Moxa Terminal Server Toolbox

[1]Ashok Kafle [2]Ernesto Paiser

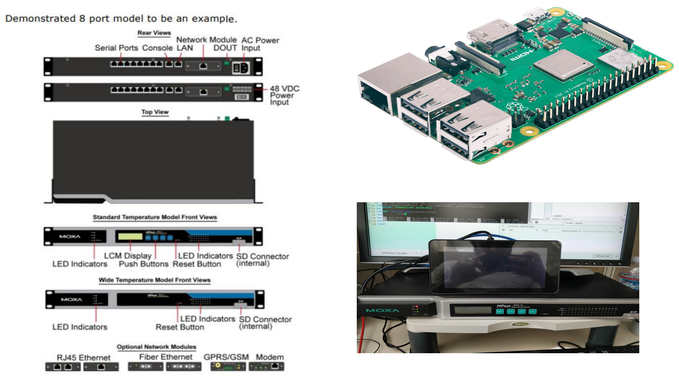
[1]Department of Electrical Engineering and Computer Science, Howard University, 2400 Sixth St., NW, Washington, D.C., 20001, USA
email: ashokkafle96@gmail.com

[2]Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.

email: paiser@slac.stanford.edu

Introduction

The NPort 6650 Moxa Terminal Server provides serial communication through Ethernet network and supports different operation modes and a powerful function of data buffering in case of a communication failure. The data is stored on the server and once the communication resumes, the buffered data is sent to the destination. This device supports 16 ports for high density environments.



Initial Approach

As the part of this project, this device along with Raspberry Pi 3B+ are used. The minimum goal is to to set the IP configuration on this device to DHCP configuration and create three users with different permissions. It is expected to do these tasks in Python Script.

First I received all the required materials:

i) A SD card with 64GB storage

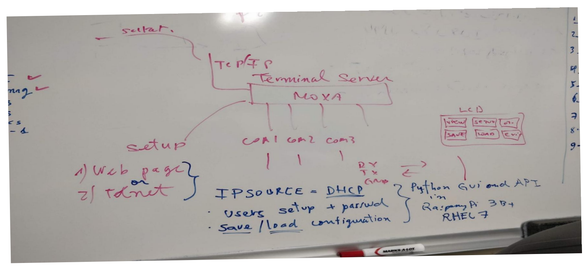
ii) SD Card Reader

iii)Raspberry Pi 3B+

iv) A Linux machine

v) A monitor and a HDMI Cable

In order to have a more clear view of the project, I have posted the following diagram by my mentor/supervisor Mr Ernesto Paiser when we walked our thought through the project timeline and steps.



So, the first step on the project was to make connections between the Raspberry Pi and the server. For that an operating system should be installed in the new Raspberry Pi. Centos 7 image for arm 32 processor was first downloaded on the machine and decompressed. Then the SD card is inserted into one of the machine and the image is copied to the SD card. To execute all these steps, one is expected to have at least some knowledge on Linux system administration. Also, various tutorial are available online which can be followed and the Raspberry Pi would get the operating system.

Loading factory defaults on Moxa Terminal Server and Installing python3 on Raspberry Pi:

- The reset button that is shown in the diagram above in introduction if pressed continuously for 5 sec with a pointed object until the Ready LED stops blinking will reset the server to factory defaults.

- After that Python3 needs to be installed.

Problem Encountered

Setting Up Moxa Terminal:

Moxa terminal can basically be set up by two methods:

i) WebPage

ii) Telnet

For my project, i decided to use Telnet. The default IP configuration of the server is 192.168.127.254 and in order to connect both the pi and server through telnet, they both must be on the same subnet .i.e The IP address of the Pi must be set to 192.168.127.XXX. I was unable to do so executing any sort of commands or referring to any materials online.

Refactored Approach

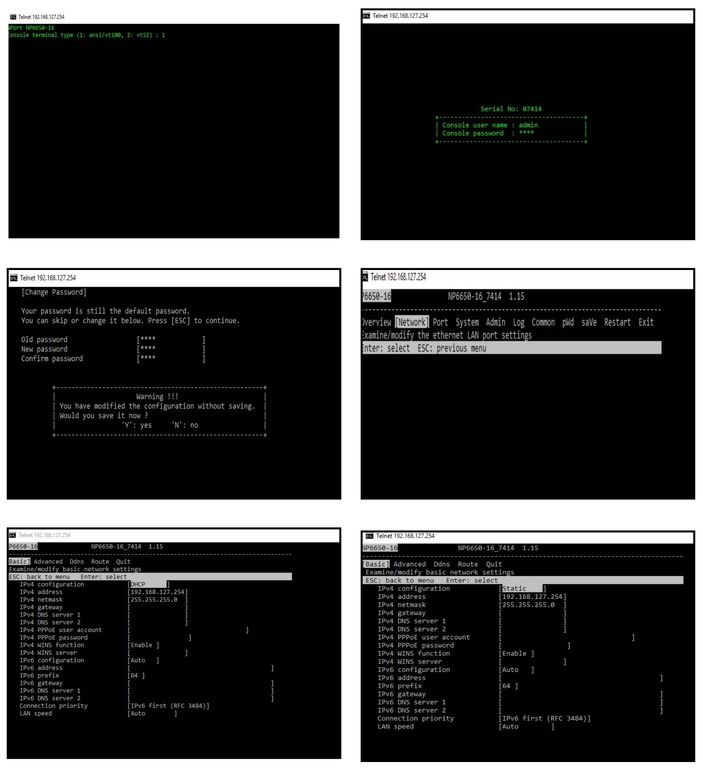
The Centos 7 GUI utilizing too much of CPU, lightdm for Centos not executing well and not able to set both on same subnet led to install Raspbian on the Pi. Raspbian is installed into the Pi by using a software called NOOBS. For details about the installation process, the official documentation of the Raspberry Pi can be referred.The link for the documentation is:

https://www.raspberrypi.org/

Once, the installation process is completed, a check for python3 is completed and it is set as default python . Next we execute the following command:

#> ifconfig eth0 192.168.127.250

This will se the ip address of interface eth0 to 192.168.127.250 and hence both pi and server are now in the same subnet which will allow us to connect to the server through telnet.In order to do so, we need to execute telnet 192.168.127.254 and then the following diagrams explains the process that undertakes within telnet for setting the IP configuration to DHCP.



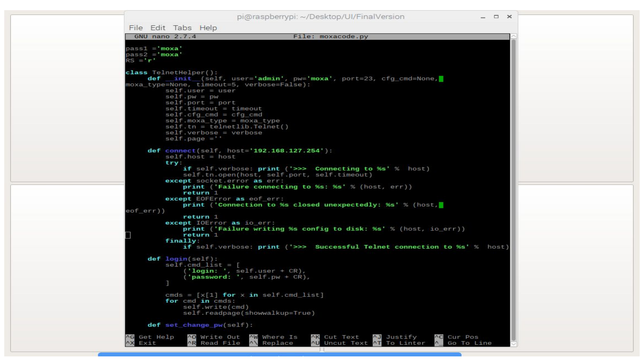
Here, the the IP configuration is manually set to DHCP, and the purpose of the project is to automate the task by using python script and also create three users with different permissions. In Raspbian, we have a tool called Wireshark that will convert the telnet commands into ascii code which we need in our python code.Hence, the script for setting up the IP configuration of Moxa Server to DHCP and creating three different with administrator, port_admin and guest permission is written and runs successfully.

Graphical User Interface & Code

Tool Used: PyQt5 & QtDesigner

First of all, PyQt is installed and then an user interface with four buttons(SETUP,SAVE, EXIT, RESTORE) are created using QT Designer.The user interface file looks like below:

On the other hand, another python script file is created which will implement the functionalities of these four buttons.



The github link for the code is:

https://github.com/Ashok96/SummerSLAC2018.git

Conclusions & Further Works

Finally, all the required scripts are completed and then the LCD is set up on the Raspberry Pi which makes it user friendly with the buttons being there. Hence, the setup, user creation, load and saving configuration of the terminal server has been automated and the server is completely standalone server.

The project could be implemented by accessing the server through web page and using API and javascript file to automate it instead of telnet.This could be one thing that can be considered.Also, the problem encountered while using CENTOS 7 can also be solved and try to complete the project using it instead of Raspbian.

Acknowledgments

I would like to express my gratitude to my mentor Mr. Ernesto Paiser for providing me guidance throughout the entire project. Similarly, i would also like to thank Omar E. Quijano,Veraldi Riccardo, and Franklin Ivanov-Pham for assisting me in different ways during the entire summer. and making it a wonderful experience for me.Furthermore, I would like to specially thanks on behalf of every intern to Dr Alan Fry, SLAC, and Department of Energy for making this happen.