EFFICIENT FIRE DETECTION SYSTEM USING FOG COMPUTING

Prepared by

Rushabh Shah (16IT126)
Rahul Tiwari(16IT139)

Under the supervision of

Mr. Ravi Patel

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology

in Information Technology

IT345 Software Group Project-V (5th sem)

Submitted at

DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology

At: Changa, Dist: Anand – 388421

November 2018

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**efficient fire detection system using fog computing**" is a bonafied work carried out by **Mr. Rushabh Shah (16IT126)** and **Mr. Rahul Tiwari(16IT126)** under the guidance and supervision of **Mr. Ravi Patel** for the subject **Software Group Project-II (IT345)** of **5th** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidates, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Mr. Ravi Patel
Associate Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S. Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421, Gujarat.

ABSTRACT

Efficient fire detection system using fog computing helps the device to take fast decision faster than any cloud device.In this Project we have use Rasberry pi as a FOG device. We are also going to compare time difference between Cloud and FOG device.In this project we have use AWS as a cloud device.When we have fire at flame sensor at that time we can receive acknowledgement more faster than cloude in our Fogg device and buzzer is ringing.We have also put LED light to see output if buzzer not work

ACKNOWLEDGEMENT

"**Efficient fire detection system using fog computing**" has been unique experience for us. We would like to thanks the entire staff of the Information Technology Dept. of Chandu Bhai S. Patel Institute of Technology College who kept helping us continuously throughout our project. A special thanks to Mr. Pritesh Prajapati for his valuable guidance, cooperation and encouragement. Without his help and guidance this study would have been unsuccessful.

With sincere regards,

Rushabh Shah(16IT126)
Rahul Tiwari(16IT139)

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 PROJECT SUMMARY

First of all we connect Buzzer,Flame sensor and wires to raspberry pi.Then we have connect our pi to Laptop for communication between pi and AWS .When we run our programme and Fire at sensor .Aws and our Fog device can receive the sence.After that our Fog device reply and than after we receive reply from our aws cloud.And we can see difference between Fog device and Aws cloud device

1.2 SCOPE

By utilizing the right set of tools, developers can seamlessly develop fog applications and deploy them whenever needed As it is an android application, it is very handy to manage expense in one's own phone. Fog applications drive the machine to function in a way according to customers need.

1.3 OBJECTIVE

- To develop this project that is fast and easy to be used for its purpose.
- Create a program useful to others and handy as well.
- To increase and enhance knowledge in Rasberry pi,AWS.

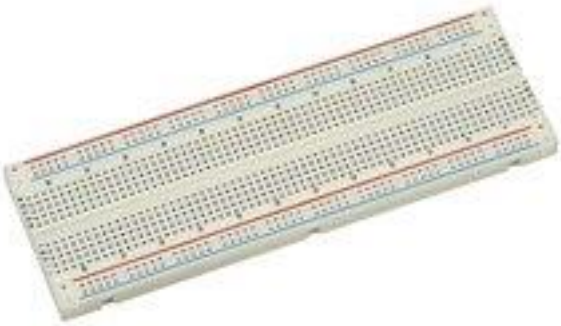
CHAPTER 2: SYSTEM ANALYSIS

2.1 TOOLS AND TECHNOLOGY

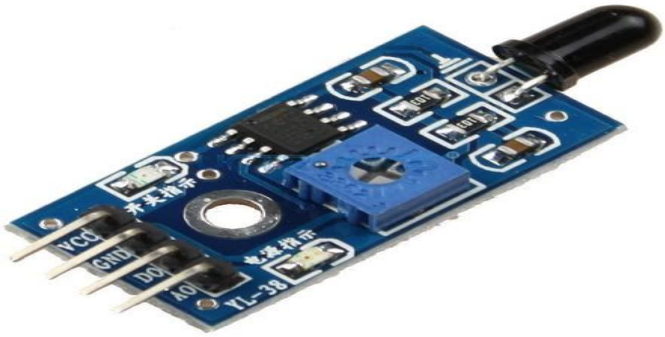
2.1.1 Hardware (Minimum):

- Raspberry pi 3 b+ RAM: 3 GB

- Bread Board



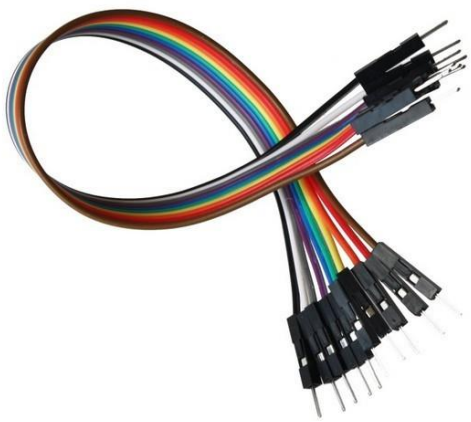
- Flame sensor



- Buzzers

- LEDs

- Jumper wires



2.1.2 Software:

- Win32DiskImager 3
- Angry IP Scanner
- MobaXtern
- AWS
- AWS IOT Python SDK for MQTT
- Python

CHAPTER 3. SYSTEM DESIGN

3.1 FLOW OF SYSTEM

- First of all we connect Buzzer,Flame sensor and wires to raspberry pi.
- Then we have connect our pi to laptop.
- When we run our programme and Fire at sensor .
- Aws and our Fog device can receive the sence.
- After that our Fog device reply and than after we receive reply from our aws cloud.
- And we can see difference between Fog device and Aws cloud device

3.2 MAJOR FUNCTIONALITY

Major Functionality of this project is that it is easy to use. And in future if home automation is successfully done than this technology is useful .

CHAPTER 4. IMPLEMENTATION

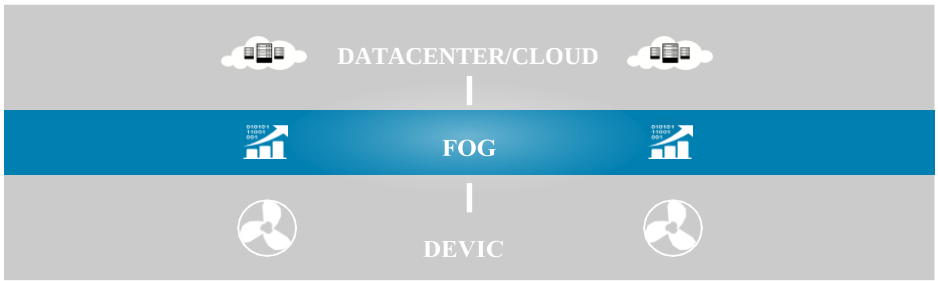
4.1 IMPLEMENTATION ENVIRONMENT

FOG COMPUTING

- The fog extends the cloud to be closer to the things that produce and act on IoT data (Figure 2). These devices, called fog nodes, can be deployed anywhere with a network connection: on a factory floor, on top of a power pole, alongside a railway track, in a vehicle, or on an oil rig. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and video surveillance cameras.

- IDC estimates that the amount of data analyzed on devices that are physically close to the Internet of Things is approaching 40 percent.1 There is good reason: analyzing IoT data close to where it is collected minimizes latency. It offloads gigabytes of network traffic from the core network, and it keeps sensitive data inside the network.

- Analyzing IoT data close to where it is collected minimizes latency. It offloads gigabytes of network traffic from the core network. And it keeps sensitive data inside the network.



4.2 CODING

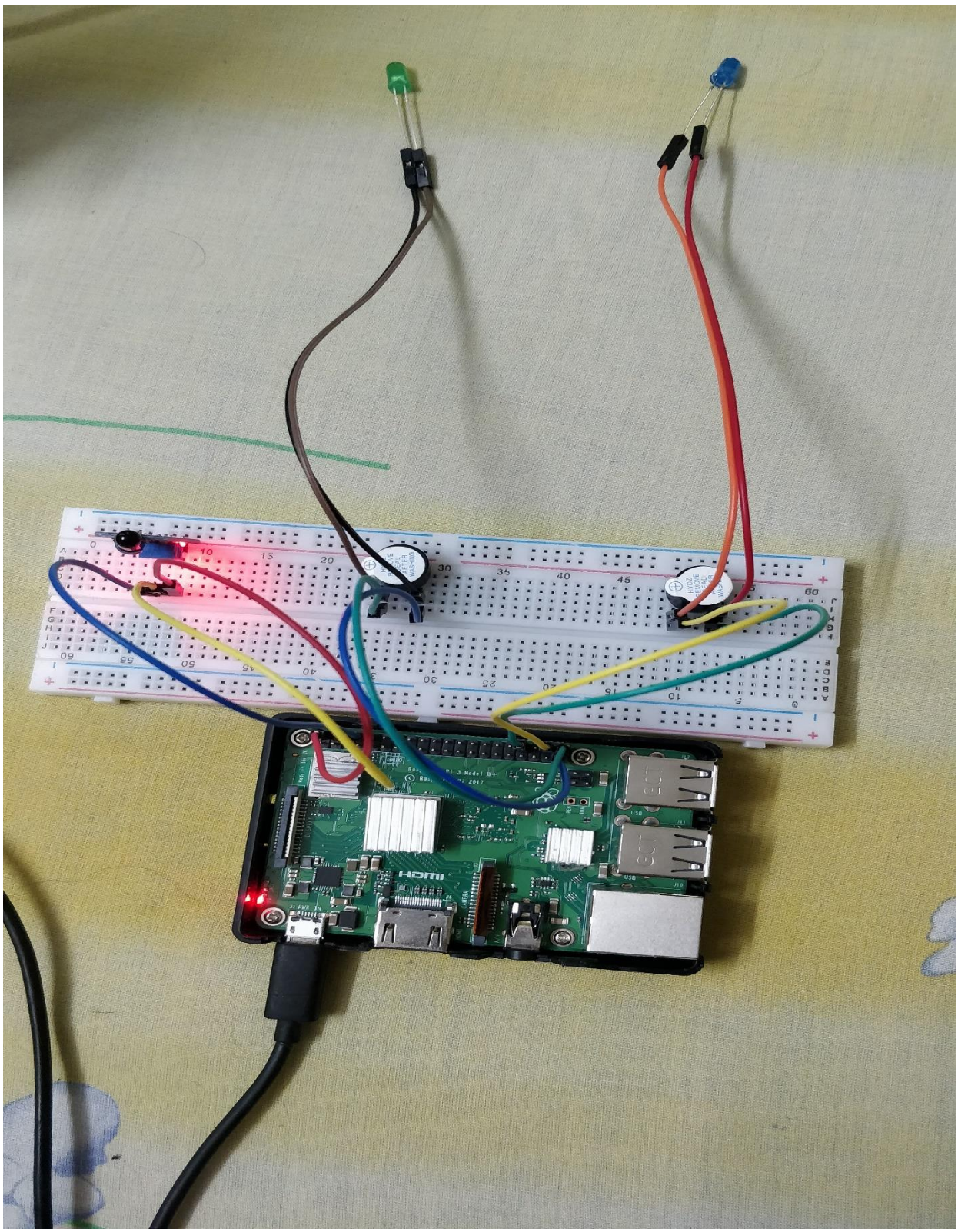
For FOG device

```
import RPi.GPIO as GPIO
import time

sensor=8
buzzer=40

GPIO.setwarnings(False)
GPIO.cleanup()

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sensor,GPIO.IN)
GPIO.setup(buzzer,GPIO.OUT)

while True:
    if(GPIO.input(sensor)==0):
        print('Flame Detected')
        GPIO.output(buzzer,True)
        time.sleep(1)

    if(GPIO.input(sensor)==1):
        #print('No Flame. Safe.')
        GPIO.output(buzzer,False)
        time.sleep(1)
```

For AWS

```
import RPi.GPIO as GPIO
import time
import sys
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
import ssl

sensor=8
buzzer=38

GPIO.setwarnings(False)
GPIO.cleanup()

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sensor,GPIO.IN)
GPIO.setup(buzzer,GPIO.OUT)

def alarm(client,userdata,message):
    GPIO.output(buzzer,True)

# AWS IoT certificate based connection
myMQTTClient = AWSIoTMQTTClient("123afhlss456")
myMQTTClient.configureEndpoint("a3dzfhkq0a3rhd.iot.us-west-
2.amazonaws.com", 8883)
myMQTTClient.configureCredentials("/home/pi/Desktop/cert/CA.crt",
"/home/pi/Desktop/cert/b0518d8a67-private.pem.key",
"/home/pi/Desktop/cert/b0518d8a67-certificate.pem.crt")
myMQTTClient.configureOfflinePublishQueueing(-1)  # Infinite offline
Publish queueing
myMQTTClient.configureDrainingFrequency(2)  # Draining: 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10)  # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5)  # 5 sec

#connect and publish
myMQTTClient.connect()
myMQTTClient.publish("rpi/connect", "connected", 0)

while True:
    GPIO.output(buzzer,False)
    if(GPIO.input(sensor)==0):
        payload='{"sensor":"fire"}'
        print(myMQTTClient.publish("rpi/fire", payload, 0))
        myMQTTClient.subscribe("rpi/fire",1,alarm)
        time.sleep(1)
```

4.3 SNAPSHOTS OF PROJECT



Chapter 5 Constraints and Future Enhancement

CONSTRAINTS

- Currently user can only for Fire detection system.
- Possible to implement for whole home automation.

BENEFITS OF FOG COMPUTING

Extending the cloud closer to the things that generate and act on data benefits the business in the following ways:

- Greater business agility: With the right tools, developers can quickly develop fog applications and deploy them where needed. Machine manufacturers can offer MaaS to their customers. Fog applications program the machine to operate in the way each customer needs.
- Better security: Protect your fog nodes using the same policy, controls, and procedures you use in other parts of your IT environment. Use the same physical security and cybersecurity solutions.
- Deeper insights, with privacy control: Analyze sensitive data locally instead of sending it to the cloud for analysis. Your IT team can monitor and control the devices that collect, analyze, and store data.
- Lower operating expense: Conserve network bandwidth by processing selected data locally instead of sending it to the cloud for analysis.

CHAPTER 6. CONCLUSION

Throughout this project we have created Fire detection system and learn how to implement AWS with Rasberri pii. We can determine and understand the continuous flow of program even in this kind of project. Using this application, we developed a real-world application which is useful to a large range of users. This project can be used for the number of users where they want fast decision like home automation,Fire alarm… etc.

REFERENCES

1. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf
2. https://docs.aws.amazon.com/iot/latest/developerguide/configure-iot.html
3. http://techblog.calvinboey.com/raspberrypi-aws-iot-python/