# Some Title

Jon Bakies, Mitchell Dunn, Elias Kapetanopoulos, and Magdy Ellabidy
Department of Computer Science and Networking
Wentworth Institute of Technology
Boston, MA 02115, USA
bakiesj@wit.edu, dunnm8@wit.edu, kapetanopoulose@wit.edu,
ellabidym@wit.edu

# I. Serial Over Lan

## A. ser2net

Ser2net is a daemon that opens a TCP or Telnet connection to the server's serial ports. After connecting all of the devices' communication ports to the Raspberry Pi ser2net allows for a single telnet connection to each device. There are some nuances that come along with this daemon. Ser2net does not provide a service for SSH, so the communication between the user and the device will be unencrypted. Although SSH would be preferred, a telnet connection is acceptable because in order to access the Serial over LAN (SoL) a user has to VPN into the network. The only way to access the network is physically or via a secure connection. Another nuance is the configuration. The way ser2net opens a connection to the serial ports is by the device in the /dev directory. The devices in the /dev directory are named by the order in which they are plugged in, meaning the needs to be an order in which they are plugged in.

*1) Configuration:* The default ser2net config file is located at /etc/ser2net.conf. The configuration file defines all of the connections ser2net provides. An example of connection for this SoL project is *192.168.1.12,5560:telnet:0:/dev/ttyUSB1:9600 8DATABITS NONE P5R3 remctl*. It is broken into parts *<network port>:<state>:<timeout >:<device>:<options >*. The network port defines the end point a user will use to connect to the serial port. The state option is used to determine the protocol used to communicate with the serial port. The third option is the timeout in seconds. The will be disconnected if there are this many seconds of no activity. This option is used to prevent inactive users from blocking others from using the device. After timeout is the device, which as previously discussed is located in the /dev directory. The final part is designated for options, separated by space. Options for this project are specifically for a Cisco console device. The P5R3 option is the name of a banner created earlier in the file that lets the user know which device they're connected to.

*2) Installation:* ser2net is available on the apt package manger, so install is simple.

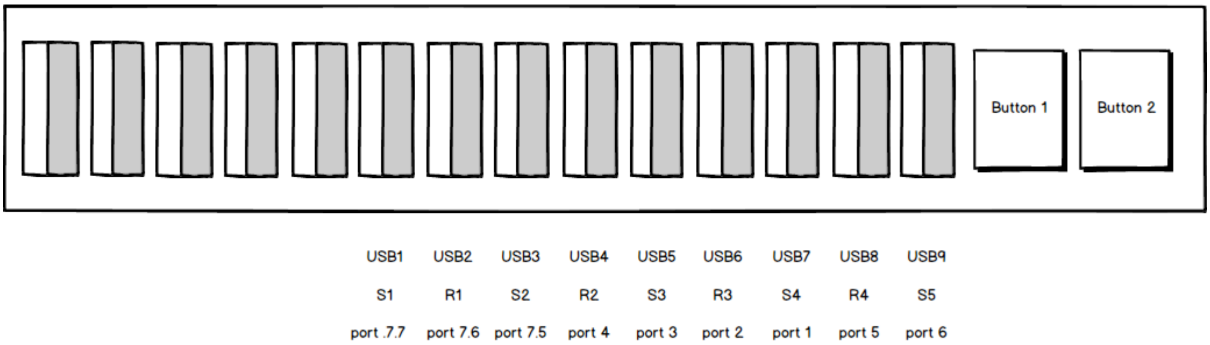
*apt install ser2net*

## B. USB hub

The USB hub is used to limit the wires run to the server room. All of the devices in the pod are connected to the USB hub. In order to constantly have the same device number, the devices need to be plugged in a specific configuration. Figure 1 shows a mockup of the usb hub. Each used port is labeled with the device that is plugged into it and the phsycial port address used by the hub. These are important to note because they are what determine ser2net's configuration. The port number describes the physical port of the device atached, which is directly mapped to the device file. The relationship between device and port is very important and should be standardized.

## C. USB-RJ45 Converter

The USB hub is plugged into a USB to RJ45 converted to extend the connection into the server room. The reason this is needed is because a USB cable is rated for only 5 meters in length without a repeater. RJ45 on the other hand has a much longer rating of 100 meters, which is more than sufficient for the lab set up. After the RJ45 line is run into the server room, it will be converted back into USB and plugged into a port on the Raspberry Pi.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| USB1 | USB2 | USB3 | USB4 | USB5 | USB6 | USB7 | USB8 | USB9 |
| S1 | R1 | S2 | R2 | S3 | R3 | S4 | R4 | S5 |
| port .7.7 | port 7.6 | port 7.5 | port 4 | port 3 | port 2 | port 1 | port 5 | port 6 |

Fig. 1: USB hub mockup