**15IT422E – INTERNET OF THINGS**

**SEMESTER - VI**

**ROOM TEMPERATURE SYSTEM**

*By*

UJJWAL YADAV (RA1611008010272)

**2018 - 2019**

**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

S.R.M. NAGAR, KATTANKULATHUR – 603 203

KANCHEEPURAM DISTRICT

**Objective:** To make a prototype of any useful product in our daily lives having the aspect of IoT.

**What is Home Automation?**

***A home automation system is a technological solution that enables automating the bulk of electronic, electrical and technology-based tasks within a home.***

***It uses a combination of hardware and software technologies that enable control and management over appliances and devices within a home.***

***Home automation is also known as domotics, and a home with an automation system is also known as a smart home.***

***Some of the processes within home automation system design, implementation and maintenance include:***

***• Installation of computer control for heating, ventilation and air conditioning systems***

***• Internet/remote/network access to all installed components and equipment***

***• Installation and maintenance of network-enabled surveillance cameras and physical security systems***

***• Central control and management capabilities over electrical fixtures and electronic appliances***

***Devices within the home automation system connect and communicate with each over a local wired or wireless network. The entire home automation system usually requires system management software, installation of device/appliance controllers, motion and temperature sensors and other components.***

**Making this using Arduino**

**Hardware Components:**

• Arduino Duemilanove or Uno board with ATmega328 microcontroller

• Spartronics Experimenter Shield (SES) v1 or v2) or EduShield v3.x

• USB (A-B) cable

• Hobby servo motor

• 9-12 VDC power supply (wall wart)

• Solderless breadboard

• 1kΩ resistor

• 3mm or 5mm red, green, or yellow LED

**Theory and steps:**

***Basically, the concept of home automation works by connecting smart devices and sensors (wired or wireless) to a system which act as the main control, and lastly to a graphical user interface for . Examples for smart devices/sensors are like humidity sensors, proximity sensors, brightness sensors, etc. that supports the relevant communication protocols (if required). While the control system may be represented by logic controllers (like PLC or similar devices) as well as SCADA and HMI to provide logical proccessing and graphical interface for end user's interaction.***

**As demonstrated, the system consists of the Arduino Uno and the Node MCU mounted on the breadboard with two other driver motors for the light and fan attached to the system. Once the wifi module and the Arduino Uno are connected to the PC via USB cables and a convenient hotspot is set up, the system uses this internet connectivity to connect to io.adafruit.com. The dashboard then contains of two**

**switched for the light and fan each and hence can be switched on or off to demonstrate the operation.**

**Schematics:**

**Code:**

**ARDUINO UNO**

a#include "DHT.h"

#include <SoftwareSerial.h>

#define DHTPIN 2 // what digital pin we're connected to

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial sw(11, 12);

void setup()

{

Serial.begin(9600);

sw.begin(9600);

dht.begin();

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.print("\*");

Serial.print(t);

Serial.print(",");

Serial.print(h);

Serial.println("#");

sw.print("\*");

sw.print(t);

sw.print(",");

sw.print(h);

sw.println("#");

delay(10000);

}

**NODE MCU:**

**#include <AdafruitIO\_WiFi.h>**

**#include <ESP8266WiFi.h>**

**#include <DNSServer.h>**

**#include <ESP8266WebServer.h>**

**#include <WiFiManager.h>**

**#include <SoftwareSerial.h>**

**#define IO\_USERNAME "roomth"**

**#define IO\_KEY "e9411606925b403e9da858eb06ede38d"**

**#define WIFI\_SSID ""**

**#define WIFI\_PASS ""**

**AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, WIFI\_SSID, WIFI\_PASS);**

**SoftwareSerial swSer(D7, D8, false, 256);**

**String response;**

**int P1;**

**int P2;**

**int current = 0;**

**int last = -1;**

**AdafruitIO\_Feed \*PR1 = io.feed("PR1");**

**AdafruitIO\_Feed \*PR2 = io.feed("PR2");**

**void setup()**

**{**

**pinMode(LED\_BUILTIN, OUTPUT);**

**digitalWrite(LED\_BUILTIN, LOW);**

**Serial.begin(9600);**

**swSer.begin(9600);**

**while(! Serial);**

**WiFiManager wifiManager;**

**wifiManager.autoConnect("WIFI MODULE");**

**Serial.println("connected...:)");**

**Serial.println("");**

**Serial.println("WiFi connected");**

**Serial.println("IP address: ");**

**Serial.println(WiFi.localIP());**

**Serial.print("Connecting to Adafruit IO");**

**io.connect();**

**while(io.status() < AIO\_CONNECTED) {**

**Serial.print(".");**

**delay(500);**

**}**

**Serial.println();**

**Serial.println(io.statusText());**

**delay(2000);**

**Serial.end();**

**for(int i=0;i<20;i++)**

**{**

**digitalWrite(LED\_BUILTIN, LOW);**

**delay(100);**

**digitalWrite(LED\_BUILTIN, HIGH);**

**delay(100);**

**}**

**Serial.begin(9600);**

**while(! Serial);**

**}**

**int ESPwait(String stopstr, int timeout\_secs)**

**{**

**bool found = false;**

**char c;**

**long timer\_init;**

**long timer;**

**response="";**

**timer\_init = millis();**

**while (!found) {**

**timer = millis();**

**if (((timer - timer\_init) / 1000) > timeout\_secs) { // Timeout?**

**Serial.println("!Timeout!");**

**return 0; // timeout**

**}**

**if (swSer.available()) {**

**c = swSer.read();**

**//Serial.print(c);**

**response += c;**

**if (response.endsWith(stopstr)) {**

**found = true;**

**delay(10);**

**swSer.flush();**

**Serial.flush();**

**Serial.println();**

**}**

**} // end Serial1\_available()**

**} // end while (!found)**

**return 1;**

**}**

**int ESPwait1(String stopstr, int timeout\_secs)**

**{**

**bool found = false;**

**char c;**

**long timer\_init;**

**long timer;**

**response="";**

**timer\_init = millis();**

**while (!found) {**

**timer = millis();**

**if (((timer - timer\_init) / 1000) > timeout\_secs) { // Timeout?**

**Serial.println("!Timeout!");**

**return 0; // timeout**

**}**

**if (Serial.available()) {**

**c = Serial.read();**

**//Serial.print(c);**

**response += c;**

**if (response.endsWith(stopstr)) {**

**found = true;**

**delay(10);**

**Serial.flush();**

**Serial.println();**

**}**

**} // end Serial1\_available()**

**} // end while (!found)**

**return 1;**

**}**

**void loop()**

**{**

**char c;**

**io.run();**

**if(swSer.available())**

**{**

**c=swSer.read();**

**swSer.println(c);**

**if(c=='\*')**

**{**

**if(ESPwait("#",3))**

**{**

**char \* strtokIndx;**

**response.remove(response.length()-1);**

**strtokIndx = strtok(const\_cast<char\*>(response.c\_str()),","); // get the first part - the string**

**P1 = atoi(strtokIndx);**

**strtokIndx = strtok(NULL, ",");**

**P2 = atoi(strtokIndx);**

**Serial.println(P1);**

**Serial.println(P2);**

**response="";**

**Serial.println("sending -> ");**

**PR1->save(P1);**

**PR2->save(P2);**

**}**

**}**

**}**

**if(Serial.available())**

**{**

**c=Serial.read();**

**Serial.println(c);**

**if(c=='\*')**

**{**

**if(ESPwait1("#",3))**

**{**

**char \* strtokIndx;**

**response.remove(response.length()-1);**

**strtokIndx = strtok(const\_cast<char\*>(response.c\_str()),","); // get the first part - the string**

**P1 = atoi(strtokIndx);**

**strtokIndx = strtok(NULL, ",");**

**P2 = atoi(strtokIndx);**

**Serial.println(P1);**

**Serial.println(P2);**

**response="";**

**Serial.println("sending -> ");**

**PR1->save(P1);**

**PR2->save(P2);**

**}**

**}**

**}**

**}**

**NODE MCU**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adafruit IO Config \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**// visit io.adafruit.com if you need to create an account,**

**// or if you need your Adafruit IO key.**

**#define IO\_USERNAME “roomth”**

**#define IO\_PASSWORD “roomth”**

**#define IO\_KEY "**e9411606925b403e9da858eb06ede38d"

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WIFI \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**// the AdafruitIO\_WiFi client will work with the following boards:**

**// - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471**

**// - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821**

**// - Feather M0 WiFi -> https://www.adafruit.com/products/3010**

**// - Feather WICED -> https://www.adafruit.com/products/3056**

**#define WIFI\_SSID “1”**

**#define WIFI\_PASS “12345678”**

**// comment out the following two lines if you are using fona or ethernet**

**#include "AdafruitIO\_WiFi.h"**

**AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, WIFI\_SSID, WIFI\_PASS);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FONA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**// the AdafruitIO\_FONA client will work with the following boards:**

**// - Feather 32u4 FONA -> https://www.adafruit.com/product/3027**

**// uncomment the following two lines for 32u4 FONA,**

**// and comment out the AdafruitIO\_WiFi client in the WIFI section**

**// #include "AdafruitIO\_FONA.h"**

**// AdafruitIO\_FONA io(IO\_USERNAME, IO\_KEY);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ETHERNET \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**// the AdafruitIO\_Ethernet client will work with the following boards:**

**// - Ethernet FeatherWing -> https://www.adafruit.com/products/3201**

**// uncomment the following two lines for ethernet,**

**// and comment out the AdafruitIO\_WiFi client in the WIFI section**

**// #include "AdafruitIO\_Ethernet.h"**

**// AdafruitIO\_Ethernet io(IO\_USERNAME, IO\_KEY);**

**Budget:**

**Items**

**Cost**

Arduino UNO

420

Wires

200

LEDs

100

Resistors

100

Other components

380

TOTAL

1200

**Conclusion**

The Home Automation system is of great utility to any household making it easier and more advanced for usage.