**Chapter 1**

# INTRODUCTION

**1.1 BACKGROUND**

In today's world need of automation is become necessary not only to reduce human effort but also to utilize maximum use of the technology and to do everything smartly and efficiently in order to reduce both energy and time consumption. The idea of PC automation basically deals with controlling the computer and applications in it through Infrared remote just like a TV which helps users to perform their tasks comfortably without any hassle [1]. The remote control is a device used to control another device remotely, usually wirelessly. In home electronics, a remote control can be used to operate devices such as a TV set, DVD player, or any other home appliance. The same remote can be used to control personal computers also.

**1.2 PROBLEM STATEMENT**

Although we have wireless mouse and keyboard for controlling the computer it is sometimes inconvenient and uncomfortable to use them for purposes like controlling audio/video while watching movies, controlling PowerPoint presentations etc. We need a system where user need not remember shortcuts to operate applications and can perform operations like opening, closing, moving through files etc. In the proposed project users can reuse their old IR remotes by mapping buttons to key strokes/functions that they require to perform. Thus using a remote to control PC is more convenient and offers more freedom of movement and comfort.

The proposed system uses IR sensor interfaced with Arduino [2, 3]. The IR sensor receives signals from the remote and decodes them into binary / hex code. A GUI application written using tkinter library in python interprets these hex codes via serial port which then maps these codes to keystrokes/functions that are performed by users for specific task using a library called pyautogui [4]. Whenever a user presses the button on remote the keystroke/function assigned to that button is executed thus simulating user actions.

**1.3 OBJECTIVE**

The major objectives of the proposed project are:

1. To study and understand the working of IR remote control and its applications.
2. To design and implement IR Remote control to control computer programs.
3. To automate user tasks that requires more than one key combination.
4. To effectively reuse and utilize old IR remotes.
5. To control computer applications with ease.

**1.4 ORGANIZATION OF MINI-PROJECT REPORT**

To make the understanding of the realised work easier, the report contains eight chapters which can be explained as follows:

**Chapter 1**: The background, problem statement and objective of this project are described in this chapter.

**Chapter 2**: This chapter describes the various literature surveys and existing work related to this project and evaluates the current work with the previous one.

**Chapter 3:** This chapter describes software requirements specification, which enlists all necessary requirements that are required for the project development.

**Chapter 4**: This chapter explains the high level design of the project. It explains about the components used, system architecture and the use case diagram.

**Chapter 5**: This chapter gives a brief overview of the circuit connection, decoding of IR signals and application design.

**Chapter 6**: This chapter has a detailed overview of the various tests performed on the system.

**Chapter 7**: In this chapter the results of the different aspects of the system are explained with figures. Finally the report ends with applications, conclusion and future enhancements.

**Chapter 2**

# LITERATURE SURVEY

In order to come up with the idea of the project and to identify various strategies and techniques used in the project, it was necessary to conduct a comprehensive literature search and a thorough review of existing work. We surveyed around 5 different research papers which are described below:

1. **IR Remote Control Signal Decoder for Home Automation by** Samiran Maiti, 2014**: This research paper** describes a design and implementation of an infrared (IR) remote control signal decoder which can be used for various home control applications.

<https://www.researchgate.net/publication/343054852_IR_Remote_Control_Signal_Decoder_For_Home_Automation>

1. Utilization of Serial Communication in Arduino by Osisiogu & Ukachi, 2015: This paper gives an overview of Arduino, serial communication and then briefly explains how Arduino utilizes such feature.

<https://www.researchgate.net/publication/327285060_Seminar_Paper_on_Serial_Communication>

1. Working Principle of Arduino and using it as a Tool for Study and Research by Leo Louis, 2018: This paper explores the working principle and applications of an Arduino board. This paper provides a glimpse of type of Arduino boards, working principles, software implementation and their applications.

<https://www.researchgate.net/publication/326316390_Working_Principle_of_Arduino_and_Using_it_as_a_Tool_for_Study_and_Research>

1. Robotic process automation by Peter Hofmann, Caroline Samp & Nils Urbach, 2019: This paper focuses on Robotic process automation (RPA) which is a technology for centralized automation of business processes. RPA automates user interaction with graphical user interfaces. <https://www.researchgate.net/publication/336769927_Robotic_Process_Automation>
2. Arduino Based Control And Data Acquisition System Using Python Graphical User Interface (GUI) by [Farid Khan](https://www.researchgate.net/profile/Farid-Khan-4), [Ahmad Masood](https://www.researchgate.net/profile/Ahmad-Masood-2) & [Atal Khattak](https://www.researchgate.net/scientific-contributions/Atal-Khattak-2101636880), 2021: This paper presents the development of a control and data acquisition system for a machines and equipments. Tkinter toolbox in Python language libraries is used to create the GUI, while Arduino acts as intermediary between the system and the computer.

<https://www.researchgate.net/publication/352786556_Arduino_Based_Control_And_Data_Acquisition_System_Using_Python_Graphical_User_Interface_GUI>

**Chapter 3**

# REQUIREMENTS SPECIFICATION

### 3.1 FUNCTIONAL REQUIREMENTS

* User should be able to map inputs to outputs.
* User should be able to control pc within 10-20m distance.
* User should be able to Control sequence of execution.
* User should be able enter COM port no. to which the device is connected to.
* The system need to display the key pressed and function performed.

### 3.2 NON-FUNCTIONAL REQUIREMENTS

* Performance
* Portability
* Extensibility
* Reliability
* Compatibility

### 3.3 HARDWARE REQUIREMENTS

* **Processor:** Intel i7 10th Gen
* **Processor Speed:** 2.4 GHz
* **Ram:** 8 Gb
* **Hard Disk:** 10 Gb
* **Others:** Arduino UNO, IR Remote, TSOP 1838 Receiver

### 3.4 SOFTWARE REQUIREMENTS

* **Operating System:** Windows 7/8/10/11
* **Integrated Development Environment:** Visual Studio Code, Arduino IDE
* **Language:** Arduino C/C++ with IR Remote Library, Python
* **Python Libraries:** serial, tkinter, pyautogui, threading

**Chapter 4**

# SYSTEM DESIGN

### 4.1 SYSTEM COMPONENTS

* **ARDUINO UNO**

The Arduino Uno is an open source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino Company.

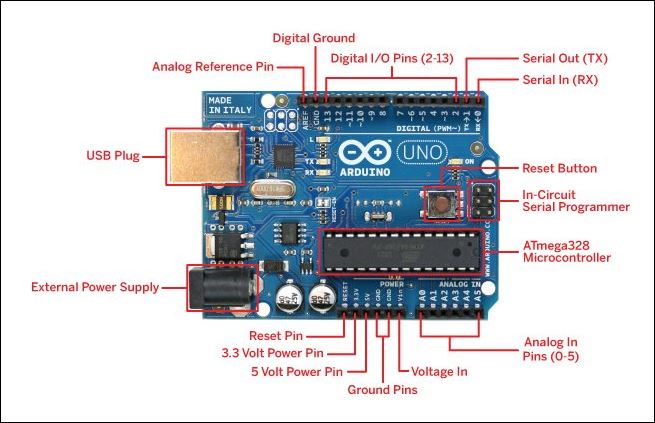
****

Fig 4.1 Arduino UNO Microcontroller

* **IR REMOTE**

The IR remote is generally used in home theatres and is based on the principle of using infrared light as the medium of communication. Today’s modern remote controls work by modulating the output from an infra-red LED. A series of pulses usually 10-20 pulses of varying width are sent to a gate that turns on or off, the modulator which is usually 38 kHz. The reason for modulation is to separate the remote IR range from the IR light emitted by other bodies in the vicinity. Usually, it requires a line of sight communication. When a button is pressed, the corresponding circuitry gets connected to bias the IR LED which emits IR light which contains the input. This output in the form of light pulses is pulse width modulated at 38 kHz frequency, which is obtained at the receiver by demodulation.

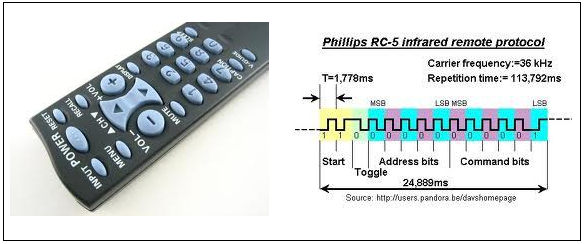
 

Fig 4.2 Infrared Remote Protocol Fig 4.3 TV Setup Box Remote used in the project

In the receiver, there is a tone decoder, which responds well to whatever signals the remote sends at a carrier frequency of 38 kHz. The microprocessor decodes the series of pulses and determines whether it is valid and if it is, will respond to that function.

* **IR RECEIVER**

TSOP1738 is an IR receiver with an amplifier that acts as a switch and converter within a circuit. The basic purpose of TSOP1738 is to convert the IR signal to electric signals. Every IR receiver has a special frequency to operate. TSOP1738 operates on 38KHz IR frequency. In case of higher or lower frequency, it may act due to a current leakage or some other errors but it won’t fully operate. It uses silicon-based technology, which works at the microlevel and very sensitive and efficient to its functions. In summary, TSOP may be smaller in size but its usage with microcontroller and microprocessors makes it smart and secure. TSOP1838 is an improved version of TSOP1738.

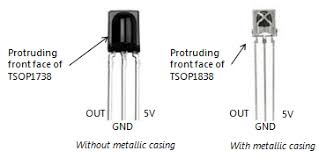


Fig 4.4 IR Led and TSOP Receivers

* **PERSONAL COMPUTER**

Fig 4.5 Desktop and Laptop

A personal computer (PC) is a multi-purpose microcomputer whose size, capabilities, and price make it feasible for individual use. Personal computers are intended to be operated directly by an [end user](https://en.wikipedia.org/wiki/End_user). Personal computer can be either a desktop or laptop. It contains system software and application software that are used by the end users. Applications like MS Word, MS PowerPoint, camera, music player and games are most common for personal use.

### 4.2 SYSTEM ARCHITECTURE

### 

Fig 4.6 System architecture of proposed project

When any button of the IR Remote is pressed, the remote sends a code in form of train of encoded pulses using 38Khz modulating frequency. These pulses are received by TSOP1838 sensor and read by Arduino and then Arduino decodes received train of pulse into a hex/bin value and in turn will display it on the serial monitor.

Once you run the Python program a GUI application designed using tkinter opens prompting for the serial port through which it communicates to Arduino. Once you establish connection.

Arduino continuously monitors the infrared receivers. Once the signal from remote is received by Arduino The python program fetches these codes through serial communication using serial module and assigns specific functions which will be simulate events like mouse click , keypress etc. using the pyautogui module that performs particular task like controlling volume , play/pause, presentation etc.

### 4.3 USE CASE DIAGRAM

### 

Fig 4.7 Use case diagram

Table 4.1 Use case for User

|  |  |
| --- | --- |
| Use Case | Establishing Connection with Arduino and PC |
| Actors | User |
| Precondition | IR Receiver to be connected to Arduino |
| Scenario | 1.User connects the Arduino to PC via USB  2.Serial Communication is established between Arduino and PC  3.User note downs the COM port no. to which Arduino is connected through device manager.  4.User starts the application and enters this port no. and establishes connection between Arduino and Application |
| Post Condition | PC is ready to receive signals from IR Remote |

Table 4.2 Use case for Arduino

|  |  |
| --- | --- |
| Use Case | Decoding IR signals |
| Actors | Arduino |
| Precondition | Arduino and Application should have connection. |
| Scenario | 1.IR receiver receives signal from IR Remote  2.Arduino interprets IR signals and decodes into HEX codes.  3.This HEX codes are read by application via serial communication. |
| Post Condition | IR Signal is converted to HEX codes |

Table 4.3 Use case for Application

|  |  |
| --- | --- |
| Use Case | Simulating events and actions |
| Actors | Application |
| Precondition | Application should be able to interpret IR signals in the form of HEX codes. |
| Scenario | 1.Application understands HEX codes.  2.Necessary actions are performed based on the mappings of functions to be performed on keypress of IR remote. |
| Post Condition | User actions and events are simulated. |

**Chapter 5**

# IMPLEMENTATION

### 5.1 CIRCUIT CONNECTIONS

Connect the signal pin (first from left) on TSOP 1838/1738 to digital pin 10 of arduino, gnd pin (middle one) of TSOP to digital pin 9 of arduino and vcc pin (right extreme) of TSOP to digital pin 8 of arduino.

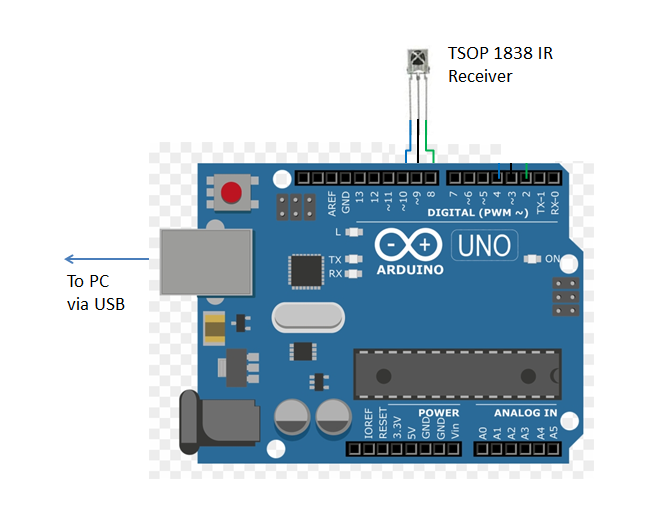


Fig 5.1 TSOP 1838 IR Receiver connected to Arduino

### 5.2 DECODING IR SIGNAL

When the IR remote is pressed it sends out IR radiation. The TSOP1738 detects the IR which is switching on and off at the rate of 38Khz. TSOP’s output is active low, means its output will continue to remain HIGH when there is no IR, and becomes low when it detects IR radiation. So as to eliminate interference, the TSOP1738 operates at a particular frequency so that other IRs in the environment can’t interfere, except the modulated IR of particular frequency.

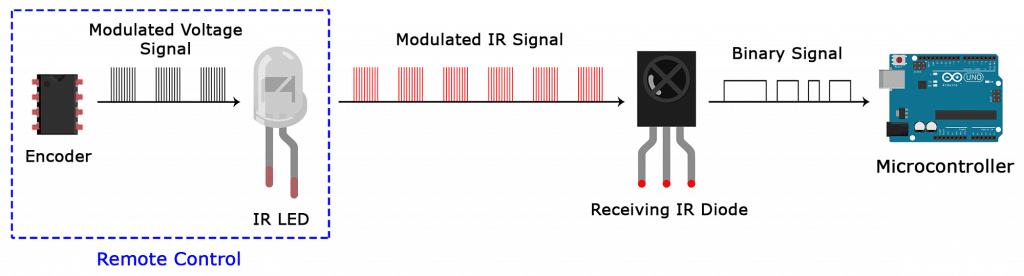


Fig 5.2 Arduino receiving signal from remote control

For decoding IR codes programmatically an Arduino program is written using IRremote library which runs repeatedly whenever Arduino is powered on. The Arduino constantly monitors the IR receiver pin for incoming signals. Once it detects the IR signal it decodes it into hexadecimal (HEX) code and sends the data through the serial communication (COM) port. This data can be accessed by other applications through the use of built-in libraries like pyserial in python which further can be processed for obtaining useful insights and other applications. The mapping of IR Codes of TV remote shown in Fig 4.3 is given below.

**Table 5.1 Mapping of TV remote shown in Fig 4.3 with its HEX Code and the events they perform.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **TV Remote**  **button** | **HEX CODE** | **Event** |
| 1 | Power button | F40B9B44 | Close/Exit |
| 2 | 1 | E41B9B44 | Opens file explorer |
| 3 | 2 | F00F9B44 | Opens spotify |
| 4 | 3 | FC039B44 | Opens game |
| 5 | 4 | E6199B44 | Opens camera |
| 6 | 5 | EE119B44 | Opens ppt presentation |
| 7 | RECALL | EF109B44 | Switches between opened applications |
| 8 | BACK | EC139B44 | Backspace or goes back to previous directory while explorer is opened |
| 9 | Arrow keys | E7189B44 (UP)  E9169B44 (DOWN)  9B649B44 (LEFT)  9A659B44 (RIGHT) | To navigate up/down/left/right |
| 10 | OK | F9069B44 | Enter |
| 11 | Vol+/- | B44B9B44 (Vol+)  B54A9B44 (Vol-) | Increases or Decreases Volume |
| 12 | CH+/- | B34C9B44 (CH-)  B24D9B44 (CH+) | Plays next song in spotify |
| 13 | PG+/- | 20DF00FF (PG+)  20DF807F (PG-) | Page up and down |
| 14 | Play button | AB549B44 | Space or Play/Pause the song |
| 15 | Bottom Red button | FA059B44 | Opens start menu |
| 16 | EPG | E31C9B44 | ESC |
| 17 | Bottom Blue button | E11E9B44 | F5/PowerPoint Fullscreen |
| 18 | TV button | EF19B44 | Minimizes window |

**5.3 APPLICATION DESIGN**

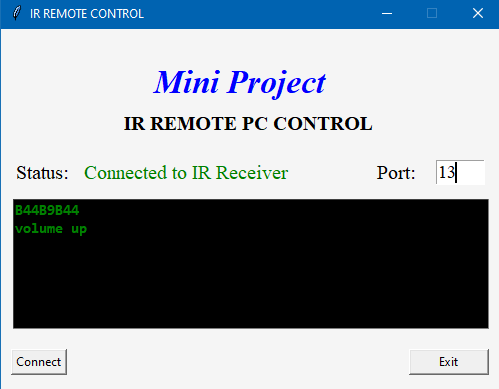


Fig 5.3 GUI Application for the proposed project

The application runs in two different threads one for updating GUI and other for serial communication between Arduino for obtaining the serial data that is stored in the serial buffer. The application interface is designed using tkinter python library. The application contains graphical user interface components like labels, textboxes and buttons. The application has a text box where the COM port of Arduino has to be entered in order to establish communication between Arduino and the application. Once the user enters the port no. and clicks connect the application shows status of the connection as connected. The application displays the HEX codes of the buttons pressed and necessary actions are performed based on how the keys are mapped to perform actions.

The mappings of IR code and actions to be performed are written using simple if else statements. If a particular IR code is present in serial data the if block is executed and pyautogui triggers the events.

**Chapter 6**

# SYSTEM TESTING

### System testing is the process of evaluating system's functionality with its requirements. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives. When the design is complete, coding follows and the finished code is then tested at the unit or module level by each programmer, at the component level by the group of programmers involved and at the system level when all components are combined together.

### 6.1 HARDWARE TESTING

The working of Arduino was tested by connecting it to PC via USB and checking whether PC recognizes the device or not. If the connection is successful the device manager should show the COM port of the PC where the Arduino is connected. If it doesn’t show then it could be hardware problem or the device driver required for Arduino may not be installed in the PC.

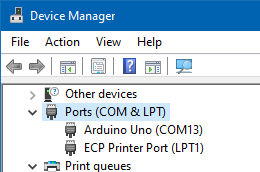
 

Fig 6.1 Windows Device Manager Fig 6.2 Arduino UNO ready to receive signal from IR remote

### 6.2 SOFTWARE TESTING

### Software testing process includes three types of testing. The first is the unit testing where in each module is tested for its functionality. The second is integration testing, where in the test is made on the modules integrated together performs the required task. The third is the functional testing, where the system as a whole meets the requirements. Finally User Acceptance testing is done to ensure that the software meets the user needs.

**6.2.1 UNIT TESTING**

Unit testing is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is smallest testable plan of any software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure etc. In object-oriented programming the smallest unit is a method which may belong to a base/super class, abstract or derived/child class.

**Table 6.1 Unit Testing for the application**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNIT TESTING** | | | | |
| **Module** | **Input Value** | **Expected Value** | **Result** | **No. of Test Cases** |
| Keybindings.py | IR HEX codes in the form of serial data | Events and actions | Pass | 18 |
| Main.py | COM Port no. of Arduino | Application must be connect to Arduino | Pass | 3 |

**6.2.2 INTEGRATION TESTING**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

**6.2.3 FUNCTIONAL TESTING**

It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.

### 6.2.3.1 GUI TESTING

GUI Testing is a process of testing the application’s graphical user interface to ensure proper functionality as per the specifications. The home screen of the project is shown in Fig 6.3. The right hand side of the screen has textbox to enter COM port no. of Arduino. The center of the screen has a window to display the available COM ports. The bottom left side of the screen has button to connect the application with Arduino once the port no. is entered. If the user presses connect button without entering the port no. it will be in disconnected state as shown in Fig 6.4.

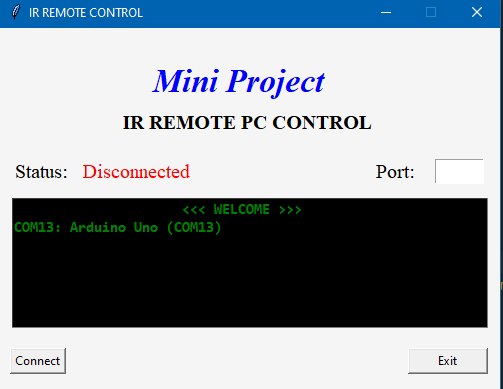


Fig 6.3 Application in disconnected state and showing available COM ports on the computer

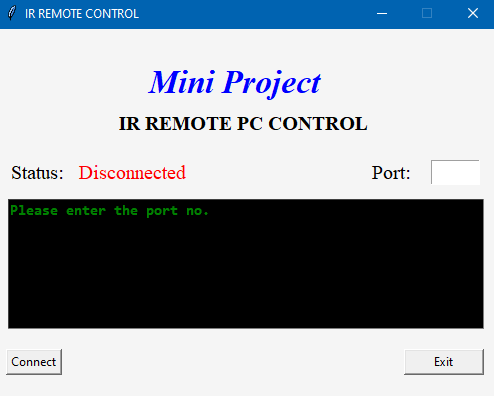


Fig 6.4 Application in disconnected state and asking the user to enter the port number

Once the user enters correct port no. and clicks connect the status of the application is changed and it enables the Arduino to receive for IR signals as shown in Fig 6.5. Once the user presses any button on the TV remote its HEX code is displayed and necessary event occurs as shown in Fig 6.6.

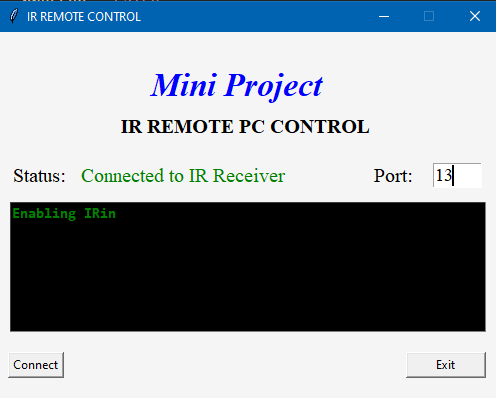


Fig 6.5 Application in connected state and enabling IR signal

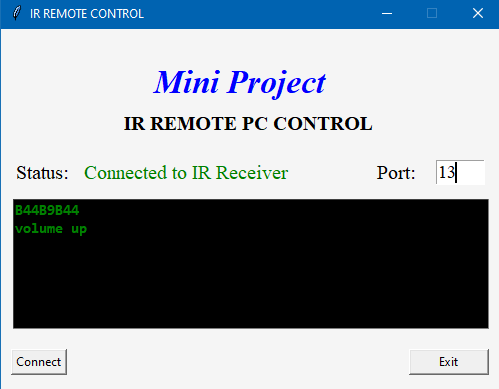


Fig 6.6 Application in connected state and showing the IR code of the volume button when pressed.

**6.2.4 USER ACCEPTANCE TESTING**

User Acceptance Testing (UAT), often known as beta or end-user evaluating, is the process of a user or client testing software to see whether it can be accepted. This is the last stage of testing after the functional, system, and regression tests have been completed. The major goal of this testing is to ensure that the software meets the company's needs. End-users who are acquainted with the business need to do this validation.

**6.2.5 TEST CASES**

A test case is a defined format for software testing required to check if a particular application/software is working or not. A test case consists of a certain set of conditions that need to be checked to test an application or software i.e. in more simple terms when conditions are checked it checks if the resultant output meets with the expected output or not. A test case consists of various parameters such as Id, condition, steps, input, expected result, result, status, and remarks. The various testcases for the propsed project is shown in Table 6.1.

**Table 6.2 Test cases for the application**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Case** | **Test Condition** | **Expected Output** | **Actual Output** | **Test Result** |
| 1 | Check if the application can connect to Arduino. | Arduino COM port no. must be entered and connect button should be clicked | Application should show status as connected. | Application showed status as connected. | Pass |
| 2 | Verify whether application responds to IR remote. | Status must be connected | Necessary events and actions should be triggered. | Necessary events and actions are triggered. | Pass |
| 3 | Verify whether application closes correctly. | Application must be opened. | Application should exit without errors. | Application exited without errors | Pass |
| 4 | Verify whether application triggers events. | Status must be connected | Necessary applications should open or events should occur whenever a button is pressed on remote. | Necessary application opens and events occur when a particular button is pressed for that event. | Pass |

**Chapter 7**

# RESULTS AND DISCUSSION

The objectives presented in Chapter 2 created a solid base for the project. We were able to control applications using IR remote. We were able to control the playback of songs like increasing or decreasing volume, play/pause the song and changing the songs as shown in Fig 7.1. We were able to open file explorer automatically by pressing button one on the remote. Also we were able to navigate through files and folders as shown in Fig 7.2. If we have more than one application opened we can switch between the applications as shown in Fig 7.3. We were able to open power point presentation automatically and moving through slides was easier through handheld TV remote as shown in Fig 7.4.



Fig 7.1 Controlling volume

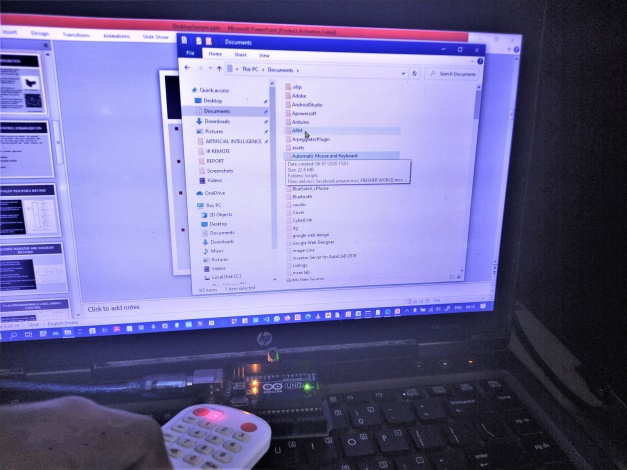


Fig 7.2 Navigating through files and folders



Fig 7.3 Switching between applications

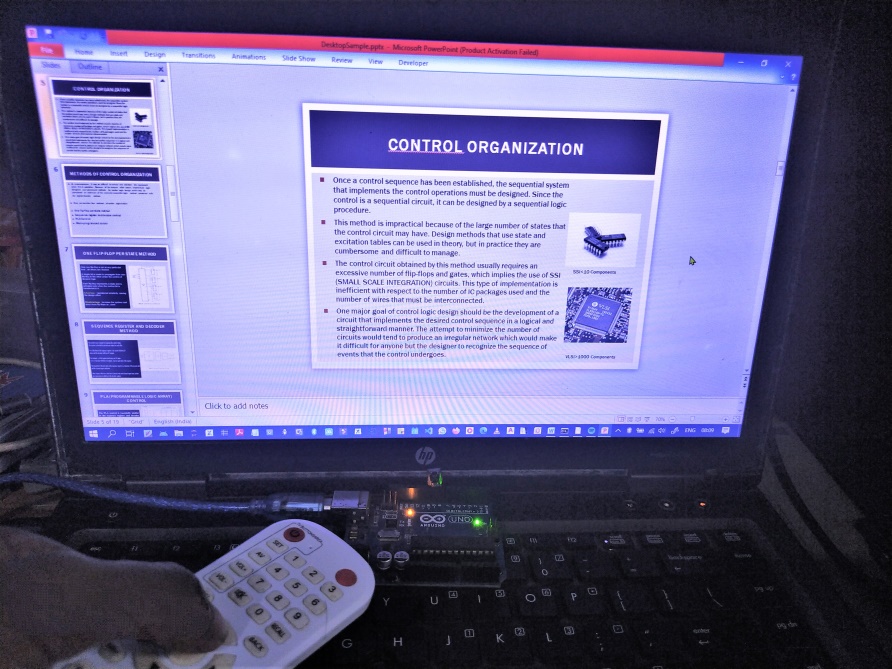


Fig 7.4 Controlling ppt

Examination of the code revealed that major time in processing the IR codes was consumed by the GUI application since it has to update the window constantly to display the necessary IR codes. Also due to environmental and other physical factors we observed that there were random IR codes generated and this increased latency when a button is pressed and when the particular event mapped to that button triggers. However, we were able to reduce the latency by making the time taken to update GUI to lesser values around zero to two. This reduced the latency and IR codes were interpreted and processed fast.

# APPLICATIONS

Some of the applications of IR Remote Control are:

1. Home Automation projects
2. Home security systems
3. Fire detection
4. Proximity sensors
5. Distance measurement
6. Infrared remote controlled toys

# 

# CONCLUSIONS AND FUTURE ENHANCEMENTS

It can be concluded that the project has met all the expectations that we had, and has been working successfully as intended. The application can perform many user actions like controlling audio/video and presentations, navigating to folders and files, opening and using favorite applications etc. And all of this can be done wirelessly via any IR remote from 10-20m distance easily giving a traditional feel of TV. Also this project can reuse old remotes that are not in use by decoding their codes and assigning user functions to these codes.

Future enhancements and improvements can be made to the project to make it even more flexible and more responsive. Cross platform support can be added to work on multiple operating systems effectively. New features can be added to the existing project where users can themselves map remote buttons to which ever keystrokes and shortcuts they need making it user friendly. Multi Remote support can be added where the application can support remotes from different vendors.

**REFERENCES**

[1] Samiran Maiti and Pabitra Kumar Nandi, “IR Remote Control Signal Decoder For Home Automation”, IJESIT, Vol3, Issue 3, 2014.

[2] Ukachi Osisiogu, “Utilization of Serial Communication in Arduino”, Research Gate, Awka, 2015.

[3] Leo Louis, “Working Principle of Arduino and Using it as a Tool for Study and Research”, International Journal of Control, Automation, Communication and Systems (IJCACS), Vol.1, No.2, 2016.

[4] Krishna Kattabiranam, “How to set up an IR remote and receiver on an Arduino”, Circuit Basisc.com, 2020.

[5] David Anmos, ”Python GUI Programming with Tkinter”, realpython.com, 2022.