# Substation Data Logging System

CS421-22 Mini Project

Naman Agarwal
1401CS28

Alan Aipe
1401CS50

# Contents

- Problem Statement
- Architecture
- Components and Technology Stack
    - Modbus
    - RS485
    - Raspberry Pi
    - Django
- Demo
- Conclusion
- Future Work

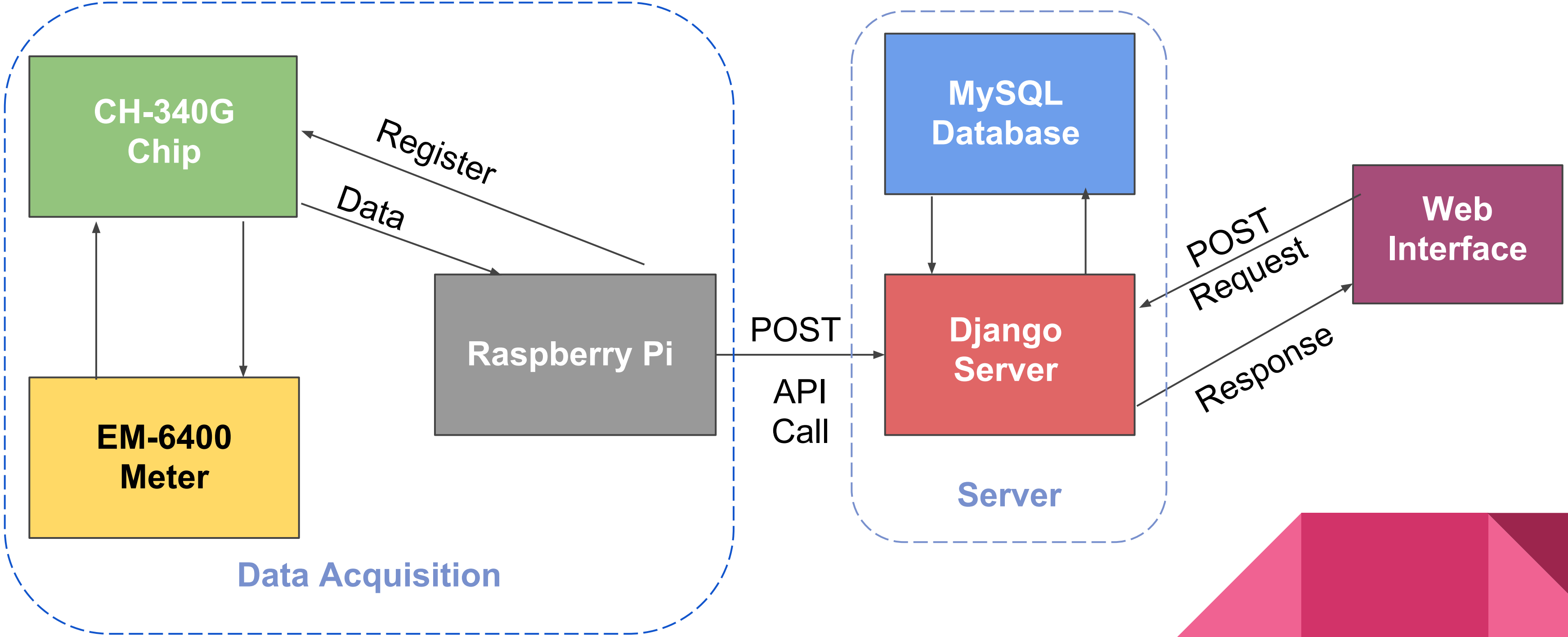
# Problem Statement

The project was adopted taking into account the human resources which goes into manual data logging at various electric substations of  IIT Patna.

Objectives of the project include:

- Automated logging of data available from EM-6400 meters installed in IIT Patna substation.
- Create user interface for better visualization of electric consumption data.
- Making data easily available for monitoring as well as analysis purposes.

# Architecture



# Components and Technology Stack

Logging system has been designed and optimized taking into consideration various hardware as well as software components as listed below:

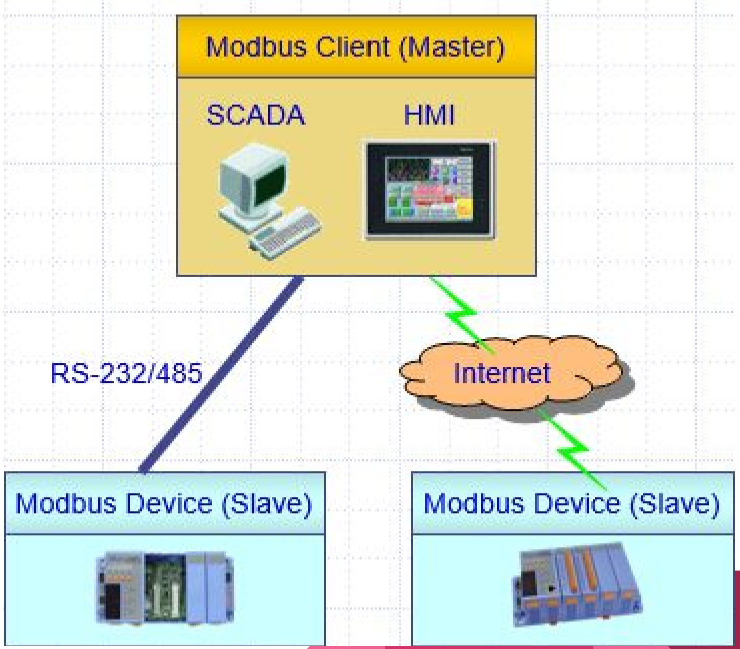
- Hardware components:
  - CH340G chip (USB to RS485 converter)
  - RaspberryPi
  - Wifi router
- Software components:
  - Libmodbus
  - CPP
  - Django
  - MySQL
  - Web Technologies like HTML, CSS and JS

# Modbus

- An open data communication protocol.
- Published by Modicon.
- Open structure &  Flexible.
- Widely known and supplied by many SCADA and HMI software.
- 2 serial transmission modes:
  - ASCII 10 bits (American Standard Code for Information Interchange)
  - RTU (Binary) 11 bits (remote terminal unit)
- Communication interface
  - RS-232/485
  - Ethernet (TCP/IP)

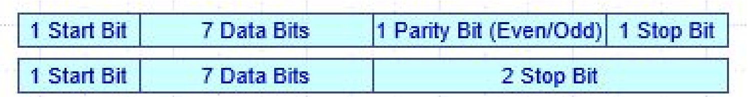
# Modbus Protocol

- Modbus protocol is defined as a master/slave protocol.
    - The master will write data to a slave device's registers,
    - and read data from a slave device's register.
    - Uses RS-485 or RS-232
- Modbus TCP is defined as a client/server protocol.
    - The slave becomes the server,
    - and the master becomes the client.
    - Uses Ethernet
- There can be multiple masters as well as multiple slaves



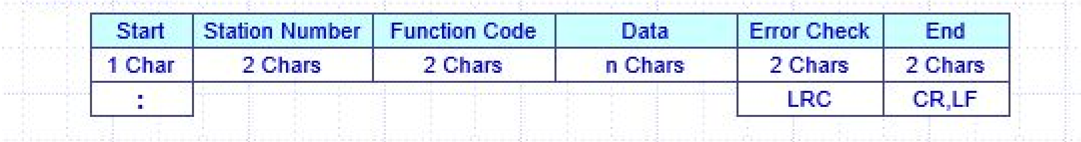
# Two Serial Transmission Modes

- ASCII Mode
  - Data system ASCII character, ‘0’~’9’,’A’~’F’
  - Bits per data unit



| | | | |
|---|---|---|---|
| 1 Start Bit | 7 Data Bits | 1 Parity Bit (Even/Odd) | 1 Stop Bit |
| 1 Start Bit | 7 Data Bits | 2 Stop Bit | |

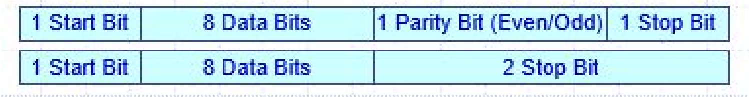
  - Error Check Field (Longitudinal Redundancy Check (LRC))
  - Message Packet



| Start | Station Number | Function Code | Data | Error Check | End |
|---|---|---|---|---|---|
| 1 Char | 2 Chars | 2 Chars | n Chars | 2 Chars | 2 Chars |
| : | | | | LRC | CR,LF |

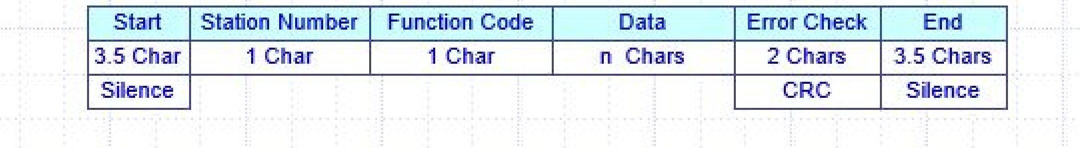
# Contd.

- RTU Mode
  - Data system 8-bit Binary, 00~FF
  - Bits per data unit



| | | | |
|---|---|---|---|
| 1 Start Bit | 8 Data Bits | 1 Parity Bit (Even/Odd) | 1 Stop Bit |
| 1 Start Bit | 8 Data Bits | 2 Stop Bit | |

  - Error Check Field (Cyclical Redundancy Check (CRC))
  - Message Packet



| Start | Station Number | Function Code | Data | Error Check | End |
|---|---|---|---|---|---|
| 3.5 Char | 1 Char | 1 Char | n Chars | 2 Chars | 3.5 Chars |
| Silence | | | | CRC | Silence |

# Modbus Function Code

- 01: read Coil (Discrete Output) (0xxxx)
- 02: read Discrete Input (1xxxx)
- 03: read Holding Registers (4xxxx)
- 04: read Input Registers (3xxxx)
- 05: write Single Coil (0xxxx)
- 06: write single Holding Register (4xxxx)
- 15: write Multiple Coils (0xxxx)
- 16: write Multiple Holding Registers (4xxxx)

# Advantages

- It is scalable in complexity and scope.
- It is simple to administer and enhance.
- There is no vendor-proprietary equipment or software needed.
- It is very high performance, limited typically by the ability of the computer operating systems to communicate.
- It can be used to communicate with the large installed base of MODBUS devices, using conversion products, which require no configuration.

# Libmodbus

- Libmodbus is a library to send/receive data with a device which respects the Modbus protocol.
- Supports RTU (serial) and TCP (Ethernet) communications.
- Written in C.
- Available for Linux, Mac OS X, FreeBSD, QNX and Windows.

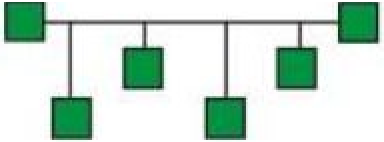
# RS485

- **RS-485**, also known as **TIA-485(-A)**, **EIA-485**, is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems.
- It is balanced data-transmission scheme that offers robust solutions for transmitting data over long distances and noisy environments.
- Used as physical layer specification by many protocols like Modbus, Profibus and many others.

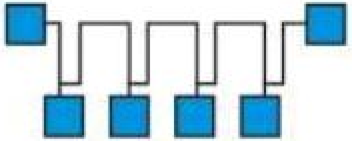
# Contd.

- Half duplex connection.
- 32 to 282 devices.
- Uses balanced differential lines for communication, usually twisted pairs of wires.
- Data is logical 1 when polarity is positive(voltage level in ‘+’ wire is higher than ‘-’ wire) and 0 when negative polarity with a noise margin of +/- 0.2 V.

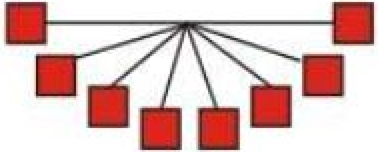
# Topology



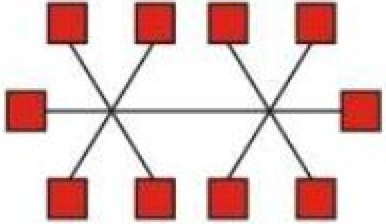
Backbone with stubs
(workable)



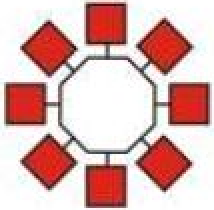
Daisy chain
(best)



Star network
(avoid)

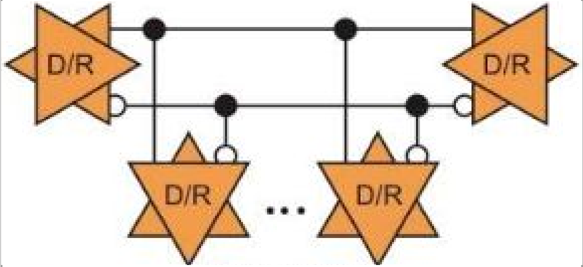
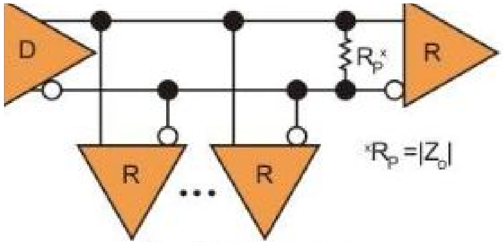
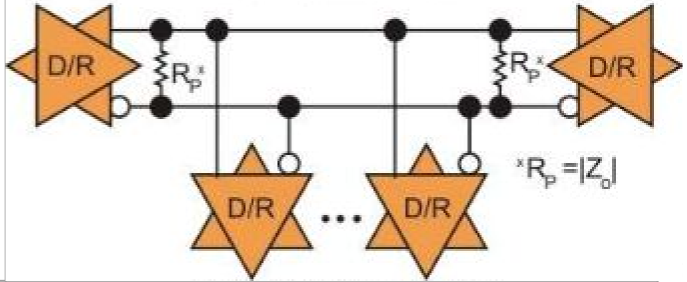


Backbone with stars or clusters
(avoid)

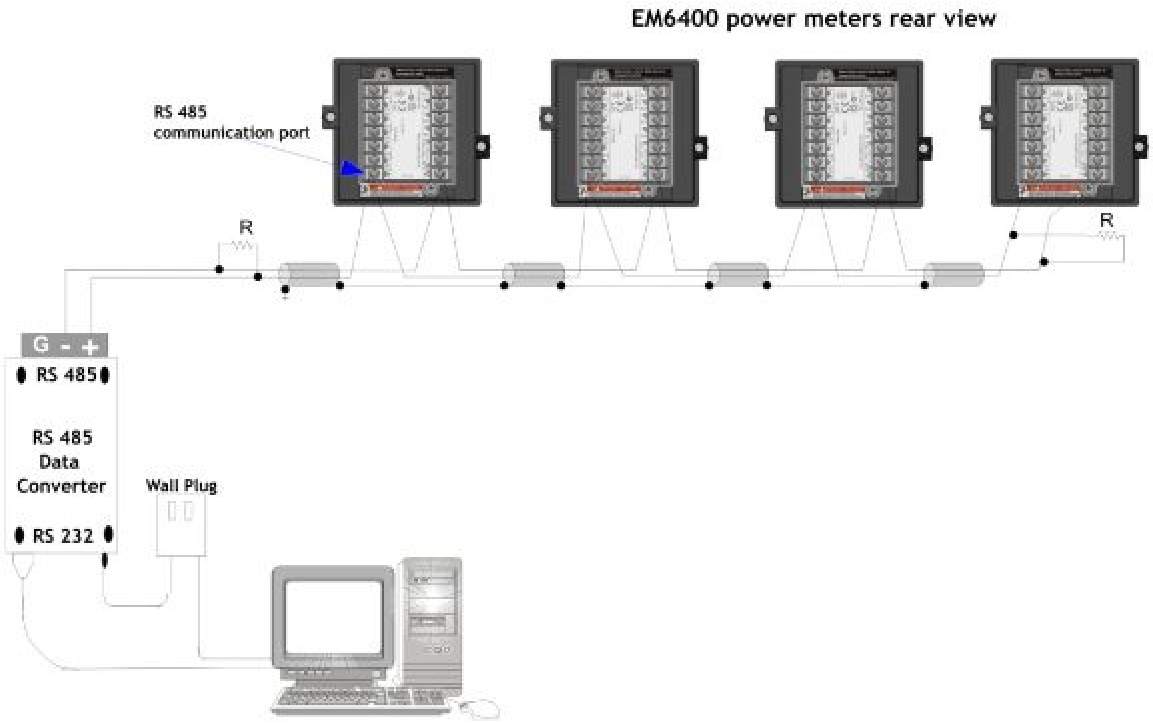


Ring
(avoid)

# Termination resistors

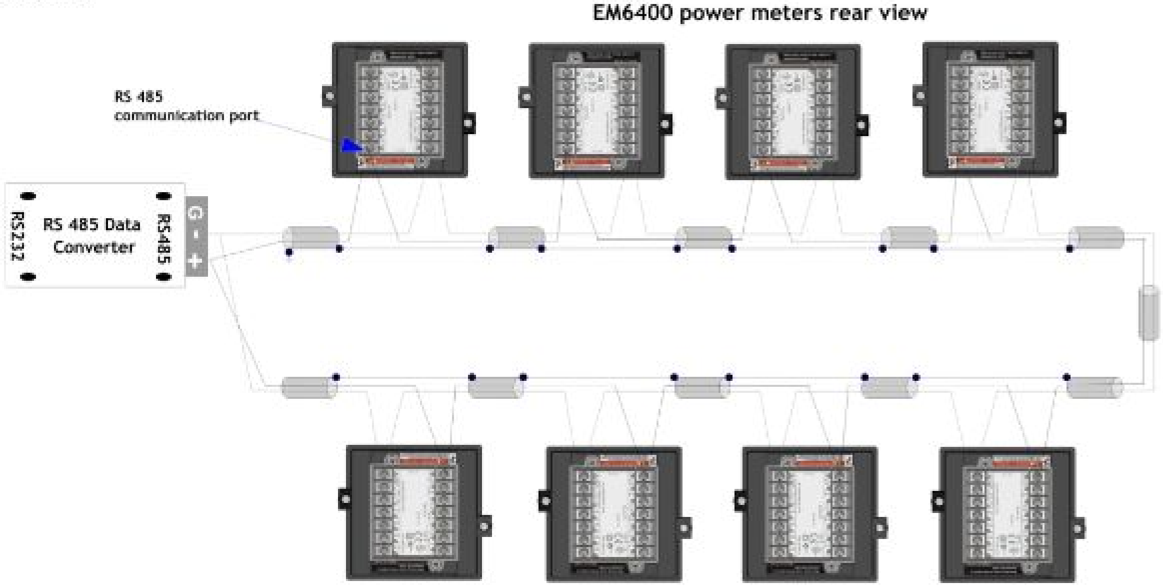
| **Unterminated** | **Parallel termination** | **Bidirectional termination** |
|---|---|---|
|  |  |   D/R<br>$R_P^*$<br>$R_P^*$<br>D/R<br>D/R<br>...<br>D/R<br>$^*R_P = \|Z_0\|$ |
| ● low power, low cost and simple to build.<br>● Data rates must be quite slow, or cable length must be short | ● Excellent data rates but limited to networks with one driver/transmitter. | ● Excellent signal Integrity.<br>● High power Consumption |

# Connection with EM6400 meter



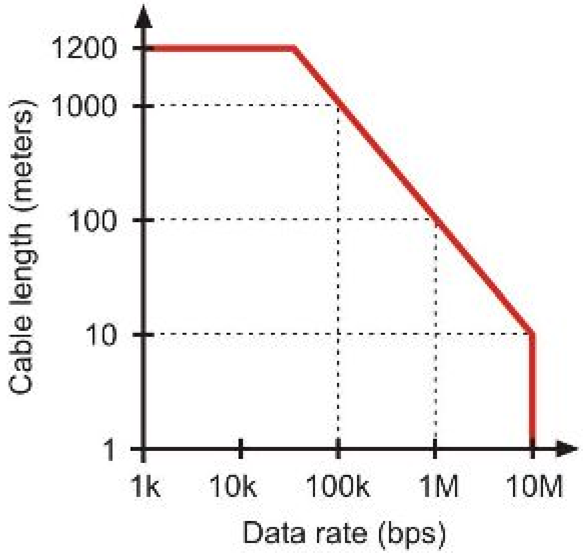
# Contd.

**Figure 6-2: Closed loop, 2-wire half duplex.**
**Advantage – Reliable communications, tolerant to one break in the cable.**



# Cable length and data rate

- Cable length (in m) * Data rate (in bps) <= $10^{8.}$
- Data-rate of 10Mbps.
- Distance up to 1200m (4000 ft)



# Raspberry Pi

Raspberry Pi is a single board computer which mimics the functionality of a computer system at a mini-level. Pi 3 has been used in this project for reception of data streams from CH340G chip and uploading in the remote server (REST API calls) using C++ programming.

Raspberry Pi 3

# Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

Why Django?

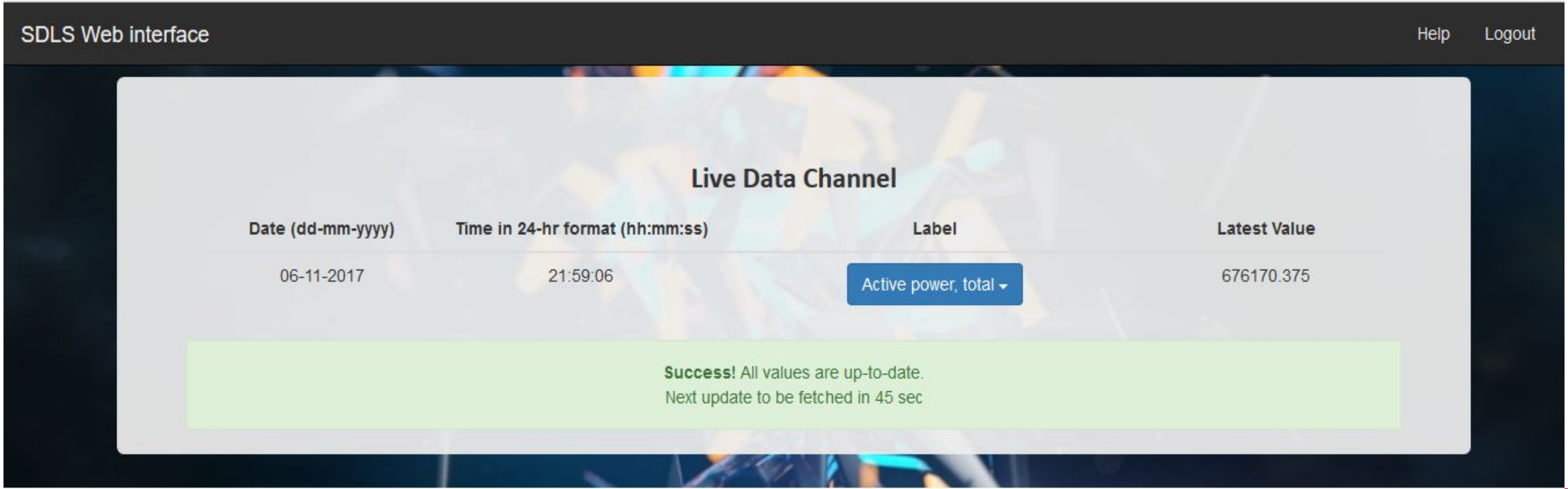
- It facilitate faster development of server apps.
- User authentication is very secure.
- Easily scalable.
- Better backend skeleton for database management.

# MySQL & Front-end Web Technologies

MySQL is a relational database which has been used in the project in order to manage previous 24 hours as well as aggregate data for each day in server. MySQL database is easier to manage and facilitate better import/export features to other data formats.

Web Technologies like HTML, CSS and JS has been used at the dashboard. REST API calls are fetched and displayed using AJAX. Libraries of JS like CanvasJS has been used to create dynamic visual representations and aggregations of data.

# Demo / Screenshots



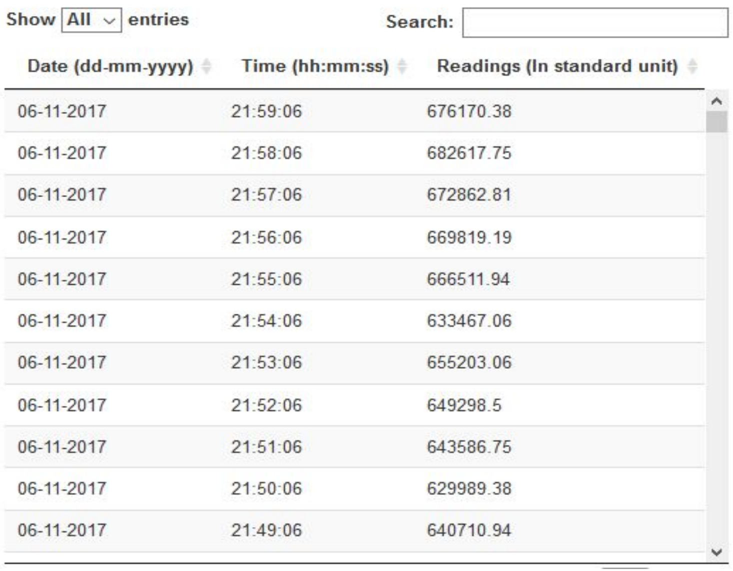
## Live Data Channel

| Date (dd-mm-yyyy) | Time in 24-hr format (hh:mm:ss) | Label | Latest Value |
|---|---|---|---|
| 06-11-2017 | 21:59:06 | Active power, total ▾ | 676170.375 |

**Success!** All values are up-to-date.
Next update to be fetched in 45 sec

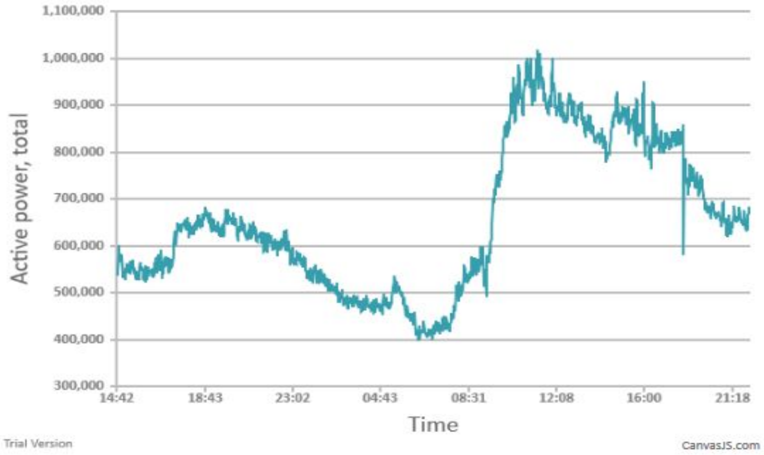
# Contd.

**24 Hour Data** Data by Date

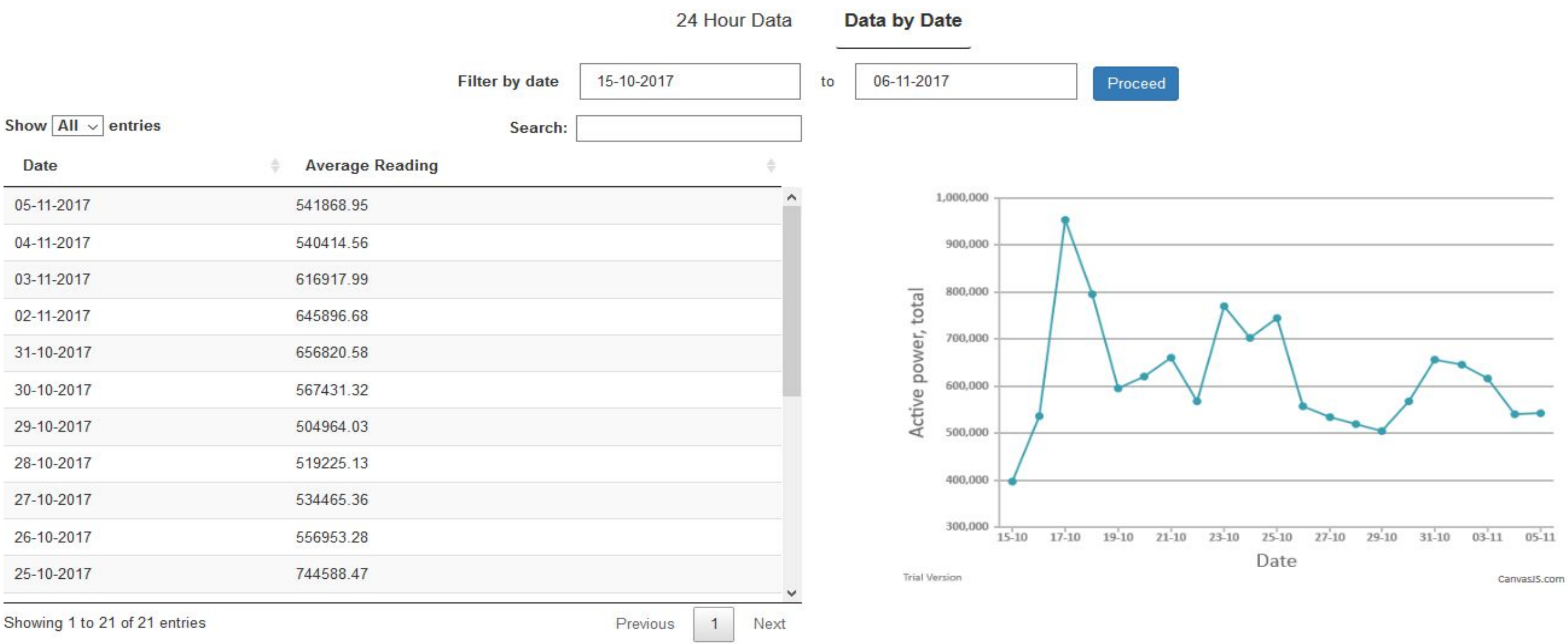


Show All entries Search:

| Date (dd-mm-yyyy) | Time (hh:mm:ss) | Readings (In standard unit) |
|---|---|---|
| 06-11-2017 | 21:59:06 | 676170.38 |
| 06-11-2017 | 21:58:06 | 682617.75 |
| 06-11-2017 | 21:57:06 | 672862.81 |
| 06-11-2017 | 21:56:06 | 669819.19 |
| 06-11-2017 | 21:55:06 | 666511.94 |
| 06-11-2017 | 21:54:06 | 633467.06 |
| 06-11-2017 | 21:53:06 | 655203.06 |
| 06-11-2017 | 21:52:06 | 649298.5 |
| 06-11-2017 | 21:51:06 | 643586.75 |
| 06-11-2017 | 21:50:06 | 629989.38 |
| 06-11-2017 | 21:49:06 | 640710.94 |



# Contd.



Filter by date 15-10-2017 to 06-11-2017

Show All entries

Search:

| Date | Average Reading |
|---|---|
| 05-11-2017 | 541868.95 |
| 04-11-2017 | 540414.56 |
| 03-11-2017 | 616917.99 |
| 02-11-2017 | 645896.68 |
| 31-10-2017 | 656820.58 |
| 30-10-2017 | 567431.32 |
| 29-10-2017 | 504964.03 |
| 28-10-2017 | 519225.13 |
| 27-10-2017 | 534465.36 |
| 26-10-2017 | 556953.28 |
| 25-10-2017 | 744588.47 |

Showing 1 to 21 of 21 entries

# Conclusion

Substation Data Logging System (SDLS) has been deployed successfully and tested in one of the substations in our campus. It streams in data every minute into the server which are dynamically shown in admin console of SDLS web interface.

This system would, inarguably, help in faster and accurate modelling of power consumption in the college, thus providing a helping hand to the upcoming research works in the field of smart grids.

# Future Work

The project can be extended to :

- Encompass data analytics to server.
- Machine learning models to predict power surges.
- Time series modelling of power consumptions.
- Automated regulatory systems for the grids.

# Thank You

Questions, if any?

## References

- http://www.modicon.com
- http://www.modbus.org/default.htm
- https://en.wikipedia.org/wiki/Modbus
- https://github.com/stephane/libmodbus
- https://en.wikipedia.org/wiki/RS-485
- https://www.djangoproject.com
- https://www.w3schools.com