Report

on

Implementation of Smart Office

Interaction Technology
(DA619A)

Submitted

by

Manaswini Kolluru
Prakriti Dhang
Monica Handa

Supervisor

Reza Malekian

Department of Computer Science,
Faculty of Technology and Society
Malmö University
Malmö, Sweden

I. Intoduction

In today's world, IoT is one of the building blocks [1] where the scope of the Internet is going to be expanded more in near future because of different things will be connected with the Internet called as Internet of Things (IoT). We can control, monitorall the activities of lights, fans, air conditioners, the microwave oven, the refrigerator, television, washing machine and many more which are not only in our homes, but also in our work places, offices, institutes, schools.

In this assgnment, we need to implement a smart office which will help us to reduce energy consumption in our daily life. Below figure 1 shows the types of sensors is used to implement this project. The details of each sensors are explained in the following subsections. The assisgnment was divided into two parts. Part A was to read the sensor data and print in serial monitor. The sensor will react based on the evironment and turns on the LEDs accordingly. Part B was to use cloudMQTT. CloudMQTT was used to publish sensory data.

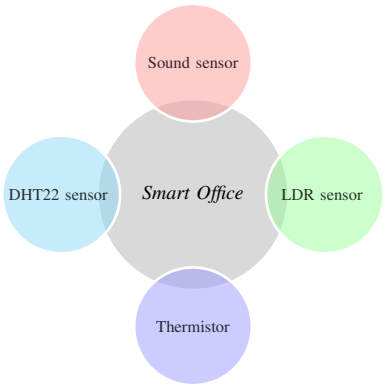

Fig. 1: Types of sensors used

A. Sensors

Sensors are devices which sense from the environment and act upon that environment through actuators. We used four different types of sensor to implement an smart office. The different types of sensors are enumerated as:

1) LDR sensor: The LDR(Light Dependent Resistor) [2] [3] is a light controlled variable resistor. LDR is use to detect light from the environment. It is also kown as photoresister. LDR have two pins, and which donot have polarity. We used LDR to control the green LED based on the light that is acquisiting from the environment.

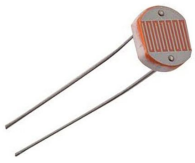

Fig. 2: LDR sensor

2) Thermistor NTC: Thermistor sensor [4] is a type of resistor whose resistance is dependent on temperature. NTC and PTC are two different types of thermistor sesnor. Thermistor requires 100K-120K ohm resistor. We used thermister to get the temperature from the environment or a room. We used thermister sensor to print the temperature value in Celcius and Fahrenheit.

Fig. 3: Thermistor NTC

3) DHT22 sensor: DHT sensor [5] [3] is the digital temperature and humidity sensor. It sense the environment air and produce the values.In DHT sensor there are 4 four pins. From the left, in figure 4 pin 1 is the VCC, pin 2 is the data, pin 3 is no connection and, pin 4 is the ground. DHT22 requires 10K ohm resister. We used DHT22 to control the yellow LED based on the temerature data.

Fig. 4: DHT22 sensor

4) Sound sensor: Sound sensor [3] is use to detect sound from the environment. It is a small board that contain a microphone, potentiomenter and some processing circuitry. There are four pins. The + sign pin is for voltage, GND is for ground, D0 is for digital output and A0 is for analog output. We used sound sensor to control the red LED whenever it detects noise from the environment.

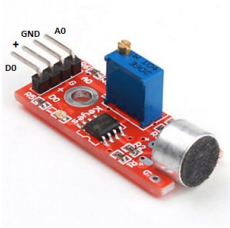

Fig. 5: Sound sensor

B. MQTT broker

MQTT have five components [6], which are as follows:

- Broker, which is the server that handles the data transmission between the clients.
- A topic, which is the place a device want to put or retrieve a message to/from.
- The message, which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.
- Publish, is the process a device does to send its message to the broker.
- Subscribe, where a device does to retrieve a message from the broker.

II. Methodology

A. Requirements

1) Hardware Components:

- LDR sensor
- Thermistor NTC
- DHT22 sensor
- Sound sensor
- Bread board
- Arduino UNO
- Ethernet shield
- 10k ohm resistor - 2
- 330 ohm resistor - 2
- 120k ohm resistor -1
- Jumper wires
- USB cable

2) Software Components:

- Arduino IDE
- CloudMQTT

B. Design

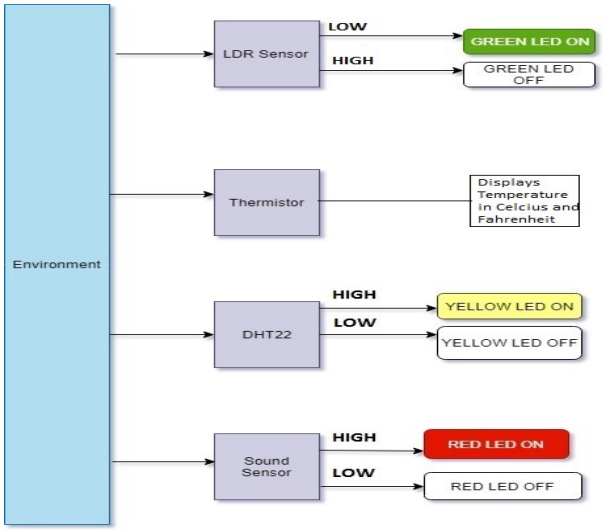

Fig. 6: Block diagram of the model

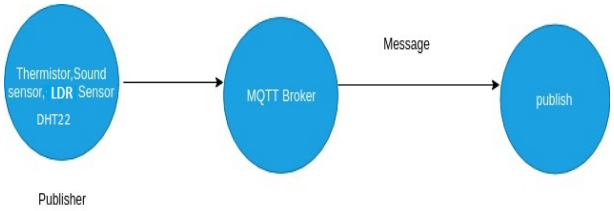

Fig. 7: Block diagram of the MQTT model

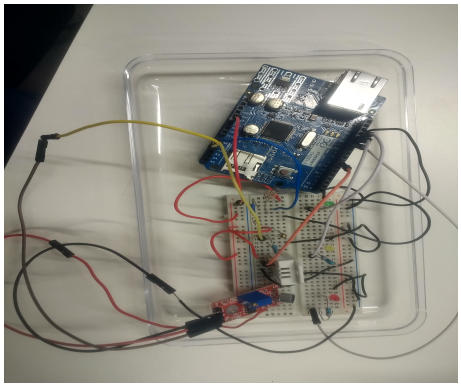

Fig. 8: Circuit diagram

III. Algorithms

A. LDR sensor

1) Initalise the LDR Status.
2) It reads the analog value and store it in the LDR Status.
3) Prints the LDR Status in the serial monitor and also publishes to MQTT.
4) Set a threshold value 750.
5) Checks if the LDR Status is less than equal to 750 or not. If yes green LED turns on, else green LED is off.
6) Publishes the status to MQTT.

B. Thermistor NTC

1) Intialise Vo and stores the anaog value from the thermistor.
2) Prints the Vo.
3) Calculates R2 = R1 * (1023.0 / (float)Vo - 1.0);
   logR2 = log(R2);
   T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
4) Calculate the temperature in Celsius Tc = T - 273.15.
5) Converts it to fahrenite Tf = (Tc * 9.0)/ 5.0 + 32.0.
6) Prints both the temperature in Celsius and Fahrenheit in serial monitor.
7) Publishes the status to MQTT.

C. DHT22

1) dht.readHumidity() reads humidity and store in float h
2) dht.readTemperature() reads the temperature in Celsius and store float t.
3) dht.readTemperature(true) reads the temperature in Fahrenheit and store in float f.
4) Prints all the value in serial monitor.
5) Checks if the temerature is higher than 60, then yellow LED turns on otherwise it will turn off.
6) Publishes the status to MQTT.

D. Sound sensor

1) Initialise noise.
2) Reads analog value from the sensor and store in noise.
3) Prints the value of noise in the serial monitor.
4) Set a threshold value
5) Check if the noise is greater equal to threshold value, if yes red LED turns on and after 1 sec it will turn off. If the noise is less than threshold value red LED will remain off.
6) Publishes the status to MQTT.

IV. Results

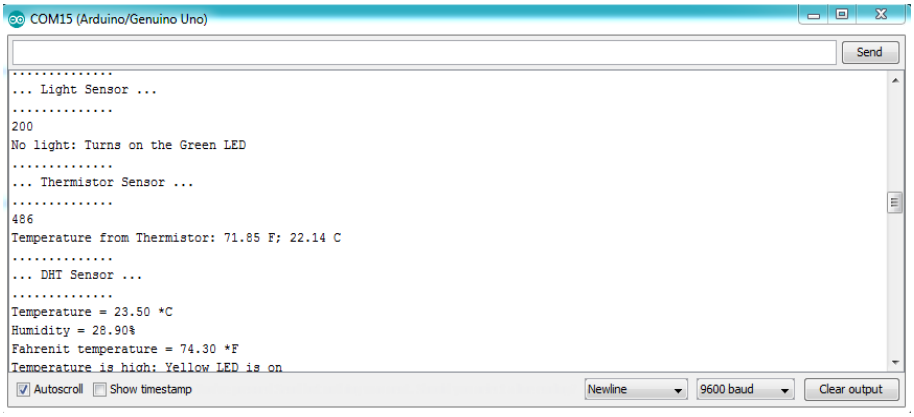

Fig. 9: Displays details about LDR sensor, Thermistor, and DHT22 sensor in the serial monitor

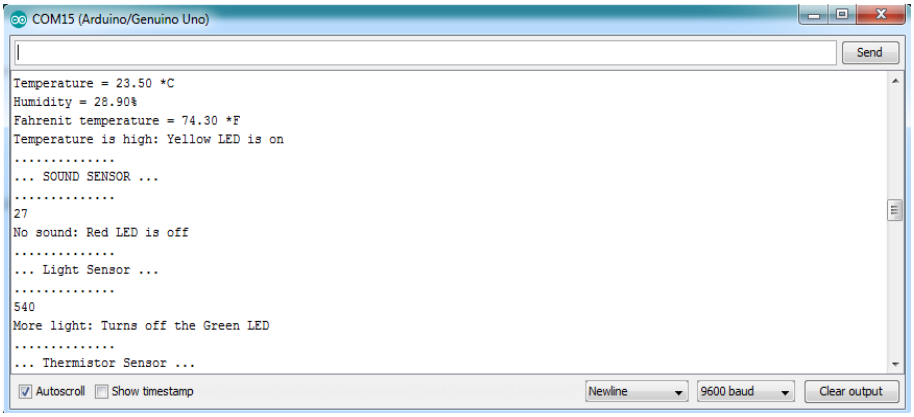

Fig. 10: Displays details about DHT22 sensor, sound sensor, and LDR sensor in the serial monitor

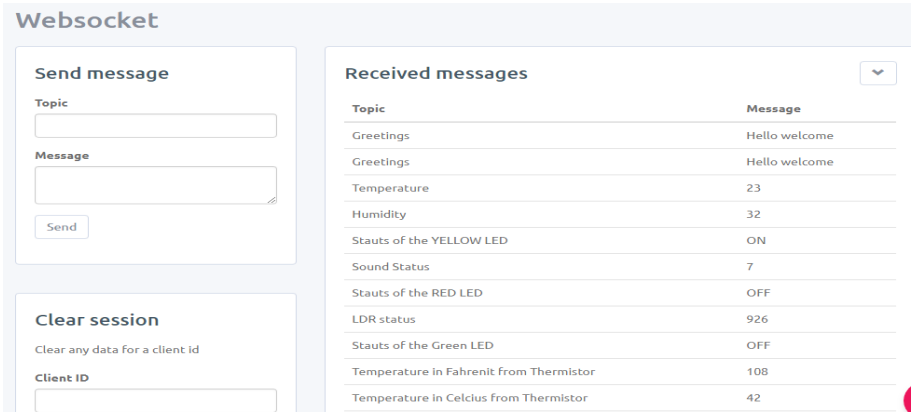

Fig. 11: Publishes the sensory data

V. Conclusion

The implementation of smart office is descibed with block diagam. Different types of sensors used and details of sensors is also discussed. CloudMQTT is used to publish the sensory data. We have also provided the algorithms for each sensor. We found the lab assignment is very interesting, as we worked with different types of sensors. The report ends with the list of references and source codes of both part A and part B in the appendix section.

References

[1] http://textofvideo.nptel.ac.in/106105166/lec1.pdf
[2] https://create.arduino.cc/projecthub/tarantula3/using-an-ldr-sensor-with-arduino-807b1c
[3] Geddes,M., Arduino project handbook Vol2
[4] http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/
[5] https://create.arduino.cc/projecthub/mafzal/temperature-monitoring-with-dht22-arduino-15b013
[6] https://1sheeld.com/mqtt-protocol/

APPENDIX

A. Code for Part A

```
#include <DHT.h> //library for DHT sensor
const int ledPinG = 13; // green led attached to digital pin 13 of the arduino board
const int ledPinY = 12; //yellow led attached to digital pin 12 of the arduino board
const int ledPinR = 11; //red led attached to digital pin 11 of the arduino board

const int ldrPin = A0; //LDR sensor attached to analog pin A0 of the arduino board

const int ThermistorPin = A1; //thermistor input attached to analog pin A1 of the arduino board
int Vo;
float R1 = 10000;
float logR2, R2, T, Tc, Tf;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

#define DHT22_PIN 7 //DHT sensor attached to digital pin 7 to the arduino board
#define DHTTYPE DHT22 //DHT sensor type
DHT dht( DHT22_PIN, DHTTYPE); // DHT pinnumber and type

const int soundpin = A2; //Sound sensor attached to analog pin A2 of the arduino board
const int threshold = 523; // fixed threshold for sound sensor

void setup() {
Serial.begin(9600);
pinMode(ledPinG, OUTPUT); // LED green output
pinMode(ldrPin, INPUT); // input for the LDR sensor

pinMode(ledPinY, OUTPUT); //LED yellow output
dht.begin();

pinMode(ledPinR, OUTPUT); //LED red output
pinMode(soundpin, INPUT); // input for the Sound sensor
}
void loop()

/* code for LDR sensor starts */
//https://create.arduino.cc/projecthub/tarantula3/using-an-ldr-sensor-with-arduino-807b1c

Serial.println(".....");
Serial.println("... Light Sensor ...");
Serial.println(".....");

int ldrStatus = analogRead(ldrPin);
Serial.print("LDR Status: ");
Serial.println(ldrStatus);
delay(1000);
if (ldrStatus ≤ 750) {

digitalWrite(ledPinG, HIGH);

Serial.println("No light: Turns on the Green LED ");

//Serial.println(ldrStatus);
} else

digitalWrite(ledPinG, LOW);
Serial.println("More light: Turns off the Green LED ");
```

```
   //Serial.println(ldrStatus);}

   /* code for LDR sensor ends */

   delay(2000);

   /*code for Thermistor sensor starts*/
//http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/

   Serial.println(".....");
Serial.println("... Thermistor Sensor ...");
Serial.println(".....");

   Vo = analogRead(ThermistorPin);
Serial.println(Vo);
R2 = R1 * (1023.0 / (float)Vo - 1.0);
logR2 = log(R2);
T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
Tc = T - 273.15;
Tf = (Tc * 9.0)/ 5.0 + 32.0;

   Serial.print("Temperature from Thermistor: ");
Serial.print(Tf);
Serial.print(" F; ");
Serial.print(Tc);
Serial.println(" C");

   /*code for Thermistor sensor ends */

   delay(2000);

   /* code for DHT sensor starts */
//https://create.arduino.cc/projecthub/mafzal/temperature-monitoring-with-dht22-arduino-15b013
//https://www.instructables.com/id/How-to-use-DHT-22-sensor-Arduino-Tutorial/

   float h =dht.readHumidity(); //reads humidity
float t =dht.readTemperature(); //reads the temperature in Celicus
float f = dht.readTemperature(true); //reads the temperature in Fahrenit

   Serial.println(".....");
Serial.println("... DHT Sensor ...");
Serial.println(".....");

   Serial.print("Temperature = ");
Serial.print(t);
Serial.println(" *C");
Serial.print("Humidity = ");
Serial.print(h);
Serial.println("%");
Serial.print("Fahrenit temperature = ");
Serial.print(f);
Serial.println(" *F");

   if(f ≥ 60){

   digitalWrite(ledPinY, HIGH);
Serial.println("Temperature is high: Yellow LED is on");
```

```
delay(1000);}

   else { digitalWrite(ledPinY, LOW);
Serial.println("Temperature is low: Yellow LED is off"); }
   /*code for DHT sensor ends */

   delay(2000);

   /* code for sound sensor starts */
//Arduino project handbook Vol2

   Serial.println(".............");
Serial.println("... SOUND SENSOR ...");
Serial.println(".............");

   int noise = analogRead(soundpin);
Serial.println(noise);

   if (noise ≥ threshold){

   digitalWrite(ledPinR, HIGH);
Serial.println("More sound: Red LED is on");
delay(1000);}

   else{ digitalWrite(ledPinR, LOW);
Serial.println("No sound: Red LED is off");

   /* code for sound sensor ends */

   }
```

B. Code for Part B

```
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

#include <DHT.h> //library for DHT sensor

const int ledPinG = 13; // green led attached to digital pin 13 of the arduino board
const int ledPinY = 12; //yellow led attached to digital pin 12 of the arduino board
const int ledPinR = 11; //red led attached to digital pin 11 of the arduino board

const int ldrPin = A0; //LDR sensor attached to analog pin A0 of the arduino board

const int ThermistorPin = A1; //thermistor input attached to analog pin A1 of the arduino board
int Vo;
float R1 = 10000;
float logR2, R2, T, Tc, Tf;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

#define DHT22_PIN 7 //DHT sensor attached to digital pin 7 to the arduino board
#define DHTTYPE DHT22 //DHT sensor type
DHT dht( DHT22_PIN, DHTTYPE); // DHT pinnumber and type
unsigned long readTime;

const int soundpin = A2; //Sound sensor attached to analog pin A2 of the arduino board
const int threshold = 523; // fixed threshold for sound sensor

// Update these with values suitable for your network.
byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
const char* server = ”m16.cloudmqtt.com”;
const char* username = ”yifzrfid”;
const char* password = ”UQW0Ni-cpe9W”;

char message_buff[100]; // this buffers our incoming messages so we can do something on certain commands
char buffer[10];
EthernetClient ethClient;
PubSubClient client(ethClient);

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

Serial.print(”Message arrived [”);
Serial.print(topic);
Serial.print(”] ”);
int i=0;
for (i=0;i¡length;i++)
Serial.print((char)payload[i]);
message_buff[i] = payload[i];}
Serial.println(); }
void reconnect() { // Loop until we’re reconnected
while (!client.connected()) Serial.print(”Attempting MQTT connection...”);

// Attempt to connect
if (client.connect(”arduinoClient”,username, password)) Serial.println(”connected”);
// Once connected, publish an announcement...
client.publish(”Greetings”,”Hello welcome”);
// ... and resubscribe
client.subscribe(”Welcome”); } else { Serial.print(”failed, rc=”);
```

```
   Serial.print(client.state());

   Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
delay(15000);
} } }
   void setup() {
   Serial.begin(9600);

   client.setServer(server, 17105);
client.setCallback(callback);

   Ethernet.begin(mac);
dht.begin();

   // Allow the hardware to sort itself out
//delay(1500);
Serial.println(Ethernet.localIP());
readTime = 0;

   pinMode(ledPinG, OUTPUT); // LED green output
pinMode(ldrPin, INPUT); // input for the LDR sensor

   pinMode(ledPinY, OUTPUT); //LED yellow output
//dht.begin();

   pinMode(ledPinR, OUTPUT); //LED red output
pinMode(soundpin, INPUT); // input for the Sound sensor

   }
   void loop() {
   if (!client.connected()) { reconnect();
}
   client.loop();

   if(millis() > readTime+200000){
   //delay(1000);
   /* code for DHT sensor starts */
//https://create.arduino.cc/projecthub/mafzal/temperature-monitoring-with-dht22-arduino-15b013
//https://www.instructables.com/id/How-to-use-DHT-22-sensor-Arduino-Tutorial/

   float h =dht.readHumidity(); //reads humidity
float t =dht.readTemperature(); //reads the temperature in Celicus
float f = dht.readTemperature(true); //reads the temperature in Fahrenit

   Serial.println(".....");
Serial.println("... DHT Sensor ...");
Serial.println(".....");
dtostrf(t,0, 0, buffer);
client.publish("Temperature",buffer);
dtostrf(h,0, 0, buffer);
client.publish("Humidity",buffer);

   Serial.print("Temperature = ");
Serial.print(t);
Serial.println(" *C");
Serial.print("Humidity = ");
Serial.print(h); Serial.println("%"); Serial.print("Fahrenit temperature = "); Serial.print(f);
```

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

   if(f>60) { digitalWrite(ledPinY, HIGH);
Serial.println("Temperature is high: Yellow LED is on");
client.publish("Stauts of the YELLOW LED","ON");

   //delay(1000);
} else { digitalWrite(ledPinY, LOW);
Serial.println("Temperature is low: Yellow LED is off");
client.publish("Stauts of the YELLOW LED","OFF");

   }
   /*code for DHT sensor ends */

   delay(1000);

   /* code for sound sensor starts */
//Arduino project handbook Vol2

   Serial.println(".....");
Serial.println("... SOUND SENSOR ...");
Serial.println(".....");

   int noise = analogRead(soundpin);
Serial.println(noise);
dtostrf(noise,0, 0, buffer);
client.publish("Sound Status",buffer);

   if (noise >= threshold) { digitalWrite(ledPinR, HIGH);
Serial.println("More sound: Red LED is on");
client.publish("Stauts of the RED LED","ON");

   delay(1000);
}
   else{ digitalWrite(ledPinR, LOW);
Serial.println("No sound: Red LED is off");
client.publish("Stauts of the RED LED","OFF");

   } /* code for sound sensor ends */
delay(1000);

   /* code for LDR sensor starts */
//https://create.arduino.cc/projecthub/tarantula3/using-an-ldr-sensor-with-arduino-807b1c

   Serial.println(".....");
Serial.println("... Light Sensor ...");
Serial.println(".....");

   int ldrStatus = analogRead(ldrPin);
Serial.print("LDR Status: ");
Serial.println(ldrStatus);

   dtostrf(ldrStatus,0, 0, buffer);
client.publish("LDR status",buffer);
//delay(1000); if (ldrStatus <= 750) {
   digitalWrite(ledPinG, HIGH);
```

```
   Serial.println(”No light: Turns on the Green LED ”);
client.publish(”Stauts of the Green LED”,”ON”);

   //Serial.println(ldrStatus);
   } else {
   digitalWrite(ledPinG, LOW);
Serial.println(”More light: Turns off the Green LED ”);
client.publish(”Stauts of the Green LED”,”OFF”);
//Serial.println(ldrStatus);

   }
   /* code for LDR sensor ends */

   delay(1000);

   /*code for Thermistor sensor starts*/
//http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/

   Serial.println(”.............”);
Serial.println(”... Thermistor Sensor ...”);
Serial.println(”.............”);

   Vo = analogRead(ThermistorPin);
Serial.println(Vo);
R2 = R1 * (1023.0 / (float)Vo - 1.0);
logR2 = log(R2);
T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
Tc = T - 273.15;
Tf = (Tc * 9.0)/ 5.0 + 32.0;

   //char buffer[10];
dtostrf(Tf,0, 0, buffer);

   Serial.print(”Temperature from Thermistor: ”);
Serial.print(Tf);
client.publish(”Temperature in Fahrenit from Thermistor”,buffer);
Serial.print(” F; ”);
Serial.print(Tc);
dtostrf(Tc,0, 0, buffer);
client.publish(”Temperature in Celcius from Thermistor”,buffer);
Serial.println(” C”);

   /*code for Thermistor sensor ends */
}
   } }
```