**HYDEL DOCUMENTATIONS**

**List of Hydel Stations:**

1. Nagjhari

2. Gerusoppa

3. Supa

4. Gattaprabha

5. Mani

6. Almatti

7. Munirabad

8. Varahi

9. Simsha

10. MGHE

11. Kadra

12. Bhadra

13. Kodasalli

**Types of Meters We Use In Sites**

1. DLMS meters

2. Modbus meters

**TWO CATEGORY OF METERS BASED ON THEIR USE**

1. **Main meters** – These meters are the meters which calculate the energy data and transmitting (only DLMS meters)

2. **Check Meters** - These meters are known as backup meters it will take the readings and store it to the DB (data base) if there is any issue in the main meters the module will take the data stored by the check meters from the DB. This meters will place near to the all main meters.

**TYPES OF METERS BASED ON METERS PLACED AREAS**

**1. GENERATOR METERS** - The meters which are place in the power generation place (ONLY MAIN METERS)

**2. GT METERS** - The meters which are placed in the lines after stepping up the voltage (after the step up transformer) (ONLY MAIN METERS)

**3. LINE METERS** - The meters which are placed in the feeders (the feeders will used to distribute the power for e.g.: one line is transmit to several number of lines to different areas sometimes it will also import power from any other substations) (BOTH MAIN METERS AND CHECK METERS).

4. **AUXILIARY METERS** -These meters are used to calculate the energy consumed by the power generation plant for their own purposes (in the generation plants they will use generated power directly for some devices e.g. lights, fans, etc.) for calculating this power usage we placed the AUX meters in the generation place. Maximum AUX meters are DLMS meters (only Almatti and Gerusoppa uses MODBUS meters)

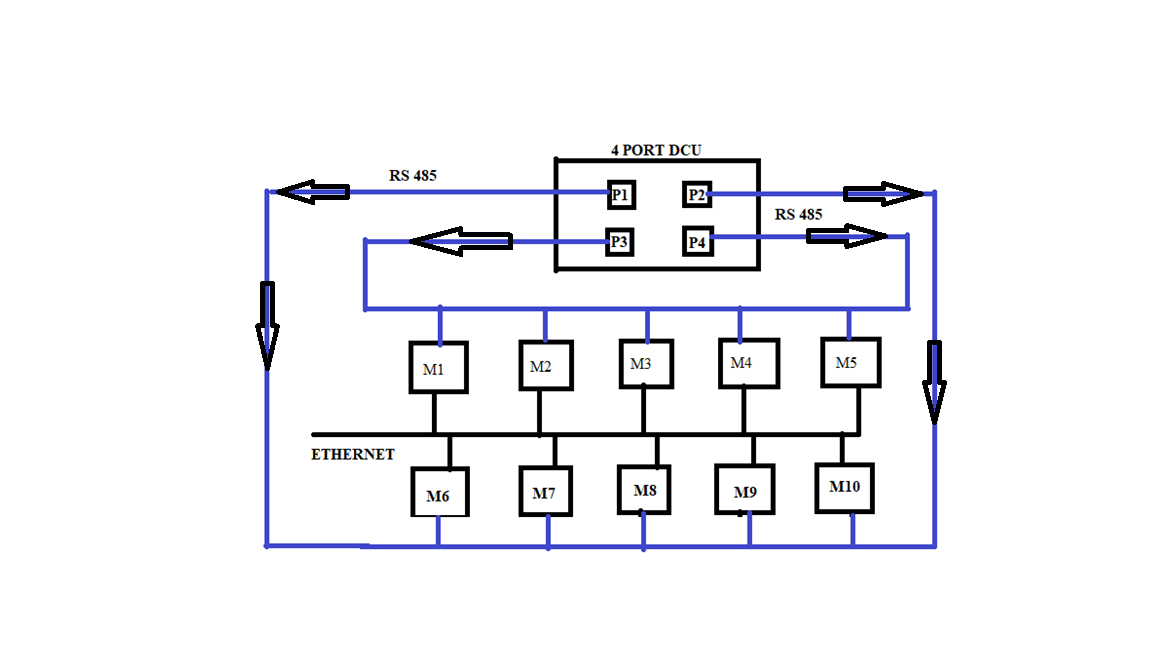
**NOTE:** All the aux meters we are using will have both Ethernet and Serial communication ports availability but we only use serial communication.

**NOTE:** In all sites our PC (SERVER) IP will be [192.168.1.10] and for generator meters starts from [192.168.1.100] and for GT meters starts from [192.168.1.110] and for LINE meters starts from [192.168.1.120]. Each meters have unique IP address by that we can access specific meters.

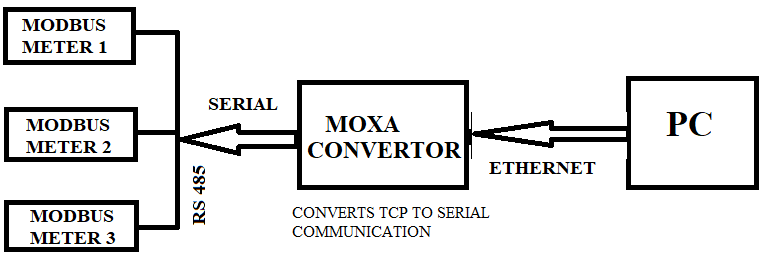
**NOTE:** The switches we are using in the sites can be use two gateways like 4 port can be in one gateway and another 4 ports will be any other gateway.

**DCU RING FIMWARE:**

This technique is used to overcome the communication loss due to communication cable damage. This is only used in serial communication in DCU. There will be two ports in each meters (DLMS) one for Ethernet and another one for serial communication. For this we will use 4 port (2MAIN PORTS and 2 STANDBY PORTS) DCU because only we configured 5 meters for each port. In this below diagram we can see that the first 5 meters are connected in series (RS485). The meter M1 is connected to the port P3 and the meter M5 is connected to the port P4 and same thing is for the 2nd meters set(6-10) so for example if there is a cable break between meter M3 and meter M4 means the port P3(main port) can’t able to poll the meter M4. So when the port P3 try to poll meter M4 and there is no response until it get timeout means then the polling will happen from port P4(Standby port). So by this the communication loss is eliminated between DCU and the meters.



AUX Modbus meters site connections (ALMATTI AND GERUSOPPA)



* **GATTAPRABHA**

There is no Ethernet meters in this sites .Only we have serial meters, connected to the DCU. So EMS (energy management system) server will take the data from the DCU via DA\_SERIAL module.

* **MUNIRABAD**

There are no line meters (feeder meters) only Generator and GT meters are there.

-----Still pending to include all other sites-----

**LIST OF HYDEL MODULES**

1. House-Keeping Module
2. DIAG Process
3. Alerts And Notification
4. File Generation Upload
5. Kerp DB Update
6. NTP Client
7. Network Health Check
8. DA Dcu
9. DA Ethernet (Reading Num Meters And Launching DA Ethernet For Each Meters)
10. DA Serial (We Are Not Using This Any Sites)
11. Report Backend Module
12. DA Aux Meter (Only For Almatti And Gerusoppa)
13. OPC Module (We Are Not Using This Any Sites)

**HOUSE KEEPING MODULE**

This module is used to trigger the other modules. If the other modules get killed, it will automatically restart the modules which got killed. It consists of all sites name and IP address of database and username and password. Billing enable /disable details and log file count, KERB db config, Aux meters table, MQTT config.

**WORKING**

**Step 1 :** It will read the DB related and Broker related configurations

**Step 2 :** Initialize the DB(data base) and connect to the DB. (Oracle or MySQL based the “**DBType**” variable in the config file)

**Step 3 :** Initialize the BROKER and connect to the BROKER.

**Step 4** : Launching all the MODULES.

**The below function are run continuously in the while loop**

**checkThresholds( )** 🡪 This will compare each meter data with the max and min data configured in the threshold config table via UI. If read data is goes above max data or goes below min data then it will send event message to the user through SMS, email, etc.

**checkOldLsData( )** 🡪 This is used to check if the up to date datum is present in DB or not. If one or two day’s datum missed due the link down or any other cause again it will again poll and get that missed data from the meter**. Note: Only for Gattaprabha we won’t provide this function because we are not supplied any main meters there.**

**UpdataHydelreport( )** 🡪 This is used to update the hydel report in the table format

**UpdateVirtualMeterData( )** 🡪 This is used to update virtual meter data to the backend (**NOT USING IT**)

**ProcessMonitor( )** 🡪 While launching each processes we have constructed name and process ID . In this function it will poll each by using its process id. If it returns **NONE** the process is running if return != NONE the process will remove and then it will launch again

**MemoryandCPUusage( )** 🡪 This is used to get how much memory and discspace is consumed in the device server because of the processes running. If the memory space or CPU space occupancy is goes above 70% then it will send event message to the user

**DBcleanUp( ) 🡪** There are list of DIAG tables in this function. It will delete more than 30 days Datum from the DB. This function will call every 6 hours.

**checkHydelReport( )** 🡪 we will display ABT report in the UI for updating that report we using this function. In the UI we have schedule management and declare management. Schedule management - the power transmitting station will give the list of how much power the generation station want to produce, the list is with the 15 mins interval. Declare management – This is list of power that the generation station can able to produce which they will give it to the transmission station.

**DIAG MODULE**

This module is used to just transmit the meter data to the DB via MQTT Protocol.

**REPORT BACKEND MODULE**

There are separate report structure for each site. This module is doing the backend report table creation for all the sites. First it will read the configurations from config file, then it will connect to the MQTT Broker.

**FindInstConsValue ( )** 🡪 this function will get the max and min parameters (voltage, load, frequency, current, pf, mvr, etc.) they generated on a particular day of each specific sites and make a report. We are storing every minute datum (voltage, load, frequency, current, pf, mvr, etc.) got from the meters **(**it will add all generator data i.e. all 5 generators produce 2Mw then it will store 10 MW) to **meter\_instant\_data** table, from that table we are taking every 5 mins data and storing it to the **station\_min\_max\_details**. **Note** : this function we implement for every sites

**FindInstMinMaxValue ( ) 🡪** from that **station\_min\_max\_details** table we getting the min and max values for voltage, act\_load, frequency. And we give to the UI to display the report. **Note**: this function we implement for every sites

**UpdateImpParameter ( ) 🡪** FindInstMinMaxValue ( ) function only get the min and max for voltage, load, frequency. But this function will calculate min and max for all the parameters (voltage, load, current, power factor, frequency, MVR) so it will use same procedure using in the FindInstMinMaxValue ( ) function. **NOTE:** this is only used in MANI site.

**UpdateAlmatiInstData ( ) 🡪** we want instantaneous parameters of each meters in separate page. Already we stored the instantaneous datum of each meters in **meter\_instant\_table.** For every month we will create billing report, this function is used to create the instantaneous report from the **meter\_instant\_table** at the time of generating billing report each month. **NOTE:** Billing data will generate at every first day of the month.

**DA ETHERNET MODULE**

Each meters will have separate Ethernet module. It will read the DB and broker details and connect with them. It will read the IP for the particular meter from the DB by using meter ID. After getting IP it will try to create a socket connection with that meter via Ethernet. Once failed it will retry every 10 seconds. After successful socket connecting it will poll the below things line by line after getting response.

1. Disc frame
2. ARQ will send
3. Name plate
4. Instantaneous
5. Midnight
6. Billing
7. Events
8. Current day Load survey
9. Old data (for helping housekeeping module to update missed data)

**DA SERIAL MODULE**

This module is same like the DA\_ETHERNET module only changes are instead of Ethernet query it will send serial query to meters via serial port other than that all the procedures are same like the DA\_ETHERNET module. **NOTE:** we are not using this module anywhere.

**DA DCU MODULE**

First initialize a socket connection with DB and broker. For each DCU separate DA\_DCU module thread will run. The listen port is **4050** same for all DCU and Device server communication. Every received message (Json format) will have the **‘!’** at the end of the whole message. The main thing of this module is to read the data (Json format) from the DCU and the Device servers and transmit it to the DB table. Based on the received **device id** it will either store the data to DCU table or Device server table. **NOTE**: **device servers is only used to communicate with Auxiliary meters**