iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 6 **Issue:** VIII **Month of publication:** August 2018

**DOI:**

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VIII, August 2018- Available at www.ijraset.com

# Different Approaches to Crack Monitoring of Buildings using IOT

Asst Prof. Kiran V.K[1] , Assoc Prof. Mohandas K A [2] , Abhishek T V[3], Akhila[4], Arya Krishna V R[5], Jafsal M A[6], Greeshma P7,
[1]Dept. of CSE , [2]Dept. of CE , [1, 2, 3, 4, 5, 6, 7]NSS College of Engineering, Palakkad, India

***Abstract: Internet of things (IoT) is a system of physical objects that can be connected to the Internet. Numerous DIY (Do It Yourself) kits for IoT based on Raspberry Pi and ESP32 are available. In our work, we try to implement an automated building monitoring system that gives an idea about nature of cracks in buildings. Different types of crack monitoring methods are available nowadays. Some of these methods are Monitoring Cracks Using Tape and Pencil, Glass and Epoxy, and paper. All these methods are mechanical and requires a human operator for taking measurements. We propose two approaches for an automated method which requires human intervention only during installation and disassembly of the system. IoT Platforms enable our solution to be accessible through web browser, mobile apps and other Web based front ends facilitating a user friendly interface.***

***Index Terms: IoT, IoT Platforms, AWS IoT, AWS Greengrass Wi-Fi, MQTT, Arduino, ESP32, Raspberry Pi, Sensor data***

## I. INTRODUCTION

The internet of things [1][3]is an evolution of mobile, home and embedded appliances that will be connected to the internet integrating greater capabilities and using data analytics to extract meaningful information. Our work proposes to design and build a system for real-time monitoring of cracks in buildings. Cracks may occur in buildings due to incorrect design, faulty construction, overloading & internal induced stress in building material. So it is necessary to identify these cracks, measure its width, rate of growth & discover whether the crack is active or passive. Two systems for monitoring of cracks have been proposed. The proposed systems is expected to be easily configurable using commercial off-the-shelf DIY (Do It Yourself) kits. Hence given the system design and installed software anyone can recreate and modify the system for specific purposes. The system is expected to be cheap , though reliable. The proposed system allows the users to interact with it using web protocols enabling easy integration with other monitoring systems. The system is expected to provide a quantitative measure of crack growth and recommend optimal positioning of sensors for following cracks in direction of its growth. For this we use low cost DIY camera module which will take image of crack periodically and will show the direction of crack growth using image recognition software.

## II. MOTIVATION

Cracks on the concrete structures are one of the earliest indications of degradation. Its maintenance is important and continuous exposure will lead to damage the environment. So cracks must be monitored to assess the damage progress and to maintain the structural integrity of the building. In current scenario there is no proper method for crack monitoring [7] in buildings bridges and roads. In these mechanical methods sketch of the cracks are taken manually and irregularities are noted. This will not provide accurate measurements because manual approach is completely dependent on specialist knowledge and lacks in quantitative analysis. Some of the mechanical methods are monitoring of cracks using tape and pencil, glass and epoxy and using paper. No other techniques have yet been established so far. In the above mentioned systems engineers have to visit the place to check for details regarding the crack. So we are introducing a real time automated system which will give continuous information about the rate of growth of the crack to engineers in their remote computer systems. This would allow online monitoring and continuous supervision of cracks giving easy access to data recorded from electronic sensors [11] improving its safety. This enables the engineers to save both their time and money needed to visit the site. Also, in case of public buildings the application of our system reduces the chances of a mass panic due to the engineer’s visit at the location, for cracks, hence leading to the belief of the building being unstable. This system can be used for monitoring cracks in multistoried buildings, water retaining structures, located in earthquake prone zones and weak soil conditions. This system can also be used for studies regarding the cracks in academic fields. Since, this system uses wireless modules the use of bulky wires and external agencies for connecting devices is not needed thus providing mobility and scalability.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VIII, August 2018- Available at www.ijraset.com

## III. EXISTING METHODOLOGY

All of the existing methodologies are mechanical and requires human intervention for taking measurements. Major drawbacks of these methods are location of the crack has to be known prior and difficulty in automation. Also these are labor intensive. These methods include:

### *A. Monitoring Cracks Using Tape And Pencil*


Fig. 1. Monitoring cracks using tape and pencil

In this method, a high quality sticky piece of paper is placed on each side of the crack. A short line is drawn on each side of tape and a ruler is used to take measurements. If there is a movement in the crack then the line on the tape will vary. This method is shown in Fig 1.

### *B. Monitoring Cracks Using Glass And Epoxy*

A small piece of single strength window glass is used to bridge over the crack. Epoxy the ends of the glass to the masonry on either side of the crack. If the glass breaks it is an indication that the walls are still moving and that the crack is still widening. This method is shown in Fig 2.

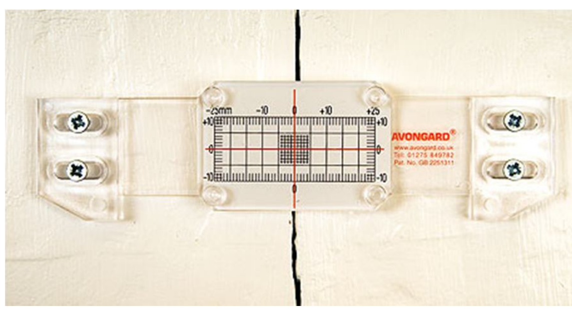


Fig. 2. Monitoring cracks using glass and epoxy

### *C. Monitoring of Cracks Using Paper*

A strip of paper is used to cover the detected crack. If this covered paper is riven then we can conclude that crack is active otherwise we conclude that the crack is passive. This requires engineers to visit the place repeatedly to verify whether the crack has become active or passive. This method is shown in Fig 3.

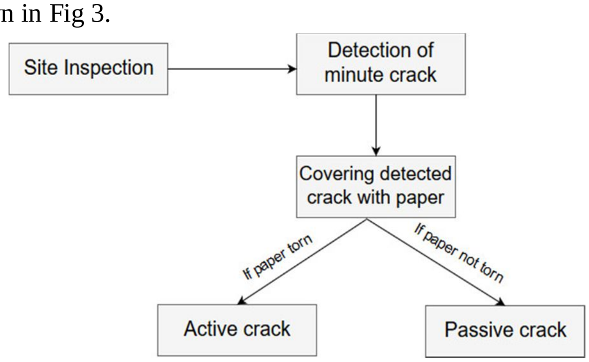


Fig. 3. Monitoring of cracks using paper

## IV. 1st APPROACH

Cracks may occur in buildings due to temperature and climate changes, foundation movements and due to internal and external stresses in buildings. So it is necessary to identify these cracks, measure its width, rate of growth & discover whether the crack is active or passive. Our work aims to simplify these tasks for an engineer by monitoring the cracks & finding its status using sensors [10]. This work consists of both hardware and software part. The data is collected in the hardware part by making things "smart" and this collected data is stored in the cloud [8] using an IoT platform [12][13]. In the hardware part the sensors [6] installed in the buildings will give the crack status to ESP32 [1] and since ESP32 has inbuilt Wi-Fi module a wireless connection is established between ESP32 and Raspberry pi [2]. AWS IoT is used as the IoT platform. From this platform users can access the crack status using mobile apps, web browsers, tablets or other web based front ends. This overall architecture [4] is illustrated in the Fig 4.

### *A. Algorithmic Operations*

1) Install sensors at appropriate places in buildings. Also a camera is installed in order to determine the rate of growth of cracks.
2) Sensors and cameras are connected to individual ESP32 using wires.
3) Emission of light to buildings by sensors to detect any presence of crack. Crack presence is determined by measuring how long the light has taken to bounce back to the sensor. Based on this measurement the sensor will return the distance between sensor and obstacle to ESP32 board. The camera placed will take images of crack for determining rate of growth of crack and passes that data to ESP32 boards.
4) A wireless connection is established since ESP32 is having built in Wi-Fi module.
5) Data collected by ESP32 are sent wirelessly to Raspberry Pi which acts as a mini computer.
6) Crack status is stored in the cloud platform, AWS IoT and users can access it using any web based front ends.

### *B. Hardware Part*

The hardware part can be installed in any intended building where the user wishes to know about the crack status. For installing this hardware part the user only needs to buy sensors, ESP32, camera and Raspberry Pi. These devices are always available and the only thing user has to do is to connect these devices. For getting building status the sensor used is VL6180X Time of Flight Micro-LIDAR Distance sensor. This work consists of four such sensors for monitoring the cracks. Sensors are arranged at one side of the crack and obstacles are placed on the other side. . Each sensor is controlled by separate ESP32 board using wires. All these four ESP32 boards are wirelessly connected to a Raspberry Pi board [15] for data collection.

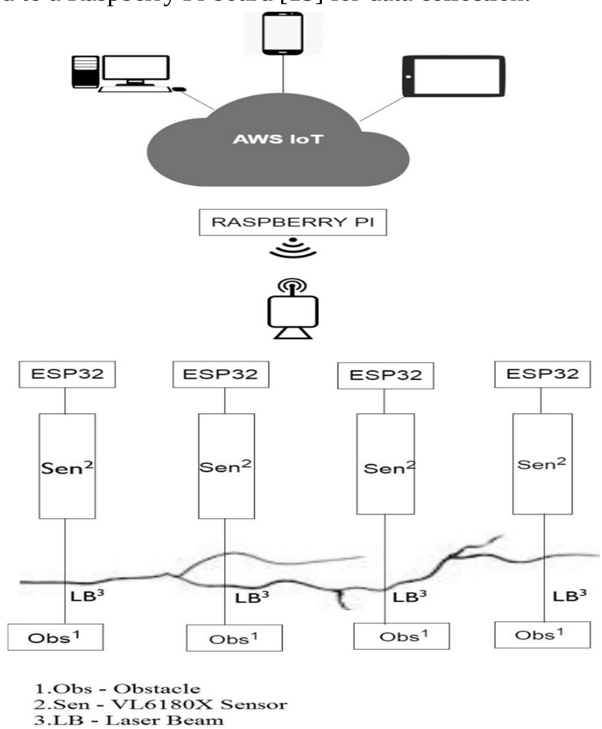


Fig. 4. Overall Architecture

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887*
*Volume 6 Issue VIII, August 2018- Available at www.ijraset.com*

The sensor emits laser light towards the obstacle across the crack and it detects how long the light has taken to bounce back to the sensor. Based on time of flight measurement the sensor returns the approximate distance between sensor and obstacle to the ESP32

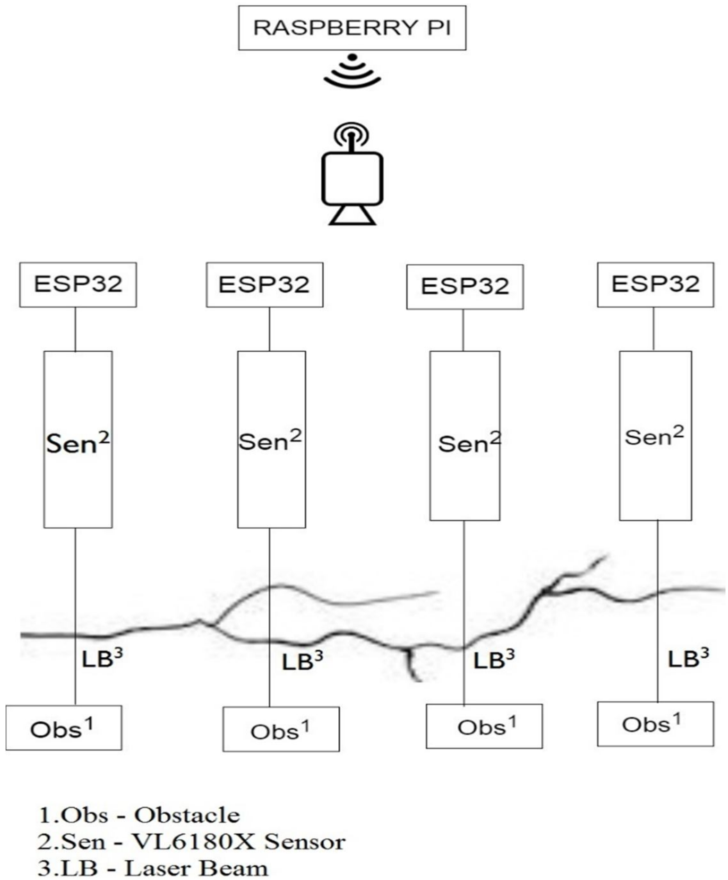


Fig. 5.Sensor data collection diagram

The system is also expected to recommend optimal positioning of sensors for following cracks in direction of its growth. For this we use low cost DIY camera module which will take image of crack periodically and will show the direction of crack growth using image recognition software. This camera is connected to Raspberry Pi using Wi-Fi module. Its pictorial representation is shown in Fig 6.Remote monitoring of these information is carried through internet using IoT platforms. This work allows engineers to know about the status of cracks in buildings from anywhere they want thus saving their time.

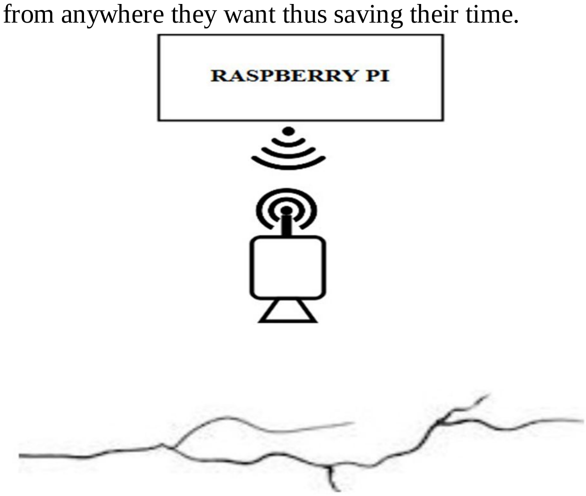


Fig. 6.Camera module diagram

 

### C. Software Part

The software part used is an IoT platform, AWS IoT. The data from Raspberry Pi is loaded to this platform for online storage and data access. AWS IoT provides secure and bidirectional communication between internet-connected devices. Here the connected devices include sensors, ESP32 and Raspberry Pi. This cloud platform enables to collect data from multiple devices and also provides a way to store and analyze data. AWS IoT consists of the following components:

### D. Device Gateway

AWS IoT device gateway allows devices to securely and efficiently communicate with AWS IoT. This gateway can exchange messages using a publication/subscription model. It enables both one-to-one and one-to-many communications. Using this one-to-many communication pattern AWS IoT makes it possible for a connected device to broadcast data to multiple subscribers for a given topic.

### E. Message Broker

It provides a mechanism for devices and AWS IoT applications to publish and receive messages from each other. Communication takes place in such a way that a client will send a message to AWS IoT addressing to a specific topic. The message broker in turn sends that message to all clients who have registered to receive messages for that topic. The act of sending messages is called publishing and the act of registering to receive messages for a topic filter is called subscribing.

### F. Rules Engine

The rules engine helps to build IoT applications that gather, process, analyze and act on data generated by connected devices. It evaluates inbound messages that are published into AWS IoT and delivers them to another device or a cloud service based on the business rules that we define.

### G. Security and Identity Service

Provides security in the AWS cloud. The devices must keep their credentials safe in order to send data to the message broker securely. The message broker and rules engine use AWS security features to send data securely to devices or other AWS services.

### H. Registry

It assigns a unique identity to each device. It also tracks metadata such as devices attributes and capabilities of a device.

### I. Group registry

Group allows to manage several devices at once. A hierarchy of groups can be build. Any action that is performed on a parent group will be applied to its child groups and to all the devices in it. Permissions granted to a group will be applied to all devices in the group and to all of its child groups.

### J. Device Shadow

It is a JSON document which is used to store and retrieve the current state information of a device. Device shadow persist the last reported state and the desired future state of each device even when the device is offline.

### K. Device Shadow Service

It helps to publish updated state information to a device’s shadow and also the device can synchronize its state when it connects. It also helps the devices to publish their current state to a shadow which can be used by applications or other devices.

### L. Device Provisioning Service

It allows to provision devices using a template that will describe the resources required for a device such as a thing, a certificate, and one or more policies.

### M. Custom authentication service

Custom authorizers can be defined which allows to manage custom specific authentication and authorization strategy using a custom authentication service and a lambda function.

### N. Jobs Service

Allows to define a set of remote operations that are sent to and executed on one or more devices connected to AWS IoT.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VIII, August 2018- Available at www.ijraset.com

The protocol used for connection is MQTT (Message Queue Telemetry Transport) [1][14]. This protocol is widely used in IoT because it is a machine to machine (M2M) protocol and extremely light – weight. It uses publish – subscribe paradigm for message transfer. Its main function is to connect, publish and subscribe. It consists of a MQTT broker publisher and subscriber. In order to publish or subscribe messages the only need to know is the hostname and port of the broker. The broker will store messages for clients that are not online. The operation of this protocol is shown in Fig 7.

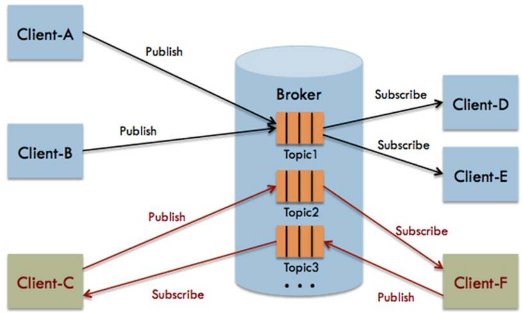


Fig. 7. MQTT Operation

## V. SYSTEM COMPONENTS

The components used to fulfill this work are:

### *A. Raspberry Pi 3 Model B*

It acts as a desktop computer for storing sensor data coming from ESP32. The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor. It is 80% faster than Raspberry Pi 2. Also it provides 10 times the performance of Raspberry Pi 1. It supports HDMI and Ethernet cable. It is having inbuilt camera and Bluetooth.

### *B. ESP32*

It is used for transmitting data from sensors to Raspberry Pi. There are four such ESP32 in our work for collecting data from four sensors. Then this data is transferred to a Raspberry Pi wirelessly. It has integrated Wi-Fi and dual-mode Bluetooth. It has a Tensilica Xtensa LX6 microprocessor and includes inbuilt antenna switches.

### *C. VL6180X Micro-Lidar Time of flight Distance Ranging Sensor*

This sensor is used for getting details about the rate of crack growth in buildings. It measures the time light takes to travel to the crack and reflect back to the sensor thereby providing crack growth. It consists of an IR emitter, a range sensor and an ambient light sensor making it a three-in-one smart optical module. This sensor is used because it can handle 5mm to 200mm range of distance and is having a very narrow cone of sensing. Unlike IR distance sensors which try to measure the amount of light bounced this sensor is much more precise and doesn't have linearity problems or double imaging where you can't tell an object is very far or very close.

### *D. Wi-Fi Module*

It is used for wireless connectivity. It is used here for transferring from camera to Raspberry Pi. Raspberry Pi and ESP32 is having inbuilt Wi-Fi module. So we do not want to provide Wi-Fi module externally for them.

### *E. Camera*

This is used to take images of crack periodically and will show the direction of crack growth. For this a low cost DIY camera module is used. This camera is connected to ESP32 using wires. The data collected from camera is then passed to Raspberry Pi using Wi-Fi module.

### *F. AWS IoT*

This is an IoT platform used to store data's coming from Raspberry Pi. It allows internet-connected devices to connect to the cloud. The applications in the cloud can interact with these devices. Devices will report their state by publishing messages in JSON format on MQTT topics. The messages that are published will be sent to the message broker, whose duty is to send all messages published on an MQTT topic to all clients subscribed to that topic.

### *G. Findings*

From table 1 the total cost of hardware components is Rs.11585 and that of AWS IoT subscription is Rs.384.33 per million messages (that is the number of messages published to AWS IoT and the number of messages delivered by AWS IoT to devices or applications).

Table 1. Components cost for 1st Approach

| HARDWARE PART | | | | |
|---|---|---|---|---|
| SL No. | Component | No's. | Price(in Rupees) | Total(in Rupees) |
| 1 | ESP32 | 4 | 899/- | 3596/- |
| 2 | Raspberry Pi 3 Model B | 1 | 3489/- | 3489/- |
| 3 | Wi-Fi Module | 1 | 2000/- | 2000/- |
| 4 | Camera module | 1 | 2500/- | 2500/- |
| TOTAL COST(in Rupees) 11585/- | | | | |
| SOFTWARE PART | | | | |
| SL No. | Component | Price(in Rupees) | | |
| 1 | AWS IoT Subscription | 384.33/- (per 1 million messages) | | |

Even though the cost of installation seem to be high for an individual, the hardware components can be reprogrammed by any authorized individual to meet the specifications of his various desired applications in the future. This trait makes the system highly desirable in spite of its installation overhead. In this project we are using this design for crack monitoring of buildings. But this design can also be modified and recreated for various other purposes too.

## VI. 2nd APPROACH

This work consists of both hardware and software part. The data is collected in the hardware part by making things "smart" and this collected data is stored in the cloud using an IoT platform. In the hardware part the sensors installed in the buildings will give the crack status to Arduino and for making the connection wireless the data from Arduino [9] is stored in Raspberry Pi [16] . AWS Greengrass is used as the IoT platform. From this platform users can access the crack status using mobile apps, web browsers, tablets or other web based front ends. This overall architecture is illustrated in the Fig 8.

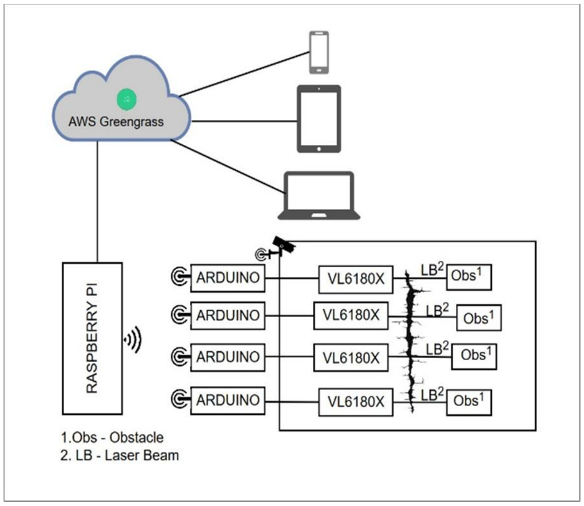


Fig. 8. Overall Architecture

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VIII, August 2018- Available at www.ijraset.com

### A. Algorithmic Operations

1) Install sensors at appropriate places in buildings. Also a camera is installed in order to determine the rate of growth of cracks.
2) Sensors and cameras are connected to individual Arduino boards using wires.
3) Emission of light to buildings by sensors to detect any presence of crack. Crack presence is determined by measuring how long the light has taken to bounce back to the sensor. Based on this measurement the sensor will return the distance between sensor and obstacle to Arduino board. The camera placed will take images of crack for determining rate of growth of crack and passes that data to Arduino boards.
4) A wireless connection is established by attaching Wi-Fi modules to Arduino boards.
5) Data collected by Arduino are sent wirelessly to Raspberry Pi which acts as a mini computer.
6) Crack status is stored in the cloud platform, AWS Greengrass and users can access it using any web based front ends.

### B. Hardware Part

The hardware part can be installed in any intended building where the user wishes to know about the crack status. For installing this hardware part the user only needs to buy sensors, Arduino, camera and Raspberry Pi. These devices are always available and the only thing user has to do is to connect these devices. For getting building status the sensor used is VL6180X Time of Flight Micro-LIDAR Distance sensor. This work consists of four such sensors for monitoring the cracks. Sensors are arranged at one side of the crack and obstacles are placed on the other side. Each sensor is controlled by separate Arduino board using wires. All these four Arduino boards are wirelessly connected to a Raspberry Pi board using Wi-Fi module [5] for data collection. The sensor emits laser light towards the obstacle across the crack and it detects how long the light has taken to bounce back to the sensor. Based on time of flight measurement the sensor returns the approximate distance between sensor and obstacle to the Arduino board. Arduino in turn returns the value back to Raspberry Pi [2] board using Wi-Fi module. This is diagrammatically represented in Fig 9.

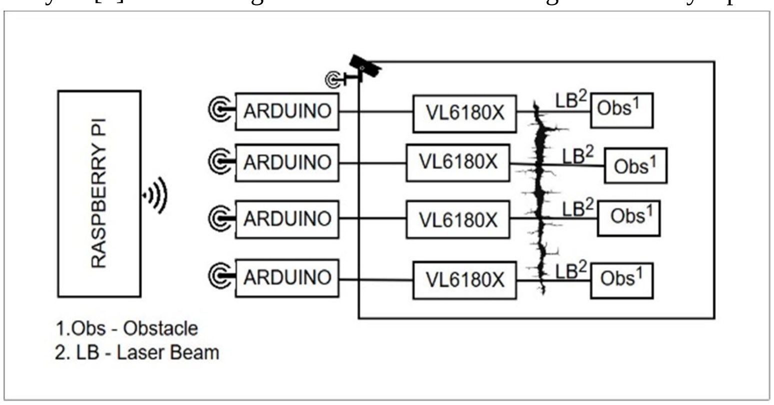


Fig. 9.Sensor data collection diagram

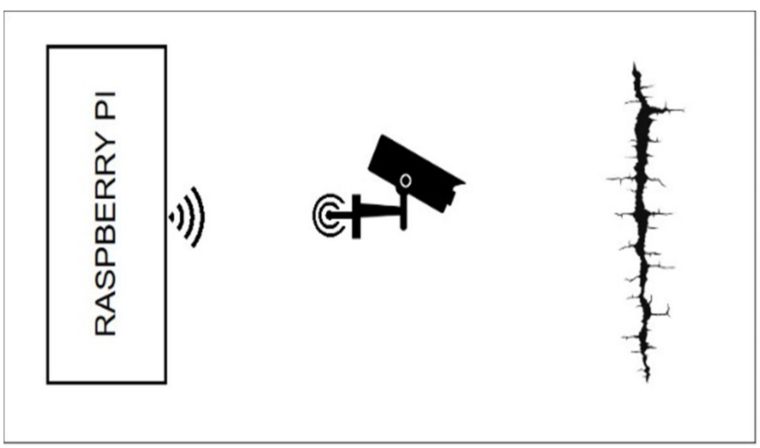


Fig. 10.Camera module diagram

The system is also expected to recommend optimal positioning of sensors for following cracks in direction of its growth. For this we use low cost DIY camera module which will take image of crack periodically and will show the direction of crack growth using image recognition software. This camera is connected to an Arduino using wires which in turn is connected to Raspberry Pi using Wi-Fi module. Its pictorial representation is shown in Fig 10.Remote monitoring of these information is carried through internet using WoT. This work allows engineers to know about the status of cracks in buildings from anywhere they want thus saving their time.

### *C. Software Part*

The software part used is an IoT platform, AWS Greengrass. The data from Raspberry Pi is loaded to this platform for online storage and data access [8] . In AWS Greengrass, connected devices will run AWS lambda functions and the device data will be synced and can communicate with other devices even when there is no internet connection. The purpose of using lambda function is to ensure that IoT devices can respond quickly to local events and it will minimize the cost of transmitting IoT data to the cloud. It can operate even in offline mode. It simplifies the device programming with AWS lambda. It is the Greengrass core that enables the local execution of AWS lambda. Actually this core acts as a hub to which other devices having AWS IoT Device SDK installed will communicate. AWS Greengrass core devices and AWS IoT device SDK-enabled devices is configured to communicate with each other in the Greengrass group. Even if the Greengrass core device loses connection to the cloud, devices in the Greengrass group can continue to communicate with each other using the local network.

The protocol used for connection is MQTT (Message Queue Telemetry Transport). This protocol is widely used in IoT because it is a machine to machine (M2M) protocol and extremely light – weight. It uses publish – subscribe paradigm for message transfer. Its main function is to connect, publish and subscribe. It consists of a MQTT broker publisher and subscriber. In order to publish or subscribe messages the only need to know is the hostname and port of the broker. The broker will store messages for clients that are not online.

### *D. System Components*

The components used to fulfill this work are:

1) *Raspberry Pi 3 Model B":* It acts as a desktop computer for storing sensor data coming from Arduino. The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor. It is 80% faster than Raspberry Pi 2. Also it provides 10 times the performance of Raspberry Pi 1. It supports HDMI and Ethernet cable. It is having inbuilt camera and Bluetooth.
2) *Arduino Uno R3:* It is used for transmitting data from sensors to Raspberry Pi. There are four such Arduino in our work for collecting data from four sensors. Then this data is transferred to a Raspberry Pi using a Wi-Fi module. It is having 14 digital input/output pins and 16 analog pins. It features the Atmega 8U2 microcontroller chip programmed as a USB to serial converter. It is simple, small in size, low cost and provides more memory and faster transfer rates
3) *VL6180X Micro-Lidar Time of flight distance ranging sensor:* This sensor is used for getting details about the rate of crack growth in buildings. It measures the time light takes to travel to the crack and reflect back to the sensor thereby providing crack growth. It consists of an IR emitter, a range sensor and an ambient light sensor making it a three-in-one smart optical module. This sensor is used because it can handle 5mm to 200mm range of distance and is having a very narrow cone of sensing. Unlike IR distance sensors which try to measure the amount of light bounced this sensor is much more precise and doesn't have linearity problems or double imaging where you can't tell an object is very far or very close.
4) *Wi-Fi Module:* It is used for wireless connectivity. It is used here for transferring from Arduino to Raspberry Pi. Raspberry Pi is having inbuilt Wi-Fi module but Arduino is not having inbuilt module. So we are providing Wi-Fi module externally.
5) *Camera:* This is used to take images of crack periodically and will show the direction of crack growth. For this a low cost DIY camera module is used. This camera is connected to an Arduino using wires. The data collected from camera is then passed to Raspberry Pi using Wi-Fi module
6) *AWS Greengrass:* This is an IoT platform used to store data's coming from Raspberry Pi. And it extends AWS cloud capabilities to local devices thereby allowing to store and analyze data both online and offline. They make use of AWS lambda functions for building IoT devices. It makes use of MQTT (Message Queue Telemetry Transport). The main peculiarity is that it can operate in offline mode. Since it is a machine to machine protocol and light weight it is adopted in IoT. This platform simplifies device programming with AWS lambda as well as reduce the cost of running IoT applications.
7) *Findings:* From table 2 the total cost of hardware components is Rs.20879 and that of AWS IoT subscription is Rs.384.33 per million messages (that is the number of messages published to AWS IoT and the number of messages delivered by AWS IoT to devices or applications).

Table2. Components Cost For 2nd Approach

| HARDWARE PART | | | | |
|---|---|---|---|---|
| SL No. | Component | No's. | Price(in Rupees) | Total(in Rupees) |
| 1 | Arduino Uno R3 | 5 | 990/- | 4950/- |
| 2 | Raspberry Pi 3 Model B | 1 | 3489/- | 3489/- |
| 3 | Wi-Fi Module | 5 | 2000/- | 10000/- |
| 4 | Camera module | 1 | 2500/- | 2500/- |
| TOTAL COST(in Rupees) 20879/- | | | | |

| SOFTWARE PART | | |
|---|---|---|
| SL No. | Component | Price(in Rupees) |
| 1 | AWS IoT Subscription | 384.33/- (per 1 million messages) |

Even though the cost of installation seem to be high for an individual, the hardware components can be reprogrammed by any authorized individual to meet the specifications of his various desired applications in the future. This trait makes the system highly desirable in spite of its installation overhead. In this project we are using this design for crack monitoring of buildings. But this design can also be modified and recreated for various other purposes too.

## VII. CONCLUSION

In this work, two designs for an automated building monitoring system have been designed to identify the nature of cracks in buildings. Rate of growth of cracks can be identified using this design and will allow engineers to find the status of cracks in their remote computer system. This system can minimize manual intervention since it has made things smart. This system also enables a user friendly interface by allowing the system to be accessible through many web based front ends.

## VIII. ACKNOWLEDGEMENT

The success and final outcome of this work required a lot of guidance and assistance from many people and we are privileged to thank all of them. First of all, we thank the Almighty God, for granting us the strength, courage and knowledge to complete this work successfully. We express our immense pleasure towards CERD (Centre for Engineering research and Development) for granting us basic financial support needed to complete this work. We would also like to thank all the teaching and non-teaching staff of our department for their constant encouragement throughout our project. This helped us in proper completion, installation and demonstration of our project. Last, but not the least, we take pleasant privilege in expressing our heart full thanks to our friends who were of precious help in completing this project.

## REFERENCES

[1] Kodali, Ravi Kishore, Snehashish Mandal, and Shahrukh Haider. "Flow Based Environmental Monitoring for Smart Cities."
[2] Dominique Dom Guinard and Vlad Trifa, “Building the Web of Things With examples in Node.js and Raspberry Pi”
[3] Zanella, Andrea, et al. "Internet of things for smart cities." IEEE Internet of Things journal 1.1 (2014): 22-32.
[4] Perles, Angel, et al. "An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage." Future Generation Computer Systems (2017).
[5] Davoli, Luca, et al. "Integration of Wi-Fi mobile nodes in a Web of Things Testbed." ICT Express 2.3 (2016): 96-99.
[6] Grgić, Krešimir, Ivan Špeh, and Ivan Heđi. "A web-based IoT solution for monitoring data using MQTT protocol." Smart Systems and Technologies (SST), International Conference on. IEEE, 2016.

[7] Basto, Camilo, Luca Pelà, and Rolando Chacón. "Open-source digital technologies for low-cost monitoring of historical constructions." Journal of Cultural Heritage 25 (2017): 31-40.
[8] Douzis, Konstantinos, et al. "Modular and generic IoT management on the cloud." Future Generation Computer Systems (2016).
[9] Ferdoush, Sheikh, and Xinrong Li. "Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications." Procedia Computer Science 34 (2014): 103-110..
[10] Perles, Angel, et al. "An energy-efficient internet of things (IoT) architecture for preventive conservation of cultural heritage." Future Generation Computer Systems (2017).
[11] Davoli, Luca, et al. "Integration of Wi-Fi mobile nodes in a Web of Things Testbed." ICT Express 2.3 (2016): 96-99. .
[12] Nagib, Ahmad M., and Haitham S. Hamza. "SIGHTED: A Framework for Semantic Integration of Heterogeneous Sensor Data on the Internet of Things." Procedia Computer Science 83 (2016): 529-536.
[13] Pham, Thanh Nam, et al. "A cloud-based smart-parking system based on Internet-of-Things technologies." IEEE Access 3 (2015): 1581-1591.
[14] Kang, Do-Hun, et al. "Room Temperature Control and Fire Alarm/Suppression IoT Service Using MQTT on AWS." Platform Technology and Service (PlatCon), 2017 International Conference on. IEEE, 2017.
[15] Ray, Partha Pratim. "A survey of IOT cloud platforms." Future Computing and Informatics Journal 1.1-2 (2016): 35-46.
[16] Sruthy, S., and Sudhish N. George. "WiFi enabled home security surveillance system using Raspberry Pi and IoT module." Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2017 IEEE International Conference on. IEEE, 2017.