Temperature Control System

Introduction:

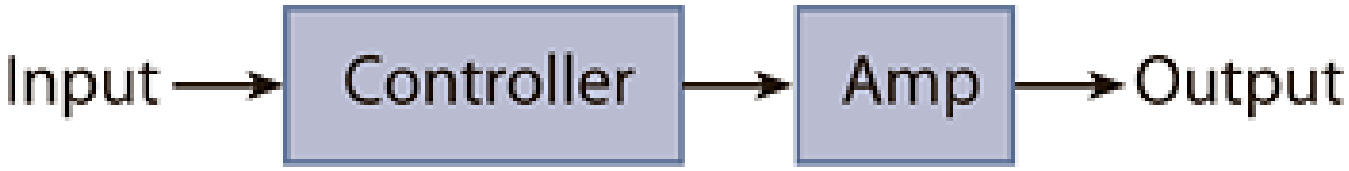
A Temperature Controlled System is a type of control system that automatically controls the temperature of an object or an area.

Temperature controllers are needed in any situation requiring a given temperature be kept stable. This can be in a situation where an object is required to be heated, cooled or both and to remain at the target temperature (set point), regardless of the changing environment around it.

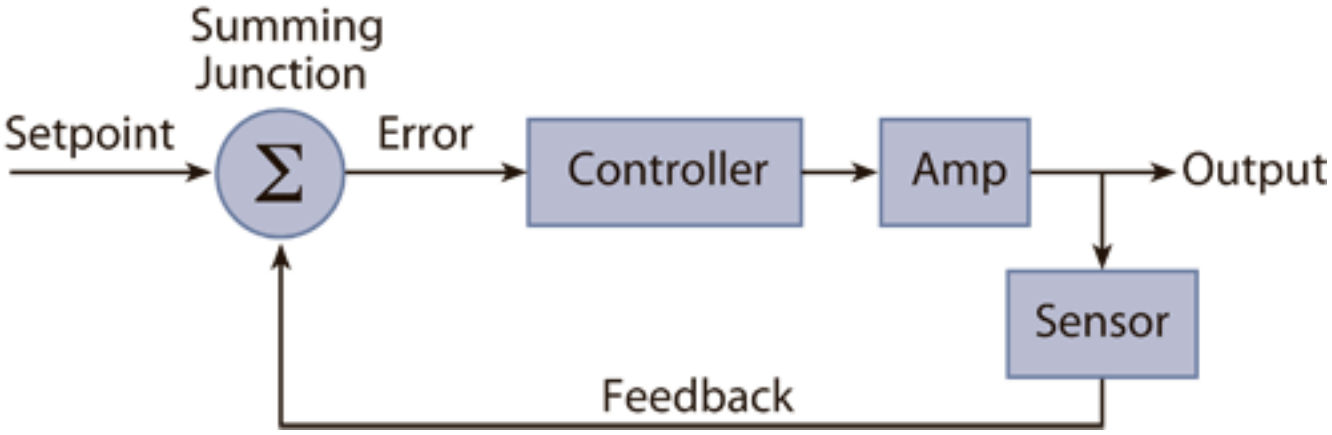
We commonly use temperature control systems in Air Conditioners, Refrigerators, geysers, etc. where the temperature is automatically adjusted as per the input settings. In order to implement a temperature control system, we need a temperature sensor, a controller, and a cooling system.

There are two fundamental types of temperature control; open loop and closed loop control.

First: Open loop is the most basic form and applies continuous heating/cooling with no regard for the actual temperature output. It is analogous to the internal heating system in a car. On a cold day, you may need to turn the heat on to full to warm the car to 75°. However, during warmer weather, the same setting would leave the inside of the car much warmer than the desired 75°.



Second: Closed loop control is far more sophisticated than open loop. In a closed loop application, the output temperature is constantly measured and adjusted to maintain a constant output at the desired temperature. Closed loop control is always conscious of the output signal and will feed this back into the control process. Closed loop control is analogous to a car with internal climate control. If you set the car temperature to 75°, the climate control will automatically adjust the heating (during cold days) or cooling (during warm days) as required to maintain the target temperature of 75°.



Applications:

Temperature controllers in industry work much the same way they do in common household applications. A basic temperature controller provides control of industrial or laboratory heating and cooling processes. In a typical application, sensors measure the actual temperature. This sensed temperature is constantly compared to a user set point. When the actual temperature deviates from the set point, the controller generates an output signal to activate other temperature regulating devices such as heating elements or refrigeration components to bring the temperature back to the set point.

It is used in:

1) Heat treat/Oven: Temperature controllers are used in ovens and in heat-treating applications within furnaces, ceramic kilns, boilers, and heat exchangers.

2) Packaging: In the packaging world, machinery equipped with seal bars, glue applicators, hot melt functions, shrink wrap tunnels or label applicators must operate at designated temperatures and process time lengths. Temperature controllers precisely regulate these operations to ensure a high quality product output.

3) Plastics: Temperature control in the plastics industry is common on portable chillers, hoppers and dryers and molding and extruding equipment. In extruding equipment, temperature controllers are used to precisely monitor and control temperatures at different critical points in the production of plastic.

4) Healthcare: Temperature controllers are used in the healthcare industry to increase the accuracy of temperature control. Common equipment using temperature controllers includes laboratory and test equipment, autoclaves, incubators, refrigeration equipment, and crystallization growing chambers and test chambers where specimens must be kept or tests must be run within specific temperature parameters.

How it works:

All controllers, from the simplest to the most complicated, operate in a similar way. Controllers maintain a preset value for a variable or parameter. The controller requires two variables: the actual input signal and the intended set point value. The process value is another name for the input signal. Depending on the controller, the input to the controller is sampled several times per second.

The value of the input, or process, is then compared to the setpoint value. If the actual value does not match the setpoint, the controller changes the output signal based on the difference between the setpoint and the process value, as well as whether the process value is approaching or departing from the setpoint. This output signal then triggers some sort of response in order to get the real value closer to the setpoint. The output power value is usually updated by the control algorithm, which is subsequently applied to the output.

The type of controller determines the type of control action taken. If the controller has an ON/OFF control, for example, the controller determines whether the output should be switched on, off, or left in its current condition.

One of the most basic types of control is the ON/OFF switch. It operates by establishing a hysteresis band. A temperature controller, for example, could be used to regulate the temperature inside a room. An error signal would show a –1° difference if the setpoint is 68° and the actual temperature is 67°. The controller would then send a signal to increase the applied heat in order to return the temperature to the setpoint of 68 degrees. The heater turns off when the temperature hits 68°. The controller does nothing and the heater remains off when the temperature is between 68° and 67°. The heater will turn on again once the temperature hits 67°.

PID control, unlike ON/OFF control, calculates the exact output value needed to maintain the specified temperature. The output power can be set anywhere between 0 and 100%. The output drive is proportional to the output power value when an analogue output type is employed. If the output is a binary type, such as a relay, SSR driver, or triac, it must be time proportioned to achieve an analogue representation.

Controllers can also have a number of additional optional features. One of these is communication capability. A communication link lets the controller communicate with a PLC or a computer. This allows data exchange between the controller and the host. An example of typical data exchange would be the host computer or PLC reading the process value.

System identification for our project:

Input: Desired temperature
Output: Heat
Controller: Temperature sensor
Heating element: A lamp with tungsten coil

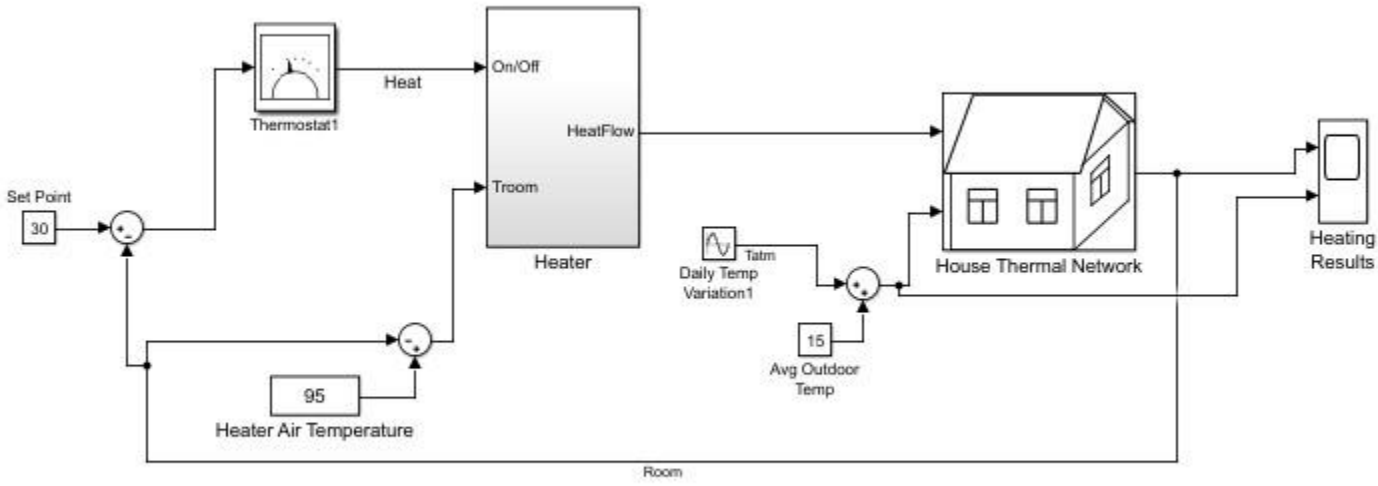
The heating element will release heat to the system which will be adjusted to our desired temperature by the sensor. Once this process is completed, the heating element will stop releasing the heat and the temperature will decrease gradually till a certain point then the heating element will start releasing heat again.

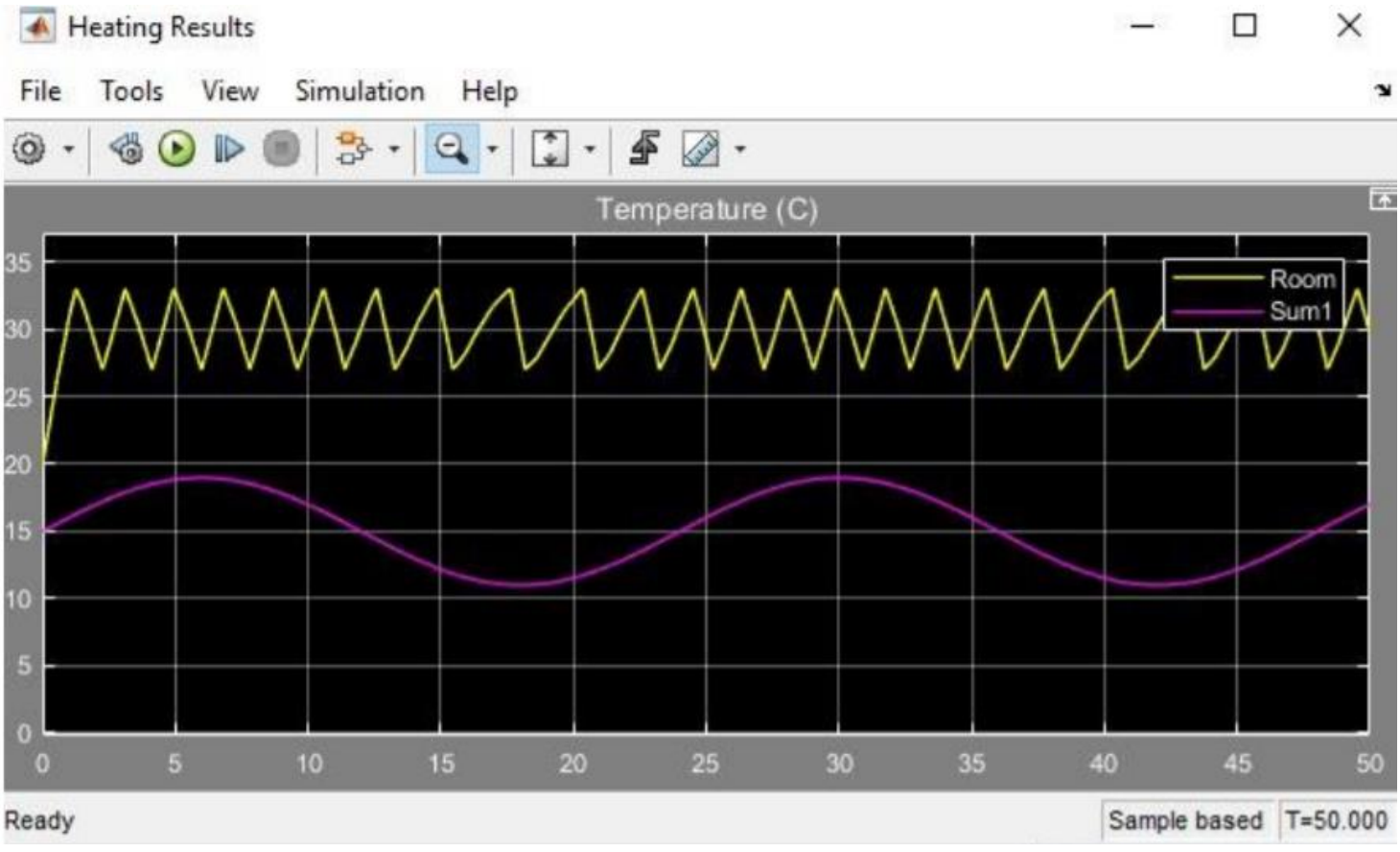
Components used:

1) Arduino Uno
2) Tungsten lamp
3) House from wood
4) LM35 temperature sensor
5) Breadboard
6) Relay

Matlab:

Whole project:

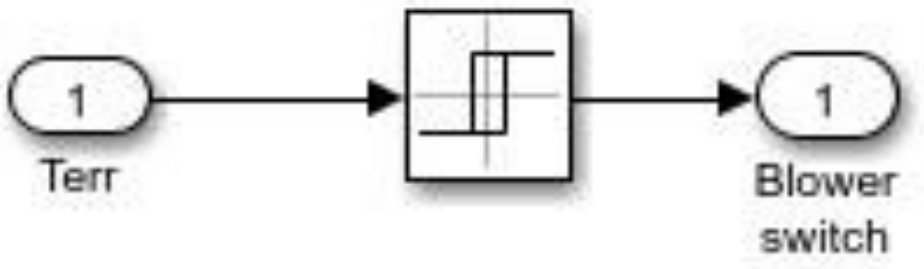




In this physical model it shows the whole temperature control system where, we set the desired temperature for the house. The thermostat make the range of temperature for the heater and the heater then changes the temperature of the house to our desired temperature taking in consideration the outdoor temperature.

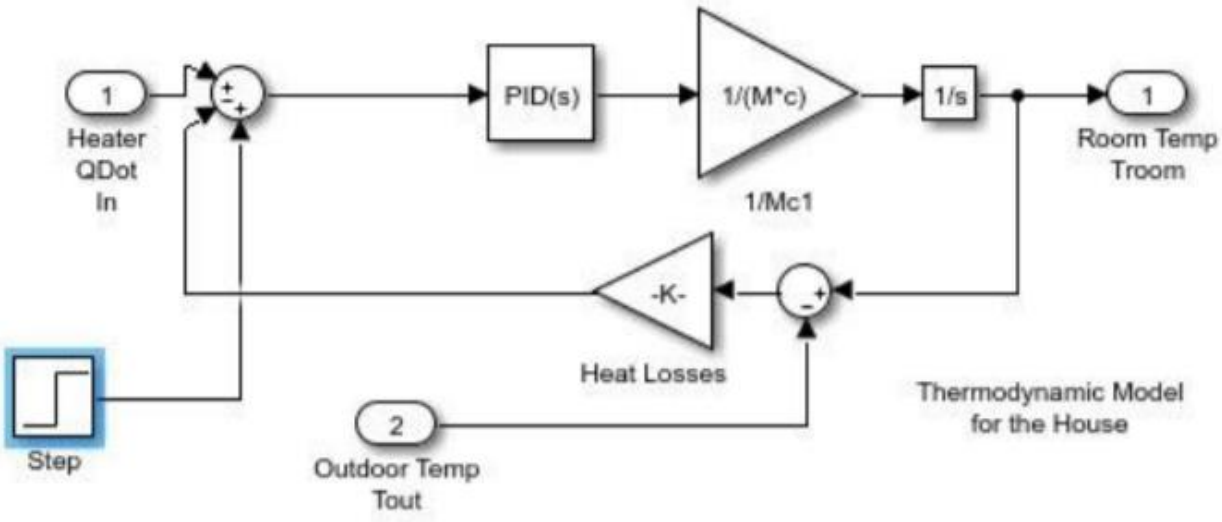
```
r2d = 180/pi;
% ------------------------------
% Define the house geometry
% ------------------------------
lenHouse = 30;
widHouse = 10;
htHouse = 4;
pitRoof = 40/r2d;
% c = cp of air (273 K) = 1005.4 J/kg-K
c = 1005.4;
% ------------------------------
% The air exiting the heater has a constant temperature which is a heater
% property. THeater = 50 deg C
THeater = 50;
% Air flow rate Mdot = 1 kg/sec = 3600 kg/hr
Mdot = 3600;  % hour is the time unit
% ------------------------------
% Determine total internal air mass = M
% ------------------------------
% Density of air at sea level = 1.2250 kg/m^3
densAir = 1.2250;
M = (lenHouse*widHouse*htHouse+tan(pitRoof)*widHouse*lenHouse)*densAir;
% TinIC = initial indoor temperature = 20 deg C
TinIC = 20;
```

Thermostat:



Thermostat Subsystem

House:



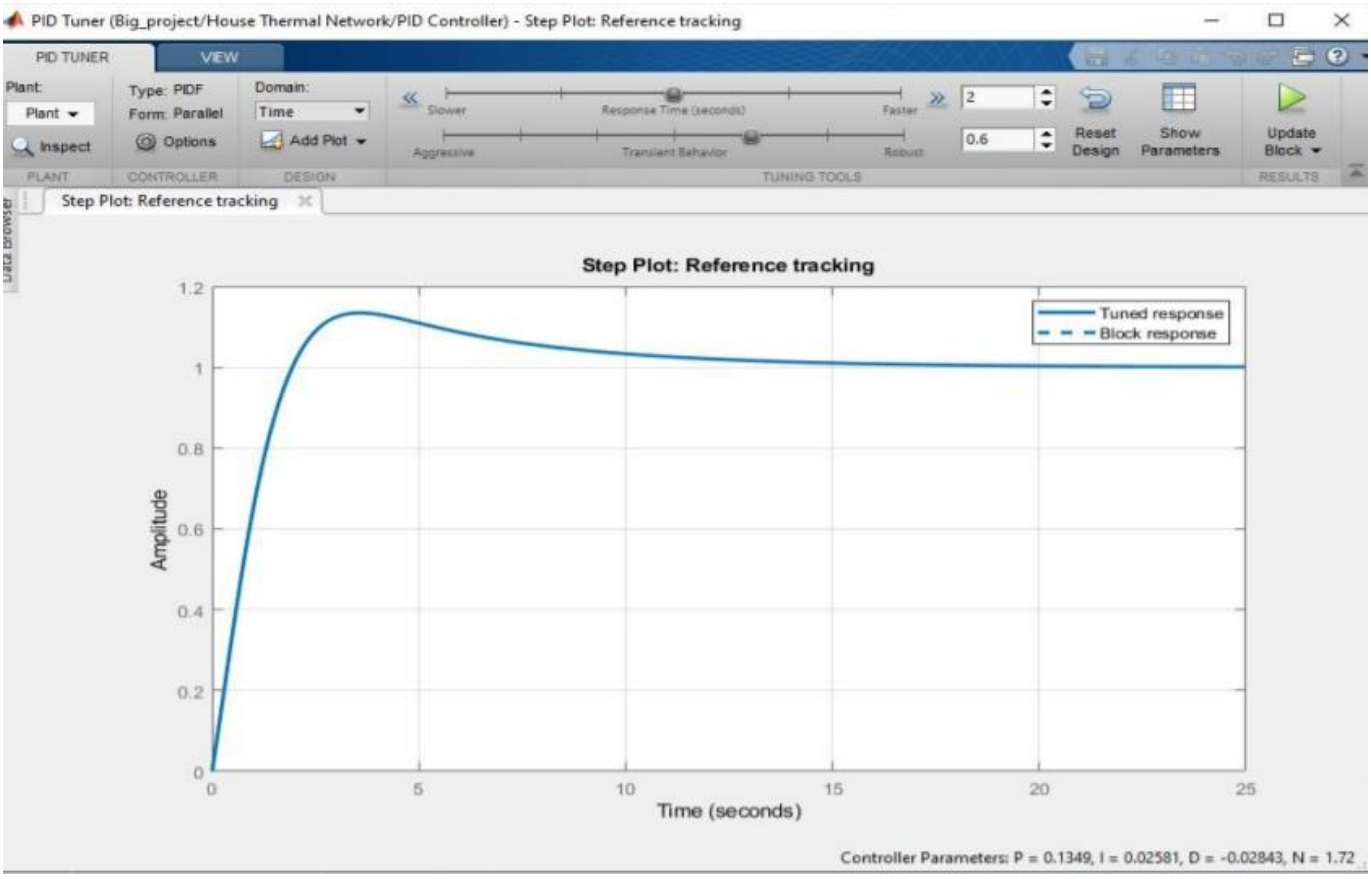
Equation 2

$$\left(\frac{dQ}{dt}\right)_{losses} = \frac{T_{room} - T_{out}}{R_{eq}}$$

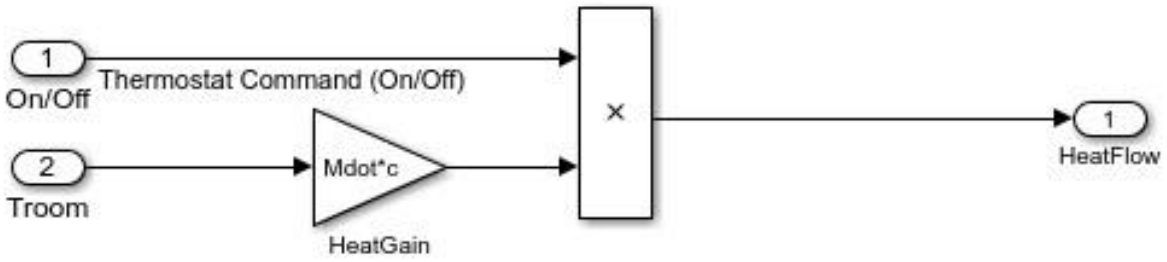
$$\frac{dT_{room}}{dt} = \frac{1}{M_{air} \cdot c} \cdot \left(\frac{dQ_{heater}}{dt} - \frac{dQ_{losses}}{dt}\right)$$

$M_{air} =$ mass of air inside the house

$R_{eq} =$ equivalent thermal resistance of the house



Heater:



Heater

The Heater subsystem models a constant air flow rate, `Mdot`, which is specifi
blows hot air at temperature THeater (50 °C = 122 °F by default) at a constan

Equation 1

$$\frac{dQ}{dt} = (T_{heater} - T_{room}) \cdot Mdot \cdot c$$

$\frac{dQ}{dt} =$ heat flow from the heater into the room

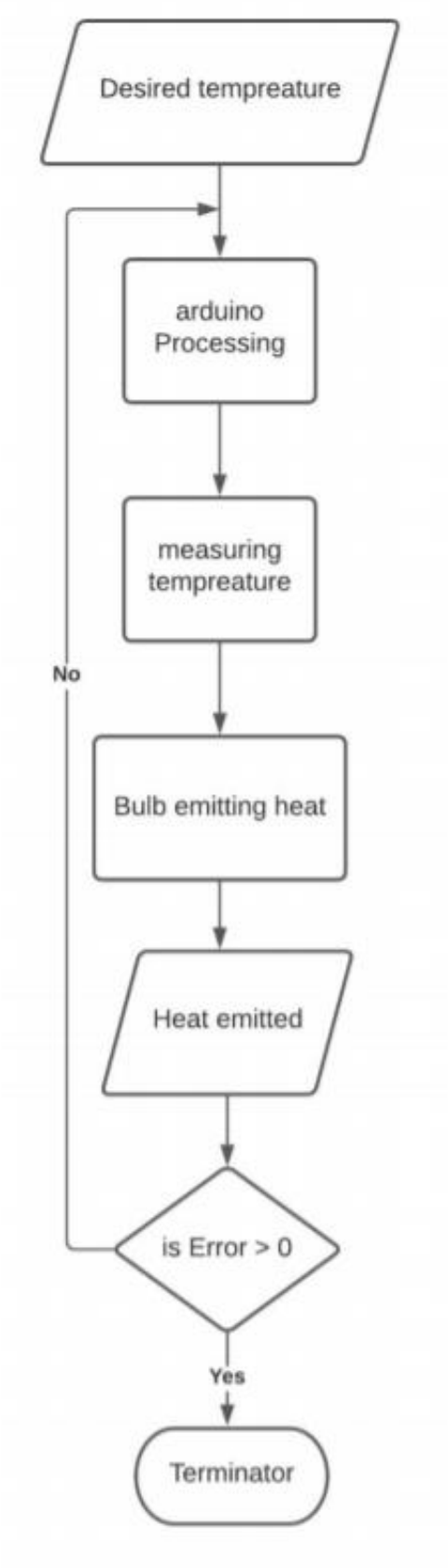
$c =$ heat capacity of air at constant pressure

$Mdot =$ air mass flow rate through heater (kg/hr)

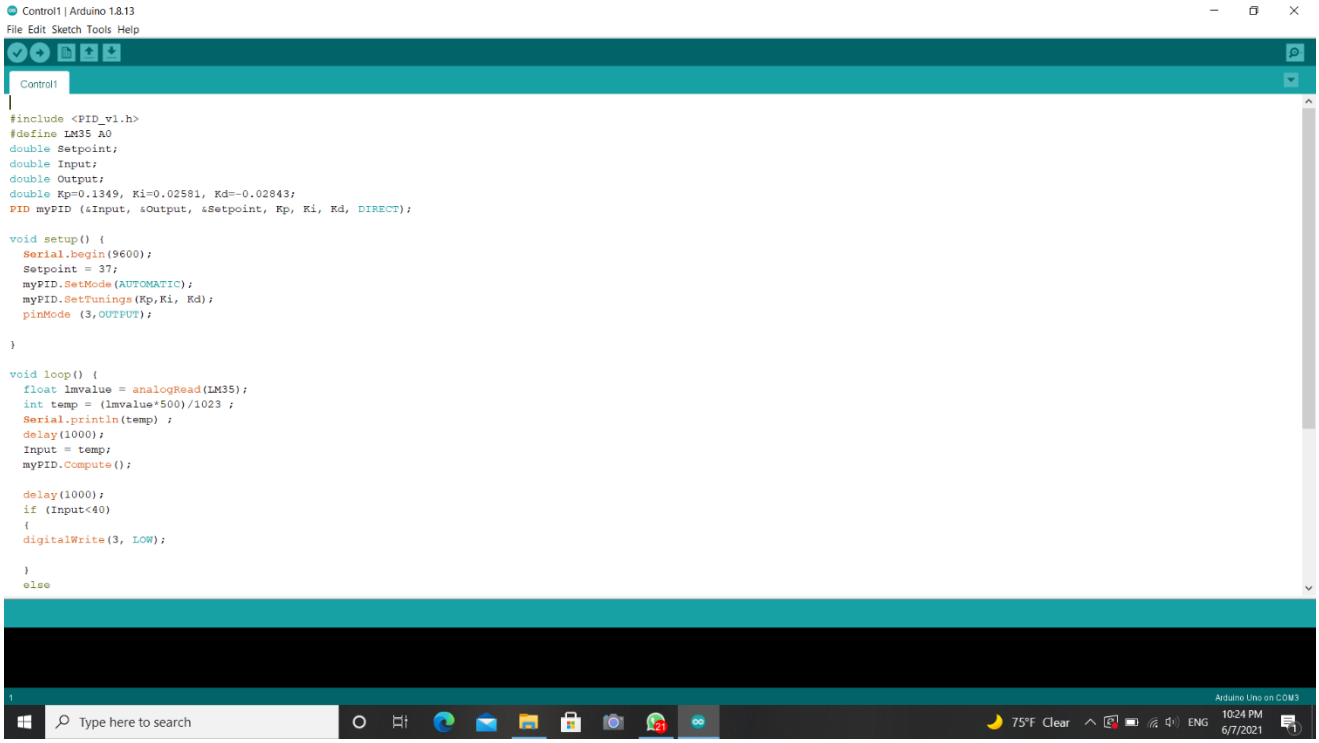
$T_{heater} =$ temperature of hot air from heater

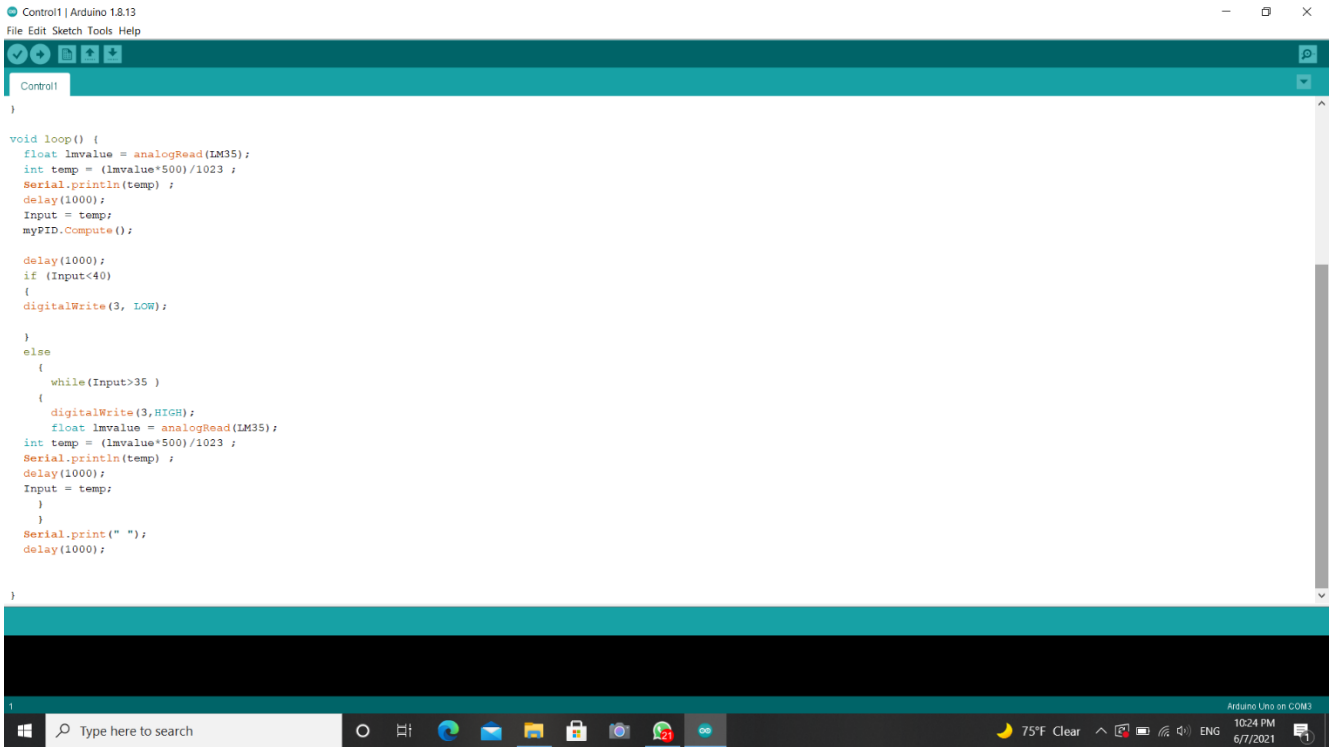
$T_{room} =$ current room air temperature

Flowchart:


Desired tempreature
arduino Processing
measuring tempreature
No
Bulb emitting heat
Heat emitted
is Error > 0
Yes
Terminator

Arduino code:





Real project: