



# Development of Message Queuing Telemetry Transport (MQTT) based Vehicle Accident Notification System

Bharat Naresh Bansal, Vivek Garg

**Abstract-** In the recent years there has been a tremendous growth in the field of engineering and sciences, which aided in the growth and development of fast and comfortable transportation media, with this development the number of automobiles have drastically increased, which for sure is a great technological achievement but sadly with this growth, the traffic and the hustle and bustle on roads is unstoppable and with it the number of accidents and road casualties have tremendously increased. But, there is no easy and practical way to reduce the usage of the automobiles. Every day the mankind read about thousands of people dying of road casualties and most of them die because the families or the concerned ones of the indulged people are not timely informed. The death casualties can be minimized to a great extent by just timely informing the families of the concerned ones. The prototype in this paper is an accident notification system ESP8266 NodeMCU and a simple vibration sensor is the heart of this system. The vibration sensor continuously senses the vibrations and on exceeding a predefined threshold limit, sends out a notification to registered numbers. In the past similar models have been proposed, which used costlier sensors such as Accelerometer but the design in this paper, used simpler and cheaper sensor. Moreover, in earlier designs GSM technology was used but proposed design uses a Wi-Fi based controller, which in comparison to GSM technology is more reliable and fast. Also earlier GSM module needed an additional microcontroller such as Arduino but the use of NodeMCU eliminates the requirement of any additional controller. The prototype system in this paper makes the use of message queuing telemetry transport (MQTT) protocol, which is a very reliable and fast communication protocol which further uses subscribe and publish technology. The IoT cloud platform used in this prototype is Adafruit IO which is quite simpler when compared to other cloud platforms such as Losant Platform and moreover the data is updated every two seconds in Adafruit IO. For the notification purpose protocol is used with the help of IFTTT platform and ClickSend platform, Applets and Triggers are created to fulfill the requirement. The controller is programmed using basic C and C++ programming languages and Arduino IDE serves as the programming environment, various library files have also been used for the programming purposes.

**Keywords:** IoT; Mqtt; Accident; Adafruit-IO; ClickSend; IFTTT; Notification.

## I. INTRODUCTION

Every day the mankind read about thousands of people dying of road casualties because their concerned ones did

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Bharat Naresh Bansal**, Assistant Professor in the department of Electronics and Communication Engineering at MIMIT, Malout, Punjab, India.

**Vivek Garg**, B. Tech Electronics and Communication Engineering at MIMIT, Malout, Punjab, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

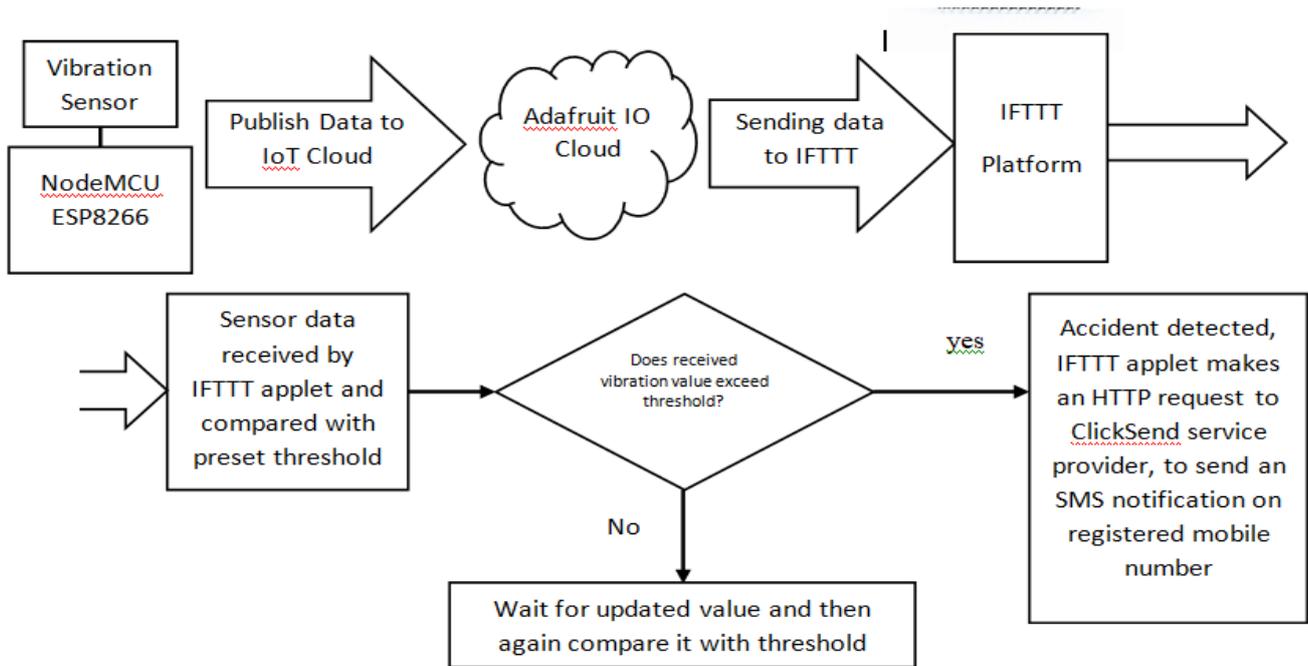
not get informed on time. This paper includes a proposed system for Accident notification, which if used in automobiles can save millions of lives and that too by just timely informing the concerned ones of the people involved in the casualty. This proposed system aims to save those lives, which can be saved if information of casualty is timely conveyed. The proposed system implements a low-power vibration sensor, a cost-effective microcontroller and a powerful IoT platform to create an application that does not depend on humans to function. This un-involvement of humans makes it fast and more reliable. Basically this prototype is an IoT system which works on the MQTT (Message Queuing Telemetry Transport) protocol. The core of this application is a NodeMCU, a very economic and extremely popular Wi-Fi based microcontroller which takes in readings from a vibration sensor. The vibration sensor constantly senses the vibrations and that data is constantly published (MQTT protocol) on the Adafruit IO cloud by the NodeMCU and also the data is being updated to the subscriber (which is IFTTT service provider) and whenever any abnormal vibrations are detected in the vehicle or the vibrations exceeds the preset threshold limit, immediately the applet created on FTTT platform makes an http (hypertext transfer protocol) request to ClickSend service provider and the email or SMS is sent to the registered mobile phone numbers. In the past, similar prototype systems to solve the same problem i.e. accident notification, have been proposed which uses either an accelerometer or GSM module which are far more expensive than a vibration sensor, also the cloud platform used in the earlier prototype was Losant[5][6][7][8] while here Adafruit IO cloud platform is used, which is comparatively simpler and easy to use, also the circuit complexity of the proposed prototype is quite simple and easy to understand as compared to the previously proposed prototype systems. The accident notification proposed system can also be used with the house doors, whenever the doors are forcefully tried to open, registered users will be notified on their phone.

## II. IMPLEMENTATION AND DESIGN

The basic block diagram that depicts the working flow of the proposed prototype has been shown below in Fig.1, it also consists of the flowchart that describes the conditional flow of the proposed prototype, where in a threshold limit is set for the vibration sensor values, which are being constantly published to the cloud platform with the help of the gateway (Node MCU),

if the values published on the cloud platform exceeds the preset threshold, the trigger created on IFTTT gets activated and performs the task assigned to it i.e. sends the

notification on registered numbers, the detailed description has been provided further in the paper.



**Figure. 1 Block Diagram Representation**

Figure 1 Shows the basic block diagram of the proposed accident notification system, the above figure describes the flow of data in the proposed system, the vibration sensor is interfaced to the NodeMCU microcontroller[4], which continuously reads the sensor data and publishes it to the Adafruit IO platform using Mqtt protocol and the IoT platform continuously sends the data to IFTTT, which has subscribed to receive the sensor data, the applet created on the IFTTT compares the received sensor value with preset threshold, and if the value exceeds threshold, IFTTT makes an http request to ClickSend to send an SMS notification on registered mobile numbers.

**A. NodeMCU ESP8266**

Figure 2 Shows a NodeMCU, it is an open-source Wi-Fi enabled microcontroller that helps to design IoT system. The firmware uses the Lua scripting language [1][9].

It can also be programmed using Arduino programming language (i.e. C & C++). NodeMCU acts as a gateway for the proposed IoT based vehicle accident notification prototype.

It continuously reads the sensor values and publishes it to the Adafruit Io cloud and then the cloud publishes the value to subscribed clients. NodeMCU eliminates the requirement of any additional microcontroller as NodeMCU itself is a Wi-Fi enabled SOC.



**Figure2. NodeMCU ESP8266 Microcontroller**

**B. Vibration Sensor**

Figure 3 shows a Vibration sensor which is nothing but a switch. It senses any kind of movement or vibration. When the switch is moved, the spring touches the center pole to make contact. So, when there's motion, the two pins will act like a closed switch.

When everything is still, the switch is open. It is connected to an analog pin on microcontroller board and the analog values are read and further conditioning can be applied on those values to perform any desired function. It is a low cost device and very simple to use and interface. The use of a vibration sensor for the proposed prototype reduces the cost up to a great extent, since a vibration sensor is cheaper.



**Figure 3. Vibration Sensor Switch**

**C. Message Queuing Telemetry Transport (MQTT)**

MQTT stands for Message Queuing Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol.

The design principles are to reduce network bandwidth and device requirements to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium. We can also encrypt the communication protocol[3][10].

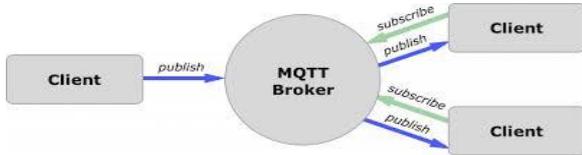


Figure 4. MQTT Protocol

Figure 4 shows the basic working principle of the MQTT protocol which is based on Subscribe and publishing of data between clients through a MQTT broker.

D. Adafruit IO IOT Platform

Adafruit IO is an easy-to-use and powerful enterprise IoT platform designed to help quick and secure building of complex systems. Adafruit IO uses open communication standards like MQTT to provide connectivity from one to millions of devices. Adafruit IO provides powerful data collection, aggregation, and visualization features to help understand and quantify vast amounts of sensor data. The following are the steps for connecting to Adafruit IO:

Step 1: Create an account on Adafruit IO platform, set up the credentials i.e. username, password and an AIO key will be generated once the account is created, copy that (it will be needed for programming).

Step 2: Create new a feed, as shown in Fig. 5.

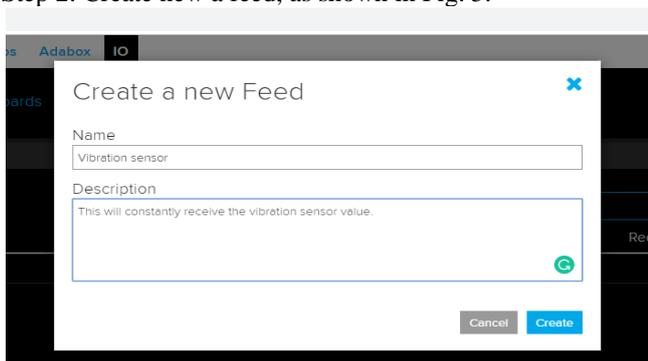


Figure 5. Creating a new Feed

Step 3: Create a new dashboard and add above created field to it

Step 4: Add the desired widgets to the dashboard

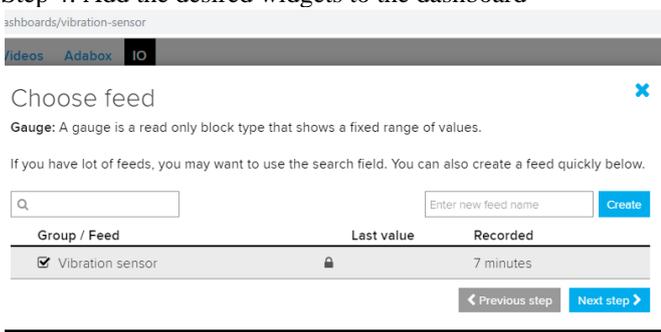


Figure 6. Addition of widgets

Figure 6 Shows the linking of feed created to the dashboard and addition of widgets to the dashboard.

E. IFTTT (If This Then That)

If This Then That, also known as IFTTT is a web-based service to create chains of simple conditional statements such as if-else, called applets.

An applet is triggered by changes that occur within other web services such as Gmail, Click Send SMS. For example, an applet may send an e-mail message if the value on vibration sensor exceeds the prescribed limit. In addition to the web-based application, the service runs on iOS and Android.

IFTTT is used for controlling the home appliances using google assistant of home by creating applets and triggers on IFTTT. The following are the steps for connecting to IFTTT:

Figure 7 displays the widgets added to the dashboard, the vibration sensor data will be displayed on these widgets, the left widget is a gauge which will display the present value of vibration sensor, which gets updated every 2 seconds and the right widget is a line chart which can display a graphical representation of values up to last 30 days.



Figure 7. Widgets

Step 1: Create an account on IFTTT platform (make sure to use the same email account).

Step 2: Search for Adafruit services and select it.

Step 3: After selecting Adafruit services, click on create new applet and create a new applet with a threshold value for vibration sensor as shown in Fig. 8., this applet works as a trigger.



Figure 8. Applet for vibration sensor

Figure 8 shows a trigger created to make an http request to ClickSend service provider, if the threshold value exceeds the value 5 (preset threshold for testing purposes).

F. ClickSend SMS

The SMS service is provided by ClickSend Service providers. ClickSend is a cloud-based gateway service that let user send bulk SMS, email, fax & letters worldwide. ClickSend services along with IFTTT is used for automation purposes, as it is used for Accident Notification system in this paper by creating trigger and applets. The steps for connecting to the service provider are same as above. IFTTT sends the http request to ClickSend and ClickSend responds to the request by sending an SMS notification on mobile phone as shown in Figure 9.



Figure 9. ClickSendSMS

III. HARDWARE REALIZATION

The basic hardware circuitry of the vehicle accident notification prototype is shown in the Fig.10 and Fig.11, The circuitry consists of a vibration sensor, a NodeMCU and a resistor interfaced together as shown below. The vibration sensor’s one pin is connected to the Analog pin of NodeMCU and the other is grounded through a 10K Resistor. The circuitry and its realization is quite simpler and inexpensive when compared to the earlier proposed prototype (i.e. The one with Accelerometer, GSM Modules[7][8]and other costlier and complex sensors and devices) [5].

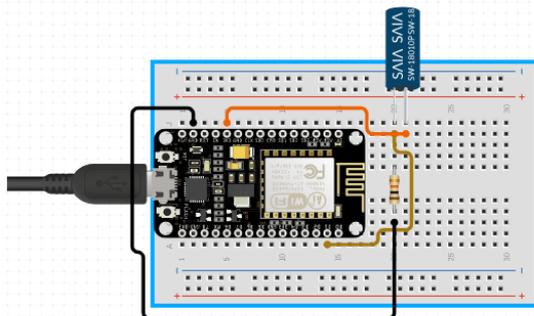


Figure 10. Hardware Realization of Accident Notification System

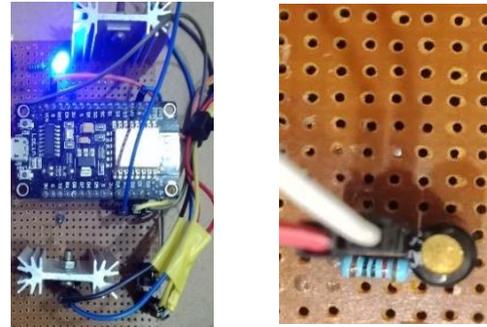


Figure 11 Magnified view of Hardware Circuit of Accident Notification System

Figure 10 & Figure11 depicts the basic hardware circuit of the vehicle accident notification system, the circuit consists of a vibration sensor whose one pin is connected to ground through a 10 kilo ohm resistor and the other pin is connected to analog input pin of NodeMCU. The realization of this circuit is very cheap and reliable as compared to the systems using Accelerometer, ultrasonic sensor and GSM Module.

IV. SOFTWARE REALIZATION

NodeMCU ESP8266 microcontroller is programmed with help of Arduino IDE and it is programmed in C/C++ language. NodeMCU cannot be directly programmed using Arduino IDE, for Programming NodeMCU with Arduino IDE, the NodeMCU board must be added to the software and for addition of the board the .JSON file is added in the preferences.

Step to add .JSON file: Open Arduino IDE software > Click on Files from Menu Bar > Select Preferences from the list > Paste the .JSON file (as shown in figure below).

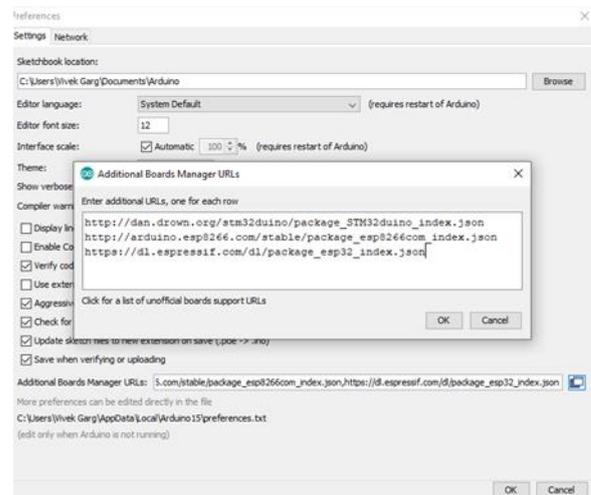


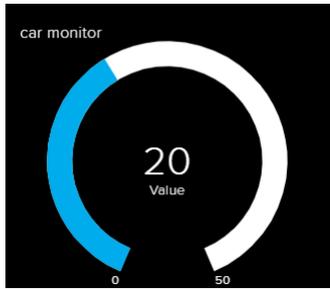
Figure 12. .JSON file Inclusion

Figure 12 depicts the additional boards manager window in Preferences where the .JSON file is included to add the additional NodeMCU ESP8266 board to Arduino IDE. The following libraries are also to be included while programming:

<ESP8266WiFi.h>, <Adafruit\_MQTT.h>, <Adafruit\_MQTT\_Client.h>,<Wire.h>

**V. TESTING AND RESULTS**

The accident notification prototype proposed in this paper has been tested in different circumstances and the results have been noted down and plotted. The results prove the proposed prototype to be working properly. The below shown in Figure 13 is a gauge which displays the vibration sensor value, which is updated every 2 seconds. The gauge ranges from 0 to 50, the limits can be changed w.r.t the practical environment.



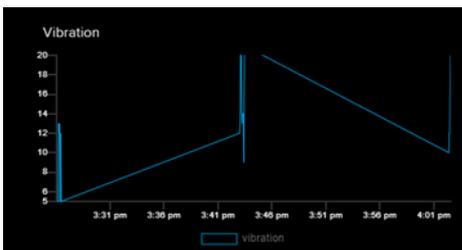
**Figure 13. Vibration sensor published on Gauge**

Figure 13 displays the Gauge which displays the vibration sensor values, the gauge value is updated every 2 seconds, the range of the gauge can be changed as per user requirement.

**Table 1. Test Values**

Time	Vibration Sensor Value
3:31 PM	6-7
3:38 PM	8-9
3:45 PM	Rapid fluctuations of values from 10 – 20
4:00 PM	8-10

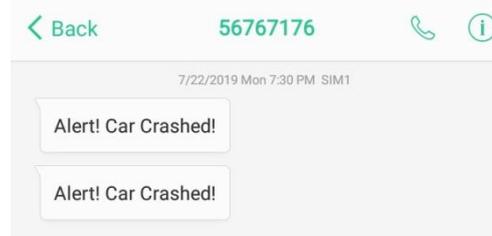
The Table 1. Display the test values of the designed system, for testing purposes the threshold value was set to a lower value but it should be increased for practical implementation in the automobiles. The sensor values of last 24 hours have been plotted below in the line graph, the similar plot can be obtained for last 30 days. Such plots are used analyze various conditions an automobile goes through. The plot is plotted between vibration sensor value on X axis and time on Y axis. Figure 14. shows the line chart which displays the vibration sensor values of last 24 hours.



**Figure 14. Graphical representation of vibration sensor values of last 24 hours**

When the value of vibration sensor exceeds the preset threshold, the trigger created on IFTTT gets activated and it

makes an http request to ClickSend service provider to send an SMS to the registered mobile phone number, as shown in Figure 15. The notification can also be received via an Email or even a computer generated voice call.



**Figure 15. SMS Notification Received**

**VI. CONCLUSION**

The designed prototype in this paper can help to save thousands of lives by timely informing the families of the concerned ones. The system designed makes use of a simple vibration sensor which senses the vibrations when the automobile is in motion. The vibration sensor continuously sends out these values to the NodeMCU which acts as a gateway between the Vibration sensor and Adafruit Io cloud platform. These values are uploaded to Adafruit every 2 seconds and are being send to the IFTTT, which is a subscriber of these values. And whenever the value exceeds preset threshold limit, IFTTT activates the trigger and as a result of trigger activation an HTTP request is made to Click Send SMS and then it sends out the notification to registered mobile phone numbers. In a nutshell, using NodeMCU along with a vibration sensor, a system can be developed which can help save many lives. This is a very simple and reliable design, the use of cheaper but reliable sensors makes it very affordable to implement in automobiles. The Accident notification prototype save lives by timely informing the concerned ones. There might be an issue of false alarms in case of speed breakers and ill maintained roads but this limitation could be overcome by increasing the threshold limit. As an extension, the future scope includes location of accident being informed to the desired persons such as nearest hospital, police and owner of vehicle through SMS sent containing coordinates obtained from GPS[2][8] along with time of accident and vehicle number. Also a camera can be located inside vehicle that will transmit real time video to see current situation of passengers inside vehicle.

**REFERENCES**

1. Behzad D. and Fujimura K.(1999, 5-8 October), "A framework for driver specific inference of danger at signalized intersections", International Conference on Intelligent Transportation Systems, PP. 195-200, Tokyo, Japan. doi:10.1109/ITSC.1999.821053.
2. C. Prabha, R. Sunitha, R. Anitha (2014, July), "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 3(7), 10723-10727. doi:10.15662/ijareeie.2014.0307062.

3. Fang C., Yu R., Huang T., Liu J. and Liu Y.(2015, February), "A Survey of Green Information-Centric Networking: Research Issues and Challenges", IEEE Communications Surveys & Tutorials,17(3), 1455–1472. doi: 10.1109/COMST.2015.2394307
4. Belshe M., Peon R., Thomson M.(2015), "Hypertext Transfer Protocol Version 2 (HTTP/2)", Internet Engineering Task Force.
5. Kushwaha Singh Vikram, Yadav Deepa and Topinkatti Abuyeed and Kumari Amrita (2015, May-June), "Car Accident Detection System using GPS And GSM", International Journal of Engineering Research and General Science, 3(3),12-17.
6. Kodali Kishore Ravi and Sahu Archana(2016, December), "An IoT based soil moisture monitoring on Losant platform, International Conference on Contemporary Computing and Informatics",PP.612-616, Noida, India. doi:10.1109/ic3i.2016.7918063.
7. Kodali K Ravi and Sahu, Shubhi (2017, December), "MQTT Based Vehicle Accident Detection and Alert System", International Conference on Applied and Theoretical Computing and Communication Technology, PP.186-189, Tumkur, India. doi: 10.1109/ICATCCT.2017.8389130
8. Chaturvedi Nimisha and Srivastava Pallika (2018,March) , "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem", International Research Journal of Engineering and Technology,5 (3),252-254.
9. Kashyap Monika, Sharma Vidushi and Gupta Neeti(2018), "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things", International Conference on Computational Intelligence and Data Science, PP.1611-1618, doi:10.1016/j.procs.2018.05.126.
10. Eurotech, International Business Machines Corporation (1999-2010), MQTT protocol specification retrieved from [http://public.dhe.ibm.com/software/dw/web\\_services/ws-mqtt/mqtt-v3r1.html](http://public.dhe.ibm.com/software/dw/web_services/ws-mqtt/mqtt-v3r1.html).

## AUTHORS PROFILE



**Bharat Nareseh Bansal** is working as Incharge and Assistant Professor in the department of Electronics and Communication Engineering at MIMIT, Malout, Punjab, India. He received his B.Tech. from Amaravati University, M.Tech. From PTU, Jalandhar and PhD from IKGPTU, Jalandhar. He has received the best teacher award from IST student chapter in March 2016. He has fourteen years of work experience. He has to his credit around twenty three research papers in journals and conferences. His areas of interest are Communication & Instrumentation. He is a life member of various technical associations. He has delivered various expert lecture sessions in Engineering Institutions across the state and have recorded lectures for EDUSAT.



**Vivek Garg** is a final year student of B. Tech Electronics and Communication Engineering at MIMIT, Malout, Punjab, India. He was amongst the Top three students in the university end semester examinations (Dec 2016). He has been trained in Embedded System and Internet of Things (IoT). He got placed in Cognizant as a Software Engineer in his final year. Also he has worked on various projects such as Wi-Fi based Home Automation System, Bluetooth Enabled voice controlled car, RFID based attendance system with database management and Hand Gesture controlled Car. His areas of interest are Embedded Systems, Wireless Communication and IoT Project Development.