Technical Manual

Network Activity Monitor

Dawson ElectroTech 2019 Final Project
(Computer Network)

Author: Yihuan Zhang

Instructor: Nick Markou

Winter 2019

Contents

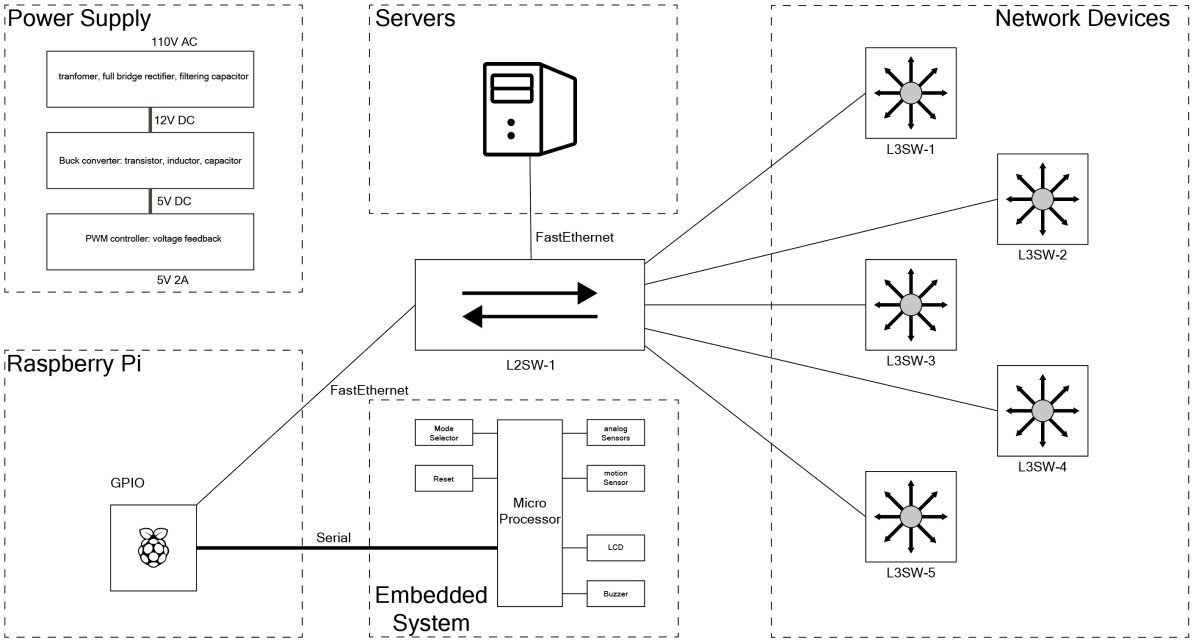
System Description

1. Introduction & Acknowledgments

The “Network Activity Monitor” (NVM) system is a final project in ElectroTech program at Dawson College. The system includes DevOps applications but not limited to softwares and networks. As System Administrators (SA) are required not only to maintain the network and servers, but also are expected to be responsible for the security (in all aspects). Therefore, an administrative system integrated with various of tools used to help administrators in troubleshooting network physically and virtually is urgently needed, as an intranet grows. In general, the Network Activity Monitor implements security in Physical Layer , Data-Link Layer, Network Layer and Transport Layer, there are multiple strategies deployed on the racks, switch ports, network edge device, and all other servers to ensure the administrated network is secure from all kinds of attack.

Before I enrolled the Electronics Engineering Technology Program at Dawson College, I was hesitating in choosing the program between Computer Science, Computer Engineering, and Information Technology. I wished I could work in network field in the future but I was also interested in electronics. Gladly, this program provides me an option to pursue the subjects in network filed. I then decided to take the advantage of my electronic and networking skills in my final project, and this is how the idea of this project born. I would like to thank my parents for supporting me in the past 3 years, I am also grateful to all my professors, chairpersons and the technicians in ElectroTech program, the lab partners I collaborated with, and all my colleagues I was working with during my last summer internship.

2. Block Diagram



The block diagram shows how the NAM system is logically deployed in a Local Area Network (LAN): a group of network devices on a rack connects to an administrative VLAN (VLAN 10). This VLAN is isolated from users access layer in order to prevent the network attack from inside, therefore it is highly confidential and is secured by “Sticky Mac Address”, RSA 1024 asymmetric key, community string, and permissive ACLs.

The administrators have the access to each network device through SSH with particular privilege or the root access through the console server, either way can be authenticated from the Radius server (on Domain Controller) or local database. In the administrative VLAN, a SNMP server monitors the network activity on each devices (SNMP nodes), sends alerts to notify the system administrator for certain SNMP traps.

In physical layer, an embedded system with sensors on it monitors the environmental activity in the Equipment Room. It updates the environmental data to the Raspberry Pi which allows authorized technicians to check them through a website.

In other words, the NAM system saves a network technicians a lot of time on their daily routine job of checking and maintaining network devices in ER. With this system, they can remotely monitor or configure a large-scale network without being presented in person.

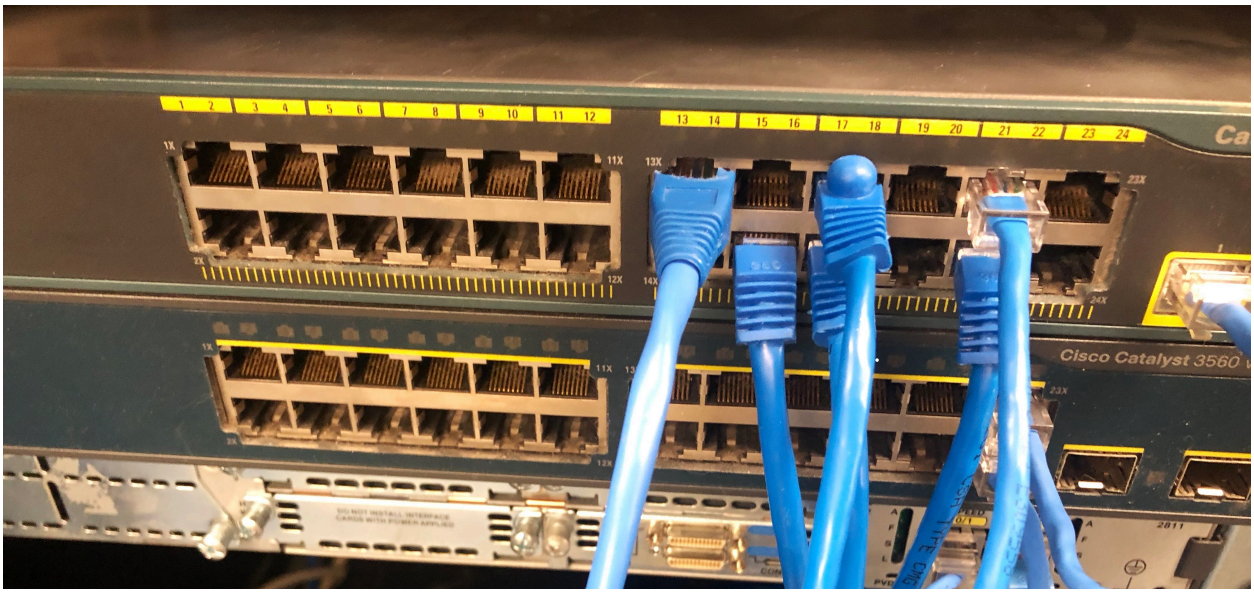
3. User Guide

To set up the system properly, make sure all the servers are functioning with correct IP addresses, embedded PCB board is power connected, and pins are mapped correctly.

Device	Interface	IP address	Subnet Mask	Default Gateway
ESXi server (phy)	R720 e0/1	10.210.20.100	255.255.0.0	10.210.0.1
DHCP server	R720 e0/1	10.210.10.3	255.255.0.0	10.210.0.1
Domain controller (DNS)	R720 e0/1	10.210.10.4	255.255.0.0	10.210.0.1
SNMP server	R720 e0/1	10.210.10.5	255.255.0.0	10.210.0.1
SAMBA	R720 e0/1	10.210.10.6	255.255.0.0	10.210.0.1
Console server (phy)	E10/100	10.210.10.7	255.255.0.0	10.210.0.1
Pi	NIC	10.210.10.8	255.255.0.0	10.210.0.1
NTP server	R720 e0/1	10.210.10.9	255.255.0.0	10.210.0.1
DMZ server (phy)	NIC	192.168.1.2	255.255.255.0	192.168.1.1
Web server	NIC	10.210.10.2	255.255.255.0	10.10.10.1
AAA server	NIC	192.168.1.4	255.255.255.0	192.168.1.1
Router1	FE0/0	192.168.1.1	255.255.255.0	n/a
	FE0/1	10.1.1.2/30	255.255.255.252	n/a
Switch1 (L3)	F0/24	10.1.1.1/30	255.255.255.252	n/a
	F0/23	10.210.0.1	255.255.0.0	n/a
	F0/0	10.210.1.1	255.255.255.0	n/a
ASA	E0/0 (VLAN 2)	DHCP	255.255.0.0	10.210.0.1
	E0/1 (VLAN 3)	192.168.2.1	255.255.255.0	n/a
	E0/2 (VLAN 1)	192.168.1.2	255.255.255.0	n/a

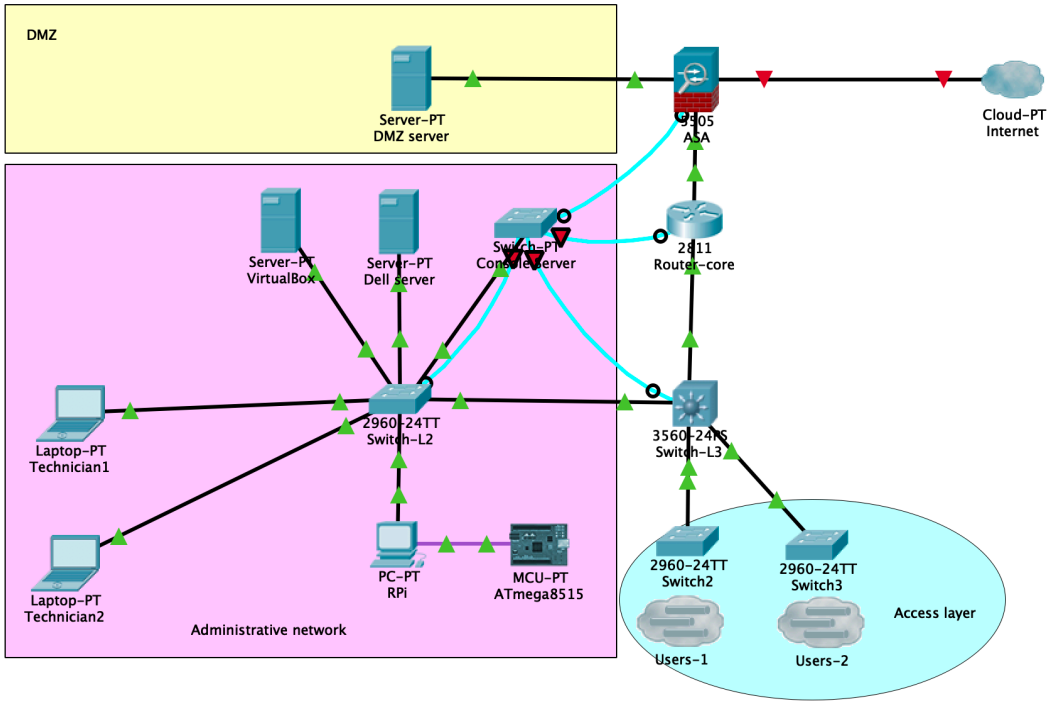
(table 1.3.1: the IP address table)

Although SAs may have their own different administrative network, the addresses for the network devices do not have to be exact same as sown above, but all the devices should be in the same broadcast domain.



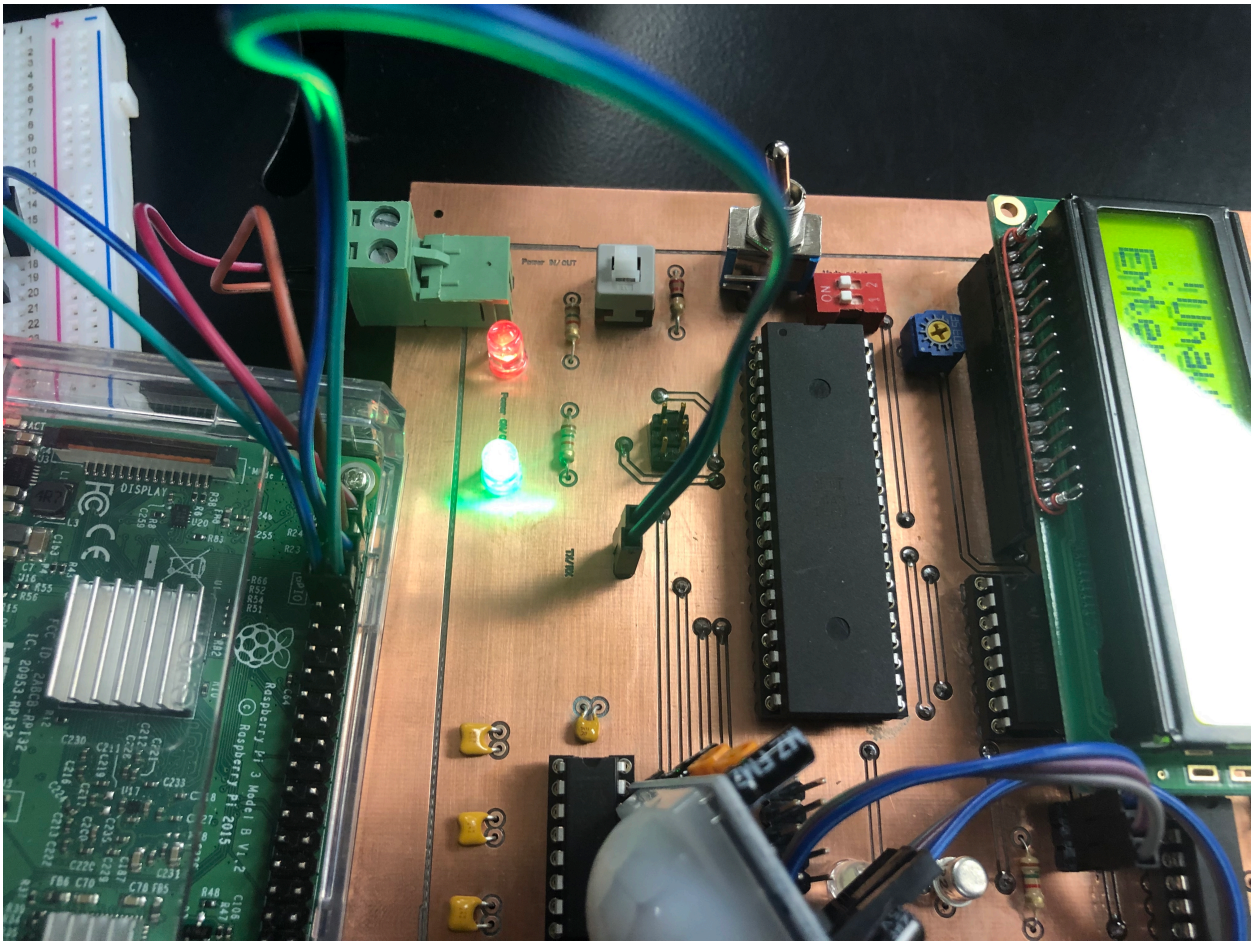
(figure 1.3.2: the switch ports)

As I configured in the layer 2 switch, ports FastEthernet0/13~24 are set as switchport access mode assigned in VLAN 10. So on the graph above, only the 6 ports on the right side of the top switch are configured in the administrative VLAN. All the rest ports are in default VLAN (VLAN 1), they are out of the administrative domain (DawsonEET2019.com).



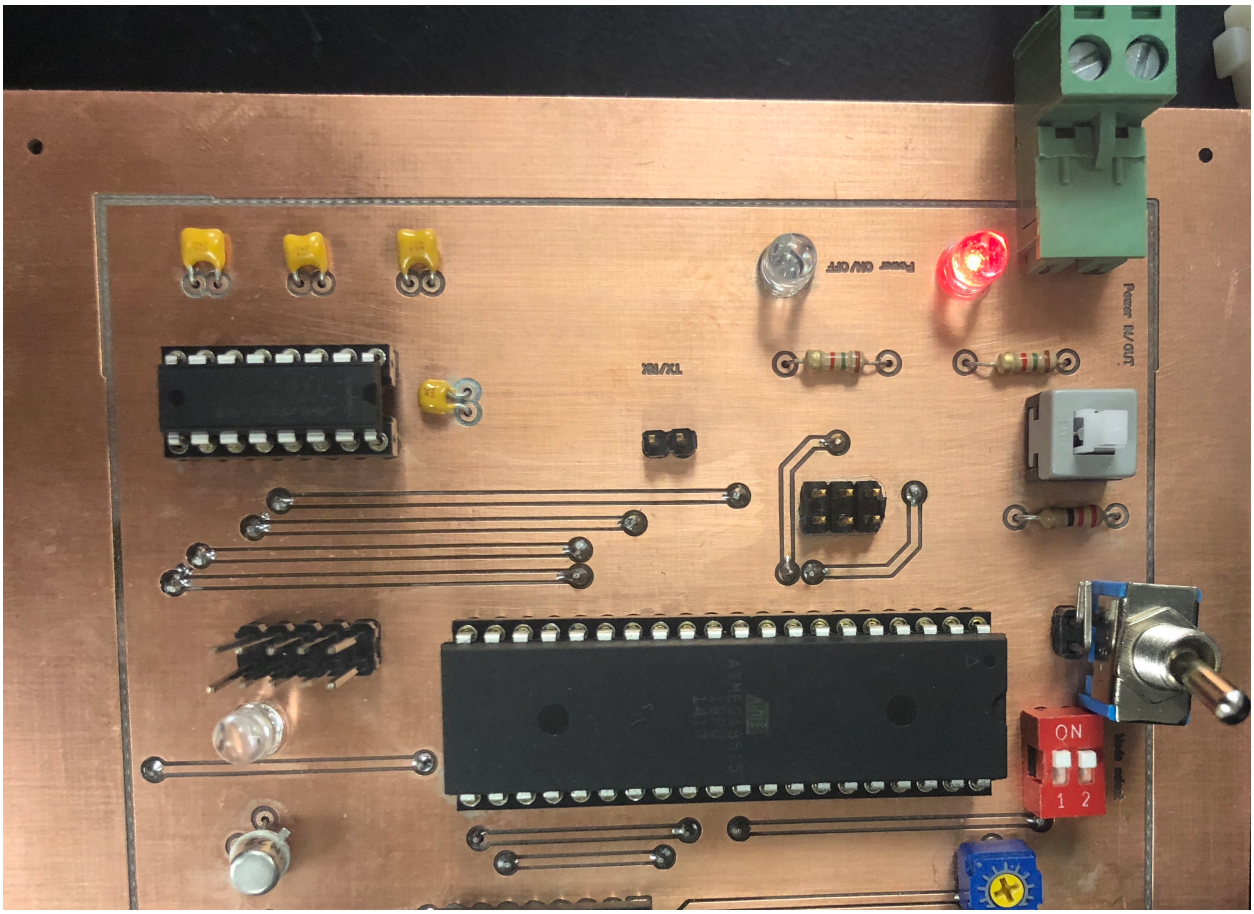
(figure 1.3.3: the network topology)

In the topology above, an administrative network (in purple) has assigned in a “DawsonEET2019.com” domain, all the devices are mounted on a 22U rack. The DMZ and access layer are not included. Administrators can change the ports, cables or servers in order to adapt their network.



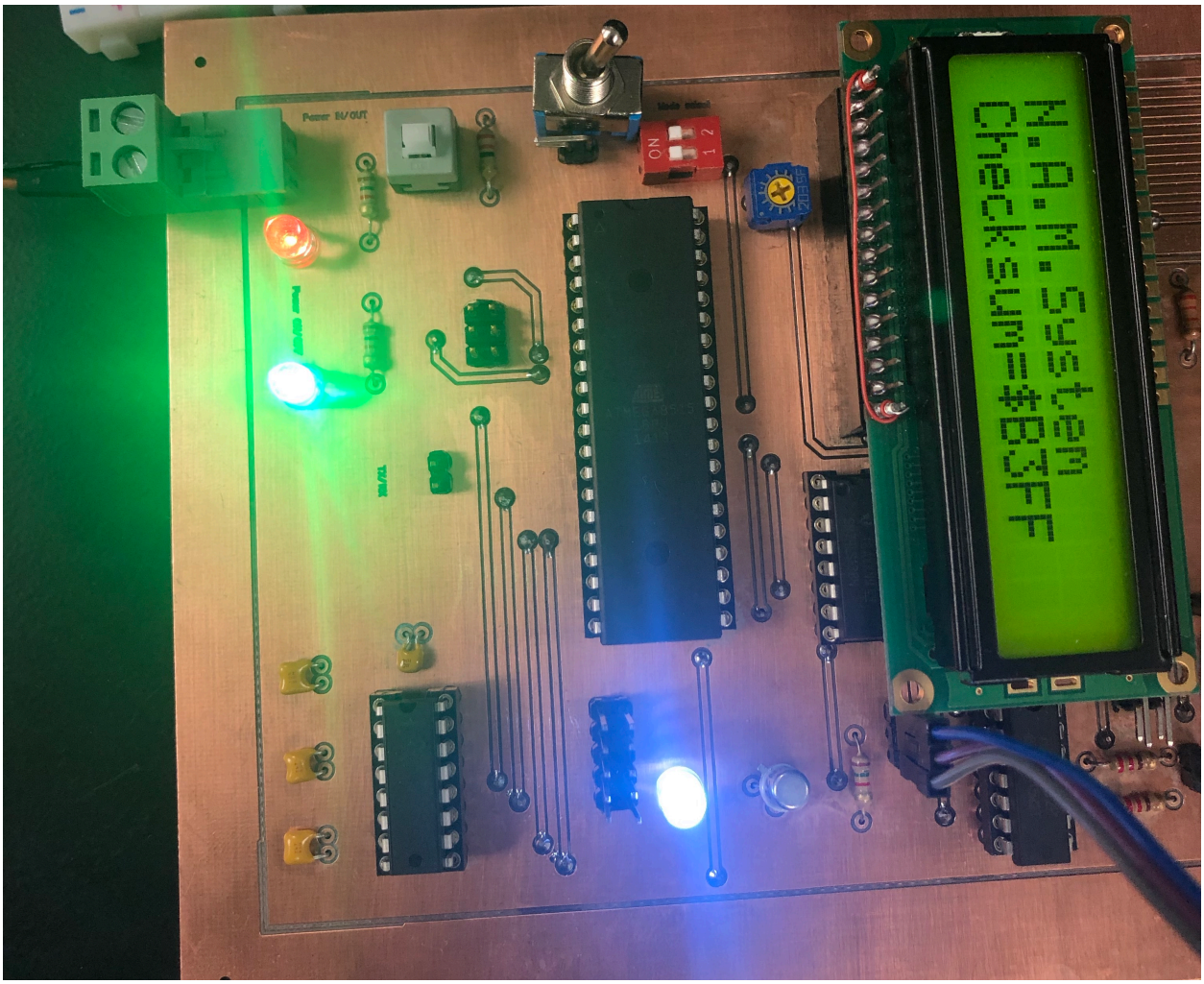
(figure 1.3.4: the serial connection)

The RPi and Embedded PCB board should be interconnected through serial ports, the RX ought to connect to the TX on the other side and vise versa.



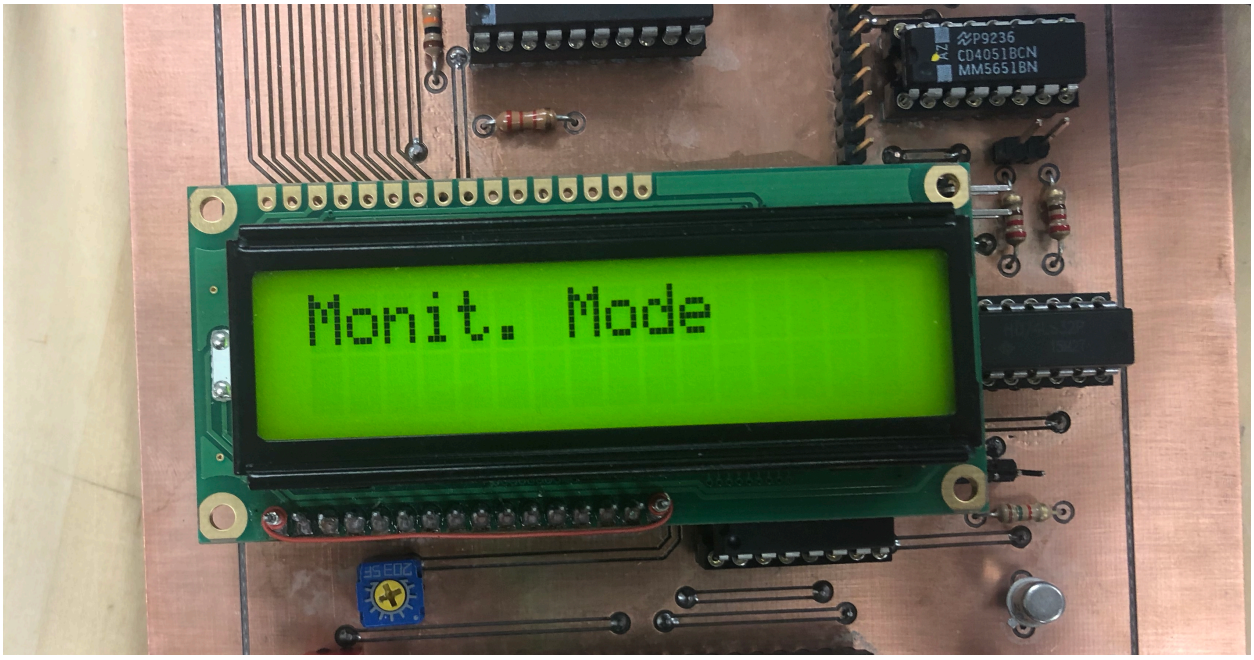
(figure 1.3.5: the power indicator)

On the embedded system, make sure the power is supplied. There are two parallel power input on this PCB board, either through the terminal (besides the power button) or the headers (besides the mode switch). When the 5V power is supplied from the screw terminal, the headers can be used as a power source to supply other devices. But when the power is supplied from the headers on the system, the screw terminal will not be functioning. Soon as a 5V power is connected to the system, only the red LED would be on to indicate the power-ready status. In this sleeping mode, the embedded system would be not functioning and only consumes 0.1W power.



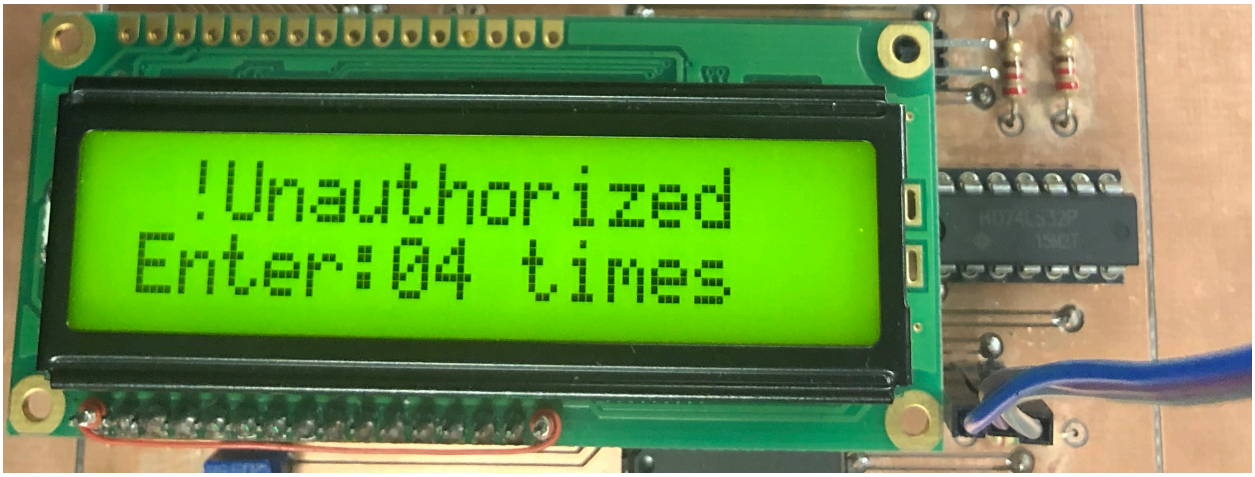
(figure 1.3.6: the power-on)

While the power button is pressed, a green LED would turn on, MCU starts to outputs the banner message and checksum information. The white LED at the bottom is serially connected to the motion sensor, it will turn on during the power-on checking stage (POST). And then the embedded system would work in either “Monitor Mode” or “Configure Mode” according to the mode selector.



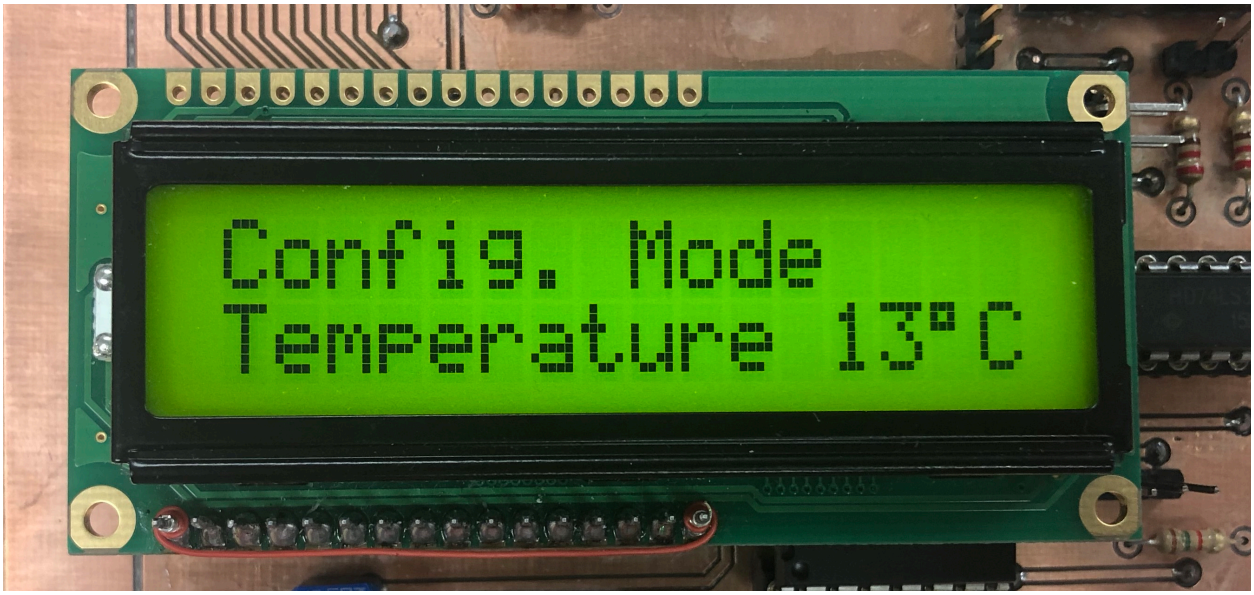
(figure 1.3.7: the Monitor Mode)

Commonly, the embedded system work in the Monitor Mode. It sends the environmental data (i.e. humidity, temperature) to the serial port (USART) at frequency of 1Hz. The MCU would constantly update the data to the Pi server until the motion detecter is activated.

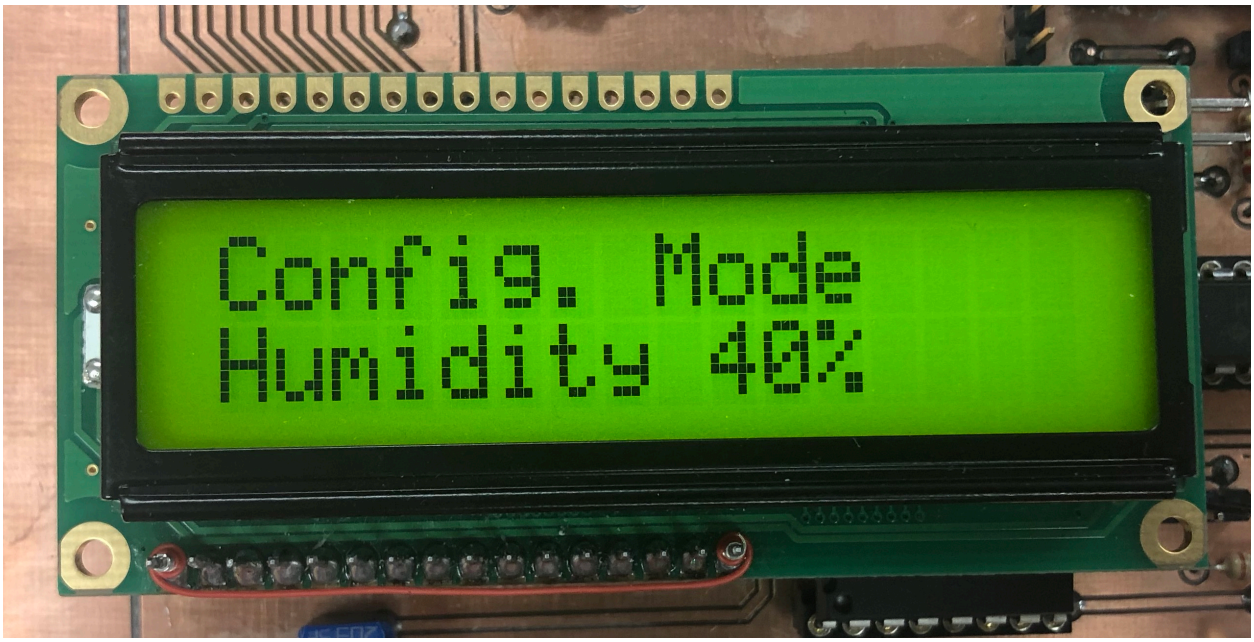


(figure 1.3.8: the Alert Mode)

The alert mode would be activated when there is a signal coming from the motion sensor (3-pin header besides the LCD). This signal is only significant in Monitor Mode since the Interrupt is disabled in Configure Mode. And there is a counter (99 times maximum by default) in Alert Mode counts the times of detection of infrared radiation within a preconfigured duration.

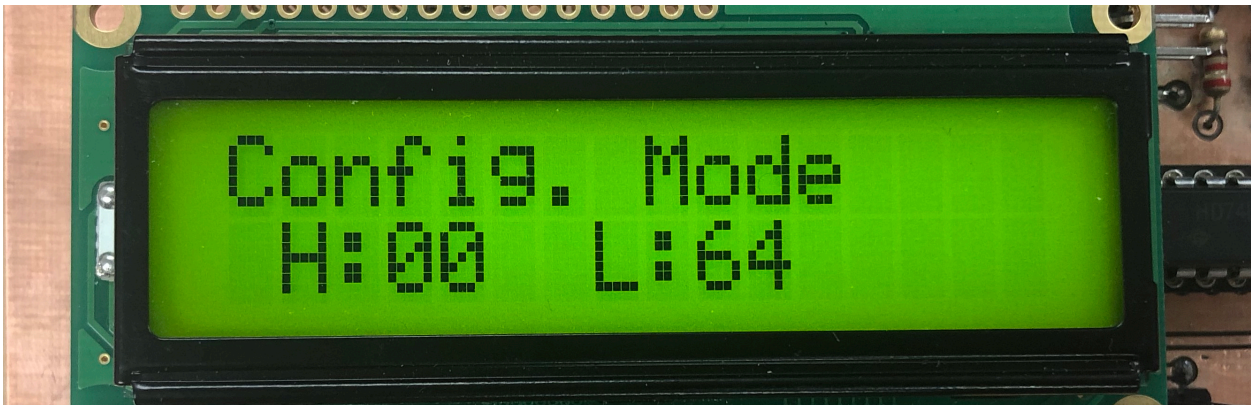


(figure 1.3.9: the data display)



(figure 1.3.10: the data display)

On the embedded system, there are analog sensors mounting on it. Instead of updating the data to the server through serial port, the data would be shown on the LCD screen in the Configure Mode. In this project, the embedded system is using replaceable humidity sensor and temperature sensor, there are also headers reserved (maximum 8) for additional sensors on the analog multiplexer. The system need to be calibrate every time when change or add a new sensor on it, please refer the “Hardware Description” section for detailed information.

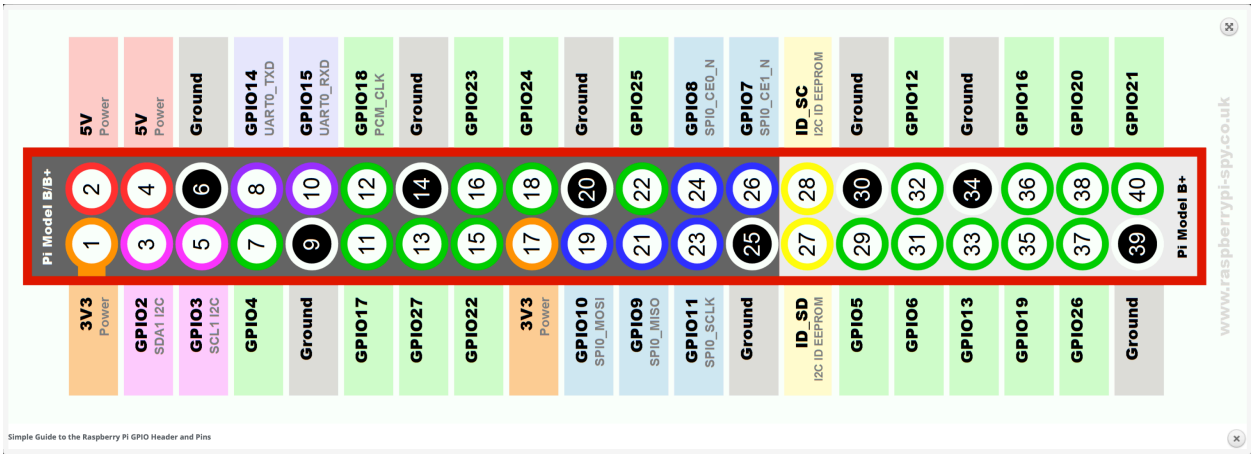


(figure 1.3.11: the trim level)

The embedded system would acquire 100 samples from the analog sensors in Configure Mode, then the MCU compares the samples with a present trim level stored in memory and displays all the samples above or below it. This application is used to monitor the environmental parameters, the operator could set up the threshold value, for example, 40˚ for temperature or 30% for humidity, the system would analyze the precise sample distribution which also reflects the variance. Notice that the value of sample is displayed in hexadecimal, and the number of samples is $64 (100) in total.

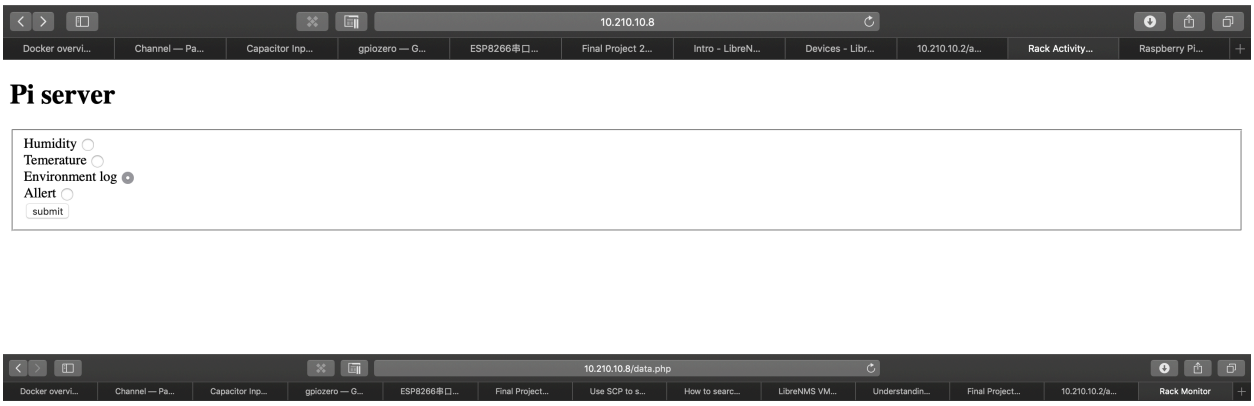
```
yihuanzhang — cu ◂ sudo — 110×36
Last login: Mon May  6 16:22:00 on ttys000
YihuandeMacBook-Pro:~ yihuanzhang$ sudo cu -s 2400 -l /dev/tty.usbserial
Password:
Connected.

E.R.Monitoring System V2.0
By Gary
1630783
B3FF[(Humidity:4B),(Temperature:0F)]
[(Humidity:4C),(Temperature:0F)]
[(Humidity:4D),(Temperature:0E)]

E.R.Monitoring System V2.0
By Gary
1630783
B3FF
Humidity = 4C
Temperature = 0F

Config. Mode:
T - Temerature
H - Humidity
S XX - Sets a new trim level
Config#T
Temperature 0F
[ H:00 L:64

Config. Mode:
T - Temerature
H - Humidity
S XX - Sets a new trim level
Config#H
Humidity 48
[ H:00 L:64
```

(figure 1.3.12: the serial interface in Configure Mode)



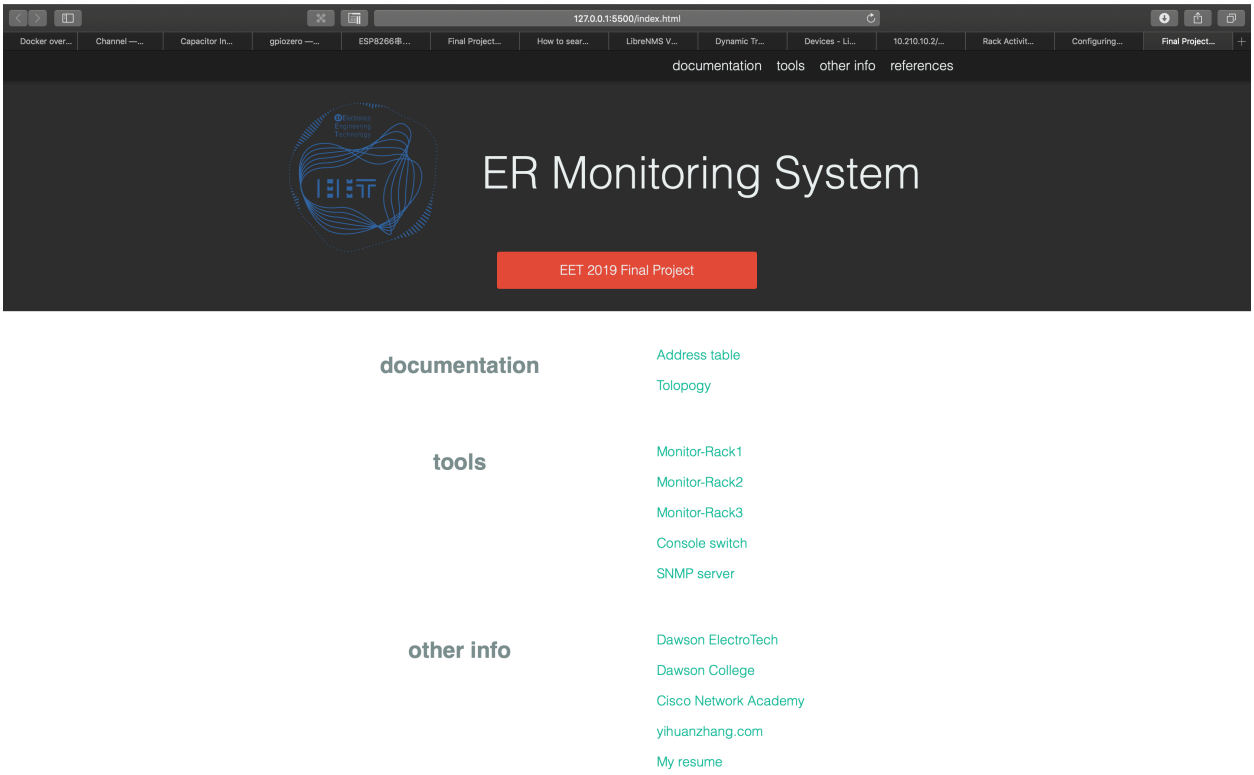
(figure 1.3.13: the RPi GPIO)

On the Raspberry Pi, the pin4 and pin 6 can be used as the power source for embedded system, pin 8 and 10 are TX/RX for serial communication with embedded system, the pin 3 controls a buzzer. Also make sure the RPi connects to the LAN through the ethernet interface, and with a 5V 2A power plugged in. This device should be running an Apache2 web server with PHP7.0 soon as the system is booted.



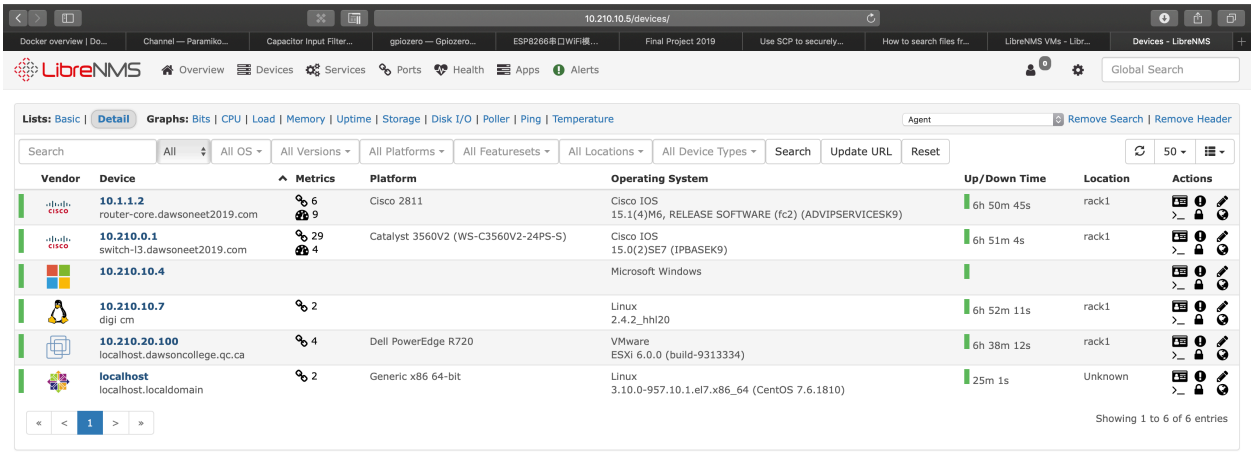
(figure 1.3.14: the RPi web page)

On the RPi website, user can choose whether to check the real time temperature or the humidity values, or the log file with all the information and timestamp (synchronized with NTP) on it. Users can also remotely activate the buzzer by choosing “alert” option in the form to submit.



(figure 1.3.15: the navigation page)

On a CentOS VM running in the VirtualBox, a Nginx web server is running in a docker container. This website provides a navigation to technicians, it redirects users to different servers as well as provides administrators relevant documents including address table and network topology. The “documentation” describes the whole administrated network, the “Monitor-Racks” binds to different embedded system unit (Pi+PCB board), each one of them is mounted on a rack or a position that needs to be monitored.

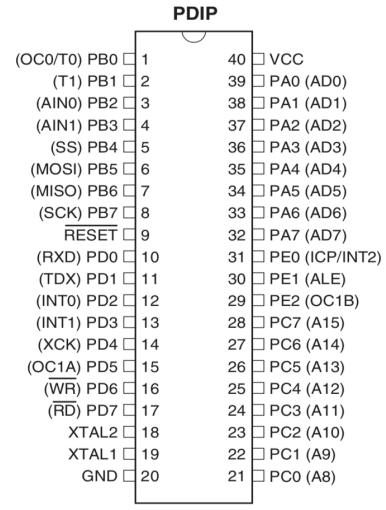


(figure 1.3.16: the SNMP manager)

The SNMP server provides a GUI on a web, all the managed SNMP nodes can be monitored and analyzed through graphs, which is useful for a network administrator to create a baseline. This Network Management System tool is using an open source software “LibreNMS”, it provides other services such as Syslog, AAA, traffic monitoring, and disk checking. The SNMP manager offers user-defined alerting rules, for more information about syntax, options, examples and API, please check their website (https://docs.librenms.org/).

4. System Specification

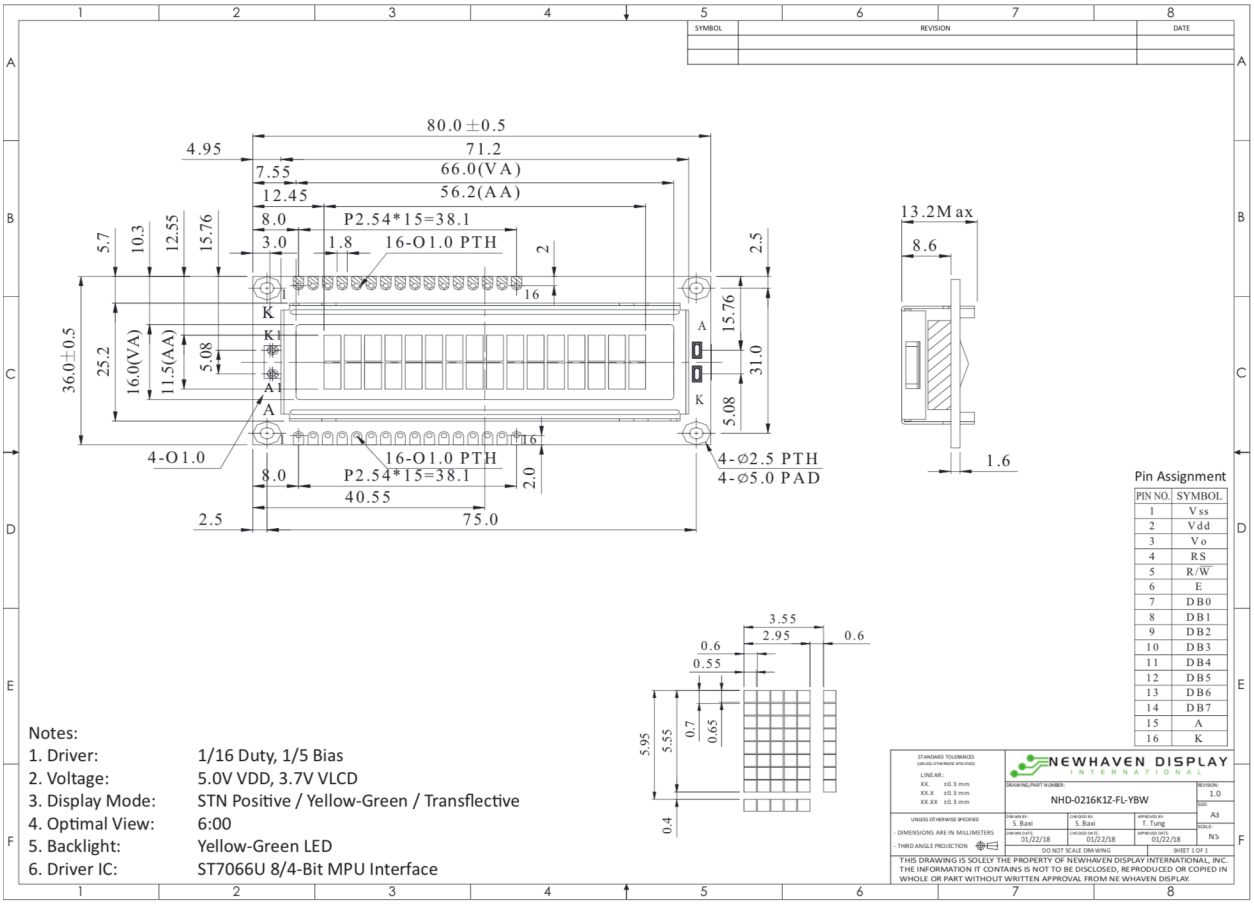
The Embedded system is PCB board based, each component has its manufacturing specification.



(figure 1.4.1: the pinout ATmega8515)

Specification	Value
Model	AVR ATmega8515
Manufacturer	Atmel (Microchip Technology)
Architecture	RISC 8-bit
Memory	8K Bytes Flash, 512 Bytes EEPROM
Operating Voltage	4.5~5.5V
Speed Grade	0~16MHz
I/O	SPI, USART, INT, BUS

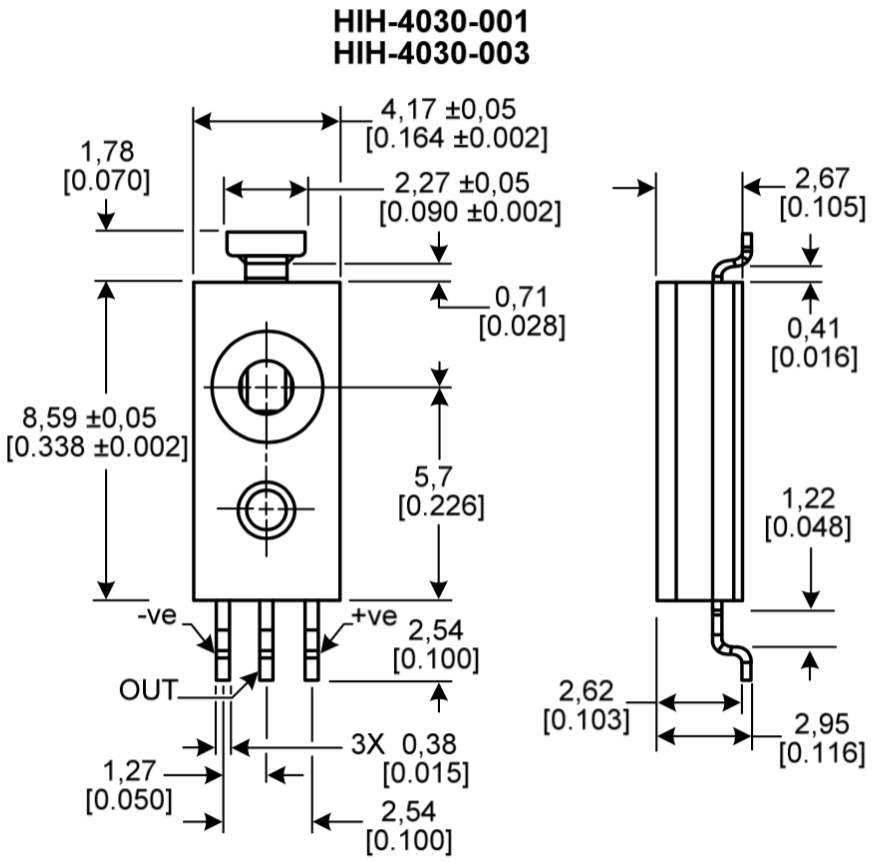
(table 1.4.2: the ATmega8515 specification)



(figure 1.4.3: the LCD diagram)

Specification	Value
Model	NHD-0216K1Z-FL-YBW
Manufacturer	Newhaven Display International Inc.
Display	2 × 16
Backlight	Y/G LED
Operating Voltage	4.5~5.5V
Response Time	150 ms (200 ms max.)
I/O	RS, R/W, E

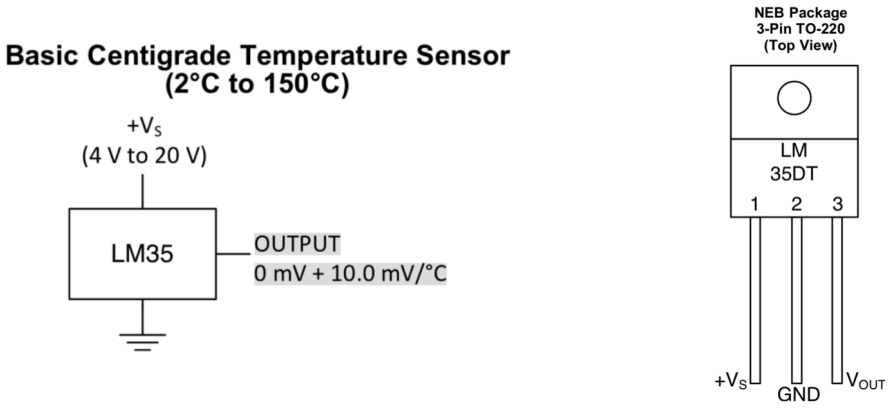
(table 1.4.4: the LCD specification)



(figure 1.4.5: the HIH-4030/31 diagram)

Specification	Value
Model	SEN-9569 Humidity Sensor
Manufacturer	Sparkfun
Operating Temperature	-40~85˚C
Accuracy	±3.5% RH
Operating Voltage	4~5.8V
Setting Time	70 ms
Signal Type	Analog, near linear

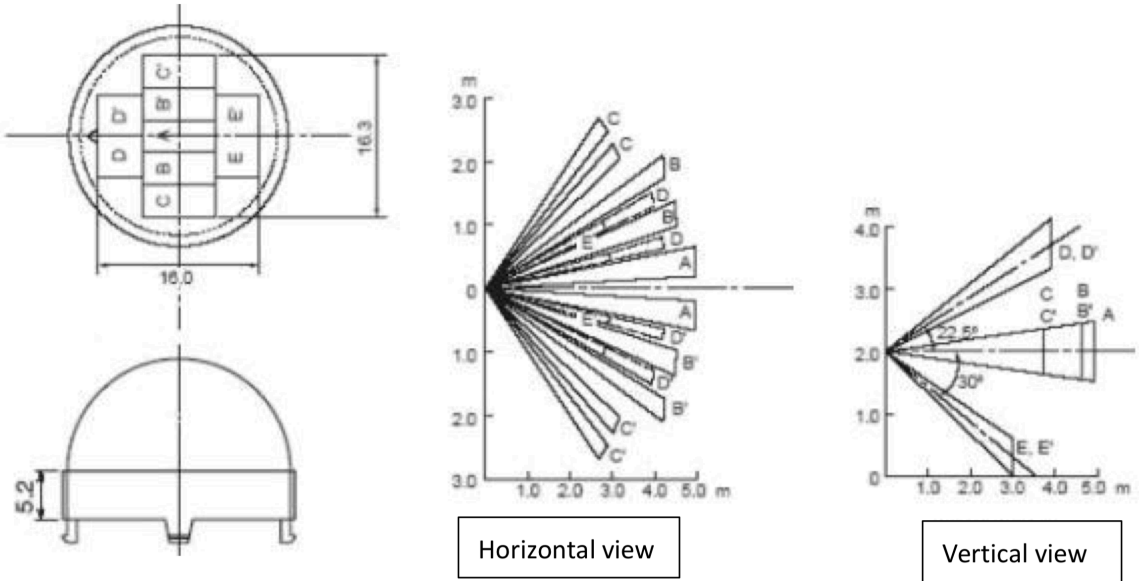
(table 1.4.6: the humidity module specification)



(figure 1.4.7: the LM35 diagram)

Specification	Value
Model	LM35 Temperature Sensor
Manufacturer	Texas Instruments
Operating Temperature	-40~125˚C
Accuracy	±1.5˚C
Operating Voltage	4~30V
Package	TO-220
Signal Type	Analog, linear 10mV/˚C scale factor

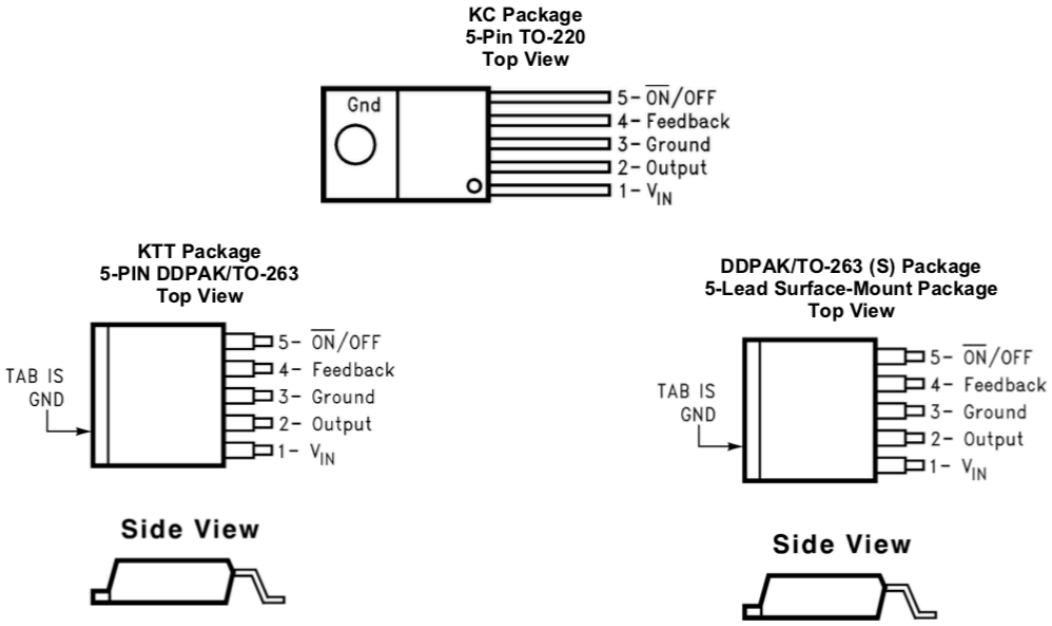
(table 1.4.8: the temperature transistor specification)



(figure 1.4.9: the RD-624 diagram)

Specification	Value
Model	HC-SR505
Manufacturer	Winsen
Operating Temperature	-30~70˚C
Wavelength	5~14μm
Operating Voltage	3~15V
Delay	Adjustable
Signal Type	TTL

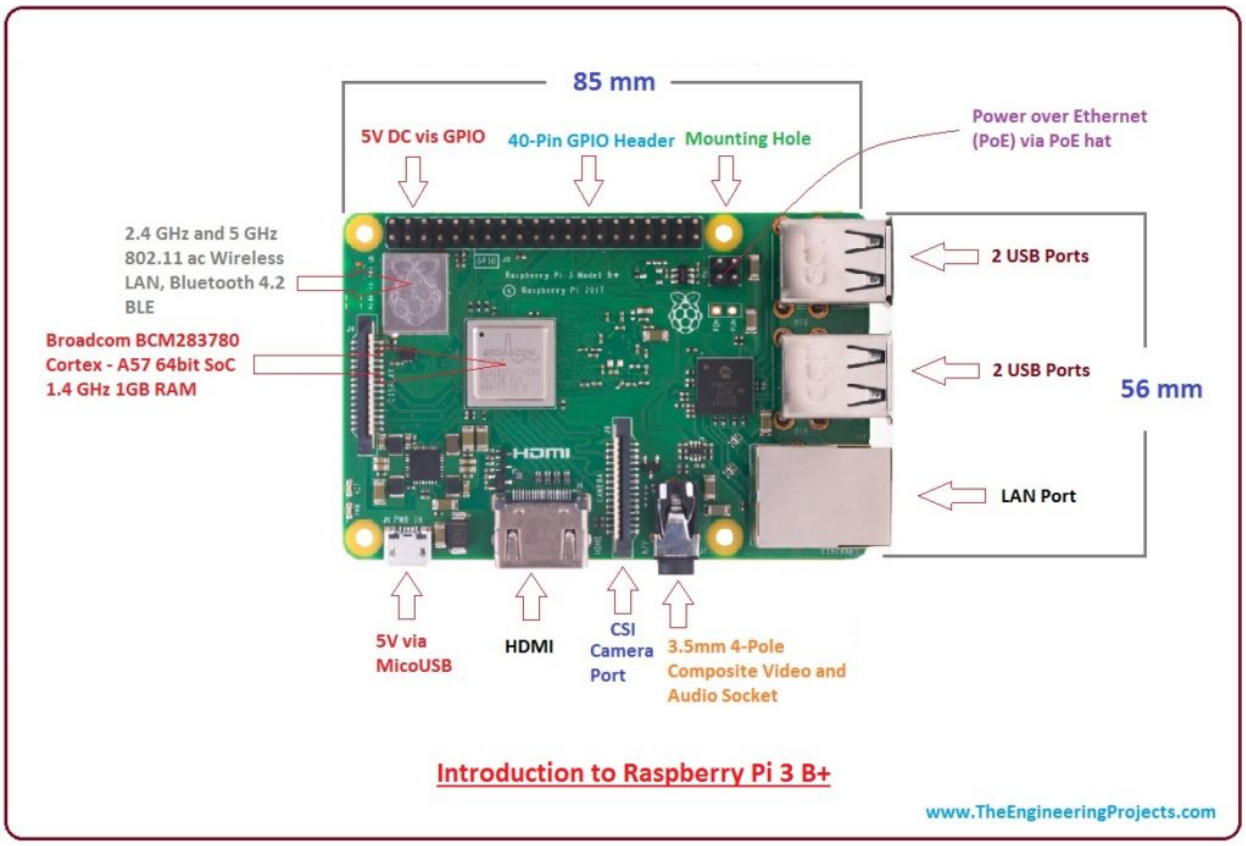
(table 1.4.10: the PIR sensor specification)



(figure 1.4.11: the LM2576 diagram)

Specification	Value
Model	LM2576
Manufacturer	Texas Instruments
Operating Temperature	-65~150˚C
Efficiency	88% @12V
Operating Voltage	40V max.
Saturation Voltage	1.4V @I_{out}=3A
Oscillator Frequency	52 kHz

(table 1.4.12: the switching regulator specification)



(figure 1.4.13: the Pi 3 Model B+ diagram)

Specification	Value
Model	Pi 3B+
Manufacturer	Raspberry Pi Fundation
CPU	A53 64-bit quad-core processor @1.4GHz
Memory	1G SDRAM, 32G Micro-SD
Operating Voltage	5.1V micro USB supply
Networking	Gigabit Ethernet port, 802.11b/g/n/ac Wi-Fi
GPIO	40-pin GPIO header, populated

(table 1.4.14: the Raspberry Pi specification)

As all the servers are virtual machines with virtual hard disks in the Dell R720 server and VirtualBox, below is the recommended requirement for each VM.

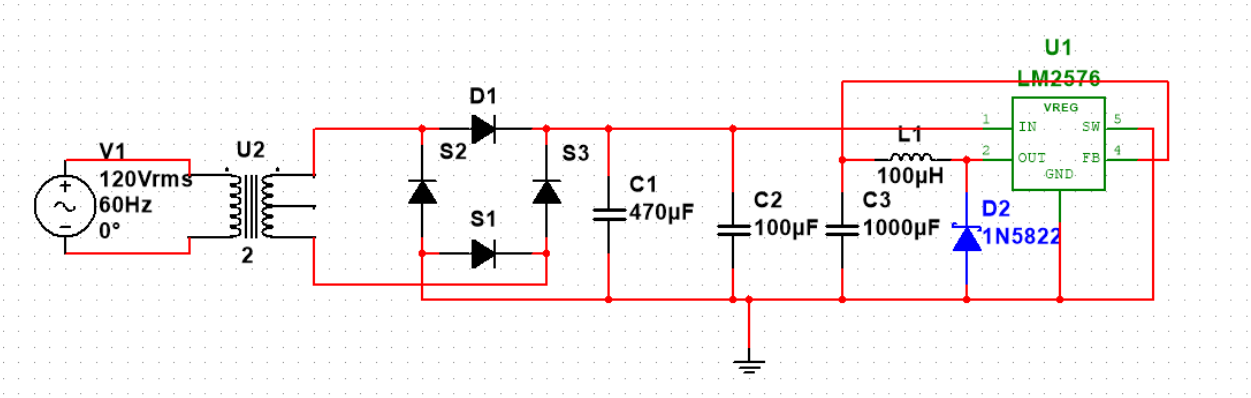
Server	O/S	RAM	Disk
Domain Controller	Windows server 2016	4G	40G
DHCP server	Windows server 2016	4G	40G
NTP server	Windows server 2016	4G	40G
SNMP server	CentOS 7.5	512M	40G
Web server (docker)	Ubuntu 16	4G	10G
NFS server	CentOS 6	2G	4G

(table 1.4.15: the hardware requirements)

Hardware Description

1. Power Supply

The switching mode power supply provides a 5V DC power (2A max.) to the embedded system. This power supply can also drive the RPi for normal condition (without running heavy programs or CPU overclock). In general there are three parts for the power supply, all of them contributes to bring the 120V 60Hz main electricity down to a 5V DC source.



(Figure 2.1.1: the schematic diagram of the power supply)

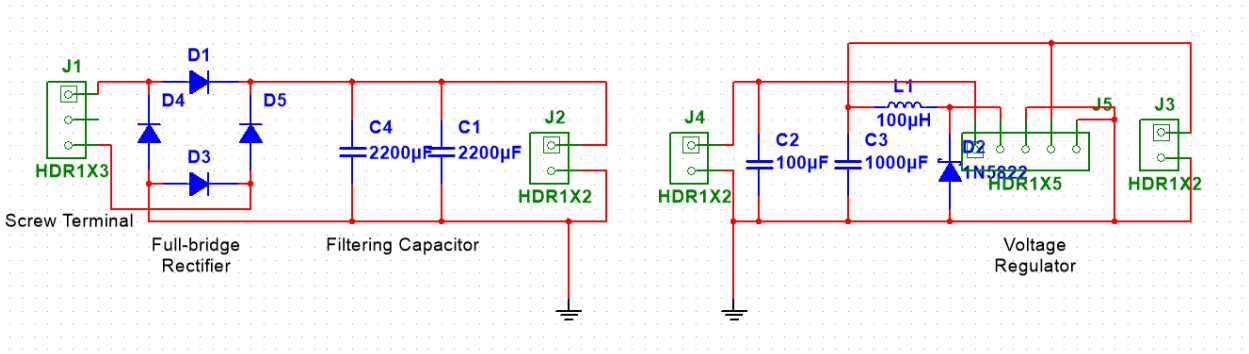
At the first stage, a transformer with winding ratio at 10:1 is used to bring down the 110V AC to about 12V AC without changing the frequency. Then a full bridge rectifier is implemented to convert all the negative power wave to positive, so that there is only one direction the current flows. The capacitors filer out the peak-to-peak ripples by temporarily store the electrons. The minimum capacitance for the capacitor (supplies 70% current of the cycle to the load) can be calculated through:

$$C = 0.7 \times I / (\Delta V \times * F)$$

As capacitors are consumables (limited charge time) and can be easily damaged by current surge, we would use multiple capacitor in parallel for redundancy in case one of them is burned during the surge happens either from the main or from the inductor (on the buck converter side). The total capacitance is the sum of each in parallel. Since smaller capacitor also reacts faster, combining different size of capacitors can additionally filter the noise and provide cleaner DC source.

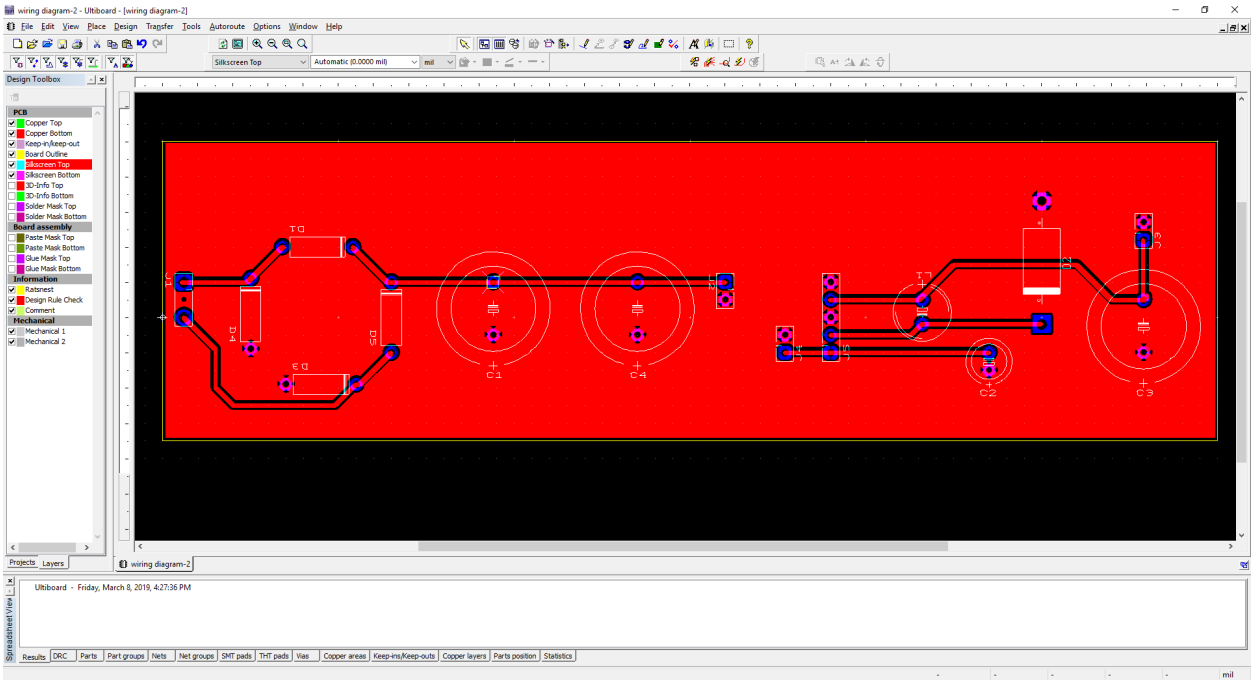
On the right side, a switching mode voltage regulator uses a integrated chip LM2576 to bring down the DC 12V to DX 5V. The output voltage returns back to the feedback (pin 4) which impact the integrated PWM to adjust the output. When the load is high, the LM2576 would

decide to increase the duty cycle to maintain the voltage output. This part of diagram uses the typical configuration, please check the data-sheet for details.



(Figure 2.1.2: the wiring diagram of the power supply)

The 4 power diodes used on the full-bridge rectifier are 1N4001s, using package "DIO8X3R10"; the two filter capacitors C4 and C1 uses package "ELKO16_5R7_5"; C2 uses "ELKO6_8R2_5", C3 uses package "ELKO16_5R7_5"; the Schottky diode D2 is 1N5822 uses "DO-201AD"; the 100µH inductor uses "ELKO8R3_5". Please note that all the footprints listed here do not match the ultimate pad sizes and drill sizes on PCB, refer to the Ultiboard file for the actual footprint information.



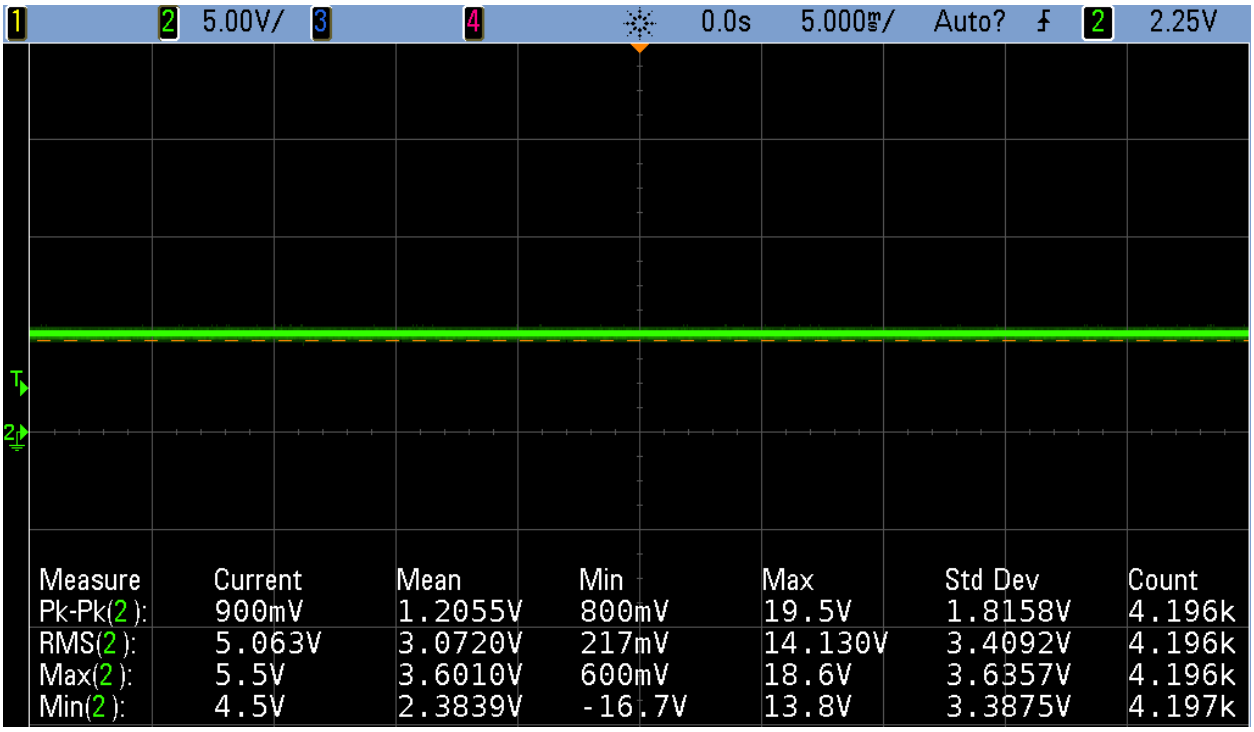
(Figure 2.1.3: the PCB diagram of the power supply)

The power supply is designed as a single board PCB. The AC-DC converter and the buck converter is separated (using a 2-pin header to connect) so they can be modularly functioning. The copper bottom is used for ground plane.

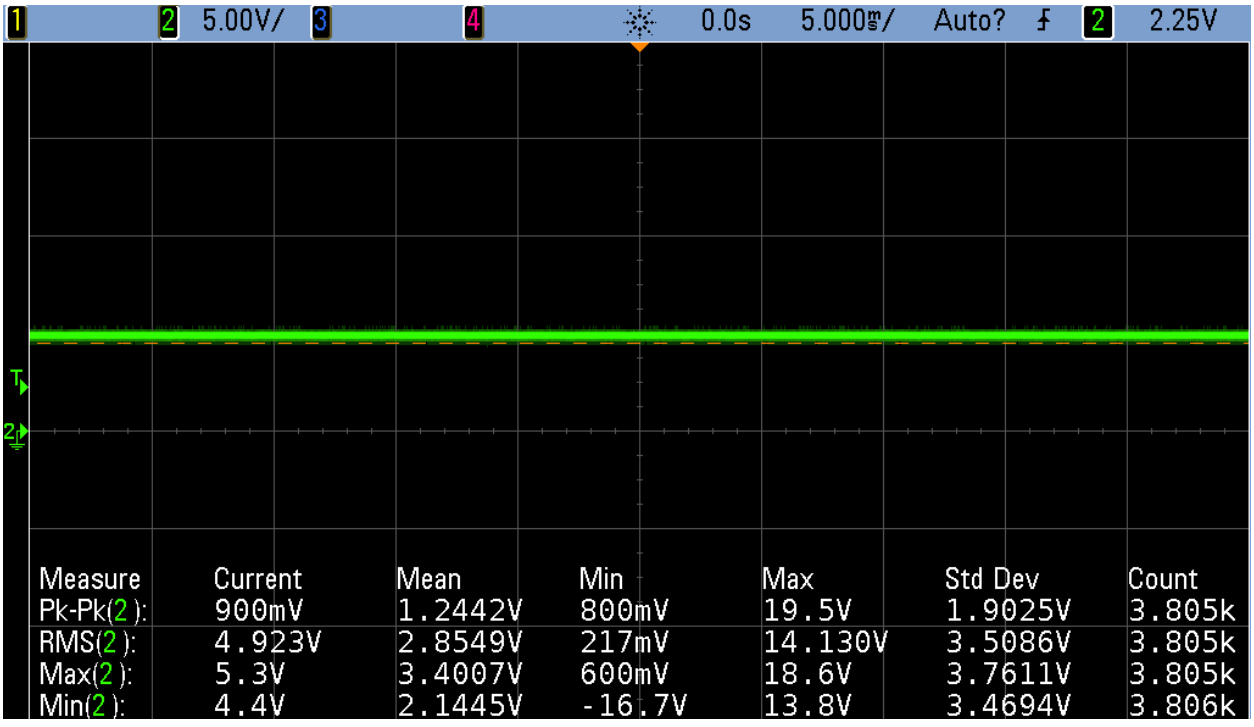
I_{out} (A) @5V	P_{out}(W)	I_{in} (A) @14V	P_{in}(W)	η (%)
0	0	0	0	n/a
0.25	0.99	0.09	1.3	76.2%
0.5	2.02	0.19	2.73	74.0%
0.75	2.96	0.31	4.4	67.3%
1	3.91	0.47	6.59	59.3%
1.25	4.66	0.64	8.88	52.5%
1.5	5.61	0.83	11.3	49.6%
1.75	6.48	0.94	12.7	51.0%
2	7.34	1.12	14.99	49.0%

(Table 2.1.4: the efficiency specification)

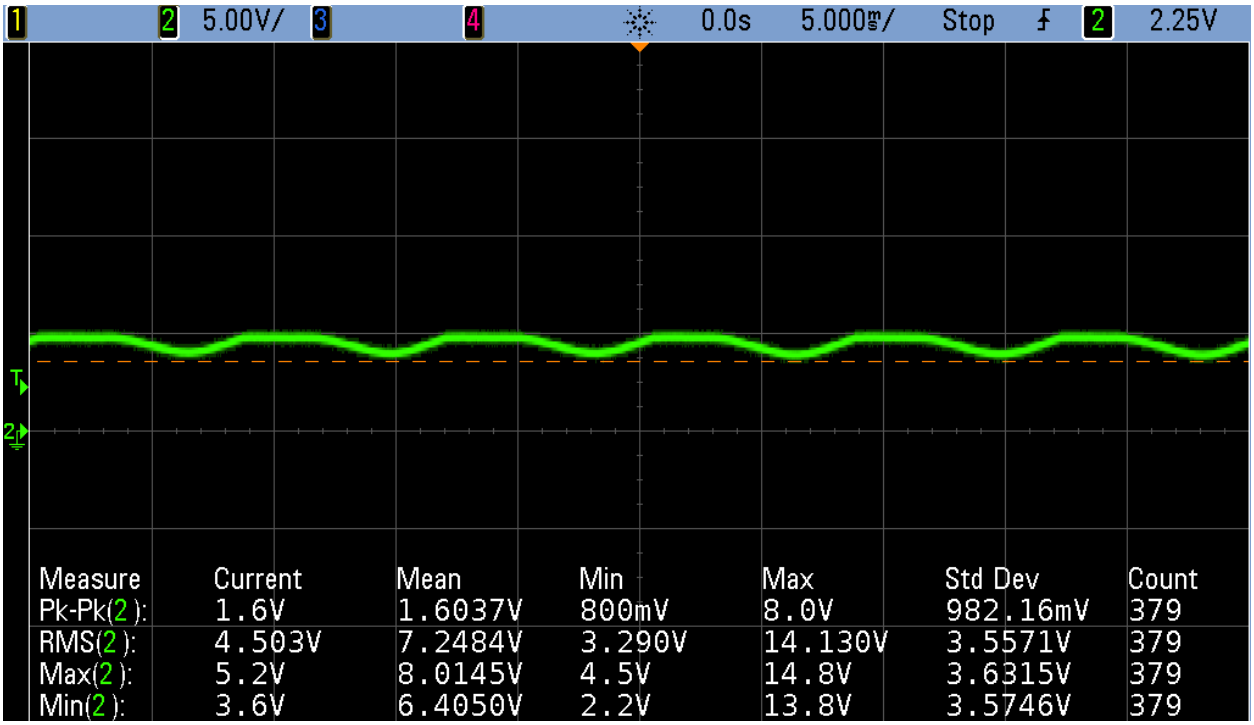
According to the measurement, this power supply has decent efficiency when the load draws less than 1A of current. In other words, the buck converter has 60%~80% of efficiency when performing DC-to-DC step down converting under 5W. If the load increases, the buck converter would loose half of the power at 10W.



(figure 2.1.5: the waveform @I_{out}=0.5A)



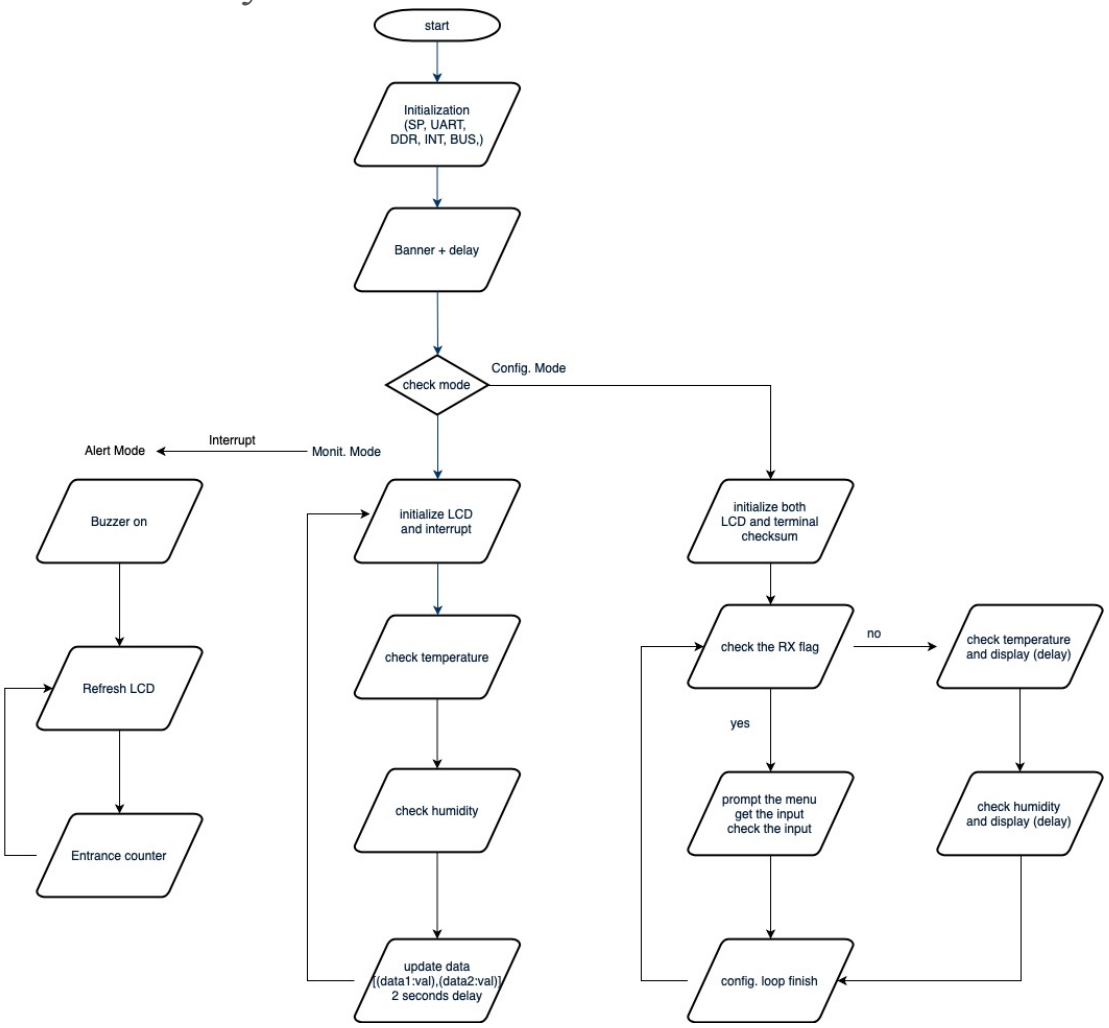
(figure 2.1.6: the waveform @I_{out}=1A)



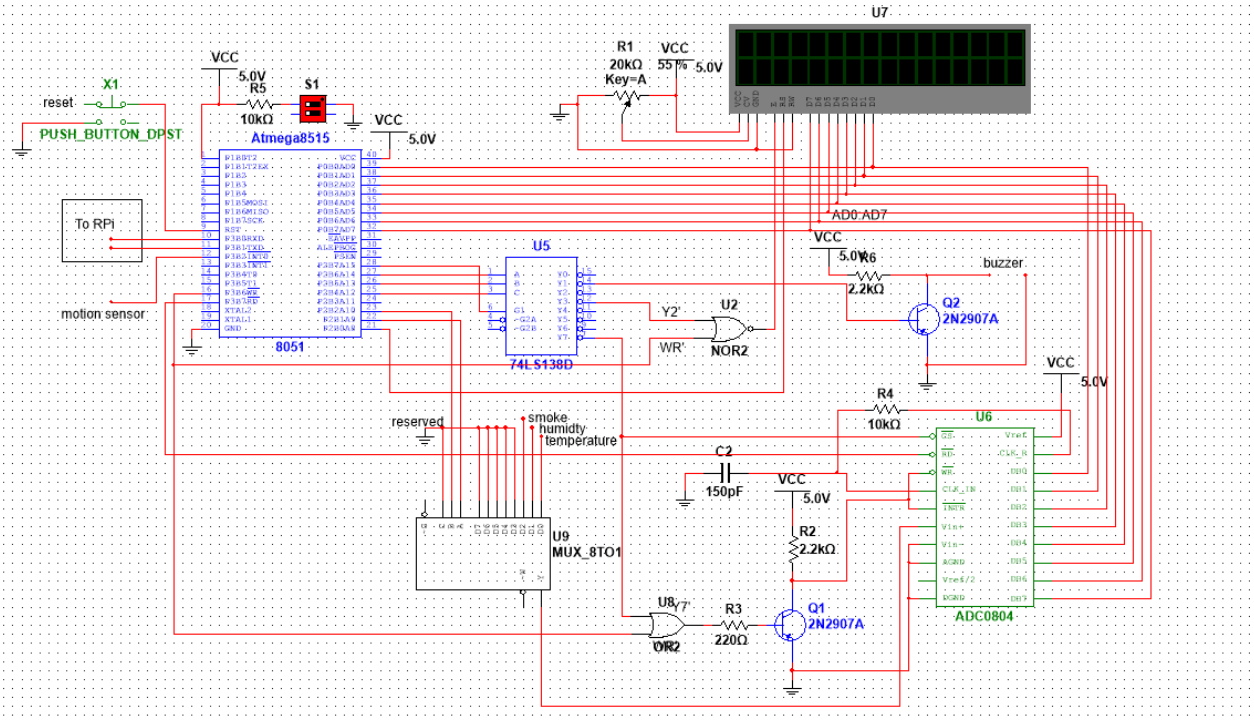
(figure 2.1.7: the waveform @I_{out}=1.5A)

As being tested in the oscilloscope, the power supply provides perfect linear waveform only when the load is under 1A, the signal starts distorting at 1.5A (7.5W).

2. Embedded System

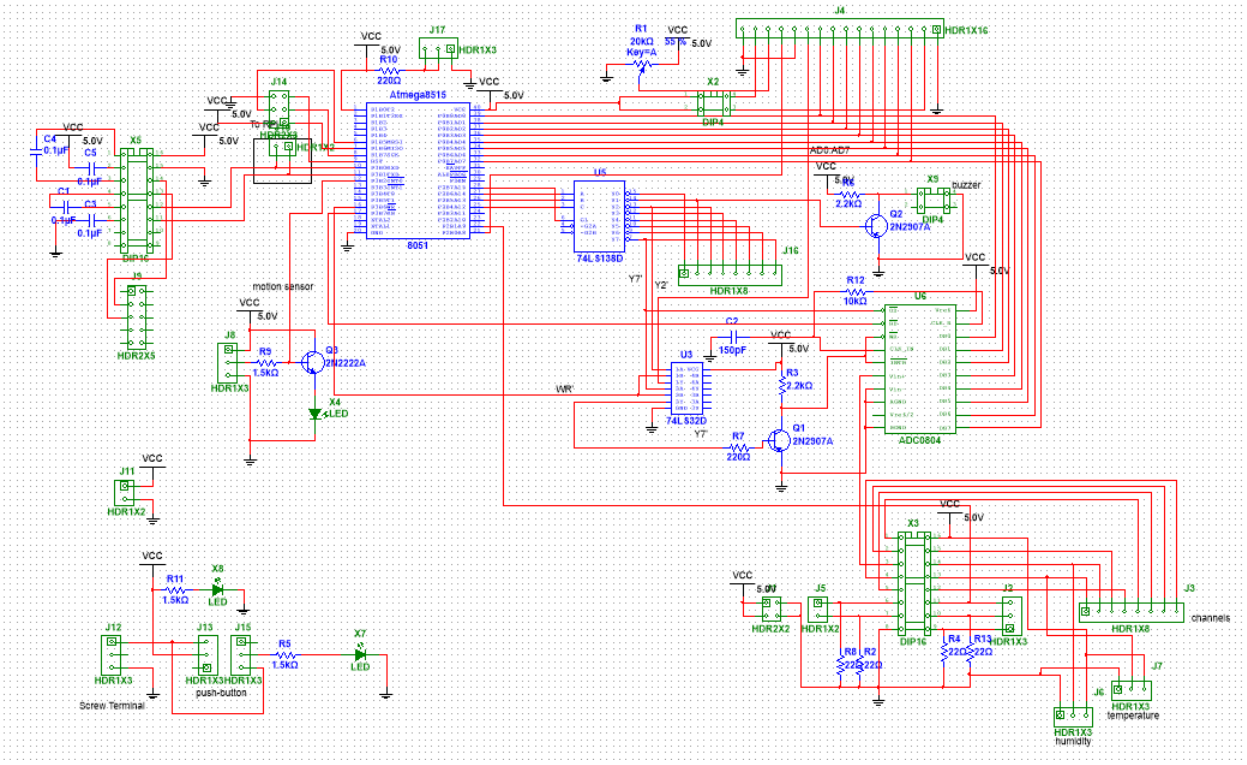


(Figure 2.2.1: the block diagram of the embedded system)



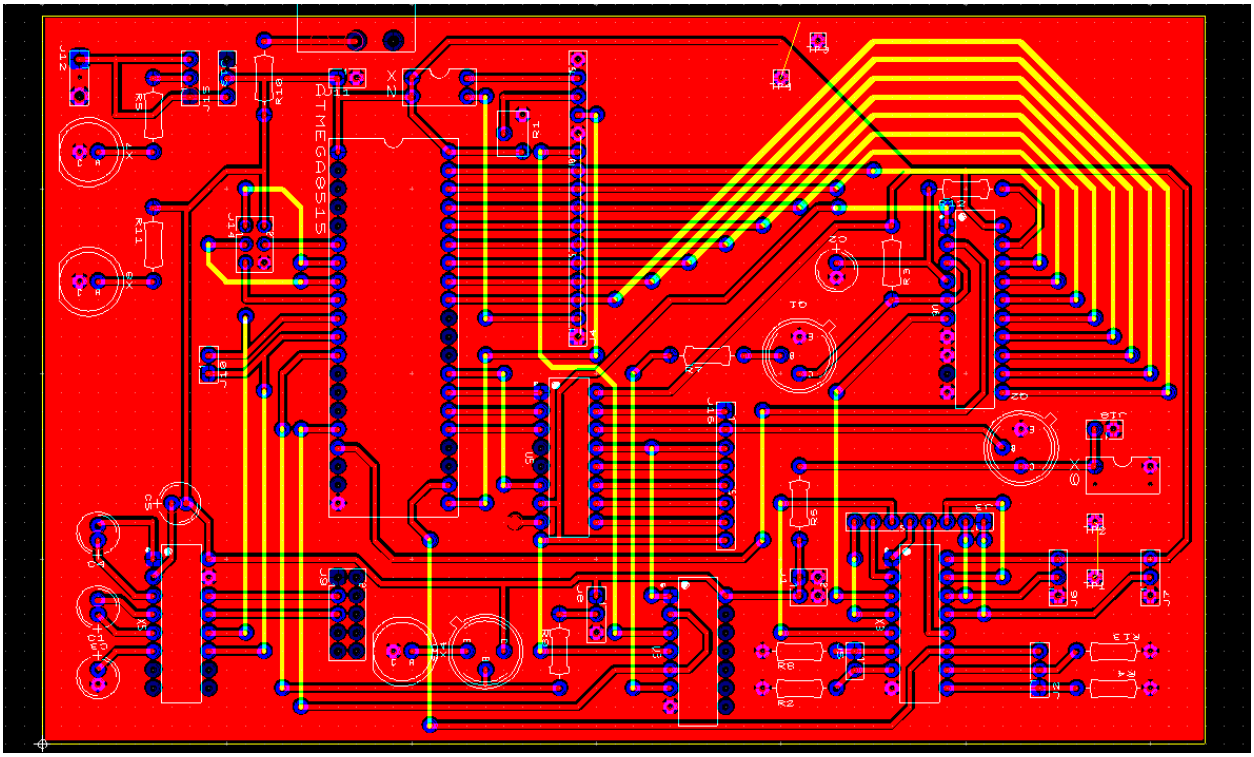
(Figure 2.2.2: the schematic diagram of the embedded system)

The embedded system is responsible for digitalize the analog signal from sensors and bridge them to the RPi through the serial transmission. The Micro-controller ATmega8515 is configured as BUS mode, with 8-bit data bus (pin 32-39) and 16-bit address bus (pin 21-28). The MCU is configured as operating clock at 1MHz, uses only 3 MSBs (A15 A14 A13) to select the external memory devices through a 138 decoder; A10 A9 choose the channel for the analog multiplexer U9; A8 enables the LCD when sending string signals or instructions to it. The interrupt INT0 is directly connected to a motion sensor, any logic high signal coming from INT0 (pin 12) will trigger the program jumping into the interrupt subroutine — “alert mode”. PB0 is initialed to select which mode to run after booting.



(Figure 2.2.3: the wiring diagram of the embedded system)

There are headers reserved for future expansion purpose on both analog multiplexer and 138 decoder, most of the pins have corresponding headers beside them (in parallel), some unused pins are connected to the ground through a 22Ω resistor, Vcc is provided through headers beside them. A 2-pin dip-switch (X2) connected to the LCD controls the LCD on/off and its backlight, for energy saving purpose. The SPI is also reserved for on-board programming, the header J14 allows writing the program (CSEG&DSEG) to the MCU through a MKII programmer. The system also provides 2 ways to access to the serial ports, a 2-pin header for RPi GPIO (3.3V/5V) and a 10-pin header for standard RS232 port (-12V).



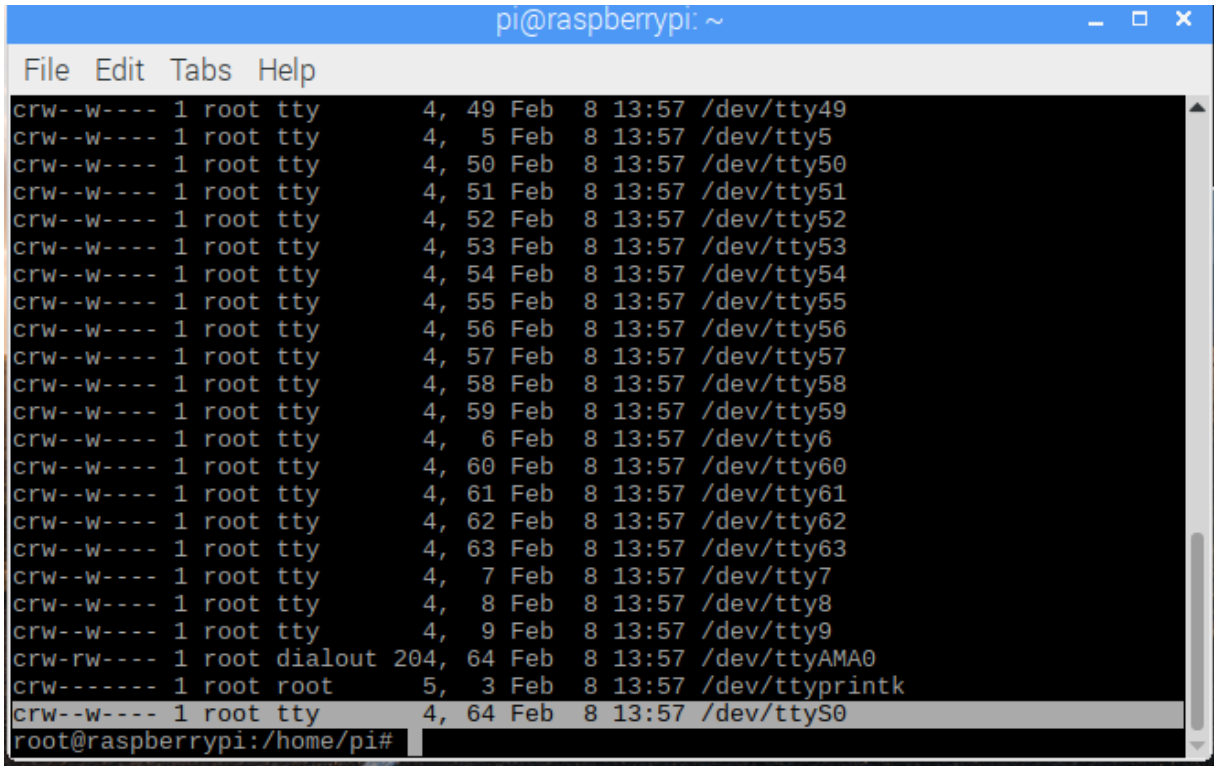
(Figure 2.2.4: the PCB diagram of the embedded system)

The embedded system PCB is designed as a double side 6inch × 4inch board. Bottom copper is configured as the ground plane, top side is used for vertical routes and data bus (to ADC). There are also jumper wires on the top to interconnect the isolated area of ground (TP). Top left are the power section, the Vcc can be either supplied from the screw terminal (J12) or from the header (J11). In addition, the other one can be used as the power source for other circuitry if there is power already connected in.

However, this design has several defects which needs to be fixed. The most important one is the pin mipmapping on 138 decoder, please double check the pinout of the decoder you are using. The selector on the top neither connects to the PB0 nor to the ground. The last pin of the RS232 connecter (ground) is not connected. These problems are fixed after soldering by using jumper wires and wire wrapping.

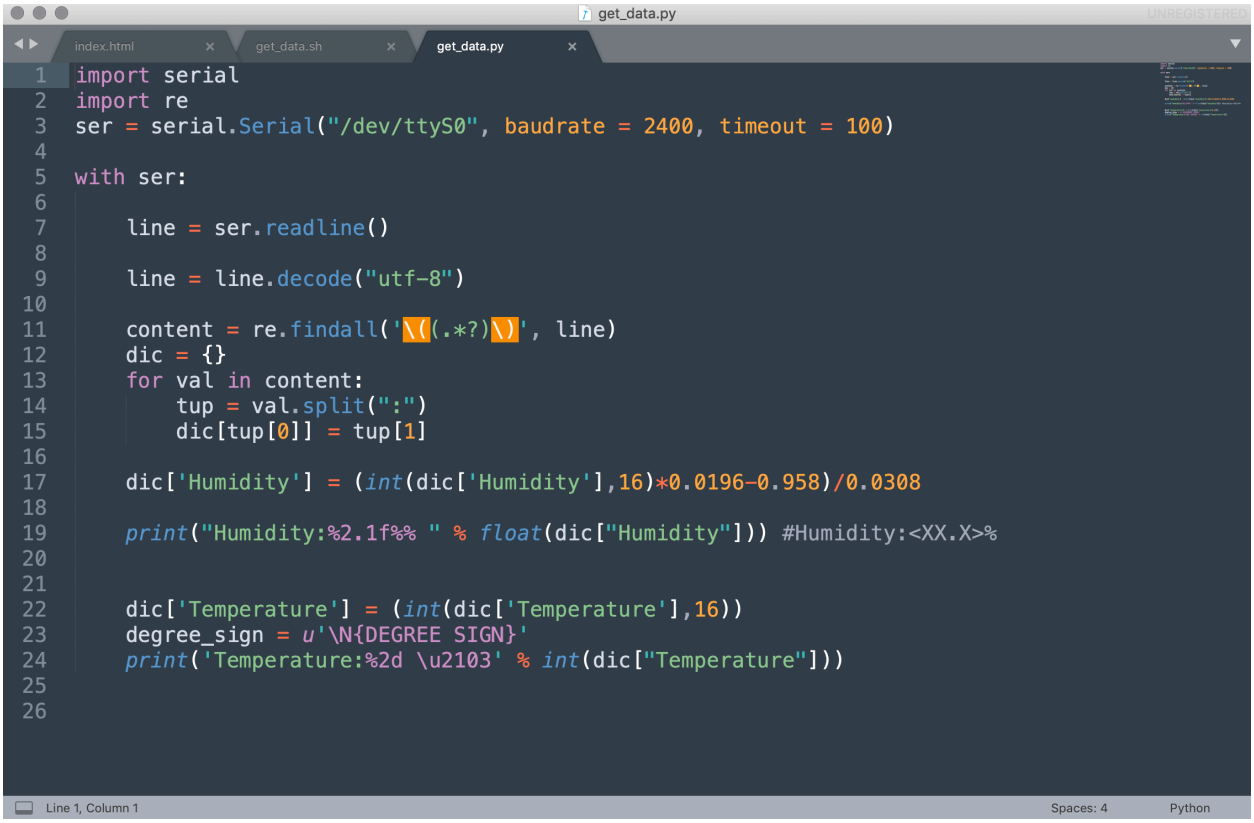
3. Raspberry Pi

The raspberry receives the data from MCU and displays on the local website. The hardware part is connecting and configuring the serial port on its GPIO headers, as the data is constantly sending out from monitor mode on embedded system. To do so, a python program is written to capture the keywords (diction) through the GPIO serial port ttyS0 and to convert the raw data (in hex) into decimal (Cesium & percentage).



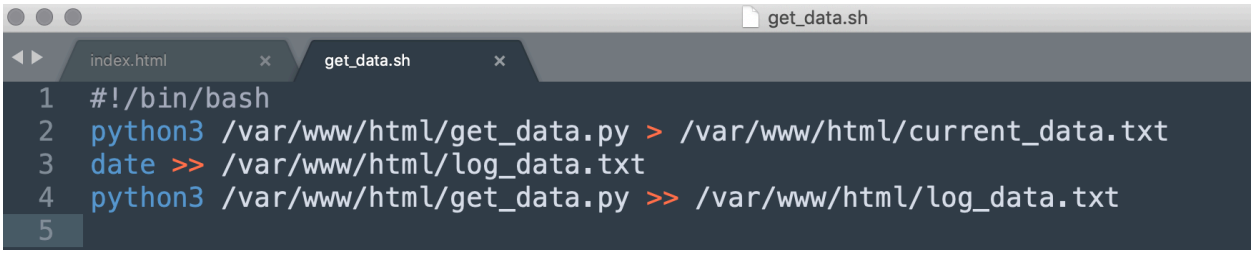
(Figure 2.3.1: the ttyS0 serial console)

With proper driver installed and pins (pin 8&10) connected, the terminal input ttyS0 (or ttyS1) can be found in /etc directory. This is the source where the python code would access from.



(Figure 2.3.2: the python program get_data.py)

The RPi must have python3 and PySerial library installed before running the program. The python program “get_data” is not only used to establish the serial communication, but also reads and processes the data. The RPi stores ASC II characters in a variable “line”, then transfers into another variable “content”. The “diction” splits the two words, converts the variables into integers before displaying them. Therefore, whenever this program is executed, it will take the next latest data from embedded system through the ttyS0 and print it to the terminal in form of “Humidity: XX.X% /n Temperature XX˚C”.



(Figure 2.3.3: the bash script get_data.sh)

In order to log the output data and save it into a log file, a bash script is created and runs periodically. The script “get_data.sh” calls the “get_data.py” program and save it into “current_data.txt” file (overwriting), as well as log it into “log_data.txt” with a time stamp.

```
For more information see the manual pages of crontab(5) and cron(8)

m h  dom mon dow   command
* * * * sh /var/www/html/get_data.sh
* * * * 1 >> /var/www/html/log_data.txt
ou have new mail in /var/mail/root
oot@raspberrypi:/home/pi#
```

(Figure 2.3.4: the crontab configuration)

In crontab file, the command is scheduled as being executed every minute.

log_data.txt — 已编辑

```
Humidity:51.0%
Temperature:21 ℃
Wed Mar 20 19:52:01 EDT 2019
Humidity:51.0%
Temperature:21 ℃
Wed Mar 20 19:53:02 EDT 2019
Humidity:60.5%
Temperature:22 ℃
Wed Mar 20 19:54:02 EDT 2019
Humidity:59.9%
Temperature:19 ℃
Wed Mar 20 19:55:03 EDT 2019
Humidity:59.3%
Temperature:22 ℃
Wed Mar 20 19:56:02 EDT 2019
Humidity:49.7%
Temperature:21 ℃
Wed Mar 20 19:56:06 EDT 2019
Humidity:49.7%
Temperature:21 ℃
Wed Mar 20 19:57:03 EDT 2019
Humidity:49.7%
Temperature:22 ℃
Wed Mar 20 19:58:03 EDT 2019
Humidity:59.3%
Temperature:19 ℃
Wed Mar 20 19:59:02 EDT 2019
Humidity:50.4%
Temperature:21 ℃
```

(Figure 2.3.5: the results in “log_data.txt”)

as the file records both temperature and humidity data every minute, the change of server room environment is shown in time domain.

Servers Setup

1. Web servers

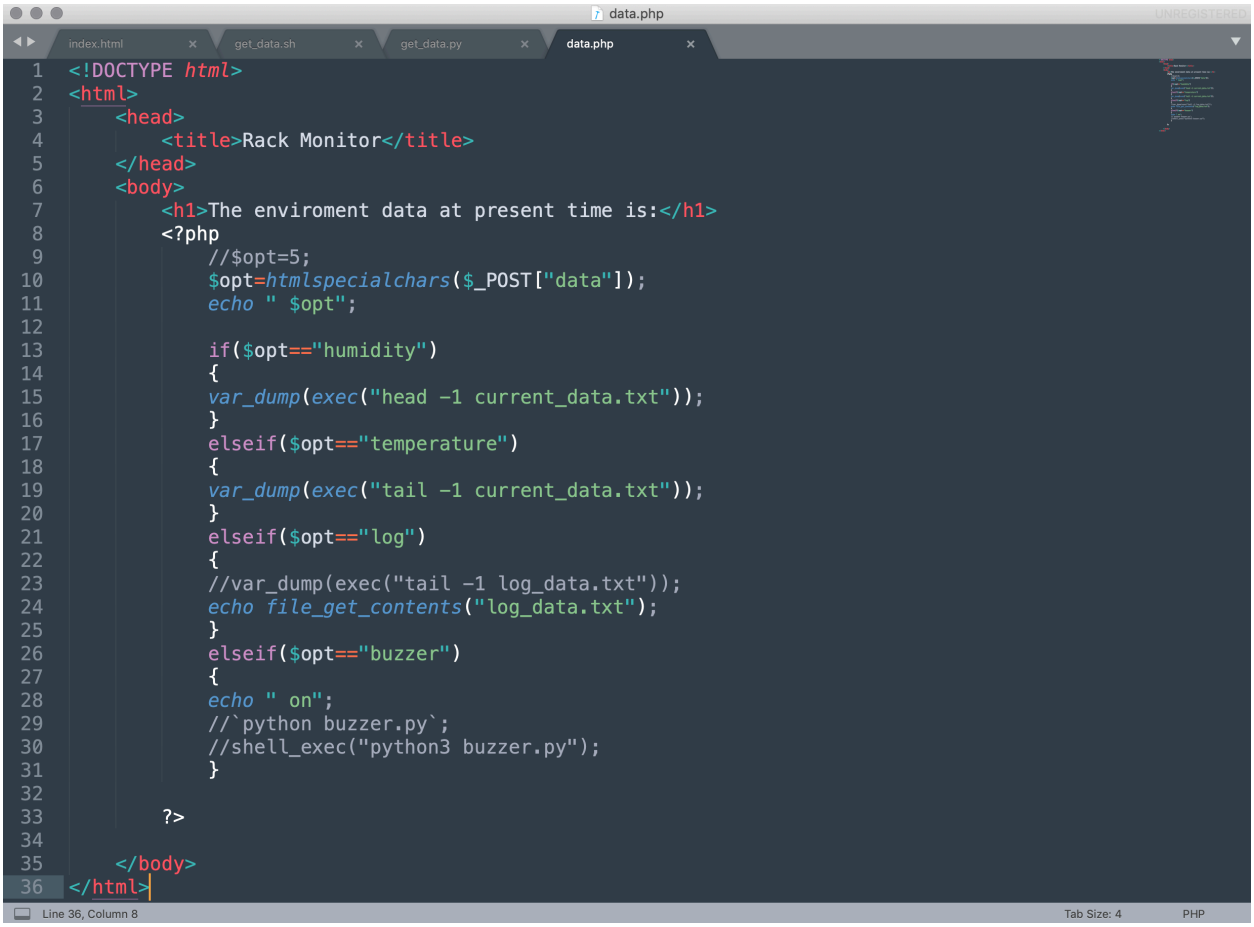
index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Rack Activity Monitor</title>
    </head>
        <style>
        .section{
            background:darkgray;
        }
        </style>

    <body>
    <H1>Pi server</H1>
    <form method="post" action="data.php">
    <fieldset>
        <label for="data">Humidity</label>
        <input type="radio" name="data" value="humidity" id="check_data"/><br>
        <label for="data">Temerature</label>
        <input type="radio" name="data" value="temperature" id="check_data"/><br>
        <label for="data">Environment log</label>
        <input type="radio" name="data" value="log" id="check_data"/><br>
        <label for="data">Allert</label>
        <input type="radio" name="data" value="buzzer" id="check_data"/><br>
        <input type="submit" value="submit"/>
    </fieldset>
    </form>
    </body>
</html>
```

Line 11, Column 1 Tab Size: 4 HTML

(Figure 3.1.1: the “index.html” on Rack1)

As for the interface to check the data from embedded system, an Apache web server is deployed on the RPi used for monitoring Rack1. The front-end (users) submits a form to the back-end (“data.php”), the PHP takes actions whether to retrieve the data, log or sends an instruction to activate the buzzer on the rack.

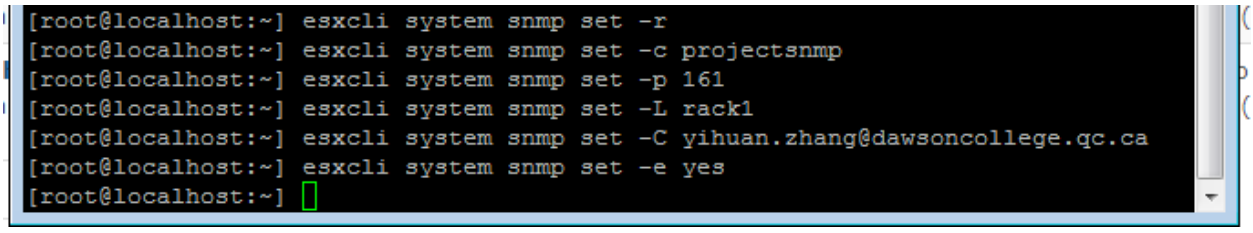


(Figure 3.1.2: the “data.php” on Rack1)

After receiving requests from HTML, the back-end uses “htmlspecialchars()” to filter out any HTML keywords or tags when retrieving the values from the users in order to prevent any malicious commands. Then the “data” stores into a variable “$opt”, different actions under different conditions are taken according to its value. The value stored in a variable “$opt” then would use “if()” condition to take actions according to the input value. The form should only use POST method when submitting form and its value has to be either “humidity”, “temperature”, “log” or “buzzer”.

2. ESXi installation

On the server Dell R720, the VMware ESXi is installed as the hypervisor. There are three virtual machines on it and all of them are running Windows server 2016 (desktop experience). The ESXi has been assigned with an IPv4 address 10.210.20.100/16, and with SSH and SNMP enabled.



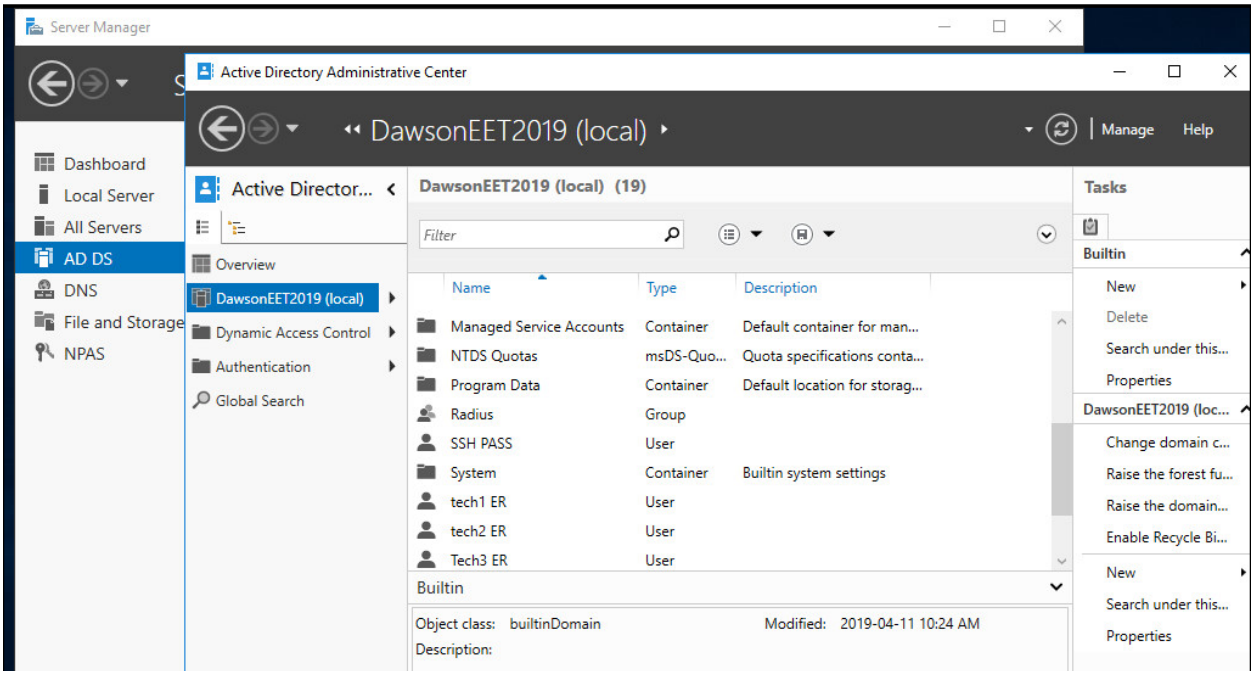
(figure 3.2.1: ESXi SNMP configuration)

Since most of the functions on ESXi can only be configured through command line, administrators can access with it through terminal (Linux) or emulator (Putty). After the SSH is enabled, on ESXi 6.0 version, the command "`#esxcli system snmp set –[options]`" is used to configure the SNMP options.

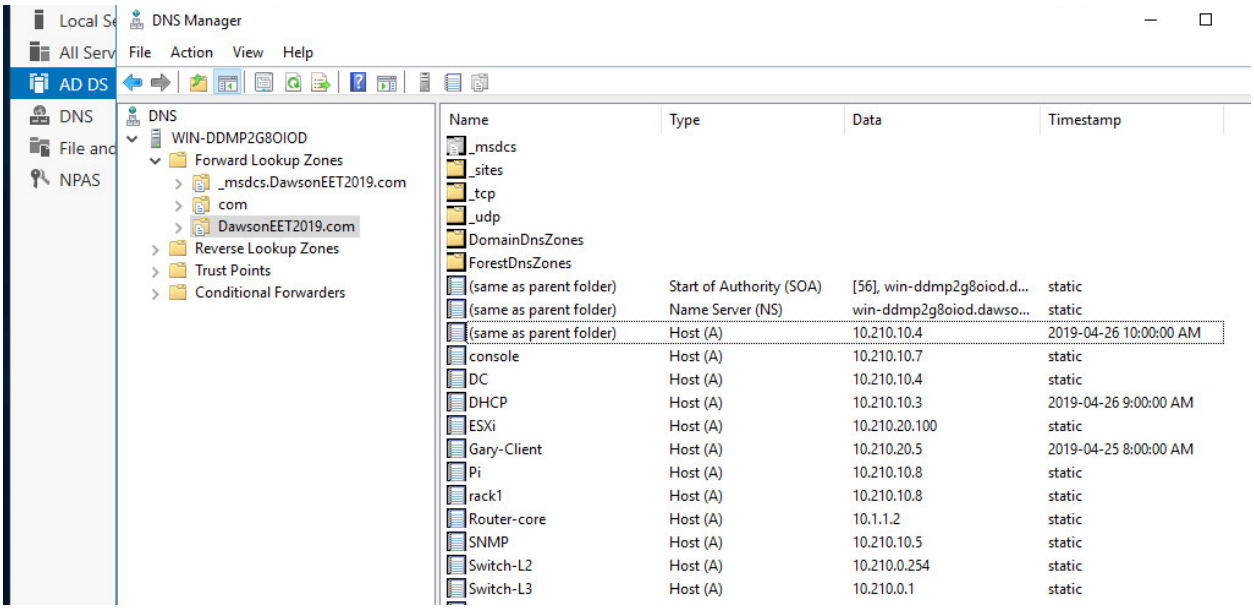
3. Domain Controller, DNS and Radius installation

The domain controller runs on the Dell R720 server, this server has AD DS, DNS, Radius and SNMP installed. As the core of the domain "DawsonEET2019.com", this server stores the data including user information and DNS lookup table. Users "technician1" and "technician2"

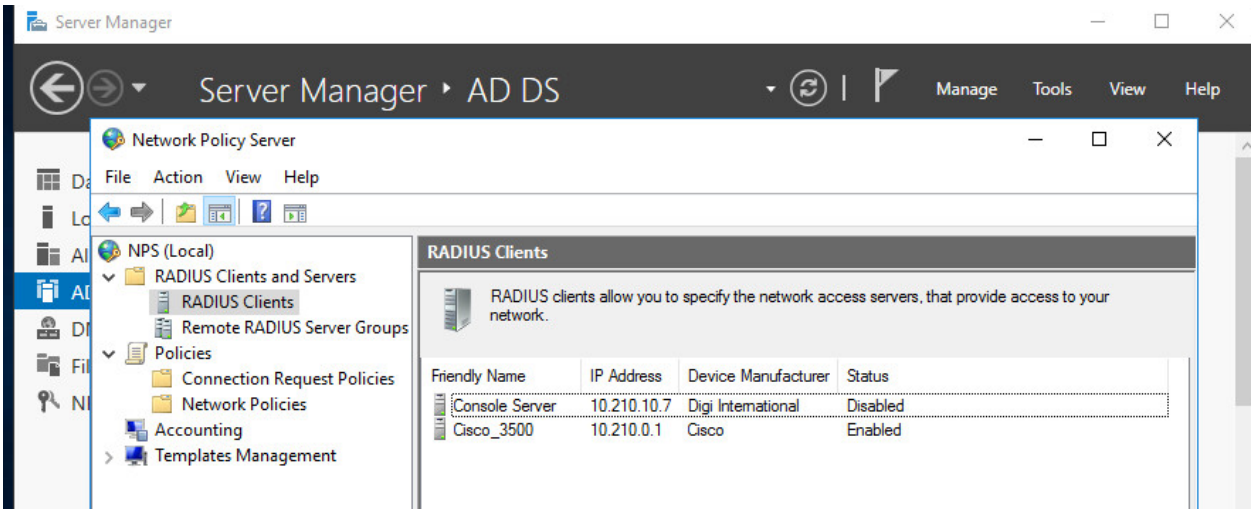
with administrative permission are created and has the Radius service enabled.



(Figure 3.3.1: the users and groups on Domain Controller)

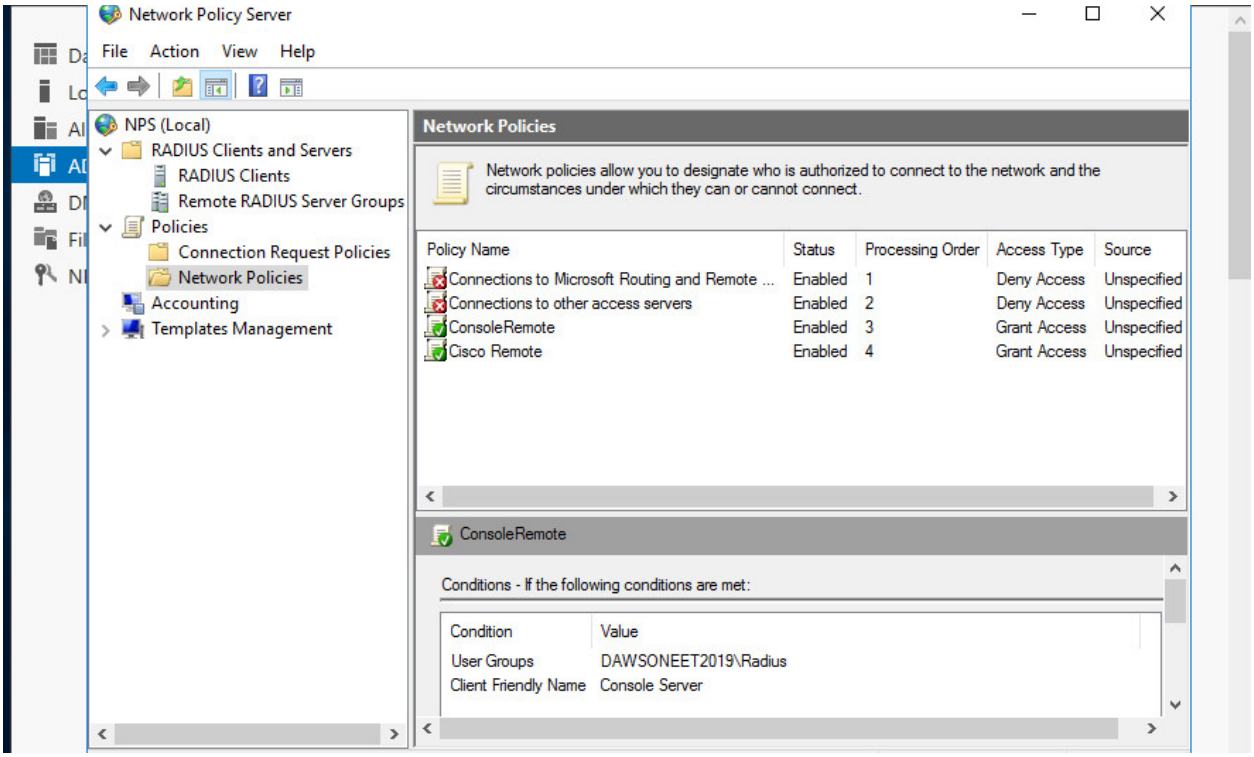


(Figure 3.3.2: the DNS database)



(Figure 3.3.3: the RADIUS Clients list)

On windows server, the Radius service uses local accounts and shared secret “Class!23” to perform authentication and encryption. To allow users being authenticated through the network, create a user group (Radius). Then in "Networking Policy Server” register server in Active Directory and create new RADIUS Clients and Policies.

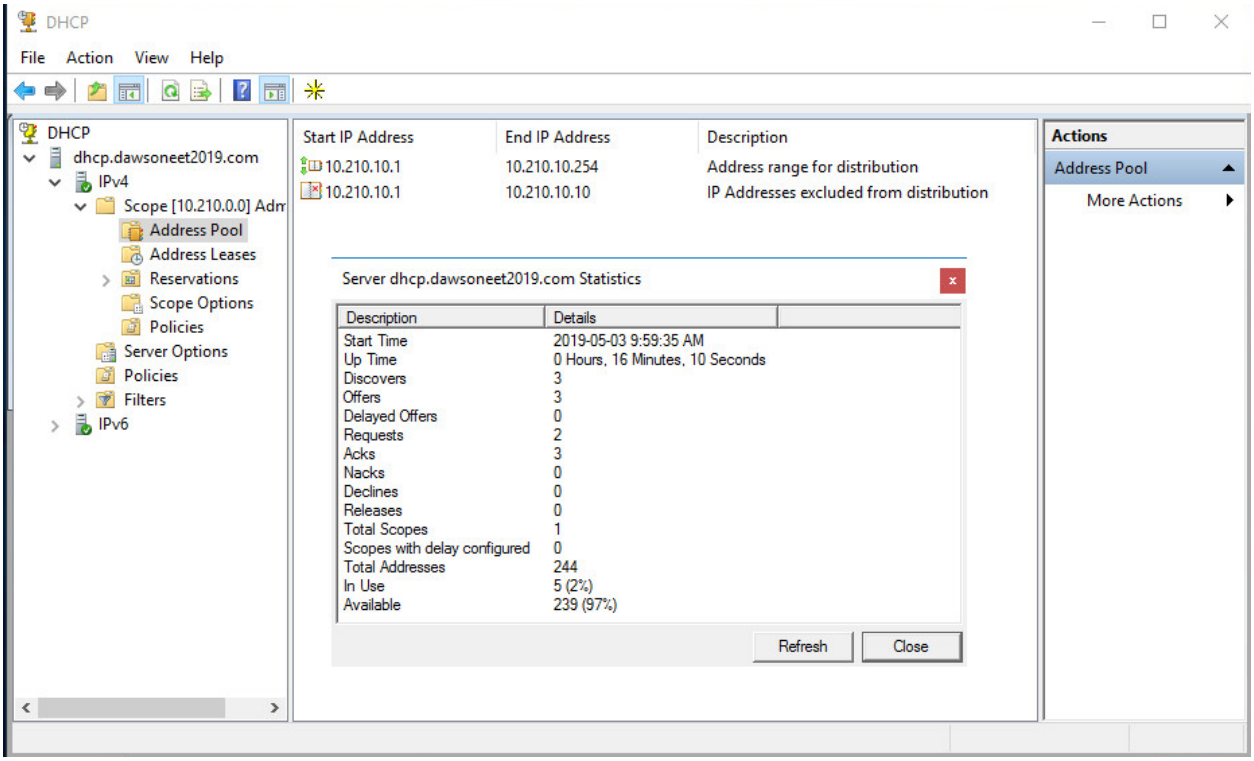


(Figure 3.3.4: the Network Policies list)

The basic logic on windows server is that we add the devices that require to be authenticated through an external AAA server (with “Friendly name”, “Address”, and “shared secret” specified), and in Policies we configure the access permission, authentication methods, and manufacturer (vendor).

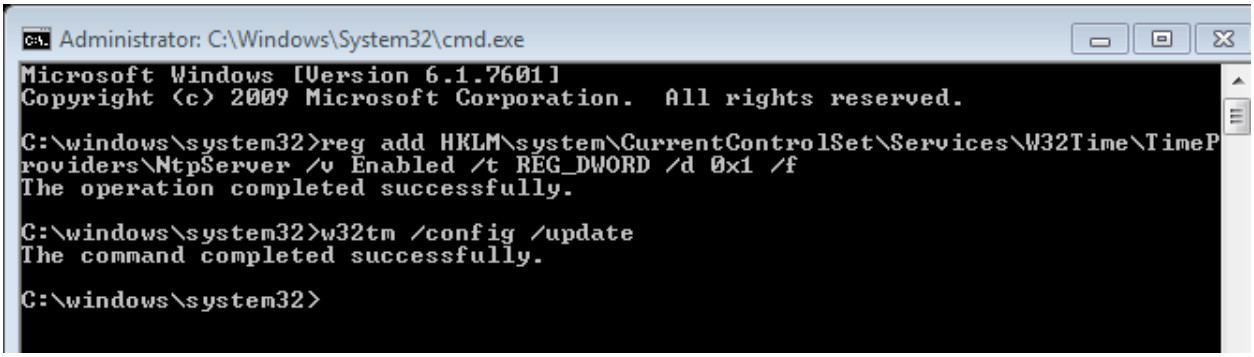
4. DHCP server installation

DHCP server is a windows 2016 server runs on Dell R720 server. After the DHCP role has been installed, the DHCP server is assigned in the domain with a static IP address 10.210.10.3/16, a new scope under IPv4 is configured with range 10.210.10.10~254.



(Figure 3.4.1: the DHCP pool statistics)

5. NTP server installation



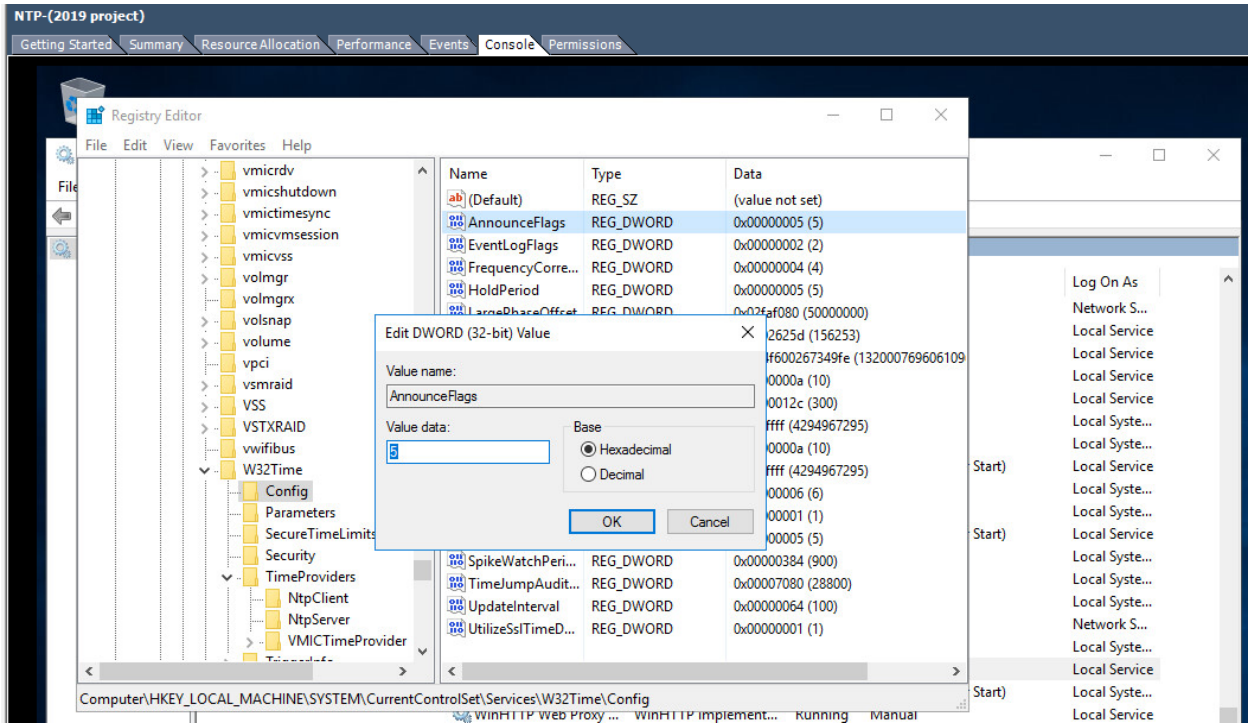
(Figure 3.5.1: using "`#w32tm`" to check or configure the flag)

The NTP server provides a universal time stamp for network devices that has external syslog and SNMP services enabled. Any centralized administrative tool requires the synchronized time or otherwise, the history logs would be in wrong sequence and meaningless. To configure the NTP server/client on a windows, we have to change the flag in OS registry.

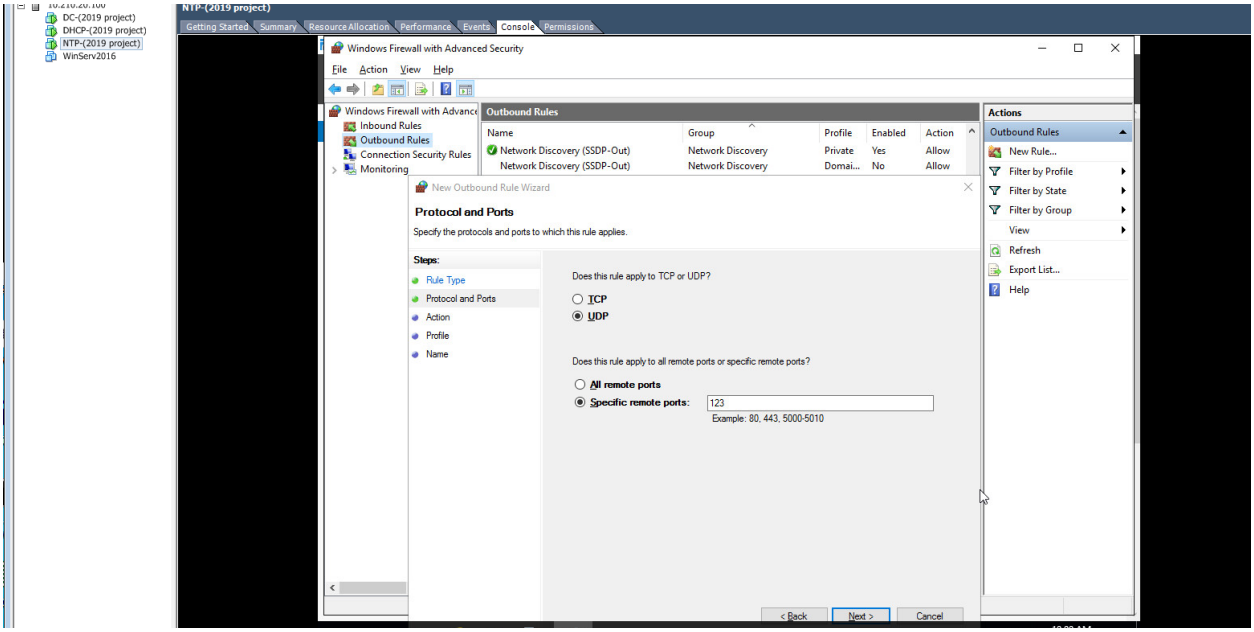
```
C:\Users\Administrator>w32tm /query /configuration
[Configuration]

EventLogFlags: 2 (Local)
AnnounceFlags: 10 (Local)
TimeJumpAuditOffset: 28800 (Local)
MinPollInterval: 6 (Local)
MaxPollInterval: 10 (Local)
MaxNegPhaseCorrection: 4294967295 (Local)
MaxPosPhaseCorrection: 4294967295 (Local)
MaxAllowedPhaseOffset: 300 (Local)

FrequencyCorrectRate: 4 (Local)
PollAdjustFactor: 5 (Local)
LargePhaseOffset: 50000000 (Local)
SpikeWatchPeriod: 900 (Local)
LocalClockDispersion: 10 (Local)
HoldPeriod: 5 (Local)
PhaseCorrectRate: 1 (Local)
UpdateInterval: 100 (Local)
```

(Figure 3.5.2: the command line of turning on NTP server)

Consequently, the W32Time parameters can also be configured in the Registry Editor. The absolute directory address is “Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config”:



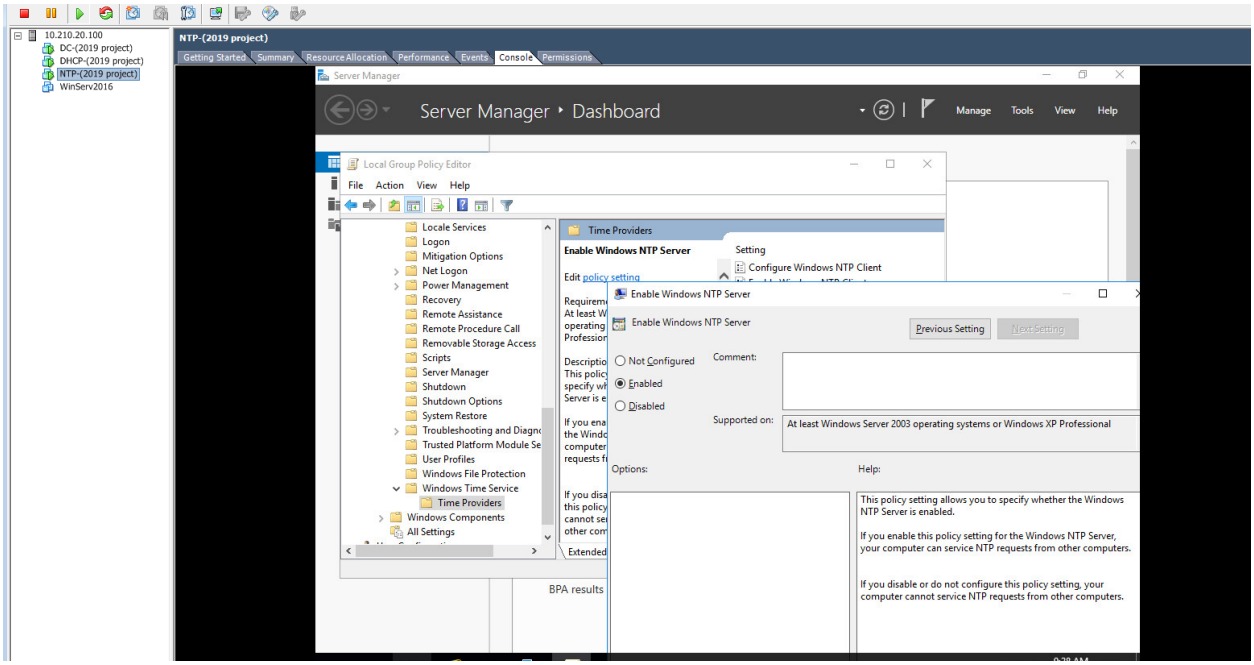
(Figure 3.5.3: configuring the NTP flag in registry editor)



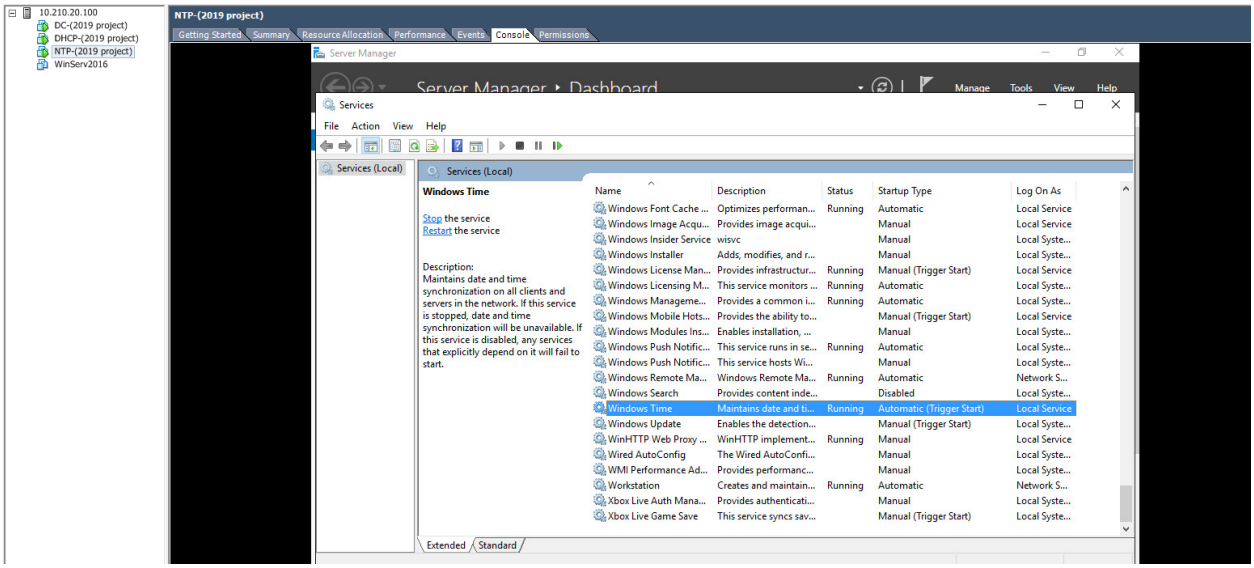
(Figure 3.5.4: adding new rule to allow port for NTP)

The Network Time Protocol uses port 123, and by default the windows firewall blocks it for security reasons. This is configured under both Inbound and Outbound Rules.

After allowing UDP port 123 in windows firewall, Enable or restart the server under "Local Group Policy Editor" to start the NTP service:

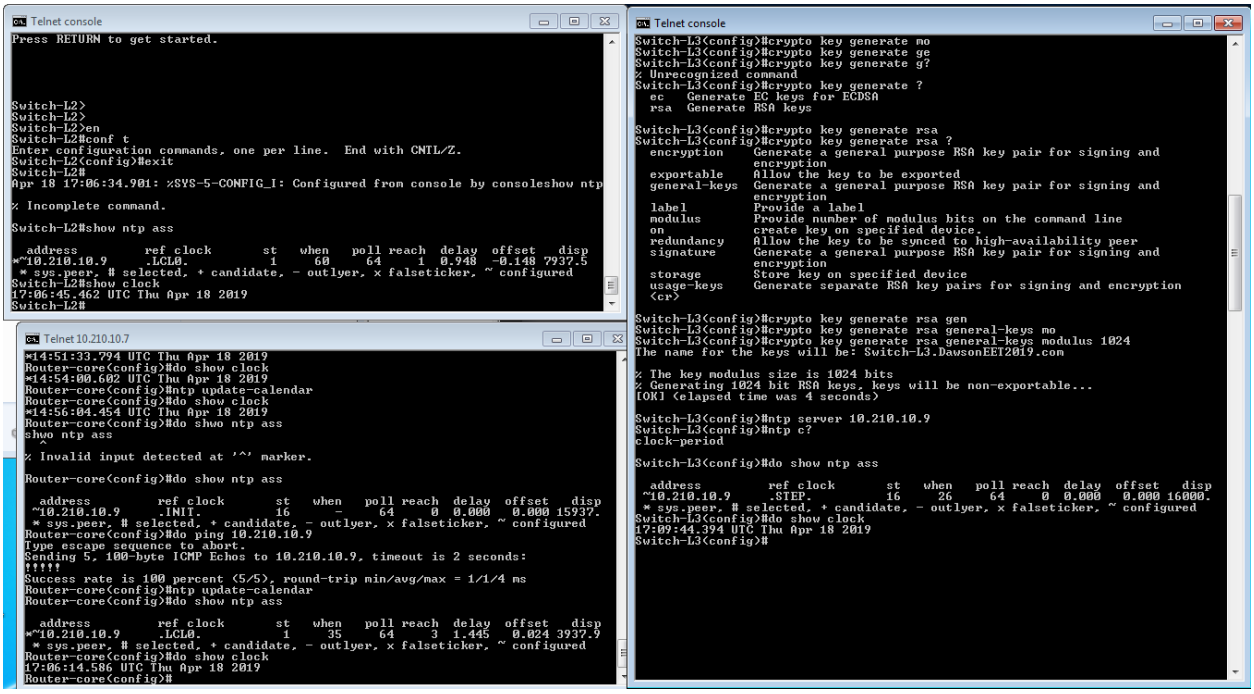


(Figure 3.5.5: enabling the NTP server)

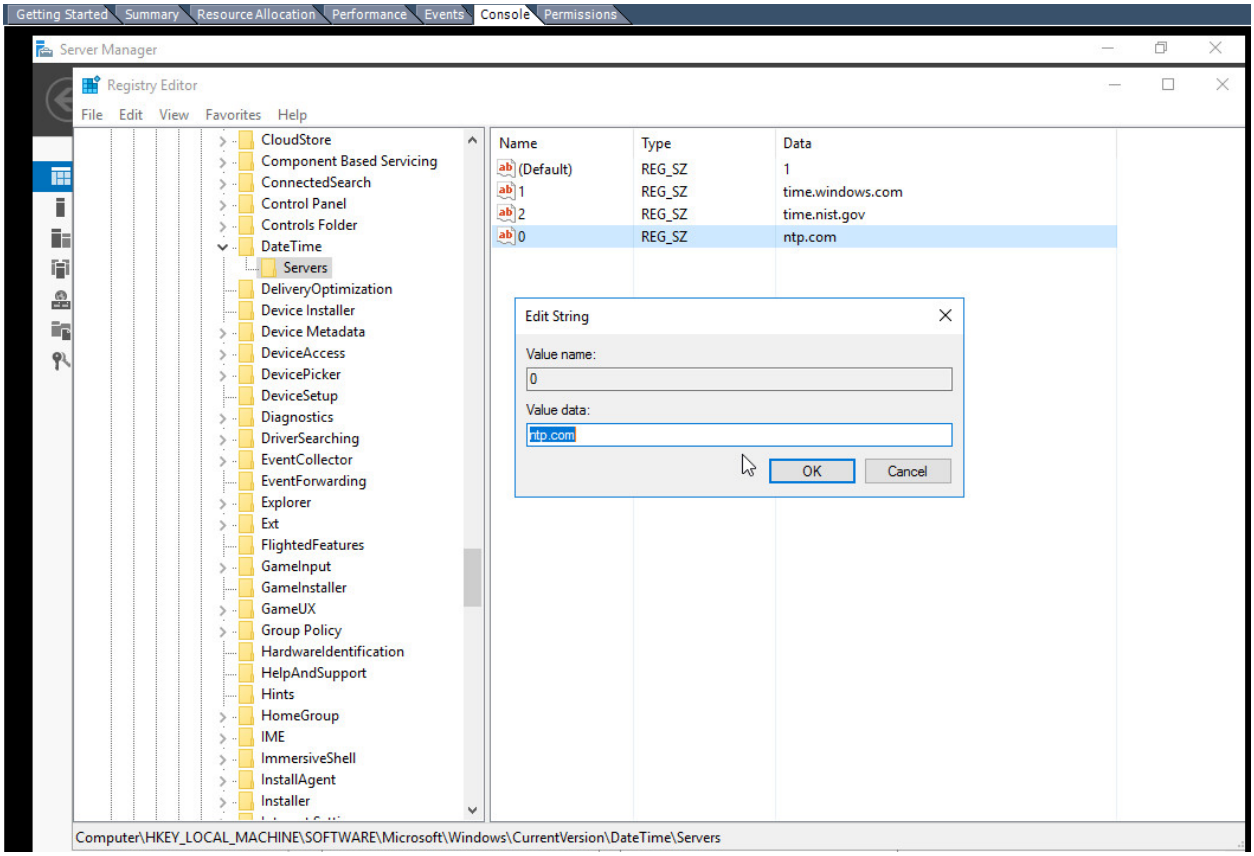


(Figure 3.5.6: checking if the NTP service is running)

The NTP service can also be activated by installing the software, the one I have tested is called "Meinberg NTP", it configures the NTP flag and is compatible with Cisco IOS devices. To see more detail about the SNMP configuration on Cisco IOS devices, please check the IOS configuration part.



(Figure 3.5.7: all the IOS devices are synchronized)



(Figure 3.5.8: configuring another windows as NTP client)

6. SNMP server installation

The SNMP manager uses an open source software *LibreNMS* (https://www.librenms.org) based on CentOS platform. The LibreNMS is a network administrator software that provides various administrative tools, it supports all SNMP version 1, 2c and 3, has advantages in terms of compatibility, authentication (supports active directory, Radius, HTTP and local database), and graphical tools (web UI and mobile application).

To install a SNMP server, I implemented a CentOS 7 server in VirtualBox, configured NIC as bridged adapter and 10.210.10.5/16 network address. Then use the command below to install required packages after adding EPEL repository and update:

```
#yum install [nginx, php72w, php72w-cli, php72w-common, php72w-curl, php72w-fpm,
php72w-gd, php72w-mbstring, php72w-mysqlnd, php72w-process, php72w-snmp, php72w-xml,
php72w-zip, mariadb-server, mariadb, git, ImageMagick, iwhois, cronie, composer,
mysql-python, net-snmp, python-memcached} -y
```

Then download the software LibreNMS:

```
#yum composer create-project –no-dev –keep-vcs librenms/librenms librenms dev-master
```

Creating a user for the SNMP manager:

```
#useradd librenms -d /opt/librenms –M -r
#usermod -a -G librenms nginx
```

Staring and configuring the local database MySQL:

```
#systemctl start mariadb
#mysql -u root
```

adding “innodb_file_per_table=1” and “lower_case_table_names=0” in /etc/my.conf file. And restart mariadb.

configure the "/etc/php-fpm.d/www.conf" as below and restart php-fpm:

```
[root@localhost ~]# vi /etc/php-fpm.d/www.conf
; Start a new pool named 'www'.
[www]

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's group
;       will be used.
; RPM: apache Choosed to be able to access some dir as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = apache

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socket to a specific IPv4 address on
;                            a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 address on
;                            a specific port;
;   'port'                 - to listen on a TCP socket to all addresses
;                            (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /var/run/php-fpm/php7.2-fpm.sock

; Set listen(2) backlog.
; Default Value: 511 (-1 on FreeBSD and OpenBSD)
;listen.backlog = 511

; Set permissions for unix socket, if one is used. In Linux, read/write
; permissions must be set in order to allow connections from a web server. Many
; BSD-derived systems allow connections regardless of permissions.
; Default Values: user and group are set as the running user
;                 mode is set to 0660
listen.owner = nginx
listen.group = nginx
listen.mode = 0660
; When POSIX Access Control Lists are supported you can set them using
; these options, value is a comma separated list of user/group names.
; When set, listen.owner and listen.group are ignored
;listen.acl_users =
;listen.acl_groups =

; List of addresses (IPv4/IPv6) of FastCGI clients which are allowed to connect.
; Equivalent to the FCGI_WEB_SERVER_ADDRS environment variable in the original
; PHP FCGI (5.2.2+). Makes sense only with a tcp listening socket. Each address
; must be separated by a comma. If this value is left blank, connections will be
; accepted from any ip address.
; Default Value: any
listen.allowed_clients = 127.0.0.1

; Specify the nice(2) priority to apply to the pool processes (only if set)
; The value can vary from -19 (highest priority) to 20 (lower priority)
; Note: - It will only work if the FPM master process is launched as root
;       - The pool processes will inherit the master process priority
;         unless it specified otherwise
; Default Value: no set
; process.priority = -19

; Choose how the process manager will control the number of child processes.
; Possible Values:
;   static  - a fixed number (pm.max_children) of child processes;
;   dynamic - the number of child processes are set dynamically based on the
;             following directives. With this process management, there will be
;             always at least 1 children.
;             pm.max_children      - the maximum number of children that can
```

configure the Nginx file "/etc/nginx/conf.d/librenms.conf" as below and restart it:

```
[root@localhost ~]# vi /etc/nginx/conf.d/librenms.conf
server {
 listen      80;
 server_name librenms.example.com;
 root        /opt/librenms/html;
 index       index.php;

 charset utf-8;
 gzip on;
 gzip_types text/css application/javascript text/javascript application/x-javascript image/svg+xml text/plain text/
xsd text/xsl text/xml image/x-icon;
 location / {
  try_files $uri $uri/ /index.php?$query_string;
 }
 location /api/v0 {
  try_files $uri $uri/ /api_v0.php?$query_string;
 }
 location ~ \.php {
  include fastcgi.conf;
  fastcgi_split_path_info ^(.+\.php)(/.+)$;
  fastcgi_pass unix:/var/run/php-fpm/php7.2-fpm.sock;
 }
 location ~ /\.ht {
  deny all;
 }
}
```

Configuring the SNMP server by simply replacing "/etc/snmp/snmpd.conf" with "/librenms/snmpd/snmpd.conf" and restart it.

forwarding the port number 80 and 443 and allowing access through firewall:

```
#firewall-cmd --zone public --add-service http
#firewall-cmd --permanent --zone  public --add-service http
#firewall-cmd --zone public --add-service https
#firewall-cmd --permanent --zone  public --add-service https
```

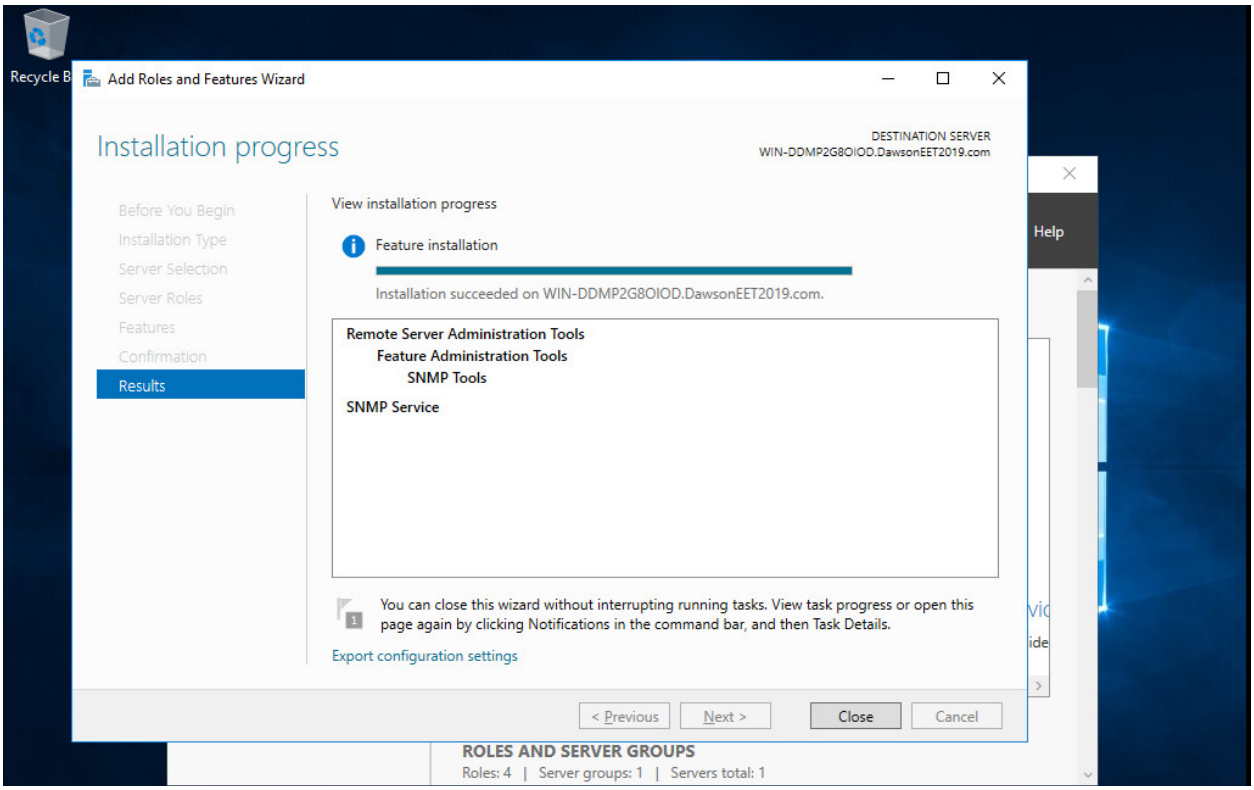
On Cisco network devices like router and switches, issue the commands below to add the SNMP server:

```
Router-core(config)#snmp-server community projectsnmp ro SNMP_ACL
Router-core(config)#snmp-server location rack1
Router-core(config)#snmp-server contact yihuan.zhang@dawsoncollege.qc.ca
Router-core(config)#snmp-server host 10.210.10.5 version 2c projectsnmp
Router-core(config)#snmp-server enable traps
Router-core(config)#ip access-list standard SNMP_ACL
Router-core(config-std-nacl)#permit 10.210.10.5
```

After the above configuration, you would see the result in IOS using "`Router-core(config)#do show snmp`" command:

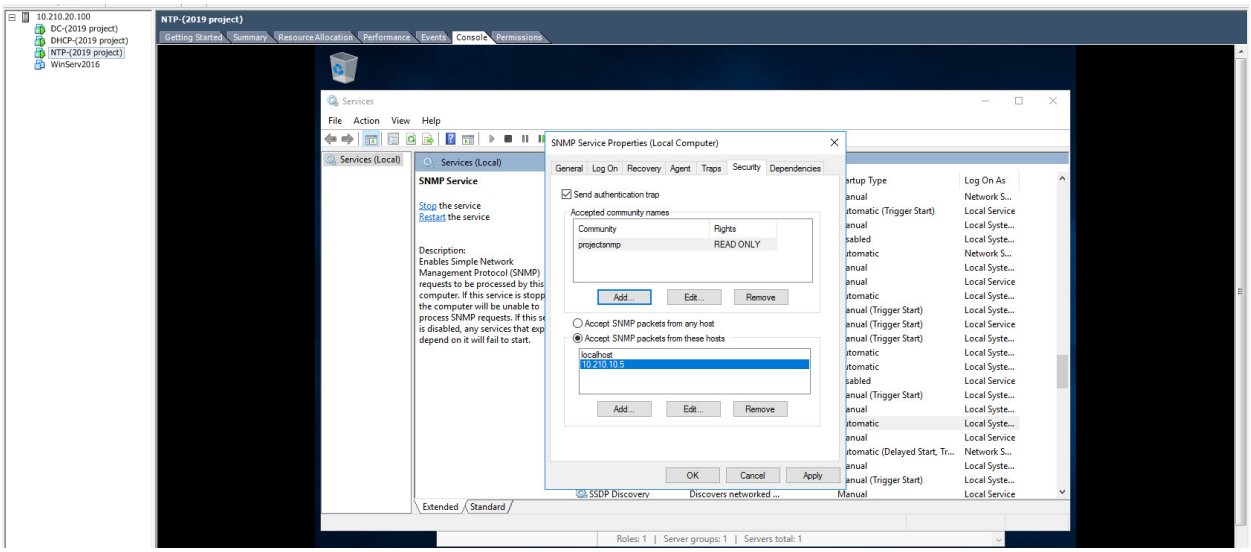
```
Chassis: FHK1446F14K
Contact: yihuan.zhang@dawsoncollege.qc.ca
Location: rack1
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
0 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs
SNMP Dispatcher:
   queue 0/75 (current/max), 0 dropped
SNMP Engine:
   queue 0/1000 (current/max), 0 dropped
SNMP logging: enabled
```

On windows servers, add the SNMP feature in order to activate its SNMP function and send the traps to the SNMP server



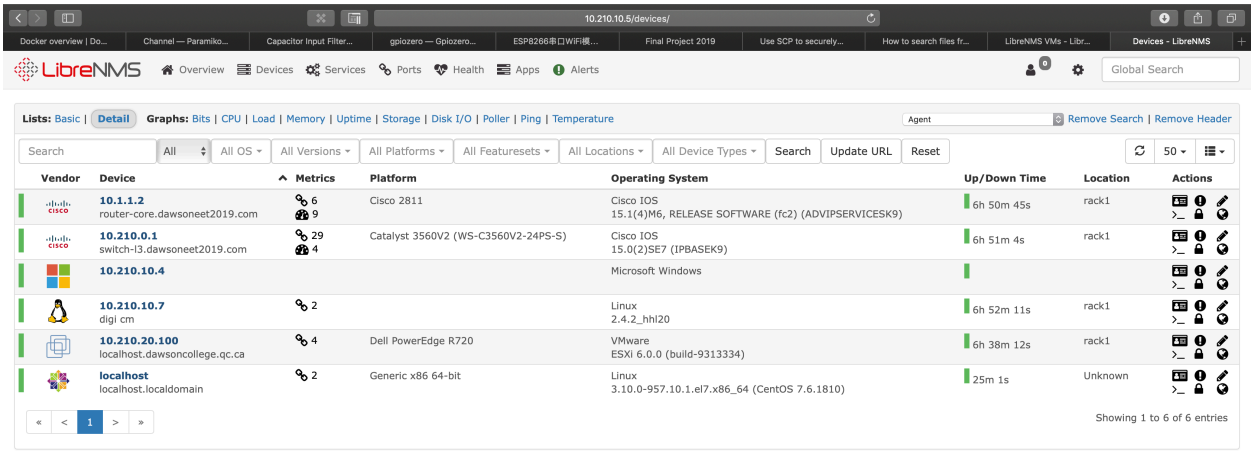
(Figure 3.6.1: the SNMP feature installing)

and then configure the community string, host address in “SNMP Service Properties -> Security”:



(Figure 3.6.1: configuring another windows as NTP client)

To add the SNMP nodes through the web UI, login into the website and click “add new server” from “Devices” on top, filling the parameters (device hostname, version 2c, port 161, community string “projectsnmp”) . After the SNMP configured devices has been added, the SNMP manager starts polling the SNMP traps and other informations from agents’ MIB.



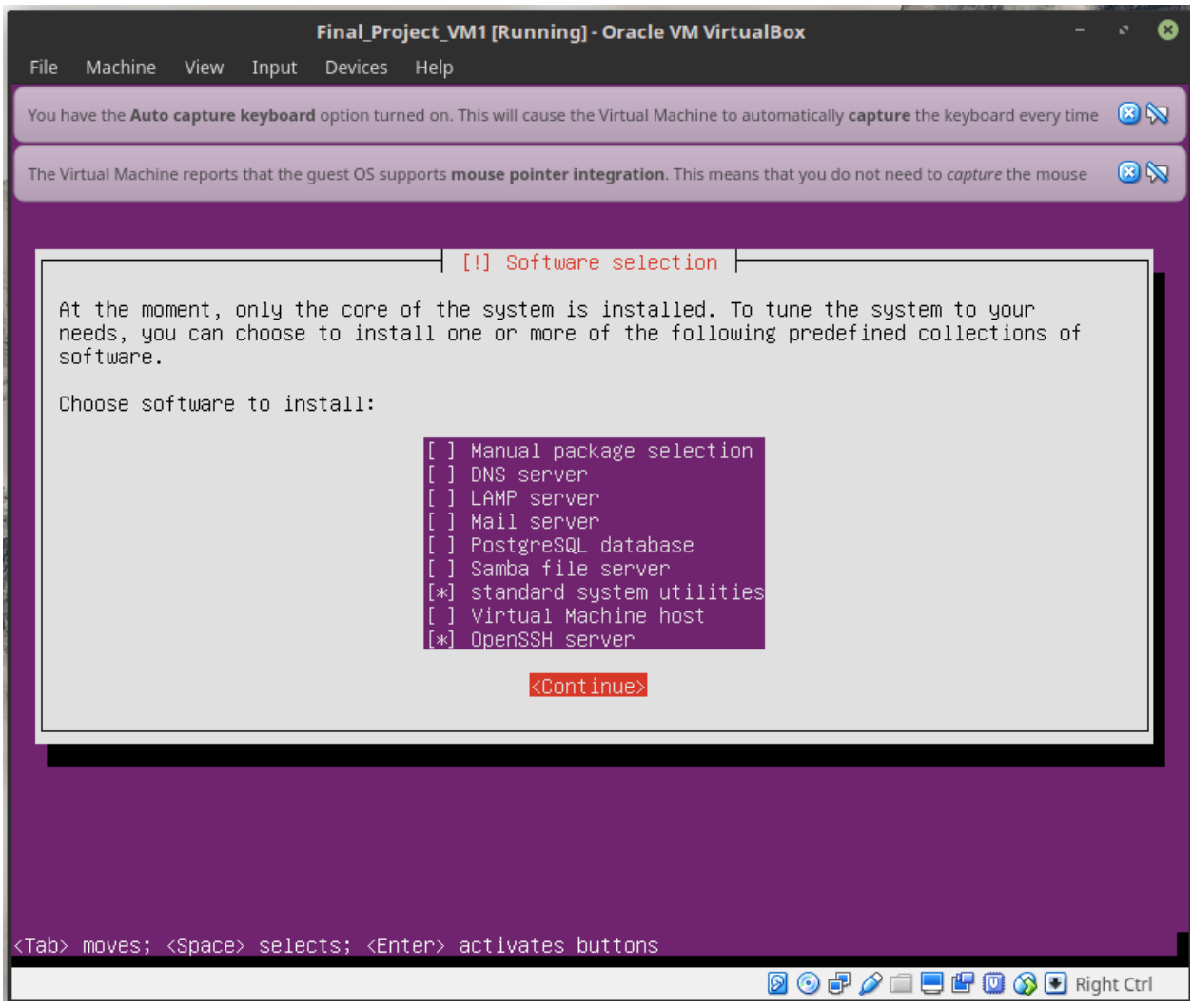
(Figure 3.6.2: the managed SNMP nodes)

I decided to uses SNMP v2c in this scenario for compatibility reasons since. The project contains different OS and legacy network devices and some of them are not able to configure the SNMP version (windows 2016).

7. Web servers installation

Besides the RPi and SNMP server, there is one more web server that are implemented in this project: the main web server (Nginx) that contains a navigation page and redirect users to different places (resources, tools, references).

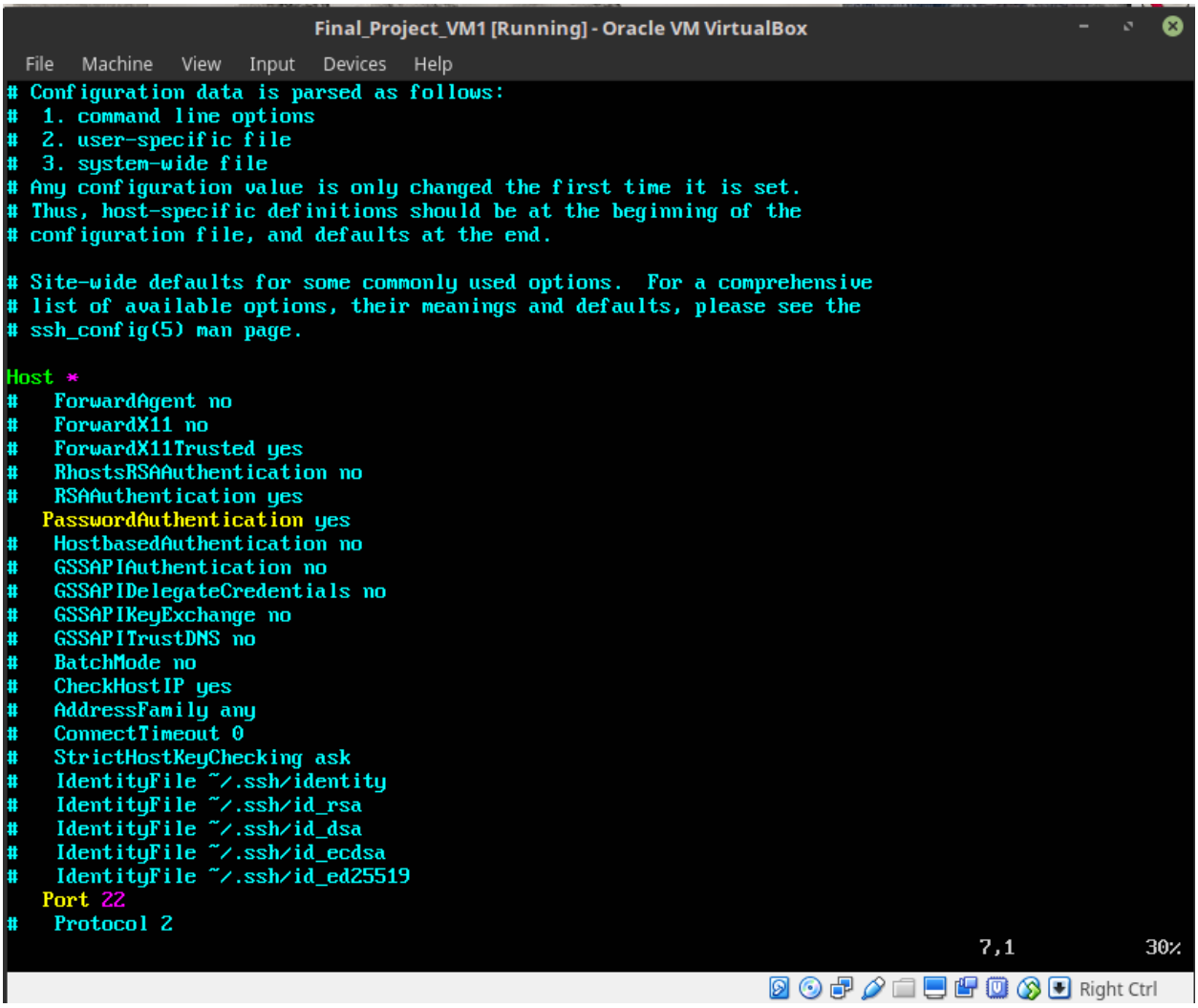
The main web server is also responsible for displaying the documentation (IP address, network topology), redirecting pages to different servers, and executing the program to check the network connectivity. A host OS based on Linux that has LAMP, NFS (SAMBA) and python installed needs to be created, these services and tools contribute to the administrative framework of the network devices.



(figure 3.7.1: Ubuntu 16.04 installation wizard)

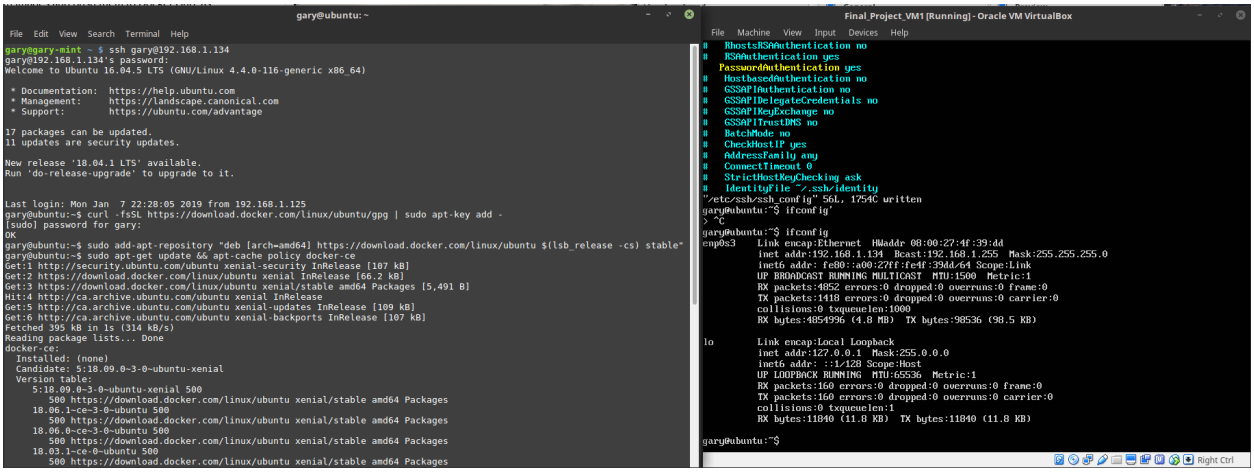
A VM with 4GB memory and 10GB disk space has been created. I choose to use Ubuntu as the host OS mainly because of its compatibility, accessibility and IDE. Moreover, as I am planning

to deploy the applications under the containers of the docker, applications are limited by the fundamental architecture of the guest OS. Therefore, the guest OS only requires minimum tools to operate and softwares will be deployed onto their respective container later. After the installation, a restore point (VM clone) is created for back up purpose.



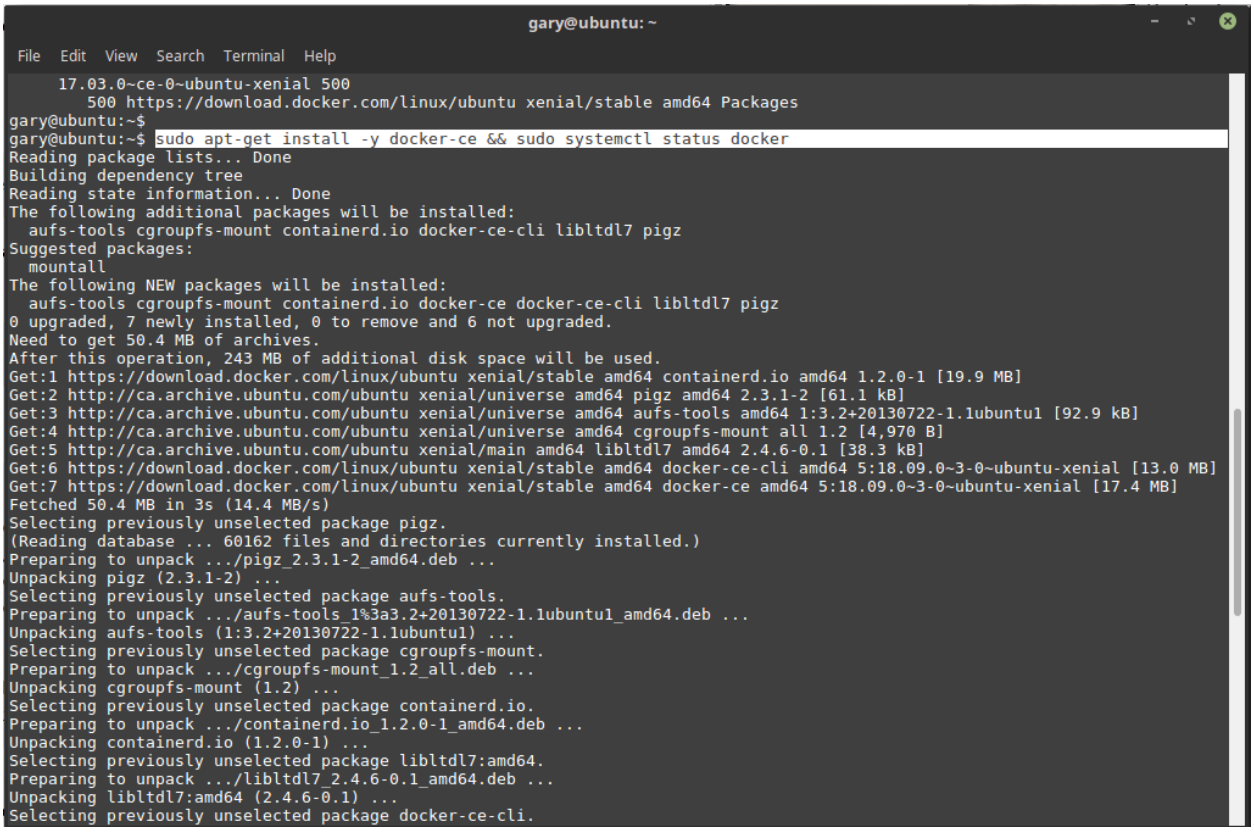
(figure 3.7.2: modification of the ssh_config file)

As the ssh has been installed in previous step, the second thing is to configure and enable the ssh service in order to activate the remote login and secure copy (SCP). I simply uncomment the two lines above (with root permission) and checked the interface IP address to test the connectivity.



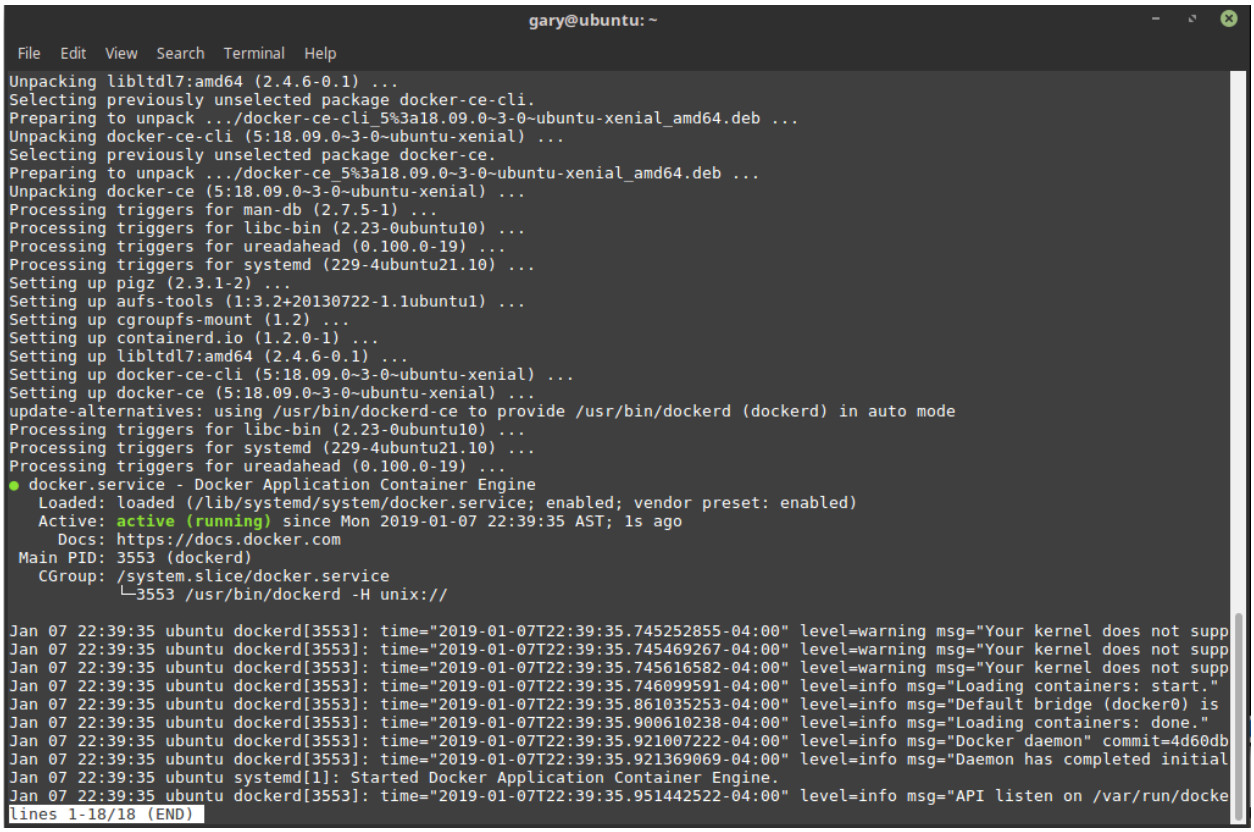
(figure 3.7.3: adding the repository for docker-ce)

Through the SSH connection, a repository from docker.com is added preparing for the docker installation.



(figure 3.7.4: installing docker engine)

From the repository we just added in previous step, the docker-ce is able to be installed using apt-get command.



(figure 3.7.5: checking the docker service)

Thus far, we can see that the docker engine has been installed and the docker daemon is in active status. Inside the docker, we are going to implement a web server using Nginx. The tutorial I referred is from DigitalOcean (https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04), and there 5 steps in general to configure a Nginx based web server in a docker.

Step 1: creating a new container “navweb” with its 80 port binds to the 80 port on host:

```
#docker run –p 80:80 ––name navweb –i –t ubuntu /bin/bash
```

Step 2: install Nginx, vim (if necessary):

```
#apt–get install –y nginx vim
```

Step 3: create a static tested web page using HTML:

```
#vi /var/www/html/index.html
```

Step 4: modify Nginx configuration file and enable it:

(figure 3.7.6: the "/etc/nginx/sites-enabled/default" file)

```
#vi /etc/nginx/sites-enabled/default
#nginx
```

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm_;
```

Step 5: binding container port to the host (host port : container port):

```
#docker ps //show container
#docker port navweb //show container port number
#docker top navweb //show container process
```

To copy a file to the container use the "`#docker cp`" command:

```
gary@ubuntu:~$ docker cp ?
"docker cp" requires exactly 2 arguments.
See 'docker cp --help'.

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem
gary@ubuntu:~$ sudo scp -rp yihuanzhang@10.210.10.20:code/html/nav.react-china.org-master/* .
Password:
404.html                                                    100% 4216     4.1KB/s   00:00
README.md                                                   100%  264     0.3KB/s   00:00
address.html                                                100% 2422     2.4KB/s   00:00
ie.css                                                      100%  200     0.2KB/s   00:00
DESA Logo(◆ ◆ ).png                                         100%  258KB 258.2KB/s   00:00
logo.png                                                    100%   62KB  62.5KB/s   00:00
reactjs.png                                                 100%   25KB  25.2KB/s   00:00
index.html                                                  100% 6914     6.8KB/s   00:00
layout.css                                                  100% 2390     2.3KB/s   00:00
topology.html                                               100%  155     0.2KB/s   00:00
gary@ubuntu:~$ ls
404.html  address.html  ie.css  img  index.html  layout.css  README.md  topology.html
gary@ubuntu:~$ docker cp topology.html ed15411dbfe1:/var/www/html/topology.html
gary@ubuntu:~$ docker cp address.html ed15411dbfe1:/var/www/html/address.html
gary@ubuntu:~$ docker cp -a img ed15411dbfe1:/var/www/html/img/
gary@ubuntu:~$ docker cp layout.css ed15411dbfe1:/var/www/html/layout.css
gary@ubuntu:~$ docker cp ie.css ed15411dbfe1:/var/www/html/ie.css
gary@ubuntu:~$ docker cp 404.html ed15411dbfe1:/var/www/html/404.html
gary@ubuntu:~$ docker cp index.html ed15411dbfe1:/var/www/html/index.html
gary@ubuntu:~$ _
```

*Python scripts

Install python on Ubuntu server:

```
#sudo apt-get install python
```

Download the python script to install pip:

```
#curl “https://bootstrap.pypa.io/get-pip.py" -o “get-pip.py”
#python get-pip.py
```

Install python library paramiko:

```
#pip install paramiko
```

The objective of the python scripts is to use the SSH function in “Paramiko” library to login into each devices and check or configure them at the same time. With these scripts, the administrators do not have to manually log into each device and check or configure network devices one by one. Instead, the network administrator only need to run a program and all the relative information would be retrieved and centralized in his terminal. Besides, the python scripts can also embedded into a PHP web server so that the above procedures can be achieved from a web UI.

show-int-br.py

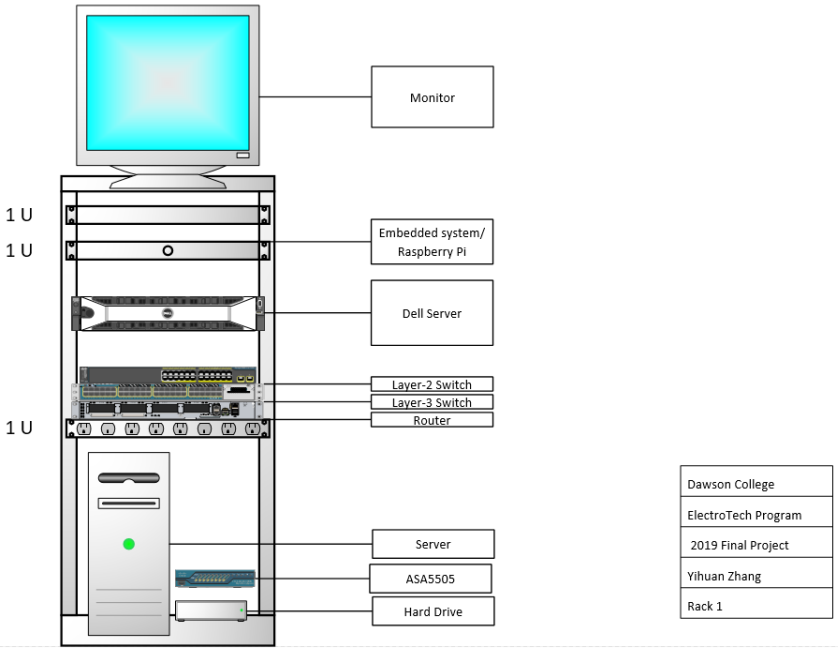
```
#!/usr/bin/env python
########################################
# Project: NetWatch (2019 Final Project)#
# Network Block – python script         #
# Author: Gary Zhang (1630783)          #
# Dawson College – Electrotech Program  #
# Required Library: paramiko            #
# Date: 2019.4.30                       #
########################################

import paramiko
import getpass
import time
print "interface checking..."
ip_router = "Router-core"   #the router
#ip_router = "10.1.1.2"   #if DNS disabled
ip_switch_l2 = "Switch-L2"
#ip_switch_l2 = "10.210.10.254"
ip_switch_l3 = "Switch-L3"
#ip_switch_l3 = "10.210.0.1"
ip_console = "console"
#ip_console = "10.210.10.7"
ip_asa = "asa"
#ip_asa = "192.168.1.2"

#admin account variable:
username = raw_input('Username: ')
password = getpass.getpass('Password: ')
####################Router####################
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip_router,username,password=password)

print "SSH connected, login to", ip_router
command = ssh_client.invoke_shell()

command.send("enable \n")
command.send("show ip interface brief \n")

time.sleep(1)
output = command.recv(65535)    # # of bytes
```

```
print output

print "Closing SSH to ", ip_router
ssh_client.close
####################Switch-L3####################
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip_switch_l3,username,password=password)

print "SSH connected, login to", ip_switch_l3
command = ssh_client.invoke_shell()

command.send("enable \n")
command.send("show ip interface brief \n")

time.sleep(1)
output = command.recv(65535)    # # of bytes
print output

print "Closing SSH to ", ip_switch_l3
ssh_client.close
####################Switch-L2####################
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip_switch_l2,username,password=password)

print "SSH connected, login to", ip_switch_l2
command = ssh_client.invoke_shell()

command.send("enable \n")
command.send("show ip interface brief \n")

time.sleep(1)
output = command.recv(65535)    # # of bytes
print output

print "Closing SSH to ", ip_switch_l2
ssh_client.close
####################Console####################
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip_console,username,password=password)

print "SSH connected, login to", ip_console
command = ssh_client.invoke_shell()

command.send("enable \n")
command.send("show ip interface brief \n")

time.sleep(1)
output = command.recv(65535)    # # of bytes
print output

print "Closing SSH to ", ip_console
ssh_client.close
####################ASA####################
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=ip_asa,username,password=password)

print "SSH connected, login to", ip_asa
command = ssh_client.invoke_shell()

command.send("enable \n")
command.send(" \n")
command.send("show switch vlan \n")

time.sleep(1)
output = command.recv(65535)    # # of bytes
print output

print "Closing SSH to ", ip_asa
ssh_client.close
```

Line 54, Column 26 Tab Size: 4

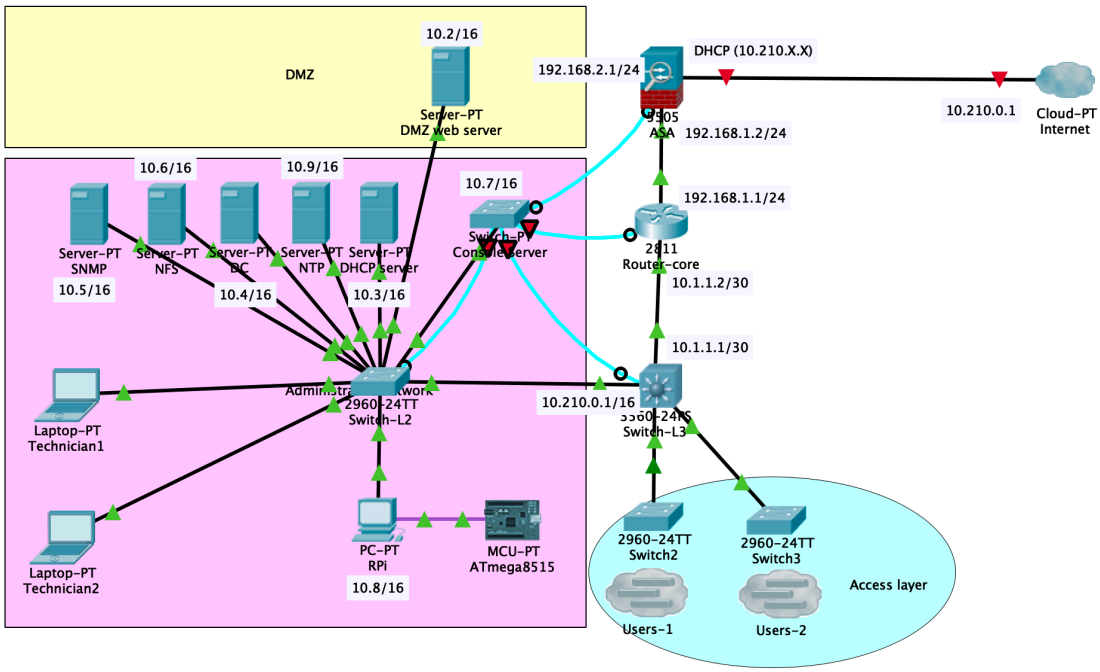
(figure 3.8.1: the “show-int” script)

Network Configuration



(Figure 4.1: the rack1 arrangement)

All the network devices within or without the administrative LAN are placed on a 22U half hight rack. The monitor, keyboard and mouse are I/O devices for all the Dell R720, Dell PC and RPi with a 4-way KVM to switch in between. All the power cords in the back are properly labeled.



(Figure 4.2: the logical topology)

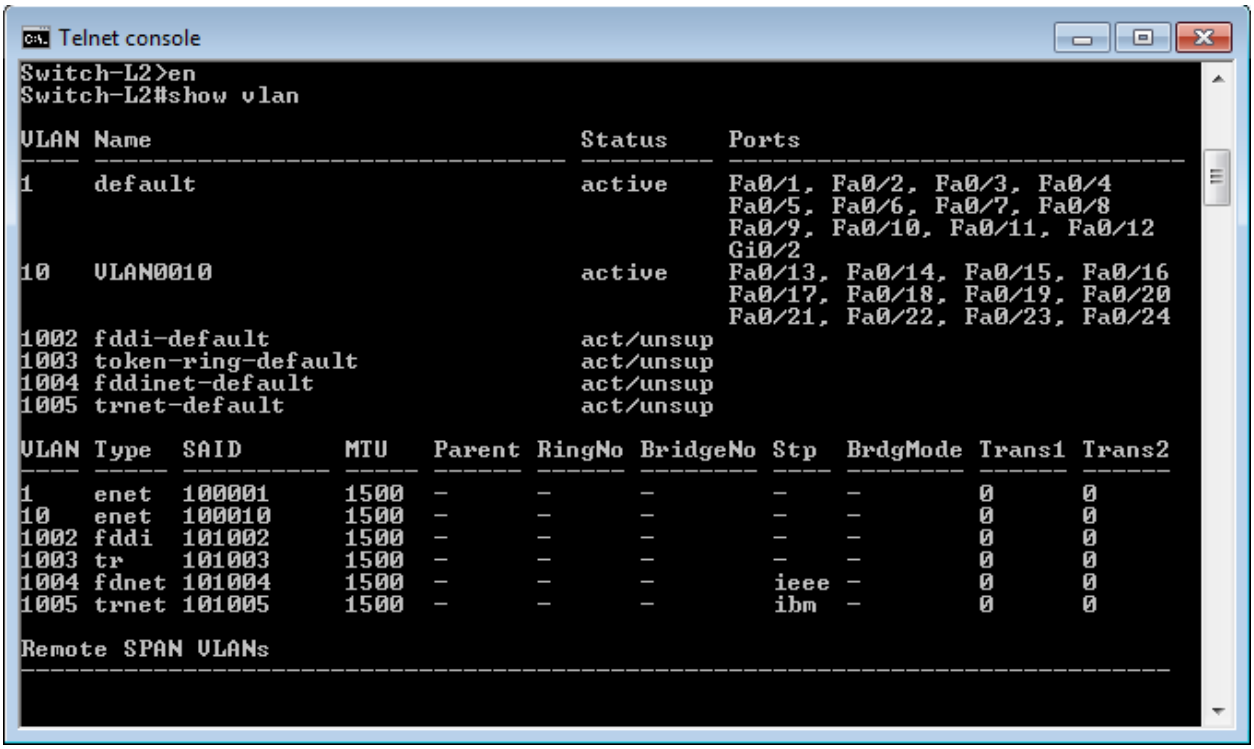
The actual network presented in the project fair is modified a bit to adapt Dawson's network. Since the inside network has the same prefix as Dawson's, there are 2 times of PAT configured on both router and ASA in order to pass through the Dawson domain and reach the internet. Also, the DMZ server is moved to the administrative VLAN for demonstration purpose. On the Switch-L2, only the ports that belong to

The first PAT on the router translates the 10.210.0.0/16 network to 192.168.1.1/24 is configured below:

```
Router-core(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.2
Router-core(config)#$ natpool 192.168.1.1 192.168.1.1 prefix-length 24
.Apr 26 15:51:32.768: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, chan
ged state to upacc
Router-core(config)#access-list 1 permit 10.0.0.0 0.255.255.255
Router-core(config)#ip nat inside source list 1 pool natpool overload
Router-core(config)#int f0/1
Router-core(config-if)#ip nat inside
Router-core(config-if)#int f0/0
Router-core(config-if)#ip nat outside
Router-core(config-if)#do ping 8.8.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

The the second PAT on ASA that translates the 192.168.1.0/24 network to a DHCP configured dynamic IP address is configured below:

```
asa(config)# object network inside-net
asa(config-network-object)# subnet 192.168.1.0 255.255.255.0
asa(config-network-object)# nat (inside,outside) dynamic interface
```



(Figure 4.3: the VLANs on the Switch-L2)

All the ports that are associated with VLAN10 have been configured with port-security. The switch will save the MAC address of the server in its table and prevent the unauthorized device continues using it to transmit traffic. The commands on the layer 2 switch are below:

```
Switch-L2(config)#int range fa0/13-24
Switch-L2(config-if-range)#switchport mode access
Switch-L2(config-if-range)#switchport port-security mac-address sticky
Switch-L2(config-if-range)#switchport port-security maximum 1
Switch-L2(config-if-range)#switchport port-security violation shut
```

Troubleshooting

1. The PCs don't get IP addresses.

Solution: access into the DHCP server (10.210.10.3) with a static IP address through the ESXi console (10.210.20.100) and check the IPv4 pool statistics. If all the IP addresses are in use, start troubleshooting from the internal DHCP snooping attach or add another scope for a new pool. If cannot access to the DHCP server, go to troubleshooting question #3 to solve the network connectivity.

2. All the servers have been shut down.

Solution: check the Pi server (10.210.10.8) mounted on the specific rack and retrieve the environmental parameters. Check the log file at specific time according with the Windows log to exclude the environmental issues. Also check the syslog and SNMP traps from the SNMP server to exclude the network issues. If they are all normal, check the LCD on the embedded system to see if there are social engineering attacks.

3. No network connectivity to any devices.

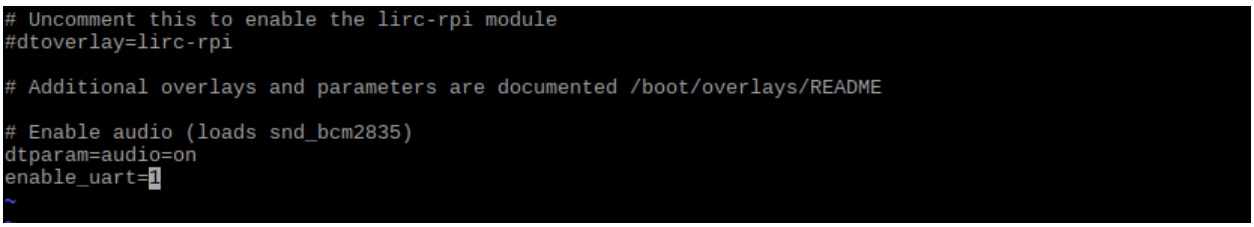
Solution: using a static IP to check if all the servers are reachable, if partially (only windows servers or Unix servers) are, check the status of the physical server or hypervisor. If cannot ping to all servers, check if the gateway is reachable. If no, testing the SVI (of VLAN10) on layer 2 switch. If still can't, troubleshoot from the local network interface card (NIC) or network adapter.

4. The "get_data.sh" stops recording the data in log file

Solution: check the log history in "/var/log/auth.log"; check the process use command "`ps –eaf |grep cron`"; use "`service cron start`" if shut down; use command "`crontab –e`" to configure the crontab schedule.

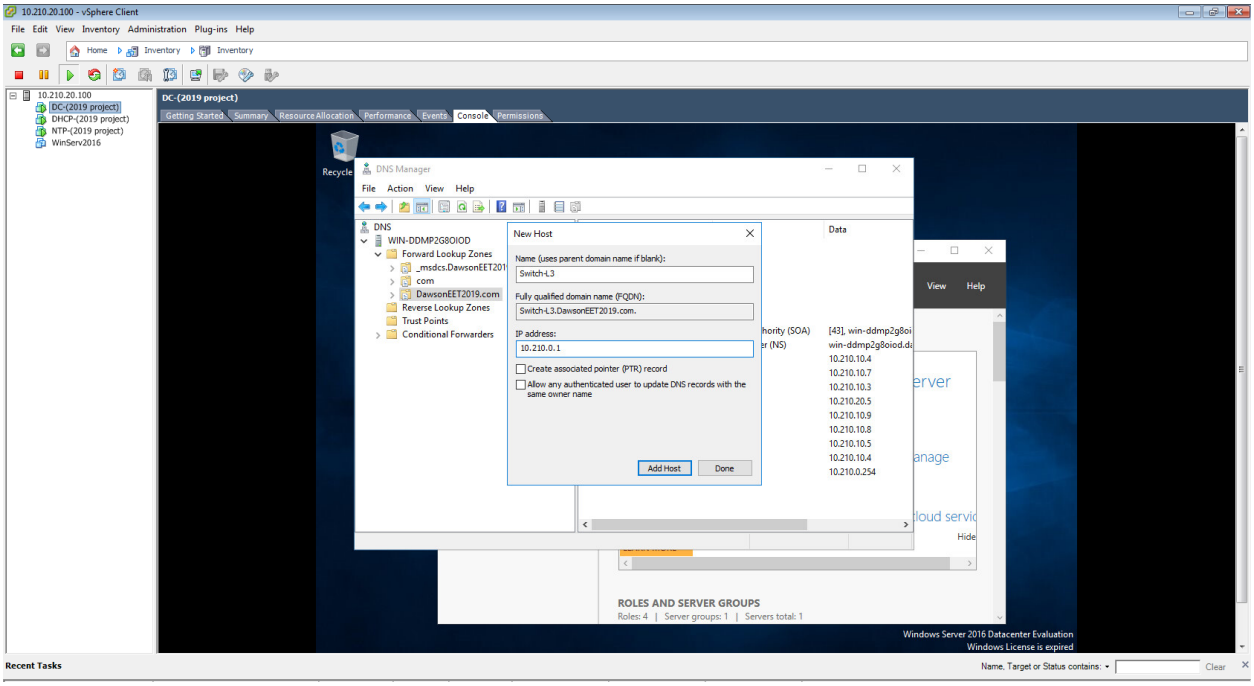
5. The serial device ttyS0 not found.

Solution: configure the file "/boot/config.txt" with root permission, and add this line at the end:



6. The web links from the navigation website are unreachable, all the websites within the LAN cannot be accessed with CNAME or A records.

Solution: Check the Domain Controller status and its DNS service. Also check the DNS list, add or modify if any hosts have changed IP addresses.



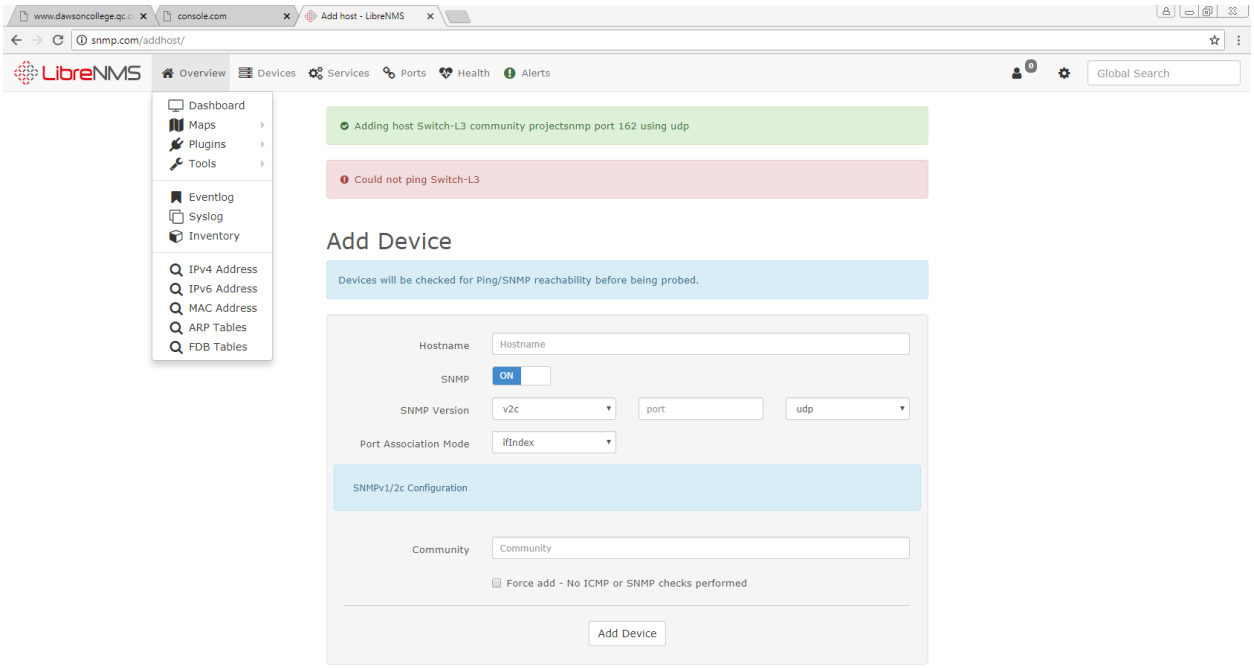
7. The navigation page is not accessible.

Solution: check the CentOS server (power status, network connectivity); check the docker using command “#docker ps -a” to display all the containers, the container needs to be started every time when the host system is booted by “#docker start <container ID/name>” and use “#docker attach <container ID/name>” to enter in; also verify the port number that the service is binding with (port forwarding).

```
root@ubuntu:/home/gary# docker start navweb
navweb
root@ubuntu:/home/gary# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
ed15411dbfe1        ubuntu              "/bin/bash"         8 days ago          Up 3 seconds
0.0.0.0:80->80/tcp   navweb
root@ubuntu:/home/gary# docker attatch navweb
docker: 'attatch' is not a docker command.
See 'docker --help'
root@ubuntu:/home/gary# docker attach navweb
root@ed15411dbfe1:/# nginx
root@ed15411dbfe1:/#
```

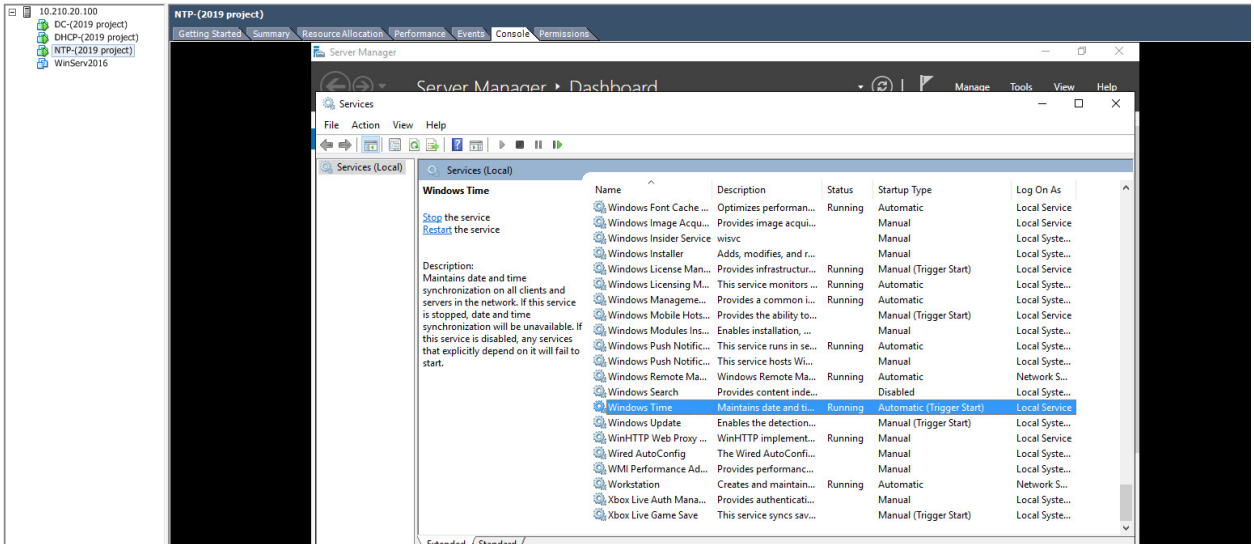
8. The SNMP manager cannot add new devices. (SNMP nodes are not polling from MIB)

Solution: the SNMP server relies on the DNS for all the host names are stored in the DNS database. The disconnection of DNS could also result the lost of all the SNMP nodes. Moreover, the firewall on windows blocks the ports 161/162 by default.



9. The NTP is synchronizing from the server.

Solution: access to the NTP server, check if the NTP is enabled in “services”; check the network activities; also check the IOS devices with “`#show ntp associations`” command.

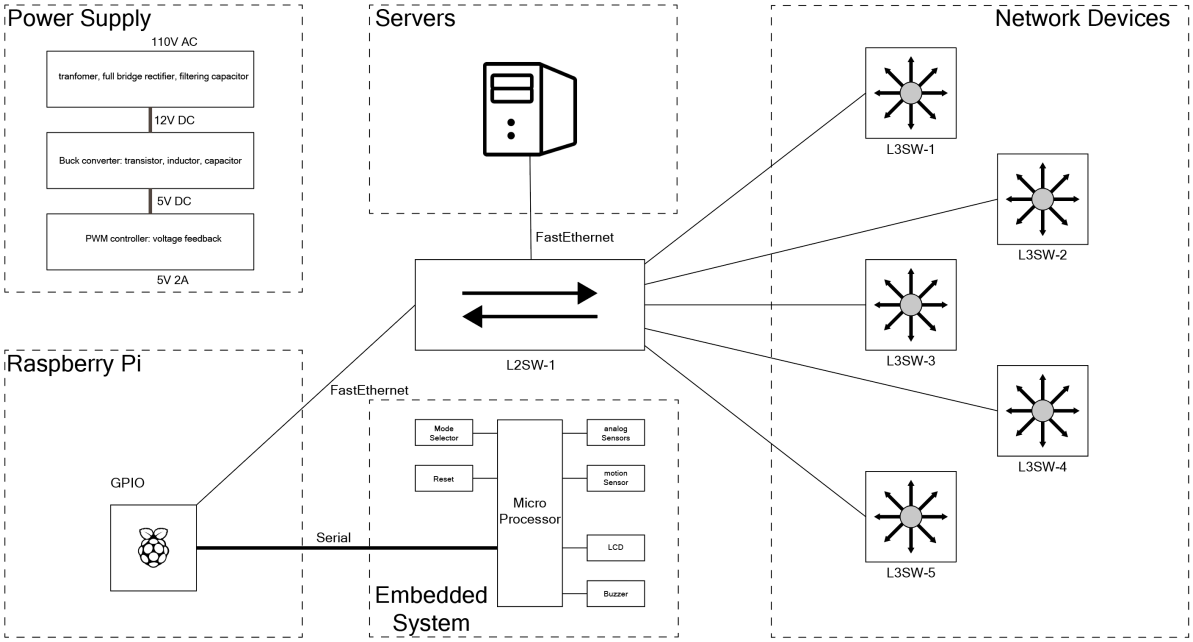


10. The administrative user cannot login into the computers.

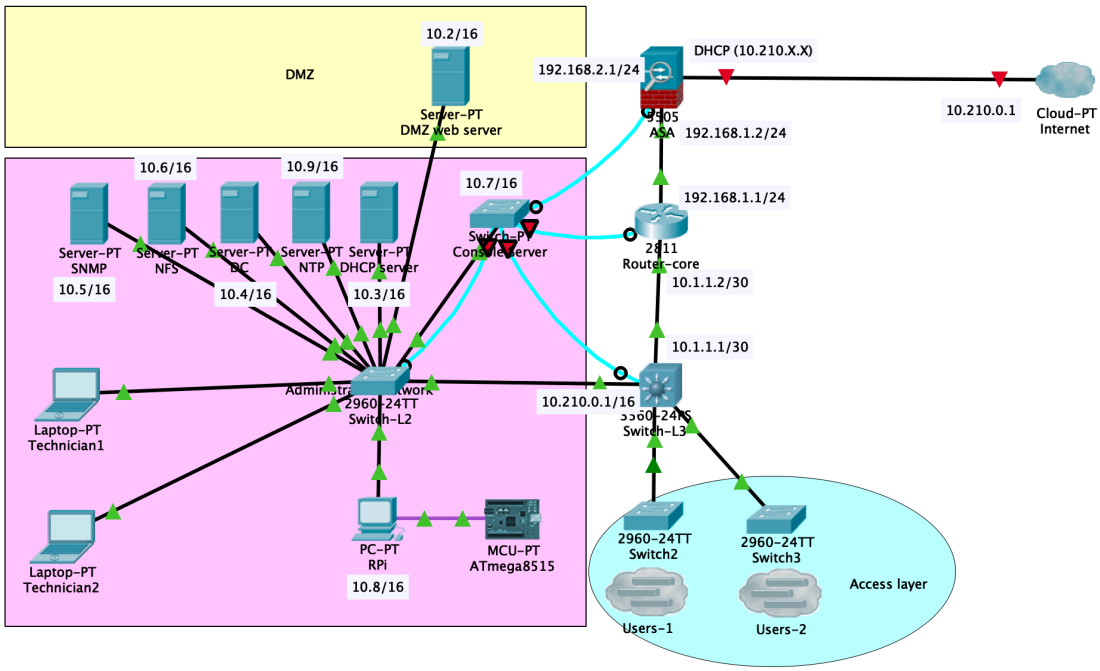
Solution: Check the DC if is down. If not, check the network connectivity.

Appendix

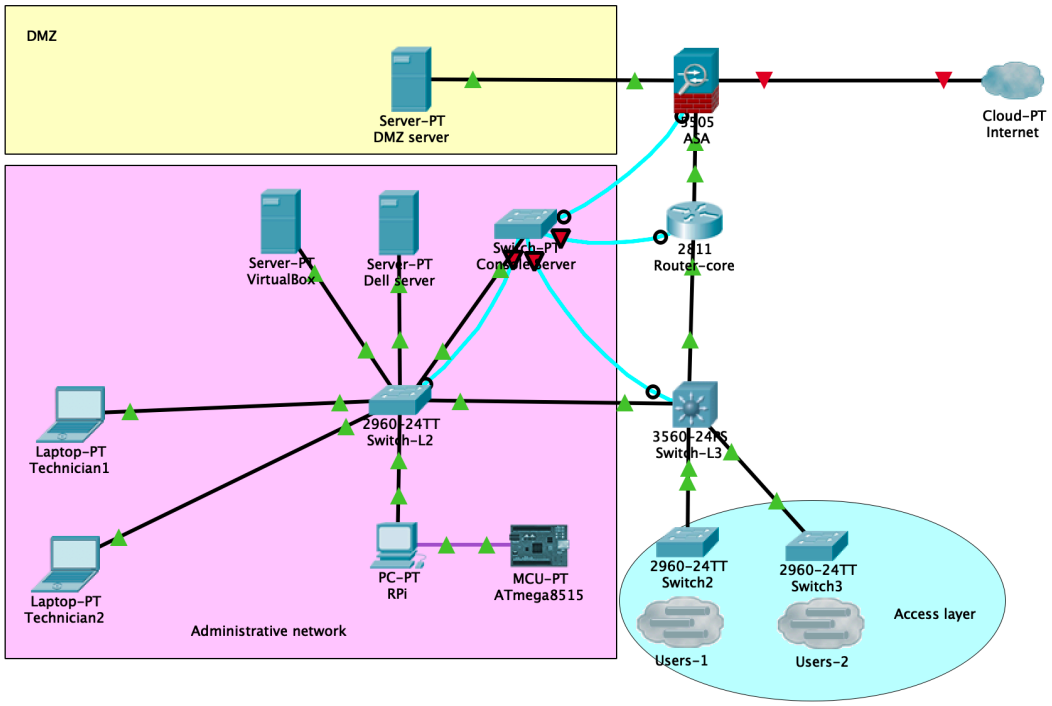
Illustrations



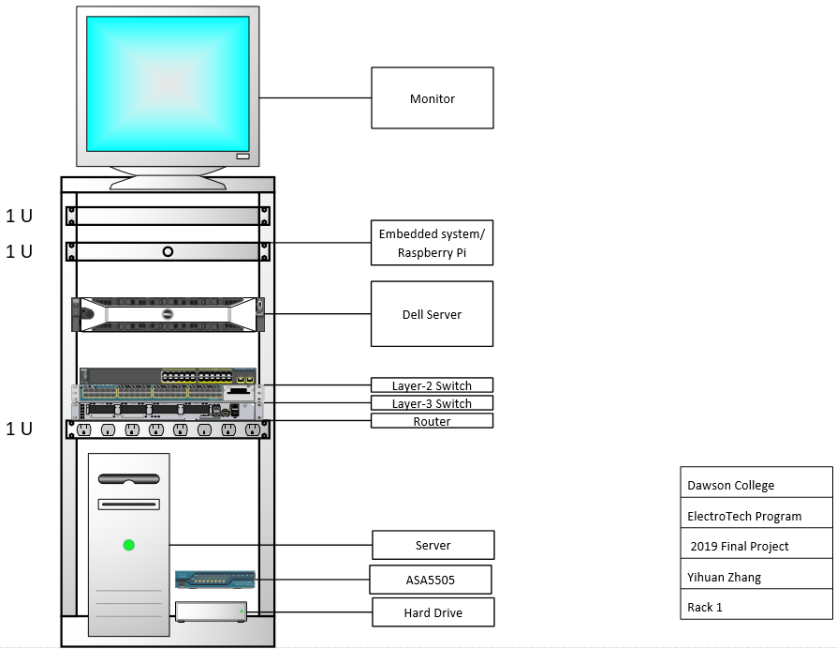
(General system diagram)



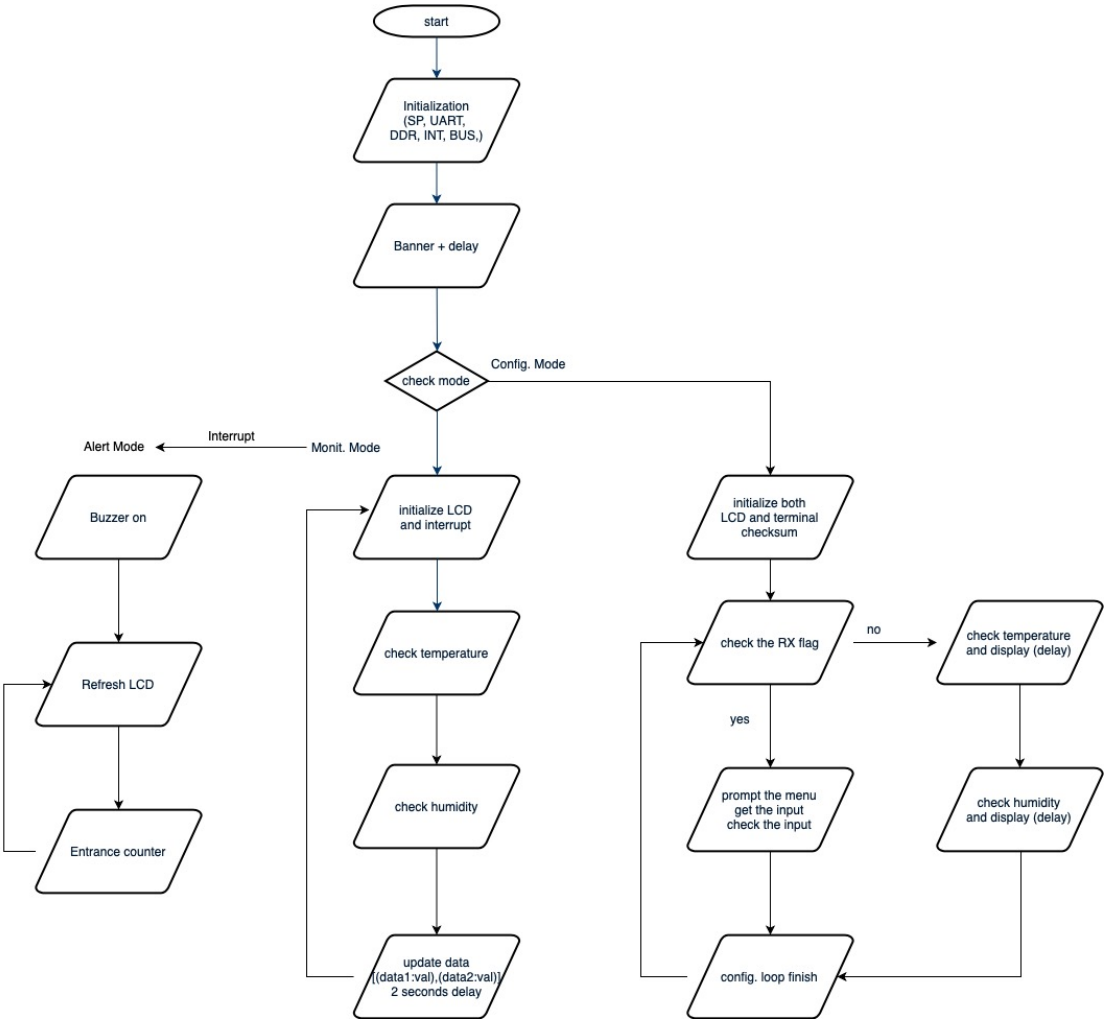
(Network logical topology)



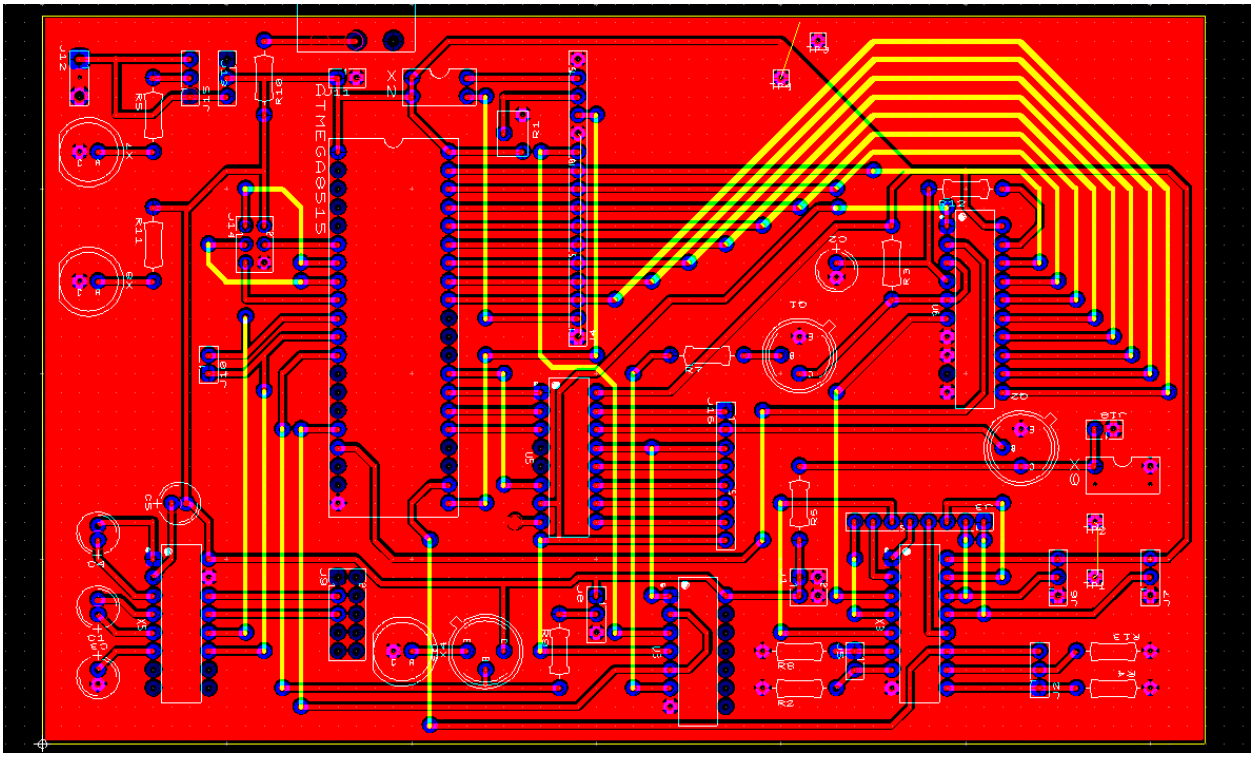
(Network physical topology)



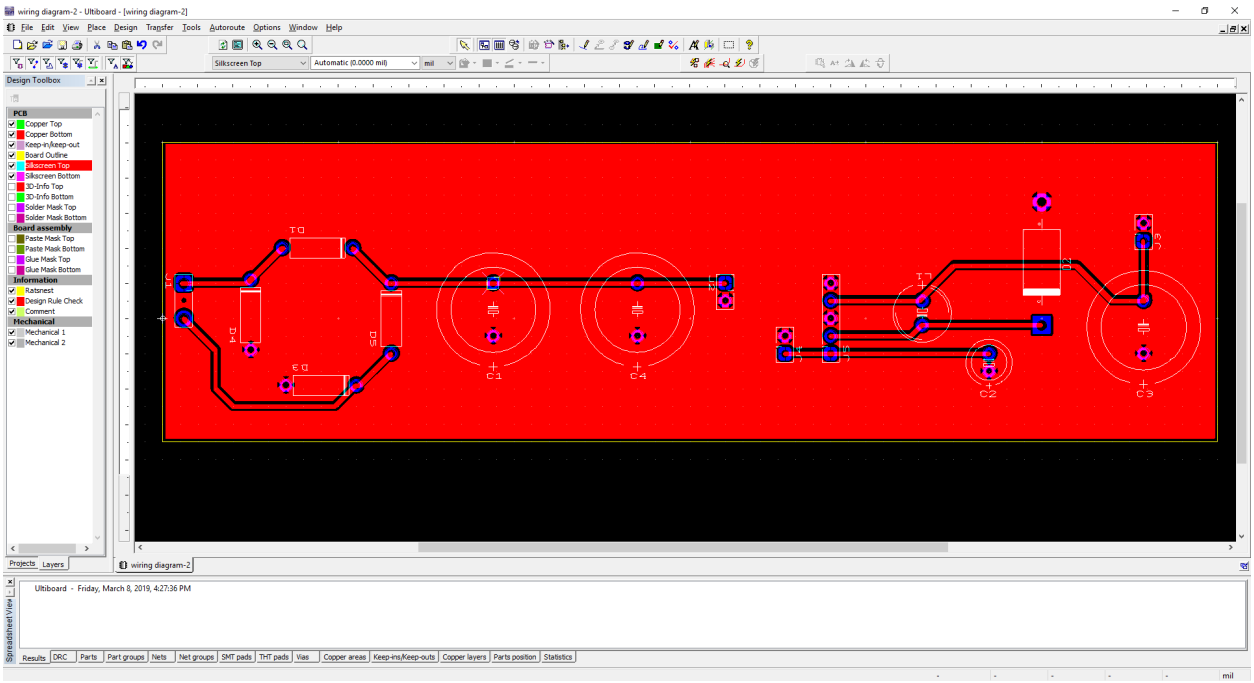
(Rack layout)



(Embedded flowchart)



(Embedded PCB diagram)



(Power supply PCB diagram)

Embedded Source Code (assembly)

```
;
; 1.0.asm
;
; Created: 1/25/2019 12:59:17 PM
; Author : 1630783
;


.equ FCPU_L  = 1000000 ;used by termio rtn
.equ BAUD  = 2400    ;desired baud rate
.EQU BUSON=$82          ;BUS and INT
.EQU VAL2=$00
.EQU ADR_ADC1=$7000  ;enable Y7' for CH0
.EQU ADR_ADC2=$7200  ;enable Y7' for CH1
.EQU ADR_LCD_CMD=$2000        ;enable Y2' and R/S for instruction
.EQU ADR_LCD_DATA=$2100       ;enable Y2' and R/S for data
;.EQU ADR_BUZ=$4000
.EQU MAX = 5
.dseg
buff:  .byte MAX


.cseg
            reset:
            rjmp init
            rjmp isr0 ;int
            .org $1E ;reserved for int
            clr R28   ;clear Y reg
            clr R29
;=========stack pointer=========
init:
            ldi R16,low(ramend)          ;stack pointer
            out spl,R16
            ldi R16,high(ramend)
            out sph,R16
            rcall init_uart       ;sbr in "termio.inc"
```

```
;=========init INT=========
        init_0:
                ldi R16,$40         ;enable int0
                out GICR,R16
;===========init bus,ddr,lcd===========
                ldi R17,BUSON
                out MCUCR,R17  ;falling edge, on Datasheet P77
                sts ADR_ADC1,R17          ;trigger ADC
                sts ADR_ADC2,R17          ;trigger ADC
                ldi R17,2;mask to enable first pin only
                out DDRB,R17     ;initialize DDR
                rcall init_lcd
;===========below only executes once===========
                in R17,PINB         ;selector
                andi R17,1          ;mask
                cpi R17,NUL
                brne monit_mode
                rjmp config_mode

;===========below executes periodly===========
monit_mode:

        rcall clear_lcd
        rcall DELAY1520uS
        rcall dis_svsr       ;first line display Supervisor mode
        clr R25   ;initialize counters
        clr R27
;       rcall terminal_menu_monit ;to be removed

monit_loop:
        rcall dis_second_line
        sei         ;enable motion sensor
        cpi R25,2           ;entering times limit
        brsh alt
;       in    R16,UCSRA  ;check the flag of RX
;       andi  R16, $80
;       breq  monit_menu_end      ;skip the terminal menu
```

```
;           rcall terminal_get_monit    ;*should put somewhere else
;           rcall checkdeta_monit
;--------print: [(data1:val),(data2:val)]
            rcall print_part_1

            rcall adc_get_data1         ;data saved in R18
            mov R16,R18
            rcall hex2asc
            rcall putchar
            mov R16,R17
            rcall putchar

            rcall print_part_2

            rcall adc_get_data2
            mov R16,R18
            rcall hex2asc
            rcall putchar
            mov R16,R17
            rcall putchar

            rcall print_part_3
            rcall DELAY1S
            rcall DELAY1S
            ;rcall DELAY1S
            rjmp monit_loop

alt:        ;alert procedure
            ldi R17, 2
            out PORTB,R17   ;turn on the buzzer signal
            rcall clear_lcd
            rcall DELAY1520uS
            rcall dis_alert1
            rcall dis_loop
refresh:    ;counting enter times until reset
            rcall lcd_alt
            rjmp refresh
```

```
config_mode:
        rcall banner_terminal
        rcall banner_lcd
        rcall dis_second_line
        rcall checksum      ;on both lcd and terminal
        rcall DELAY1S       ;delay for banner
        rcall DELAY1S       ; and motion detector
        rcall DELAY1S
        rcall DELAY1S
        rcall DELAY1S
        cli             ;disable sensor
        rcall clear_lcd
        rcall DELAY1520uS
        rcall dis_node      ;first line dispaly Node mode
default:
        ldi R19,128         ;set default trim level
        rcall terminal_data             ;general value of T and H
        rcall dis_second_line
        ;rcall terminal_config
background:
        in    R16,UCSRA  ;check the flag of RX
        andi  R16, $80
        brne  terminal_config       ;skip the terminal menu
        rcall checkdata1   ;also display on LCD, only in configuring mode
        rcall DELAY1S
        rcall checkdata2
        rcall DELAY1S
        rjmp background
fini:
        rjmp fini

terminal_config:
        rcall terminal_menu_conf
        rcall terminal_get_conf     ;data saved in R16
        rcall check_input_conf
        rjmp background
```

```
;===========Interrupt subroutine===========
isr0:
        inc R25
        rcall calc_int_time
        reti

calc_int_time:
        cpi R25,$64         ;100 times
        brsh one_hundred
        ret
one_hundred:
        clr R25
        cpi R27,$64
        brsh cnt_reset
        inc R27
cnt_reset:
        ret
ten_thousands:
        clr R25
        rjmp cnt_reset
;===========Checksum===========
checksum:
        clr R20
        clr R21
        clr R0
        ldi R30,0
        ldi R31,0
        ldi R28,high(esum<<1)      ;store the last address in Y reg
        ldi R29,low(esum<<1)
checksum_calc:
        lpm R19,Z+          ;to hold the value
        add R20,R19         ;add the number on the lsb
        adc R21,R0          ;hold or add the carry on the msb
        sbiw R28:R29,1
        brne checksum_calc
        rcall dis_checksum
        rcall dis_loop
```

```
        mov R16,R21
        rcall hex2asc       ;R17,R16
        rcall ascii_on_lcd
        rcall ascii_on_terminal
        mov R16,R20
        rcall hex2asc
        rcall ascii_on_lcd
        rcall ascii_on_terminal
        ret

;===========LCD results===========
checkdata1:
        rcall DELAY1520uS
        rcall dis_second_line
        rcall dis_data1     ;"data1 = "
        rcall dis_loop
        rcall adc_get_data1         ;aquire data in R18, record H L in R1 R2
        rcall percentage_humidity  ;convert hex of R18 into percentage
        mov R16,R18
        rcall hex2bcd8
        rcall dis_value
        ldi R22,$25         ;'%'
        rcall dis_char
        rcall DELAY1S
        rcall DELAY1S
        rcall DELAY1S
        rcall clear_second_line
        rcall dis_HL        ;display H:L
        ret

checkdata2:
        rcall DELAY1520uS
        rcall dis_second_line
        rcall dis_data2
        rcall dis_loop
        rcall adc_get_data2
        ;rcall percentage_trim
```

```
        mov R16,R18
        rcall hex2bcd8
        rcall dis_value
        ldi R22,$DF         ;check P14 of "ST7066U"
        rcall dis_char
        ldi R22,$43         ;'%'
        rcall dis_char
        rcall DELAY1S
        rcall DELAY1S
        rcall DELAY1S
        rcall clear_second_line
        rcall dis_HL        ;display H:L
        ret

dis_value:      ;put the displaying value in R16 in hex
                rcall hex2asc
                mov R22,R16
                rcall dis_char      ;display MSB
                mov R22,R17
                rcall dis_char      ;display LSB
                ret

lcd_alt:    ;a counter
        rcall dis_second_line
        rcall dis_alert2
        rcall dis_loop
;       mov R16,R27     ;if more than a hundred
;       rcall hex2bcd8
;       rcall hex2asc
;       rcall ascii_on_lcd
        mov R16,R25
        rcall hex2bcd8
        rcall hex2asc
        rcall ascii_on_lcd
        rcall dis_alert3
        rcall dis_loop
        ret
```

```
;===========ADC subroutine===========
;R19 is the trim level;
;R2 stores the values exceeded
;R1 stores the values below

adc_get_data1:
        clr R3    ;L
        clr R1    ;H
        clr R18
        ldi R16,$64;100 samples
        data_loop1:
                lds R18,ADR_ADC1 ;read the data from ADC
                cp R18,R19
                brsh cnt_high1    ;R1++
                inc R3    ;R3++
        data_loop1_end:
                dec R16
                brne data_loop1
        ret
cnt_high1:
        inc R1
        rjmp data_loop1_end

adc_get_data2:
        clr R3
        clr R1
        clr R18
        ldi R16,$64;100 samples
        data_loop2:
                lds R18,ADR_ADC2 ;read the data from ADC
                cp R18,R19
                brsh cnt_high2    ;R1++
                inc R3    ;R3++
                data_loop2_end:
                dec R16
                brne data_loop2
        ret
```

```
cnt_high2:
        inc R1
        rjmp data_loop2_end

;================

ascii_on_lcd:       ;display two vals (R16,R17) on LCD (R22)
        push R16
        push R17
        mov R22,R16
        rcall dis_char      ;display MSB
        mov R22,R17
        rcall dis_char      ;display LSB
        pop R16
        pop R17
        ret
ascii_on_terminal:          ;send R16,R17 to UDR (R16)
        push R16
        push R17
        mov R15,R16
        mov R16,R17
        rcall putchar
        mov R16,R15
        rcall putchar
        pop R16
        pop R17
        ret
;include files must be in the same folder
.include "termio.inc"   ;routines to do terminal io using AVR's USART
.include "numio.inc"    ;routines from same folder
.include "lcdio_project.inc"
.include "delay_project.inc"
esum: .db 0

.exit
```

(The “main.asm” code)

```
;
; Filename: lcdio_project.inc
;
;this library is used for LCD display
;
;Author: Yihuan Zhang
;
;Changelog:
;1v0 01/28/19 - include lcd init, dis_char
;1v1 01/31/10 - banner msg
;
;
init_lcd:
        rcall DELAY40mS
        ;Funtion set
        ldi R17,$38
        sts ADR_LCD_CMD,R17 ;sending data
        ldi R16,9           ;37us=9loop * 4clk
        rcall DELAY4clk
        ;Function set
        ldi R17,$38
        sts ADR_LCD_CMD,R17
        ldi R16,9           ;37us
        rcall DELAY4clk
        ;Display ON/OFF control
        ldi R17,$0C
        sts ADR_LCD_CMD,R17
        ldi R16,9           ;37us
        rcall DELAY4clk
        ;Display clear
        ldi R17,$01
        sts ADR_LCD_CMD,R17
        rcall DELAY1520uS
        ;Entry mode set
        ldi R17,$06
        sts ADR_LCD_CMD,R17
        ldi R16,9           ;37us
```

```
        rcall DELAY4clk
        ret
clear_lcd:
        rcall DELAY1520uS
        ldi R17,$01         ;P17 of datasheet
        sts ADR_LCD_CMD,R17
        rcall DELAY1520uS
        ret
clear_second_line:          ;filling the line with blank
        rcall dis_second_line
        ldi R31,high(blank<<1)
        ldi R30,low(blank<<1)
        rcall dis_loop
        blank: .db "                ",NUL
        rcall dis_second_line
        ret

dis_second_line:
        ldi R17,$C0         ;DDRAM
        ldi R16,9           ;37us
        rcall DELAY4clk
        sts ADR_LCD_CMD,R17
        ldi R16,9           ;37us
        rcall DELAY4clk
        ret


dis_loop:
                lpm R22, Z+
                cpi R22,NUL
                breq dis_loop_end
                rcall dis_char
                rjmp dis_loop
        dis_loop_end:
                ret

dis_char:       ;load R22 into LCD to display
```

```
        push R17        ;for time delay
        sts ADR_LCD_DATA,R22
        ldi R17, $FA    ;1mS = 250loop * 4clk
        rcall DELAY4clk
        pop R17
        ret

banner_lcd:
;to init
        ldi R16,9       ;37us
        rcall DELAY4clk
        rcall dis_lcd_banr      ;first line
        ldi R17,$C0     ;DDRAM
        ldi R16,9       ;37us
        rcall DELAY4clk
        sts ADR_LCD_CMD,R17
        ldi R16,9       ;37us
        rcall DELAY4clk
        ret
dis_lcd_banr:   ;display the message in the first line
        ldi R31, high(lcd_banr<<1)
        ldi R30,low(lcd_banr<<1)
        rcall dis_loop
        lcd_banr: .db "N.A.M.System",NUL
        ret
dis_checksum:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(dis_cks<<1)
        ldi R30,low(dis_cks<<1)
        dis_cks: .db "Checksum=$",NUL
        ret
dis_svsr:       ;display the message in the second line
        ldi R16,9       ;37us
        rcall DELAY4clk
        ldi R31, high(lcd_svsr<<1)
        ldi R30,low(lcd_svsr<<1)
```

```
        rcall dis_loop
lcd_svsr: .db "Monit. Mode",NUL
        ret
dis_node:       ;display the message in the second line
        ldi R16,9       ;37us
        rcall DELAY4clk
        ldi R31, high(lcd_node<<1)
        ldi R30,low(lcd_node<<1)
        rcall dis_loop
lcd_node: .db "Config. Mode",NUL
        ret
dis_alert1:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(alert1<<1)
        ldi R30,low(alert1<<1)
        alert1: .db " !Unauthorized",NUL
        ret
dis_alert2:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(alert2<<1)
        ldi R30,low(alert2<<1)
        alert2: .db "Enter:",NUL
        ret
dis_alert3:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(alert3<<1)
        ldi R30,low(alert3<<1)
        alert3: .db " times",NUL
        ret

dis_data1:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(data1<<1)
```

```
        ldi R30,low(data1<<1)
        data1: .db "Humidity ",NUL
        ret
dis_data2:
        ldi R16,9
        rcall DELAY4clk
        ldi R31,high(data2<<1)
        ldi R30,low(data2<<1)
        data2: .db "Temperature ",NUL
        ret
dis_HL: ;take R1 R3 and display in H:L
        ldi R31,high(highval<<1)
        ldi R30,low(highval<<1)
        rcall dis_loop
        highval: .db " H:",NUL
        ;load H data
        mov R16,R1
        rcall dis_value
        ldi R22,$20
        rcall dis_char
        ldi R31,high(lowval<<1)
        ldi R30,low(lowval<<1)
        rcall dis_loop
        lowval: .db " L:",NUL
        ;load L data
        mov R16,R3
        rcall dis_value
        ret
```

(The “lcd_project.inc” library)

```
;
; Filename: termio.inc
;
; The following code supplied to the Fall 2018 243-513-DW students for educational/study
; purposes. The use of these routines, in whole or in part, without proper reference to
; origin is a violation of ISEP regarding Academic Integrity and Plagerism.
;
;Description: Subroutine framework for use in the project (do not modify)
;
;original Author: Mr. Markou
;
;Changelog:
;0v9 11/04/17 - beta version of serialio.asm
;1v0 11/10/17 - finalized and commented version
;2v0 11/11/18 - forked into this version with extra routines
;2v1 11/12/18 - restructured for multi device usability e.g.UART/LCD
;2v2 11/20/18 - added some bells and whistles, and cookie (find it).
;3v0 01/28/19 - modified by Gary 1630783 for project purpose
;3v1 01/31/19 - changed the banner msg
.equ  UBRR  = (FCPU_L /(16 * BAUD)) -1   ;see p.138 (important)
.equ  FRAME = $86     ;8N1 standard frame
.equ  TXE = $18       ;Transmit & receive enable
.equ  LF = $0A                ;ASCII line feed
.equ  CR = $0D                ;ASCII carriage return
.equ  NUL = 0         ;string terminating value
.equ  EOL = 0         ;string terminating value
.equ  EOT = $04       ;string terminating value


;
; Standard USART init which logically belongs here
;
init_uart:
          ldi R16, 0              ;always zero (mostly)
          out UBRRH, R16
          ldi R16, UBRR
          out UBRRL, R16    ;config. the rate of data tx
          ldi R16, TXE
```

```
        out UCSRB, R16    ;enable port tx (see p.158)
        ldi R16, FRAME    ;defined in calling
        out UCSRC, R16    ;config. frame elements
        ret
;
;*gets - asm workalike C routine which inputs from UART RxD
; Entry: Z reg must point to buffer
;       R17 must contain the size of the buffer.
gets:
        dec   R17               ;leave room for NUL w/o exceeding buffer limit
gtx:
   rcall getche
        st          Z+, R16
        cpi         R16, CR     ;check for returm
        breq  gty
        dec         R17
        brne  gtx
gty:
        ldi   R16, NUL    ;place end of string
        st    Z, R16
        ret
;
;*getch - asm workalike C routine to receive char from UART
; Exit:  R16 contain rx char
getch:
        in    R16,UCSRA
        andi  R16, $80
        breq  getch                 ;poll status for key press
        in          R16,UDR     ;get char from UART
        ret
;
;*getche - like above but w/echo
; Exit:  R16 contain rx char
getche:
        rcall getch
        push  R16         ;save R16 on stack
        rcall putchar
```

```
        pop  R16        ;and retrieve it back
        ret
;
;*putchar - char tx UART routine
; Entry: R16  char. to send
putchar:
   out  UDR,R16    ;txmt char. out the TxD
putc1:
   in   R16, UCSRA    ;poll status reg
        andi R16, $20      ;check for tx complete
        breq putc1
        ret
;
;*puts - asm workalike routine to puts() in C
; Entry: Z index points to an NUL terminated string in cseg or dseg
puts:
        lpm   R16, Z+      ;get char from memory
        cpi   R16, NUL
        breq  px           ;this is the end, my only friend, the end
        rcall putchar      ;send it to uart
        rjmp  puts
px: ret
;
;*newl - issues a new line (CR&LF) which comes in handy
; Entry: R16
newl:
        ldi   R16, LF      ;nothing new here
   rcall putchar
        ldi   R16, CR
   rcall putchar
        ret
;
;//insert following two directives to main program a line before .exit
;.nolist
;.include "termio.inc"   ;append library subroutines from same folder
```

(The “termio.inc” library)

```
;
; Filename: numio.inc
;
; The following code supplied to the Fall 2018 243-513-DW students for educational/study
; purposes. The use of these routines, in whole or in part, without proper reference to
; origin is a violation of ISEP regarding Academic Integrity and Plagerism.
;
;Description: Subroutine framework for use in the project (do not modify)
;
;Author: Mr. Markou
;
;Changelog:
;1v0 11/10/17 - finalized and commented version
;2v0 11/11/18 - forked into this version with extra routine
;2.1 11/12/19 - added hex2bcd8
;3v0 02/7/19 - modified by Gary 1630783 for project purpose
;----------------
;*hex2asc - converts an 8 bit hex value into a valid ASCII characters by
;*              masking and shifting ASCII character into a valid binary form
; Entry: R16 should contain the hex number to convert
; Usage: R15,R17,R18 **save before calling rtn if used
; Exit:  R17:R16 contain msd:lsd respectively
hex2asc:
    clr   R17
    mov   R15,R16      ;make copy of hex number
    lsr   R16          ;shift ms nyble to ls
    lsr   R16
    lsr   R16
    lsr   R16
h2a:
    andi  R16,$0F      ;mask off upper nyble
    ldi   R18,$30
    add   R16,R18      ;add $30 to adjust to ascii 0 - 9
    cpi   R16,$3A      ;check if valid bcd range
    brlo  h2b
    ldi   R18,7        ;adjust to hex char into 'A' - 'F'
    add   R16,R18
```

```
h2b:
    push  R16                ;save msd ascii
    mov   R16,R15
    inc   R17
    cpi   R17,1
    breq  h2a
          pop  R17          ;R17:R16 contain msd:lsd ascii numbers
    pop  R16
    ret
;---------------
;* hex2BCD8 - 8-bit Binary to BCD rtn which converts a hex number
;           from $00 - $63 to $00 - $99 in bcd
; Entry: R16 should contain the hex number to convert
; Exit: R16 contains converted number
hex2bcd8:
          push   R17          ;save on stack in case of zombies
          clr    R17          ;clear R16 reg
msd:
          subi   R16,10       ;input = input - 10
          brcs   lsd          ;abort if carry set
          subi   R17,-$10     ;R17 = R17 + 10
          rjmp   msd          ;loop again
lsd:
          subi   R16,-10      ;compensate extra subtraction
          add    R16,R17
          pop    R17          ;restore it back
          ret


;-------------$00~$FF to 0~100 lookup table----------------
percentage_trim:;compress a hex number into a bcd number
          cpi R18,$1A         ;check if 10%
          brlo pct0
          cpi R18,$33         ;20%
          brlo pct10
          cpi R18,$4D         ;30%
          brlo pct20
          cpi R18,$66
```

```
        brlo pct30
        cpi R18,$80
        brlo pct40
        cpi R18,$99
        brlo pct50
        cpi R18,$B3
        brlo pct60
        cpi R18,$CC
        brlo pct70
        cpi R18,$E6
        brlo pct80
        cpi R18,$FF
        brsh pct100
percentage_humidity:;linear output between 0%~75.3% and 75.3%~100%
        cpi R18,$31
        brlo pct0
        cpi R18,$40
        brlo pct10
        cpi R18,$50
        brlo pct20
        cpi R18,$5F
        brlo pct30
        cpi R18,$6F
        brlo pct40
        cpi R18,$7F
        brlo pct50
        cpi R18,$8F
        brlo pct60
        cpi R18,$9F
        brlo pct70
        cpi R18,$AF
        brlo pct80
        cpi R18,$BF
        brlo pct90
        cpi R18,$CF
        brsh pct100
table_end:
```

```
        ret

pct0:
        ldi R18,$0
        rjmp table_end
pct10:
        ldi R18,$A
        rjmp table_end
pct20:
        ldi R18,$14
        rjmp table_end
pct30:
        ldi R18,$1E
        rjmp table_end
pct40:
        ldi R18,$28
        rjmp table_end
pct50:
        ldi R18,$32
        rjmp table_end
pct60:
        ldi R18,$3C
        rjmp table_end
pct70:
        ldi R18,$46
        rjmp table_end
pct80:
        ldi R18,$50
        rjmp table_end
pct90:
        ldi R18,$5A
        rjmp table_end
pct100:
        ldi R18,$63
        rjmp table_end
;
;//insert following two directives to main program a line before .exit
```

```
;.nolist
;.include "numio.inc"   ;append library subroutines from same folder
```

(The “numio.inc” library)

Networking Configuration (IOS)

Switch-L2 running configuration:

```
Switch-L2#show run
Building configuration...

Current configuration : 4600 bytes
!
! Last configuration change at 18:22:17 UTC Fri May 3 2019
! NVRAM config last updated at 18:22:24 UTC Fri May 3 2019
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch-L2
!
boot-start-marker
boot-end-marker
!
!
username admin privilege 15 secret 4 3Paji2ygor46xBsNLb2Dyi9cbUlP3gjcpz/VwBMy1zE
no aaa new-model
system mtu routing 1500
!
!
ip domain-name DawsonEET2019.com
ip name-server 10.210.10.4
!
!
crypto pki trustpoint TP-self-signed-2709964672
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2709964672
 revocation-check none
 rsakeypair TP-self-signed-2709964672
!
!
crypto pki certificate chain TP-self-signed-2709964672
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32373039 39363436 3732301E 170D3933 30333031 30303030
```

```
  35355A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 37303939
  36343637 3230819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100A1D2 C4DCA671 F8837AFF 0D8B42FF F4798958 FDDB9951 26127272 BCABCBBC
  0F183ABD 31B1AD3B DD3D6C9C 0FC79EB9 421B3A86 9D15BCEB E5811991 C8467A7E
  430E45B4 65D935DB 63A89F54 1A9F78F2 15035CFA 312D71E6 7931C7FC A23C9E27
  649232BB 57742D37 BABB9D7F C9F9E40A DE61138B 6DDBCDD2 93814DFF 30BF30D6
  56E50203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 14E6E86D C1E65A10 2554E4BE 0A683AD2 620156DC 79301D06
  03551D0E 04160414 E6E86DC1 E65A1025 54E4BE0A 683AD262 0156DC79 300D0609
  2A864886 F70D0101 05050003 81810074 528C4220 B7923AA5 A063FE0A FC201FB2
  7DAE19CB 5440F8FA 349D45DE 1A522F1D EC4D5FA6 56C0635C 23BA475C AC0FE925
  0208C038 72B2549B EDC9DDA1 07E4B9C1 959DB0E5 2097BFA9 1246D5DF 4881773D
  B69292CA 686ADBB0 FF165DC8 FF4BDB13 631C57E7 A383D3AC 1BBD8CEA 305E309F
  E2456002 75AEB510 CB84DE15 8B1795
        quit
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
```

```
interface FastEthernet0/12
!
interface FastEthernet0/13
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/14
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/15
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/16
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/17
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/18
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/19
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/20
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/21
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/22
 switchport access vlan 10
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0040.9d29.0d2c
```

```
!
interface FastEthernet0/23
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface FastEthernet0/24
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address sticky
!
interface GigabitEthernet0/1
 switchport access vlan 10
 switchport mode trunk
!
interface GigabitEthernet0/2
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan10
 ip address 10.210.0.254 255.255.0.0
!
ip http server
ip http secure-server
!
!
line con 0
line vty 0 4
 login local
 transport input ssh
line vty 5 15
 login
!
ntp authentication-key 1234 md5 072C2D4D5D1A585744 7
ntp authenticate
ntp trusted-key 1234
ntp server 10.210.10.9
end
```

Router-core running configuration:

```
Router-core#show run
Building configuration...


Current configuration : 6052 bytes
!
! Last configuration change at 18:04:47 UTC Thu May 2 2019
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router-core
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!
!
dot11 syslog
ip source-route
!
!
ip cef
!
!
!
ip domain name DawsonEET2019.com
ip name-server 10.210.10.4
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
voice-card 0
!
crypto pki token default removal timeout 0
!
!
!
!
```

```
license udi pid CISCO2811 sn FHK1446F14K
username admin privilege 15 secret 4 3Paji2ygor46xBsNLb2Dyi9cbUlP3gjcpz/VwBMy1zE
!
redundancy
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.1.1.2 255.255.255.252
 ip nat inside
 ip virtual-reassembly in
 duplex auto
 speed auto
!
interface Serial0/0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip nat pool natpool 192.168.1.1 192.168.1.1 prefix-length 30
ip nat inside source list 1 pool natpool overload
ip route 0.0.0.0 0.0.0.0 192.168.1.2
ip route 10.210.0.0 255.255.0.0 FastEthernet0/1
!
ip access-list standard SNMP-ACL
 permit 10.210.10.5
!
access-list 1 permit 10.0.0.0 0.255.255.255
!
```

```
!
!
!
snmp-server community projectsnmp RO SNMP_ACL
snmp-server location Rack1
snmp-server contact yihuan.zhang@dawsoncollege.qc.ca
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps xgcp
snmp-server enable traps flash insertion removal
snmp-server enable traps c3g
snmp-server enable traps adslline
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
snmp-server enable traps vstack
snmp-server enable traps license
snmp-server enable traps disassociate
snmp-server enable traps deauthenticate
snmp-server enable traps authenticate-fail
snmp-server enable traps dot11-qos
snmp-server enable traps switch-over
snmp-server enable traps rogue-ap
snmp-server enable traps wlan-wep
snmp-server enable traps rf
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps bfd
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps memory bufferpeak
```

```
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps dial
snmp-server enable traps dsp card-status
snmp-server enable traps dsp oper-state
snmp-server enable traps entity
snmp-server enable traps fru-ctrl
snmp-server enable traps resource-policy
snmp-server enable traps event-manager
snmp-server enable traps frame-relay multilink bundle-mismatch
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps isis
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps ipsla
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps l2tun pseudowire status
snmp-server enable traps vtp
snmp-server enable traps pw vc
snmp-server enable traps firewall serverstatus
snmp-server enable traps ipmobile
snmp-server enable traps nhrp nhs
snmp-server enable traps nhrp nhc
snmp-server enable traps nhrp nhp
snmp-server enable traps nhrp quota-exceeded
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps ccme
snmp-server enable traps srst
snmp-server enable traps mpls vpn
```

```
snmp-server enable traps voice
snmp-server enable traps dnis
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
snmp-server host 10.210.10.5 version 2c projectsnmp
!
!
control-plane
!
!
!
!
mgcp profile default
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login local
 transport input ssh
!
scheduler allocate 20000 1000
ntp server 10.210.10.4
end
```

Switch-L3 running configuration:

```
Switch-L3#show run
Building configuration...

Current configuration : 6407 bytes
!
! No configuration change since last restart
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch-L3
!
boot-start-marker
boot-end-marker
!
!
username admin privilege 15 secret 5 $1$W59B$ZJWUAoJqifj9m/tYH8Ddi/
```

```
no aaa new-model
system mtu routing 1500
ip routing
ip domain-name DawsonEET2019.com
ip name-server 10.210.10.4
!
!
!
!
crypto pki trustpoint TP-self-signed-1074126208
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1074126208
 revocation-check none
 rsakeypair TP-self-signed-1074126208
!
!
crypto pki certificate chain TP-self-signed-1074126208
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 31303734 31323632 3038301E 170D3933 30333031 30303031
  32335A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 30373431
  32363230 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100AD37 D3B48E2F EA1CA934 8750CD55 DE2AC840 8E75374D BEE8EA4F F332F623
  56E5F61A 26CB3C5C D1124A11 4E90C81D AA05E838 72E1B112 D2B396D9 441929E2
  8413E2BA D0A0C6B3 1F32D0CA 0C4D7AD9 E40265BD 217642B0 A87D29D3 9D21331F
  A2FC3E90 9FAB2713 662A6A91 0E6D3E85 956CB1CC 43EB02AB 64974D23 2F8A76A0
  9A3D0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 14D87488 EECADAFC 05735BFB 6D876572 24ED9518 10301D06
  03551D0E 04160414 D87488EE CADAFC05 735BFB6D 87657224 ED951810 300D0609
  2A864886 F70D0101 05050003 81810074 A9FCA58D 294E5920 CBF2C9E8 039C7D40
  F76341CC F950BA6C 33ABA8EF 3D12D14C C27FE329 A3DBA00A 2168C055 9DE542A8
  89F7FC47 68FD686A D9A5E985 B5C6181F 4C4FB705 B028BF4F 16FBAC75 DBAAF6FE
  F8B8C6A9 7673C6BE 3A5C4464 56779AF9 F449B031 3AF1FEF6 E29C4298 32F0A814
  D642F4BE 5BA86B43 8CBE224F 558D70
       quit
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
```

```
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/22
!
interface FastEthernet0/23
 switchport access vlan 10
 switchport trunk encapsulation dot1q
!
interface FastEthernet0/24
 no switchport
```

```
 ip address 10.1.1.1 255.255.255.252
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan10
 ip address 10.210.0.1 255.255.0.0
!
ip http server
ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 10.1.1.2
!
ip access-list standard SNMP_ACL
 permit 10.210.10.5
!
snmp-server community projectsnmp RO SNMP_ACL
snmp-server location rack1
snmp-server contact yihuan.zhang@dawsoncollege.qc.ca
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps flowmon
snmp-server enable traps transceiver all
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps cluster
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps cpu threshold
snmp-server enable traps power-ethernet police
snmp-server enable traps rep
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps flash insertion removal
snmp-server enable traps port-security
snmp-server enable traps auth-framework sec-violation
snmp-server enable traps dot1x auth-fail-vlan guest-vlan no-auth-fail-vlan no-guest-vlan
```

```
snmp-server enable traps envmon fan shutdown supply temperature status
snmp-server enable traps event-manager
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps energywise
snmp-server enable traps vstack
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps ipsla
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ike tunnel start
snmp-server enable traps ike tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps mac-notification change move threshold
snmp-server enable traps vlan-membership
snmp-server enable traps errdisable
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
snmp-server host 10.210.10.5 version 2c projectsnmp
!
!
line con 0
line vty 0 4
 login local
 transport input ssh
line vty 5 15
 login
!
ntp server 10.210.10.9
end
```

ASA running configuration:

```
asa# show run
: Saved
:
ASA Version 8.4(6)5
!
hostname asa
domain-name dawsoneet2019.com
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
names
!
interface Ethernet0/0
 switchport access vlan 2
!
interface Ethernet0/1
 switchport access vlan 3
!
interface Ethernet0/2
!
interface Ethernet0/3
!
interface Ethernet0/4
!
interface Ethernet0/5
!
interface Ethernet0/6
!
interface Ethernet0/7
!
interface Vlan1
 nameif inside
 security-level 100
 ip address 192.168.1.2 255.255.255.0
!
interface Vlan2
 nameif outside
 security-level 0
 ip address dhcp
!
interface Vlan3
 no forward interface Vlan1
 nameif DMZ
 security-level 50
 ip address 192.168.2.1 255.255.255.0
!
ftp mode passive
dns server-group DefaultDNS
 domain-name dawsoneet2019.com
object network obj_any
 subnet 0.0.0.0 0.0.0.0
```

```
object network inside-net
 subnet 192.168.1.0 255.255.255.0
pager lines 24
logging asdm informational
mtu inside 1500
mtu outside 1500
mtu DMZ 1500
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
no arp permit-nonconnected
!
object network obj_any
 nat (inside,outside) dynamic interface
object network inside-net
 nat (inside,outside) dynamic interface
route outside 0.0.0.0 0.0.0.0 10.210.0.1 1
timeout xlate 3:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
dynamic-access-policy-record DfltAccessPolicy
user-identity default-domain LOCAL
http server enable
http 192.168.1.0 255.255.255.0 inside
snmp-server location Rack1
snmp-server contact yihuan.zhang@dawsoncollege.qc.ca
snmp-server community *****
snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
telnet timeout 5
ssh timeout 5
ssh key-exchange group dh-group1-sha1
console timeout 0

dhcpd auto_config outside
!
threat-detection basic-threat
threat-detection statistics access-list
no threat-detection statistics tcp-intercept
webvpn
username admin password Q6cxZHFxNEbArjj8 encrypted
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
```

```
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect esmtp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
  inspect ip-options
  inspect icmp
!
service-policy global_policy global
prompt hostname context
Cryptochecksum:db4ac3e831607ff38ec5ddaf21fba547
: end
```

Project Work Hours:

Part:	Work on:	Hours:
Hardware section	System designing	4
	Sensors testing	4
	Programming & troubleshooting	8
	Embedded PCB designing	4
	Embedded system soldering	2
	Power supply designing	2
	Power supply testing	4
	Power supply PCB designing	1
	Power supply soldering	1
Hardware sub-total		30
RPi section	Serial port connection	2
	Python code programming	6
	Web server establishing	4
	Integrating (shell script, cron, PHP)	4
RPi sub-total		16
Network section	Network designing	1
	Rack installing	4
	Network configuring	4
	Console server	1
	*Python script (SDN)	2
Network sub-total		12
Server section	Hypervisor implementing	1
	SNMP	3
	NTP	4
	DC/DNS/Radius	4
	DHCP	2
	Web server (docker)	4

	*NFS	0
Server sub-total		16
Total hours		74

Project Hardware Cost:

Part:	Components:	Price (CAD):
Embedded system	MCU	2
	ICs	10
	LCD	15
	Copper board (double side)	8
	Sensors	30
	Others (sockets resistors capacitors)	5
Sub-total		70
Power supply	Regulator (IC)	3
	Inductor	3
	Transformer	20
	Copper board (single side)	6
	Others (diodes, wires)	5
Subtotal		37
Raspberry Pi	Pi 3B+	60
	Buzzer	2
	SD card	10
Sub-total		72
Total cost		179