IBM Project Build-A-Thon
On
Smart Security System For Homes

Submitted By: Ankur Acharjee
CSE(Hons.) IBM – Information Security
Chandigarh University
Semester: 3rd

1. Introduction

A smart home security system typically connects controlled devices to a central hub or "gateway". The user interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface that may also be accessible off-site through the Internet.

While there are many competing vendors, there are increasing efforts towards open source systems. However, there are issues with the current state of home automation including a lack of standardized security measures and deprecation of older devices without backwards compatibility.

Smart Home has high potential for sharing data between family members or trusted individuals for personal security and could lead to energy saving measures with a positive environmental impact in the future.

‘Smart cities’ is a term used to describe the use of smart technologies and data as the means to solve cities’ sustainability challenges. Many cities are in the process of making themselves smart, using data and technology to improve transport, energy use, health and air quality or to drive economic growth. Others are being built to be smart from the start. So this is a term that relates to the present and to the future.

1.1 Overview

My proposed solution involves using a large number of sensors that determine the situation and environment of the house and makes adjustments to the environment control system accordingly. We also employ the usage of a simple app that gives a warning when gas and smoke levels reach dangerous values, gives basic home temperature and humidity levels and a cam feed from the main door security system camera.

1.2 Purpose

This proposed system aims to help IOT enabled devices in our homes achieve efficiency and remove the need for user inputs while still being in complete control of its user.

2. Literature Survey

2.1 Existing Problem

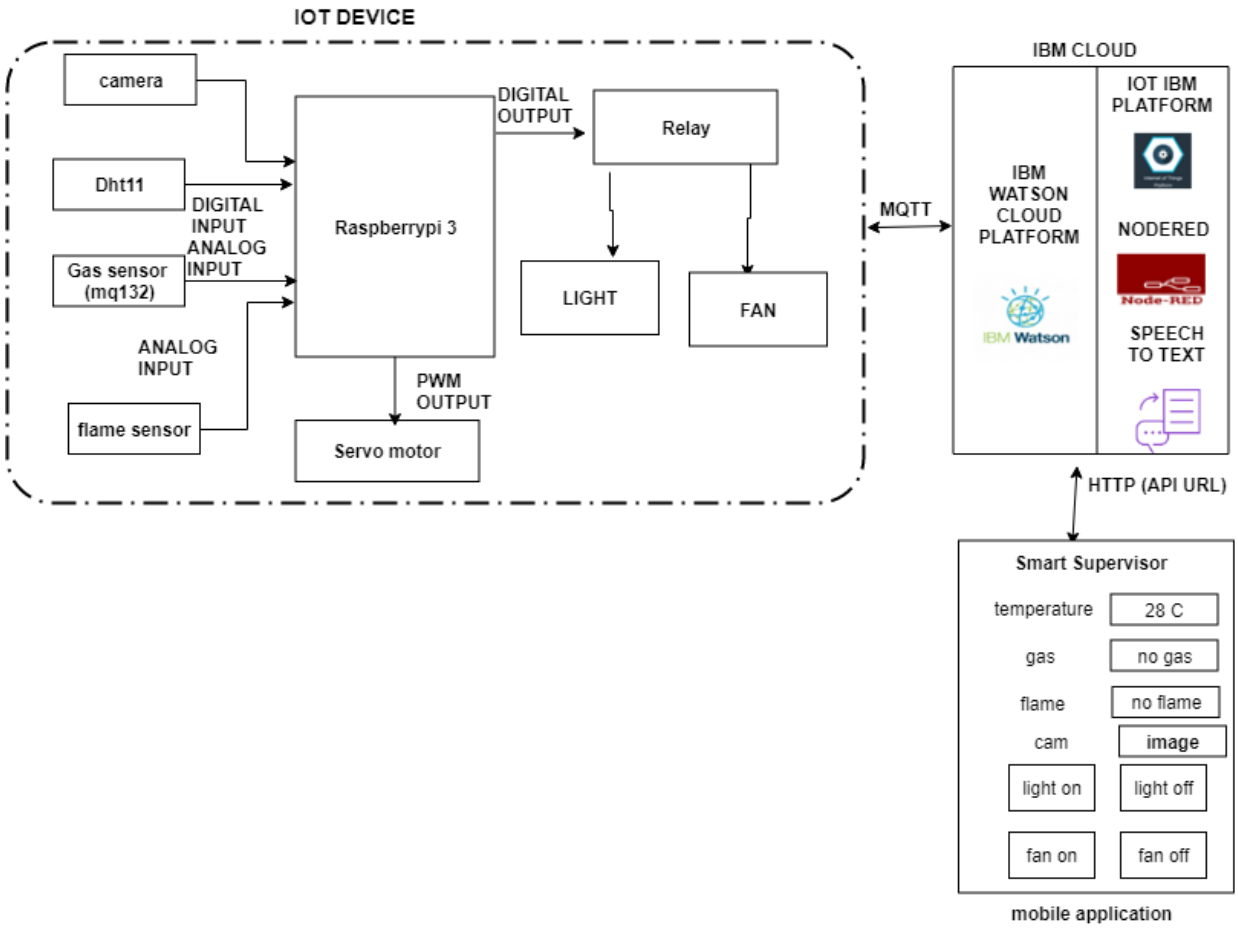
The main problem arises from the need for constant user intervention to make changes to the devices in order to get the desired output from the devices that control the environment or any service in their houses. The user might find these tasks tedious and hence we need to eliminate this.

2.2 Proposed Solution

This problem can be resolved by employing automation through IOT enabled smart homes. The IOT system finds the suitable results from the user once and then makes automatic decisions based on inputs from the sensors placed around the house and sends those decisions in the form on instructions to the environment control system/ electrical devices.

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware / Software designing

Hardware:

- Raspberry Pie 3
- Picamera
- Dht11
- Mq132
- Flame sensor
- ESP8266 Board
- Servo Motor

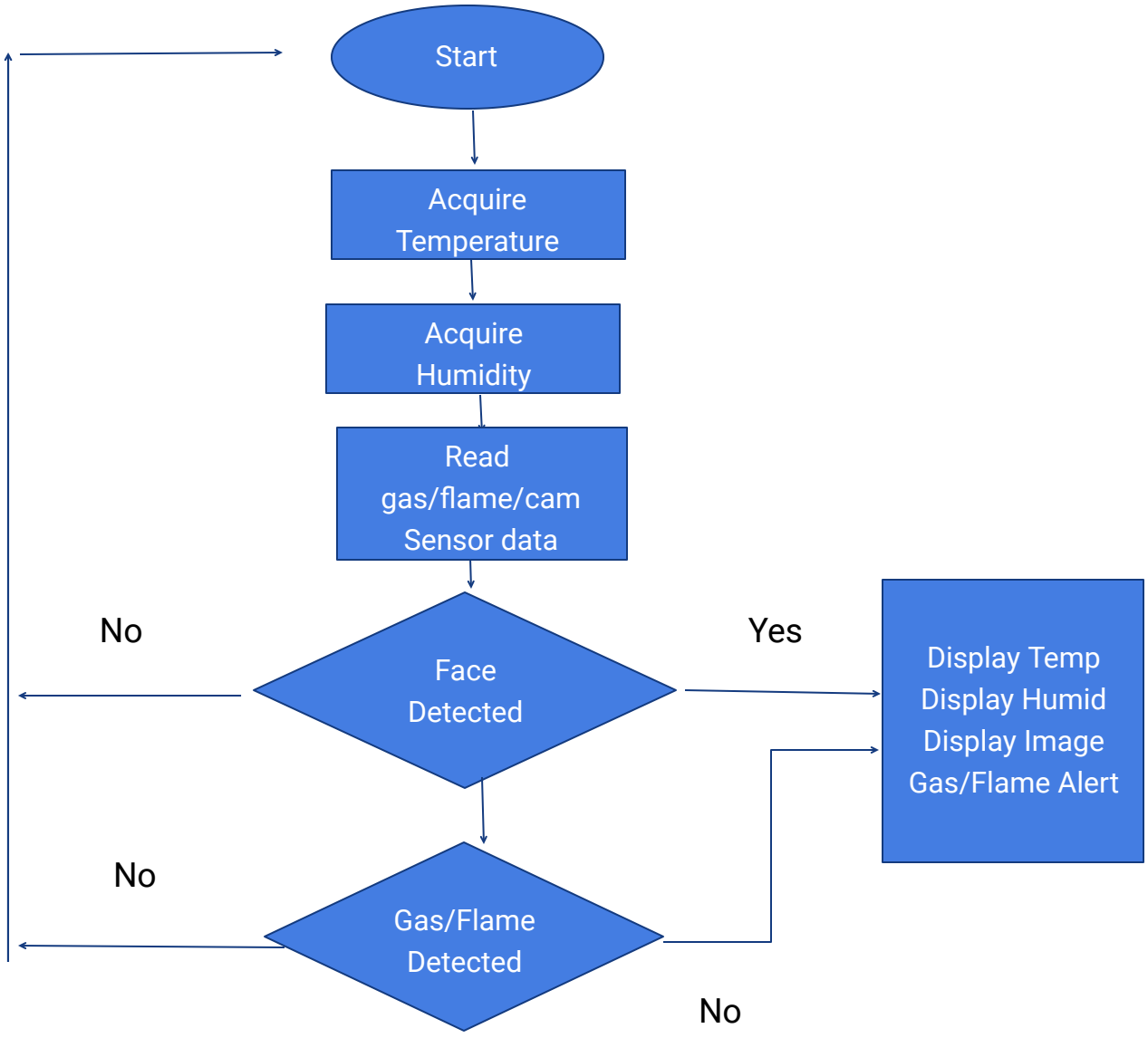
Software:

- Python 3
- MIT App Inventor
- Genymotion

4. Experimental Investigation

Due to not having an Raspberry Pi 3 board or any of its components I got the chance to experiment with various emulators. I noticed that almost all do not have any method to add the ESP8266 board that was needed for my project. So, I've written a Python 3 script which would be inputting the sensor's data directly to the IBM Cloud. Furthermore, the data will also be used to interact with Node-RED & MIT App Inventor.

5. Flowchart



6. Result

The result is the complete automation of the temperature control system along with slight automation in the security system of the house. The warning/ alerting system is also automated to some extent.

7. Advantages & Disadvantages

Advantages:

- Managing all of your home devices from one place.
- Flexibility for new devices and appliances.
- Maximizing home security.
- Remote control of home functions.

Disadvantages:

- Significant installation costs.
- Reliable internet connection is crucial.
- Security issues.
- You may lock yourself out of your own house.
- Helplessness if technology fails.

8. Applications

The sensors placed around the house constantly sends the data over to the IBM Cloud & then they're sent to Node-RED and ultimately to the app created by MIT App Inventor. The user is able to see the readings of various parameters on the app including alert warnings.

9. Conclusion

The Smart Security System For Homes using Internet of Things has been experimentally proven to work satisfactorily by connecting simple appliances to it and the appliances were successfully controlled remotely through Internet. The designed system not only monitors the sensor data, like temperature, humidity, gas, flame, but also actuates a process according to the requirement, for example switching on the light when pressed in the app. It also stores the sensor parameters in the cloud (IBM

Cloud) in a timely manner. This will help the user to analyze the condition of various parameters in the home anytime anywhere.

10. Future Scope

Home of the future is a space for the digital natives. With the invention of lots of automation technologies featuring IoT and AI, home automation has become a reality. One can implement several of their tasks with just a single command of verbal instructions. These technologies can be used to build a fully functional home automation system and control smart home devices including smart lights, connected thermostats, and appliances.

There are several new technologies which can become a part of the home in the near future:

- Increased efficiency, control, and customization: Artificial intelligence is set to make you lazy in the near future. Technology will become much more efficient and one will be able to control everything from volume to security from one central place. The devices will work automatically and you don't need to waste your energy it will act upon user's preferences. AI would revolutionize home by automatic threat detection and proactive alertness.
- Integration of Smart home devices: One can command it to control small things of the home through voice and Smartphones. All the tech giants are working in the field of IoT to bring advancements in the home automation devices. In the near future, homes will be equipped with such IoT devices which will make your daily lives work faster smoother and more accurate.
  Mark Zuckerberg came up with a goofy proof-of-concept video showing off an idealized version of how his Jarvis system actually works. Google Home, which is Google's smart speaker loaded with Google Assistant, was updated at last year's Google I/O with a bunch of new features, including "proactive assistance", also known as push notifications, hands-free free calling, Spotify, SoundCloud, and Deezer

integrations, and more. Also, more recently, Google launched two more Google Home speakers, Home Max and Home Mini.

- Smart spaces outside homes: Smart parking through sensors will help to recognize whether the parking is available or not. Camera monitoring can be done and with the help of artificial intelligence and computer vision, both parking facilities and security can be provided. It would be a faster and smoother process and act as a reference for other smart systems to be build accordingly. Streetlights can also be automated through sensors and build for effective use for the people nearby.
- Development of smart appliances: The devices which we use to use like television, refrigerator and even the mirror is getting smarter today with the evolution of technology. The smart mirror should not only act as a face video but also help to other tasks like listening to music and stuff. Televisions have become part of a centralized entertainment and can also be used for social media. The refrigerator has been upgraded to sense the temperature outside and operate accordingly. The washing machine will wash the clothes according to the clothes material and switch off after drying. They will keep on advancing as the technology evolves.
- Personal home delivery: Drones will be used to deliver the packages at the right time. They will replace the normal salesman job. They might also be used for several other tasks like monitoring the weather outside the home, returning something back to a relative's home nearby and so on. They can also be used for monitoring the traffic in our locality.

One can build several amazing projects using the concepts of home automation. There are several projects already done by developers and available on the Internet. They might help you to start work with IoT. You can add new skills to own a smart device. You can make your smart home device work according to your life works and habits. Even we can build

many projects around it by discovering new areas of the internet of things and make the world a smarter place to live in.

11. Bibliography

- http://www.t3.com/features/the-smart-home-guide
- https://web.archive.org/web/20160505124414/https://www.reuters.com/article/research-and-markets-idUSnBw195490a%2B100%2BBSW20150119
- https://archive.org/details/homeautomationwi0000gerh

Appendix

```
# -*- coding: utf-8 -*-

import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import cv2
import numpy as np
import sys
import ibmiotf.application
import ibmiotf.device
import random
import time

from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
#Provide your IBM Watson Device Credentials
organization = "g7khju"
deviceType = "rasppi"
deviceId = "101"
authMethod = "token"
```

```
authToken = "Ynmw-&Ub@FJyP-N*jG"
#API token = nI?4p&RNgF80yA_A@&

def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)
        print(cmd.data['command'])

        if(cmd.data['command']=="dooropen"):
                print("door open")

        if(cmd.data['command']=="doorclose"):
                print("door close")



        if(cmd.data['command']=="lighton"):
                print("light on")

        if(cmd.data['command']=="lightoff"):
                print("light off")

        if(cmd.data['command']=="fanon"):
                print("fan on")
        if(cmd.data['command']=="fanoff"):
                print("fan off")

try:
          deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
          deviceCli = ibmiotf.device.Client(deviceOptions)
          #.....................................

except Exception as e:
```

```
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as
an event of type "greeting" 10 times
deviceCli.connect()
face_classifier=cv2.CascadeClassifier("haar-face.xml")

#It will read the first frame/image of the video
video=cv2.VideoCapture(0)



COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
# Current list avaiable at
https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "1SrM9cn9KI4cuavhgARdYJlS-FpaLmip3uwlX6Gi3Qgy"
# eg "W00YiRnLW4a3fTjMB-oiB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN =
"crn:v1:bluemix:public:cloud-object-storage:global:a/74e549a90bfb4d9094
5dd193dd7b6f67:510c2d9f-eb9c-4df7-87e2-ccc2dab11423::"



client = Cloudant("c0fecf4c-33b6-4643-a0e8-317729a00e34-bluemix",
"7af9cd73b462f1103340e1106ca52c362506db632a492544eade875f47b9
9e08",
url="https://c0fecf4c-33b6-4643-a0e8-317729a00e34-bluemix:7af9cd73b4
62f1103340e1106ca52c362506db632a492544eade875f47b99e08@c0fecf
4c-33b6-4643-a0e8-317729a00e34-bluemix.cloudantnosqldb.appdomain.cl
oud")
client.connect()
```

```
database_name = "imagesdb"

# Create resource
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_RESOURCE_CRN,
    ibm_auth_endpoint=COS_AUTH_ENDPOINT,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)



def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,
bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part
upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
```

```
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))



while True:
    #capture the first frame
    check,frame=video.read()
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #detect the faces from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)

    #drawing rectangle boundaries for the detected face
    for(x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        picname=picname+".jpg"
        pic=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname,frame)

        my_database = client.create_database(database_name)
```

```
multi_part_upload("cloud-object-storage-wo-standard",picname,pic+".jpg")
        if my_database.exists():
            print("'{database_name}' successfully created.")
            json_document = {
                    "_id": pic,

"link":COS_ENDPOINT+"/cloud-object-storage-wo-standard/"+picname
                    }
            new_document = my_database.create_document(json_document)
            if new_document.exists():
                print("Document '{new_document}' successfully created.")
        time.sleep(1)
        t=str(random.randint(-100.0,100.0)) + "°C"
        h=str(random.randint(0,100)) + "%"
        g=random.choice(['Gas', 'No Gas'])
        f=random.choice(['Flame', 'No Flame'])
        data = {"d":{ 'temperature' : t, 'humidity': h, 'gas' : g, 'flame' : f}}
        #print data
        def myOnPublishCallback():
            print ("Published data to IBM Watson")

        success = deviceCli.publishEvent("Data", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback

    #waitKey(1)- for every 1 millisecond new frame will be captured
        Key=cv2.waitKey(1)
        if Key==ord('q'):
        #release the camera
```

```
            video.release()
        #destroy all windows
            cv2.destroyAllWindows()
            break
deviceCli.disconnect()
```