A Project Report

On

SMART BUILDING WITH IBM WATSON

Submitted by

Team: The Elite

Abhigna Ogirala

Chandra Mileeni Chowdary

Movva Sai Laxmi

Katikam Vishali

Chandana Poorvaja Challa

INDEX

S. No	Description	Page. No
1	Abstract	1
2	Introduction	2
3	Problem Statement	3
4	Circuit Diagram	4
5	List of components	5
6	Components Description	6
7	Working	12
8	Appendix	14
9	NodeRed Flow	19
10	Prototype	20
11	Advantages	21
12	Disadvantages	22
13	Application	23
14	Conclusion	24
15	References	25

ABSTRACT

Electrical energy consumption is the need of the hour. In today's technological advance and comfortable lives, conserving electrical energy has become even more imperative. We therefore intend to make a step forward in the same direction by developing this project; "Smart Buildings with IBM Watson". This project as the name suggests employs the idea of automation to enable users to decrease the usage of electricity by switching off the appliances when not in need. This project will not just help reduce one's electricity bills but will also decrease the detrimental repercussions caused by gaseous and liquid pollutants that are liberated by the appliances we use. For that reason this project is aimed at creating an energy saving system that makes use of what is called as the Green building concept to convert any regular, conventional building into an energy saving one.

The model designed illustrates energy consumption through data obtained from the sensors known as the energy saving parameters, like temperature, humidity, light intensity etc. The energy saving system is built in such a way that it enables the user to operate their home appliances through three modes of operation i.e., Eco Mode, Away Mode and Manual Mode which turns the appliances on/off as and when required thereby saving electrical energy. The eco mode is a completely automatic mode which switches all lights, fans and ACs of the building when it detects the presence of humans in the building area. If there is no movement of humans detected, then automatically all the appliances are switched off without any human intervention.

The next mode is the away mode which when chosen switches off all appliances by default, independent of whether humans are there or not. The last mode is the manual mode which operates the appliances based only on the user input. Apart from these three modes, we have also integrated a DHT sensor through which weather parameters can be monitored. The LDR and PIR sensor have also been used for efficient working of the system and this entire sensor data are stored by creating a database using IBM Watson cloud services. These stored parameters can be observed and the modes can be selected through a User Interface created in Node Red that assists in controlling of appliances and consequent energy consumption.

INTRODUCTION

Internet of Things (IoT) has been a budding technology in the past few years. This miraculous advancement has therefore led to development of many useful and interesting technological projects. One of these is the concept of Smart Buildings that implements Building Automation Systems to automatically perform actions with no or less human intervention. The Smart Building project is a substantial parent of the home automation system that is already in existence. And hence the core of this energy saving system is motion detection which is also the vital part of home automation. This process of motion detection enabled through the incorporation of IoT technologies in buildings consequently assists and provides users the privilege to monitor a building all round the clock.

This kind of access is extremely valuable as, in addition to providing anytime, anywhere control it also helps track real-time occupancy during fire accidents or natural disasters. Therefore it helps in search and rescue process in situations of such distress. This Smart Building notion has also been proved to be instrumental in lower the energy consumption in a significant manner due to its ability to identify occupancy patterns. Recognising such patterns, according to recent studies is much more efficient than fixed usage and occupancy patterns.

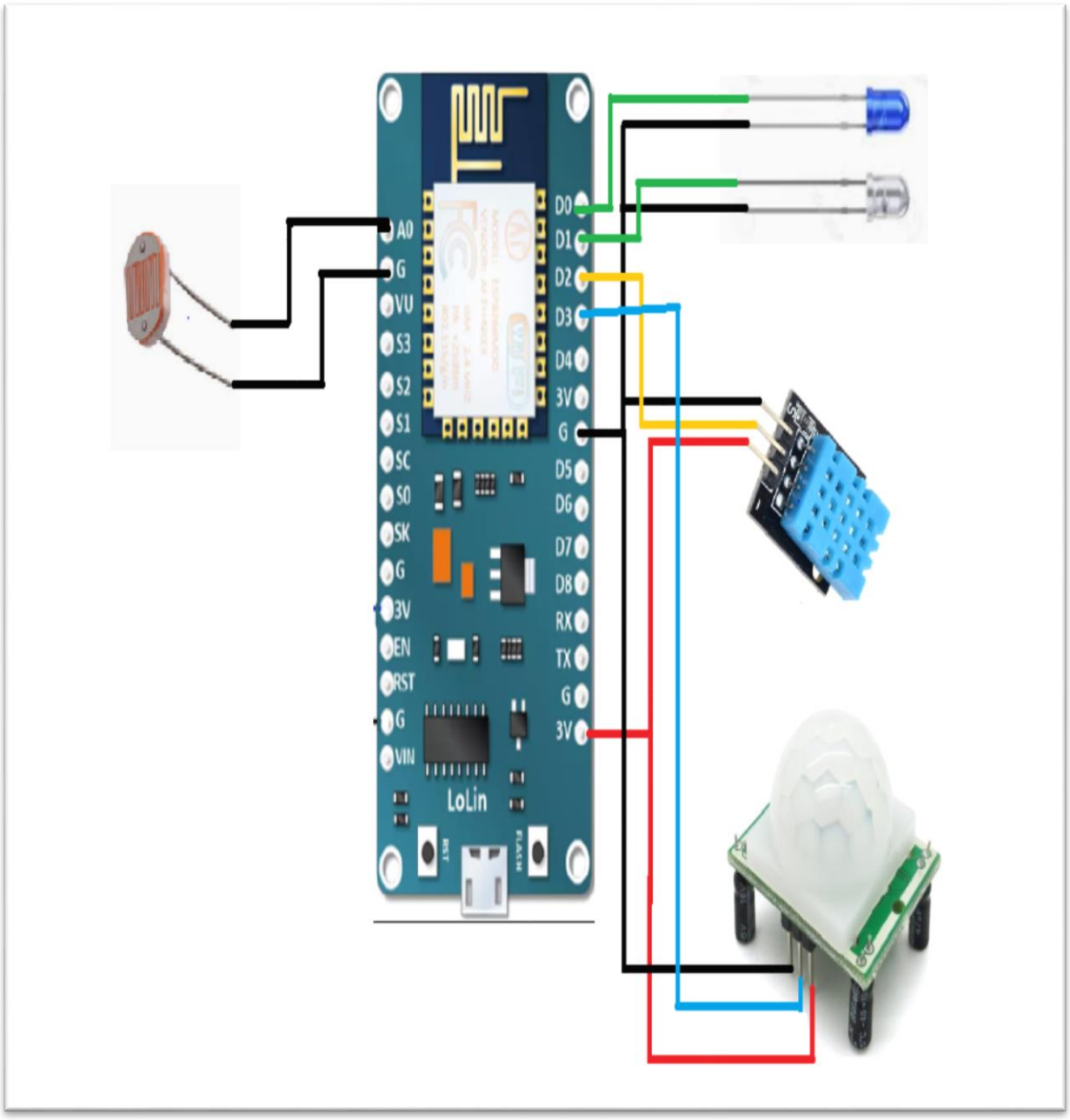
Keeping all the benefits that this system has to offer in view, we have proposed a system where a created User Interface which acts as a medium of communication between the user and the appliances. The user can not just operate the appliances but also has access to all data like temperature, humidity and all other sensor values which are also updated in a database for permanent storage of all sensor information. The application offers he user with choices between three modes of operation namely eco mode, away mode and manual mode respectively.

The designing of three such modes is to give the best possible control to the user to effectively run the appliances at his/her comfort and at the same time conserve maximum energy possible by supplying all required and necessary information of the environment in the building or room. The Smart Building project is almost like creating a virtual image of the environment while giving all information about the parameters concerned to create a real-time access to have complete control of the electrical appliances.

PROBLEM STATEMENT

The project proposes a solution to the existing tedious system of physically going till the switch board and then switching the appliances on/off. In the existing system appliances cannot be operated remotely and hence if the person forgets to turn off the appliances before leaving, it will result in wastage of huge amount of electricity. To avoid such situations this Smart Building project has been developed where the appliances work automatically based on the sensor data. This system also provides an option to make the appliances work based on user input alone thereby reducing the electricity consumption. The proposed system is achieved by devising an interface through Node Red which allows the consumer to access different modes of operation through an application designed using MIT app inventor. The sensor data is sent to IBM Cloud and stored in cloudant for further usage.

CIRCUIT DIAGRAM



LIST OF COMPONENTS

Hardware

- Node MCU
- DHT11
- LDR
- PIR sensor
- LEDs
- Jumper Wires
- Micro USB Cable

Software

- Arduino IDE
- MIT App Inventor 2
- IBM Watson

COMPONENT DESCRIPTION

Hardware

NodeMCU

NodeMCU is an open source, interactive, programmable, low cost, simple, smart, and WI-FI enabled IoT platform. It includes firmware which runs on the ESP8266 WiFi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware and development kit helps to create IOT products with Lua scripting language.

Specifications

Developer: ESP8266 Open source Community
Type : Single-board microcontroller
Operating system : XTOS
CPU : ESP8266
Memory : 128kBytes
Storage : 4MBytes
Power By : USB
Power Voltage : 3v ,5v
Code : Arduino Cpp
IDE Used : Arduino IDE
Digital I/O pins :10
Analog Input pin: 1

DHT11 Sensor

DHT is a digital humidity and temperature sensor. It consists of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate

changes due to which the resistance changes. This change in resistance is measured and processed by the IC and passed to the microcontroller.

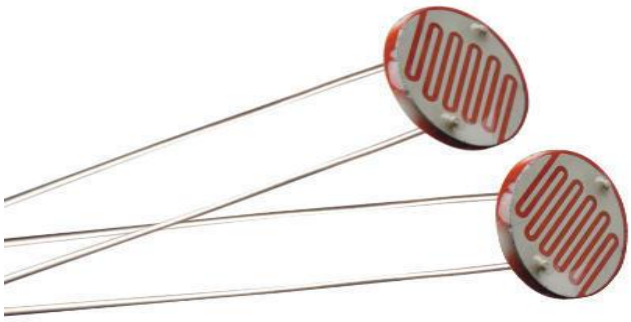
And, for measuring temperature these sensors use a NTC temperature sensor or a thermistor. Negative Temperature Coefficient means that the resistance decreases with increase in temperature. A thermistor is actually a variable resistor that changes its resistance with change of the temperature. These sensors are made up of semiconductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature.

Specifications

Operating Voltage: 3.5V to 5.5V
Operating current: 0.3mA (measuring) 60uA (standby)
Output: Serial data
Temperature Range: 0°C to 50°C.
Humidity Range: 20% to 90%
Resolution: Temperature and Humidity both are 16-bit.
Accuracy: ±1°C and ±1%

LDR

A light dependent resistor/ photoresistor are a component that is sensitive to light. When light falls on it, the resistance changes. Values of the resistance of the LDR may change over wide range of magnitudes. A LDR is made of a high resistance semiconductor. In the dark, a LDR can have a resistance as high as several MΩ, while in the light; a LDR can have a resistance as low as a few hundred ohms. If incident light on a LDR exceeds certain frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electrons conduct electricity, thereby lowering resistance.



Specifications

Resistance: 400ohm to 400Kohm
Sensitivity: about 3msec

PIR Sensor

Passive Infra Red Sensors are fundamentally made of a pyro electric sensor, which can detect infrared radiation. It is a motion detector which detects the heat (infrared) emitted naturally by humans and animals. It consists of two halves to detect the radiation and the change in condition that occurs when a target enters its field. These changes in the amount of infrared radiation on the element in turn change the voltages generated. On closely observing the top region of the sensor, the beehive structure, curved segments are seen. These curved segments are Fresnel lenses which constitute an array that increases the detection zone of the sensor. Fresnel lens array is known to capture more infrared radiation and focus it to a relatively smaller point. Detection is more stable and maximum distance for detection is also increased. Fresnel lens has been crafted to be translucent so that it can capture only infrared radiation without getting unwanted radiations from visible spectrum of light.

Specifications

Operating Voltage: 5V – 20V
Power Consumption: 65mA
TTL output: 3.3V, 0V
Delay time: Adjustable (0.3->5min)
Lock time: 0.2 sec
Sensing range: less than 120 degree, within 7 meters
Operating Temperature: – 15 ~ +70

LED

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. This effect is called electroluminescence. The colour of the light is determined by the energy required for electrons to cross the band gap of the semiconductor.

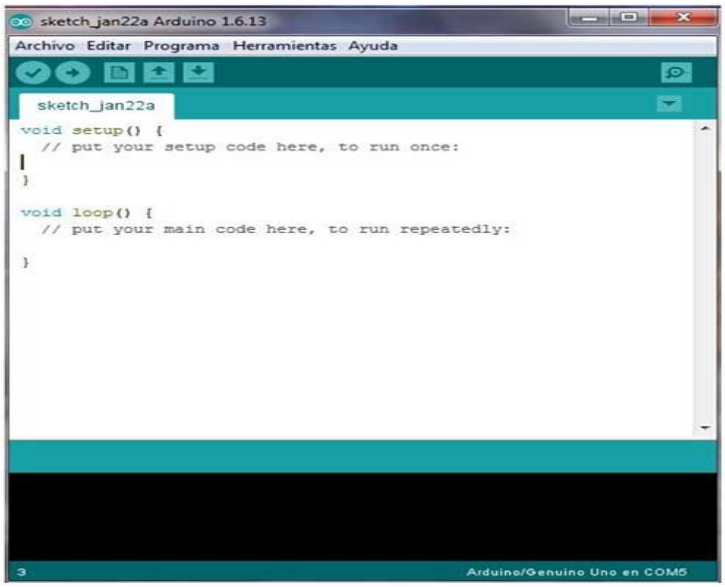
Specifications

Forward Drop: 1.8 to 2.2V
Maximum current: 20mA
Suggested using current: 16-18mA
Luminous Intensity: 150-200mcd

Software

Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application for operating systems like MAC, Windows, Linux. It runs on the Java Platform that comes with inbuilt functions and commands that helps debugging, editing and compiling the code in the environment. Arduino IDE is a open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonard etc. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts, Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.



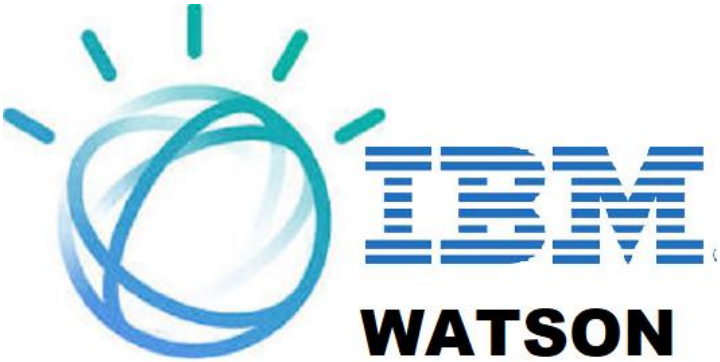
MIT App Inventor 2

MIT App Inventor is a programming environment that allows us to build fully functional apps for Smartphone and tablets. This helps in creation of complex, high-impact apps in significantly less time than traditional programming environments. The MIT App Inventor

enables users to create customized applications by providing easy access to develop software thereby motivating young students to build technologically advanced projects.

IBM Watson

IBM Watson is a platform that assists in development of projects and enterprises. It is a very reliable and trusted environment. It acts as front end for creating, managing, and viewing cloud resources. It is user friendly.



WORKING

To begin with, the most important part of an automation system is the coding which is done here in Arduino IDE (Integrated Development Environment). The programming for this energy saving system has been done so as to employ a three mode operation system to enable user to access the appliances.

The program has been split into functions according the purpose they serve in helping the network connectivity, data transmission, data retrieving and the outputs required based on the selected mode of operation. The code therefore is dumped onto the NodeMCU and the then obtained sensor data received by the microcontroller is sent to IBM Cloud.

In accordance with the written code a NodeRed flow is created which helps in create a visual interface by retrieving data from the cloud. The data obtained by the IBM IoT input node in the flow is then parsed with the help of a function node and stored in an already created cloudant (database) for further use.

The NodeRed as mentioned earlier helps in creation of visual interface through which commands can be sent to perform the required operations. The MQTT (Message Queuing Telemetry Transport) protocol is used for communication between NodeRed visual interface and the NodeMCU.

In the same NodeRed flow an URL is generated and an http request is sent from MIT to NodeRed requesting information such as humidity, temperature and LDR and PIR values which are then parsed and then passed onto the designed application. All these sensor data as mentioned earlier is taken from the cloud by NodeRed.

Integrating these, this project has been created such that it operates in three different modes providing more flexibility of operation to the end user.

The three modes in which the intended application functions are

- Echo mode
- Away mode
- Manual mode

Echo Mode

This mode when chosen, entirely automates the operation of all lights, fans and air conditioners of the building based on the presence of humans. The PIR sensor is used which sends a sensor value of one to the NodeMCU when any humans are present. This data has been programmed such that all the appliances are automatically turned on when a value of one is received from the PIR sensor. Alternatively on no presence of humans the PIR gives a value zero as output which consequently leads to switching off of all appliances without human intervention.

The eco mode also automatically dims or brightens the light intensity in room based on light intensity of the environment in room which is obtained using the LDR sensor. This sensor has been therefore integrated with LEDs to display the variation in brightness of the bulb depending on the existing natural light.

Away Mode

This mode when selected sends the command "away" from the app which is programmed by default to switch off all appliances irrespective of whether humans are present or not. That is, independent of the PIR sensor value when this command is sent from the application shuts down all connected appliances.

Manual Mode

This mode when selected, unlike the echo mode is designed so as to operate the appliances based on the user input alone. Through the application the user can select any option to operate the appliances individually as per choice and then the required commands are sent to perform necessary actions. Therefore through manual mode of operation the appliances can be operated through app using wireless network.

APPENDIX

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#define DHTPIN D2
#define DHTTYPE DHT11

DHT dht (DHTPIN, DHTTYPE);

float temperature;
int humidity;
int ldr_data;
int pir_data;
String command;
String data="";

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

//WIFI CREDENTIALS
const char* ssid = "milee";
const char* password = "mahi485ka";

//DEVICE CREDENTIALS
#define ORG "4y51d2"
#define DEVICE_TYPE "smartBuild"
#define DEVICE_ID "141414"
#define TOKEN "hzfkbw60xr66R&yj!7"

//PIN DECLARATIONS
#define LED1 D0
#define LED2 D1
#define PIR D3
#define ldr A0


const char publishTopic[] = "iot-2/evt/sensorsdata/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;
```

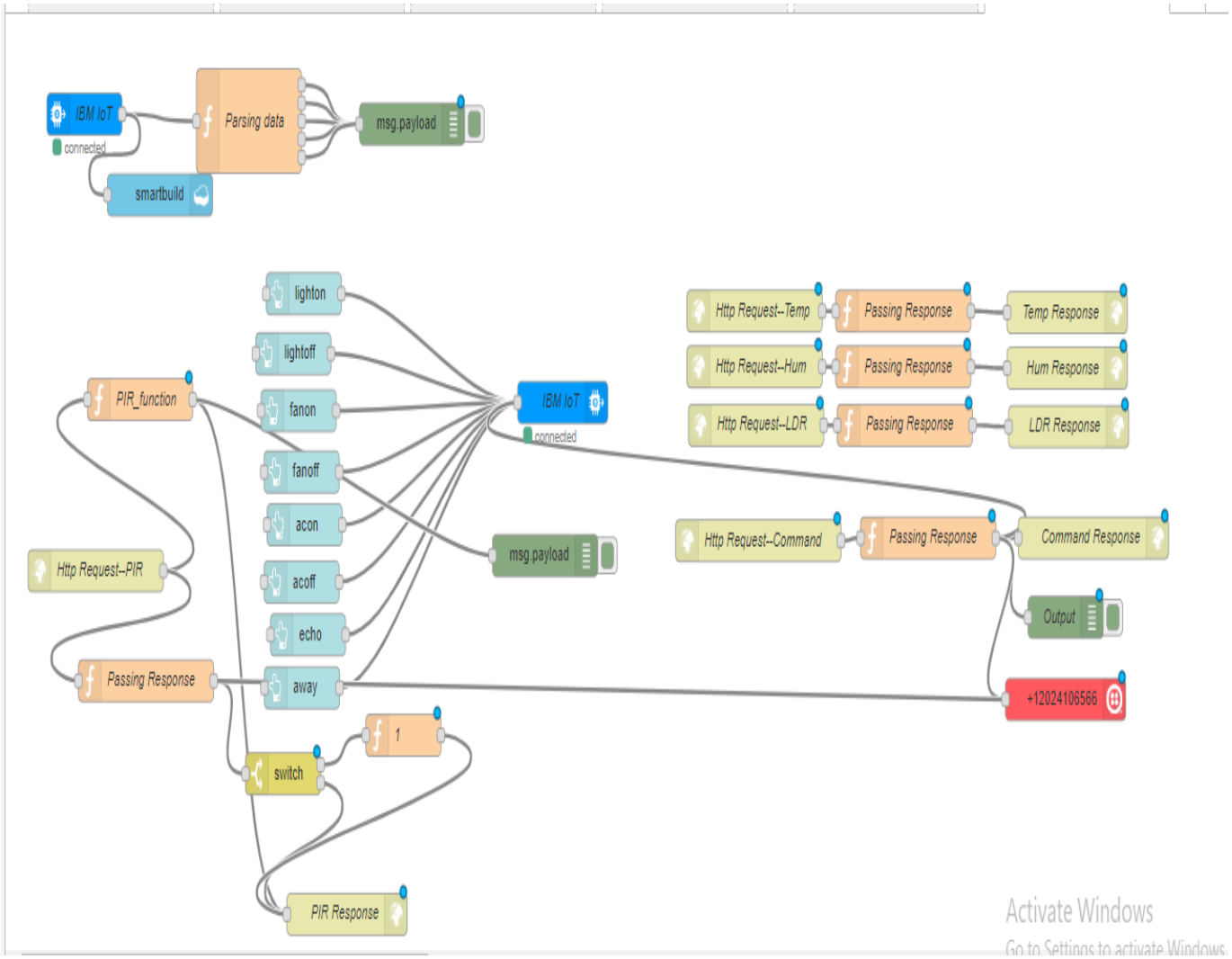
```
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback, wifiClient);

int publishInterval = 5000;
long lastPublishMillis;
void publishData();

void setup()
{
  Serial.begin(115200);//to begin serial communication
  Serial.println();
  pinMode(ldr,INPUT);//to declare whether the pins are used as I/O
  pinMode(PIR,INPUT);
  pinMode(LED1,OUTPUT);
  pinMode(LED2,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 initManagedDevice()
{
 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=="echo")
 {
  if(pir_data)
  {
```

```
    if(ldr_data>500) //based on light intensity the leds are operated
    {
      analogWrite(LED1,255);
      analogWrite(LED2,255);
    }
     else
     {
      analogWrite(LED1,20);
      analogWrite(LED2,20);
     }
  }
 else
  {
    digitalWrite(LED1,LOW);
    digitalWrite(LED2,LOW);
    Serial.println("Nobody is present");
    Serial.println("Lights are OFF");
  }
 }

 else if(command=="away")
 {
  digitalWrite(LED1,LOW);
  digitalWrite(LED2,LOW);
  Serial.println("Lights are OFF");
 }

 else if(command== "lightoff")
 {
  digitalWrite(LED1,LOW);
  digitalWrite(LED2,LOW);
  Serial.println("Lights are OFF");
 }

 else if(command== "lighton")
 {
  digitalWrite(LED1,HIGH);
  digitalWrite(LED2,HIGH);
  Serial.println("Lights are ON");
 }

 else
 {
  Serial.println("No commands have been subscribed");
 }
}
```

```
void publishData()
{
  humidity= dht.readHumidity();
  temperature= dht.readTemperature();
  ldr_data=analogRead(A0);
  pir_data=digitalRead(D3);

  if (isnan(humidity) || isnan(temperature)||isnan(ldr_data)||isnan(pir_data))
  {
    Serial.println("Failed to read from the sensors!");
    return;
  }

  String payload = "{\"d\":{\"Temperature\":";
  payload += temperature;
  payload += ","\"Humidity\":";
  payload +=  humidity;
  payload += ","\"LDR\":";
  payload +=  ldr_data;
  payload += ","\"PIR\":";
  payload +=  pir_data;
  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");
  }
}
```

NODERED FLOW



PROTOTYPE

ADVANTAGES

- Power consumption is minimized.
- Efficient
- Reliable
- Wireless access
- Choice of operation
- Reduces the usage of appliances when not required
- Assists in better maintenance of appliances
- Environment of the room can be analyzed
- Helps in switching off appliances immediately in case of fire accidents
- Movement in the buildings can be monitored
- Helps in decreasing the release of harmful pollutants from appliances.
- Saves time and energy
- Reduces electricity bills
- Better quality of life

.

DISADVANTAGES

- The appliances cannot be operated without wireless connectivity
- Failure of the appliances cannot be detected
- Repeated monitoring of sensors is needed to ensure they are in working condition
- The PIR detects animal movements as well
- Privacy of data stored is not guaranteed
- Not compatible with all appliances

APPLICATIONS

- Smart Lighting Control
- HVAC Regulation
- The energy saving system can be employed in new land development projects
- Gated Communities can make use of this system
- Can make your appliances smart
- Security systems can be developed

CONCLUSION

The Smart Building Project is an energy saving system aimed at decreasing the electricity consumption. By augmented all above mentioned sensors, app and cloud through Arduino IDE, NodeRed, MIT App Inventor 2 and IBM Watson platform the project has been developed to allow users to operate the connected appliances through the three modes of operation (Eco, Away and Manual). This project can be enhanced by integrating more sensors to increase the efficiency and enhance its features and consequently can be utilized for many more applications and future uses.

REFERENCES

- https://www.buildings.com/news/industry-news/articleid/21603/title/iot-smart-building-technology
- https://smartify.in/knowledgebase/iot-based-home-automation-system/
- https://www.electronicslovers.com
- http://www.iotgyan.com/iot-projects/smart-buildings-with-ibm-watson
- https://www.elprocus.com
- https://www.ahirlabs.com/2017/10/21/what-is-nodemcu-esp8266/