Smart Security System for Homes

BY: Suhail Ahmed

CONTENTS

1.INTRODUCTION

Now a days the world has switched over to Industry 4.0 with the inclusion of a very efficient way of dealing with data sharing and interlinking of devices. This technology is called the Internet of Things. We have tried to make our lives as comfortable as possible by employing machines to do tedious tasks or repetitive ones. It is no surprise that our homes (where we spend most of our time) have been fitted with a large number of such devices. They make our lives comfortable but still don’t reach their true potential because of constant inputs from users (the only inefficiency). IOT aims to eradicate this by employing sensors that take the inputs and “send” this data in suitable forms to the concerned devices and hence remove the need for constant human interaction or otherwise known as automation of machines. Only then, is true efficiency achieved.

1.1 Overview

Our 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.

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

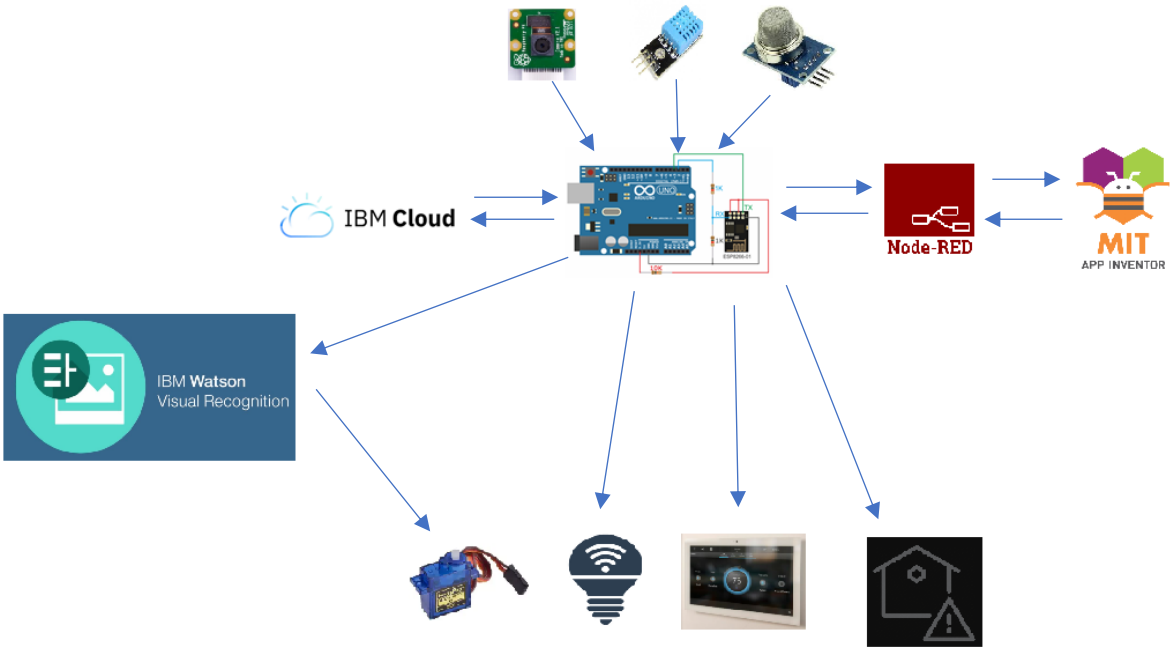
The main problem arises from the need for constant user intervention to make changes to the devices in order to get the desired output from the devices that control the environment or any service in their houses. The user might find these tasks tedious and hence we need to eliminate this.

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 Requirements

1. MIT AI2 companion App

2.Arduino Uno

3.ESP8266 Board

4.Connecting wires

6.Picam

7.DHT11 sensor

8.MQ-2 Gas sensor

9.Servo motor

10.IOT enabled Light bulb/ LED

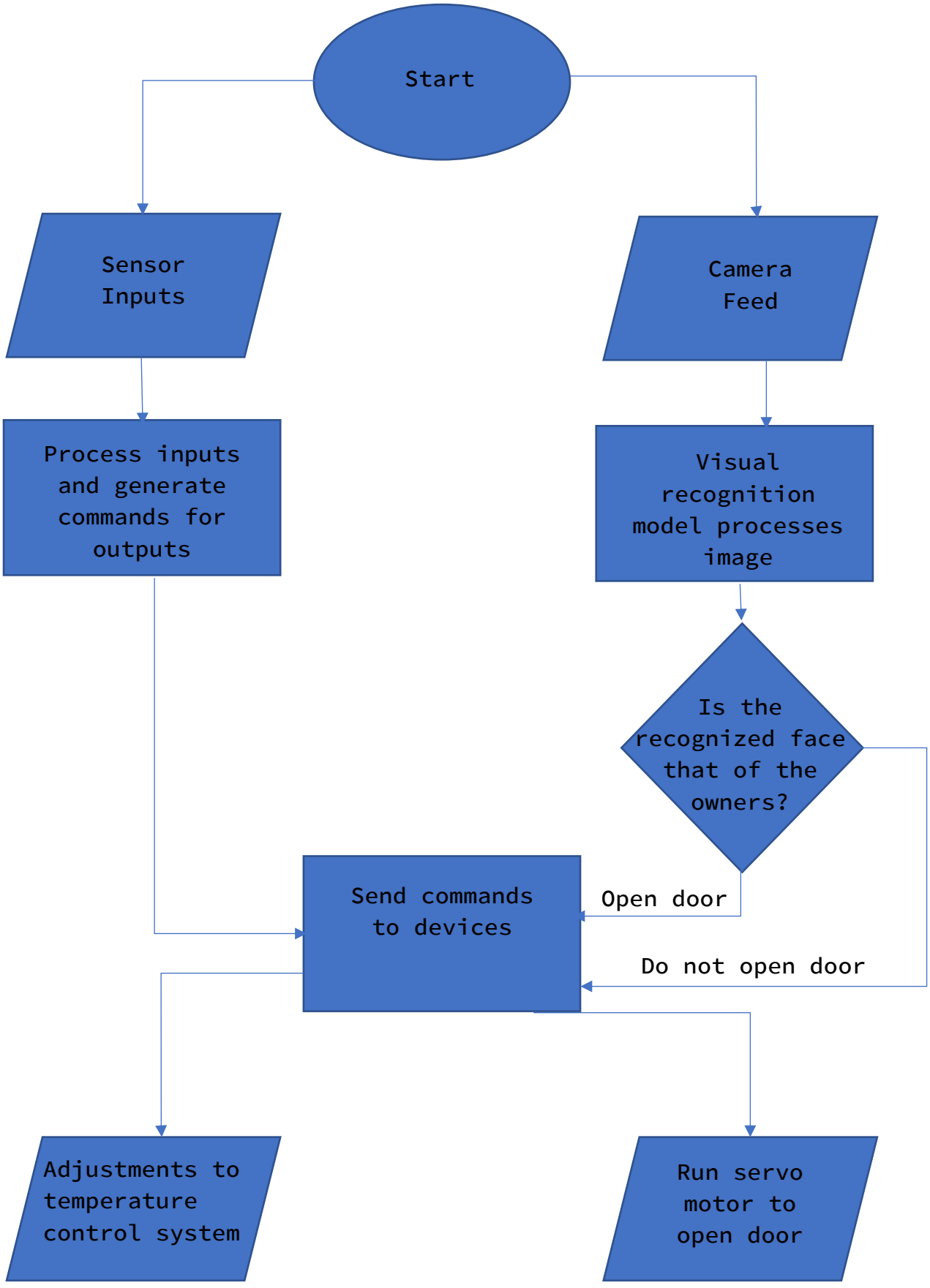
11.IOT enabled temperature control system

12.Android device

4.EXPERIMENTAL INVESTIGATION

Due to not having an Arduino 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. Apart from this I took some surveys on how interested or comfortable they would be with an IOT enabled home with the participants being some members of my family. All 7 of the participants were comfortable of the idea. One of them was conforming as long as the final overriding control still stayed with them. 5 of the 7 thought it would be worth investing in such a system for their own home. The rest thought that thought the idea was great the cost wasn’t worth it. Overall, the general idea of having an automated house was liked by all. But the cost of such a system caused them to change their decision of actually using it.

5.FLOWCHART


Start
Sensor Inputs
Camera Feed
Process inputs and generate commands for outputs
Visual recognition model processes image
Is the recognized face that of the owners?
Send commands to devices
Open door
Do not open door
Adjustments to temperature control system
Run servo motor to open door

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 include the complete automation of the house thus reducing the need for human intervention and hence easing the work to be done by the user.

Disadvantages include high initial costs and some security risks.

8.APPLICATION

The sensors placed around the house constantly send the inputs over to the Arduino board from where they are sent to Node Red and ultimately to the MIT AI2 companion app.

The user is able to see the readings of various parameters on the app.

9.CONCLUSION

The system helps in home automation and is a more efficient way to deal with home control systems.

10.FUTURE SCOPE

The system as of now only displays the sensor readings but it can be upgraded to also control the various devices based on the inputs it receives and the user preferences.

A more secure mode of information interchange can be used to ensure security of data and hence the house.

11.BIBLIOGRAPHY

11.1 An Innovative Heuristic Algorithm for IoT-Enabled Smart Homes for Developing Countries
11.2 OpenCV Documentation
11.3 ESP8266 Documentation

APPENDIX

```
#include<ESP8266WiFi.h>

#include<PubSubClient.h>

#include "DHT.h"

#define DHTPIN D2

#define DHTTYPE DHT11

#define MQ2pin (0)

DHT dht (DHTPIN, DHTTYPE);

float t,s;

int h;

String command;

String data="";

void callback(char* topic, byte* payload, unsigned int
payloadLength);

const char* ssid = "Wifi";

const char* password = "very101Secure";

#define ORG "2twfpy"

#define DEVICE_TYPE "MyDevice"

#define DEVICE_ID "HomeIOT"

#define TOKEN "&j6U@W80ipoFTBOuSS"

#define led1 D0

const char publishTopic[] = "iot-2/evt/Data/fmt/json";

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_Type ":" DEVICE_ID;

WiFiClient wifiClient;
```

```
PubSubClient client(server, 1883, callback, wifiClient);
int publishInterval = 5000;
long lastPublishMillis;
void publishData();

void setup(){
  Serial.begin(115200);
  Serial.println();
  pinMode(led1, OUTPUT);
  dht.begin();
  wifiConnect();
  mqttConnect();
}

void loop(){
  if(millis() - lastPublishMillis > publishInterval){
    publishData();
    lastPublishMillis = millis();
  }
  if(!client.loop()){
    mqttConnect();
  }
}

void wifiConnect(){
  Serial.print("Connecting to "); Serial.print(ssid);
  WiFi.begin(ssid, password);
  while(WiFi.status() != WL_CONNECTED){
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void mqttConnect(){
  if(!client.connected()){
    Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
    while(!client.connect(clientId, authMethod, token)){
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevices(){
  if(client.subscribe(topic)){
    Serial.println("Subscribe to cmd OK");
  } else{
    Serial.println("Subscribe to cmd FAILED");
  }
}

void callback(char* topic, byte* payload, unsigned int
payloadLength){
```

```
  Serial.print("callback invoked for topic: ");
  Serial.println(topic);
  for(int i=0; i< payloadLength; i++){
    command+= (char)payload[i];
  }
  Serial.print("data: "+ command);
  control_func();
  command="";
}

void control_func(){
  if(command=="lightoff"){
    digitalWrite(led1, LOW);
    Serial.println("Lights were switched Off.");
  }
  else if(command == "lighton"){
    digitalWrite(led1,HIGH);
    Serial.println("Lights were switched On.");
  }
  else{
    Serial.println("No commands subscribed");
  }
}

void publishData(){
  h = dht.readHumidity();
  t = dht.readTemperature();
  s = analogRead(MQ2pin);
  if(isnan(h) || isnan(t)){
```

```
    Serial.println("Failed to read the DHT sensor.");
    return;
  }
  if(isnan(s)){
    Serial.println("Failed to read the MQ-2 sensor.");
    return;
  }
  String payload = "{\"d\":{\"temperature\":";
  payload += t;
  payload += ","\"humidity\":";
  payload += h;
  payload += ","\"smoke\":";
  payload += s;
  payload += "}}";
  Serial.print("\n");
  Serial.print("Sending payload: ");Serial.println(payload);
  if(client.publish(publishTopic, (char*) payload.c_str())){
    Serial.println("Publish OK");
  }
  else{
    Serial.println("Publish FAILED");
  }
}
```