House Heating Manager

Alexandre Soldano
Yosra Ibrahem
Jasser Hmadi

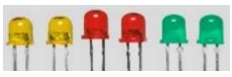
ESILV LÉONARD DE VINCI | ÉCOLE D'INGÉNIEURS PARIS-LA DÉFENSE

House Heating Manager

The main goal of embedded systems is to ease everyone's life, making ordinary devices intelligent. For this final project, our goal was to design and conceive an intelligent heater which depending on the users' timetable, would auto regulate the temperature inside a house. With this tutorial of House Heating Manager, we will propose you an entertaining simulation to show you how it works.

Step 1: Get the material required

In order to make your own house heating manager simulation proceed by getting all these parts first:

- Breadboard 
- Arduino Yun board (Uno works as well) 
- Temperature sensor TMP36 
- 6 LEDs (10mm) 
- 6 resistors 
- Cables 
- Power supply cable (USB) 

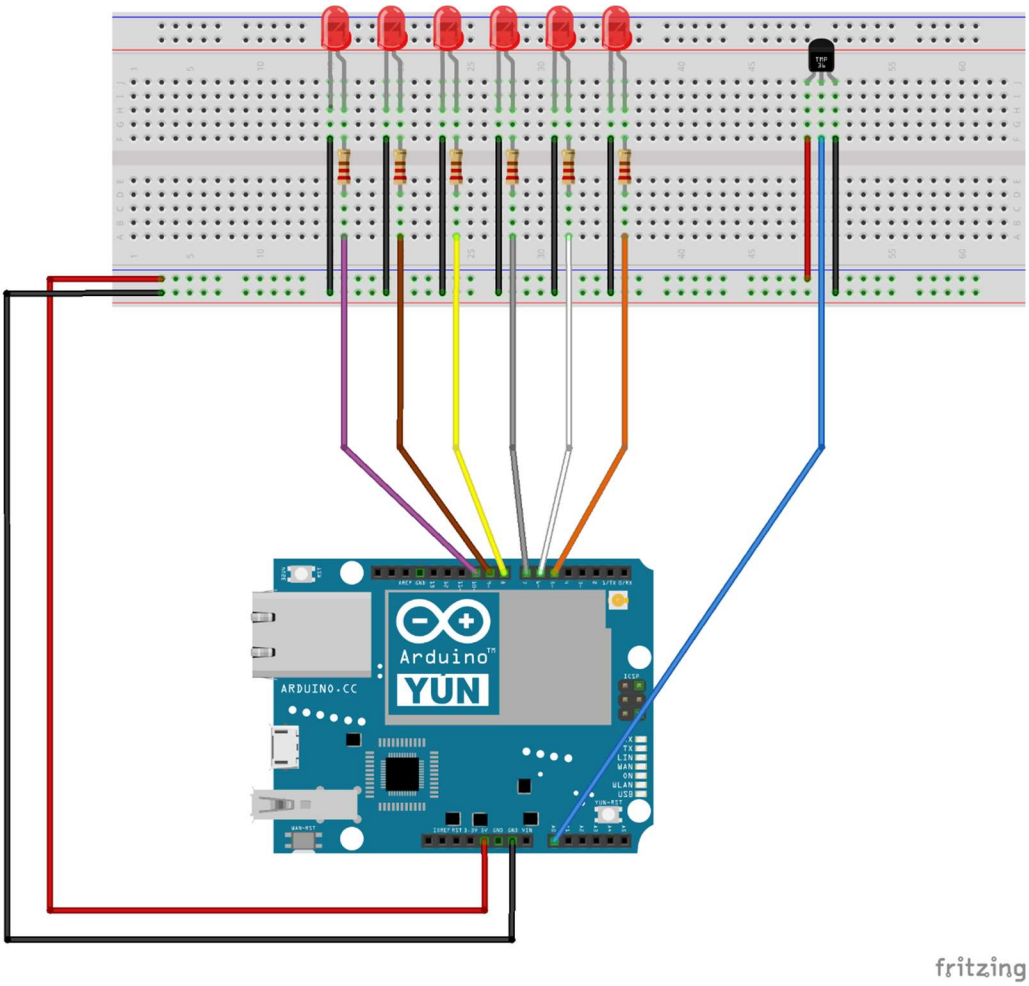
Theoretical only:

You will need these parts to get further and make your own House Heating Manager (if you are only interested in that, skip the next steps and go directly to step 5):

- Raspberry Pi 
- Micro SD card 
- Ethernet cable 
- Another power supply cable 

Step 2: Assemble the parts

Here is a Fritzing schema that shows how to build the simulation with all the parts listed above on the first kit (do not forget that the left part of the breadboard is independent from the right one and you'll need to connect one to another to use both parts):



Once you have realized the montage, you can go to the next step. If you need any information on any electronic material, you can search on the Internet their datasheet.

Step 3: Arduino code

For this part, download Arduino here: https://www.arduino.cc/en/Main/Software

You will then run Arduino after downloading and installing it. Open the **Arduino.ino** file that is on the GitHub of this project, or just copy the following code and paste it on Arduino:

```
#include <Wire.h>

//Definition of the Pins places
const int ledPin1 = 10;
const int ledPin2 = 9;
const int ledPin3 = 8;
const int ledPin4 = 7;
const int ledPin5 = 6;
const int ledPin6 = 5;
const int sensorPin = A0; // TMP 36


// variables:
double sensorValue = 0; // the sensor value
int index = 0; // variable to put all the led to LOW when there is a particular change in the temperature, to put the right leds to HIGH

//Important variable !!!
float mean_temperature = 18; // Define the mean temperature in the room

void setup() {
  // initialize the digital output of the Pins
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
    Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

  // read the sensor:
  sensorValue = analogRead(sensorPin);
      // apply the calibration to the sensor reading
    double  sensorV = ((sensorValue*5000/1023)-500)/10;
          // *5000/1023 : convertion of the analogic data for 5V, so 5000mV, on a range of 1023.
          // -500 : corresponding to an offset of 500mV from the data sheet (link below, page 8) for the TMP36.
          // /10 : a scaling of 10mV/°C

      //Write the value of the sensor on the serial monitor
    Serial.print("Temperature : ");
    Serial.println(sensorV);

      //Defining the difference of temperature between the actual and the mean temperature
    float difference = sensorV - mean_temperature;
```

```
if(difference <-3)
{
  if(index != 1)
  {
    LedOff();
  }
  digitalWrite(ledPin1, HIGH);
  index = 1;
}

if(difference <-1 && difference >= -3)
{
  if(index != 2)
  {
    LedOff();
  }
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
  index = 2;
}

if(difference <1 && difference >= -1)
{
  if(index != 3)
  {
    LedOff();
  }
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
  digitalWrite(ledPin3, HIGH);
  index = 3;
}

if(difference <3 && difference >= 1)
{
  if(index != 4)
  {
    LedOff();
  }
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
  digitalWrite(ledPin3, HIGH);
  digitalWrite(ledPin4, HIGH);
  index = 4;
}

if(difference <5 && difference >= 3)
{
  if(index != 5)
  {
    LedOff();
  }
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, HIGH);
  digitalWrite(ledPin3, HIGH);
  digitalWrite(ledPin4, HIGH);
  digitalWrite(ledPin5, HIGH);
```

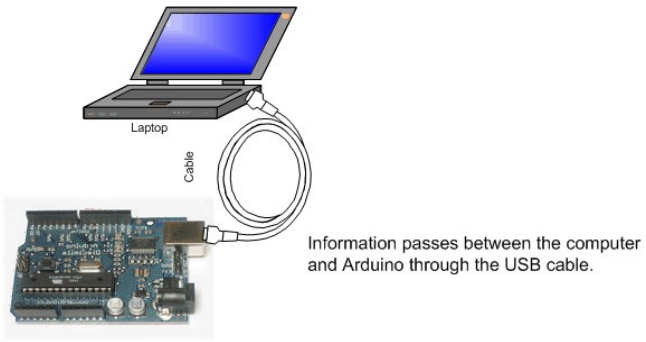
```
    index = 5;
  }

  if(difference >= 5)
  {
    if(index != 6)
    {
      LedOff();
    }
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    digitalWrite(ledPin3, HIGH);
    digitalWrite(ledPin4, HIGH);
    digitalWrite(ledPin5, HIGH);
    digitalWrite(ledPin6, HIGH);
    index = 6;
  }
    //Running the regulation system
      delay(1000);
}

 //To turn off all the LEDs
void LedOff() {
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, LOW);
  digitalWrite(ledPin3, LOW);
  digitalWrite(ledPin4, LOW);
  digitalWrite(ledPin5, LOW);
  digitalWrite(ledPin6, LOW);
}
```

Please, put the value of the usual temperature of the room you are in right now into the variable **mean_temperature**, at line 14, as it is an important variable that vary from one another.

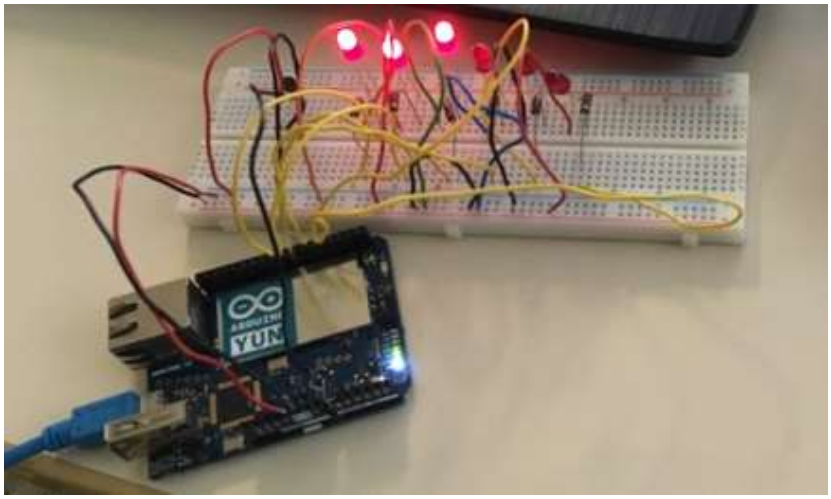
You can now connect the Arduino to your Laptop with the power supply cable.



The Arduino should be recognized within the application in your laptop. You can now upload the previous code.

Step 4: Test and enjoy

You can now see the LEDs light up. If the temperature of the room is close to the value you have given to the variable mean_temperature, then 3 LEDs should be lighted up as seen in this image:



If, however, you have less than 3 LEDs lighted up, it means that the temperature of the room you're in is colder than it usually is. And if you have more than 3, then the room you're in is hotter than it usually is.

You can regulate the temperature of the room and watch the LEDs light up or being turned off as the temperature changes. Or if you just want to test it, you can put a cold surface close to the temperature sensor or a hot one, such as the tip of your fingers which should be hot enough to light up all the LEDs in most cases.

Step 5: Going further

From this point on, it's only theoretical since it hasn't been done.

Firstly, you will connect your Raspberry to your laptop with the Ethernet cable and the power supply cable and also insert the micro SD card into the Raspberry. You will, then, enable an Ethernet connection from your laptop to your Raspberry. Here is how you can do it: http://www.dummies.com/computers/operating-systems/windows-7/how-to-share-an-internet-connection-in-windows-7/

Then, find the IP address associated to the Ethernet port of your laptop. (You can use this: https://kb.wisc.edu/helpdesk/page.php?id=6526 or open a Terminal and try the command ifconfig)

Now, download PuTTY (here: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html) and connect to your Raspberry using its IP address obtained earlier.

The goal here is now to create a web page using an HTML file to upload the timetable. Then, you have to create a file for the database. You can make a Python file that will stock the timetable of the user.

You can create this time a C file that can regulate the temperature before the user arrives at home or after he leaves home, using an embedded system connected to the Raspberry that can regulate the temperature of the thermostat.

We could have also implemented a lot of other features into this project, such as an InfraRed sensor to capture movements and start the regulation system, and also a LCD display. A lot more could be implemented from this project.

If you want to share what you have accomplished following this project, you can post it in the GitHub of this project: https://github.com/EmbeddedS8/House_Heating_Manager