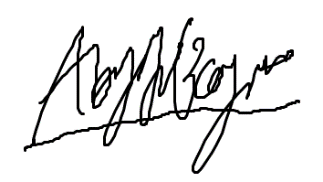


## **System programming and computer control**

Module Code: CT047-3-2-SPCC

Intake: APU2F2102CS(IS)

**Project Title: Smart Home Control System**

**Name: Abiyyu Taj Mahasin Bagindo**

**Signature:**

Table of Contents

**Introduction3**

**Hardware Requirements3**

**System Design**............................................................................................................................................ **4**

**Protocol Design**.......................................................................................................................................... **5**

**System Review**.......................................................................................................................................... **12**

**User Manual**............................................................................................................................................. **14**

**References**................................................................................................................................................ **22**

**Introduction**

Humans, unlike other creatures in the animal kingdom, have the sufficient level of intelligence to utilize sophisticated tools. The purpose of these tools are to solve problems or simply make their daily life easier. It started with the first spark of flames that allow humans to keep themselves warm and light their surroundings, and then the invention of the wheel revolutionized transportation. All of a sudden, heavy objects can be transported over great distances (Page, 2016). These tools evolve and new ones are invented as well. Nowadays, in the 21st century, we harness the power of electricity to fuel our daily lives. And of course, computers allow us humans to do things that cannot even be comprehended by the general population half a century ago. Now, we are in the information era, with computers embedded in various objects. In this project, we are embedding computers to home appliances. Allowing lights and doors to be manipulated through the press of a button on a panel.

A prototype of a smart home control system has been developed, with LabView being the software development tool of choice. LabView is a software developed by National Instruments, that allows the engineering of software within a virtualized environment. This means, a system can be developed with minimal consumption of time and materials. This removes various limitations that may hinder the development of the prototype.

**Hardware Requirements**

The proposed smart home control system uses the following hardware components which are needed for a physical implementation of the prototype.

* Raspberry Pi Computer System
* Smart Bulb
* Login Panel
* Smart TV
* Plugs
* Curtains Control
* Electric Doors
* Heat Sensor/Thermometer
* Air Conditioner
* Ambient Light Sensor

**System Design**

Below are flowcharts that show the flow of both the client side of the system and the server side of the system.

**System Start**

Diagram

Description automatically generated

Figure 1 Start of system.

**Client Module**

**Diagram

Description automatically generated**

Figure 2 Client module.

**Server Module**

Diagram

Description automatically generated

Figure 3 Server Module

**Protocol Design**

The system is comprised of two separate modules. A client module, and a server module. These two modules are connected using TCP/IP. TCP/IP is used as a method of communication between the two modules in the system because when compared with UDP, TCP/IP is simply more suitable and provide more advantages than UDP. TCP/IP has various advantages that UDP does not have such as larger capacity of packets that can be sent with around 20-60 bytes variable length header as opposed to UDP’s 8 bytes fixed-length header. TCP/IP is also more reliable due to it being able to guarantee the delivery of data while UDP cannot. The only advantage UDP has over TCP/IP is it is faster and more lightweight when compared to TCP/IP. However, this is an insignificant factor considering the whole system runs locally. Therefore, quality is favored more than speed which leads to TCP/IP being the winner and not UDP in this scenario (Jain, 2021).

**Client Module**

The client module’s task is to send packets containing data that will be used to interact with the system. The client module is essentially a control panel that allows the user to control the system. The following are the TCP/IP functions used in the client module and their purposes.

**TCP Open Connection**

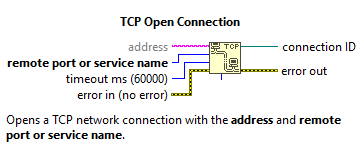


Figure 4 TCP Open Connection function context help.

The TCP Open Connection function allows the program to open a TCP network connection with a designated address and its specified port. In the prototype, a TCP network connection is opened to the localhost with port number 50000. The timeout mark can also be set, with 60000 milliseconds being the determined timeout mark.

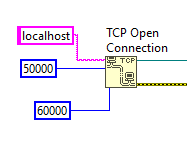


Figure 5 TCP Open Connection being used in the client module.

**TCP Write**

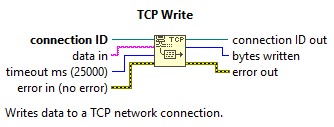


Figure 6 TCP Write function context help menu.

The TCP Write function allows packets to be sent through the TCP/IP network protocol. The function is connected with the previously mentioned TCP Open Connection function through the connection ID port, which takes in the connection ID that lets the system know where to send the packets to.

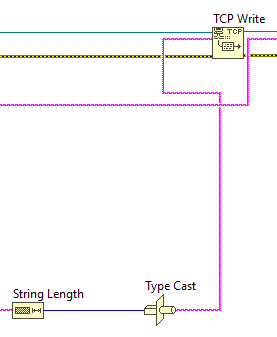


Figure 7 The TCP Write function being used to send a packet in the form of a string's length after being converted to a string type data from numeric type data.

**TCP Close Connection**

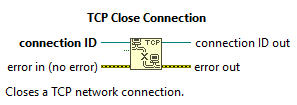


Figure 8 TCP Close Connection context help menu.

The TCP Close Connection function allows the system to close a previously opened connection for all open connections must be closed once it is done being used. The function simply takes in the connection ID of the connection that needs to be closed.

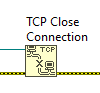


Figure 9 TCP Close Connection function being used in the client module to close a connection.

**Server Module**

The server module’s task is to receive packets from the client module and process it into a meaningful output. For example, if the client module sends a packet, the server module will decipher the packet and commits the appropriate routine according to the received packet. An analogy would be the user is a customer ordering food in a restaurant, and then the client module is a waiter taking orders, and then delivering the order to the chef which is the server module who will cook the food according to the order received.

**TCP Listen**

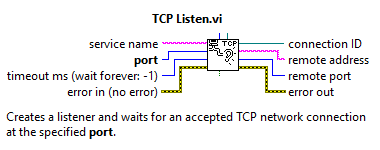


Figure 10 TCP Listen VI context help menu.

The TCP Listen VI is used to accept opened TCP connections at the specified port. The VI waits and listens at the specified port and receives packets sent through that port. This is how the server module receives the packets sent by the client module. Just like the TCP Open Connection function, the TCP Listen VI returns the connection ID of the open connection which can be passed to other TCP functions.

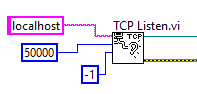


Figure 11 The TCP Listen VI being used to listen for open connections at port 50000.

**TCP Read**

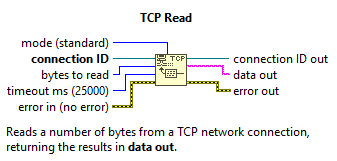


Figure 12 TCP Read function context help menu.

The TCP Read function reads packets that it receives through the opened TCP connection. In the server module, the TCP Read function receives its packets from the TCP Listen VI. It takes in connection ID which is returned by the TCP Listen VI. The function needs a specified number of bytes to be read from the received packet. Once the packet is read, the function returns it as a string that can be processed by other components in the system.

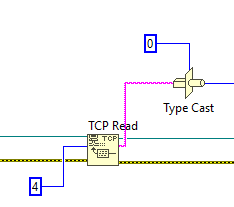


Figure 13 The TCP Read function reading a packet and returning string data which is then converted to numeric.

**TCP Close Connection**

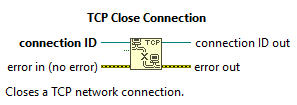


Figure 14 TCP Close Connection context help menu.

As mentioned previously, any opened connection has to be closed at once it is finished being used. So, just like the client module, the server module uses the TCP Close Connection function as well to close the communication between the two modules.

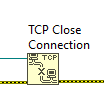


Figure 15 The TCP Close Connection function being used to close the connection in the server module.

**System Review**

A major limitation of the system is the fact that a plain text file is used to store login credentials. Which is a very significant security risk due to the login credentials being exposed without any protection. If a person with malicious intent would like to break into the system, all they have to do is open the account.txt text file which stores the login credentials of all users without any encryption or security. Another limitation of the system is it relies on the house electrical supply and does not have any backup power source. If there is a power outage, the system will simply stop working and will only be able to be activated again once the house’s power comes back.

To fix the security issue, an external tool called the LabView Encryption Compendium might be able to be implemented. This addon allows LabView to encrypt data using methods such as hashing or advanced encryption. For power problems, an emergency battery can be implemented to power vital components which allows the system to be active at a minimum level such as opening doors.

**User Manual**

**Login and Registration**

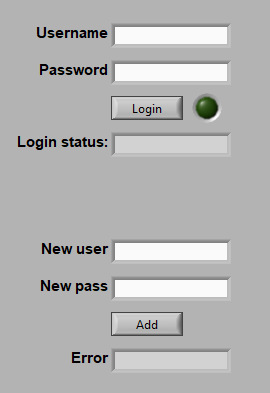


Figure 16 Login and registration page.

When starting the system, the user will be greeted with two sets of text fields. If the user is already registered, please input your credentials in the first two text fields and click on the Login button. Otherwise, please register by inputting the desired credentials at the last two text fields and click the Add button. The user can now Login.

**Client**

After a successful login, the user will see both the client and server module. This section will cover the Client module.

**Control**

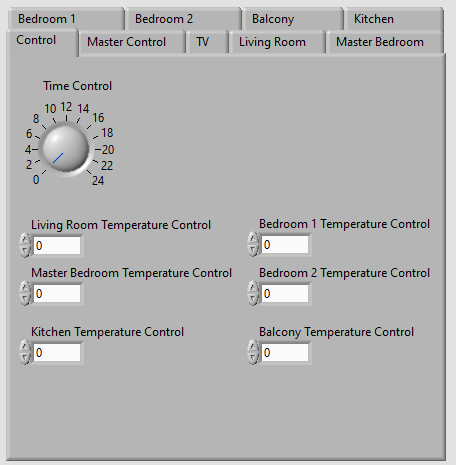


Figure 17 Client control tab.

The Control tab allows the user to modify the environment conditions that cannot be controlled in real life. The purpose of this is to test the system in various conditions without having to experience them. Turning the dial will change the time, the dial represents all 24 hours in a day. The numeric control below the dial allows the user to control the temperature in each room.

**Master Control**



Figure 18 Client Master Control tab.

The master control tab gives the user access to override the various functions of the system. For example turning on the three master buttons will disable the corresponding buttons. If a category of buttons have its master button turned on is turned on, an alarm of the room where the button is pressed will be triggered. For example, if the Light Master Button is on, and someone were to turn on the lights in the living room, an alarm in the living room will be triggered. This applies to doors and curtains as well. The six alarm buttons below the master buttons allow the user to disable the alarms if any are triggered or if they want to trigger the alarms themselves directly.

The two buttons at the bottom are extra features. Turning on the Automatic Curtain button will cause the curtains around the house to open in the morning and close in the evening automatically. While the Light Rotation button will cause the lights in the house to turn on and off in sequence, simulating a person’s presence in the house.

**TV**

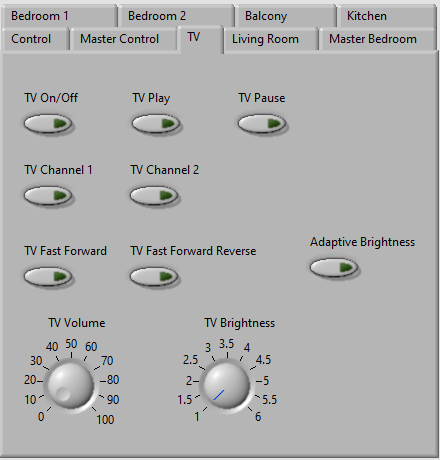


Figure 19 Client TV tab.

The TV tab, just like the name implies, allows the user to control the TV. The On/Off button toggles the visibility of the TV, the TV Play button plays the video on the TV, the TV Pause button pauses the video that is currently playing on the TV. The TV Channel 1 button sets the TV to channel 1 while the TV Channel 2 button sets the TV to channel 2. The TV Fast Forward and Fast Forward Reverse button allows the user the skip or go back through the currently playing video. The TV Volume dial controls the volume while the TV Brightness dial controls the TV brightness. The Adaptive Brightness button overrides the brightness controls and sets the TV brightness according to the room brightness.

**Room Controls**

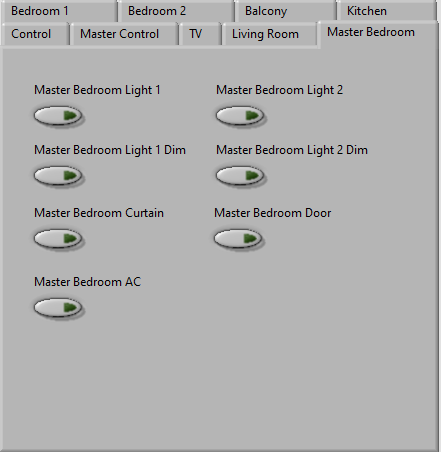


Figure 20 Client Room Controls tab.

There is a total of six rooms in the house. Each room has at least one light, one door, and some rooms have curtains. The control is the same for all lights, doors, and curtains so the following applies to all rooms. The Room Light button turns on or turns off the light. When the corresponding light’s dim button is turned on, the light turns dimmer and changes color to blue. However when the room temperature goes above 25 degrees Celsius, the lights change color to red. The Room Curtain and Room Door buttons opens or closes their respective curtain or door. The AC button turns on the AC which will lower room temperature. The AC turns on automatically when a room with AC in it reaches a temperature of 28 degrees Celsius.

**Server**

This section covers the server module of the system.

**Master Control**

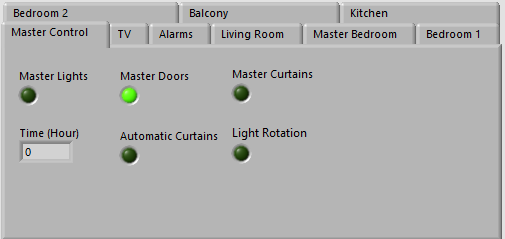


Figure 21 Server Master Control tab.

This tab shows the time of the day in hours, and the currently in effect master functions

**TV**



Figure 22 Server TV tab.

This tab shows the properties of the TV such as the volume, brightness, and features such as Adaptive Brightness.

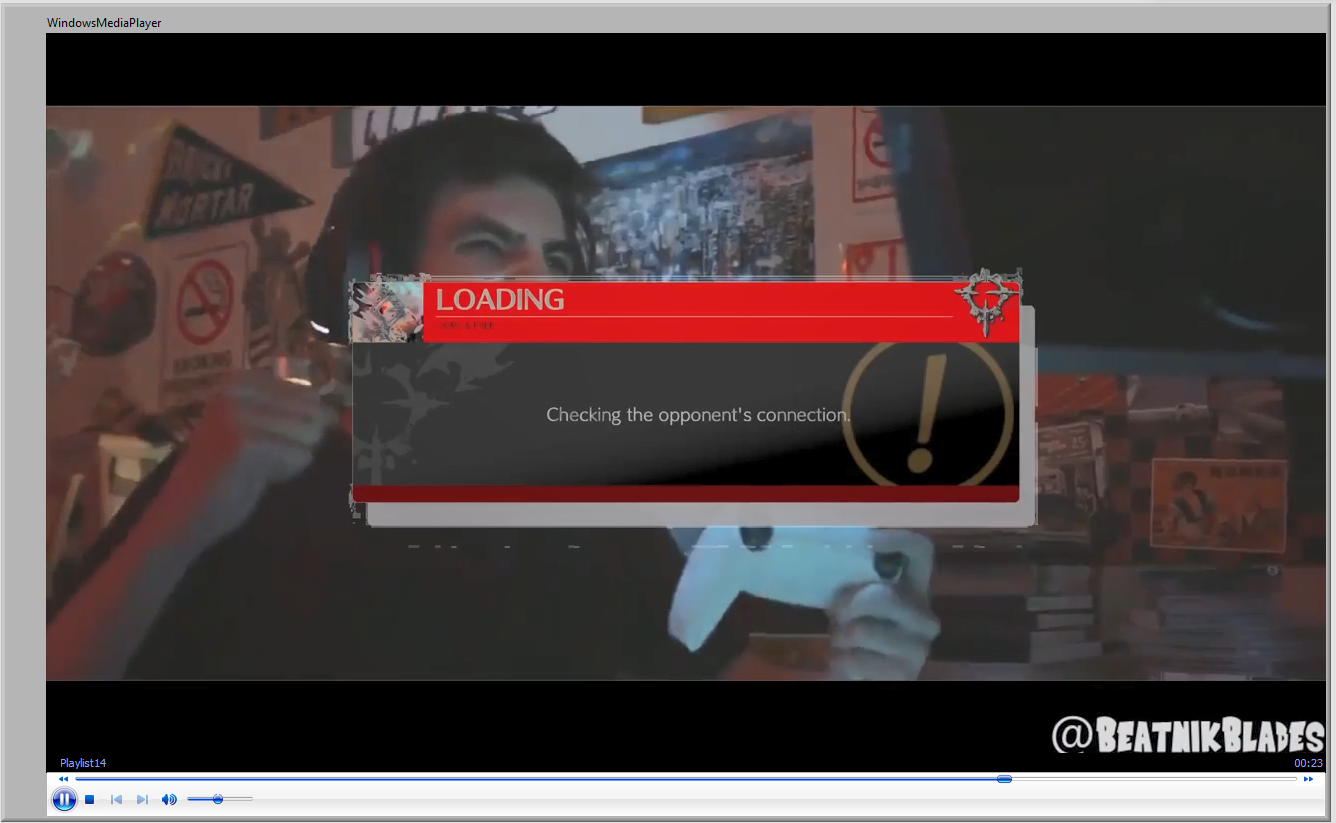


Figure 23 Server TV playing a video.

**Alarms**

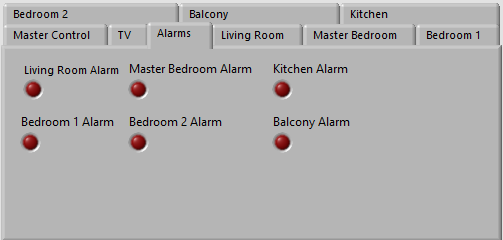


Figure 24 Server Alarms tab.

This tab shows all the alarms in the house. If the LED turns on, it means the alarm has been triggered.

**Room**

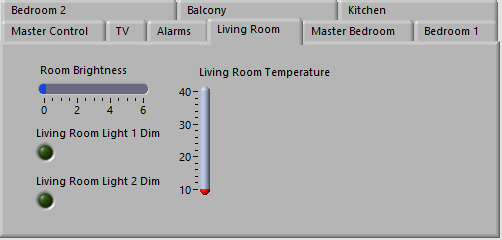


Figure 25 Server Room tab.

All six rooms tabs in the house typically shows whether the light is dimmed or not, and the temperature of the room. The living room has a Room Brightness bar because it has a TV which uses the Room Brightness level for its Adaptive Brightness feature.

**References**

Jain, P., 2021. *Differences between TCP and UDP - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/> [Accessed 13 June 2021].

Page, H., 2016. *Our greatest invention the wheel will become fundamental to the transport industry and the world we live in*. [online] The Sun. Available at: <https://www.thesun.co.uk/news/hold-ye-front-page/1909443/wheels-our-greatest-invention-mans-greatest-invention-will-become-fundamental-to-transport-and-later-to-agriculture-industry-and-the-world-in-which-we-live-today/> [Accessed 13 June 2021].