

### **MQTT Broker List**

```
myMQTT(1) = mqtt('tcp://192.168.0.124:1883'); % V1
myMQTT(2) = mqtt('tcp://192.168.0.123:1883'); % V2
myMQTT(3) = mqtt('tcp://192.168.0.112:1883'); % V3
myMQTT(4) = mqtt('tcp://192.168.0.113:1883'); % I1
myMQTT(5) = mqtt('tcp://192.168.0.120:1883'); % I2
myMQTT(6) = mqtt('tcp://192.168.0.119:1883'); % I3
myMQTT(7) = mqtt('tcp://192.168.0.118:1883'); % Vn
myMQTT(8) = mqtt('tcp://192.168.0.117:1883'); % In
myMQTT(9) = mqtt('tcp://192.168.0.116:1883'); % Freq
myMQTT(10) = mqtt('tcp://192.168.0.115:1883'); % Load
myMQTT(11) = mqtt('tcp://192.168.0.114:1883'); % Subscribe
```

### **Feeders:**

**\*Install dashboard and table UI nodes if MQTT is not connecting to the feeders**

### **Raspberry Pi Allocations**

PLC - 192.168.0.102  
IED1 - 192.168.0.121  
IED2 - 192.168.0.117  
SCADA - 192.168.0.124

### **Bus 1:**

Feeder 1: 192.168.0.101  
Feeder 2: 192.168.0.103  
Feeder 3: 192.168.0.104  
Feeder 4: 192.168.0.105  
Feeder 5: 192.168.0.107  
Feeder 6: 192.168.0.108  
Feeder 7: 192.168.0.109  
Feeder 8: 192.168.0.110  
Feeder 9: 192.168.0.111

### **Bus 2:**

Feeder 10: 192.168.0.112  
Feeder 11: 192.168.0.113  
Feeder 12: 192.168.0.114  
Feeder 13: 192.168.0.115  
Feeder 14: 192.168.0.116  
Feeder 15: 192.168.0.118  
Feeder 16: 192.168.0.119  
Feeder 17: 192.168.0.120  
Feeder 18: 192.168.0.123

**Problem Raspberry Pis:**

192.168.0.106

192.168.0.122

### Database Parameters:

- Timestamp
  - Collects data every 10 minutes
- 9 input values
- CB status
- CB decision
- Events that occur
  - At every event, collect all data regardless of 10 minute cycle

### Database address (Stored locally in PI)

To access in PI,

```
cd /tmp                                // to directory
sqlite3 Input_data_ied1;
sqlite3 output_data_ied1;             // file names for 1-20, SCADA & Individual IED
select rowid, * from Input_data_ied1;
Physical location for ied1 >> /tmp/pi_s1_ied1
Physical location for SCADA historian >> /tmp/historian
```

To COPY and paste from SCADA node to auto delete for last 500 entries for all 20 ieds

If nodered shows EADDRESS 0.0.0.0 in use,

```
ps -e | grep -i node-red
Sudo kill process no.
```

## **MQTT**

Configuration: Matlab

Broker - 192.168.0.124

Matlab file consist of:

- Broker and supply configuration
- Hardcoded values for Supply 1 and 2
- MQTT channels/topics for hardcoded values
- Publish and subscribe settings
- Display status for both CB

### **Topics:**

#### **Supply 1**

Voltage Phase 1 - Matlab\_to\_IED1/S1\_Vphase1

Voltage Phase 2 - Matlab\_to\_IED1/S1\_Vphase2

Voltage Phase 3 - Matlab\_to\_IED1/S1\_Vphase3

Voltage Neutral - Matlab\_to\_IED1/S1\_Vn

Current Phase 1 - Matlab\_to\_IED1/S1\_Iphase1

Current Phase 2 - Matlab\_to\_IED1/S1\_Iphase2

Current Phase 3 - Matlab\_to\_IED1/S1\_Iphase3

Current Neutral - Matlab\_to\_IED1/S1\_In

Frequency - Matlab\_to\_IED1/S1\_frequency

#### **Supply 2**

Voltage Phase 1 - Matlab\_to\_IED2/S2\_Vphase1

Voltage Phase 2 - Matlab\_to\_IED2/S2\_Vphase2

Voltage Phase 3 - Matlab\_to\_IED2/S2\_Vphase3

Voltage Neutral - Matlab\_to\_IED2/S2\_Vn

Current Phase 1 - Matlab\_to\_IED2/S2\_Iphase1

Current Phase 2 - Matlab\_to\_IED2/S2\_Iphase2

Current Phase 3 - Matlab\_to\_IED2/S2\_Iphase3

Current Neutral - Matlab\_to\_IED2/S2\_In

Frequency - Matlab\_to\_IED1/S2\_frequency

## Modbus

Server location: 192.168.0.104

### PyModbus Setup

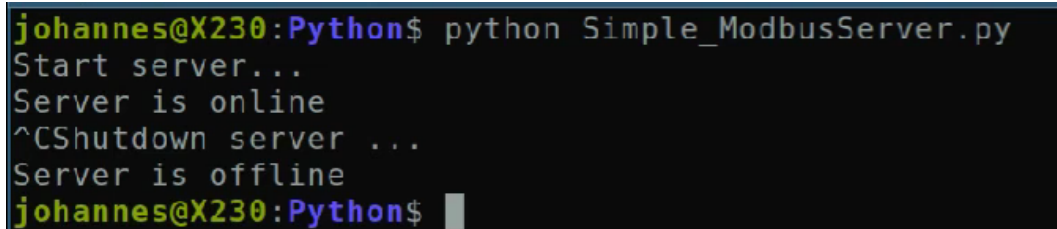
To activate: Go to command prompt/terminal

Type:

cd Python

python Simple\_ModbusServer.py

Example:

A terminal window with a dark background and light-colored text. The prompt is 'johannes@X230:Python\$'. The user enters 'python Simple\_ModbusServer.py'. The output is: 'Start server...', 'Server is online', '^CShutdown server ...', 'Server is offline', and the prompt returns to 'johannes@X230:Python\$'.

```
johannes@X230:Python$ python Simple_ModbusServer.py
Start server...
Server is online
^CShutdown server ...
Server is offline
johannes@X230:Python$
```

### Modbus Addresses

#### IED1 → PLC:

- FC: 6 , Unit ID: 2, Address: 15

#### IED2 → PLC:

- FC: 6, Unit ID: 4, Address: 15

#### Feeder 1 IED → PLC

- FC: 6, Unit ID: 2, Address: 20

#### Feeder 2 IED → PLC

- FC: 6, Unit ID: 2, Address: 21

#### Feeder 3 IED → PLC

- FC: 6, Unit ID: 2, Address: 22

#### Feeder 4 IED → PLC

- FC: 6, Unit ID: 2, Address: 23

#### Feeder 5 IED → PLC

- FC: 6, Unit ID: 2, Address: 24

#### Feeder 6 IED → PLC

- FC: 6, Unit ID: 2, Address: 25

#### Feeder 7 IED → PLC

- FC: 6, Unit ID: 2, Address: 26

#### Feeder 8 IED → PLC

- FC: 6, Unit ID: 2, Address: 27

#### Feeder 9 IED → PLC

- FC: 6, Unit ID: 2, Address: 28

#### Feeder 10-18 IED → PLC

- FC: 6, Unit ID: 2, Address: 29-37

**IED1 → SCADA**

- FC: 16, Unit ID: 10, Address: 50, Quantity: 9

**IED2 → SCADA**

- FC: 16, Unit ID: 10, Address: 75, Quantity: 9

**Feeder 1 → SCADA**

FC: 16, Unit ID: 10, Address:100, Quantity: 9

**Feeder2→ SCADA**

FC: 16, Unit ID: 10, Address: 125, Quantity: 9

**Feeder3→ SCADA**

FC: 16, Unit ID: 10, Address: 150, Quantity: 9

**Feeder4→ SCADA**

FC: 16, Unit ID: 10, Address: 175, Quantity: 9

**Feeder5→ SCADA**

FC: 16, Unit ID: 10, Address:200, Quantity: 9

**Feeder6→ SCADA**

FC: 16, Unit ID: 10, Address:225, Quantity: 9

**Feeder7→ SCADA**

FC: 16, Unit ID: 10, Address:250, Quantity: 9

**Feeder8→ SCADA**

FC: 16, Unit ID: 10, Address: 275, Quantity: 9

**Feeder9→ SCADA**

FC: 16, Unit ID: 10, Address: 300, Quantity: 9

**Feeder10→ SCADA**

FC: 16, Unit ID: 10, Address: 325, Quantity: 9

**Feeder11→ SCADA**

FC: 16, Unit ID: 10, Address: 350, Quantity: 9

**Feeder12→ SCADA**

FC: 16, Unit ID: 10, Address: 375, Quantity: 9

**Feeder13→ SCADA**

FC: 16, Unit ID: 10, Address: 400, Quantity: 9

**Feeder14→ SCADA**

FC: 16, Unit ID: 10, Address: 425, Quantity: 9

**Feeder15→ SCADA**

FC: 16, Unit ID: 10, Address: 450, Quantity: 9

**Feeder16→ SCADA**

FC: 16, Unit ID: 10, Address: 475, Quantity: 9

**Feeder17→ SCADA**

FC: 16, Unit ID: 10, Address: 500, Quantity: 9

**Feeder18→ SCADA**

FC: 16, Unit ID: 10, Address: 525, Quantity: 9

**Scada → IED 1, IED2 ,Feeder 1 to 18**

FC: 6, Unit ID: 2, Address: **551 to 570** , Quantity: 1