

## Graphical Representation of Real-Time Data from IoT

*Shashwat Siddhant, Gaurav Gupta, and Puneet Sharma*

Centre for Nano Science Engineering (CeNSE),  
Indian Institute of Science,  
Bengaluru, India

Copyright © 2016 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT:** The amount of data in this world has been exploding and expanding at an astounding rate. There is data coming in from almost every possible source. This data has to be carefully monitored and utilized in order to draw meaningful conclusions. For this sole purpose, analysis is done. Analysis of data makes it usable and transparent. This enables people with almost no technical knowledge understand the significance of the data captured. This paper focuses on the implementation of a medium for real-time graphical representation of data transmitted by IoT devices using favorable transmission process depending on the device. The medium chosen for this purpose is a website with a friendly user interface which makes it easy for people to study data, understand its importance and utilize it accordingly.

**KEYWORDS:** Internet of things(IoT), Raspberry Pi, Arduino, real-time graph, HTTP, TCP/IP, Requests, Interfacing, Model-View-Controller(MVC).

### 1 INTRODUCTION

With the constant progress in various fields of technology comes the need to understand and realize the importance of the purpose it solves. A device when used to solve a problem delivers the result in the form of information or values specific to that particular problem statement. This data is valuable and has to be stored in a secure location. The amount of incoming data is also unknown in some cases and a storage location with a suitable size has to be selected in order to avoid any discrepancies. A suitable and practical transmission process has to be implemented so as to avoid any loss of data or unnecessary overhead. The growing applications of IoT devices have resulted in the requirement of a medium to manage and monitor the data they provide. This paper explains the implementation of a website that has been created to study and analyze the data coming from IoT devices using the most viable option for the transmission process. The data captured is used to display a real-time graph in accordance with the values received.

Users can register themselves on the website and make their profiles before studying the data and getting the graph displayed. They can keep track of their previous records to compare with the current data in order to check the improvements or descent in the performance. The website offers the functionality of displaying a multiple lined real-time graph which can be used to display the values received as co-ordinates. The users will have to enter specific details of their device during the registration process such as device id, number of sensors, timeframe, parameters etc. This information will be used to draw the corresponding graph with correct parameters to make it easy for the user to see the data through a neat lined graph making the studying process simpler.

### 2 BACKGROUND

The internet of things (IoT) is the network of physical devices, vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In this paper, the implementation of data transmission on two of the most popular IoT devices has been done. A computer which in this case is on the receiving end, acts as the server. The medium for graph analysis of the data sent on the server is a

website that supports various functionalities which are discussed later in the paper. A real-time graph is displayed on the website for the user to analyze and/or monitor the data coming from the external device. The server stores the data in a database and fetches it simultaneously to plot a real-time graph. Common networking protocols have been used to transmit the data from the device to the server. The incoming data are in the form of values for which suitable packages and libraries have been made use of. The transmission and plotting process work simultaneously to display a real-time graphical representation of the values.

### 3 DATA TRANSMISSION BETWEEN IOT DEVICE AND SERVER

The initial part of this paper involves connecting the IoT device and the server system so as to enable the transmission of data between the two devices. This paper focuses on two of the very popular IoT devices that are Raspberry Pi and Arduino. This entire setup procedure has two parts. The first part is the transmission of values to the server system from the IoT device. The second part is receiving the incoming values and storing them on the database server which has been explained later in the paper.

#### 3.1 RASPBERRY PI

The Raspberry Pi is a capable little IoT device that can do everything a desktop does. The calculations and computations to solve the problem statement is done on the Pi itself. This device can run various scripts and codes and acts like a system that can be attached to other devices such as sensors to read its output or act as a component in drones etc. To establish connection between the Raspberry Pi and the server, a common networking protocol called HTTP is used. HTTP makes use of the TCP/IP protocol for transferring the values in order to avoid losing the data packets coming from a remote machine. UDP is not used because in case a UDP packet is lost, it is not retransmitted and UDP has no error recovery mechanism to retrieve the lost value.

To enable transmission of values, a python script is written on the Raspberry Pi. It can either be written in a separate file or incorporated in the original python file. HTTP has a package called Requests available which makes the value transmission fairly simpler. The Requests package can be imported in the beginning of the code to make use of its functionalities. Using this package, the GET method is implemented in order to send the values to the server such that they directly get stored in the database server. Now the values to be transmitted can be stored in an array, list, dict type etc. This preferred data structure is passed as a parameter in the GET method along with the URL of the server. The URL has to have the IP address of the server and the path of the PHP script written on the server side which makes sure the values go to the correct location in the database. Here is the code snippet for the mentioned procedure:

```
#sample data set
data={'id':'1','value':'2000'}
#URL containing IP and path of the PHP script
url='http://10.56.17.246/py_insert/insert_val.php'
#passing parameters in get method
r=requests.get(url,params=data)
```

This ensures that the PHP script written on the server side is correctly called so as to take care of the values received on the server end. The final URL can be printed to check if the script is running correctly or not in the following way:

```
print(r.url)
```

Expected output should look something like this:

```
http://10.56.17.246/py_insert/insert_val.php?id=1&value=2000
```

The value corresponding to id i.e. 1 gets stored in the id column in the database and same goes for the value column. In case the values are in large quantity and they have to be transmitted in a similar way, they will have to be stored in an array and this array will have to be passed as a parameter in the GET method in Requests library in the following way:

```
for i in range (specify range):
```

```
url='http://10.56.17.246/py_insert/insert_val.php'
r=requests.get(url,params=arr)
```

The loop runs until the entire array is traversed and the values keep getting sent to the server with the help of the GET method in a similar fashion as was done for hard-coded values. This completes the Raspberry Pi's role in transmission of data

### 3.2 ARDUINO

Arduino is an open-source platform consisting of both a physical programmable circuit board and a piece of software, or IDE (Integrated Development Environment) that runs on your computer which can be used to write and upload computer code to the physical board.

Arduino has numerous applications in a wide variety of fields. Most of these projects require data handling, transmission, manipulation and analysis. The transmission process in Arduino uses the same GET method of Requests library in HTTP. A small piece of code can be added in the original code written on the Arduino's side to enable smooth transferring of data to the server. Again, transmission of values is focused upon so as to get real-time for the server to be able to plot a real-time graph of the corresponding values. The URL and the port number have to be declared in the code. The server's IP address is passed as a parameter in the GET method much like the case in Raspberry Pi. Following lines of code are required to be send hard-coded values to the server:

```
if (client.connect(Server_IP_addr, portNum)) {
  // Make a HTTP request:
  client.print("GET /py_insert/insert_val.php?id=1&value=2000");
  client.println("HTTP/1.1");
  client.print("Host:");
  client.println(Server_IP_addr);
  client.stop();
}
```

In case the values to be transmitted are stored in a variable, the variable itself is passed as a parameter in the GET method in the following manner:

```
//Define and declare the variable to be transmitted
int i=1;
int val=2000;
//To be written inside the transmission function
client.print("GET /py_insert/insert_val.php?id=");
client.print(i);
client.print("&value=");
client.print(val);
client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(Server_IP_addr);
```

This procedure is to be followed on the Arduino device's side in order to transmit values successfully to the server's database.

### 4 ROLE OF SERVER

The server is responsible for receiving the values coming from a remote device and storing it in a secure place accurately to keep it intact. The obvious choice for storage purpose is a database created on the server itself which can store large quantity of data. This database has to be first created on the server as per the device's requirements and then connected to the script that is responsible for loading the values into the database. The database can be connected using a basic PHP script which contains the username, password, localhost and the table name where the data has to be inserted.

The script that inserts the values into the desired table of the database contains an SQL query for the same. This SQL query makes use of the GET method and the exact same identification string which was used in the python script written on the Raspberry Pi has to be written in order to avoid any confusion and error. The SQL query in the PHP script that solves this purpose looks like this:

```
$$SQL = "INSERT INTO sensor_log.table_sensor_log (sensor_id, timestamp, sensor_value) VALUES ('. $_GET ["id"]."',
unix_timestamp(), "._GET["value"].")";
```

This method works for both hard-coded and real-time values. This completes the process of loading values into the database server.

The source which in this case is an IoT device transmits data using the python script mentioned earlier in the paper using the HTTP Requests library. Interfacing between the remote device and the server is done so as to establish a connection. This is the base on top of which the actual transmission takes place. The dispatched data traverses this medium and reaches the database created on the server. The populating of data in the database in the correct corresponding table and column is made sure by the virtue of the PHP script written on the server itself which contains the necessary SQL query to do the same. Once the database has been populated, the values are fetched in order to plot the graph for the user. This part is implemented by the back-end of the website. The back-end acts as the intermediate stage between the source and the server system as it fetches the values and supplies them to the server. The graph plotting is also a part of the back-end and it uses the fetched values to display the graph on the front-end. The whole transmission procedure runs on TCP/IP protocol using HTTP request and response mechanism between the two systems.

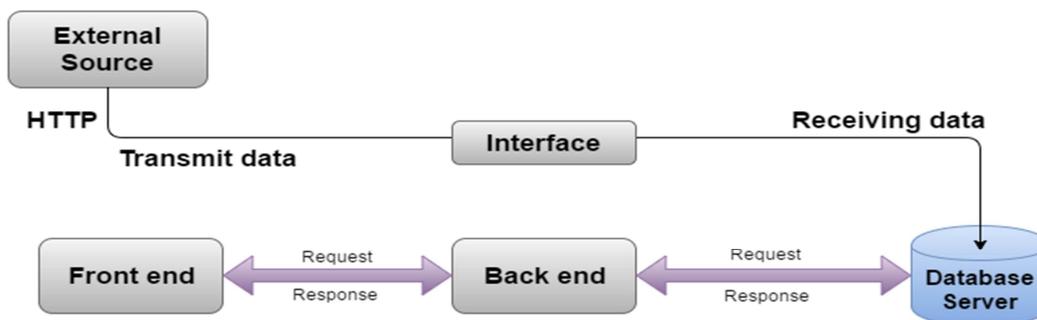


Fig. 1. The diagram shows the flow of data from the source and its handling on the server

## 5 WEBSITE IMPLEMENTATION

The two parts of website are the front-end which is responsible for the user-interface and the back-end which implements the various functionalities shown by the user-interface. The entire graphical representation medium i.e. the website makes use of the Model-View-Controller framework (MVC) which is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**. Each of these components is built to handle specific development aspects of an application. A Model View Controller pattern is made up of the following three parts:

- 1. Model:** The Model component corresponds to all the data related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic related data. The Model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.
- 2. View:** The View component is used for all the UI logic of the application. For example, the Customer view would include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with. It is basically a presentation of data in a particular format, triggered by a controller's decision to present the data.
- 3. Controller:** Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. The controller is responsible for responding to user input and performs interactions on the data model objects. The controller receives the input, validates it and then performs the business operation that modifies the state of the data model.

## 6 BACK END

The back-end of this website consists of creating various functionalities involved in the website. The modules are to be coded individually and integrated later. The various parts of the back-end are:

- Creating a suitable database on the server as per requirements.
- Deciding the credentials required to access the database through a PHP file.
- Registration page.
- Login page.
- Connecting the registration page and login page to the database.
- Creating an authentication system that verifies the user's credentials.
- Fetching values from the database.
- Using a suitable data structure to store the values and calling the function to plot it.
- Storing and displaying the overview of the device.

The system of the database is taken care entirely by the back-end. The maintenance of the three principles of security which are confidentiality, integrity and authority in the database is also handled by the back-end. The website gives the functionality of registering multiple devices under the same username. This is done so that a person's account doesn't stay limited to just one device. A unique device id is given after registration of every device. This unique string can then be entered in the login page along with the password to monitor and study the data of that particular device.

Since the value transmission process is fast, more than one value reaches the database from the source. This might create a discrepancy when plotting the graph as a device can have only one value at a time on the graph. Hence, the following query is written to avoid this problem:

```
"SELECT distinct (unix_timestamp) as times, device_id FROM table_sensor_log WHERE device_id = " . $deviceId . """;
```

The above mentioned modules have been implemented using PHP and JavaScript codes and later called or included in the final HTML file that displays the webpage. The website uses MVC framework as mentioned earlier in the paper and for the back-end part, the two components of MVC are used namely Model and Controller.

The Model part involves creating a connection between the database present on the server and the login page where a user enters his/her credentials to login. The definitions of the authentication details of the database such as hostname, username, password and database name are specified. A function is created to get the necessary details of the user and another function checks the credentials and returns the value to a separate PHP file which is the Controller part of the login procedure.

The Controller part involves verification procedure of the user login and returns the corresponding value to the main file PHP file. It is also responsible for the logout event of the website.

A separate PHP file is written which is outside the Model-View-Controller part. It is used for the actions taking place on the login page i.e. filling of text fields and clicking on the submit button which triggers an event.

## 7 FRONT END

The front-end makes use of the View part. All the necessary images, themes and packages are stored here. It also keeps record of all the static attributes of the website which does not change according to the user's needs. It contains the images of the layout and the Bootstrap files required for the front-end. The Bootstrap files are basically the CSS, JavaScript and font folders that are to be included to match the requirements of the front-end. These files have classes for designing and implementation which are used later in the main file for the final webpage. Another feature called Templates is used inside View. This contains two sub parts namely Layout and Partial. Layout is responsible for deciding the actual layout of the web pages that are involved in the project. The layout for the home page which in this case is the login page itself is where the user can enter his/her credentials to access his/her profile. A separate file is used which creates the dashboard. This is used when the user successfully logs in and his/her profile gets displayed. The profile has a dashboard with various functionalities related to the analysis and device information. These layout files contain the libraries and packages of jQuery, Bootstrap, FusionCharts etc.

The Partial part contains files and PHP scripts that define the classes of features and functionalities implemented on the webpage. These are the classes that can be used directly later and hence act as small modules or API's for the front-end. In

this project, the Partials folder contains a PHP file for the admin layout and three other files for navigation bar, login form and sidebar. Twig is a composer used for combining the previously created files for various minor or major functionalities. A composer called Twig is used which is used to combine the layout pages created earlier in the main PHP file.

## 8 APPLICATIONS

An analysis based application is a necessity in almost all the projects that are under development or are being used already. Monitoring and representation of data is required everywhere in so as to be able to understand it and draw logical conclusions. Below are a couple of applications where this project is already being implemented:

### 8.1 BREATH ANALYSER

A breath analyser analyses the breath of the user using sensors which transmits value as analog output. This analog output is converted to a digital output using an Analog to Digital Converter (ADC) and received by the IoT device. This device can run the python script on its end and transmit the values to the server system. The server system fetches the values from the database to plot a real-time graph of the same. This will help the users monitor and get a good understanding of the data through a graphical representation.

### 8.2 DIET MONITOR

This device is used to monitor the amount of fat burn in the user's body. It has a gas sensor and a pulse sensor to measure the acetone level in the user's breath and the pulse of the user respectively. Accordingly, it generates ppm values for acetone and pulse value. If both these values are in a specified range, the device draws the conclusion that fat burn is possible. This generates data such as ppm values, pulse and amount of fat burn which have to be stored and analysed so that users can study their body better to lose fat accordingly.

## 9 CONCLUSION

Data is only valuable to us if it can be understood to draw meaningful conclusions about the source of the data. This makes the analysis the most important procedure when it comes to handling data. Many more features can be inculcated in order to support the graphical representation of the data for better analysis. For example, one can take the current data set, train it using a simple training algorithm and then implement a prediction algorithm to predict the future values on the server side system. Hence if given a proper investment of time and resources, this system can be applied to numerous projects in a wide variety of fields.

## ACKNOWLEDGEMENT

This work was done as a part of summer internship at Centre for Nano Science and Engineering, Indian Institute of Science, Bangalore during the summer of 2016.

## REFERENCES

- [1] Loble, A., Lee Whitmore, and B. A. Wallace. "DICHROWEB: an interactive website for the analysis of protein secondary structure from circular dichroism spectra." *Bioinformatics* 18.1 (2002): 211-212.
- [2] Jain, Eric, et al. "Infrastructure for the life sciences: design and implementation of the UniProt website." *BMC bioinformatics* 10.1 (2009): 1.
- [3] Boute, Wim, and Jos Van Hove. "System and method for transmission of medical and like data from a patient to a dedicated internet website." U.S. Patent No. 7,082,334. 25 Jul. 2006.
- [4] Richardson, Matt, and Shawn Wallace. *Getting started with raspberry Pi*. "O'Reilly Media, Inc.", 2012.
- [5] Vujović, Vladimir, and Mirjana Maksimović. "Raspberry Pi as a wireless sensor node: performances and constraints." *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014.
- [6] *Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014.
- [7] Banerjee, Soham, et al. "Secure sensor node with Raspberry Pi." *Multimedia, Signal Processing and Communication Technologies (IMPACT), 2013 International Conference on*. IEEE, 2013.