

Smart System

*** What is our smart system?**

Our smart system consists of three main modules , It is first job that controls is to adjust the room temperature whenever it is hot It will automatically turn hot the Air conditioner or a custom cooler system and if when the temperature goes back to normal acceptable temperature it will automatically turn the cooler off and also we made a room to control a custom heater system it will turn on when the temperature gets too cold till it setls down and go back to normal.

And then we supposed that our system is made for houses so we made one more additional thing , we wanted to make something to control the existence of the water in the water tank automatically so that no one needs to turn the water on if the tank is empty and when the tank gets full the water should stop flowing to the tank .

Last but not least we made it possible for us to monitor what happens in our system in the current moment so we installed an LCD to that data on it , On the LCD it displays the information like the current room temperature and if the heater or the cooler is on , it also displays the current water level of the water tank whether the water pump is on or off.

Description of the modules:

To make it easier for explaining in detail how our system is working we divided the explanation into three sections and every section explains a specific module.

- Temperature module:

In this module we will control the temperature of our room , when sensor measure the temperature if it is hot then a signal will be sent to the Air conditioner or cooler system to be on until the temperature goes back to normal (To specific degree) , and if it is cold then a signal will be sent to the Heater to be on until it reaches the normal temperature .

if the temperature of our room is normal then our devices (Air conditioner & Heater) will not be activated ...

What will we need *instruments*? We need:

LM35 sensor ==> about 40 EGP, 2*Single Channel
"220V-10A 5V" Relay ==> about 50 EGP, resistors, wires,
2 leds.

How to connect it *connecting*? LM35 sensor's range is (-50 to 150) , it has 3 pins , first one will be connected to

Vcc & the third to GND , and the middle pin is the output of our sensor which is analog voltage will be connected to (A0) analog pin .

In our simulation we will use leds to watch the output without using relays or air conditioner or heater, so we need 2 leds connecting to Arduino pin every led need resistor 330Ω to GND.

Coding : we defined some variables to our pins number , and used an array to store the strings which will be displayed on the LCD and some of the array elements will change when temperature change .

To help LCD to display the temperature, we used a variable to store the output of sensor then we used a float variable to store our temperature in Celsius.

$\text{temp} = (\text{Reading} * 5.0 * 100.0) / 1024.0 \implies$ Reading is the analog voltage output of sensor.

then we used if-elseif statement to control in our leds which is supposed to be in reality is our heater and cooler systems . we used `*strcpy*` which is C method helped us to change values in our array .

The normal temperature of the room we put is as $\in]20,30[$ if the current temperature is not in this period then one of the systems will work till the temp is normal.

```

char* snetences_list[] = {
    // temprature
    "Tmp is" ,
    // cooler on or heater
    "No cooler or heater",
    // water level
    "water lvl is ",
    // pumb on or off
    "pumb off"
};

void measureTemp() {
    // Temperature alarm
    reading = analogRead(tempsensor);
    temp = (reading * (5.0/1024.0)*100.0);
    if(temp >= 30 ){
        digitalWrite(Air_condi_Led,HIGH);
        digitalWrite(Heating_Led,LOW);
        strcpy(snetences_list[1] , "Cooler is on");
    }
    else if(temp <= 20 ){
        digitalWrite(Heating_Led,HIGH);
        digitalWrite(Air_condi_Led,LOW);
        strcpy(snetences_list[1] , "Heater is on");
    }
    else if (temp >= 20 and temp <=30 ){
        digitalWrite(Air_condi_Led,LOW);
        digitalWrite(Heating_Led,LOW);
        strcpy(snetences_list[1] , "Temp is good");
    }
}

```

- Water level module:

We use ULTRA SONIC Sensor to measure distance, this sensor has two parts trigger and echo, Trigger sends sound waves to hit the body we need to measure the distance between and Echo receive it, We measure the timing of this

process and calculate the distance throws this law : $D = T \cdot S$ (D = distance , T = time , S =sound speed).

In our project we used the ULTRA SONIC Sensor to measure water level in the tank , we put the sensor at the top of the tank and measure the distance between sensor and Surface of the water and if the distance is greater than 30cm - for example - we order the water pump to run and fill the tank to reach our point and then stops .

```
void measureWaterLvl() {
    // water level
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    long duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    waterlvl= duration*0.034/2;
    // Prints the distance on the Serial Monitor
    Serial.print("Distance:");
    Serial.println(waterlvl);
    //delay(1000);

    controlPumb();
}

void controlPumb() {
    // start the pumb
    if (waterlvl > 100) {
        digitalWrite(motor, HIGH);
        strcpy(snetences_list[3] , "Pumb on");
    } else {
        digitalWrite(motor, LOW);
        strcpy(snetences_list[3] , "Pumb off");
    }
}
```

- LCD module:

This module is responsible for displaying the data on LCD.

The LCD functionality is in the main thread because it is the most frequent operation.

The temp thread is supposed to happen every specific time interval measures the temperature and changes the value of temp variable.

And the water level thread also supposed to happen every specific time to measure the water level.

We first measure the temperature and water level.

Then we display the measured temperature, water level and which device is on.

```

void loop(){
    tempThread.check();
    waterThread.check();
    // lcd in the main thread to display the sentences we want to show
    // the lcd functionality in the main thread because it is the most frequent operation
    // the temp thread is supposed to happen every specific time interval measures the temp and change the value
    // in the sentences_list array about the temperature;
    int i;
    if (temp != 0 and waterlvl != 0){
        for (i = 0; i <= sizeof(sentences_list)/sizeof(char *); i++){

            //tempThread.check();

            lcd.clear();

            tempThread.check();
            waterThread.check();

            lcd.setCursor(0, 0);
            lcd.print(sentences_list[i]);

            if (i == 0){
                lcd.setCursor(0, 1);
                lcd.print(temp);
            } else if (i == 2){
                // write the water lvl
                lcd.setCursor(0, 1);
                lcd.print(waterlvl);
            }

            delay(2000);
        }
    }
}

```

*** Problems we faced:**

- **No multithreading !** : We faced big issue in dealing with doing every functionality of the system parallelly , we wanted to do it parallelly because every functionality of the system is separate from each other and at the same time every functionality supposed to happen frequently and because the Arduino's microprocessor is a single core processor it doesn't support multi-threading .

- **Monitor the system through an LCD** : At some point we wanted to know what is going on inside of our system to know what is happening and give better visualization of it but that got hard when we are dealing with just 16 * 2 LCD which means it is only has 2 rows to display all information of the system on which was a problem as the information was larger than being displayed on just 2 rows !

*** Our solutions:**

- **For No multithreading support** : We handled this issue by using library in the Arduino is called Timed Action , it is used to do some actions every specific time intervals the we define , we used this library to handle the measurements of temperature and water level as we gave made them only happen at specific time and in the main Loop function the main thread we put the LCD controlling code as it is the

most frequent thing that happens in our system as it always displays the data from the system.

By using timed action, we made it possible for everything works in the Arduino in harmony as they didn't interrupt each other anymore, our solution is called **Prototherian**.

- **For multi viewing on LCD :** We used some little hack to deal with this idea , we basically showed every sentence of information we wanted to show separately every specific time the LCD shows a different sentence on the screen display something about our system , in our code we defined an array that contains our sentences and data that we wanted to display about the system and this array gets changed from the temperature function and water level function every time they done their measurements .

* **Ideas for development:**

- we could upgrade this system by using a better way for monitoring the system such as teeing it to a cloud and controlling our system from the internet or we could link it to a wifi module and monitor the system through a phone app while we are in the house

- we could add more sensors to control the home in more smart way.

* **Code:**

On GitHub:

<https://github.com/hossamhasanin/smart-home-system>

*** Done by :**

حسام حسانين عبد التواب عبد الرحيم

عبد الرحمن زغلول محمد الهادي

عبد الرحمن امين ابراهيم امين سبع

المعتصم بالله عبد الهادي عثمان