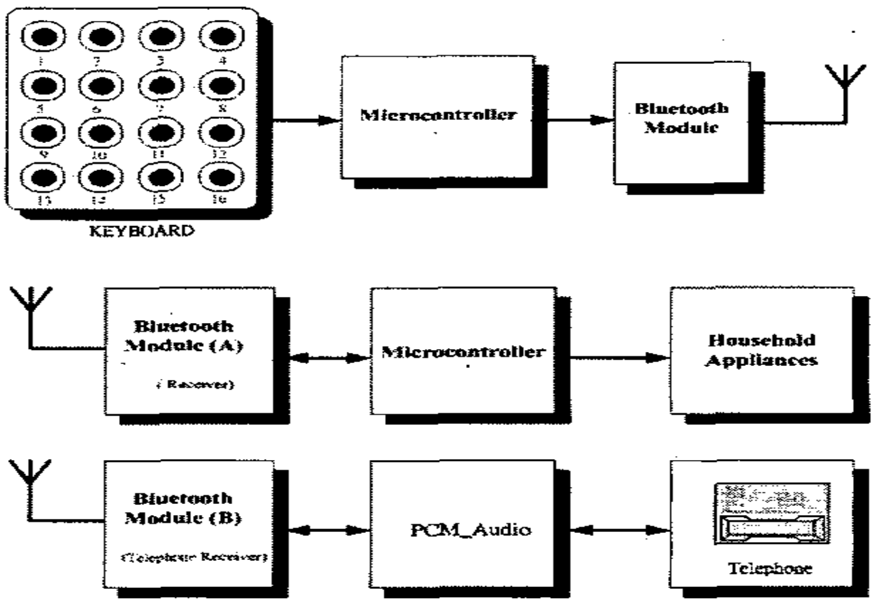


Figure 2.6: Block Flow Diagram Of The Information Home Appliance Control System -- A Bluetooth Universal Type Remote Controller

(Source : Cheng-Chi T., 2004)

CHAPTER THREE

SYSTEM DESIGN AND ANALYSIS

3.1. PREAMBLE

This chapter outlines the design and analysis of this Voice Control System for Infrared Devices and sheds more light on how the Voice Control module communicates with the Infrared emitter module to control the devices. The basic unit of the system are given below:

- Power Supply Unit
- Bluetooth Serial Module
- Infrared Emitter
- Control Unit
    1. Arduino Nano Microcontroller
- Android Application
    1. MIT App Inventor
    2. Android Speech-to-Text
    3. Tiny DB
    4. Tiny Web DB

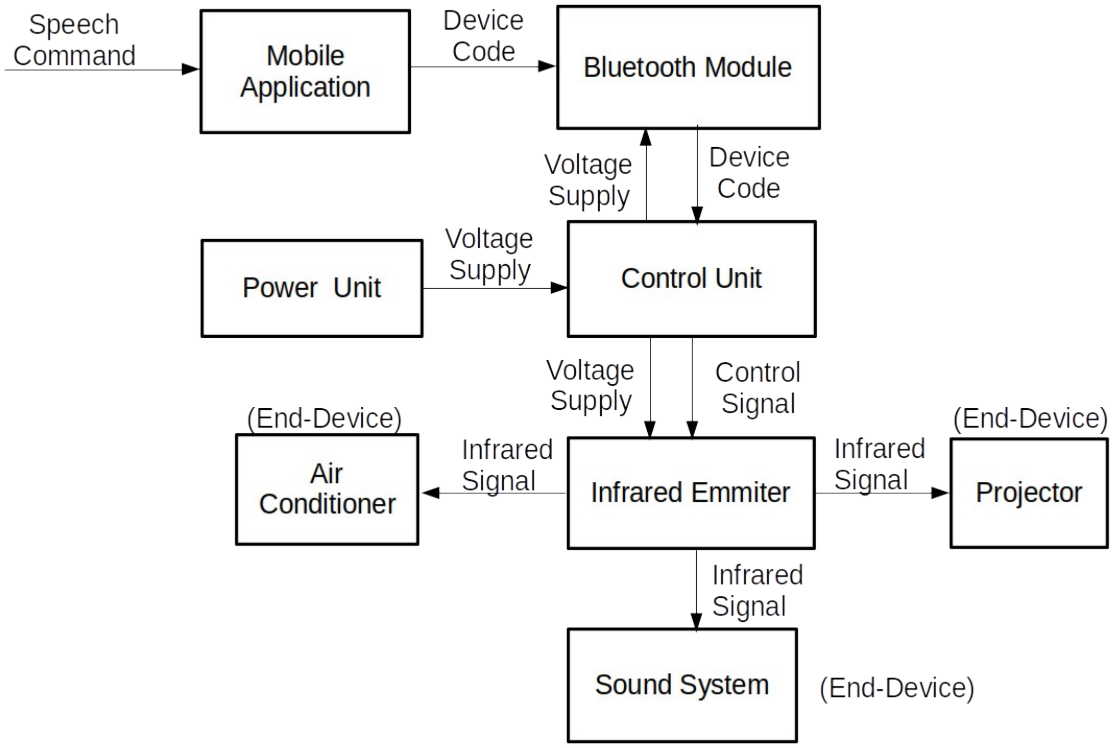


Figure 3.1: Proposed Block Flow Diagram Of The Speech Activated Control System For Infrared Appliances

3.2. POWER SUPPLY UNIT

All stages in the system require a +5volts DC supply except the Android application which is supported on the mobile device. This is achieved by connecting a 9volts DC battery to the V_{in} and GND pins of the microcontroller. A switch toggle is added to enable switching between an "OFF" and "ON" state. A schematic diagram of the power supply unit is shown in Figure 3.2 below:

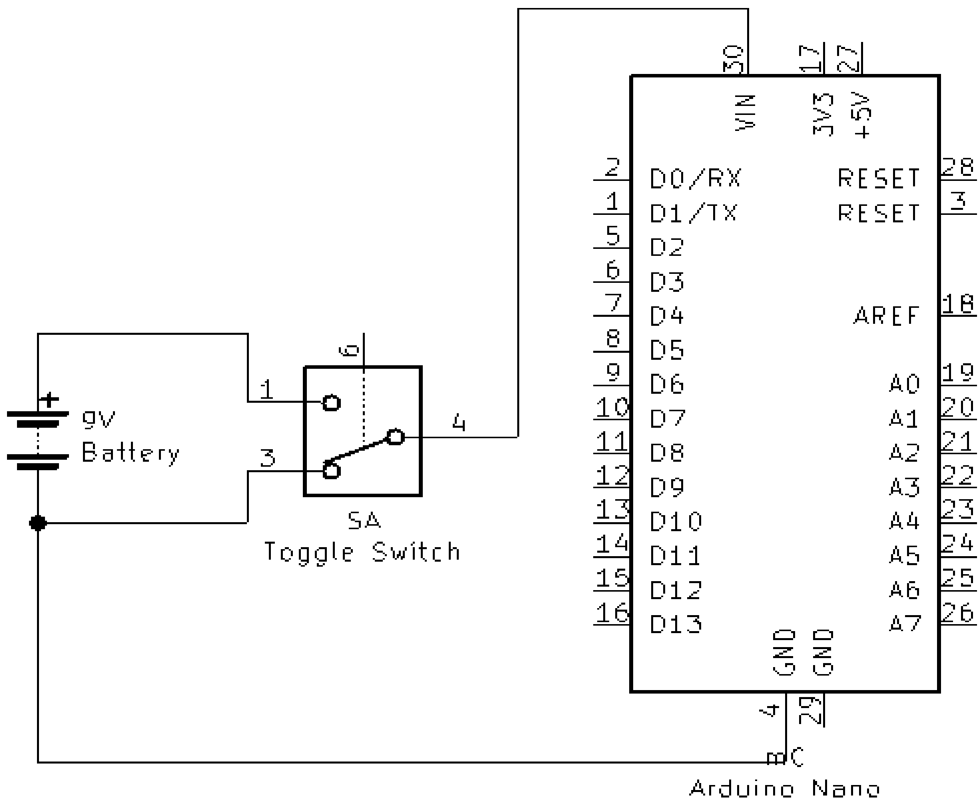


Figure 3.2: Proposed Schematic Diagram Of The Power Supply Unit Connection

3.3. BLUETOOTH SERIAL MODULE

This module is used to send data between the device and the accompanying android application. The Bluetooth module used for this device is the HC-05 Bluetooth Module. This type is an easy to use Bluetooth Serial Port Protocol Module for wireless serial connection setup. It is fully qualified V2.0+EDR (Enhanced Data Rate) 3Mbps modulation with complete 2.4Ghz radio transceiver and baseband. It operates with a default command mode Baud rate of 38400 and data mode Baud rate of 9600 with support for 19200, 38400, 57600 and so on. It has an operating current of 30mA and votage in the range of +4Volts to +7Volts. The HC-05 has a maximum connection range of 100m. The HC-05 employs a full-duplex connectivity which implies data can be transmitted and received concurrently.

The diagram below shows a 2-D model of the HC-05 Bluetooth module:

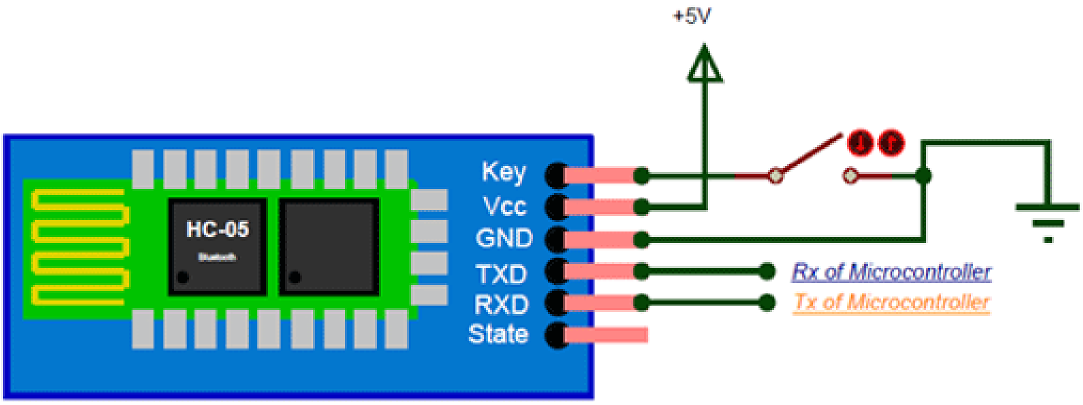


Figure 3.3: 2-D Diagram Of Hc-05 Bluetooth Module
(Source : Components101, 2019)

3.4. INFRARED EMITTER

The function of this component is to blast the controlling signal(s) to the various Infrared devices from the system. The controlling code is sent from the android application to the device, decoded and then blasted to the air by the Infrared emitter. The Infrared emitter used in the device is the Open-Smart Infrared emitter which is based on the HL-A838 integrated infrared receiver module. It operates at a frequency of 38Khz. It has a supply voltage in the range of +3.3Volts to +5.5Volts.

The diagram below shows the diagram of the Infrared emitter:

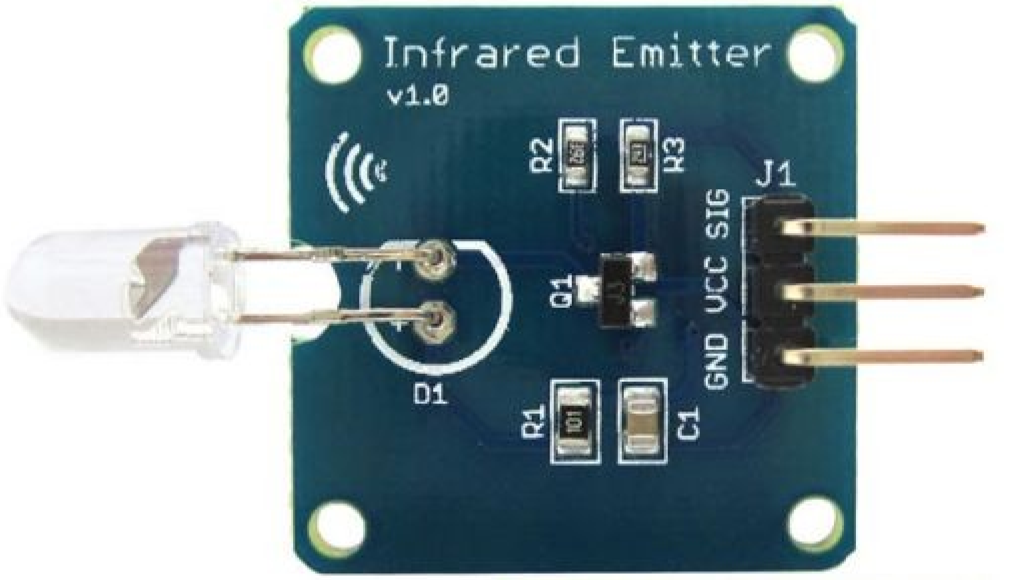


Figure 3.4: Image Of The Infrared Emitter
(Source : DealeXtreme, 2019)

3.5. CONTROL UNIT

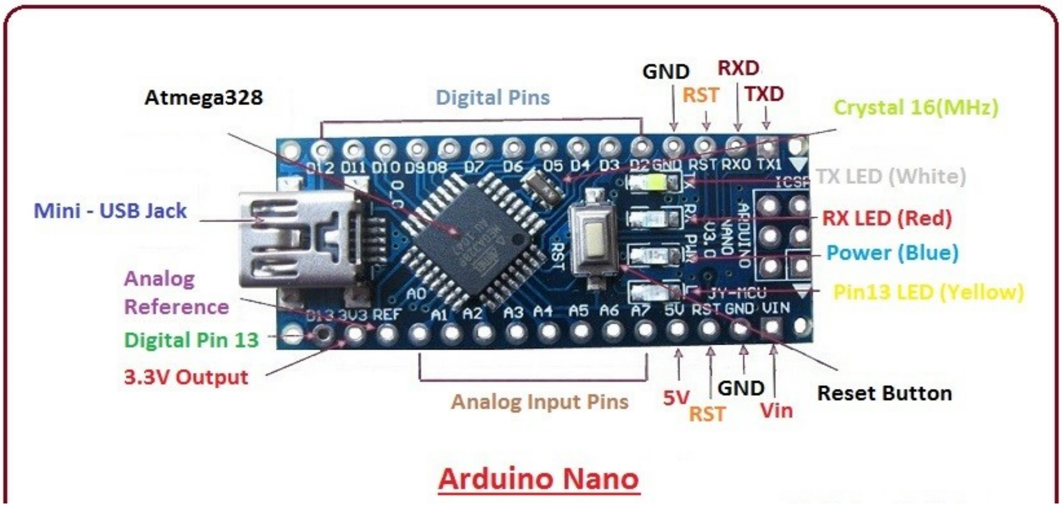
This unit processes the data received from the android application, processes it, and sends it to the IR transmitter to blast the signal. It is also tasked with regulating temperature if an AC unit is configured with the system. It consists of an Arduino Nano microcontroller with an ATMEL ATmega328 IC with a 32KB non-volatile Flash program memory that is parallel programmable.

A **microcontroller** (**uC** | **µC** | **MCU**) is a small computer on a single IC comprising a processor core, memory, and programmable I/O peripherals. Program memory in the form of NOR flash or OTP ROM is often included on the chip as well as a small amount of RAM. Microcontrollers are designed for embedded systems so that they can control the features or actions of the system.

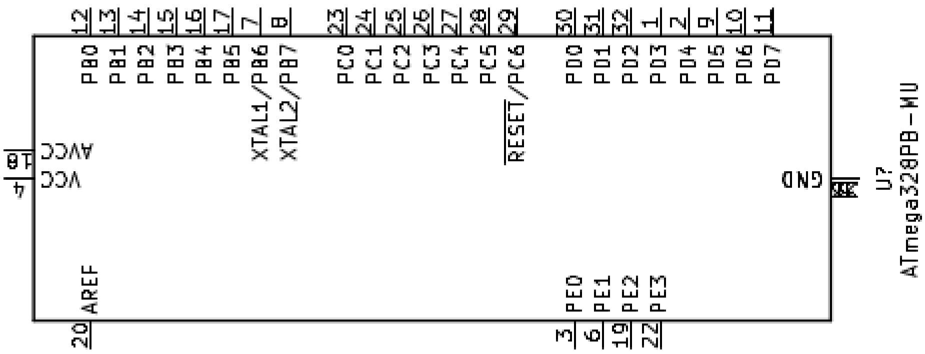
The basic features of the Arduino Nano are given below:

- Operating Voltage: +5Volts
- Input Voltage (limit) : +6V - +12V

- Flash Memory : 32KB (ATmega328) of which 2KB is used by the bootloader
- EEPROM : 1KB (Atmega328)
- SPAM: 2KB (Atmega328)
- Digital I/O ports : 14 of which 6 provide PWM output
- DC Current per I/O port : 40mA
- Analogue Input pins : 8
- Clock Speed : 16MHz



(a.)



(b.)

(c.)

Figure 3.5: (a.) Diagram Of The Arduino Nano (b.) Schematic Diagram Of The Atmega328p (c.) Schematic Diagram Of The Arduino Nano
(Source : TheEngineeringProjects, 2019)

3.5.1. USING THE MICROCONTROLLER

The Arduino Nano microcontroller is relatively straightforward to setup and use. Provided the Arduino desktop IDE is available, a simple Arduino code can be written, compiled and uploaded to the board from the IDE. The Arduino Nano has a V_{in} pin for connecting a voltage source to power the board.

Figure 3.6: Interface of The Arduino IDE

3.5.2. WRITING THE CONTROL PROGRAM

The Arduino programming language is based on the C++ programming language, so it is very easy to comprehend. A program written with the Arduino IDE is called a sketch and is saved on the development computer with a **.ino** or **.pde** extension. The structure of the program entails two main parts:

- Setup(): This function is called once and is used to initialize variables and I/O pin modes.
- Loop(): This contains the main body of the sketch. It includes possible function calls and other functionalities the sketch might have.

A simple Arduino sketch for blinking an LED is shown below:

```
#define LED 13
void setup(){
     pinMode(LED,OUTPUT);
}
void loop(){
     digitalWrite(LED,HIGH);
     delay(1000);
     digitalWrite(LED,LOW);
     delay(1000);
}
```

3.5.3. PROGRAMMING THE MICROCONTROLLER

The Arduino Nano sketch is uploaded to the board by connecting it to the serial (USB) port on the development computer. This is done using a USB type A Male to a USB mini B Male cable. The sketch is compiled by the IDE and then uploaded to the board.

3.6. ANDROID APPLICATION

The Android application is tasked with obtaining vocal command from the user and then sending the relevant data to the control unit through the Bluetooth module. The application is built using the **MIT App Inventor** software.

3.6.1. MIT APP INVENTOR

MIT App Inventor is an open-source web application originally provided by Google but now maintained by the Massachusetts Institute of Technology (MIT). It allows users to create software applications for the Android operating system. It is easy to use because it employs a drag-and-drop model for design and development. It also has an emulator feature for live testing of applications.

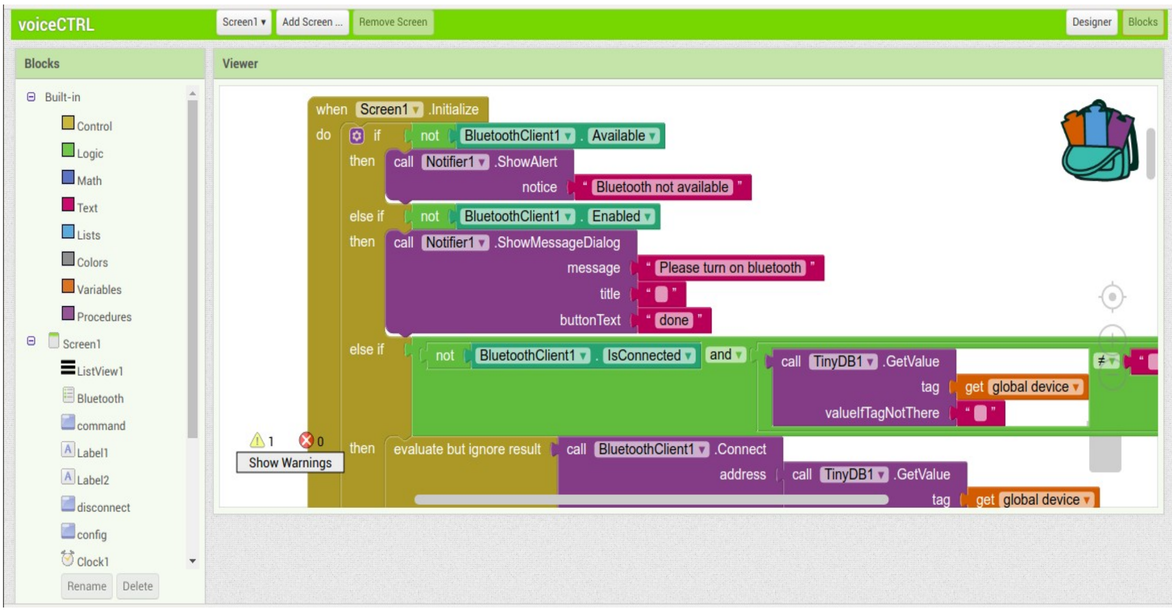


Figure 3.7: Block View Of Mit App Inventor

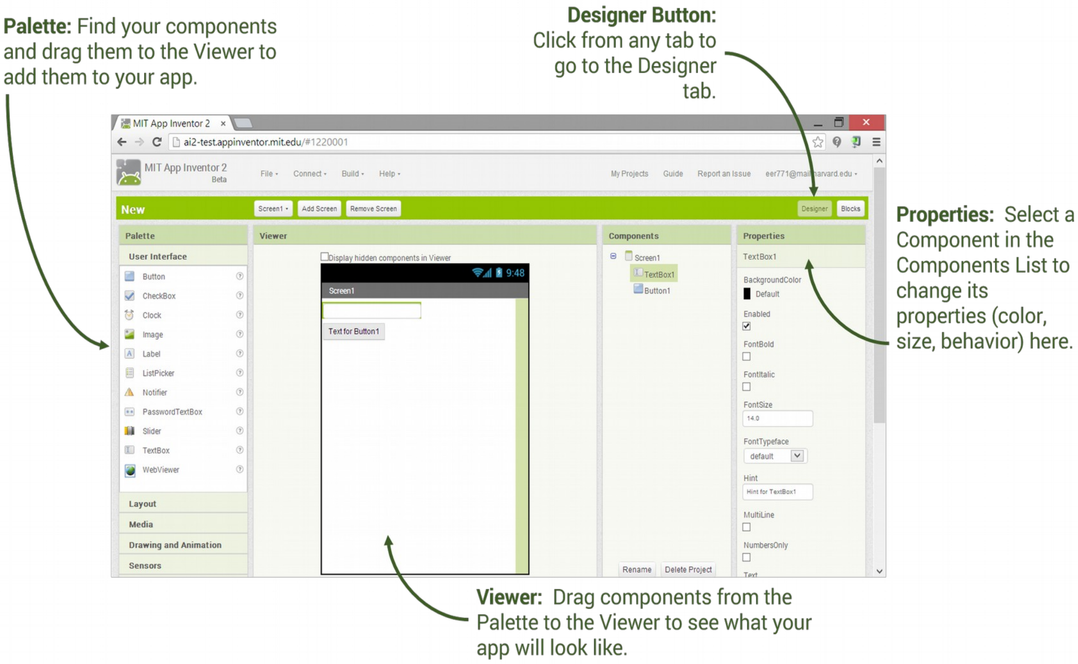


Figure 3.8: Design View Of Mit App Inventor

(Source : MIT App Inventor, 2019)

3.6.2. ANDROID SPEECH-TO-TEXT

The Android operating system comes with a built-in speech-to-text feature through which a user can provide a speech input and it can be converted to plain text. This is incorporated in the application to receive the voice command from the user which can then be sent to the control unit.

3.6.3. TINY DB

Tiny DB is a persistent data store for the application, the data stored here will be available each time the application is run. Data items are stored under tags. To store a data item, a tag is specified under which it will be stored. To retrieved stored data, its corresponding tag just has to be provided. This component is used to store the infrared code and corresponding voice command as data and tag respectively to facilitate the functioning of the application.

3.6.4. TINY WEB DB

Tiny Web DB is an App Inventor component that allows you to store data persistently in a database on the web. This component is used to store Infrared code for all the possible devices that can be configured for control on the application. The data is retrieved from the Tiny Web DB during configuration and stored in the Tiny DB after configuration.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND TESTING

4.1. INTRODUCTION

This chapter gives a precise documentation of the implementation of the speech activated control system. This chapter comprises development tools, algorithms and challenges encountered during the development process of the system.

4.2. DEVELOPMENT TOOLS

4.2.1. HARDWARE

- **Connecting cables:** These are used to connect the various components together in the circuit.
- **Bread-board**: This formed the base upon which the circuit was built. It is a solder-less board and multiple circuits can be placed on it.
- **Wire strippers:** These were used to remove the outer covering of wires to enable better connection.
- **Android device:** This was used to test the application during construction.
- **Infrared Learner:** This tool is used to receive and decode infrared signals. It was used to learn the infrared control signals for the various supported infrared appliances.

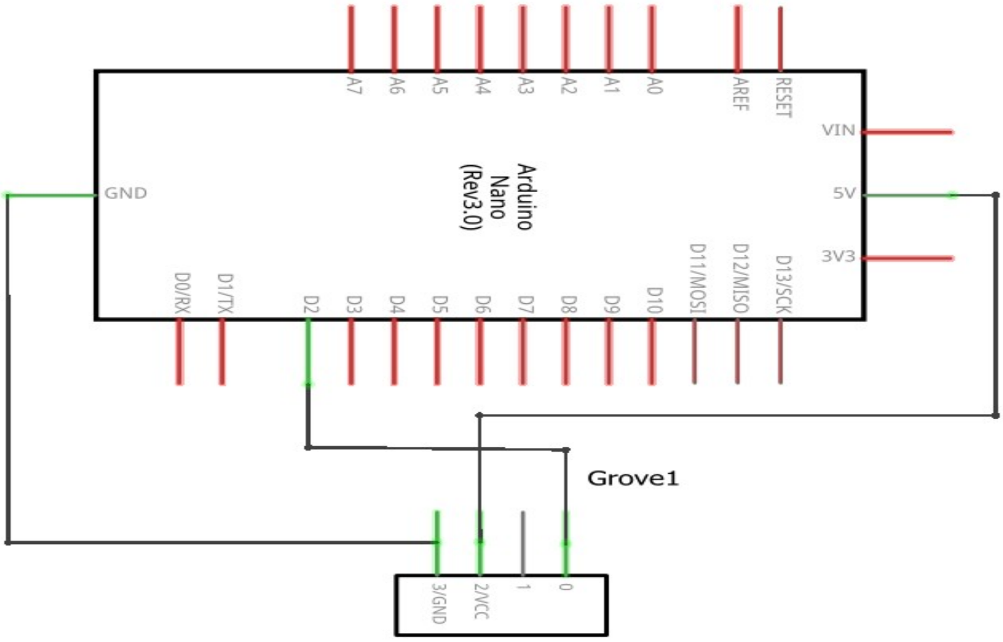


Figure 4.1: Proposed Circuit Diagram Of The Infrared Learner

4.2.2. SOFTWARE

- **Arduino IDE**: The Arduino IDE is necessary to program the microcontroller to perform the prescribed tasks.
- **MIT App Inventor:** This is used to create the Android application that receives the speech command and sends the corresponding code over Bluetooth.
- **Libraries:**

  StandardCplusplus : This library provides support for C++ data structure that are not supported by the Arduino programming language.

  Vector : data structure inherited from StandardCplusplus. Used to temporarily store the infrared signal.

IRLibSendBase : This library is needed fo initialization of the workings of the infrared emitter.

IRLib_HashRaw : This is responsible for facilitating the sending of the infrared signal through the infrared emitter.

4.3. SYSTEM REQUIREMENTS

4.3.1. HARDWARE REQUIREMENTS

- **Android device**: The Android device must have Bluetooth.
- **Memory**: 11MB storage space or higher is required for installation and optimal functioning.

4.3.2. SOFTWARE REQUIREMENTS

- **Android:** An android version of at least 4.4
- **Internet access**

4.4. SYSTEM IMPLEMENTATION

The Speech activated control system comprises two modules – An Android application and A physical control unit.

The Android application is run on and Android OS supported device and requires just a standard installation of its .apk file.

All components other than the microcontroller require 5V electricity to function. The power supply unit provides 9V to power the microcontroller which in turn provides 5V to power all other components.

The system functionality is described below:

A user configures the devices to be controlled by choosing the device type and manufacturer and then assigning ON and OFF commands to activate those functionalities. After configuration, the infrared code corresponding to the device type and manufacturer chosen is retrieved from the Tiny WebDB and stored locally on the applications TinyDB. The infrared code is stored as a string but has to be transmitted as an integer array. When power is given to the physical control unit, a Bluetooth connection is established between the Android device and the control unit by choosing the device from the list of available Bluetooth devices on the Android device.

The user presses the button on the Android application to receive the speech command. When the speech command is received, it is converted to text and compared against tags in the TinyDB. If the speech command is found in the TinyDB, its corresponding infrared code is converted into a list and each element is sent over Bluetooth as a 2-Byte number to the control unit terminating in 1000. At every instance, the control unit checks if data is available in the Serial buffer and temporarily store all data received. As the infrared code is sent as a 2-Byte number, every second data received is multiplied by 256 and added to the preceeding data before being stored in a temporary array. If the data recieved is 1000, the control unit blasts the accumulated infrared code through the infrared emitter at a frequency of 35 Khz. All temporary data structures are then unset.

The complete circuit diagram of the control unit is shown below:

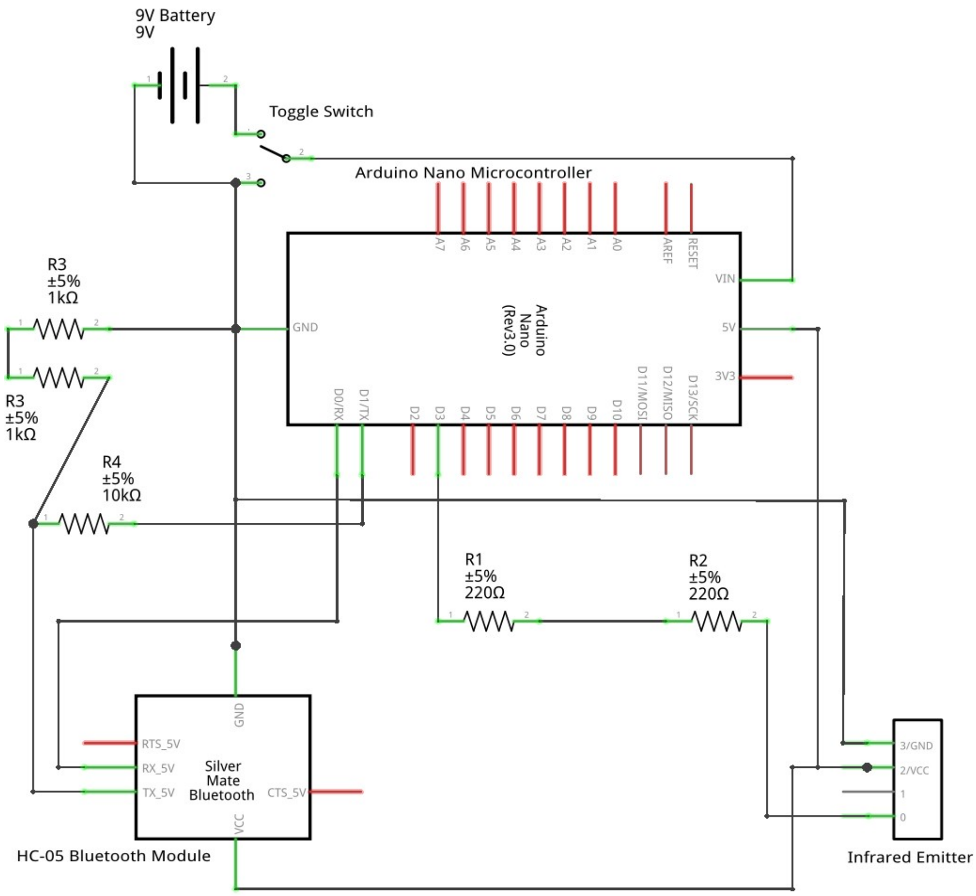


Figure 4.2: Proposed Circuit Diagram Of Control Unit

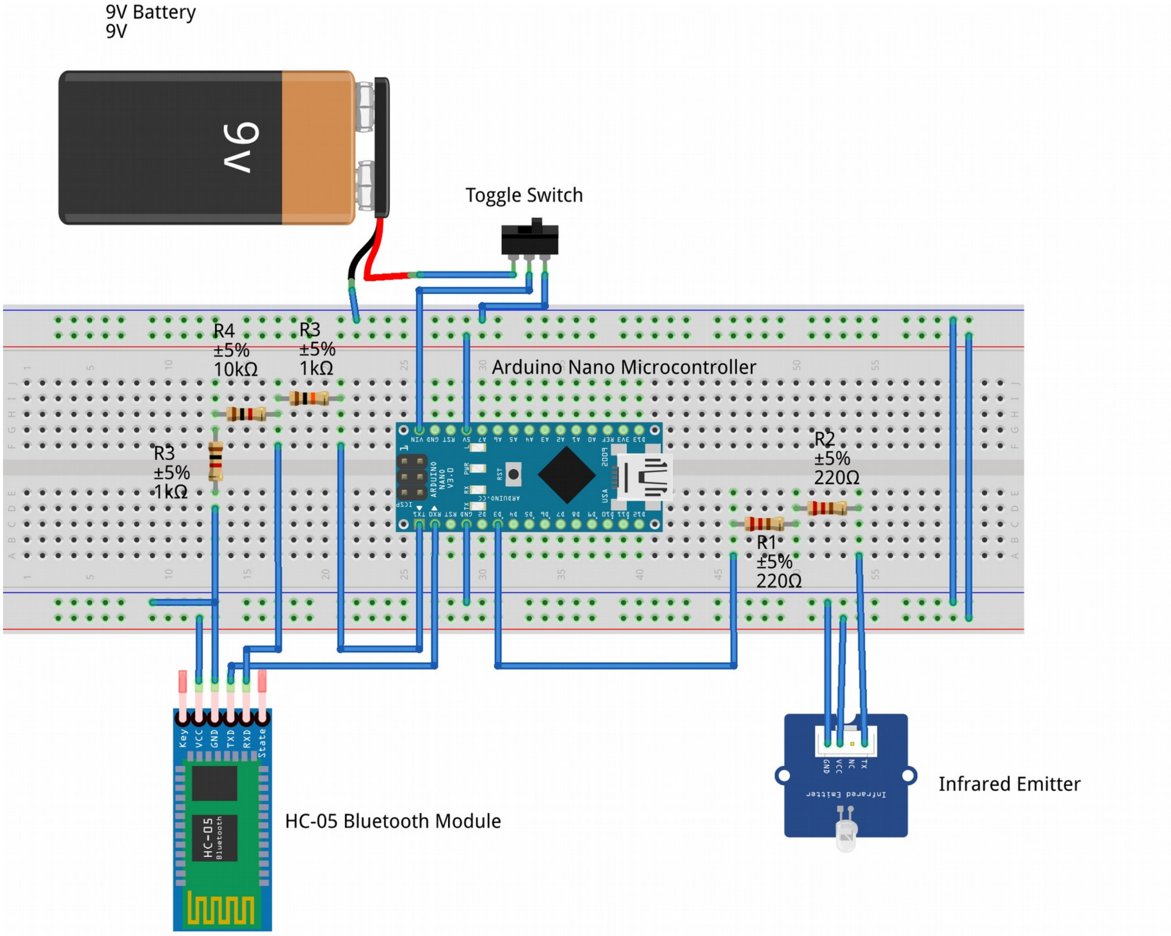


Figure 4.3: Proposed Breadboard Connection Of Control Unit

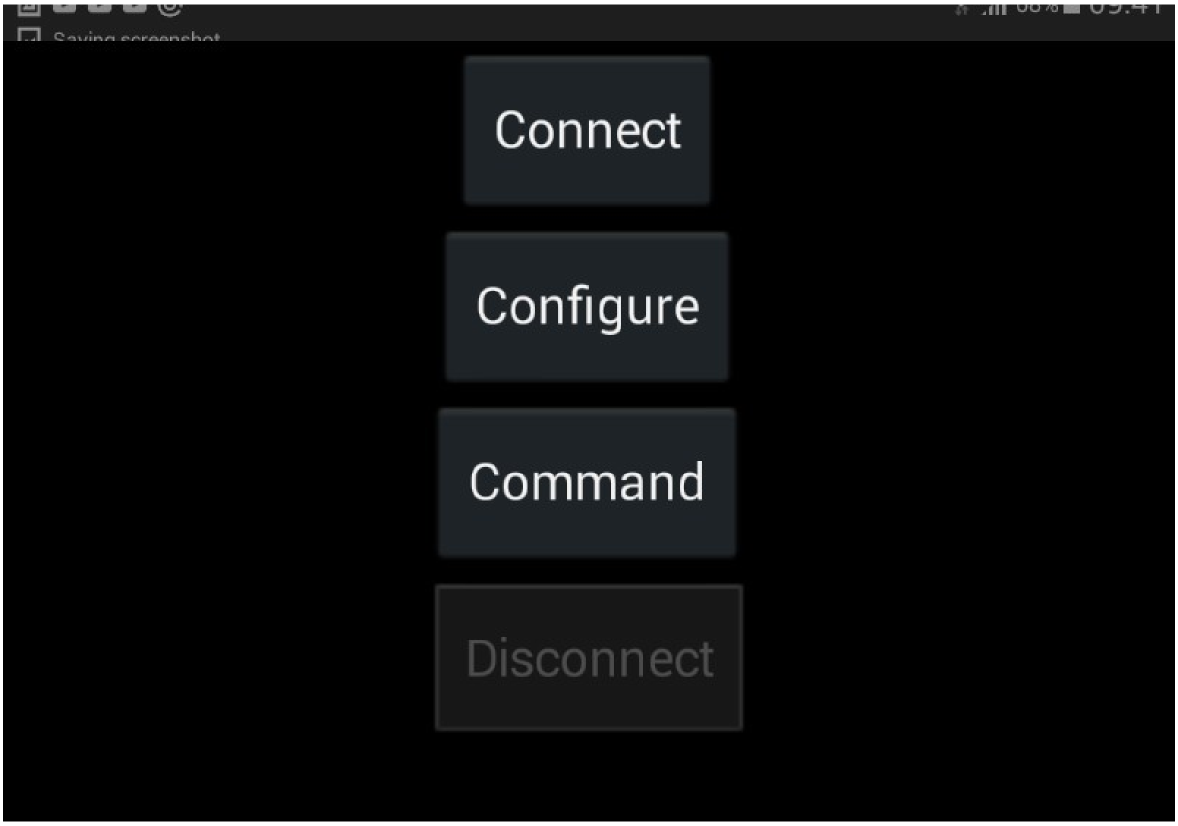


Figure 4.4: Proposed Layout Of Main Page Of Android Application

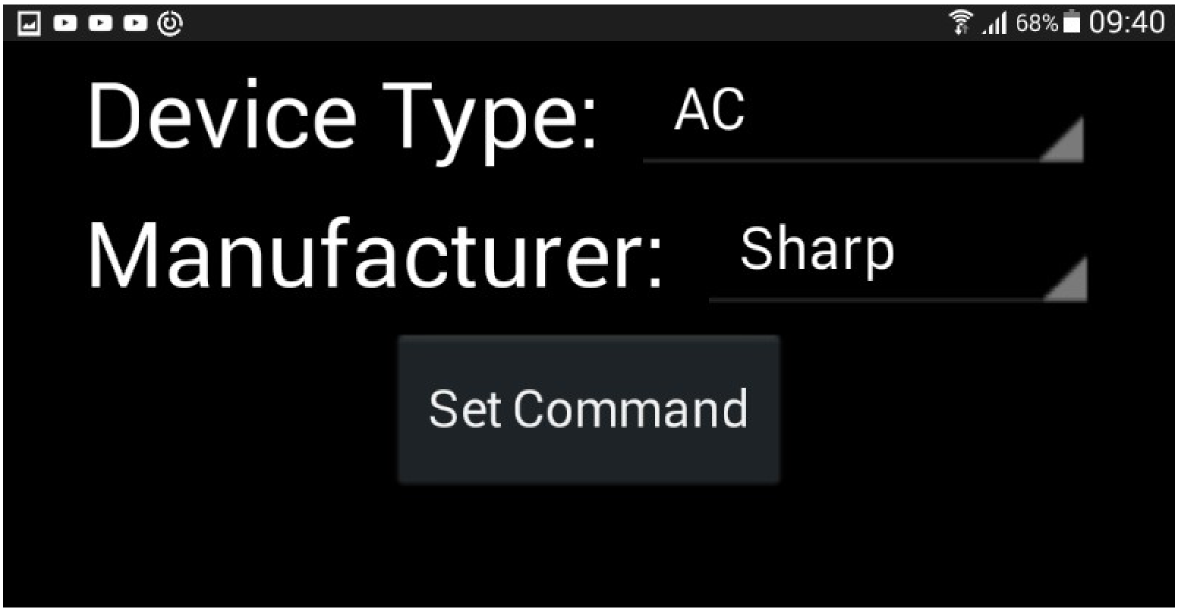


Figure 4.5: Proposed Layout Of Device Configuration Page Of Android Application

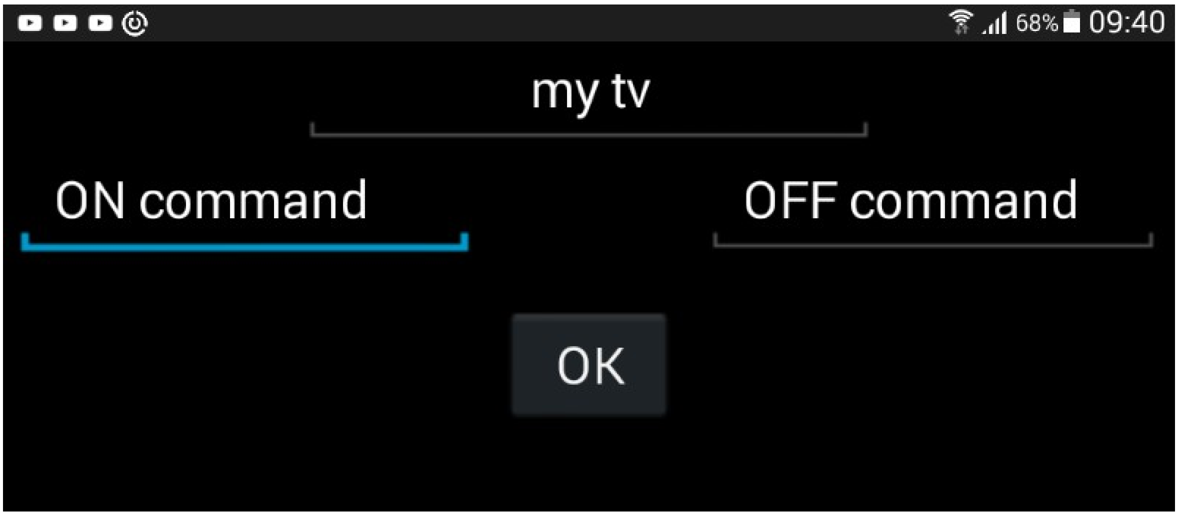


Figure 4.6: Proposed Layout Of Command Configuration Page Of Android Application

4.5. TESTING AND ANALYSIS

Both modules were tested during development. During construction of the system, the following faults in each module were encountered and corrective measures taken shall be explained further below:

Android Applications

- Bluetooth disconnection on screen transition
- Format to transfer stored infrared code

Control Unit

- Malfunctioning of the infrared emitter

4.4.1. BLUETOOTH DISCONNECTION ON SCREEN TRANSITION

The Bluetooth connection between the Android device and the control unit kept breaking when the application transitioned from one screen to another. The problem arose because the connection had been refreshed on every screen. This was solved through the implementation of virtual screens so all the application's activities ocurred on only one screen.

4.4.2. FORMAT TO TRANSFER STORED INFRARED CODE

The infrared code is stored as a string in the TinyDB of the application while the format in which it is to be emitted is an integer array, a problem arose on how to transmit the code. The initial method was to send the entire code as text and format at the control unit but the data proved to be too large for the microcontroller flash memory and the Bluetooth serial buffer. The problem was solved by transferring each element in the array sequentially at 10µs intervals as 2-Byte numbers.

4.4.3. MALFUNCTIONING OF INFRARED EMITTER

The infrared emitter was not emitting the infrared code in the first implementation. After analyzing the flow of data at each point in the hardware code, the problem identified was the method of encoding the raw signal the infrared remote Arduino library that was being used. The problem was solved by finding and using a simpler and better infrared remote Arduino library.

CHAPTER FIVE

RECOMMENDATIONS AND CONCLUSION

5.1. INTRODUCTION

This chapter gives a comprehensive summary of the entire project and suggests adequate recommendations towards how to improve certain features present. It also gives a conclusion on the design and methodology employed as well as what the project stands for.

5.2. RECOMMENDATIONS

Although the projects provides support for a plethora of devices, there is still a large set of devices and brands it does not which could not be implemented due to time constraints, technical know-how and complexity of their infrared encoding algorithm. The system could be expanded to provide support for more devices. A variety of more control features could be implemented to make the system more dynamic.

5.3. CONCLUSION

Having developed a control system to provide support for older model infrared appliances, it is important to know that for the system to work effectively, a good environment where it can be centrally installed and adequate maintenance practices will need to be provided. The system is user-friendly and requires little to no orientation to understand its workings as long as the intended appliance is supported.

REFERENCES

B.V.S Sai Chaitanya, T.Chandra Sekhar and N.V.K Ramseh. (2016). IoT Based Smart Device Using CC32000. *Indian Journal of Science and Technology, 9(16).* DOI:10.17485/ijst/2016/v9i16/87057

S.Sen, S.Chakrabarty, R,Toshniwal, A.Bhaumik. (2015). Design of an Intelligent Voice Controlled Home Automation System. *International Journal of Computer Applications, 121(15),39-42*. DOI:10.5120/21619-4904

A.Saiskumar(2016).Voice Controlled Smart Home. *International Journal of Emerging Technology and Advanced Engineering. 5(1).*

K.Pandaya (2017, March 10). Voice Controlled Home Automation System. Retrieved from *http://electronicsforu.com/electronics-project/voice-controlled-home-automation-system*

J.Porter. (2018, November 12). Histoy of IoT(Internet of Things) and How its Changed Today. Retrieved from *http://www.google.com/amp/s/www.techprevue.com/history-iot-changed-today*

Microcontrollers lab (2017, February 11). Voice controlled Home Automation using adruino. Retrieved from *http://microcontrollerslab.com/voice-controlled-home-automation*

History of Infrared and Infrared Technology. Retrieved from *http://biosmartsolutions.com/shop/buyers-guide/technology/history-of-infrared-technology*

Short History of the Remote Control (2015, July 4). Retrieved from *http://www.sureuniversal.com/short-history-remote-control*

Evolution Of The Remote Control (2016 February) Retrieved from *http://www.ave4u.net/the-evolution-of-the-remote-control-a-along-strange-trip*

Parameshachari, B. D., Gopy, S. K., Hurry, G., & Gopaul, T. T. (2013). A Study on Smart *Home Control System through Speech. International Journal of Computer Applications, 69(19).*

Ali, A. T., Eltayeb, E. B., & Abusail, E. A. A. *Voice Recognition Based Smart Home Control System.*

Mahmud, K., Joarder, M. M. A., & Roy, A. K. *Research Article Voice Activated Electronic Devices Control System For Home Appliances.*

Chien, J. R. C., & Tai, C. C. (2004, March). The information home appliance control system-a Bluetooth universal type remote controller. In *IEEE International Conference on Networking, Sensing and Control, 2004 (Vol. 1, pp. 399-400).* IEEE.

The Engineering Projects. (2019). Introduction to Arduino Nano - The Engineering Projects. [online] Available at: *https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html* [Accessed 20 Feb. 2019].

Designer and Blocks Editor. (n.d.). Retrieved from *http://appinventor.mit.edu/explore/designer-blocks.html*

OPEN-SMART Infrared Emitter Module IR Transmitter Module for Arduino. *DX.com*, *www.dx.com/p/produino-ir-transmitter-module-for-arduino-blue-works-with-official-arduino-boards-2038634#.XG0yg1VKjqQ.*

Brun, C. L. (n.d.). 8 benefits of using automated systems. (2017, January 24). Retrieved from *https://blog.cimpl.com/8-benefits-of-using-automated-systems*

Lueth, K. L. (n.d.). Why the Internet of Things is called Internet of Things: Definition, history, disambiguation. (2014, December 19). Retrieved from *https://iot-analytics.com/internet-of-things-definition/*

Rowan-Robinson, M. (2013). *Night vision: Exploring the infrared universe*. Cambridge University Press.

Farhi, P. (2007). *The Inventor Who Deserves a Sitting Ovation*. Washington Post. February, 17.

Gertner, J. (2007, December 30). The Lives They Lived - Robert Adler - Remote Control - Television. Retrieved from *https://www.nytimes.com/2007/12/30/magazine/30Adler-t.html*

Pinola, M. (2011). *Speech recognition through the decades: How we ended up with siri*. Web log post. TechHive. IDGTechNetwork, 2.

Haynes, W. M. (2014). *CRC handbook of chemistry and physics*. CRC press.

APPENDIX I

SOURCE CODE

control_unit.ino

```
#include <StandardCplusplus.h>     //Library for all c++ data
                                   structures not included in
                                   Arduino
#include <vector>                  //Library for vector data
                                   structure
#include <IRLibSendBase.h>         //We need the base code
#include <IRLib_HashRaw.h>         //Only use raw sender

using namespace std;               //Standard namespace for
                                   c++ methods

IRsendRaw mySender;                //Initialize IR sender
                                   object at digital pin 3

int tx=1;                          //transmitter terminal
int rx=0;                          //receiver terminal
char buf;                          //buffer to hold received
                                   bytes from Bluetooth
vector<uint16_t> sample;           //Global Data structure to
                                   temporarily store IR code
```

```
                                    received through Bluetooth
                                    Module
int i=0;
int state = 0;

void setup() {
  Serial.begin(9600);               //Set BAUD rate for serial
                                    port
  delay(2000); while (!Serial);

//Set transmitter and receiver terminal to INPUT and OUTPUT
respectively
  pinMode(tx,OUTPUT);
  pinMode(rx,INPUT);

}

void loop() {
    if(Serial.available() > 0){     // Checks whether data is
                                    comming from the serial
                                    port
 /*
     ** The Android application sends elements in IR code as
     2-Byte numbers with the last element being 1000.
     ** Bluetooth module receives data 1 Byte at a time.
```

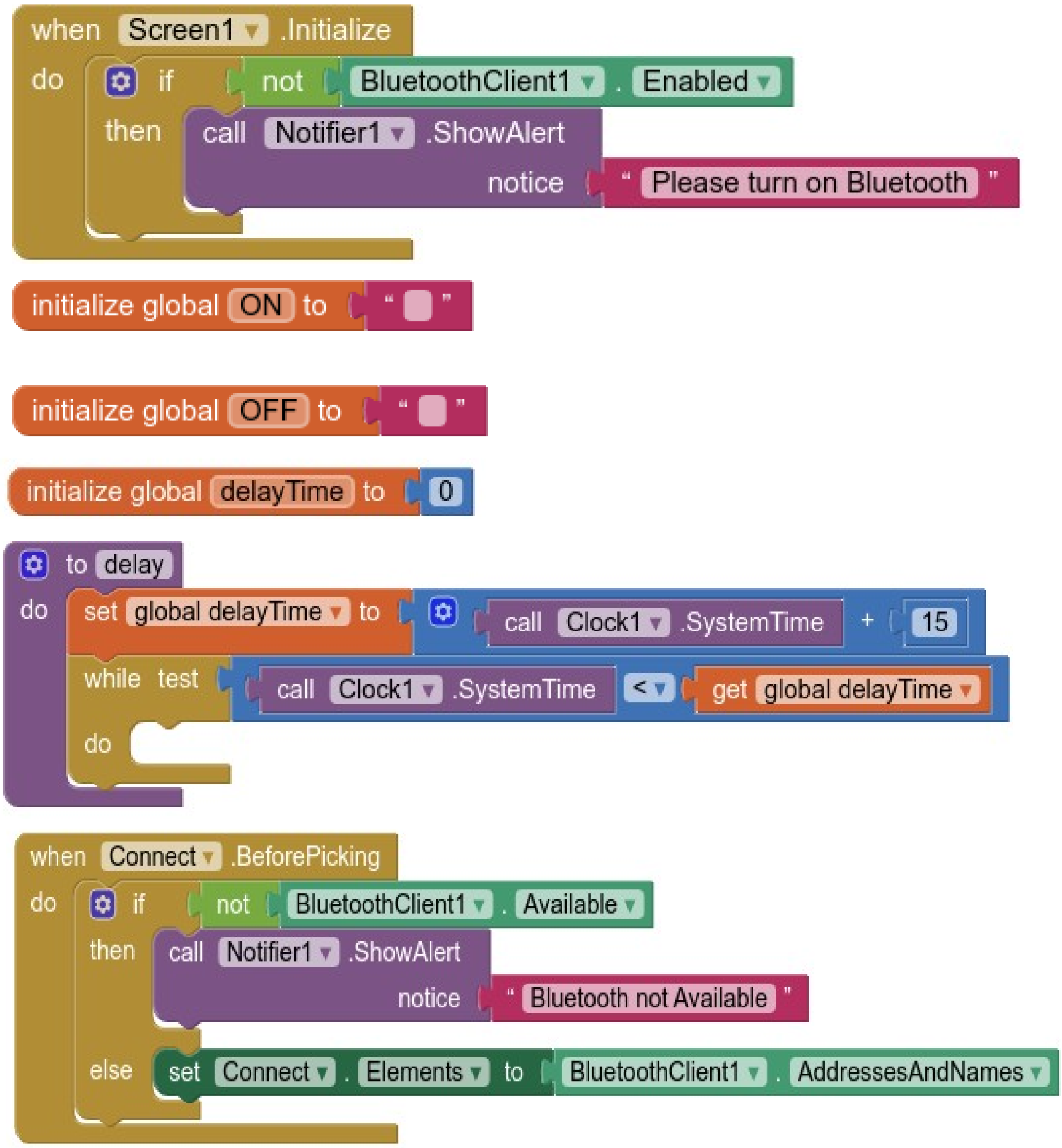
```
     ** Reconstruct data by multiplying every second data
     received  by 256 and summing it with preceeding data.
     ** Store reconstructed code in sample vector.
     ** If reconstructed data is 1000, call send_code()
     function.
     ** After send_code() is completed, clear sample vector
     memory.
*/
    while(Serial.available()>0){

      if(i%2==0){
        state=(int) Serial.read();
      }
      else{
        state+=(int) Serial.read() * 256;
        sample.push_back(state);
        state=0;
      }
      i++;
    Serial.println(i);
    Serial.println();
    delay(10);
      }


    }
```

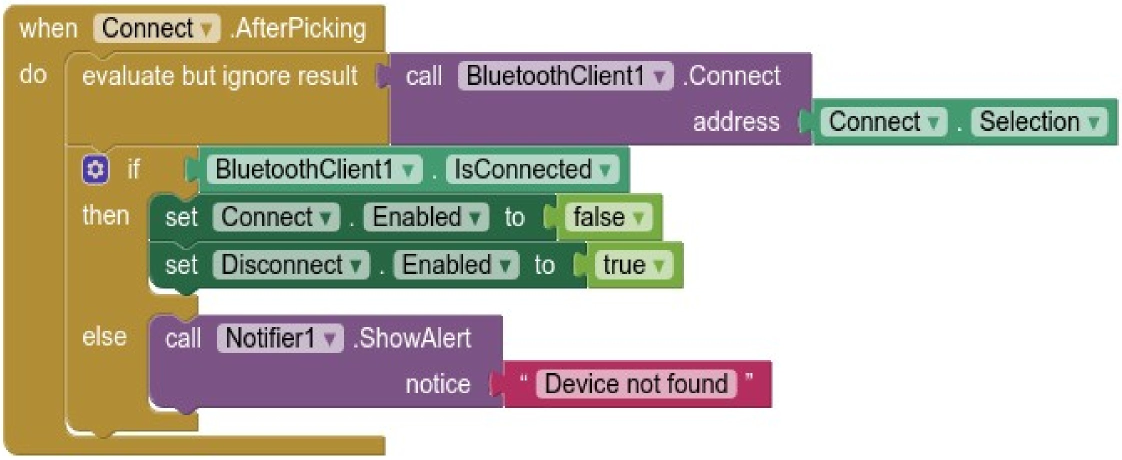
```
    if(sample[(i/2)-1]==1000){
        send_code();
        sample.clear();
        i=0;
      }
    }

/*
Purpose:  This function formats the received infrared code
          into a format supported by the emitter library and
          then blasts the IR code at a rate of 36Hz.

args: None

Pre-conditions:     * sample vector is non-empty
                    * IR emitter is up and connected to
                    digital pin 3

Post-condition:     * IR signal is blasted through the IR
                    emitter.
*/
void send_code(){
  uint16_t code[sample.size()];
```

```
  copy(sample.begin(),sample.end(),code);    //copy contents
                                             of sample vector
                                             into code array
  mySender.send(code,sample.size(),36);      //Pass the
                                             buffer,length,
                                             optional
                                             frequency
  delay(40);
}
```

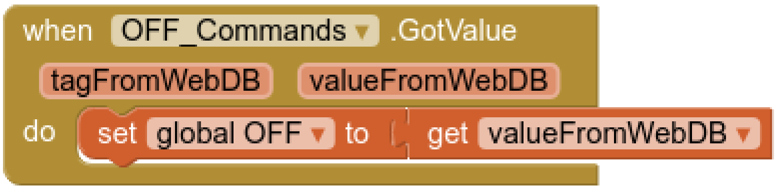
APPENDIX II

ANDROID APPLICATION BLOCK VIEW

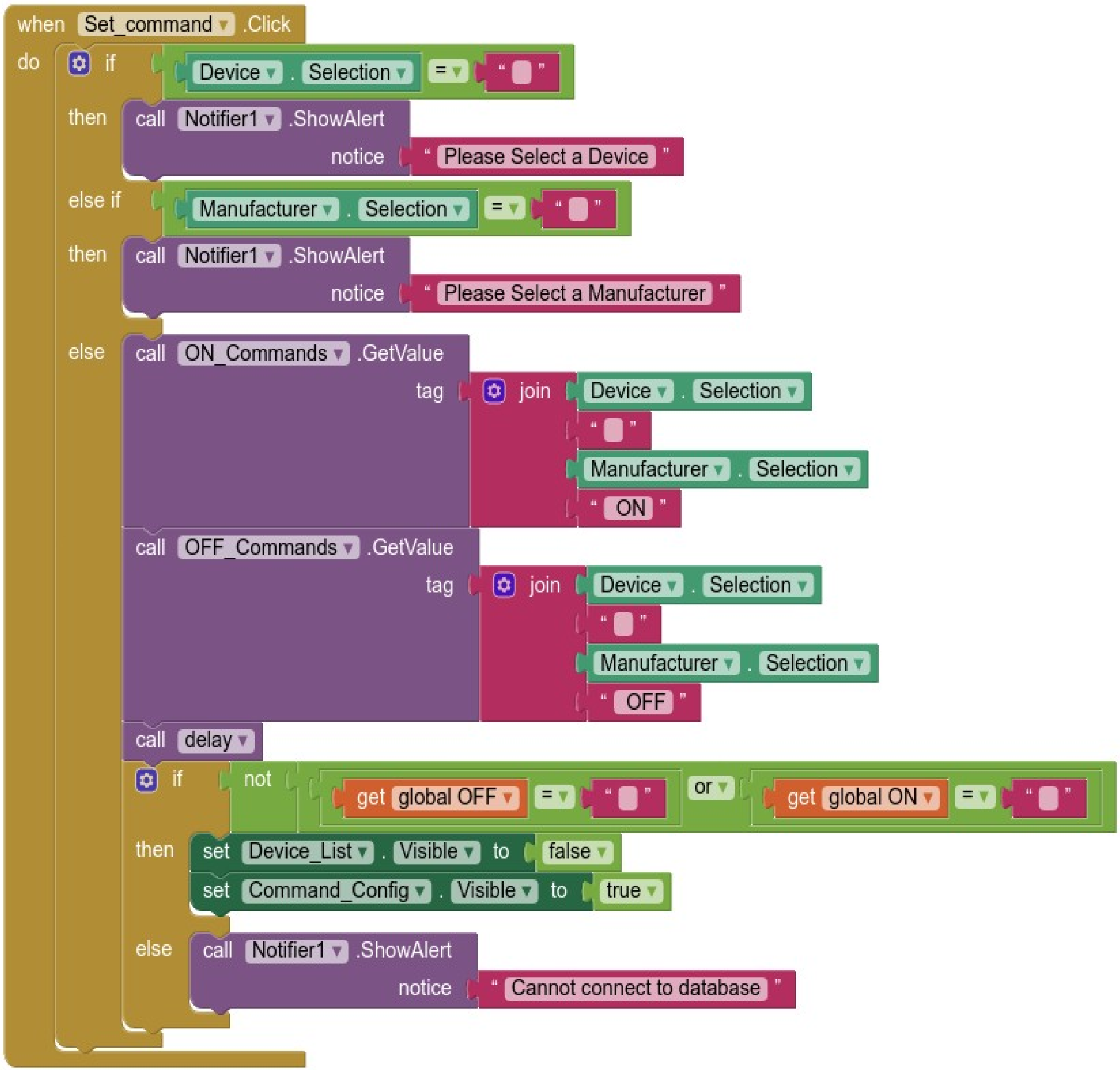


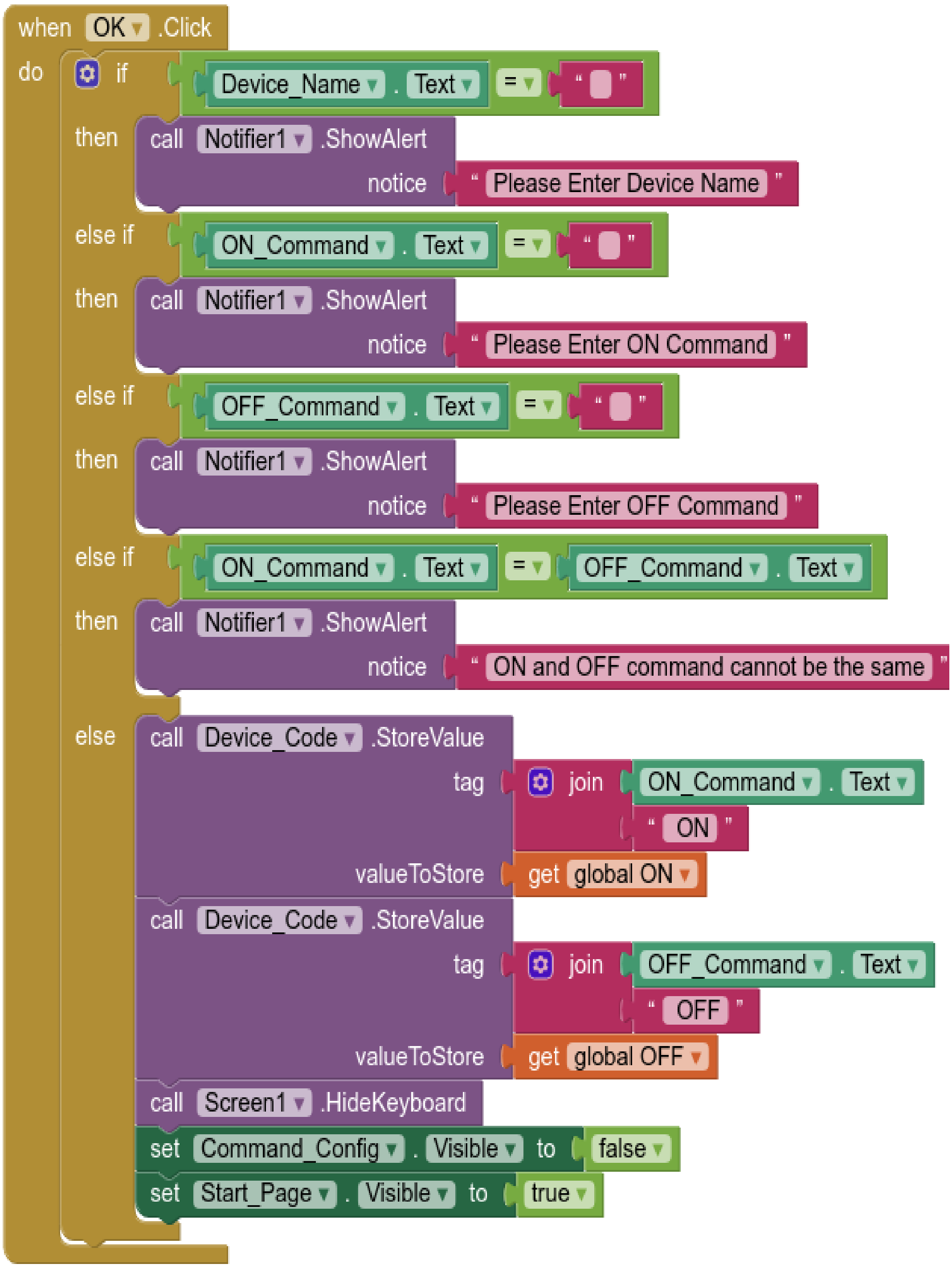

when Connect .AfterPicking
do evaluate but ignore result call BluetoothClient1 .Connect
address Connect . Selection
if BluetoothClient1 . IsConnected
then set Connect . Enabled to false
set Disconnect . Enabled to true
else call Notifier1 .ShowAlert
notice " Device not found "


when Configure .Click
do set Start_Page . Visible to false
set Device_List . Visible to true

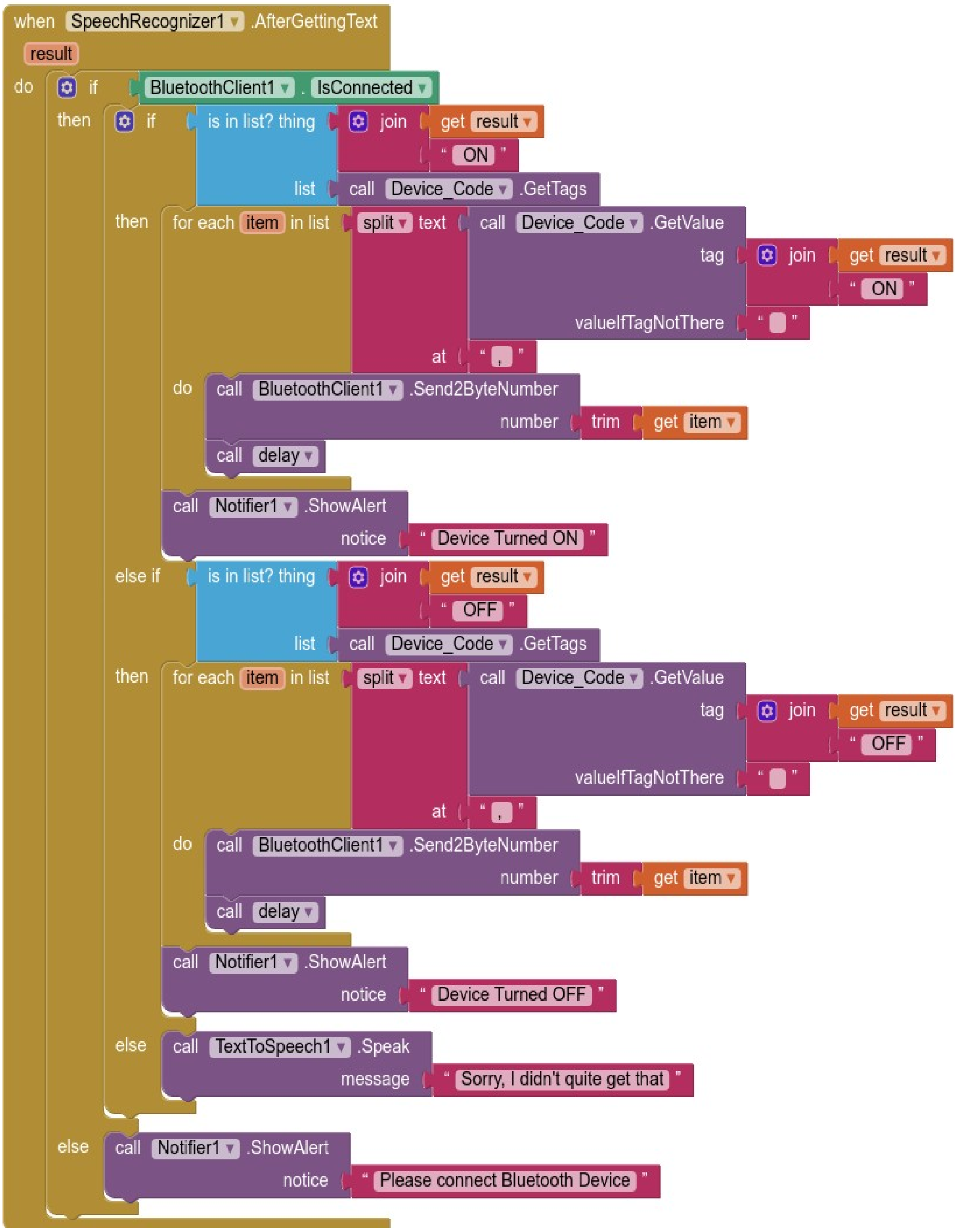

when OFF_Commands .GotValue
tagFromWebDB valueFromWebDB
do set global OFF to get valueFromWebDB

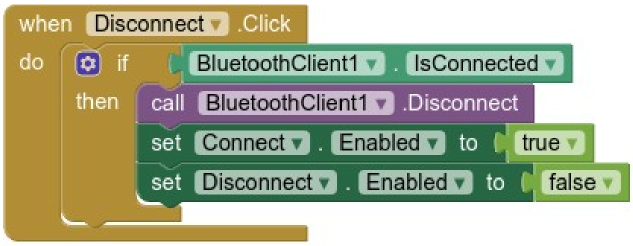

when ON_Commands .GotValue
tagFromWebDB valueFromWebDB
do set global ON to get valueFromWebDB

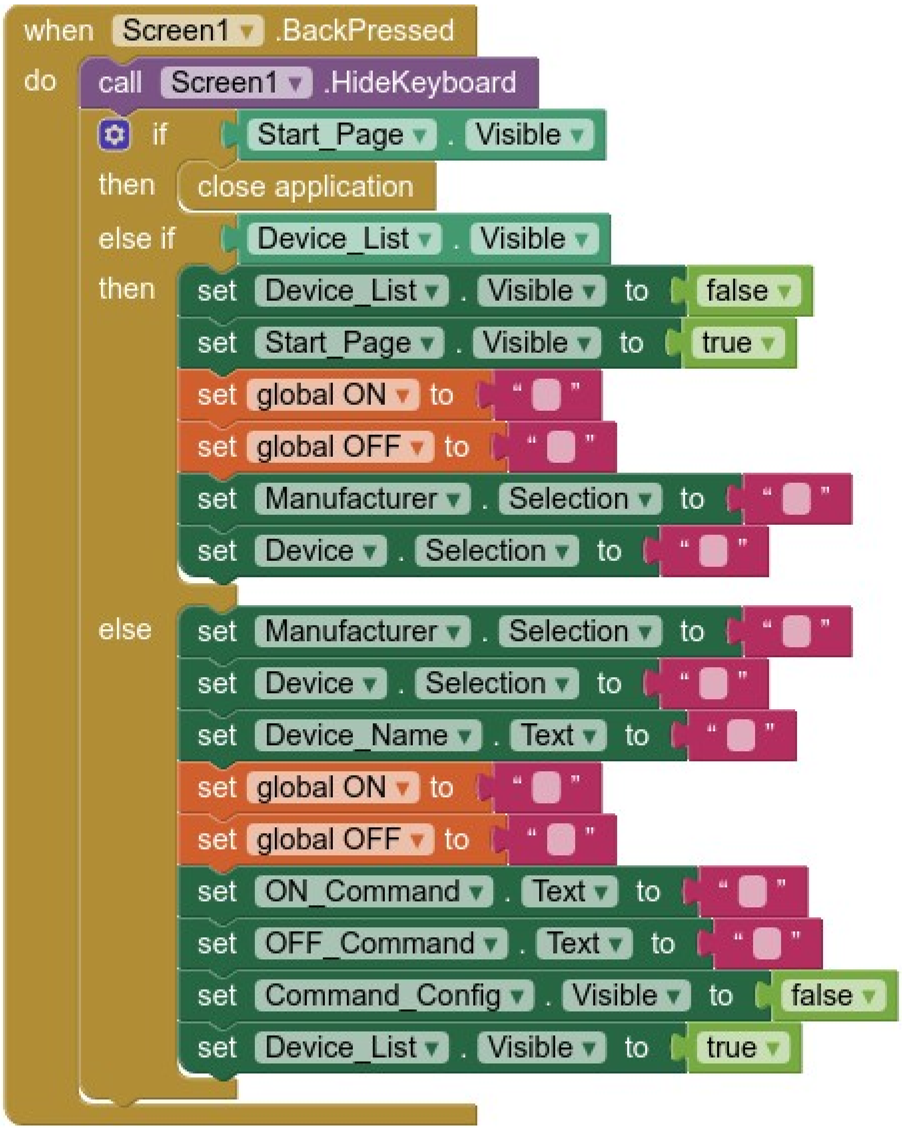

when Set_command .Click
do if Device . Selection = " "
then call Notifier1 .ShowAlert
notice " Please Select a Device "
else if Manufacturer . Selection = " "
then call Notifier1 .ShowAlert
notice " Please Select a Manufacturer "
else call ON_Commands .GetValue
tag join Device . Selection
" "
Manufacturer . Selection
" ON "
call OFF_Commands .GetValue
tag join Device . Selection
" "
Manufacturer . Selection
" OFF "
call delay
if not get global OFF = " " or get global ON = " "
then set Device_List . Visible to false
set Command_Config . Visible to true
else call Notifier1 .ShowAlert
notice " Cannot connect to database "


when OK .Click
do if Device_Name . Text = " "
then call Notifier1 .ShowAlert
notice " Please Enter Device Name "
else if ON_Command . Text = " "
then call Notifier1 .ShowAlert
notice " Please Enter ON Command "
else if OFF_Command . Text = " "
then call Notifier1 .ShowAlert
notice " Please Enter OFF Command "
else if ON_Command . Text = OFF_Command . Text
then call Notifier1 .ShowAlert
notice " ON and OFF command cannot be the same "
else call Device_Code .StoreValue
tag join ON_Command . Text
" ON "
valueToStore get global ON
call Device_Code .StoreValue
tag join OFF_Command . Text
" OFF "
valueToStore get global OFF
call Screen1 .HideKeyboard
set Command_Config . Visible to false
set Start_Page . Visible to true


when Command .Click
do call SpeechRecognizer1 .GetText


when SpeechRecognizer1 .AfterGettingText
result
do if BluetoothClient1 . IsConnected
then if is in list? thing join get result
" ON "
list call Device_Code .GetTags
then for each item in list split text call Device_Code .GetValue
tag join get result
" ON "
valueIfTagNotThere " "
at " , "
do call BluetoothClient1 .Send2ByteNumber
number trim get item
call delay
call Notifier1 .ShowAlert
notice " Device Turned ON "
else if is in list? thing join get result
" OFF "
list call Device_Code .GetTags
then for each item in list split text call Device_Code .GetValue
tag join get result
" OFF "
valueIfTagNotThere " "
at " , "
do call BluetoothClient1 .Send2ByteNumber
number trim get item
call delay
call Notifier1 .ShowAlert
notice " Device Turned OFF "
else call TextToSpeech1 .Speak
message " Sorry, I didn't quite get that "
else call Notifier1 .ShowAlert
notice " Please connect Bluetooth Device "


when Disconnect .Click
do if BluetoothClient1 . IsConnected
then call BluetoothClient1 .Disconnect
set Connect . Enabled to true
set Disconnect . Enabled to false


when Screen1 .BackPressed
do call Screen1 .HideKeyboard
if Start_Page . Visible
then close application
else if Device_List . Visible
then set Device_List . Visible to false
set Start_Page . Visible to true
set global ON to
set global OFF to
set Manufacturer . Selection to
set Device . Selection to
else set Manufacturer . Selection to
set Device . Selection to
set Device_Name . Text to
set global ON to
set global OFF to
set ON_Command . Text to
set OFF_Command . Text to
set Command_Config . Visible to false
set Device_List . Visible to true

APPENDIX III

LIST OF COMPONENTS USED

- Resistors
- Arduino Nano Microcontroller
- A 9V Battery and snap connector
- Jumper Cables
- HC-05 Bluetooth Module
- Infrared Emitter
- Infrared Receiver
- Breadboard