**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.

**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:


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:

```bash
#!/usr/bin/python3

import ifconfig

# DHCP configuration

ifconfig eth0 192.168.127.250

# This will set the ip address of interface eth0 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.
```

**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:

![User Interface Diagram](image1)

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

![GitHub Link](image2)

**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 Ricardo, 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.