REMOTE POWER USING ANDROID, POE AND VPN

Muhammad Amir Rozlan Bin Mohd Zamri
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
Kuala Lumpur Malaysia
amirrozlan007@gmail.com

Sri Banu A/P Munisamy
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
Kuala Lumpur Malaysia
sribanu@unikl.edu.my

***Abstract*— This final year project develops a simplified device that will replace the old tools to perform remote power. This project will help the worker to control the power of the device remotely from outside of the working area by using a VPN. The status of the power such as voltage, ampere, and watt provided by the controller is shown in an Android application. This project uses System Development Life Cycle (SDLC) model which is a prototype model as focusing on building, test, and then reworked when needed until an acceptable prototype is achieved. The result of the project should be the android application can perform remote power and notify the user of power failure.**

Keywords—VPN; remote power; Android application; Controller

I. INTRODUCTION

A remote control is an electronic device used to operate another device from a distance usually wirelessly. In consumer electronics, for example, a remote control may be used to operate devices from a short distance, such as televisions, DVD players or other home appliances. A remote control is primarily a user convenience feature, and can enable devices that are out of convenient reach for direct control operation to work. In certain situations, remote controls allow a person to manipulate a device that they would otherwise not be able to access, such as when a garage door opener is activated from the outside or when a high ceiling mounted Digital Light Processing projector is operated from the floor level by a person.

Consumer infrared devices that transmit digitally-coded infrared radiation pulses to monitor functions such as power, volume, channels, replay, track change, heat, fan speed, or other device-to-device features are usually present-day remote controls. Tiny wireless handheld objects with a variety of buttons are typically remote controls for adjusting various settings for these devices.

The remote control provides all the feature controls for several applications, while the managed computer itself has just a handful of critical primary controls. The purpose is same as remote control using Bluetooth and Wireless Local Area Network (WLAN) to operate another device from a distance but in latest technology people can control and can monitor the devices at the same time by using smartphone or any displayed devices.

This project suggests on making the remote power control to operate another device from a distance through Virtual Private Network (VPN) and internet. The remote part is using android application that provide controlling and monitoring system. The controller is a router-switch, will be performed as network gateway and power supplier to the device. The power provided to the device is Power over Ethernet (PoE) which is the same power that have been use nowadays to power the Access Point (AP).

II. RELATED PROJECT

A. Microtubine Power Controller

The Power Controller is a study performed by Mark Gilbreth [1]. In this research, a device was designed to control the turbine power motor's delivery power, and the turbine can work in heating and cooling cogeneration systems at high speed and low temperatures.

The current invention provides a power controller that provides a distributed power generation networking system in which a common DC bus is used with bi-directional power converters to allow compatibility between different components of energy. Each power converter functions essentially as a specialised bi-directional switching converter designed to provide, under the control of the power controller, an interface to the DC bus for a particular energy component.

B. Power Monitoring module with display unit for electrical power source device

The Power Monitoring Module with an electrical power source display unit is a research conducted by Daniel Liu (2000)[2]. In this study, a computer was designed to track the power source device and, more precisely, an electrical monitoring module with a display unit capable of showing various electrical parameters of the electrical appliance plugged into the power source device.

The other purpose of the present invention is to provide a display unit for the power monitoring module capable of showing the associated electrical Parameters of the power source system and the electrical appliances connected to the power source device. The electrical parameters include voltage value, current value, watts, kilowatt-hours, apparent power value, power factor, frequency, and so on.

C. Monitoring and Controlling Power using Zigbee Communications

Monitoring and Control Power using Zigbee Communications is a research conducted by Javaid et al. (2012)[3]. The main aim of this research is to create a home monitoring system that can regulate power by using the Zigbee module. In this project, a device consisting of an electrical grid was built where generation, transmission, distribution and customers are not only electrically linked, but also through a strong communication network with each other as well as with the sector, operation and service provider. It is very important to find an acceptable protocol to achieve a good communication connection between them.

This paper discusses different hardware techniques for home and transmission side power monitoring, power management and remote power control, and also discusses Zigbee's suitability for the required communication link. Zigbee plays a key role in monitoring and direct load control for successful power usage. This covers enough space required for connectivity and runs at a low data rate of 20Kbps to 250Kbps with limited power consumption. This paper describes the user-friendly control of home appliances, on / off power over the internet, PDA via Graphical User Interface (GUI) and GSM cellular mobile phones.

D. Networkable Power Controller

Networkable power control is a study conducted in March 1999 by Adamson (1999) of Power Circuit Innovations Inc. [4]. The main purposes of this research are to develop and deploy a Networkable Power Controller with a monitoring system.

In general, a building control system enables a building operator to control a building system inside one or more buildings, such as a heating,

Ventilation and air conditioning system (HVAC) system, a lighting system, a water and waste system, or a protection system. For example, a building control system can include a centralised or remote building control station from which thermostat setting schedules can be programmed by a building operator to track temperatures in different building zones. In this way, during different hours of the day, a building operator can control energy consumption and occupant comfort in conjunction with the planned building usage.

Furthermore, BACnet, the open system standard for building control system networks, has become an important standard in the construction control industry. For building automation and control networks, BACnet is a data transmission protocol. A building operator can manage and track building- related devices spread in a building through a network by using BACnet. Such BACnet-compliant devices which include, but are not limited to, furnaces, air-conditioning systems, cooling towers, heat exchangers, lighting systems, dampers, actuators , sensors, security cameras and other devices relating to construction.

E. Smart Home Monitoring using Android and Wireless Sensors

Smart Home Monitoring using Android and Wireless Sensors is a research conducted by Sankaranarayanan et al. (2013) from Institute Teknologi Bruniei Jalan Tunku Link, Gadong Brunei Darussalam[5]. In this research, their main purposes are to create a system for monitoring the electrical power using android and wireless sensor network (WSN).

In this project, the "well-being" of power delivery in the home is optimally controlled by the use of the wireless sensor network (WSN) in home monitoring to mitigate risks caused by electrical faults. Incidentally, this approach helps to monitor the tariff by optimally maintaining the power use. It has also been found that not much research has yet been conducted on the use of WSN for electrical power control and the consequent risks in order to ensure the safety of residents at home.

III. PROPOSED METHOD

SDLC is known as Software Development Life Cycle where it is a framework defining tasks performed at each step in the software development process. There few types of SDLC model that can be used. In this project, Prototyping model is used because it is an iterative, trial, and error method. In this model, projects can be developed successfully, and it has rapid processes to be achieved.

Prototype methodology in figure 1 is shown characterized as a model of software development in which a prototype is constructed, tested, and then reworked until an appropriate prototype is achieved if necessary. It also provides a foundation for the final system to be created.

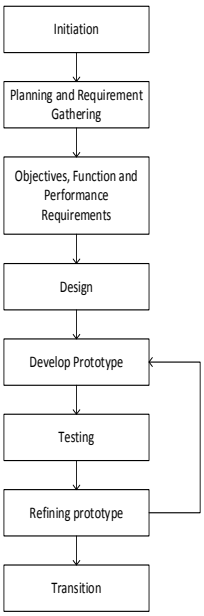

Figure 1: A Prototype Model

A. Analysis Phase

This project using qualitative research which is concerned with understanding human behavior from the informant's perspective. The practical problem is an undesirable situation in real world. The data collection method is using existing data which is need to review and analyze the previous project that related to the power control.

B. Design Phase

In this section, figure 2 is shown the topology of the Power controller using POE, VPN, and Android in a diagram form. Mikrotik RB750UP will be used as a router, controller, VPN, and PoE provider for this project. Android users will be connected to the private network by using a VPN. The android box will be power up using power from PoE. The results were sent through the controller where the results will be shown to an android user.

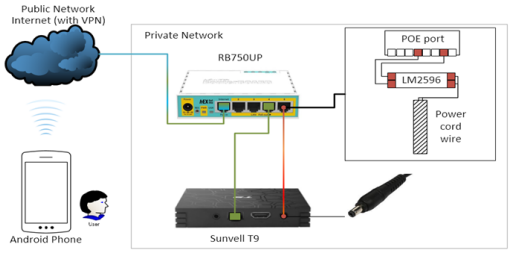

Figure 2: Prototype Design

C. Hardware Development

Hardware development need to modify the cable and convert the power into acceptable voltage. The latest converter makes the cable no need to be soldered, except just attached the cable on board. In figure 3.6 showing the pin functions and polarity of UTP cable mode B. The pin at switch 4, 5, 7 and 8 will be used in this project.

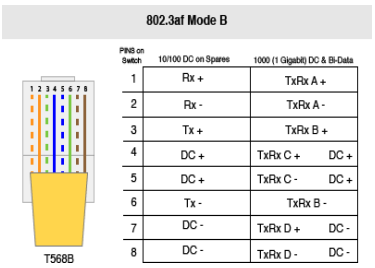

Figure 3: T568B pin function

The power transmitter pins are connected together in figure 4, followed by power polarity. The positive power transmitted in blue and white-blue and negative transmitted in chocolate and white-chocolate.

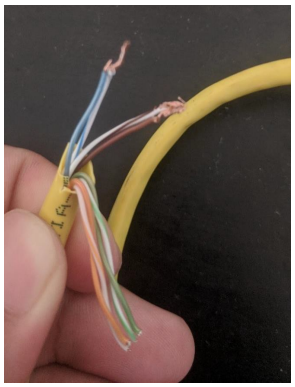

Figure 4: Power transmitter pins

In figure 3.8, both transmitted cables are attached to the board together at the input side of converter board.

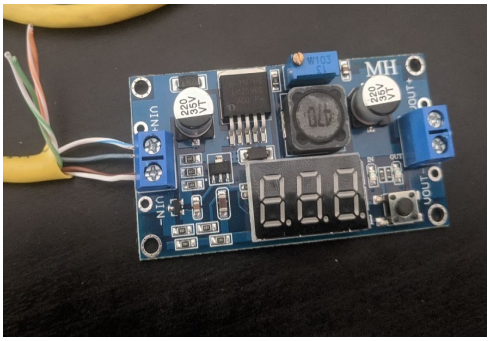

Figure 5: Converter with input attached

After done with UTP cable, at the power adapter part make an insulated wire by terminate the cable from the power adapter as shown in figure 6. Check the polarity which positive and negative can be detected by using a Multimeter or observed the colour of the cable.

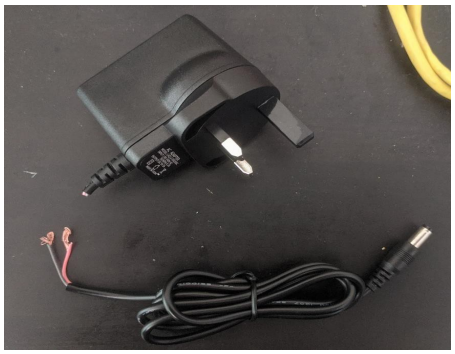

Figure 6: Power adapter cable

By understanding the polarity of the cable, the cable now can be attached to the output side at the converter board as shown in figure 3.10.

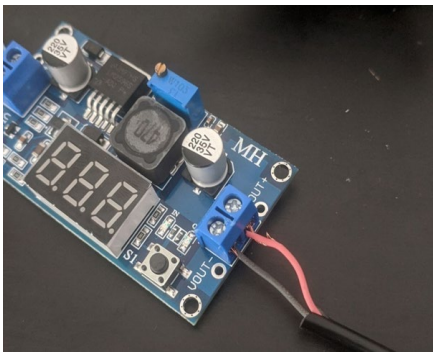

Figure 7: Converter with output attached

Now, connect the UTP cable to the port at the controller as shown in figure 3.11.

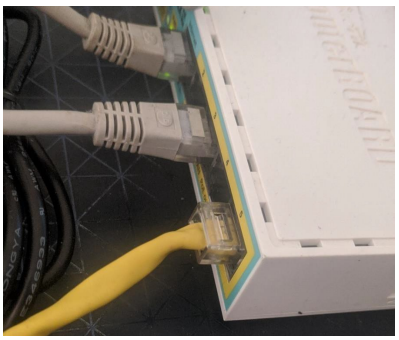

Figure 8: UTP cable connected to the controller

Check the power of the converter by turned on the PoE at the attached port. In figure 9 shown how to turn on the PoE port by using software Winbox. Change the configuration by select auto on at the PoE Out located in "interfaces>interface>PoE".

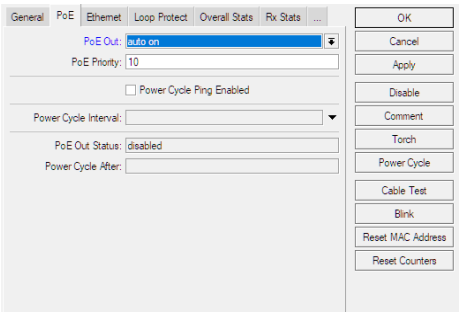

Figure 9: Turn on PoE using Winbox

When the PoE is on, the converter will detect the power that delivers by the router board. Next, change the voltage output before deliver to the android box by using a flat head screwdriver as shown in figure 10. The android box will detect the power delivered by the converter (convert to 5 volt) and the android box is turned on.

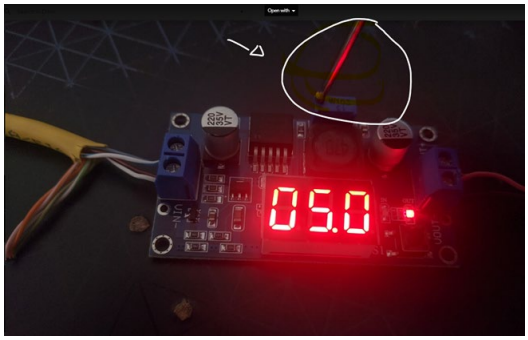

Figure 10: Change the power output

D. Software Development

VPN Configuration

The VPN extends a private network through a public network and gives the user able to send and receive data across public networks as for virtual network is directly connected to the private network like a tunnel. The purpose of a VPN in this project is to make the application is able to connect to the controller across a public network.

The type of VPN using on this project is PPTP (Point-To-Point Tunneling Protocol) allows to connect securely from a remote location (such as home) to an LAN (Local Area Network) located in another location, such as workplace, business office, etc.

The VPN configuration such as user, profile, and IP pool is on the controller, so other than the power delivery, the VPN server is also handled by the controller. In figure 11, create an IP pool to assign an address to the remote user by building the remote address IP pool located in "IP > IP Pool > pool > +".

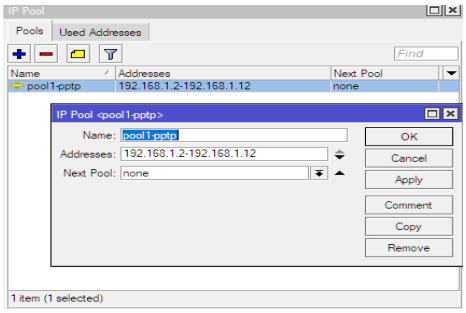

Figure 11 IP Pool for remote address

After that, the figure 12 is shown masquerade the VPN network to allow access from the outside network (Internet) located on "IP > Firewall > NAT > + > General > (Src address)" and "Action > masquerade". IP masquerading is a technique that hides an entire IP address space, usually consisting of private IP addresses, behind a single IP address in another, usually public address space.

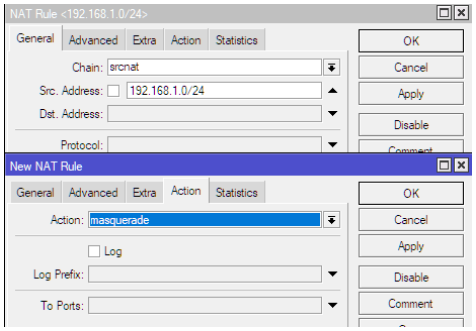

Figure 13: Masquerade the network

Then, set remote profile for PPTP on "PPP > profile" as shown in figure 14. Set local address as the IP gateway of IP Pool. Then select IP Pool profile at remote address.

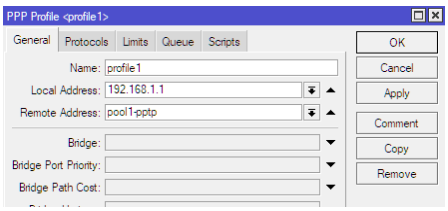

Figure 14: PPP Profile

Next, create user profile on "PPP > secret" as shown in figure 15, set the name and password for confidential and secure connection. Select the type of service as PPTP and PPP profile at profile.

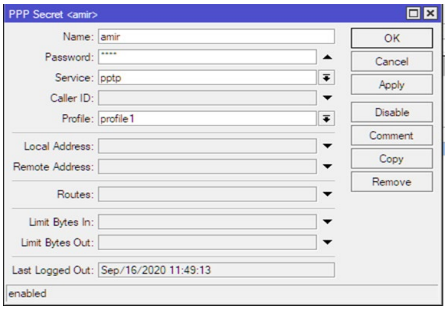

Figure 15: PPP Secret

Finally, enable the PPTP service on "PPP > Interface > PPTP Server" and select PPP profile as "Default Profile' as shown in figure 16.

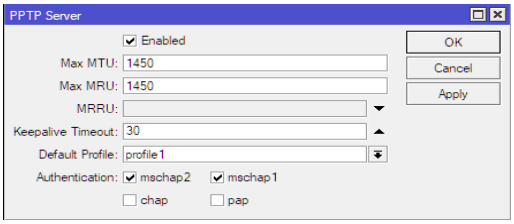

Figure 16: PPTP server

Since the PPTP port is blocked by the gateway, figure 17 shown the configuration of the port forwarding to allow the link from outside of the network to the VPN port.

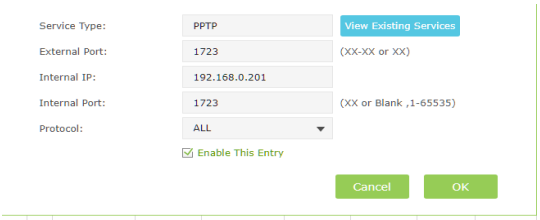

Figure 17: Port forwarding for PPTP

Android Studio

Develop application for this project is using Android Studio [2]. The program language that been use on Android Studio are Java language and Kotlin language. The Kotlin language is a lot easier than the java language but the Mikrotik API only provided for Java language. The availability of using java is highly recommended because it has many references on the internet compared to Kotlin.

The java development can be divide by two, first frontend and second backend. In Android studio, the frontend is called interface and backend is called activity. So in this project have three interface, three activity and one library (contain four java).

Figure 18 is showing the java file contain inside the library directory provided by Mikrotik API.

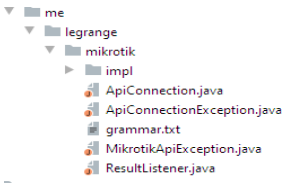

Figure 18: Java Library Directory

Figure 19 is showing the java file contain inside the activity directory called com.example. The process of a program that runs from the background while using the application.

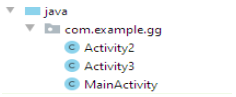

Figure 19: Java activity

Figure 20 is showing the interface file for the user interface application, this interface is what the user will see on their android phone.

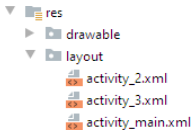

Figure 20: Interfaces

The priority part is to make the connection between the application and the controller. The link can create many ways like Telnet, SSH, and API. The API has been selected for use in this project because Mikrotik already provides the API for Java so the connection between the android and the controller is easier. By the way, the java application is not the same as java on Android Studio that has a different implementation. Figure 3.26 showing the code in java API connection that used to establish a connection between android application and controller.

```
Log.d(LOG_TAG, msg: "start");

ApiConnection con = ApiConnection.connect("192.168.0.201");

Log.d(LOG_TAG, msg: "start2");
con.login( username: "admin", password: "admin");
```

Figure 21: API Connection

The API command is different than the usual normal command for Mikrotik devices. This API command contains parameter in the command. Figure 22 showing the java code inside con.execute(from the library) to send the API command to the controller.

```
if(choice == 1) {
    con.execute( cmd: "/interface/ethernet/poe/set .id=ether4 poe-out=auto-on");

}else if(choice == 2) {
    con.execute( cmd: "/interface/ethernet/poe/set .id=ether4 poe-out=off");
}else if(choice == 4){
```

Figure 22: Execute command

The output from the controller is delivered by using the Array List and Map methods, the output result is separate and arranges by the separate array as shown in figure 23.

```
List<Map<String, String>> results =  con.execute( cmd: "/interface/ethernet/poe/monitor .id=ether4 once");
String list[];
String list1;
String list2;
String list3;
String list4;
String list5;
String list6;


for(Map<String, String> res : results)
{
    Log.d(LOG_TAG, res.toString());
    TextView tv = (TextView) findViewById(R.id.textView);

    list = res.toString().split( regex: ",");
    list1 = list[1].replace( target: "data={poe-out-status=", replacement: "").replace( target: "data={poe-out-pow
    list2 = list[2].replace( target: "name=ether", replacement: "");
    list3 = list[3].replace( target: "poe-out=", replacement: "").replace( target: "}", replacement: "");


    tv.setText("Port : " + list2 + "\nPort Status : " + list1 + "\nPoE Status : " + list3 );
```

Figure 23: Program output

Also due on gateway, the port for Mikrotik API is blocked to travel outside the network so need to create a port forwarding on the gateway as shown in figure 24.

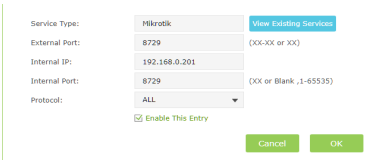

Figure 24: Port forwarding for API

After the coding is developed, the software can be run in two ways by using Android Emulator and a Physical device. Android Emulator (Android Studio, n.d.) provided by the android studio is Android Virtual Device, AVD is a configuration that defines the characteristics of an Android phone, tablet, Wear OS, Android TV, or Automotive OS device to simulate in the Android Emulator. The AVD Manager is an interface that can launch from Android Studio to helps create and manage AVDs. Figure 25 shown how to open the AVD setup and configuration.

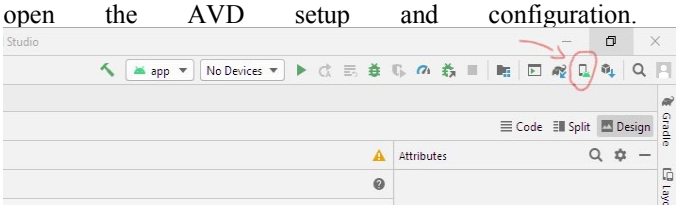

Figure 25: AVD option

Select the system image as an operating system for the device that has been created. Figure 26 shown the android version available for the system image is from android 7.0 until android 10.0 also the image can be imported outside of the android studio.

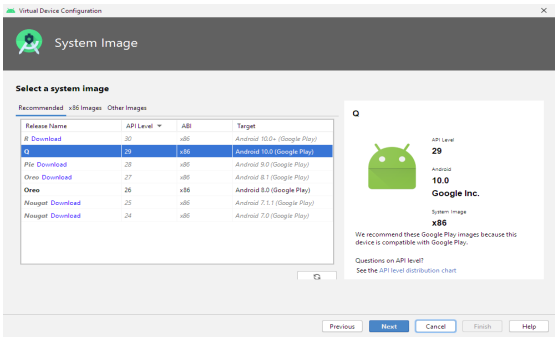

Figure 26: AVD Configuration

The other option to run the application is using Android Emulator from outside the Android Studio like LD Player, Nox or Game Loop. Figure 3.37 shows the Android Studio detect the external emulator other than AVD device.

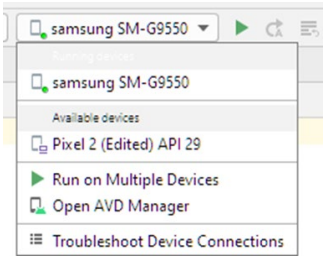

Figure 27: External emulator

The second way is using a physical device, the device needs to enable USB debugging inside "Setting > Developer option" as shown in figure 28. The different types of phones require different ways to enable developer functions. It is because the stock ROMs for android OS is being modified by the different manufacturer to create their trademark and unique for their devices.

Figure 28: USB debugging

Figure 29 shown the Android Studio detects the physical device and ready to run the program directly into the device. The program will install the application through a USB connection and run the application automatically.

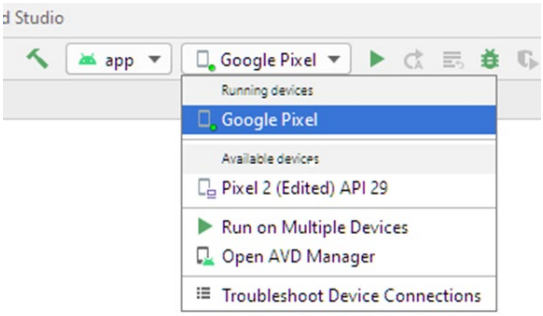

Figure 29: Physical device

IV. TESTING & DISCUSSION

In this section, the power delivery from the controller into an android box can be tested using the software winbox. When the power is turned on the android box should start without any power issues. In Figure 30, the android box is turned on without any issues and the led on the screen is shown.

Figure 30: Android box is turned on

The application shown the information on the certain PoE port for this project. The information is shown in three different ways. First, status disconnected, status off, and status on.

A. Status Application

Status Disconnected

Figure 31 shows the status "Disconnected" appeared when the application failed to connect to the controller. The problem can be the connection itself inside the local network or the outside network like VPN misconfiguration. Also, the issue can occur when the port of the controller did not forward by the gateway so the connection is blocked by the gateway.

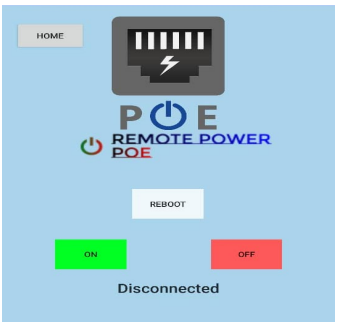

Figure 31: Status disconnected

Status Off

Figure 32 shows the status "Off" appeared when the connection is established and the PoE port is turned off. When the application is connected to the controller, the application will seek the status of the controller is it the PoE is on or off and show that status inside the application. Status off is the PoE function is disabled and the power is not provided by the controller.

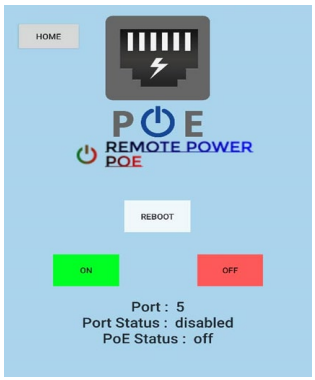

Figure 32: Status off

Status On

Figure 33 shows the status "On" appeared when the connection is established and the PoE port is turned on. That means the controller is provided the power at the PoE port. This status also provides additional status which is the detailed information of the power itself. The additional status is power ampere, voltage, and watt.

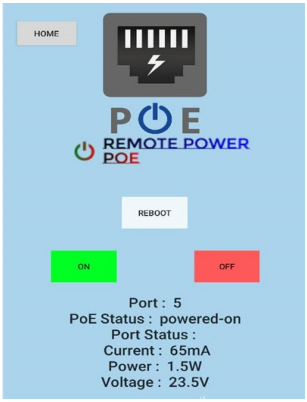

Figure 33: Status on

B. *Remote Power*

Figure 34 shows the touch button in this application has three buttons are Reboot, On, and Off. All of this button are worked fine and the controller accepted the command delivered by the application. The ON button is to make the PoE port to turn on and tell the controller to deliver power. The OFF button is to tell the controller to stop deliver the power. The REBOOT button is to send a command to the controller to restart the PoE port. The reboot takes 12 seconds to make the android box start completely and ready to be used.

Figure 34: Application buttons

C. *Notification Alert*

The Notification alert is to notify the user when the power failure event is happening so the user can take action on the spot to recover the failure device. The notification will triggered on the power failure at the controller (VIN) or at the android box (VOUT).

The notification at the VIN side means the power failure happened on the controller as shown in figure 35. The guess is going to be the PoE port or the system itself.

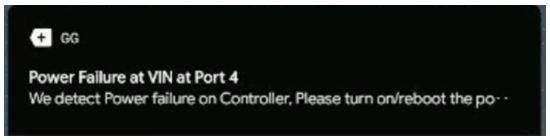

Figure 35: Notification of power failure at the Controller

The notification at the VOUT side means the power failure happened on the Android box device as shown in figure 36. The guess is going to be cable disconnected, device hangs or short circuit.

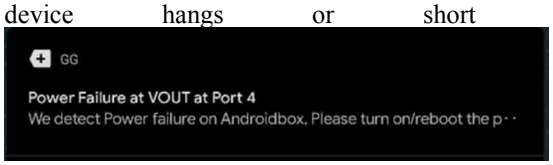

Figure 36: Notification of power failure at the Android box

D. *VPN Connection*

The VPN connection is tested by using the default Android VPN as shown in figure 37. The server address is going to be the public address of the network. The information on the configuration of PPP and the service type is filled in the detail inside the VPN profile.

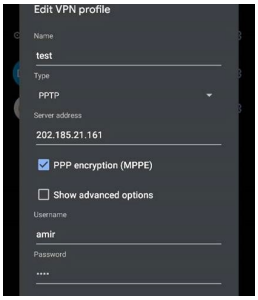

Figure 37: VPN Client Profile

Figure 38 shows the connection is established and the status of connection is recorded.

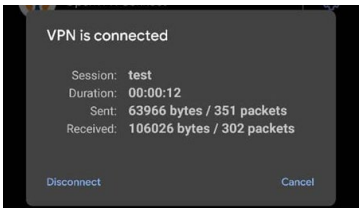

Figure 38: VPN connection status

Figure 39 shows the connection of the user to the server using PPTP in the controller. Now, The user is able to

connect the controller from outside of the network example from home.

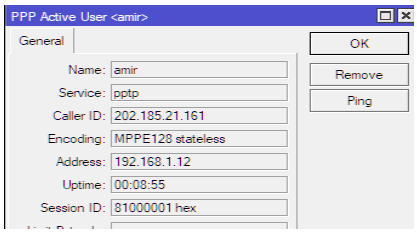

Figure 39: PPP Active User

Analysis

The VPN connection is important when the user is connecting the controller from outside of the local network because the VPN gives access from the remote to the local network across the public network. While the user is using the local network the VPN is no required to use because the connection to the controller is direct. Here the performance test between controller and application.

The performance connection of the local network which is the best because the connection is direct. Even though the connection of a VPN is not good as a local network but the main purpose of the VPN is mobility. The performance connection of the VPN network is affected by the Internet performance itself. So the hypothesis makes for this test is, if the internet connection is good, then the VPN will be stable.

Table 1 Comparison performance of the application

VPN network (U mobile)	Local Network
The speed for download is 3.90mbps and upload 1.56mbps.	The speed for download is 205mbps and upload 160mbps.
Average ping is 127ms	Average ping is 7.39ms
Mobility	Immobility
The connection affected by Internet speed	The connection affected by Network speed
Delayed on touch	Normal on touch

The power comparison is made by using two different PoE ports, so it can observe the power information by open both PoE ports then shut either one port. Figure 40 shows the both PoE port is turned on.

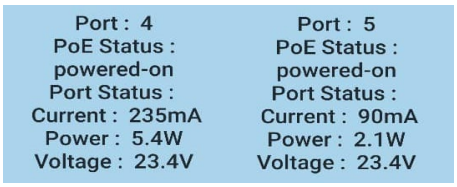

Figure 40: PoE port 4 and 5 is on

Figure 41 shows only one PoE port is turned on.

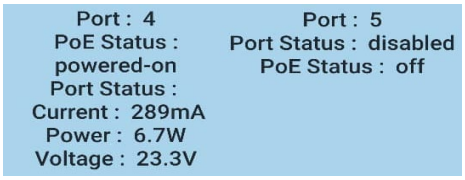

Figure 41: Only Poe port 4 is on

From the observation, the only change that can see is the current drop by 54mA, and the voltage increased by 0.1V.

This is proved the modification that has been made on PoE cable, the controller not have any issue and works fine as usual. The changes that have been made on the controller are acceptable and not harmful.

V. CONCLUSION

As a conclusion, this prototype of "Remote Power Using Android, PoE and VPN" is successfully developed and tested. It is proved that using Android application as a remote power of the device is a great help to the user.

As the result, for this project, right now the great platform to implement in the network system in order to control the power of the devices by using network devices as a controller. The controller can be replaced with a PoE switch for more features in providing the power to the devices also can terminate from using the converter.

Few recommendations for anyone that want to do this project in the future, need to develop a good Remote Power Controller that has other more features such as more accessibility for another network equipment brand. Next, using more futuristic network devices that can control the power of voltage that supply to the devices. Other than that, the VPN of the connection can be implemented inside the application for users to less difficulty in use.

ACKNOWLEGMENT

I would also want to express my gratitude to my supervisor Ms Sri Banu A/P Munisamy because of his guidance, crucial helps, encouragement and patience in every aspect until the end of the project. His countless comments, criticisms and recommendations during the preparation of this project are highly appreciated. Not forgetting to all the lecturers of UniKL MIIT who helped and guided me in the development of this project.

In addition, I also would like to thank my friends and the people who directly or indirectly helped me in completing Remote Power Using Android, POE and VPN project.

Motivations, moral supports and their encouragement help me to complete this project. Thank you very much.

REFERENCES

[1] "US6487096B1 - Power controller - Google Patents." [Online]. Available: https://patents.google.com/patent/US6487096B1/en. [Accessed: 15-Jan-2021].

[2] "Power monitoring module with display unit for electrical power source device," Aug. 2000.

[3] N. Javaid, A. Sharif, A. Mahmood, S. Ahmed, U. Qasim, and Z. A. Khan, "Monitoring and controlling power using zigbee communications," in *Proceedings - 2012 7th International Conference on Broadband, Wireless Computing, Communication and Applications, BWCCA 2012*, 2012, pp. 608–613, doi: 10.1109/BWCCA.2012.107.

[4] "Networkable power controller," Mar. 2000.

[5] S. Sankaranarayanan, A. Thien Wan, and A. Harayani Pusa, "Smart Home Monitoring using Android and Wireless Sensors," *Int. J. Eng. Manuf.*, vol. 4, no. 2, pp. 12–30, Aug. 2014,

[6] "Android Studio and SDK tools :   Android Developers," Android Developers. [Online]. Available: https://developer.android.com/studio. [Accessed: 15-Jan-2021].