VIT UNIVERSITY

SCHOOL OF ELECTRONICS ENGINEERING

Fall Semester 2018-2019

SENSORS AND INSTRUMENTATION

Professor :Muthu Raja
Slot:TB2
Group members:
Member 1: Md Haque 18BEC0556
Member 2:Utsab Suvro Biswas 18BEC0914
Member 3:Tirthajit Das 18BEC0919
Member 4: Akshat Manral 18BEC0550

Home Automation using Google Assistant with NodeMCU

Abstract

Comfort is becoming a major priority in the 21st century. So the revolutions of computing and smartenvironment came into existence. Some technologies like Ubiquitous/pervasive and ambient intelligence satisfy themaximum need of smart world but these technologies are not tightly coupled with the internet, so the people need anothertechnology extension. Internet of Things (IoT) is an ideal buzzing technology to influence the internet and communicationtechnologies. IoT allows people and things to be connected anytime, anyplace, with anything and anyone, by using ideallyin any path/network and any service. Home automation system achieved great popularity in the last decades and it increasesthe quality of life. In this paper, an overview of home automation systems is discussed. This paper presents a design and prototype of the Home Automation system. The main part is NodeMCU, which has inbuilt Wi-Fi module, which will helpin controlling devices over the Internet. It supports a wide range of home automation devices like power managementsystems. In addition to that, an app would be developed which will allow the user to control their devices using the Google assistant.

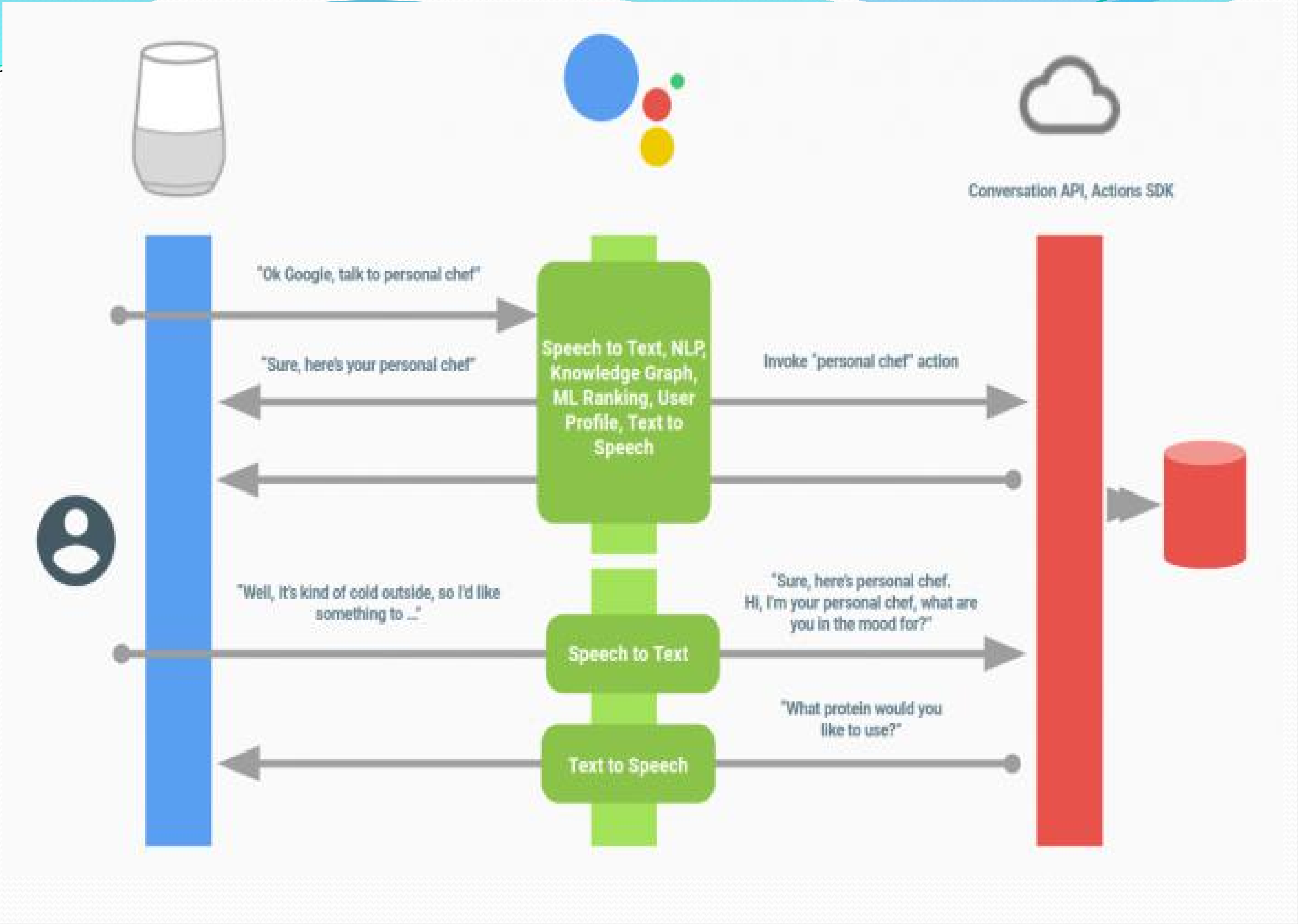
Read full abstract using the link:
https://www.academia.edu/38683395/Home_Automation_Based_On_IoT_Using_Google_Assistant

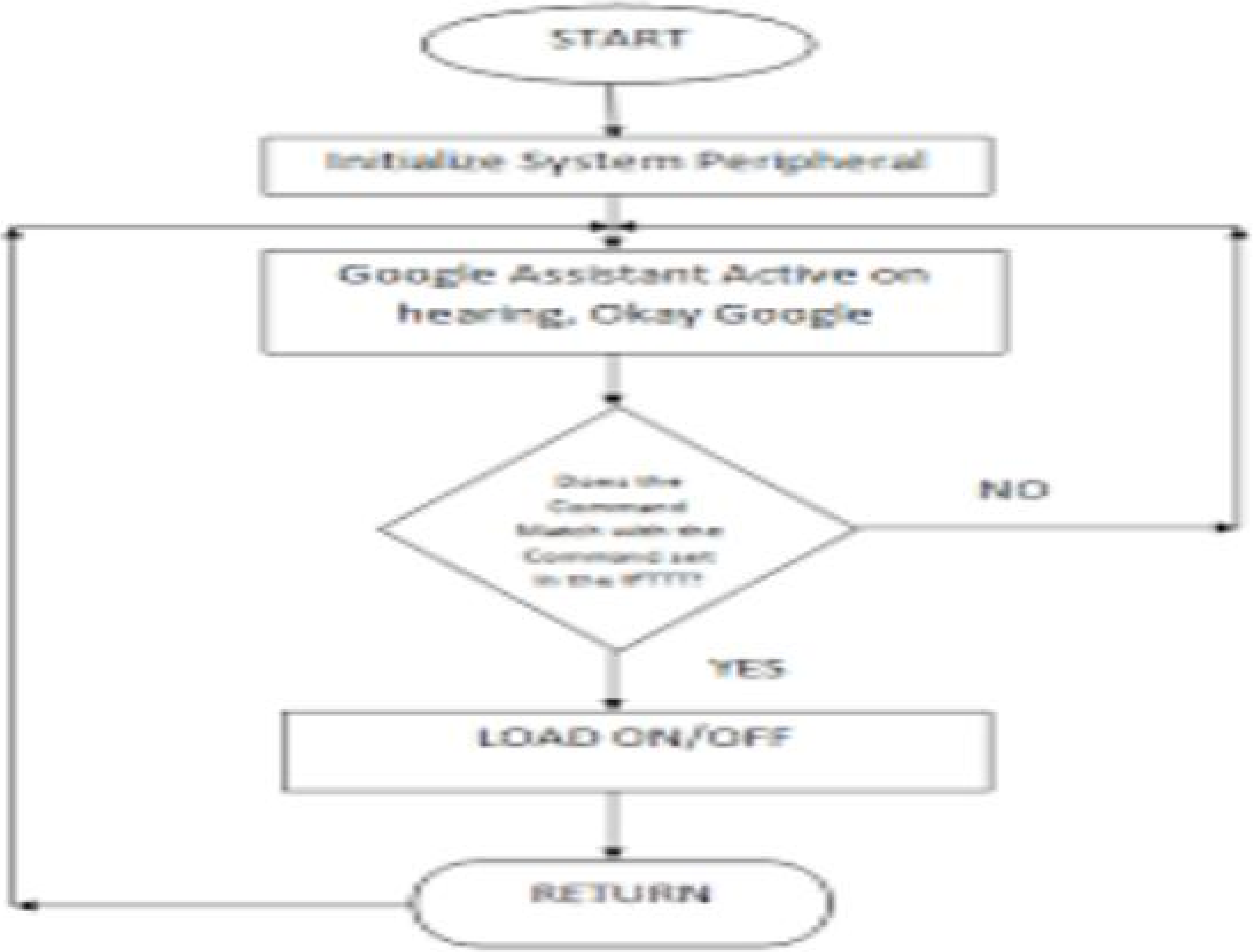
Required components:

- 1.Node MCU ESPR266(AURDINO IDE)
- 2. 5*Relay Module
- 3.Transistor BC547
- 4.IC 7805
- 5.2V Battery
- 6.Battery Connectors
- 7.Power supply
- 8.Jumper Wires
- 9.Bread Board
- 10.LED

METHODOLOGY:

The ESP8266 chip-set that has been used in this project has a built in Wifi-module and a 32 bit MCU which operates all the functionalism. The NodeMCU is programmed using Arduino IDE and it is connected to the internet with the wifi-module with the given username and password. The NodeMCU is programmed in such way so that it takes commands from the Applets set in io.adafruit.com which are controlled with Google Assistant. Hence we can change the pin output signals of the NodeMCU and switch on and off the 9v relay module connected to it which is further connected with a plug to plug in various electronic devices. In this way we can use this device for home automation.


Conversation API, Actions SDK
"Ok Google, talk to personal chef"
"Sure, here's your personal chef"
Speech to Text, NLP, Knowledge Graph, ML Ranking, User Profile, Text to Speech
Invoke "personal chef" action
"Well, it's kind of cold outside, so I'd like something to ..."
"Sure, here's personal chef. Hi, I'm your personal chef, what are you in the mood for?"
Speech to Text
"What protein would you like to use?"
Text to Speech



START

Initialize System Peripheral

Google Assistant Active on hearing, Okay Google

Does the Command Match with the Command set in the IFTTT?

NO

YES

LOAD ON/OFF

RETURN

Arduino Code

```
//NodeMCU Home Automation Using Google Assistant
//Code By Utsab Suvro Biswas (18BEC0914)
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define Relay1          D1
#define Relay2          D2
#define Relay3          D3
#define Relay4          D4

#define WLAN_SSID       "Redmi101"          // Your SSID
#define WLAN_PASS       "123456789"       // Your password

/************************* Adafruit.io Setup *********************************/

#define AIO_SERVER      "io.adafruit.com" //Adafruit Server
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "utsi101"          // Username
#define AIO_KEY         "2647a91359e14fdfa359f0d1f2402592"   // Authentication Key

//WIFI CLIENT
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME"/feeds/Relay1"); // Feeds name should be same everywhere
Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay2");
Adafruit_MQTT_Subscribe Light3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay3");
Adafruit_MQTT_Subscribe Light4 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay4");

void MQTT_connect();

void setup() {
```

Arduino Code

```
Serial.begin(115200);

  pinMode(Relay1, OUTPUT);
  pinMode(Relay2, OUTPUT);
  pinMode(Relay3, OUTPUT);
  pinMode(Relay4, OUTPUT);

  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);

  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  mqtt.subscribe(&Light1);
  mqtt.subscribe(&Light3);
  mqtt.subscribe(&Light2);
  mqtt.subscribe(&Light4);
}

void loop() {

  MQTT_connect();
```

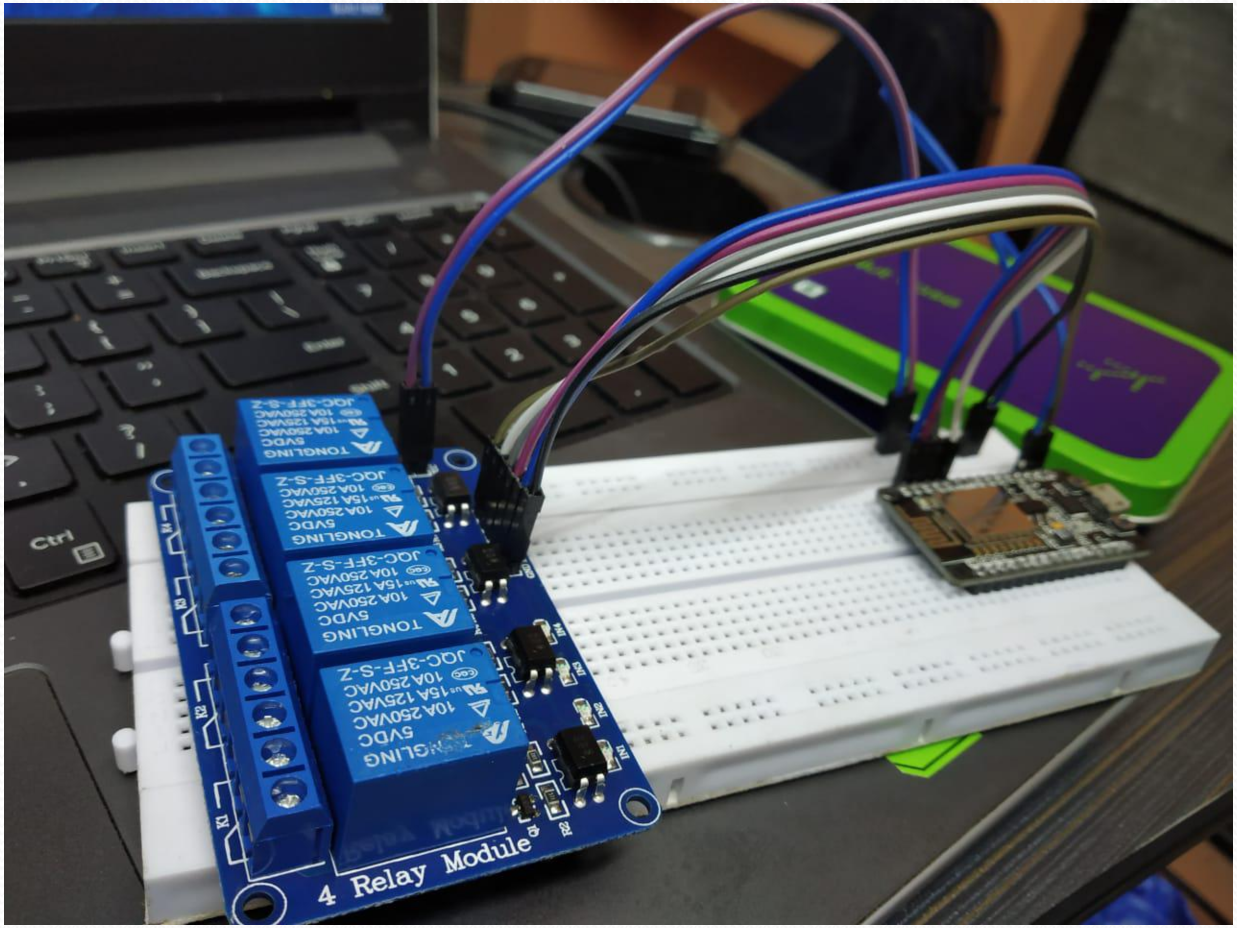
Arduino Code

```
 Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(20000))) {
    if (subscription == &Light1) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light1.lastread);
      int Light1_State = atoi((char *)Light1.lastread);
      digitalWrite(Relay1, Light1_State);

    }
    if (subscription == &Light2) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light2.lastread);
      int Light2_State = atoi((char *)Light2.lastread);
      digitalWrite(Relay2, Light2_State);
    }
    if (subscription == &Light3) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light3.lastread);
      int Light3_State = atoi((char *)Light3.lastread);
      digitalWrite(Relay3, Light3_State);
    }
    if (subscription == &Light4) {
      Serial.print(F("Got: "));
      Serial.println((char *)Light4.lastread);
      int Light4_State = atoi((char *)Light4.lastread);
      digitalWrite(Relay4, Light4_State);

    }
  }


}

void MQTT_connect() {
```

Arduino Code

```
  int8_t ret;

  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;

  while ((ret = mqtt.connect()) != 0) {
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);
    retries--;
    if (retries == 0) {
      while (1);
    }
  }
  Serial.println("MQTT Connected!");

}
```

Working Model



Thank You!