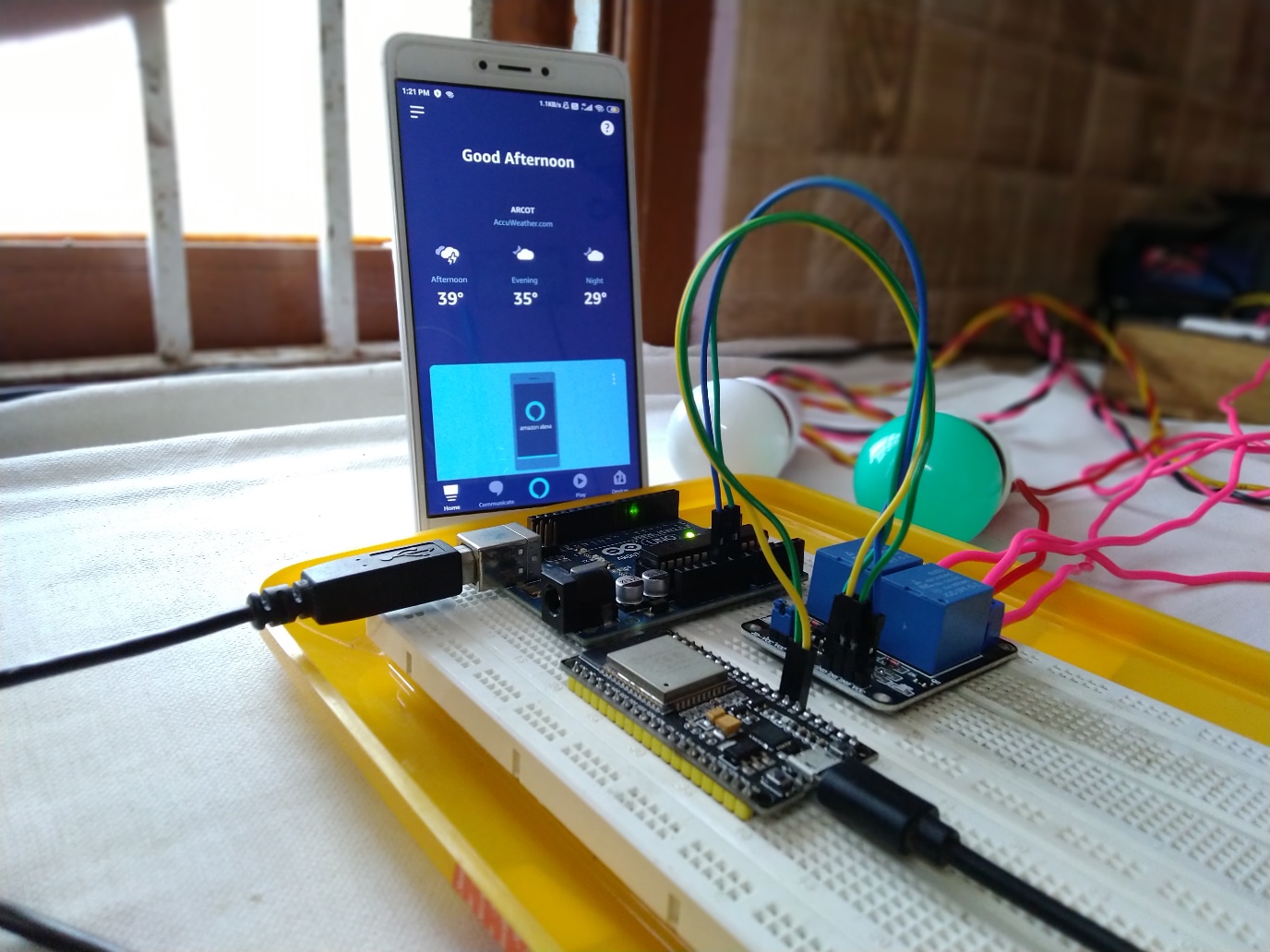
C4Projects

Capstone Project

Name: Nirmal kumar

Project Name: IoT based home Automation using Alexa



Short Description:

-In the modern world Iot plays an important role in major area’s .one of that is Home Automation.I implemented this idea using the Amazon alexa mobile application,hence you can control the any appliances on your home through the voice command and it really helpful for the physically challenged persons.

Required parts:

1)ESP-32

2)Relay

3)Power source

4)Alexa app

5)Sinric plugin

Simple Working model:

* Setup the sinric plugin to get the API and Device details.
* Then setup the commands in the Alexa app.
* When we give the command ,the Alexa recognize the words and send the commands to the sinric .
* The ESP-32 collects the data from the sinric and performs the action like ON and OFF, which uses the json to communicate with the sinric.

Source codes:

/\*

C4Projects

Capstone project : "IOT based HomeAutomation using Alexa"

Name: Nirmal kumar

Github:: https://github.com/NirmalKnock

\*/

int white\_light=16; //light1

int light=4; // light 2

int light1=18; //optional light 3

//required header files

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiMulti.h>

#include <WebSocketsClient.h>

#include <ArduinoJson.h>

#include <StreamString.h>

WiFiMulti WiFiMulti;

WebSocketsClient webSocket;

WiFiClient client;

#define MyApiKey "ff817bda-9f72-4e56-85f7-dee9961159a3" // sinric API Key.

#define MySSID "nirmal kumar" // Wifi network SSID

#define MyWifiPassword "nirmal@1234" // Wifi network password

#define HEARTBEAT\_INTERVAL 300000 // 5 Minutes

uint64\_t heartbeatTimestamp = 0;

bool isConnected = false;

void turnOn(String deviceId) {

if (deviceId == "5d850b87aa5bc849af8b16fe") // Device ID of first device

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(white\_light,HIGH); //lights ON

}

else if (deviceId == "5d86173c554c6f0923042d57") // Device ID of second device

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(light,HIGH);

}

else {

Serial.print("5d8e63a1acb3630272a5082f");

Serial.println(deviceId);

digitalWrite(light1,HIGH);

}

}

void turnOff(String deviceId) {

if (deviceId == "5d850b87aa5bc849af8b16fe") // Device ID of first device

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(white\_light,LOW);

}

else if (deviceId == "5d86173c554c6f0923042d57") // Device ID of second device

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(light,LOW);

}

else {

Serial.print("Turn off for unknown device id: ");

Serial.println(deviceId);

digitalWrite(light1,LOW);

}

}

void setup() {

Serial.begin(115200);

pinMode(white\_light,OUTPUT);

pinMode(light,OUTPUT);

pinMode(light1,OUTPUT);

WiFiMulti.addAP(MySSID, MyWifiPassword);

Serial.println();

Serial.print("Connecting to Wifi: ");

Serial.println(MySSID);

// Waiting for Wifi connect

while(WiFiMulti.run() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

if(WiFiMulti.run() == WL\_CONNECTED) {

Serial.println("");

Serial.print("WiFi connected. ");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

}

// server address, port and URL

webSocket.begin("iot.sinric.com", 80, "/");

// event handler

webSocket.onEvent(webSocketEvent);

webSocket.setAuthorization("apikey", MyApiKey);

// try again every 5000ms if connection has failed

webSocket.setReconnectInterval(5000);

}

void loop() {

webSocket.loop();

if(isConnected) {

uint64\_t now = millis();

// Send heartbeat in order to avoid disconnections during ISP resetting IPs over night

if((now - heartbeatTimestamp) > HEARTBEAT\_INTERVAL) {

heartbeatTimestamp = now;

webSocket.sendTXT("H");

}

}

}

void webSocketEvent(WStype\_t type, uint8\_t \* payload, size\_t length) {

switch(type) {

case WStype\_DISCONNECTED:

isConnected = false;

Serial.printf("[WSc] Webservice disconnected from sinric.com!\n");

break;

case WStype\_CONNECTED: {

isConnected = true;

Serial.printf("[WSc] Service connected to sinric.com at url: %s\n", payload);

Serial.printf("Waiting for commands from sinric.com ...\n");

}

break;

case WStype\_TEXT: {

Serial.printf("[WSc] get text: %s\n", payload);

#if ARDUINOJSON\_VERSION\_MAJOR == 5

DynamicJsonBuffer jsonBuffer;

JsonObject& json = jsonBuffer.parseObject((char\*)payload);

#endif

#if ARDUINOJSON\_VERSION\_MAJOR == 6

DynamicJsonDocument json(1024);

deserializeJson(json, (char\*) payload);

#endif

String deviceId = json ["deviceId"];

String action = json ["action"];

if(action == "setPowerState") { // Switch or Light

String value = json ["value"];

if(value == "ON") {

turnOn(deviceId);

} else {

turnOff(deviceId);

}

}

else if (action == "SetTargetTemperature") {

String deviceId = json ["deviceId"];

String action = json ["action"];

String value = json ["value"];

}

else if (action == "test") {

Serial.println("[WSc] received test command from sinric.com");

}

}

break;

case WStype\_BIN:

Serial.printf("[WSc] get binary length: %u\n", length);

break;

}

}