SMART AUTOMATION USING MACHINE LEARNING ALGORITHMS AND NEURAL INTERFACE

A PROJECT REPORT

Submitted by

AKSHAY SANKAR J (CCV16CS005)

HAFEED K (CCV16CS020)

MINHAJ FASALU RAHMAN (CCV16CS023)

to

the A P J Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

In

Computer Science Engineering

DEPT. OF COMPUTER SCIENCE ENGINEERING

MGM COLLEGE OF ENGINEERING AND PHARMACEUTICAL SCIENCES

VALANCHERY

JUNE 2021

DECLARATION

We undersigned hereby declare that the project report " SMART AUTOMATION USING MACHINE LEARNING ALGORITHMS AND NEURAL INTERFACE", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of Ms.SHABNA M, Asst professor, Department of Computer Science and Engineering. This submission represents our ideas in our own words and where ideas or words of others have been included. We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission.We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Akshay Sankar J

Hafeed k

Minhaj Faslu Rahman

Place: Valanchery

Date: 11-07-2021

DEPT. OF COMPUTER SCIENCE & ENGINEERING

MGM COLLEGE OF ENGINEERING AND PHARMACEUTICAL SCIENCES

VALANCHERY

2020 - 2021

CERTIFICATE

This is to certify that the report entitled **SMART AUTOMATION USING MACHINE LEARNING ALGORITHMS AND NEURAL INTERFACE** , submitted by **AKSHAY SANKAR J (CCV16CS005), HAFEED K (CCV16CS020), MINHAJ FASALU RAHMAN (CCV16CS023)**, to the APJ Abdul Kalam Technological University in partial fulfillment of the B.tech. degree in Computer Science Engineering is a bonafide record of the project work carried out by them under my/our guidance and supervision. *This report in any form has not been submitted to any other University or Institute for any purpose.*

Ms.Shabna M
Internal Supervisor

External Supervisor

Ms.Nahan Rahman M.k
Project Coordinator

Mr.Vipin Krishnan C.V
Head of Department

ACKNOWLEDGEMENT

We take this opportunity to express our deep sense of gratitude and sincere thanks to all who helped us to complete the work successfully. Our first and foremost thanks goes to God Almighty who showered in immense blessings on our effort.

We wish to express our sincere thanks to our principal **Dr.KP Indira Devi** for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to **Mr. Vipin Krishnan C V** (Head of the department), for his support and co-operation. We wish to express our sincere gratitude towards all the teaching and non teaching staff members of our Department.

Finally we thank our parents, all our friends, near and dear ones who directly and indirectly contribute to the successful of this work.

AKSHAY SANKAR J (CCV16CS005)

HAFEED K (CCV16CS020)

MINHAJ FASALU RAHMAN (CCV16CS023)

ABSTRACT

A home automation system controls lighting, temperature, multimedia systems, and appliances. Since these devices and sensors are connected to common infrastructure, they form the Internet of Things. A home automation system links multiple controllable devices to a centralized server. These devices have a user interface for controlling and monitoring, which can be accessed by using a tablet or a mobile application, which can be accessed remotely as well. Ideally, anything that can be connected to a network can be automated and controlled remotely. Smart homes must be artificially intelligent systems that need to adapt themselves based on user actions and surroundings. These systems need to carefully analyze the user needs and the conditions of the surroundings in order to predict future actions and also minimizes user interaction. Traditional home automation systems that provide only remote access and control are not that effective in terms of being 'smart', so in this project we put forward the use of concepts of different machine learning algorithms along with computer vision to shape together a smart learning automated system that controls lighting, sound and other devices based on the user's emotion.

Index Terms:- Home automation, Emotion Recognition, Neural wave Sensing, Supervised learning, Smart Home System.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AMP	Amplifier
Ch	Channel
ECG	Electro Cardio Graph
EEG	Electro Encephalograph
L1	Light1
L2	Light2
L3	Light3
ML	Machine Learning
NI	Neural Interface
R1	Relay1
R2	Relay2
R3	Relay3
R4	Relay4
RPI	Raspberry pi
SVM	Support Vector Machine
VRAM	Virtual Random Access Memory

Chapter 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Smart Home Systems are the subset of everyday computing which includes smart technology for providing comfort, health, safety, security and energy reduction. When this application is controlled by machine intelligence to provide circumstance-aware settings, services and facilitate remote control it significantly improves user comfort. Further addition of automated appliance control and accessibility services can improve the quality of life as well. Internet of things that contain multiple sensors can detect temperature, light, sound, distance, air pressure, motion which act as different points of data sources. Since there is a huge amount of data involved Machine learning can be applied to the existing Home automation systems to make it perform exceptionally well based off the users emotions. In this paper we define an improved Home automation system using multiple machine learning algorithms that can detect human facial expressions and adjust the environment conditions accordingly.

1.2 OBJECTIVE

Our main goal is to bring down manual control to zero and emphasize on strict power control. Also adding circumstance awareness to the system will give us a intelligent control system which saves energy and user comfort.

1.3 SCOPE

Scope of the project result is limited due to limitations in available sensor and lack of availability of desired sensor. full range of emotion recognition may not be detected,neural interface can only function in a limited fashion

1.4 SCHEME OF PROJECT WORK

Project is divided into 3 stages

Stage 1

- Build Manual Control and Existing System.
- Connect ESP8266 to Network Via WIFI Authentication.
- Control relay via Blynk App through Blynk App.
- Relay Controls Connected Appliances.

Stage 2

- Build Emotion Recognition to the Existing System.
- Train Face Data to recognize emotions.
- Link Program To ESP8266 And Raspberry Pi 3.

Stage 3

- Build Neural Interface and connect to Existing System along Emotion recognition.
- Link Neural Interface To ESP8266 And Raspberry Pi 3.
- Link Program To ESP8266 And Raspberry Pi 3.

Chapter 2

LITERATURE SURVEY

2.1 Smart Energy Efficient Home Automation System Using IoT

Advancement in IoT based application has become the state-of-the art technology among the researcher due to the availability of Internet everywhere. To make the application more user friendly, web based and android based technologies have gained their importance in this cutting edge technology. In this paper, smart energy efficient home automation system is proposed that can access and control the home equipments from every corner of the world. For this system, Internet connectivity module is attached to the main supply unit of the home system which can be accessed through the Internet. For wireless connectivity, the static IP address is used. Home automation is based on multi modal application that can be operated using voice recognition command of the user using the Google Assistant or through a web based application. Thus, main objective of this work is to make our home automation system more secure and intelligent.

2.2 i-learning IoT: An intelligent self learning system for home automation using IoT

Internet of Things (IoT) is extension of current internet to provide communication, connection, and inter-networking between various devices or physical objects also known as "Things." In this paper we have reported an effective use of IoT for Environmental Condition Monitoring and Controlling in Homes. We also provide fault detection and correction in any devices connected to this system automatically. Home Automation is nothing but automation of regular activities inside the home. Now a day's due to huge advancement in wireless sensor network and other computation technologies, it is possible to provide flexible and low cost home automation system. However there is no any system available in market which provide home automation as well as error detection in the devices efficiently. In this system we use prediction to find out the required solution if any problem occurs in any device connected to the system. To achieve that we are applying Data Mining concept. For efficient data mining we use Naive Bayes Classifier algorithm to find out the best possible solution. This gives a huge upper hand on other available home automation system, and we actually manage to provide a real intelligent system.

2.3 Enhanced Smart Home Automation System based on Internet of Things

The Internet of Things (IoT) connects users with interconnection of things to facilitate the life. IoT is now shifted towards 'Thing to Thing'. Smart home concept brings comfort and convenience in our lives with the aid of IoT. Major issues in current smart home scenario are automation and security. Problem in security arises due to network of devices in the home with internet. Focus is sifted towards providing confidentiality, authenticity, and integrity of data sensed and exchanged by smart home objects. Computation overhead is also a concern for smart home solutions. Comfort and user requirements as per scenario or situation are basic need for automation. Automation with learning human behavior is

also a major concern with smart home concept. Paper represents IoT based smart home automation approach which is secure and also reduces computation overhead.

2.4 Enabling IoT automation using local clouds

Various forms of cloud computing principles and technologies are becoming important recently. This paper addresses cloud computing for automation and control applications. It's argued that the open Internet cloud idea has such limitations that its not appropriate for automation. Since automation is physically and geographically local, it is inevitable to introduce the concept of local automation clouds. It's here proposed that local automation clouds should be self contained an be able to execute the intended automation functionalities without any external resources. Thus providing a fence at the rim of the local cloud preventing any inbound or outbound communication. Such a local cloud provides possibilities to address key requirements of both todays and future automation solutions. Adding mechanisms for secure inter-cloud administration and data tranfere enables local automation cloud to meet IoT automation system requirements as: 1) Interoperability of a wide range of IoT and legacy devices 2) Automation requirement on latency guarantee/prediction for communication and control computations. 3) Scalability of automation systems enabling very large integrated automation systems 4) Security and related safety of automation systems 5) Ease of application engineering 6) Multi stakeholder integration and operations agility. How these requirements can be met in such a local automation cloud is discussed with references to proposed solutions. The local automation cloud concept is further verified for a compartment climate control application. The control application included an IoT controller, four IoT sensors and actuators, and a physical layer communication gateway. The gateway acted as host for local cloud core functionalities. The climate control application has successfully been implemented using the open source Arrowhead Framework and its supports for design and implementation of self contained local automation clouds.

2.5 Smart Automation using Machine Learning Algorithms

A home automation system controls lighting, temperature, multimedia systems, and appliances. Since these devices and sensors are connected to common infrastructure, they form the Internet of Things. A home automation system links multiple controllable devices to a centralized server. These devices have a user interface for controlling and monitoring, which can be accessed by using a tablet or a mobile application, which can be accessed remotely as well. Ideally, anything that can be connected to a network can be automated and controlled remotely. Smart homes must be artificially intelligent systems that need to adapt themselves based on user actions and surroundings. These systems need to carefully analyze the user needs and the conditions of the surroundings in order to predict future actions and also minimizes user interaction. Traditional home automation systems that provide only remote access and control are not that effective in terms of being 'smart', so in this paper we put forward the use of concepts of different machine learning algorithms along with computer vision to shape together a smart learning automated system that controls lighting, sound and other devices based on the user's emotion.

Chapter 3

MODELLING

In this section, we discuss the principles and methods behind the proposed system.

3.1 Three modes of operations

3.1.1 Manual Mode:

In this mode the User must use the user interface like a remote control for controlling the appliances. The System will respond by Automating only those devices that the User requests.

3.1.2 Emotion Recognition Mode:

The system will use the connected camera to detect the users facial expression and accordingly will automate the lights, fan etc.

3.1.3 Neural Interface Mode:

System Uses EEG(Electroencaphilograph)sensor to detect beta waves from our brain, These waves are transmitted over Bluetooth or Wi-Fi to Our system for processing According to wave pattern respective profiles are switched

3.2 Principle Of Working.

Different Sensors and devices are interfaced with the input/output ports on the Raspberry Pi (Single Board Computer) as displayed in Fig. 3.1.

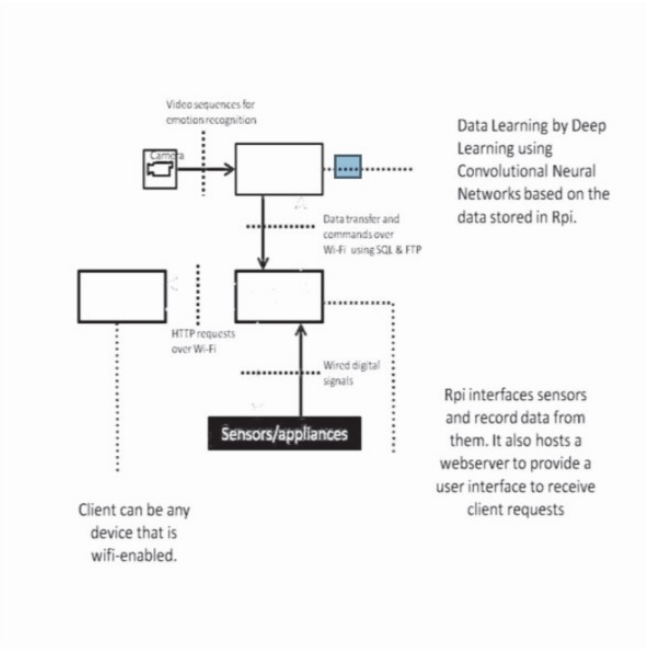

Figure 3.1: Overall setup of the home automation system.

.

It also hosts an Apache powered Web-server that incorporates the User interface for any device on the network to remotely trigger any other appliances in the home. These requests are forwarded by typical Hypertext Transfer Protocol (http) to the Raspberry Pi. Various requests from the User to the sensors/ devices that were used are recorded in a database so

that they serve as the data that can be used by the server running deep learning algorithms. This is done to learn the pattern of how the user uses different sensors at different times of the day. Using this data, while the system is in Automatic mode the Server can predict which sensors need to be activated without the user's input like switching off lights past 11pm, or if the temperature rises, cooling devices kick in.

While using the system in Emotion detection mode, a camera is used to detect the user's face and by the deep learning algorithm implemented using convolutional neural networks it sets the room environment as per the emotion recognized. For this purpose, each emotions have a pre-set configuration of lights, music, temperature level, and specific devices like television can be turned on. User can disable this mode on command. All these Control functions for various devices can be accessed by any device that is Wi-Fi enabled like a smartphone, tablet or a computer. With further modifications (like port forwarding from the internet router), this system can be even connected to the Internet which enables access from outside the home network. The Server shown in Fig. 1. is only used for training and testing purposes. This can be removed since the trained Machine learning model can be directly loaded on to the raspberry pi. During such modification, the camera is to be interfaced directly with the raspberry pi.

3.3 Emotion Recognition.

Here, we use the application of emotion recognition to detect the user's state (emotion) and using this emotion to create a comfortable environment throughout the house by using appropriate lighting, sounds and other parameters. Camera vision and image recognition concepts are important in-order to achieve face detection and emotion recognition. Apart from identifying the user's face, the computer will utilize the composition, order and shape of different faciallandmarks like eyebrows, lips to aid in the detection of facial expression and gives us the detected emotion of the user.

We used following three methods as a major part of our proposed automation system:

- Convolutional Neural Network (CNN) approach.
- Haarscascade Classifier approach.
- Support Vector Machine (SVM) approach.

CNN has an input layer, output layer and other hidden layers. Convolutional layers, pooling layers, fully connected layers and normalization layers comprise the hidden layer of a Convolutional Neural Network. First, Haar Cascade classifier is used to detect faces in each frame of the webcam feed. The region of image containing the face is resized to 48x48 and is passed as input to the ConvNet. The network outputs a list of softmax scores for the seven classes (emotions). The class with the maximum score is displayed. The architecture of neural network is shown in Figure 3.2.

Fisher Face classifier is a face detection algorithm which can be used to classify different emotions based on the input image data. The extracted features can be applied in multiple ways for image pattern identification and facial recognition. The Training data set is labelled images, so a better optimized method can implemented by Dimensionality Reduction.

A Support Vector Machine is the classifier that is defined by a separating hyper plane. By using Supervised Learning, the algorithm will output a hyper plane which categorizes new examples. To use SVM to recognize facial expression, we extract facial landmark key points first and then it is given to SVM classifier to detect user's mood.

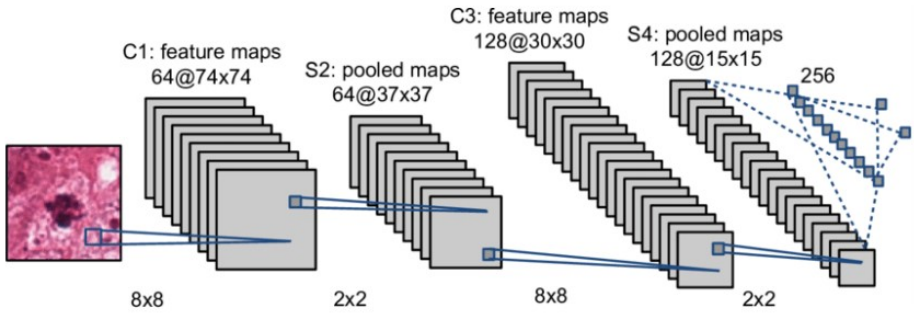

Figure 3.2: Architecture of used CNN.

.

The original network starts with an input layer of 48 by 48, matching the size of the input data. This layer is followed by one convolutional layer, a local contrast normalization layer, The network initially has input layer of 48 x 48, which is matched to the input data size. The next layer is a convolutional layer followed by a contrast normalization layer and max-pooling layer. The neural network is displayed in Figure 3.2. The CNN is finished with convolutional layer followed by a fully connected layer. These are connected to a output(softmax) layer. Dropout given to the fully connected layer and other layers contain ReLu units. Also another max-pooling layer is added to reduce the number of parameters.

To detect the landmarks that are crucial points required for identifying the emotions/-expressions, we use DLib library for python. This will result in some useful key-points like eyebrows, lips, eyes. The landmarks are of course important because it is what makes emotional expressions possible. Normalization of key-points is done by calculating the mean of x and y axes which gives us the coordinates for center of gravity in all facial landmarks. Now calculation of position of points that are relative to the center of gravity is possible. In case of Tilted or disoriented faces, the classifier might make mistakes.
This can be prevented by Rotation of the face which is done by offset angle calculation by assuming the nose bridge angle].We used was the Extended CohnKanade (CK+) dataset with 80-20 training and testing ratio.

3.4 Neural Interface

Neural Interface Uses EEG (Electro Encephalograph)sensor to detect beta waves from the brain These waves are transmitted over Bluetooth or Wi-Fi to Our system for processing According to wave pattern, respective profiles are switched.

Neural Interface Has 2 parts

- Sensor Electrodes.
- Amplifier and signal Filtering.

Sensor Electrodes

We measure the EEG activity as the difference in voltage between two electrodes. As a general rule, one fixed electrode is chosen as the reference for all the other electrodes.

Therefore, an EEG headset contains three types of electrodes:
Recording electrodes: placed over the specific scalp locations that we want to measure.
The reference electrode: one whose signal is subtracted from each of the recording electrodes.
The ground electrode: used to place both the amplifier and the body to the same potential and to reduce common-mode interference.
The number of EEG electrodes determines how much and what type of waves to process.

Amplifier and Signal Filtering

The signal From Electrodes are processed in the amplifier and the required waves are filtered out and transmit the wave pattern to the system to process.

Working

System Acquires the wave pattern and normalizes the required value that is set in the system if the chosen value is reached system chooses the corresponding profile and switch relay accordingly this is the general Working of Neural Interface.

Chapter 4

EXPERIMENTAL ANALYSIS

4.1 Experimental Setup

The Experimental Setup Consist of Aurduino Uno,Ambient Lighting,4CH Relay,AD8232 ECG Sensor,Bluetooth module and Powersupply with Voltage Divider to convert 5v to 3.3 for ECG Sensor And Bluetooth Module The Output is can be viewed on Serial Monitor Via Bluetooth or wifi.System Can Be accessed From Anywhere while connecting sysem to internet via a nodemcu module.While running the program we are asked for choosing modes there are 3 modes

1.Manual Mode

2.Emotion Recognition Mode

3.Neural Interface Mode

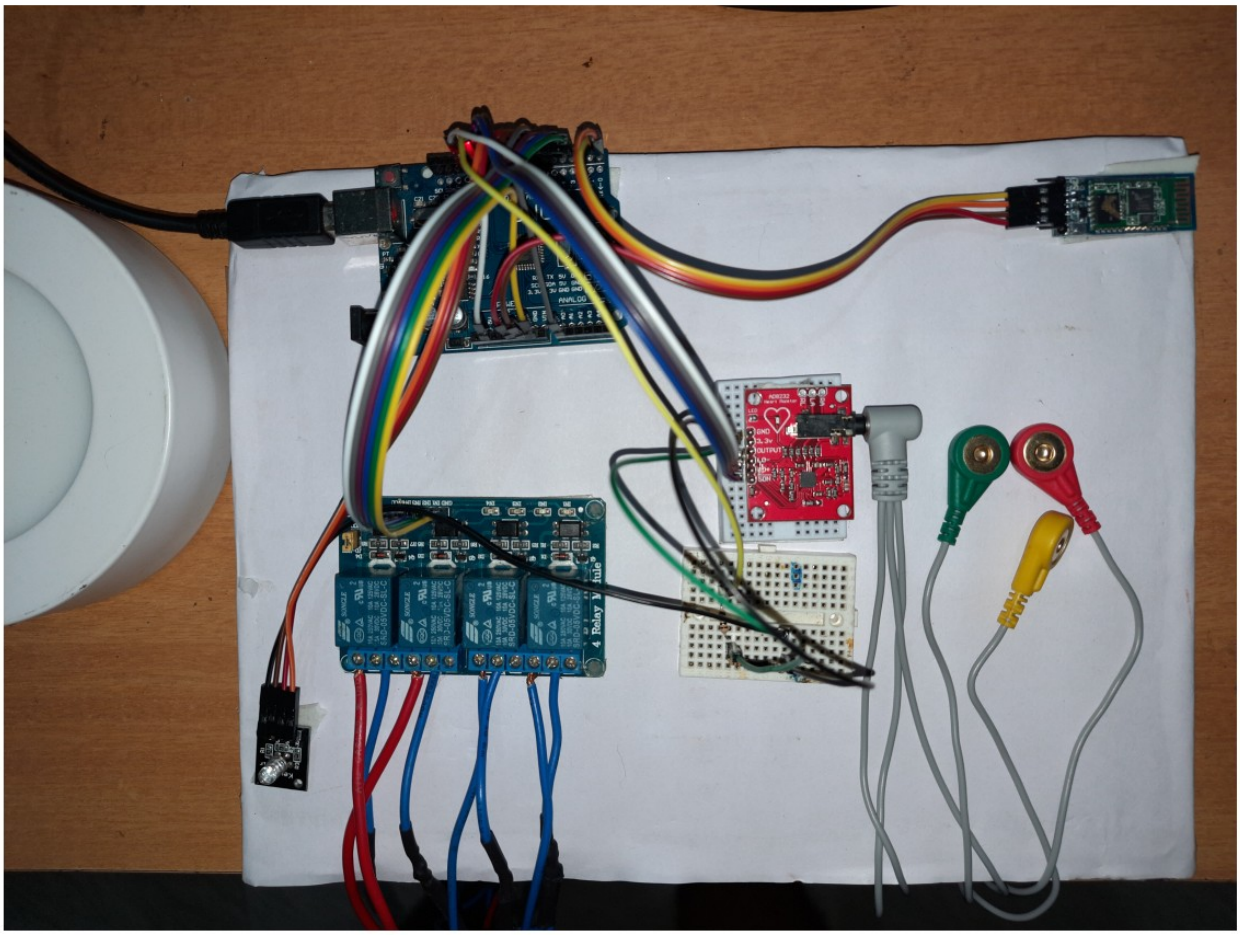

Figure 4.1: Experimental setup of System

.

In Manual Mode The system Can be overridden To manual Switching App Based or Voice Based Activation Can Be Used

In Emotion Recognition Mode, Connected Webcam Is Activated To Detect Emotion,when Emotion Detected Is Happy, Happy Profile Activated Where Ambient Lighting,Music,Heating or Cooling Apparatus related to profile is turned on same goes for all seven emotions.

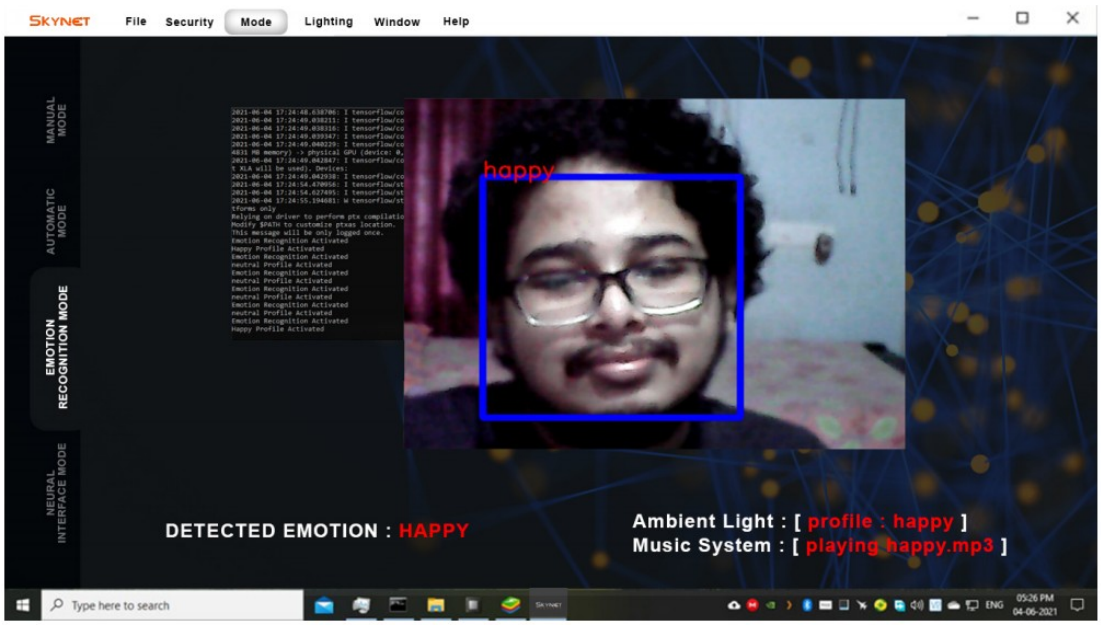

Figure 4.2: Developed User Interface of Emotion Recognition Mode

.

Table 4.1: Accuracy Obtained from Different Method

Sl.No	Approach used	No: of Classes	Emotions	Accuracy
1	Convolutional Neural Network	8	Anger, Contempt, Disgust, Fear, Happiness, Sad, Neutral, Surprise.	66.6%.
2	Haarcascade Classifier	8	All 8 emotions	79.80%
3	Linear Support Vector Machine	8	All 8 emotions	81.94%
4	Linear Support Vector Machine	5	Anger, Neutral, Disgust, Surprise, Happy	91.66%
5	Linear Support Vector Machine	3	Anger, Sadness, Happy	92.86%

From Table 1, it is clear that linear SVM is a powerful Machine Learning Classifier algorithm for the CK+ data set.But we used Haarcascade classifier due to its transparency and easy to work with than SVM, also it has almost 80% accuracy for all 8 emotions. It is also observed that on reduction of number of classes from 8 to 5 the accuracy improves from almost 82% to 91%. On further reduction of number of classes from 5 to 3 the accuracy goes up to 92.86% which is a better result. Hence, we have used Haarcascade learning model and have deployed it into the application where it recognizes if the user is angry, sad or happy. These results are solely on the basis of the data set that we have used. In case we use a different data set for training, these results will vary.

From our results, it is observed that decreasing the number of classes has increased the accuracy. This is because certain images in the data set for emotions like Surprise and Fear are similar and they have a significant number of samples that are unclassified between them. The removal of such similar classes which have increased chances of being unclassified will decrease the errors. The effects of reducing the number of classes are dependent on both the algorithm and the data set.

4.2 Neural Interface Mode

Electrodes are placed in the head such that they can detect electrical impulses from the frontal lobe of the brain, through the electrodes the detected impulses/waves are transmitted to ECG sensor. in EEG Sensor the Detected electrical impulses are measured as attention level and level value is transmitted to Arduino. in Arduino the level value is obtained by reading analog value through analog A0 pin. This value is cross-checked against the preset value in Arduino if both values equals connected light/relay will turn on else the system will keep reading on new values via analog pin A0.

Every Serial Port Commuincation Is Done Via Bluetooth And System Status,Graph Data and System Errors Are Logged To an class 10 Flash Memory Using Flash Memory Module for Offline Analysis

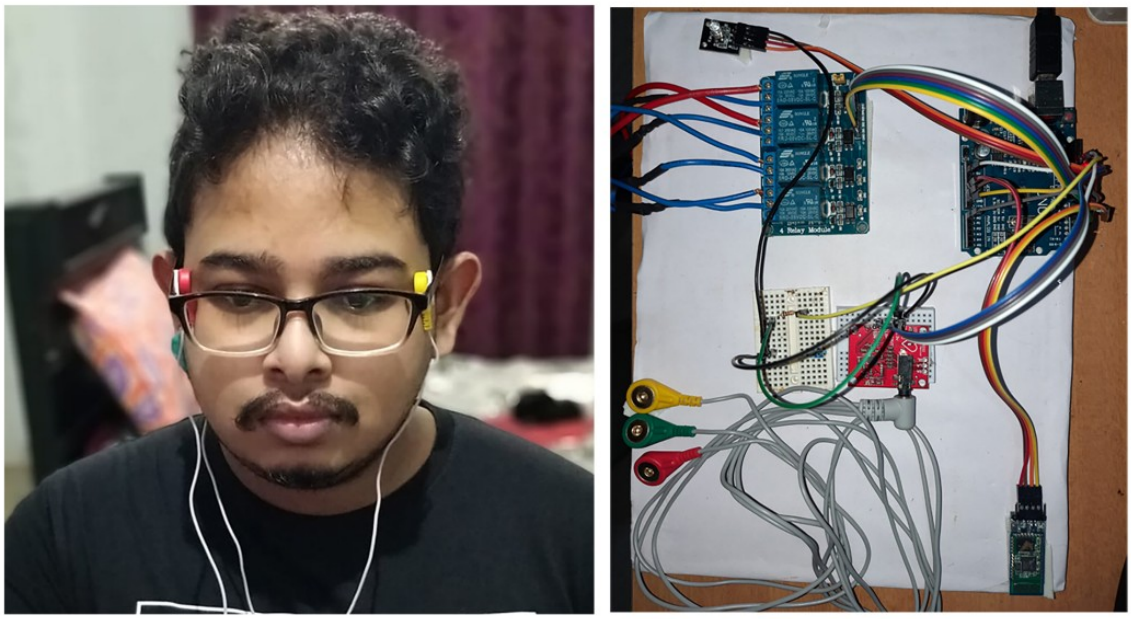

Figure 4.3: Circuit of Neural Interface Mode

.

```
void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the  pin as an output:

  pinMode(3,  OUTPUT); //Relay 1 gray
  pinMode(4,  OUTPUT); //Relay 2 black
  pinMode(5,  OUTPUT); //Relay 3 white
  pinMode(6,  OUTPUT); //Relay 4 orange
  pinMode(7,  OUTPUT);
  pinMode(8,  OUTPUT);
  pinMode(9,  OUTPUT);
  pinMode(10, INPUT); // Setup for leads off
  pinMode(11, INPUT); // Setup for leads off
  pinMode(12,  OUTPUT);
  pinMode(13,  OUTPUT);
```

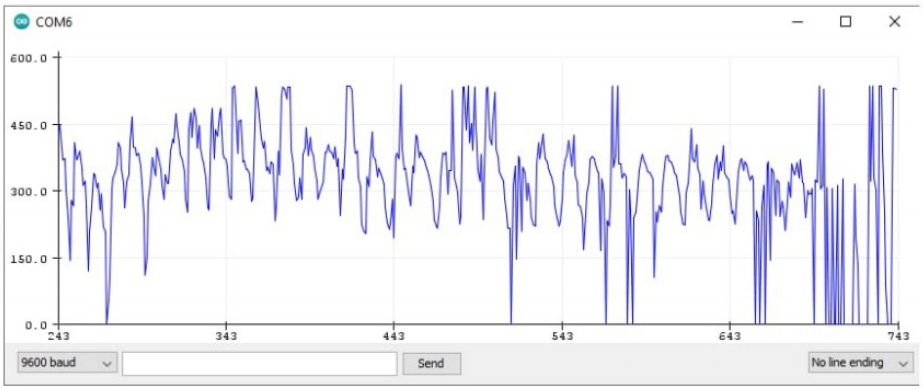

Figure 4.4: Brain Attention Level Detection Graph

.

Chapter 5

CONCLUSION

From Our Studies this System shows the potential to achieve an intelligent control system. It benefits in strict power control that helps in achieving energy savings.It also provide control and reliability. It provides security and comfort at the touch of your fingertips. incorporating neural interface to a automation system brings certain comfort to our day to day life. it speeds up our activities by directly sensing it from our thoughts and implementing it through our developed system,we don't have to waste any action to do those tasks. Moreover, remote monitoring can be setup easily and allows cost-efficient and intelligent conversion of existing home automation systems by incorporating advanced learning technology. The goal of Home Automation systems powered by This Learning Technology is to bring down any manual settings to zero. Due to ever expanding researches on brainwave controlled automation and emergence of new technologies on machine Learning we could see this system in full action in near future. Its Possible to Imagine Every Human Beings will be having a implant in their brains to interface with a computer and search the web in comforts of their own brain in near future. This System will not only change our workplace but will also change the way we live in our homes.

BIBLIOGRAPHY

[1] **Smart Home Automation Using Machine Learning Algorithms**
John Jaihar;Neehal Lingayat;Patel Sapan Vijaybhai;Gautam Venkatesh;K. P. Upla 2020 International Conference for Emerging Technology (INCET)

[2] **Youtube**
https://www.youtube.com/watch?v=DtBu1u5aBsc – Emotion Recognition keras model Training.
https://www.youtube.com/watch?v=fkgpvkqcoJc
https://www.youtube.com/watch?v=MrRGVOhARYY&t

[3] **Stack Overflow/python.**
https://stackoverflow.com/questions/59596748/warn0-global-sourcereadercbsourcereadercb-terminating-async-callback-wa
https://stackoverflow.com/questions/34550437/serialexception-could-not-open-port-access-is-denied
https://stackoverflow.com/questions/46288224/opencv-attributeerror-module-cv2-has-no-attribute-face

[4] **Git/dataset**
"Emotion Recognition using Facial Landmarks,Python, DLib and OpenCV"
https://github.com/vishalsingh020997/Emotion-Recognition-using-Facial-Landmarks0

[5] **Hackster.io.**

https://www.hackster.io/Imetomi/use-the-force-or-your-brainwaves-9e839b

https://www.hackster.io/akshay6766

https://blynk.hackster.io/ashshaks/control-your-home-appliances-using-your-voice-81d702

[6] **Other Sites**

https://neuralink.com/neuralink

https://www.analog.com/en/products/ad8232.html

https://www.tomsonelectronics.com/products/ecg-monitoring-module-arduino

Appendix A

APPENDIX A - TEST RESULTS

A.1 SCREENSHOTS

This section contains screenshots of system modelling and system working on different modes and its light description

Screenshots Include:-

Figure 6.1 :-Complete Setup Automation Using Emotion Recognition And Neural Interface

Figure 6.2: model is trained using fer2013.csv data set and Cuda ToolKit

Figure 6.3: Emotion Of user Is Detected and Respective profile is Switched

Figure 6.4: Brain Waves is Detected Using ECG Sensor

Figure 6.5: Attention Level Recording

Figure 6.6: Relay is On

A.1.1 Complete Setup

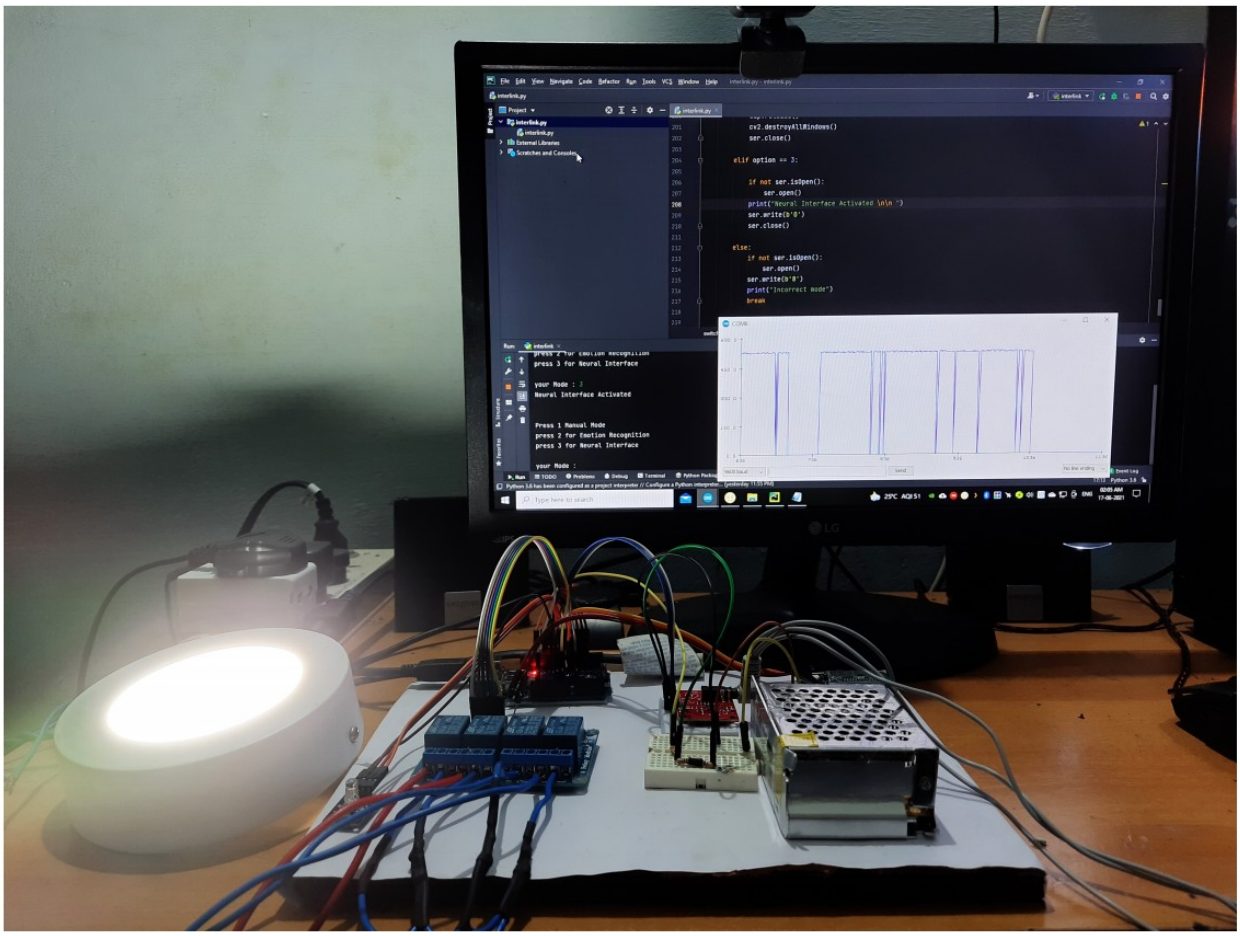

Figure A.1: Complete Setup Automation Using Emotion Recognition And Neural Interface

A.1.2 Stage 2 Emotion Recognition Model Training

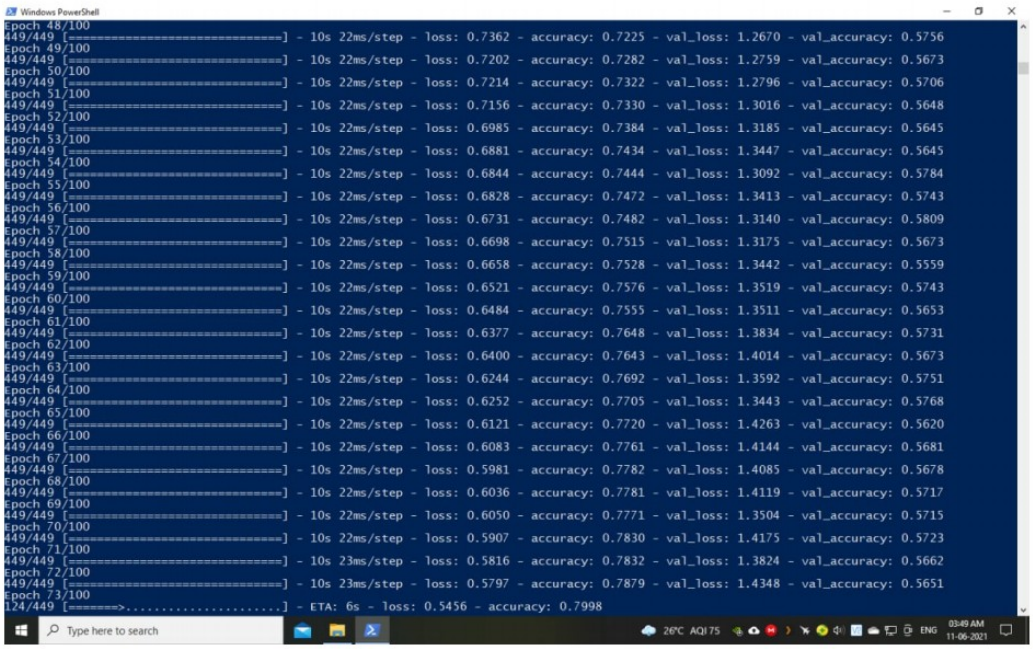

Figure A.2: model is trained using fer2013.csv data set and Cuda ToolKit

A.1.3 Stage 2 Face Detection And Emotion Recognition

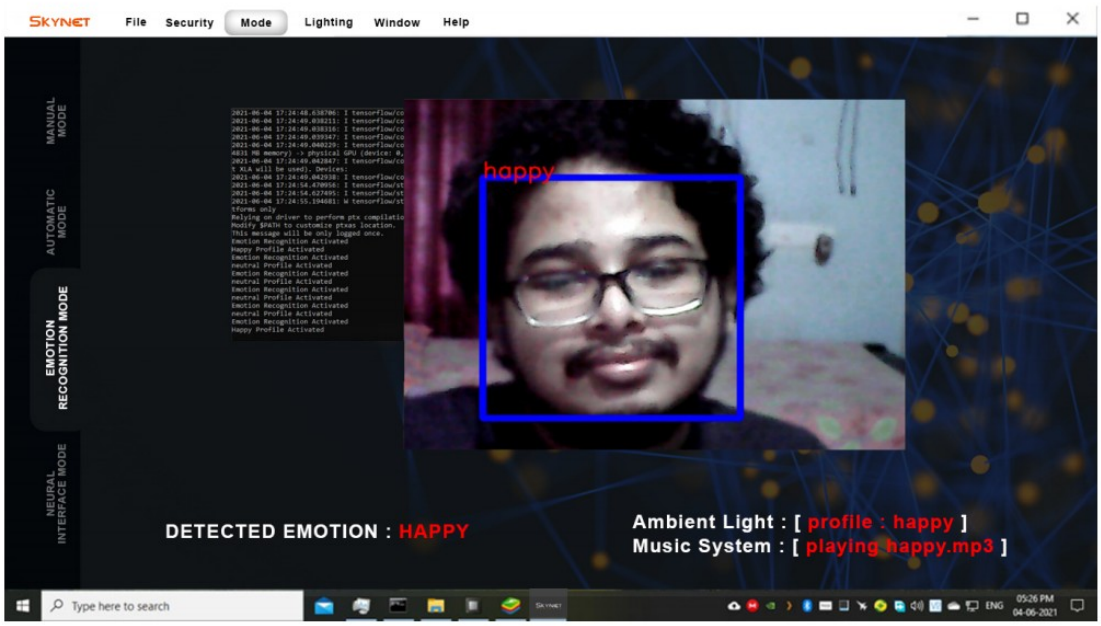

Figure A.3: Emotion Of user Is Detected and Respective profile is Switched

A.1.4 Stage 3 Neural Interface : ECG Sensor

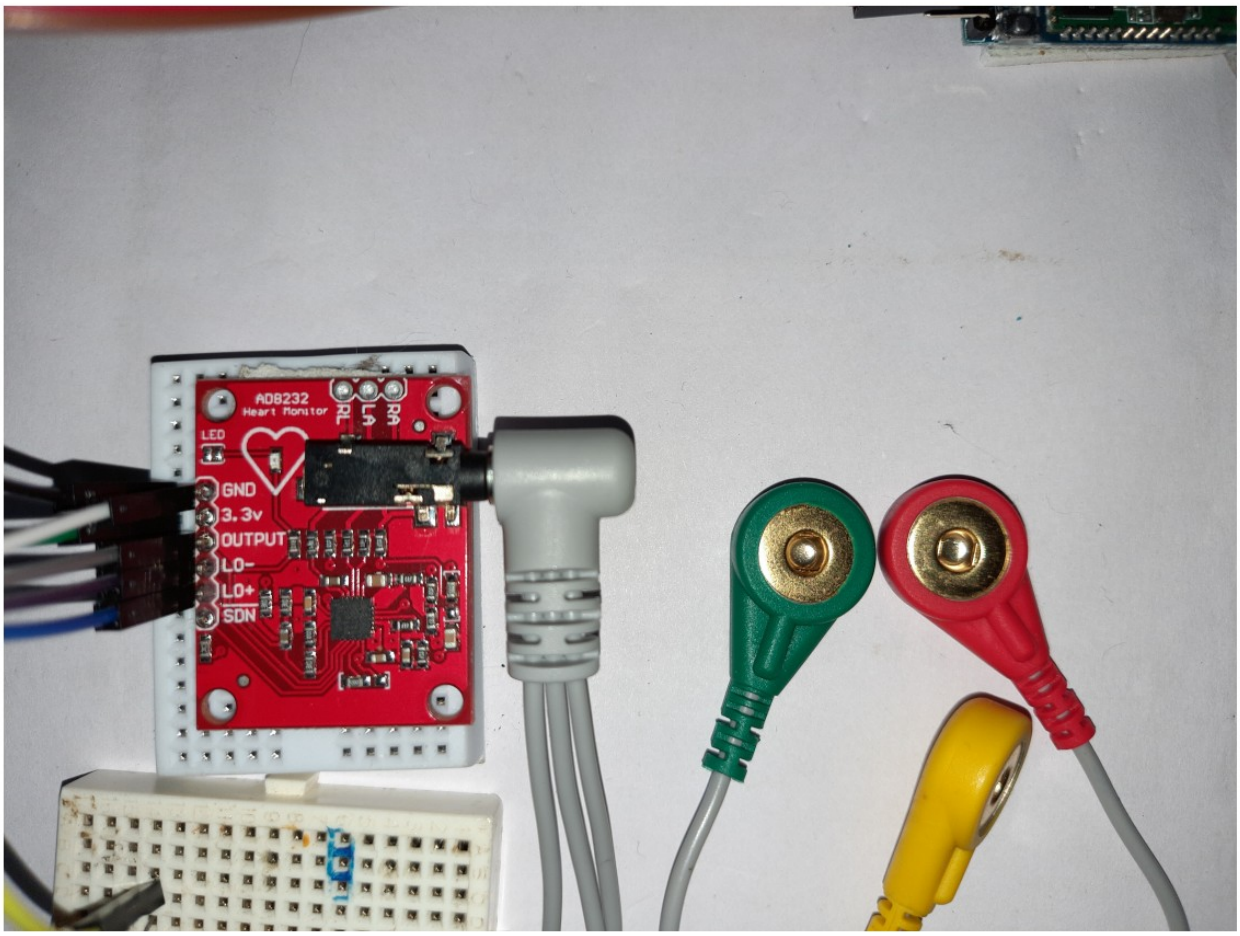

Figure A.4: Brain Waves is Detected Using ECG Sensor

A.1.5 Stage 3 Neural Interface Attention Level Is Recorded

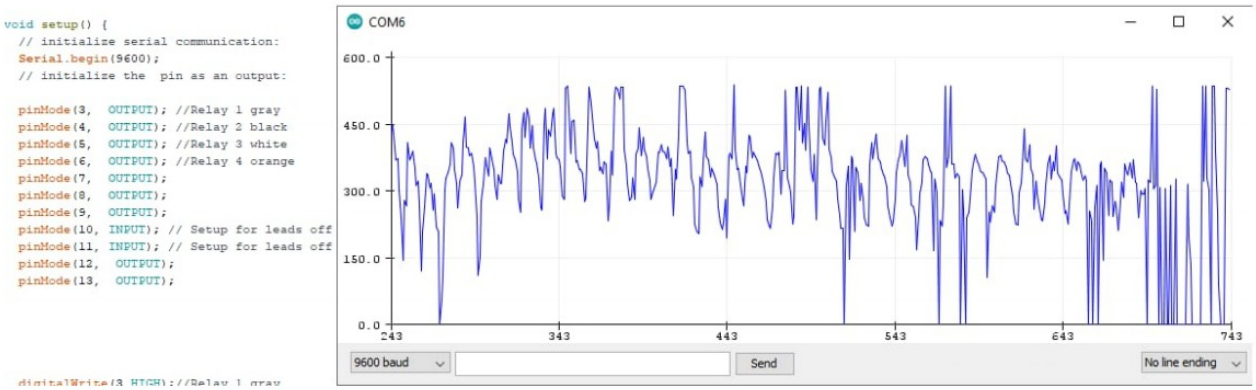

Figure A.5: Attention Level Recording

A.1.6 Stage 3 Neural Interface Attention Level Peek Detected

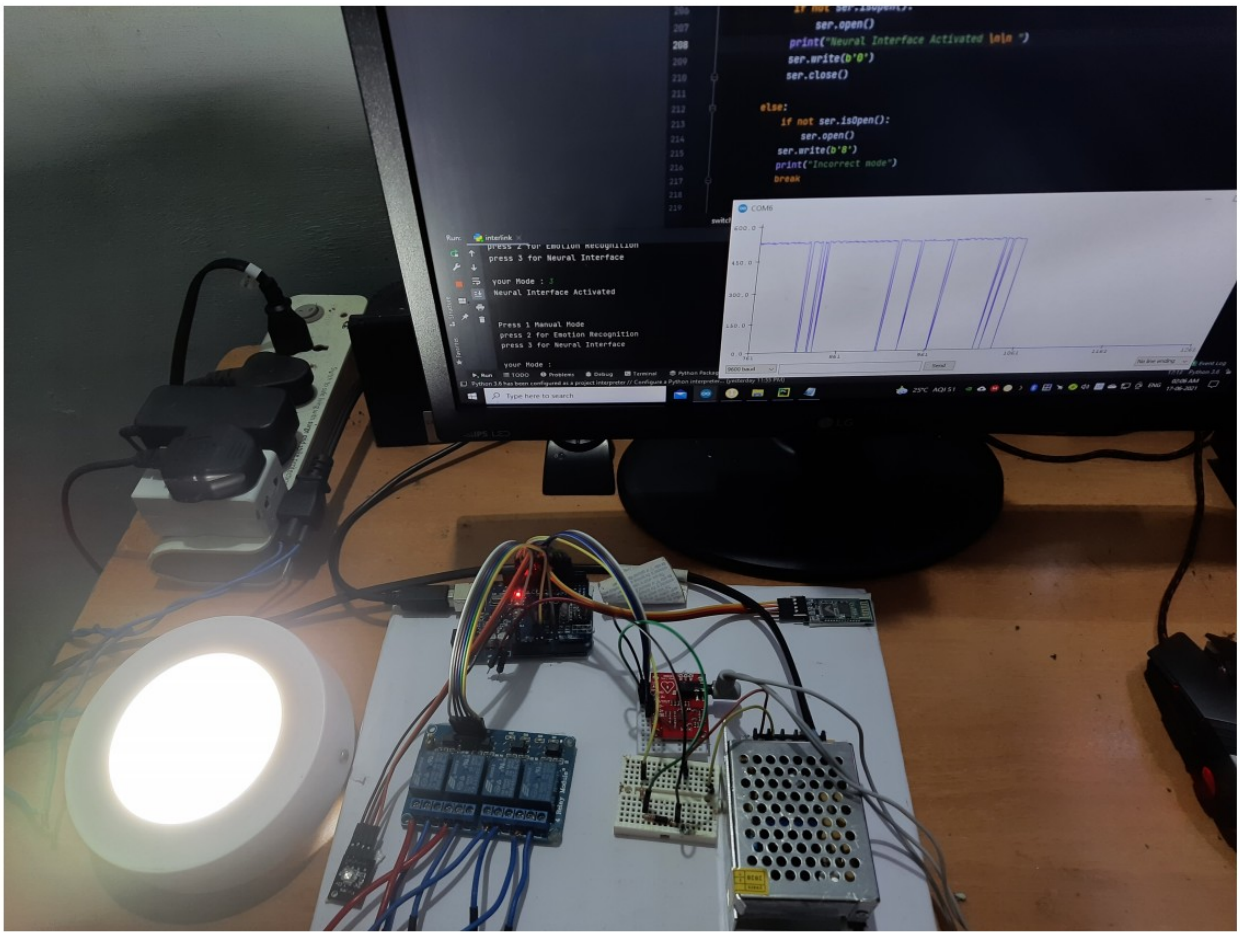

Figure A.6: Relay is On

Appendix B

APPENDIX B - TOOLS AND SPECIFICATIONS

Here we list out the used Hardware and software tools and its specifications used in this project

B.1 Hardware Requirements

Raspberry Pi.

Arduino.

Node MCU.

Graphic Card (Nvidia Geforce GTX 1060 6gb VRAM).

Relay 5V —— 230V 10A x 4.

Webcam 1080p.

EEG/ECG Sensor.

HC-05 Bluetooth Module x2.

SD Card Module.

jumper Wires.

LED Strips or Mood Light.

B.2 Software Requirements

Python version 3.9

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for web development (server-side), software development, mathematics, system scripting. Python can be used on a server to create web applications, alongside software to create workflows,to handle big data and perform complex mathematics. It can connect to database systems and also read and modify files. It is used for rapid prototyping, or for production-ready software development. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). It has a simple syntax similar to the English language and the syntax that allows developers to write programs with fewer lines than some other programming languages. The libraries used in this work are Opencv, Tensorflow, sklearn, Pandas, Numpy, Matplotlib, DlibPython runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. It can be treated in a procedural way, an object-orientated way or a functional way. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. It relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. Natural Language Processing with Python provides a practical introduction to programming for language

processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3. The libraries used in this work are Opencv, Tensorflow, sklearn, Pandas, Numpy, Matplotlib and Dlib. Python is a must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. The advantages of using Python are :-

• Easy-to-learn :Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

• Easy-to-read : Python code is more clearly defined and visible to the eyes.

• Easy-to-maintain : Python's source code is fairly easy-to-maintain.

• A broad standard library : Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.

• Interactive Mode : Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

• Portable : Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

• Extendable : You can add low-level modules to the Python interpreter.

• Databases : Python provides interfaces to all major commercial databases.

• GUI Programming :Python supports GUI applications that can be created and supported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

• Scalable : Python provides a better structure and support for large programs than shell scripting.

Aurduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the

help of third-party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision,also machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. Features of OpenCV Library are:-

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc

Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of version 2.4, only TensorFlow is supported.

TensorFlow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming.

CUDA Tool Kit

CUDA Toolkit Develop, Optimize and Deploy GPU-Accelerated Apps The NVIDIA CUDA Toolkit provides a development environment for creating high performance GPU-accelerated applications. With the CUDA Toolkit, you can develop, optimize, and deploy your applications on GPU-accelerated embedded systems, desktop workstations, enterprise data centers, cloud-based platforms and HPC supercomputers.

Appendix C

APPENDIX C - CODE

C.1 SYSTEM CODE

This Section Contains Complete Program Code Of Our System

C.1.1 Model Training Code

```
#Written In Python
#last updated 15/06/2021 by Akshay6766
import numpy as np
    import pandas as pd
    from keras.layers import Conv2D, MaxPooling2D
    from keras.layers import Dense, Dropout, Flatten
    from keras.losses import categorical_crossentropy
    from keras.models import Sequential
    from keras.optimizers import Adam
    from keras.utils import np_utils

    # pd.set_option('display.max_rows', 500)
    # pd.set_option('display.max_columns', 500)
```

```
# pd.set_option('display.width', 1000)

df = pd.read_csv('fer2013.csv')

# print(df.info())
# print(df["Usage"].value_counts())

# print(df.head())
X_train, train_y, X_test, test_y = [], [], [], []

for index, row in df.iterrows():
    val = row['pixels'].split(" ")
    try:
        if 'Training' in row['Usage']:
            X_train.append(np.array(val, 'float32'))
            train_y.append(row['emotion'])
        elif 'PublicTest' in row['Usage']:
            X_test.append(np.array(val, 'float32'))
            test_y.append(row['emotion'])
    except:
        print(f"error occurred at index :{index} and row:{row}")

num_features = 64
num_labels = 7
batch_size = 64
epochs = 100
width, height = 48, 48

X_train = np.array(X_train, 'float32')
```

```
train_y = np.array(train_y, 'float32')
X_test = np.array(X_test, 'float32')
test_y = np.array(test_y, 'float32')

train_y = np_utils.to_categorical(train_y, num_classes=num_labels)
test_y = np_utils.to_categorical(test_y, num_classes=num_labels)

# cannot produce
# normalizing data between o and 1
X_train -= np.mean(X_train, axis=0)
X_train /= np.std(X_train, axis=0)

X_test -= np.mean(X_test, axis=0)
X_test /= np.std(X_test, axis=0)

X_train = X_train.reshape(X_train.shape[0], 48, 48, 1)

X_test = X_test.reshape(X_test.shape[0], 48, 48, 1)

# print(f"shape:{X_train.shape}")
# designing the cnn
# 1st convolution layer
model = Sequential()

model.add(Conv2D(64, kernel_size=(3, 3),
activation='relu', input_shape=(X_train.shape[1:])))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
```

```
model.add(Dropout(0.5))

# 2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))

# 3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

model.add(Flatten())

# fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(num_labels, activation='softmax'))

# model.summary()

# Compiling the model
model.compile(loss=categorical_crossentropy,
```

```
                optimizer=Adam(),
                metrics=['accuracy'])

    # Training the model
    model.fit(X_train, train_y,
              batch_size=batch_size,
              epochs=epochs,
              verbose=1,
              validation_data=(X_test, test_y),
              shuffle=True)

    # Saving the  model to  use it later on
    fer_json = model.to_json()
    with open("fer.json", "w") as json_file:
        json_file.write(fer_json)
    model.save_weights("fer.h5")
```

C.1.2 Main Interlink Code

```
#Written In Python
#last updated 15/06/2021 by Akshay6766
import cv2
import pygame
from pygame import *
import numpy as np
import serial
from keras.models import model_from_json
```

```
from keras.preprocessing import image

ser = serial.Serial('COM10', 9600)


def switch():
    global ser
    while True:

    print("Press 1 Manual Mode\npress 2 for Emotion Recognition \n
    press 3 for Neural Interface \n")
       option = int(input("your Mode : "))

       if option == 1:

            if not ser.isOpen():
                ser.open()
            ser.write(b'9')
            ser.close()

        elif option == 2:

            if not ser.isOpen():
                ser.open()
            print("Emotion Recognition Camera Activated")

            # load model
            model = model_from_json(open("fer.json", "r").read())
            # load weights
```

```
model.load_weights('fer.h5')

face_haar_cascade = cv2.CascadeClassifier
('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)

while True:
    ret, test_img = cap.read()
    # captures frame and returns boolean value and captured image
    if not ret:
        continue
    gray_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)

    faces_detected = face_haar_cascade.detectMultiScale
    (gray_img, 1.32, 5)

    for (x, y, w, h) in faces_detected:
        cv2.rectangle(test_img, (x, y), (x + w, y + h),
        (255, 0, 0), thickness=7)
        roi_gray = gray_img[y:y + w, x:x + h]
        # cropping region of interest i.e. face area from  image
        roi_gray = cv2.resize(roi_gray, (48, 48))
        img_pixels = image.img_to_array(roi_gray)
        img_pixels = np.expand_dims(img_pixels, axis=0)
        img_pixels /= 255

        predictions = model.predict(img_pixels)
```

```
# find max indexed array
max_index = np.argmax(predictions[0])

emotions = ('angry', 'disgust', 'fear', 'happy', 'sad',
'surprise', 'neutral')
predicted_emotion = emotions[max_index]

cv2.putText(test_img, predicted_emotion,
(int(x), int(y)), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
            2)

if cv2.waitKey(10) == ord('e'):
    print("Emotion Recognition Activated")
    mixer.init()
    # profile 1
    if predicted_emotion == 'sad':
        # time.sleep(1)
        print("Sad Profile Activated")
        ser.write(b'3')
        # mixer.init()
        mixer.music.load('sad.ogg')
        mixer.music.play()

        while mixer.music.get_busy():
            time.Clock().tick(10)
            if cv2.waitKey(10) == ord('e'):
                pygame.mixer.fadeout(2)
                pygame.mixer.music.stop()
                pygame.mixer.quit()
```

```
                break


    break
# profile 2
if predicted_emotion == 'angry':
    # time.sleep(1)
    print("angry Profile Activated")
    ser.write(b'2')
    # mixer.init()
    mixer.music.load('angry.ogg')
    mixer.music.play()

    while mixer.music.get_busy():
        time.Clock().tick(10)
        if cv2.waitKey(10) == ord('e'):
            pygame.mixer.fadeout(2)
            pygame.mixer.music.stop()
            pygame.mixer.quit()
            break


    break
# profile 3
if predicted_emotion == 'disgust':
    # time.sleep(1)
    print("disgust Profile Activated")
    ser.write(b'7')
    # mixer.init()
    mixer.music.load('disgust.ogg')
    mixer.music.play()
```

```
        while mixer.music.get_busy():
            time.Clock().tick(10)
            if cv2.waitKey(10) == ord('e'):
                pygame.mixer.fadeout(2)
                pygame.mixer.music.stop()
                pygame.mixer.quit()
                break

        break
    # profile 4
    if predicted_emotion == 'fear':
        # time.sleep(1)
        print("fear Profile Activated")
        ser.write(b'6')
        # mixer.init()
        mixer.music.load('disgust.ogg')
        mixer.music.play()

        while mixer.music.get_busy():
            time.Clock().tick(10)
            if cv2.waitKey(10) == ord('e'):
                pygame.mixer.fadeout(2)
                pygame.mixer.music.stop()
                pygame.mixer.quit()
                break

        break
    # profile 5
```

```
if predicted_emotion == 'surprise':
    # time.sleep(1)
    print("surprise Profile Activated")
    ser.write(b'1')
    # mixer.init()
    mixer.music.load('disgust.ogg')
    mixer.music.play()

    while mixer.music.get_busy():
        time.Clock().tick(10)
        if cv2.waitKey(10) == ord('e'):
            pygame.mixer.fadeout(2)
            pygame.mixer.music.stop()
            pygame.mixer.quit()
            break

    break
# profile 6
if predicted_emotion == 'neutral':
    # time.sleep(1)
    print("neutral Profile Activated")
    ser.write(b'4')
    pygame.mixer.fadeout(2)
    pygame.mixer.music.stop()
    pygame.mixer.quit()

    break
# profile 7
if predicted_emotion == 'happy':
```

```
                # time.sleep(1)
                print("Happy Profile Activated")
                ser.write(b'5')
                # mixer.init()
                mixer.music.load('happy.ogg')
                mixer.music.play()

                while mixer.music.get_busy():
                    time.Clock().tick(10)
                    if cv2.waitKey(10) == ord('e'):
                        pygame.mixer.fadeout(2)
                        pygame.mixer.music.stop()
                        pygame.mixer.quit()
                        break

                break

    resized_img = cv2.resize(test_img, (800, 600))
    cv2.imshow('Skynet ', resized_img)

    if cv2.waitKey(10) == ord('q'):
        ser.write(b'8')

        pygame.mixer.music.stop()
        pygame.mixer.quit()
        break

cap.release()
cv2.destroyAllWindows()
```

```
            ser.close()

        elif option == 3:

            if not ser.isOpen():
                ser.open()
            print("Neural Interface Activated \n\n ")
            ser.write(b'0')
            ser.close()

        else:
            if not ser.isOpen():
                ser.open()
            ser.write(b'8')
            print("Incorrect mode")
            break


switch()
ser.close()
```

C.1.3 Aurduino Interlink Code

```
//#Written In C

//last updated 15/06/2021 by Akshay6766
```

```
int incomingByte;// a variable to read incoming serial data into
int count = 0;
int i = 0;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the  pin as an output:

  pinMode(3,  OUTPUT); //Relay 1 gray
  pinMode(4,  OUTPUT); //Relay 2 black
  pinMode(5,  OUTPUT); //Relay 3 white
  pinMode(6,  OUTPUT); //Relay 4 orange
  pinMode(7,  OUTPUT);
  pinMode(8,  OUTPUT);
  pinMode(9,  OUTPUT);
  pinMode(10, INPUT); // Setup for leads off detection LO +
  pinMode(11, INPUT); // Setup for leads off detection LO -
  pinMode(12,  OUTPUT);
  pinMode(13,  OUTPUT);




  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,HIGH);//Relay 2 black
```

```
  digitalWrite(5,HIGH);//Relay 3 white
  digitalWrite(6,HIGH);//Relay 4 orange
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);//status



}

void loop() {
  // see if there's incoming serial data:
  if (Serial.available() >= 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();

    if (incomingByte == '1') {
      //profile 1
      delay(100);

      digitalWrite(3,LOW);//Relay 1 gray
      digitalWrite(4,HIGH);//Relay 2 black
      digitalWrite(5,HIGH);//Relay 3 white
      digitalWrite(6,HIGH);//Relay 4 orange not using
      digitalWrite(7,LOW);
      digitalWrite(8,LOW);
```

```
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '2') {
  //profile 2
  delay(100);
  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,LOW);//Relay 2 black
  digitalWrite(5,HIGH);//Relay 3 white
  digitalWrite(6,HIGH);//Relay 4 orange not using
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '3') {

  //profile 3
  delay(100);
  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,HIGH);//Relay 2 black
  digitalWrite(5,LOW);//Relay 3 white
  digitalWrite(6,HIGH);//Relay 4 orange not using
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
```

```
if (incomingByte == '4') {

  //profile 4
  delay(100);
  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,HIGH);//Relay 2 black
  digitalWrite(5,HIGH);//Relay 3 white
  digitalWrite(6,LOW);//Relay 4 orange not using
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '5') {

  //profile 5
  delay(100);
  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,HIGH);//Relay 2 black
  digitalWrite(5,HIGH);//Relay 3 white
  digitalWrite(6,HIGH);//Relay 4 orange not using
  digitalWrite(7,HIGH);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '6') {
        //profile 6
```

```
  delay(100);
  digitalWrite(3,HIGH);//Relay 1 gray
  digitalWrite(4,HIGH);//Relay 2 black
  digitalWrite(5,HIGH);//Relay 3 white
  digitalWrite(6,HIGH);//Relay 4 orange not using
  digitalWrite(7,LOW);
  digitalWrite(8,HIGH);
  digitalWrite(9,LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '7') {
  //profile 7
  delay(100);
 digitalWrite(3,HIGH);//Relay 1 gray
 digitalWrite(4,HIGH);//Relay 2 black
 digitalWrite(5,HIGH);//Relay 3 white
 digitalWrite(6,HIGH);//Relay 4 orange not using
 digitalWrite(7,LOW);
 digitalWrite(8,LOW);
 digitalWrite(9,HIGH);
 digitalWrite(13,HIGH);
}
if (incomingByte == '8') {
  //all off
  //profile 8
  delay(100);
  digitalWrite(3, HIGH);//Relay 1 gray
  digitalWrite(4, HIGH);//Relay 2 black
```

```
  digitalWrite(5, HIGH);//Relay 3 white
  digitalWrite(6, HIGH);//Relay 4 orange not using
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(13,LOW);
}
if (incomingByte == '9') {
  //all on
  //Conected Appliance Self Test
  //test connected lights,massager
  //profile 9
  delay(100);
  while(i<=50)
    {
    digitalWrite(3, LOW);//Relay 1 gray
    delay(500);
    digitalWrite(3, HIGH);//Relay 1 gray
    delay(500);
    digitalWrite(4, LOW);//Relay 2 black
    delay(500);
    digitalWrite(4, HIGH);//Relay 2 black
    delay(500);
    digitalWrite(5, LOW);//Relay 3 white
    delay(500);
    digitalWrite(5, HIGH);//Relay 3 white
    delay(500);
    digitalWrite(6, LOW);
```

```
      delay(500);
      digitalWrite(6, HIGH);//Relay 4 orange not using
      delay(500);
      digitalWrite(7, HIGH);
      delay(500);
      digitalWrite(7, LOW);
      delay(500);
      digitalWrite(8, HIGH);
      delay(500);
      digitalWrite(8, LOW);
      delay(500);
      digitalWrite(9, HIGH);
      delay(500);
      digitalWrite(9, LOW);//Relay 3 white
      delay(500);
      //digitalWrite(13,HIGH);
      i++;

      }
}
if (incomingByte == '0') {
  //profile 10
  digitalWrite(12,HIGH);
  while (true)
  {
    if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
    //Serial.println('!');
      }
```

```
            else{
            // send the value of analog input 0:
            Serial.println(analogRead(A0));
            delay(100);
            if (analogRead(A0) >= 450)

                digitalWrite(7,HIGH);
            else
                digitalWrite(7,LOW);




            }
            //Wait for a bit to keep serial data from saturating
            delay(1);



        }


    }
  }
}
```