Industrial IoT Data Simulator

submitted by

K Siddharth

16BIS0052

in

Electronics and Communication with specialisation in Internet of Things and Sensors

Vellore Institute of Technology, Vellore

May-June, 2018

Certificate

This is to certify that **K Siddharth** , pursuing **B.Tech** from **Vellore Institute of Technology, Vellore** has successfully completed the **Project** on the topic entitled **'Industrial IoT Data Simulator'** from **May 14th to the 29th June, 2018,**which is a record of bonafide work, under the supervision of **Arunkumar Jayaraman**.The contents of this project have not been taken from any other source.

Place: Bangalore

Date: 29/06/2018 (tentative)

(Signature of Candidate)

Acknowledgement

I would like to express my special thanks of gratitude to my supervisor, **Mr. Arunkumar Jayaraman**, who has given me opportunity to do this project on the topic '**Industrial IoT Data Simulator'**, and also for his continuous guidance and encouragement in carrying out this project. I would also like to thank my **Colleagues**, without whose help this project would not have been a success. I would also like to thank **everyone**, who directly or indirectly, has lent their helping hand in this venture.

(Signature of Supervisor)

Table of Contents

1. Abstract

- Equipments play a major role in any given industry and can be used for a multitude of purposes. They need to be monitored constantly to ensure that they function according to certain requirements.
- To ensure this, we connect certain devices to these equipments, which are responsible for collecting required data.
- This data can now be analysed to review the performance of the equipment.

- SuperAxis is an Industrial IoT and Analytics Platform created by EcoAxis and deployed at multiple customer locations - such as Thermal Power Plants, Discrete Manufacturing Industries, Textile and Sugar Industries, Large Enterprises.
- While developing and testing SaaS (software as a service) applications in the platform, an IoT Data simulator that simulates close to real world data, as per the SuperAxis configuration, is quite important and vital for the platform and applications.
- Once the data is generated, it is made available in different transports such as CSV files, zip file, FTP etc.

2. Software/System Requirements

After careful evaluation of different technologies under the guidance of the EcoAxis team, the following set of technologies is chosen to build the IoT Data Simulator.

- Java - version 8
- Eclipse IDE - release 4.7.3a
- MySQL Workbench - Version 6.3.9
- HighCharts version 6.1.0
- Xampp - version 5.6.35

3. Introduction

The data simulator helps us analyse the performance of an equipment and its parameters, by generating data. Typically the real world data from the Equipment is sensitive and protected on one side and on the other, it doesn't provide all data probabilities within a considered short time span. While building functional, performant and scalable Analytical systems, it is key to have a data set close to real world.

- To simulate real world situation, a pattern for a defined configuration is created that enables Data simulator to generate the required data set.
- Using the Data Simulator, one can generate data for multiple such equipment, each of them carrying many such parameters of different kinds.
- The data simulator first acquires the relevant information that is stored in the database of each equipment and its parameters, their nature,configuration and even pattern of data generation.
- It sorts the data based on the equipments and their parameters.
- For each equipment that is read by the data simulator,data is generated depending on the nature and the configuration of each parameter of the selected equipment.
- This process is repeated for all equipments that are present in the database’s table.
- Files consisting of the data of all parameters, are created(timestamp of occurrence, value of data generated) in a certain folder, which is useful in the analysis of the equipment’s functioning and behaviour.

4. Systematic approach for the Data Simulator

4.1 Agile methodologies used

- Scrum sessions were held on a daily basis, which also included a standup meeting, that included:
  - Reflection on the work done previously and the problems encountered.
  - Visualization of the work to be done, for the day.
  - Parts of the project were developed and tested on a weekly basis.
  - New features were developed between sprint sessions.
  - Many features of the end product were discussed frequently, such as: Quality of the product, it's flexibility, Customer satisfaction and even problems encountered.
- Enhancing the flow of work done, by implementing alternate methods, if required.
- Visualization of the end goal, in each stage of work done.

4.2 Gathering of Requirements

- The requirements for the data simulator have to be acquired before it is implemented.
- The time frame between which data is generated, should be provided as the start and end timestamps, on the equipment level.
- In this case, the requirements include a list of all the equipments and their parameters with their nature, patterns and their configurations of generating data. Each parameter will be assigned a pattern, according to which data is generated. The patterns are each represented with a unique identification number which consists of the configuration:

4.2.1 Analog parameter Specifications

- Maximum and minimum values between which the data is generated.
- Maximum and minimum possible deviation values between any 2 consecutive data points.
- Publish and Capture frequency values (specified on the equipment level).

4.2.2 Digital parameter Specifications

- Maximum and minimum zero occurrence values (can also behave like a breakdown).
- Maximum and minimum time durations of each zero occurrence.
- Publish and capture frequency values (specified on the equipment level).

- The requirements also include the details of how data is to be represented to the end user/customer. Data can be written locally, via FTP/HTTP onto a specific server. The data can also be collected for a specific amount of time and can be saved in a .zip format, for easy access.

All the above requirements are stored on a database using MySQL.

4.3 Design of the Data Simulator

- The data simulator has many parts to its construction:
    - ➔ Importing of data from the database should be done in a quick and effective manner. This process may vary slightly depending on the amount of data that is present.
    - ➔ Separation of Equipments and parameters.
    - ➔ Further separation of the parameters of an equipment, based on their type.
    - ➔ Data generation for analog and digital parameters of the equipment is done. Algorithms for each type of data generator should be created based on the requirements. Data Generation should be done quickly, taking the required conditions into consideration.
    - ➔ Process all equipments that are read as data.
    - ➔ Generated data is made available as per the FTP Format in a File that can be ingested directly into the SuperAxis System.
    - ➔ Files can be used for representation of data in the form of charts and graphs, with the help of external software.

Note:

There should not be a compromise in the system’s performance while reading and generating data quickly.

5. Reason for selection of Software/System Requirements

5.1 Java - Version 8

- Some reasons why Java is used in this project and why it is one of the most popular programming languages are:
  - ➔ Java is an ideal programming language for IoT related applications that demands high scalability and performance.
  - ➔ It is a platform-independent language.
  - ➔ Java is an object-oriented language.
  - ➔ It is free and very easy to use.
  - ➔ Java has many useful in-built Libraries/Frameworks.
  - ➔ Java has multithreading capabilities
  - ➔ Java has high interoperability with various systems

5.2 Eclipse IDE - release 4.7.3a

- Eclipse is one of the most popular IDE's (integrated development environment) that is used due to the following reasons:
  - ➔ It is free to use.
  - ➔ It is user-friendly.
  - ➔ Eclipse can be used for Java, Python, C, C++ development.
  - ➔ More features can be added to Eclipse by installing plugins.
  - ➔ Has good support in the form of developer forums and online resources.

5.3 MySQL Workbench - version 6.3.9

- MySQL is used as the relational database management system in this project and it is the most popular DBMS in the world for the some of the following reasons:
  - ➔ It is a relational database that support SQL to query data.
  - ➔ It is straightforward and easy to use.
  - ➔ It is free to use.
  - ➔ It is easy to set up MySQL server.

- ➔ Interoperable with Java and other languages.

5.4 HighCharts - version 6.1.0

- HighCharts is used for data visualization and is a popular charting library, which serves many large companies in the world, since it helps in solving few of the following purposes:
  - ➔ It can be easily integrated into web frameworks.
  - ➔ Data can be dynamically represented in highcharts very easily.
  - ➔ It is easily customisable, according to the user's convenience.
  - ➔ It has a bunch of libraries that can be used to represent data in different formats.
  - ➔ Highcharts can also be used on older browsers.
  - ➔ The online version of HighCharts is free to use.

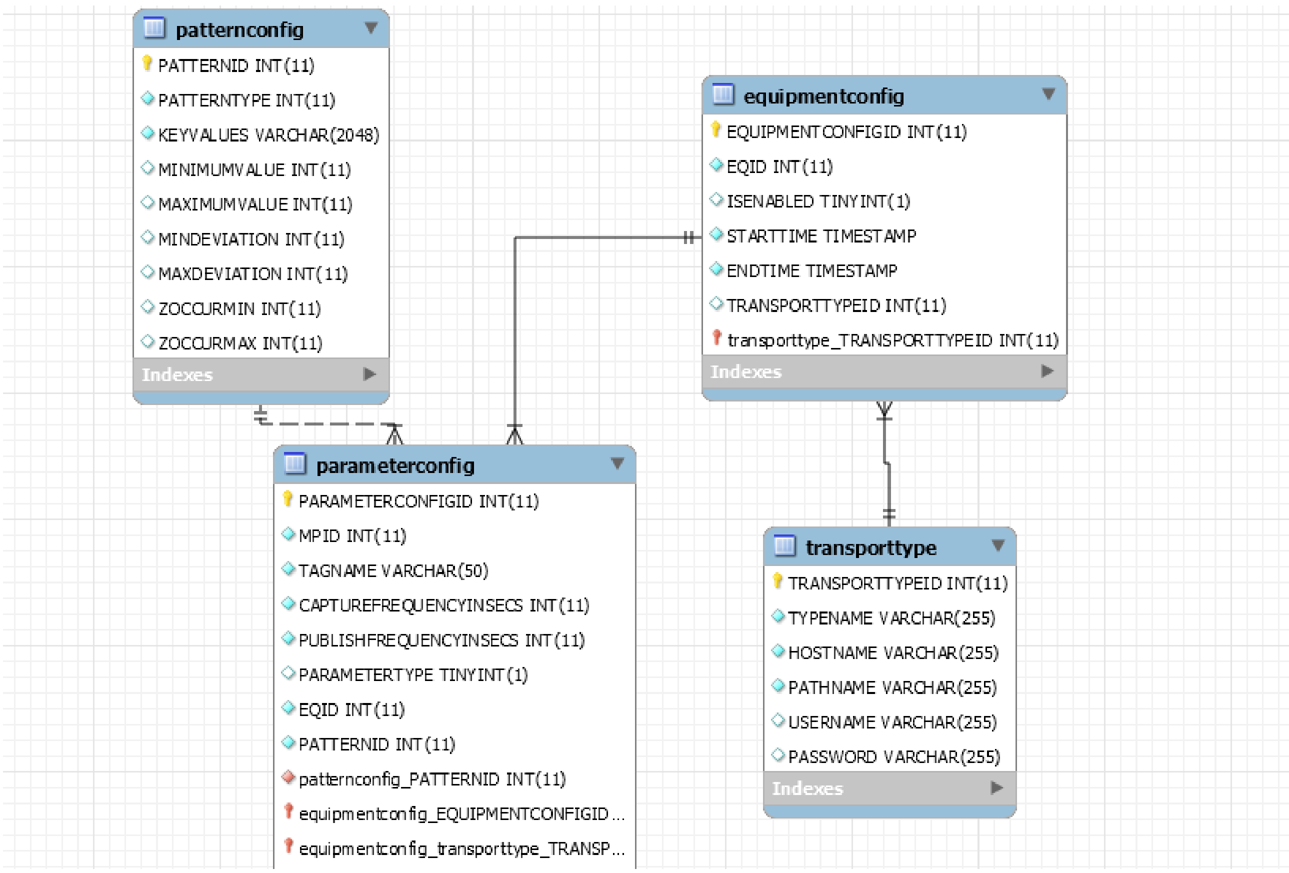
5.5 Xampp - version 5.6.35

- It is a free and easy to use cross platform web-server solution.
- It is lightweight.
- It provides easy transitioning from a local test server to a live server

6. Workflow

6.1 - MySQL Implementation

- MySQL, a relational Database Management System, hosts a database where all the relevant data is stored and can be used for CRUD operations.
- The database consists of 4 tables, each having their own functionality.



- The above chart is a block diagram representation of the tables in their atomic state.
- This also depicts the relation established between tables.
- The first table consists of the configuration of all the equipments for which data has to be generated.
    - We identify each equipment by its unique ID.
    - Data is only generated for those equipments which are enabled.
    - Start and End times are provided, between which data is generated.

    - This table is linked to the Transport table, which indicates the type of file writing method to be implemented.

- The second table consists of the parameters, that is linked to the equipments table.
    - Each parameter has its own parameter ID, which is related to an equipment.
    - Each parameter has a tag name.
    - The rate at which data is captured and written is based on the capture and publish frequencies.
    - The parameters are distinguished from one another bases on their type (analog, digital)
    - This table is linked to the equipments table and the pattern table (data for the parameters is generated based on the pattern that is assigned to each parameter).

- The third table consists of the pattern configuration, which forms the basis for the data generation of each parameter.
    - Each pattern has its own pattern ID and type, so that they can be distinguished easily.
    - Data is generated between the minimum and maximum values, in the case of an analog parameter. For a digital parameter, the minimum and maximum values depicts the time range for a zero occurrence.
    - In the case of analog parameters, minimum and maximum deviations are the values between which two consecutive data points can differ by.
    - In the case of digital parameters, minimum and maximum zero occurrences are values between which a zero occurrence value is used in data generation.

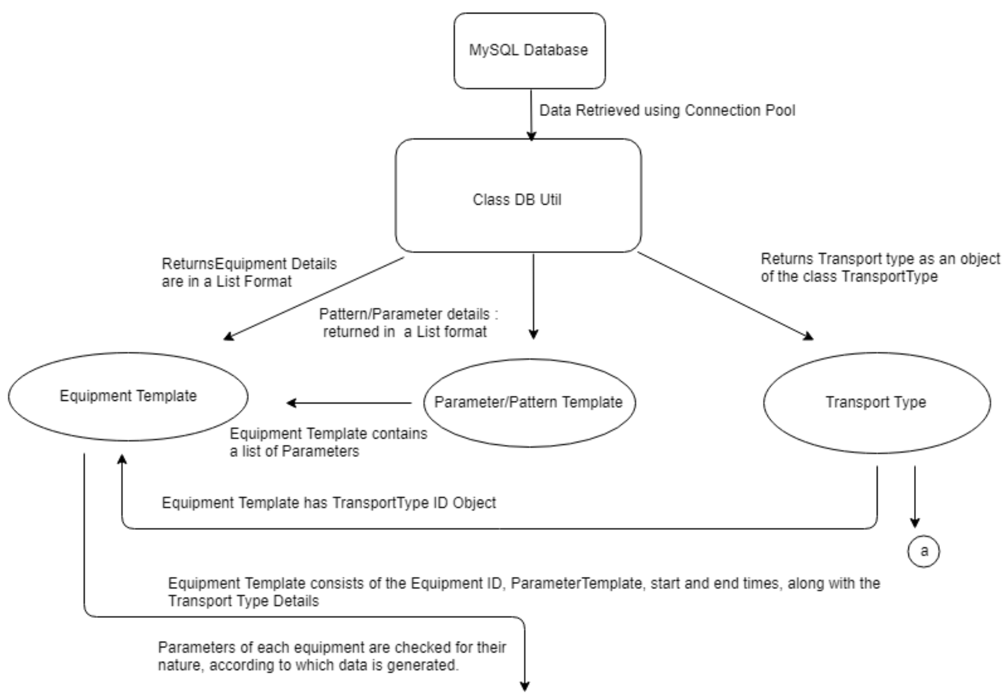
- The fourth table is the transport table, which has the following information:

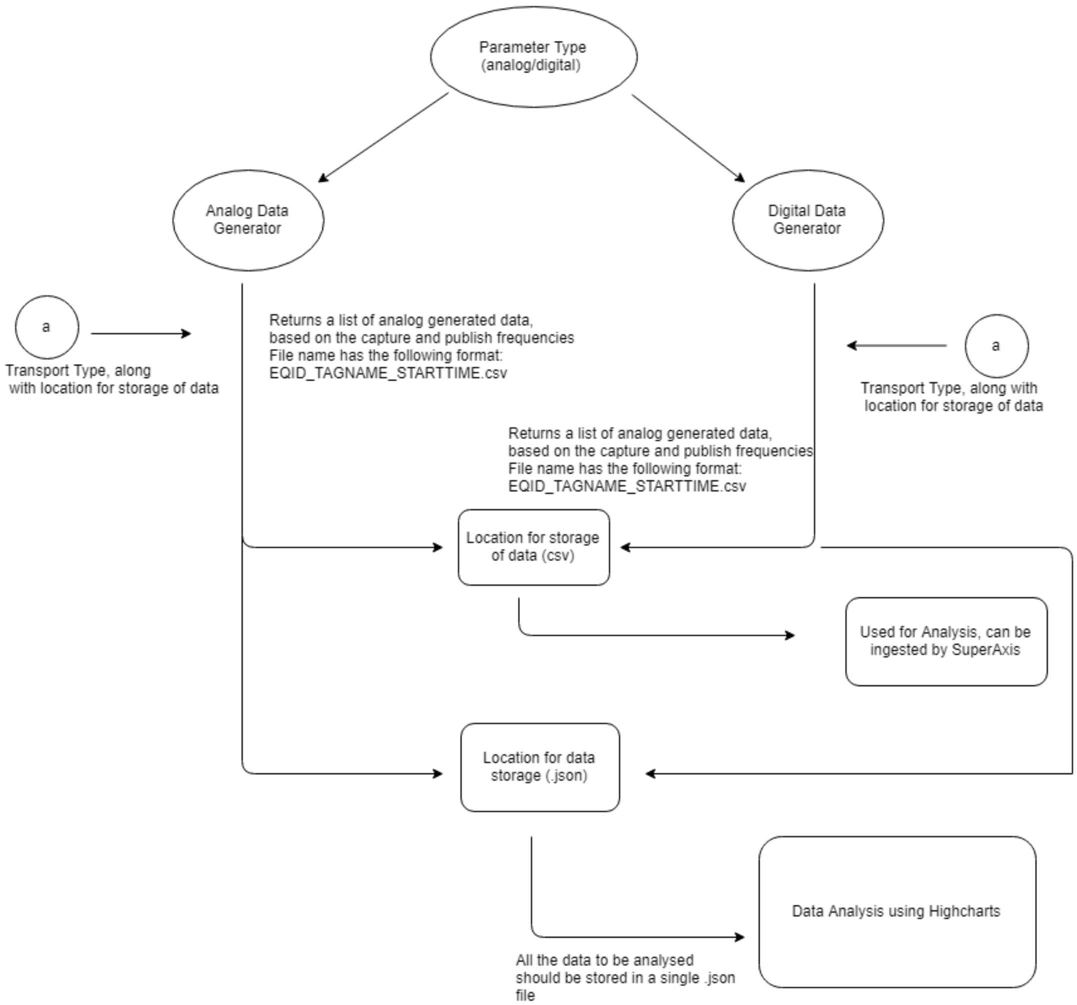
- ID and name of each transport type to be used for data writing, for any equipment.
- It has a hostname and a pathname, in the case of FTP. Data is written onto the host's directory, into the specified path.
- Username and password are used in the case when the data is secured and should only be accessed by authorised people.

6.2 - Java Implementation using Eclipse IDE

- After the database is configured, the main part of the project is implemented in java, with the help of the Eclipse IDE.

6.2.1 Flowchart





6.2.2. Connection Establishment

- To obtain data from the database, we establish a connection to it, from the Eclipse IDE. The connection is established to the database, in the form of a connection pool which will ease in a multi-threaded environment. Used Hikari CP (connection pool) for creating a pool of connections.
- Constantly opening and closing connections can be expensive. Cache and reuse.
- When activity spikes you can limit the number of connections to the database. This will force code to block until a connection is available. This is especially helpful in distributed environments.
- The main advantage of using a connection pool is faster readability. The connection pool establishes a certain number of connections with database.

6.2.3. Initial Processing

- Once the connection has been established, the data that is read from the database, is sorted equipment-wise.
- For each equipment, all its corresponding parameters are sorted, based on their nature (analog/digital).

Note:

➔ *Only the data of those equipments which are enabled, are read.*

6.2.4. Analog Data Generation

- For Analog data generation, the corresponding configuration (minimum/maximum values, minimum/maximum deviations) is read from the database (for each unique pattern that is used).
- The above step is repeated for each analog parameter of an equipment, followed by all the equipments in a similar manner.
- Data for the selected parameter is now generated in the following manner:
    - ➢ First, the number of data points for which data has to be generated are obtained. This is done by subtracting the time difference between the start and end times and dividing this result by the capture frequency.
    - ➢ Initially, a random value of data is obtained between the maximum and minimum values.
    - ➢ To this, we add another random value that is generated between the minimum and maximum deviations.
    - ➢ This process is repeated for all the data points of the parameter.

- The data that has been obtained is written onto a file in every interval of the publish frequency, in a certain format. The data points also have the timestamp of their occurrence.

6.2.5. Digital Data Generation

- ○ For Digital data generation, the corresponding configuration (minimum/maximum values, minimum/maximum deviations) is read from the database (for each unique pattern that is used).
- ○ The above step is repeated for each digital parameter of an equipment, followed by all the equipments in a similar manner.

    - Data is generated for the selected parameter in the following manner:
        - ➢ A random value of zero occurrence is obtained, which lies between the minimum and maximum zero occurrence values.
        - ➢ Each of these xero occurrence is now assigned a time value, which lies between the minimum and maximum zero time.
        - ➢ Random start times are allocated in the time slot in such a manner that when a breakdown is simulated, the next start point should not lie in the breakdown duration.
        - ➢ After all the breakdowns and their values are simulated, we store the values in intervals of the capture frequency, along with their timestamp of occurrence.

- The data that has been obtained is written onto a file in every interval of the publish frequency, in a certain format. The data points also have the timestamp of their occurrence.

In both cases of data generation, initially an api called TSimulus was implemented in Java. It was a ready-to-use data generator. It was then dropped since its implementation was not easy.

The algorithm for the current data simulators was then written and executed for analog and digital parameters.

6.2.6. ExecutorService

- ❖ The entire process from retrieving the parameter of an equipment till the writing of data in a file, is accomplished with the help of the ExecutorService in java.
- ❖ The ExecutorService consists of a multi-thread pool.
- ❖ Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU
- ❖ Thread in java can be implemented in 2 ways:
    - ➢ Extending the thread class

- ➢ Implementing the runnable interface

❖ A thread from the thread-pool is assigned to each parameter of an equipment, which carries out the process and is then destroyed.

❖ This significantly reduces the time for overall processing.

❖ This is repeated for all the parameters of all the equipments.

❖ When the thread has completed its work, it is destroyed.

6.2.7. Data Writing

❖ Once the data has been generated, it is written onto a file in a specific format (timestamp, value).

❖ The file name is in the following format : ‘EquipmentID_TagName_StartTime.csv’

❖ In this case, we only write files onto the local directory, but there are options where we can directly publish the files via FTP, HTTP or even obtain the the files in a specific format (.zip).

❖ Alternately, the files are written in another specific format (.json) to visualise them graphically, using HighCharts.

6.3 Setting up a local server using Xampp

- A local server is setup using Xampp so that the html and js file can be run to view the data from the .json file.
- The .json file is run within the .js file.

6.4 Visualization of Data using HighCharts

- ➢ All the files consisting of data to be viewed should exist in a single ‘.json’ file.

- ➢ Along with the help of a '.html' and a '.js' file, the '.json' is run and the graph can be viewed on HighCharts.

6.4.1 Creation of the .html file/.js file

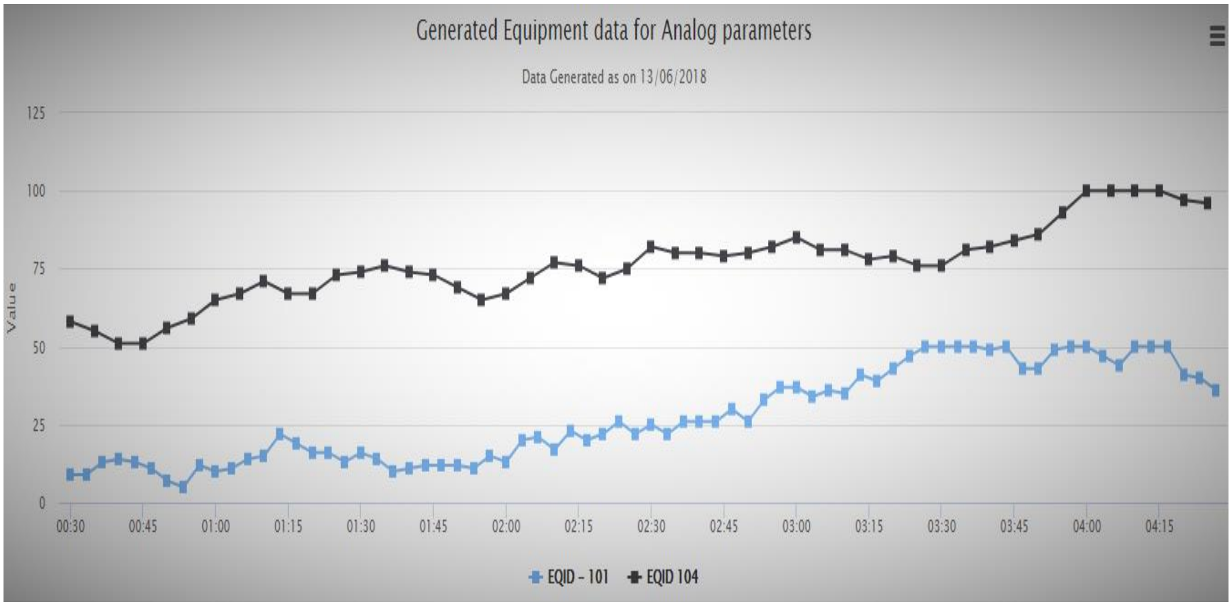
- ➢ A standard html and js file have to be created, to view the data on the local server. This can be done by referring the jsfiddle webpage
- ➢ The html and js files can be modified to represent data in a specific manner.

6.4.2 Running the .html/.js file

- ➢ The .json, .html and the .js file should be present in the same folder.
- ➢ A function should be written within the .js file that reads the data from the .json file.
- ➢ On running the .js file, we can open our web browser to view the charted data.

7. Data Analysis

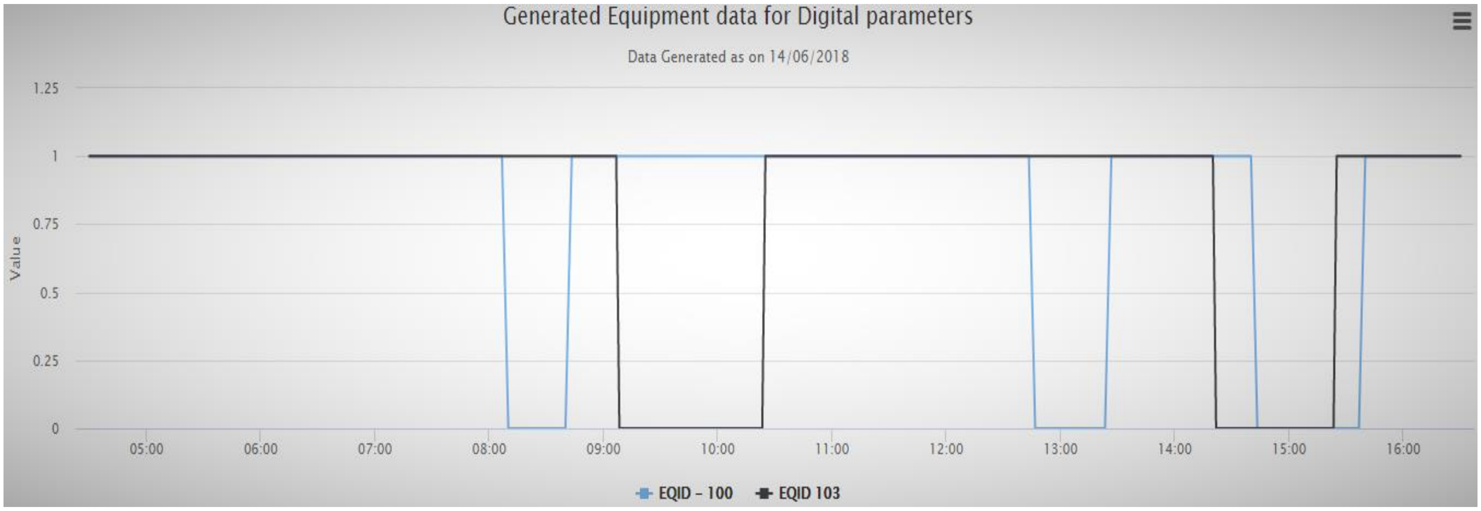
7.1 Analog Data Analysis



Two Equipments with IDs: 101,104 have analog parameters, whose data has been generated on the 13th of June, 2018 and viewed using highcharts.

- Equipment 101 takes analog values ranging from 0 to 50
- Equipement 104 takes analog values ranging from 50-100

7.2 Digital Data Analysis



Two Equipments with IDs: 100,103 have digital parameters, whose data has been generated on the 14th of June, 2018 and viewed using highcharts.

- Equipment 100 has 3 zero occurrences in the total time for which data is simulated.
    - Each zero occurrence has a different amount of time for which is occurs.
    - The machine is seen to be operational at all points other than these.
    - None of the zero occurrences overlap with another.
- Equipment 103 has 2 zero occurrences in the total time for which data is simulated.
    - Each zero occurrence has a different amount of time for which it occurs.
    - The machine is operational at other points other than these.
    - Again, none of the zero occurrences overlap with another.

The variation in values for analog and digital parameters of equipments is therefore seen using HighCharts.

8. Conclusion

The Data Simulator produces results as shown above, which is visualised using HighCharts for a detailed view/analysis of the equipment's functioning. The data simulator can generate data for extended time durations that may vary from a few days to a few months and even years.

More specifications can be added to the requirements, to generate data in a much more specific manner.

Thus we can visualise the data that has been generated, which can be very useful in identifying problems in the equipment, improving its performance as well as using it for detailed analysis and research purposes.

9. Possible Improvements

There are improvements that can be implemented into the data simulator, which may be as follows:

- Parameter dependencies of an equipment can be listed in the requirements, so that data is generated accordingly.
- Some equipments may still consume power even if they do not function. In this case, random data cannot be generated, as the power consumed will never reduce at any stage.
- Data can be generated for all possible devices and appliances in different conditions.
- There may be faster methods of generating data, which can be implemented.

The Data Simulator can be implemented with these changes so that the generation of data becomes much more seamless and can be applicable to a wide variety of equipments, appliances and devices with different parameter dependencies and conditions.

10. References

https://stackoverflow.com/

https://mvnrepository.com/

https://www.w3schools.com/

https://www.mysql.com/downloads/

https://www.mkyong.com/

https://www.highcharts.com/

https://jsfiddle.net/

https://www.scrum.org/resources/what-is-a-sprint-in-scrum

https://www.thescrummaster.co.uk/scrum/benefits-scrum-agile/

https://www.oracle.com/in/java/index.html

https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/

https://iot-analytics.com/