NANYANG TECHNOLOGICAL UNIVERSITY
Energy Research Institute @ NTU

SPMS Sub-Metering Data Collection Program (on BBB)

Revision

Revision	Date	Description	Name	Approved
1	09/07/2014	Initial Version	Wang Xiaochen (Harry)	

Remark: If any one changes this document, please approach Wang Xiaochen (Harry) or Zhang Longqi for approval.

Contents

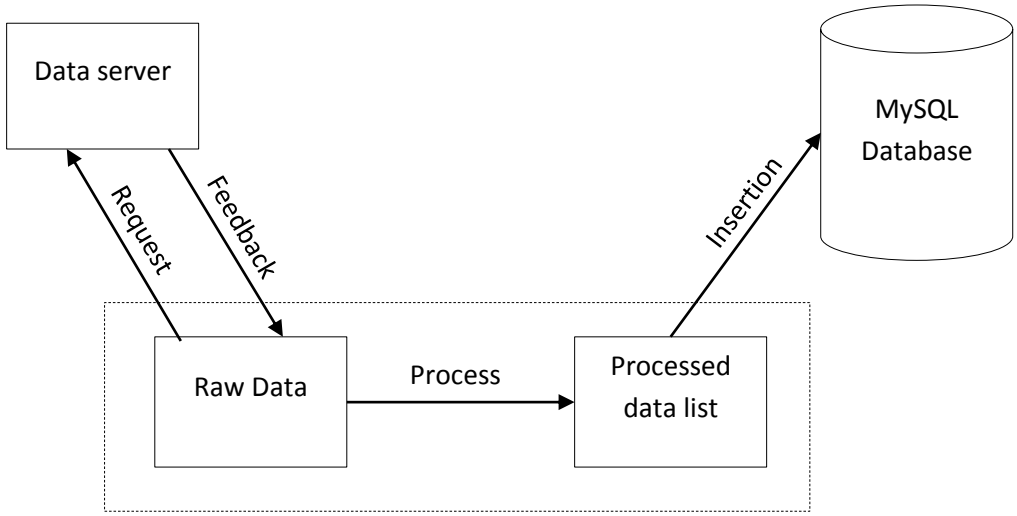
1. Overview

The program described here is in charge of the data collection section of SPMS Sub-Metering project for both newly installed measurement devices (power meter, BTU meter etc.) as well as on top of the existing system, and this program is running on BeagleBone Black and collect the data every 1 minute.

The program described here is suitable with the existence of a Modbus server (either series or TCP/IP) which could response accordingly to the request sent by the program.

1.1 Service Description

The major process of this program has been shown in the block diagram below:



Data Collection Program

The program would request the data from different measurement devices to the "data server" according to the IP, port number and register address, then process the raw data and keep the processed data in a temporal list, after all the meters have been gone through, the data in the list would be inserted into MySQL database. Then program would wait for another insertion according to the pre-set timer.

1.2 Technology

The program is written in Java and would be running on Lunix Debian on BeagleBone Black board with NTU network access.

1.3 Development Tools

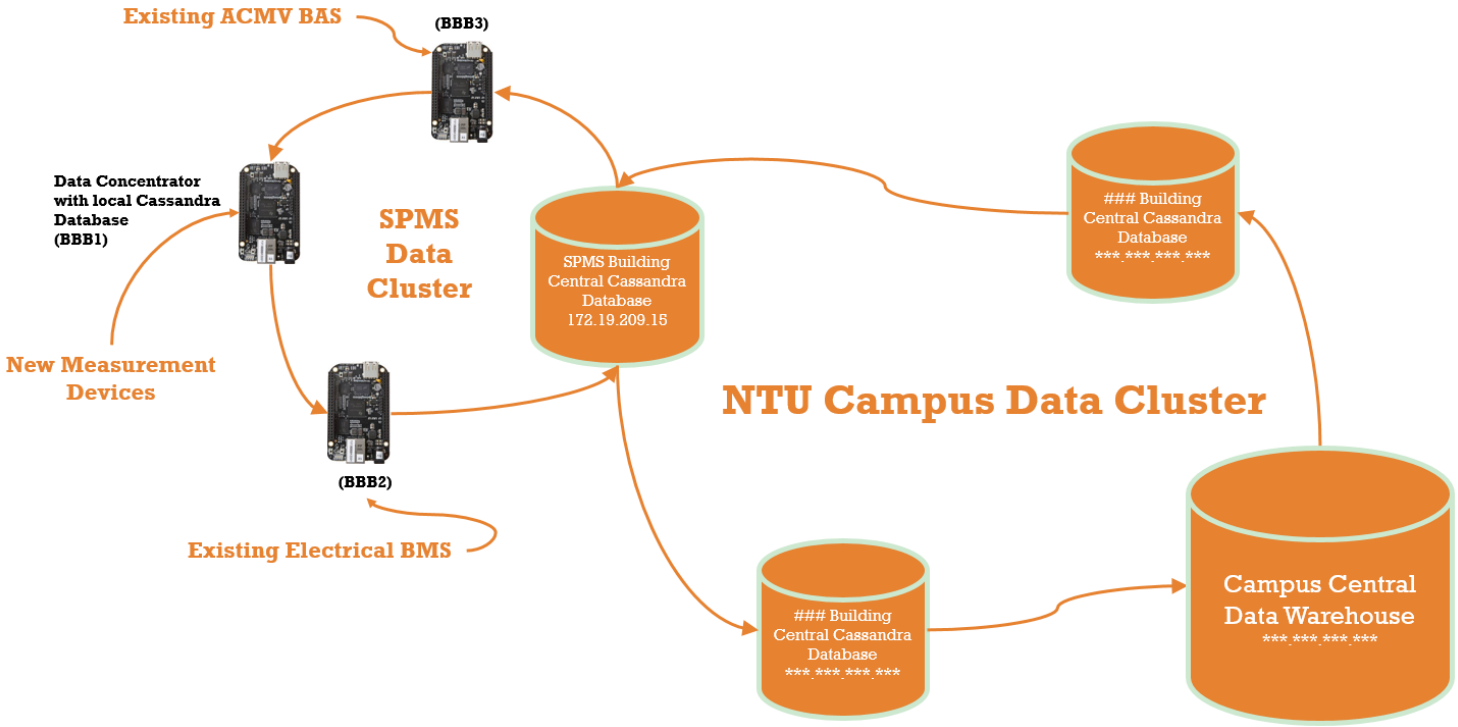
The development environment implemented here is: NetBeans IDE 8.0 Beta.

1.4 Access, Authentication and Authorization

In the command line -> go to project folder -> go to the folder "dist" -> run the program by keying "java –jar PROGRAM_NAME.jar".

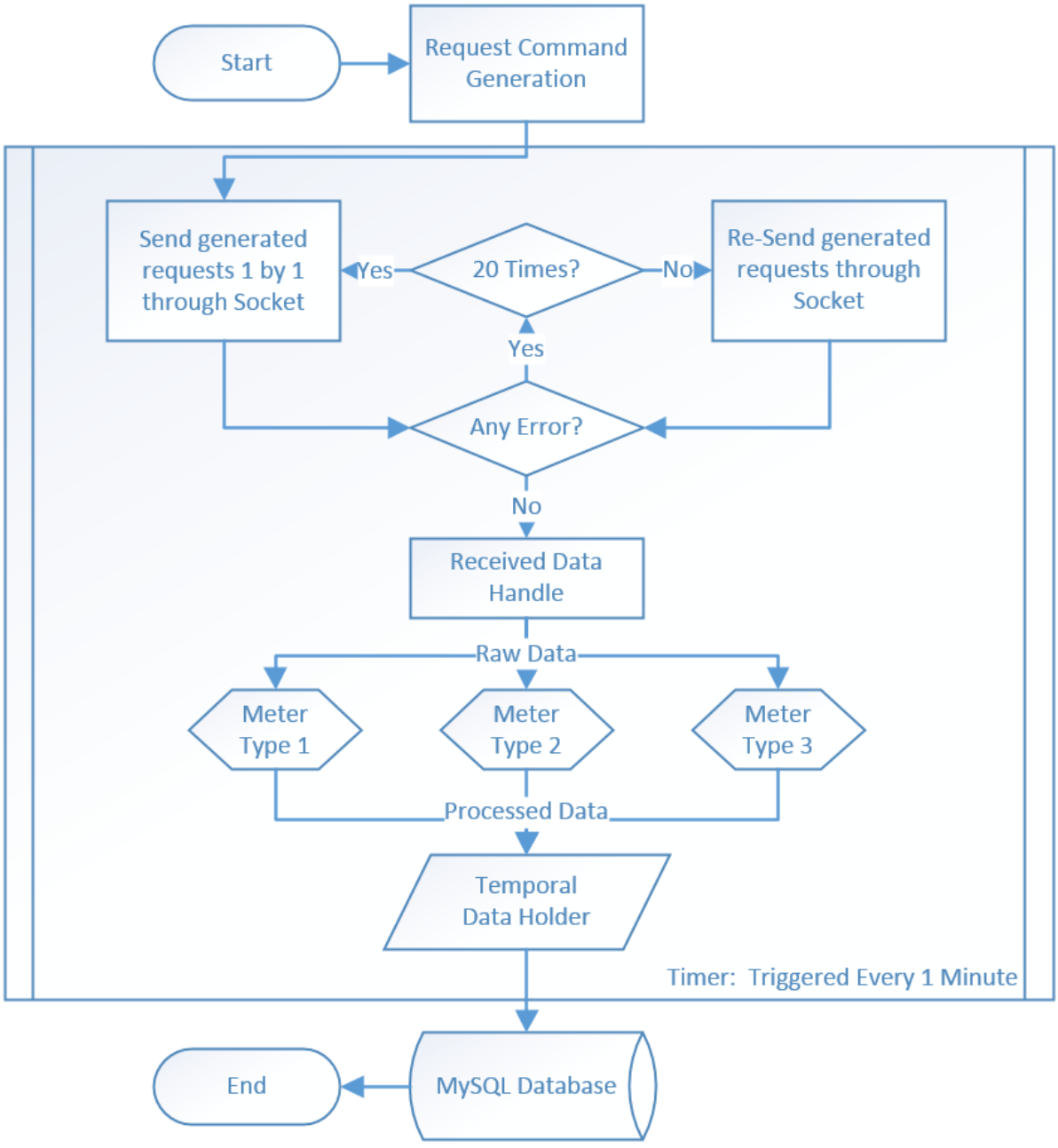
1.5 Delivery

This data collection program could be part of the practice case study on Eco-Campus campus wide data collection implementation. In the future, local Cassandra database would be applied and synchronize the data depending on the database configuration, all the data would be centralized to the campus data center for further processing, analysis and visualization.



2. Program Flowchart

The general flow chart of the collection program with existence of Modbus server has been shown below.



The timer is set 1 minute which means the process in the big block would be triggered every 1 minute, which is considered the plan data frequency that could meet most of the applications.

3. Explanations

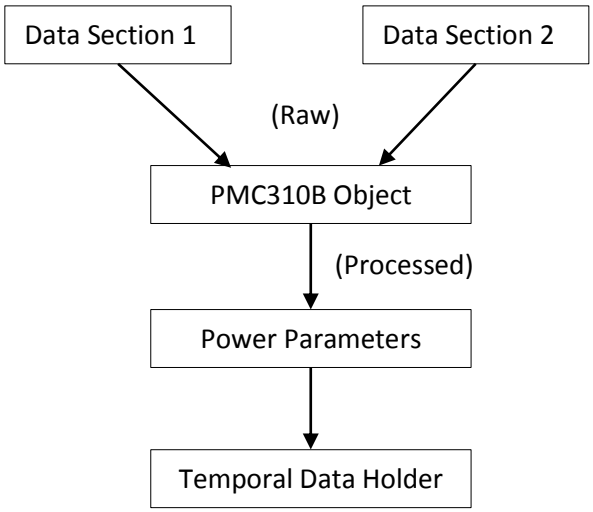
There is certain replied data check according to the request sent in this program (welcome to improve), and they are listed below:

- Received data too short;
- Received data does not have the function code in the request;
- Received data length is incorrect;
- According IO exceptions.

Once the error detected, program would re-send the request that failed every 0.5 seconds until reach 20 times, then program would jump to next request and the former data is lost in this cycle.

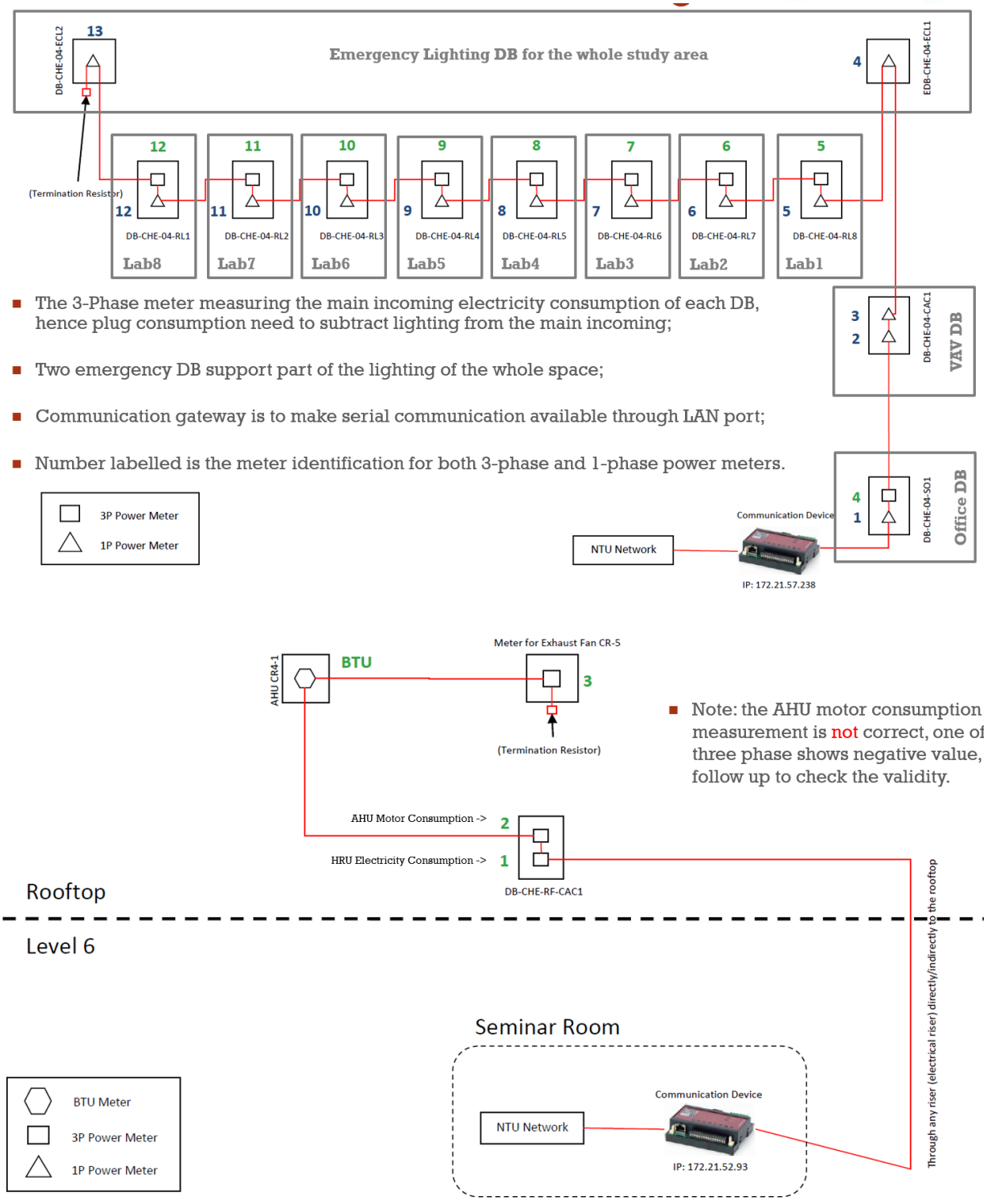
The received correct data would be handled based on either “slave ID”, “received data length” or “transaction ID (TCP)” to allocate to each meter object for further processing and stored in a temporal data holder. Once all the data request session has been finished, the data holder would insert all the processed data to the different MySQL database tables, then we can consider one data collection cycle has been finished (1 minute).

For different measurement devices, the data section we’re focusing in the device register table might be far away from each other, therefore in this condition, several requests for one device would be necessary. Take the device “PMC310B” as an example:



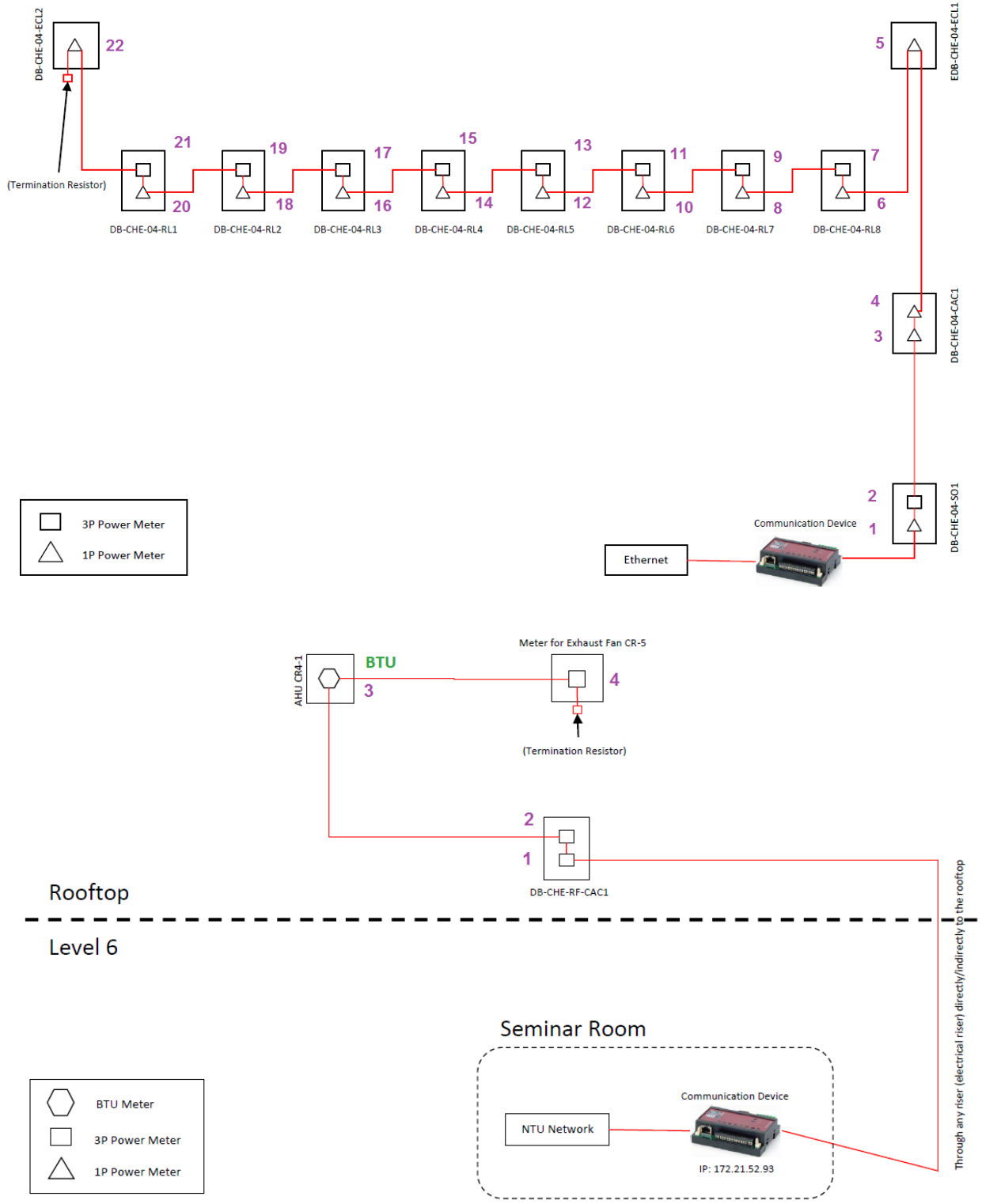
3.1 Additional Measurement Devices

We applied Modbus RS485 in additional measurement devices data collection on both level 4 and rooftop, the two figures shown below illustrate the Daisy chain structure:



The gateway we've used is to make the serial communication available through the NTU network (not to translate serial to TCP/IP communication). The two gateways have both allocated with fixed IP address

with a special port number (10002) is configured by the provider to access. Hence, in this case, Modbus RS485 request need to be sent for the data query and the slave number allocation is shown below:

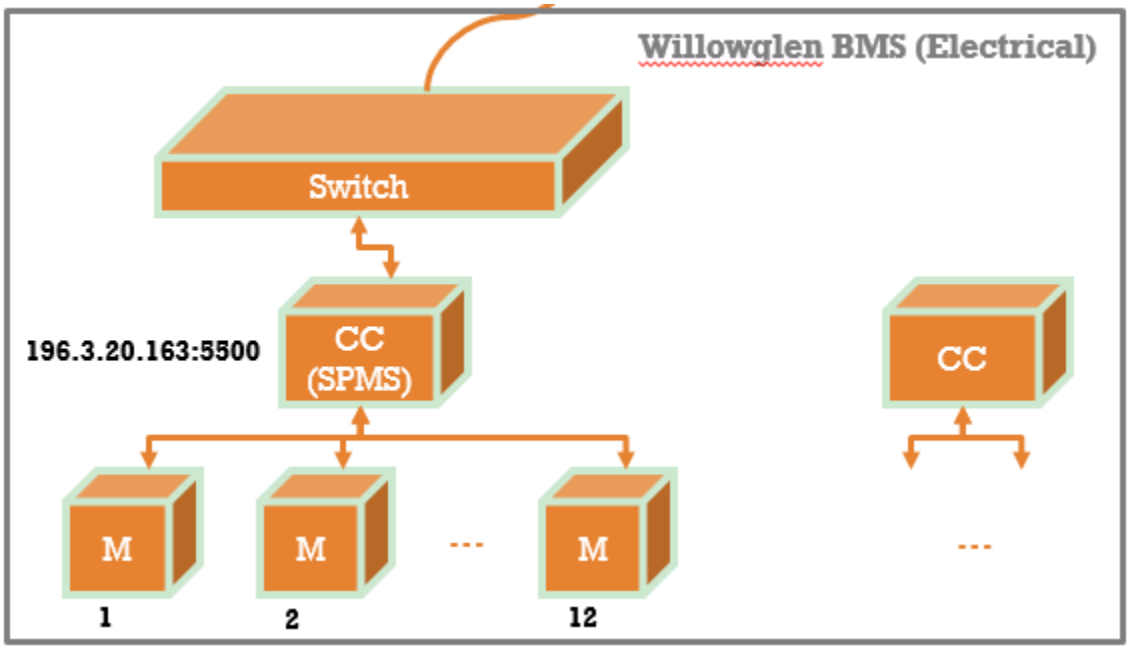


3.2 Existing Building System (Electrical-Willowglen)

The Willowglen system rough structure can be shown as the graph below, their engineer configured a port which support Modbus TCP/IP standard and the program would send TCP request in this case.

The network of the client device (send request for data) should be set to the following setting to access the CC shown in the figure below:

- Address: 196.3.20.120
- Netmask: 255.255.255.0
- Network: 196.3.20.0



There are 12 meters whose slave ID has already been defined as the table below:

Slave ID	Name of Meter/Location
1	SPMS MSB1-1 Meter
2	SPMS MSB1-2 Meter
3	SPMS MSB1-3 Meter
4	SPMS MSB2-1 Meter
5	SPMS MSB2-2 Meter
6	SPMS EMSB-1 Meter
7	SPMS SSB-1 Meter
8	SPMS SSB-2 Meter
9	SPMS ESSB-1 Meter
10	SPMS IC-SSB1 Meter
11	SPMS IC-SSB2 Meter

12	SPMS H7-LT Meter

While the locations of the meters need to be further investigated, while the data is kept on collecting every 1 minute.

3.3 Existing Building System (ACMV-Quantum)

There is no way to have direct access to the actual meter through their intranet and they just save the data in a shared file for storage purpose. Hence the method of data collection from this system is to have a continuous monitoring on all the data files, once the file has been modified, it would trigger the event to read the most recent record in the file and kept in the temporal data holder.

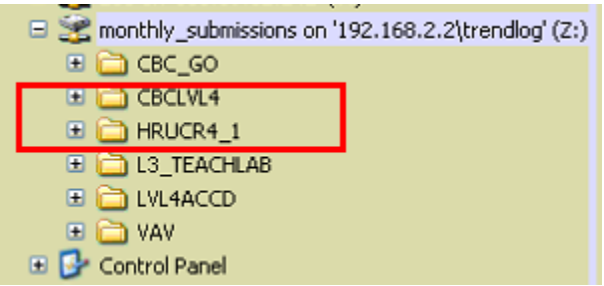
The network of the client device (send request for data) should be set to the following setting to access their shared drive for data storage:

- Address: 192.168.2.200
- Netmask: 255.255.0.0

And then mount a network drive with the following IP address and folder directory:

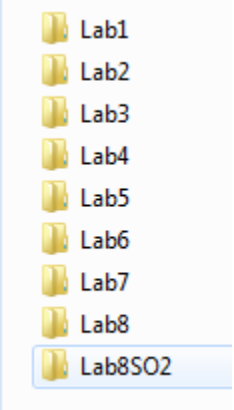
\\192.168.2.2\trendlog\monthly_submissions

And the folder “CBCLVL4” and “HRUCR4_1” are the target folders:

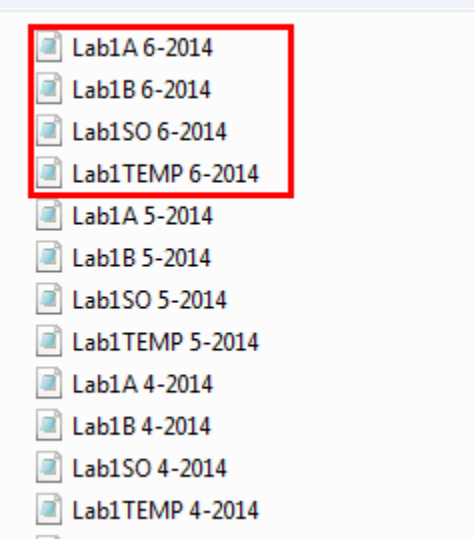


3.3.1 CBCLVL4

There are 9 folder in this folder:



And in the first 8 folder, 4 recent DAT files we need to take care of as highlighted in the figure below, and only these 4 files will change during the system operation:



Normally, the file would be named with "Lab[LAB_NUM][Identifier] [CURRENT_MONTH]-[CURRENT_YEAR]".

File Name	Parameters	Remarks
Lab1A	Date Time FCV1 (CMH) FCV2 (CMH) FCV3 (CMH) FCV4 (CMH) Double Gang Supply (CMH)	Lab airflow part A
Lab1B	Date Time FCV5 (CMH) FCV6 (CMH) FCV7 (CMH) FCV8 (CMH) Single Gang Supply (CMH)	Lab airflow part B
Lab1SO	Date Time Temp (Deg C) BoxFlow (CMH)	Office temperature and airflow
Lab1Temp	Date Time Temp (Deg C)	Lab temperature

The Lab1A, Lab1B and Lab1Temp combined according to the date and time with the addition on lab ID, and then inserted to the "lab" table in MySQL database. The Lab1SO would be inserted to the "so" table with the addition of SO_ID (lab1->so1, lab8so2->so9).

Moreover, the lab total air supply and exhaust and office total air supply should be calculated as following for the insertion of table "airflow" in the database:

- Lab total air supply:

$$Lab1Air_{TotalSupply} = DoubleGang + SingleGang$$

…

$$Lab8Air_{TotalSupply} = DoubleGang + SingleGang$$

$$LabAir_{TotalSupply} = Lab1_{TotalSupply} + \cdots + Lab8_{TotalSupply}$$

- Lab total air exhaust:

$$Lab1Air_{TotalExhaust} = FCV1_{Lab1} + FCV2_{Lab1} + \cdots + FCV8_{Lab1}$$

…

$$Lab8Air_{TotalExhaust} = FCV1_{Lab8} + FCV2_{Lab8} + \cdots + FCV8_{Lab8}$$

$$LabAir_{TotalExhaust} = Lab1Air_{TotalExhaust} + \cdots + Lab8Air_{TotalExhaust}$$

- Office total air supply:

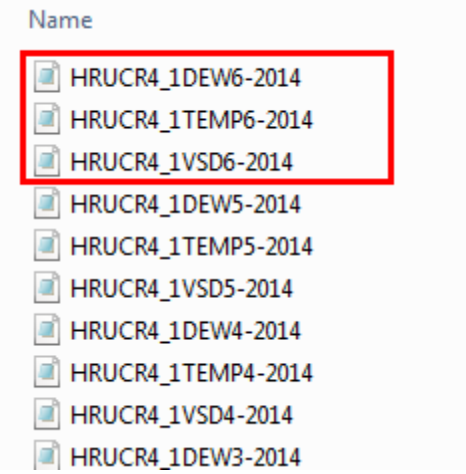
$$OfficeAir_{TotalSupply} = Boxflow_{SO1} + Boxflow_{SO2} + \cdots + Boxflow_{SO9}$$

Additionally, the average temperature of the office area should be calculate as below and inserted into table "so_average" together with the total office air supply in to the database:

$$OfficeTemp_{Average} = \frac{Temp_{SO1} + Temp_{SO2} + \cdots + Temp_{SO9}}{9}$$

3.3.2 HRUCR4_1

In this folder, 3 most recent file we should take care of as highlighted in the figure below:



And combine all the parameters in the three files together according to the same date and time, and then insert to the table "ahu" in MySQL database.

File Name	Parameters	Remarks
HRUCR4_1DEW	Date Time Dew Point (Deg C) Valve Position (%)	HRU Dew Point temperature and valve position parameters
HRUCR4_1TEMP	Date Time HRU Temp (Deg C) HRU Temp SetPt (Deg C) HRU Relative Humidity (%) HRU RH SetPt (%) KwH (kWh)	HRU temperature and humidity parameters
HRUCR4_1VSD	Date Time SP (Pa) VSD Control (Hz) VSD Speed (Hz)	Variable Speed Drive control and monitoring parameters

P.S. The parameters here are all heat recovery unit parameters, hence the table name in database might be a little bit improper.

4. References

[1] SPMS Data Collection Scenario and Future Plan (one of the documentations in the same folder);
[2] Data Collection Program Structure (one of the documentations in the same folder);
[3] SPMS Meters Register Data Request (one of the documentations in the same folder).