Dawson College – Electronics Engineering Technology
243-698-DW Computer Network Project

# Project Proposal

# Configuring and Monitoring System for Server Room

**Student Name: Yihuan Zhang**
**Student Number: 1630783**
**Date created: October 28, 2018**
**Date modified: January 24, 2019**
**Submitted to: Nick Markou**

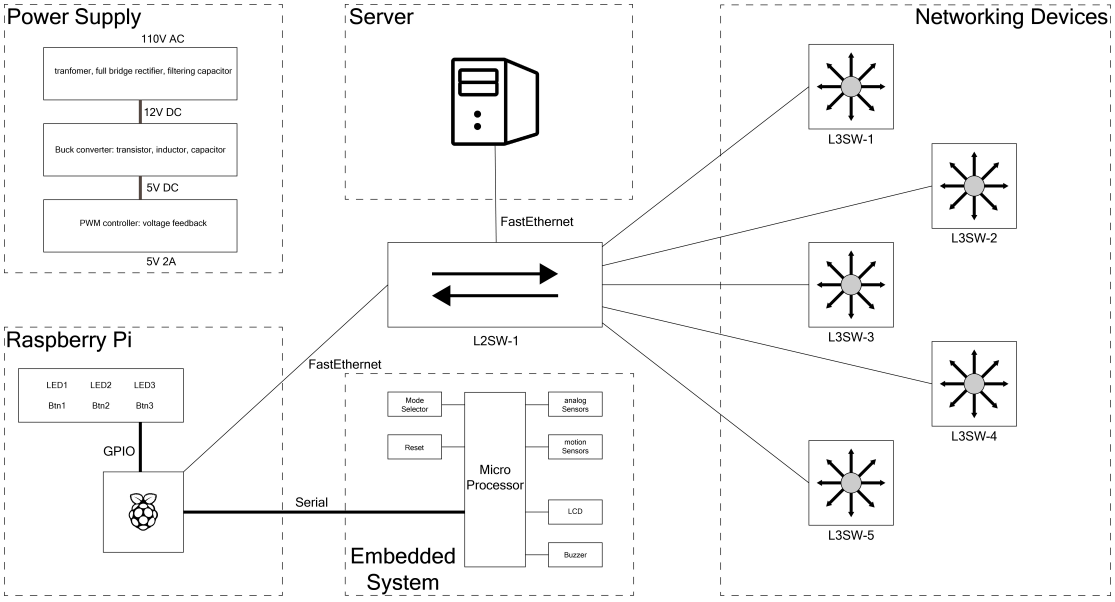
# Description

In most network based companies (or organizations that have campus networks) like ISP, school, government, system integrator, outsourced provider and social media data center, the server room is the essential part that requires a lot of resource to maintain, including checking the physical environment and device's statuses. Whether the network devices (servers, databases, routers and switches) function properly or not, there are always technicians doing routine maintenance tasks everyday to ensure that bugs can be found as early as possible. Based on such scenario, there are different types of sensors or detectors that exist in electronic world that can replace most of the maintenance tasks technicians regularly do. This project is an automatic controlling system used in DevOps domain based on Embedded system and Software Defined Network.

By taking advantages from embedded hardware system, the technicians or the team who are responsible for a certain server room do not need to be physically presented there (in the actual location) to check the potential hazards but can inspect through the data returned from sensor units. Each unit contains various sensors that achieve different purpose, and the units placed in the server room can also be programmed or scheduled from the terminal, the web page, the controller, or the state machine according to the specific requirement.

Since one significant part of the technicians' job is configuring the networking devices using console and command line interface, these are duplicated tasks which can also, at least partially, be substituted by scripts. Implementing the Configuring and Monitoring System reduce the use of human resource on modern server rooms, thus the efficiency of the network operation/maintenance increases.

# Block Diagram

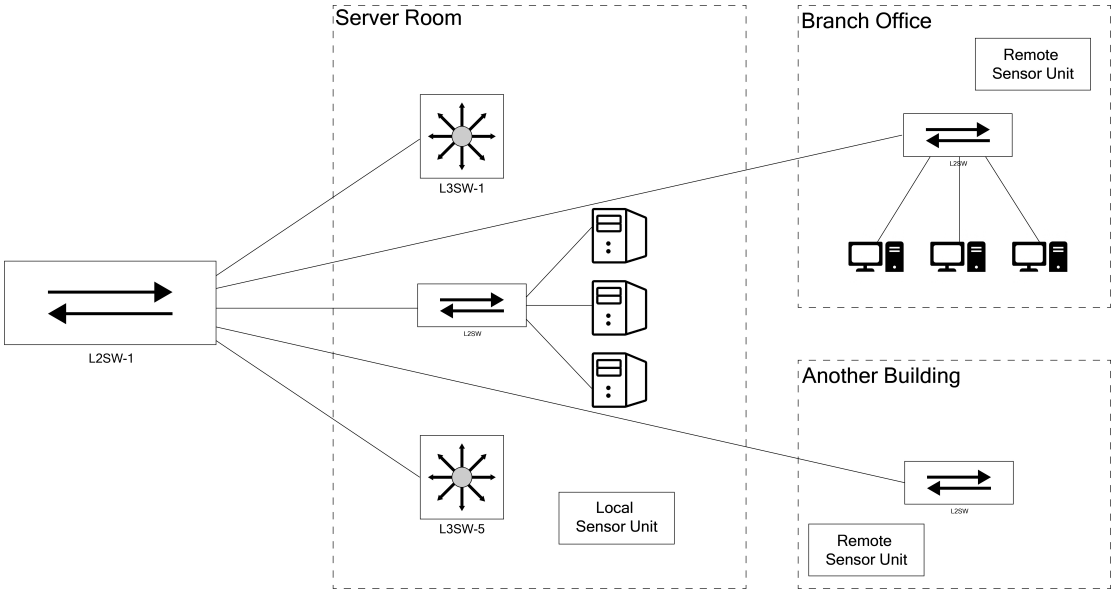



Before introducing the block diagram, keep in mind that everything above should be implemented in the server room, and the targets are to monitoring the connectivity and physical status of networking devices as well as providing necessary notification. The right block represents the routers and switches we need to monitor or configure, they all are connecting to a layer 2 switch and their corresponding ports needs to be assigned to the administrative network; in the Embedded System block, a microprocessor where multiple sensor units attached to communicates to a Raspberry Pi through a serial interface; the Pi takes environment data from its serial pins, and grabs the results of "show" or "ping" commands from

the NFS server, and uses the wireless connection to exchange simple instructions or statuses with the controller; the controller drove by a portable power supply uses switches or buttons sending simple instructions to turn on/off the Embedded system, it also outputs a signal to notify the abnormal condition received from Pi; there is a vm running in the server part, it is in the administration network so that it is able to SSH into any of the networking devices as well as sharing the results with Pi.

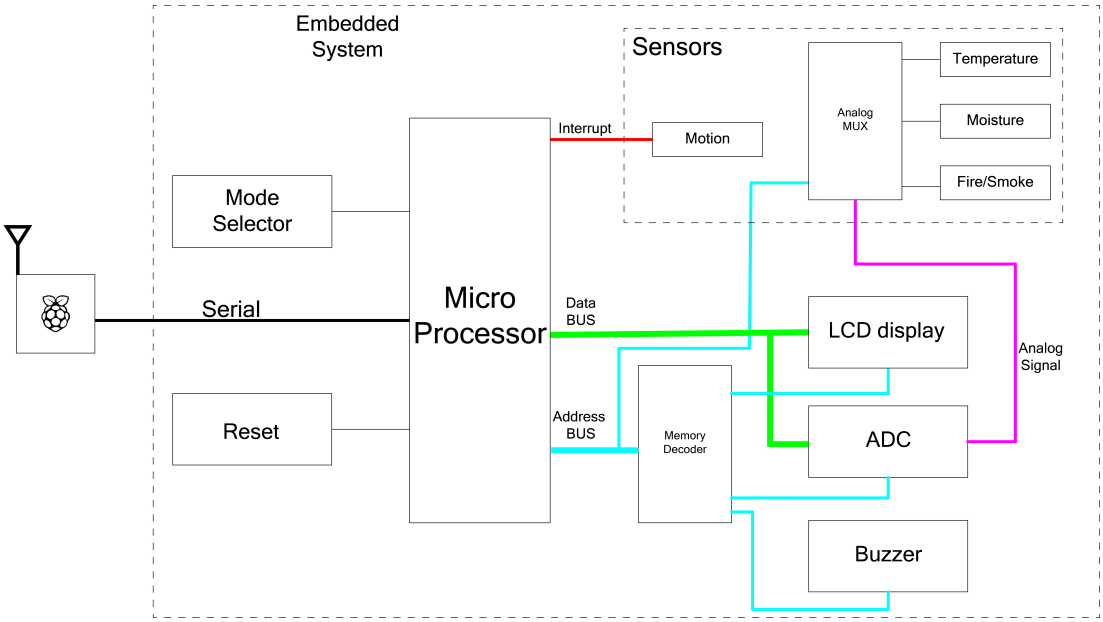
# Block Description

## Networking Devices




This block contains all the devices and instruments in the server room (or outside the building). Though a layer 2 switch (L2SW-1), these devices are accessible by an external server using SSH.  Besides, The sensor units that contain different sensors are deployed close to them so that the change of their physical status can be monitored immediately.

## Embedded System




In the Embedded System block, the Micro Processor is configured as BUS mode and responsible for exchanging data (from sensors) with the Pi. One of the main purposes in this system is data acquisition. Sensors attach to the Micro Processor through an analog multiplexer and return the digital value back to the Data Bus. The analog MUX interfaces the sensors through its channels and selected by Address Bus. There is also a motion sensor that interfaces with the interrupt directly, to monitor any unauthorized entrance in network security concern.

In the beginning of the program, a dip switch connected to an available port pin acts as a mode selector which selects the level of strictness or sensitiveness. The mode selector determines the reactions under the Interrupt subroutines as well as affects the triggering level in serial communication with Pi.

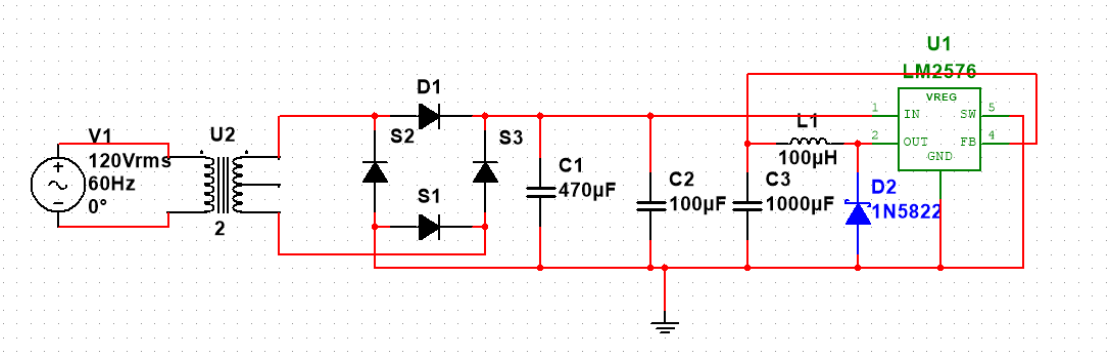
Then, the LCD will display digitalized data returned from sensors correspondingly. Motion sensor will activate the interrupt to send alert message to the RPi as well as turn on the buzzer until the embedded system being reset.

## Raspberry Pi

The main function of the Pi is acting as a web server that provides the user interface. Setting the Crontab file to execute certain scripts periodically so that the log information can be taken and shown on a web page dynamically. Specifically, the “HTTPd (Apache)”, “MySQL”, “NFS”, and “Python (GPIOzero)” services are enabled on Pi, the serial pins on GPIO are used to communicate with the Micro Processor, a few parallel pins attached with an external antenna (or I.R. Comm.) to communicate with controller, it uses RJ45 connecting the administrative network thus the scripts and the results on the server is also accessible.

The Raspberry Pi has LEDs and buttons attached on it through its GPIO. The LEDs indicates the status of embedded system, the server, and the services running on Raspbian. Buttons are used to send reset command to each block system.

## Power Supply




To provide DC power to the Raspberry Pi and the embedded system, a 5V2A switching power supply is required. In general, the procedure of transforming 110V 50Hz AC main into a 10Watts 5V2A DC power has two parts. The first part is bringing down the voltage using a step-down transformer and regulating the voltage by a rectifier and a capacitor; second part is to bring down the voltage further more and adjust the PWM control signal by the output voltage feedback. A switching power supply maximizes the power efficiency comparing with a linear power supply hence it has been chosen in this application.

## Server

As part of the configuring and monitoring system, the server's job is to allow technicians configuring the routers and switches remotely, and checking the connectivity or even providing the corresponding solutions. To achieve it, a virtual machine with installing python libraries such as Paramiko, Netmiko (for SSH) or Telnetlib (for telnet) needs to be installed. After executing the python program, the results or the log save into a directory where is also shared with the Pi. For instance, one of the program is used to check the interface status, the vm would run to issue the "show ip int brief" on each switches, then grep and save results as a log file; the Pi takes the log from the shared directory, later it will send a notification to the controller once it notices that certain important ports are down.

In addition, we can also use docker engine to deploy softwares or services (docker images) inside the containers (in stead of virtual machines) on the actual server in convenience of development, immigration and maintenance.

# Software Description

- IOS commands: networking devices, version 14.2 or higher
- Assembly: Microprocessor
- WinCUPL: PLD in embedded system
- Multisim&Ultiboard: Sensor unit, power supply circuit, wireless module
- Verilog: FPGA in controller
- Hyper-V: server
- CentOS: VM on the server
- Python: on the CentOS and on the Raspberry Pi

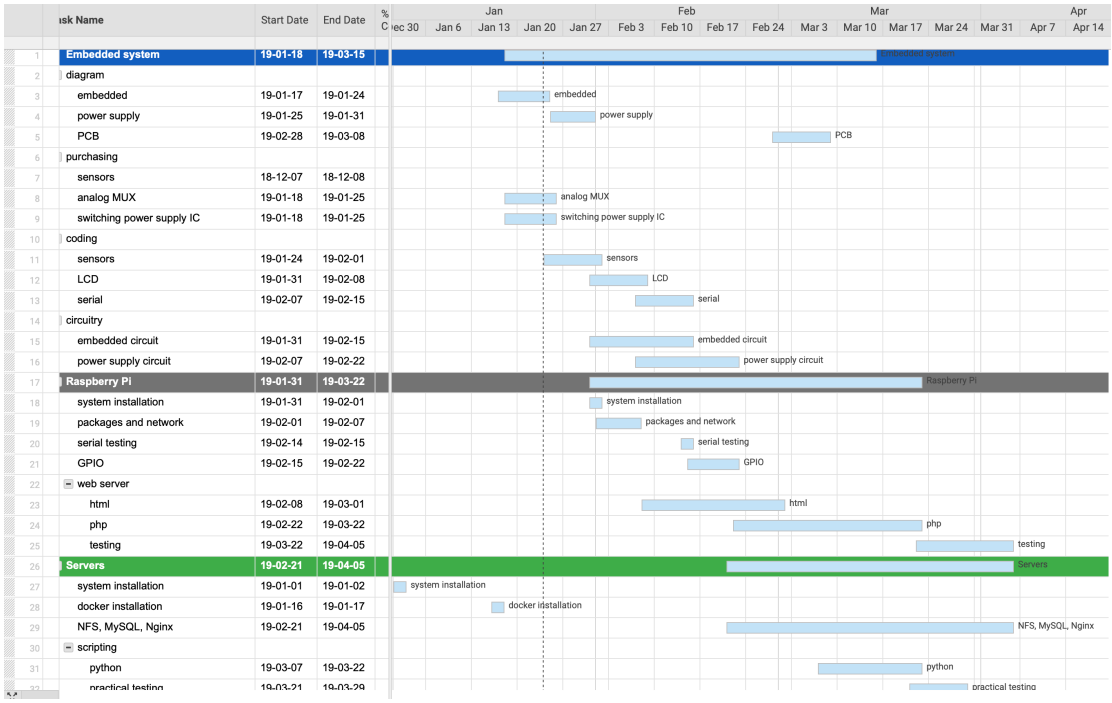
# Course Inventory

- Client Based OS and Server Environment: to install and manage the VM on the server.
- Unix I&Unix II: to script and operate the raspberry Pi and the VM.
- Fundamentals of Web Servers: httpd service on the Pi, PHP, GPIO, Python.
- Embedded: microprocessor, serial transmission.
- Wireless LANs: wireless transmission between Pi and controller.
- Power Supplies: the switching circuit and DC-DC buck converter circuit.
- Electronic Techniques: PCB designing and soldering.
- Network Routing: switch configuration.
- Network Security: physical access control, AAA server
- Network Planning: server room management, rack installation
- LAN switching and Configuration: router configuration.

# References

- Python SSH tutorial and how to connect to Cisco devices: https://www.ictshore.com/sdn/python-ssh-tutorial/
- Paramiko website: http://www.paramiko.org
- Tenetlib documentation: https://docs.python.org/2/library/telnetlib.html
- DIY buck converter: https://www.youtube.com/watch?v=m8rK9gU30v4

# Gantt chart




| | Task Name | Start Date | End Date |
|---|---|---|---|
| 1 | Embedded system | 19-01-18 | 19-03-15 |
| 2 | diagram | | |
| 3 | embedded | 19-01-17 | 19-01-24 |
| 4 | power supply | 19-01-25 | 19-01-31 |
| 5 | PCB | 19-02-28 | 19-03-08 |
| 6 | purchasing | | |
| 7 | sensors | 18-12-07 | 18-12-08 |
| 8 | analog MUX | 19-01-18 | 19-01-25 |
| 9 | switching power supply IC | 19-01-18 | 19-01-25 |
| 10 | coding | | |
| 11 | sensors | 19-01-24 | 19-02-01 |
| 12 | LCD | 19-01-31 | 19-02-08 |
| 13 | serial | 19-02-07 | 19-02-15 |
| 14 | circuitry | | |
| 15 | embedded circuit | 19-01-31 | 19-02-15 |
| 16 | power supply circuit | 19-02-07 | 19-02-22 |
| 17 | Raspberry Pi | 19-01-31 | 19-03-22 |
| 18 | system installation | 19-01-31 | 19-02-01 |
| 19 | packages and network | 19-02-01 | 19-02-07 |
| 20 | serial testing | 19-02-14 | 19-02-15 |
| 21 | GPIO | 19-02-15 | 19-02-22 |
| 22 | web server | | |
| 23 | html | 19-02-08 | 19-03-01 |
| 24 | php | 19-02-22 | 19-03-22 |
| 25 | testing | 19-03-22 | 19-04-05 |
| 26 | Servers | 19-02-21 | 19-04-05 |
| 27 | system installation | 19-01-01 | 19-01-02 |
| 28 | docker installation | 19-01-16 | 19-01-17 |
| 29 | NFS, MySQL, Nginx | 19-02-21 | 19-04-05 |
| 30 | scripting | | |
| 31 | python | 19-03-07 | 19-03-22 |
| 32 | practical testing | 19-03-21 | 19-03-29 |

# Circuit diagram