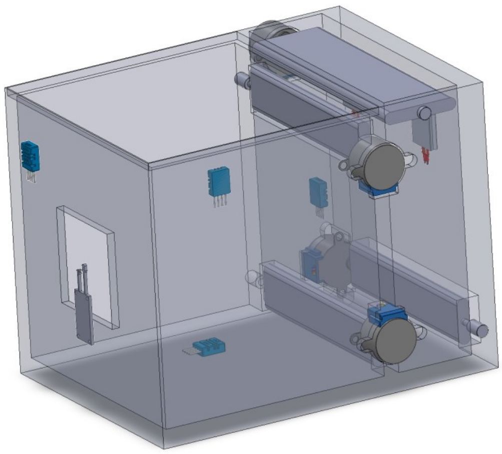
Control Sensing of a Solar House

Course Project - Control and Sensor Technology

Aarhus University



Antonio Karlovic - 202102198
Adam Kuryla - 202103839
Petar Gavran - 202102200

Supervisor:
Xuping Zhang
Uffe Jakobsen

January 2022

Contents

List of Figures

1 Introduction

This project was made for the Control and Sensor Technology course taught by professor Xuping Zhang in the fall semester 2021., under supervision of Uffe Jakobsen and professor Xuping Zhang itself. The focus of the project was on proving that system of solar house works and designing the control and sensing system for a solar house.

1.1 Background and Motivation

Cooling and heating are a significant part of the world's energy consumption. To reduce the energy input, required passive or semi passive solutions may be useful. Such a system can be solar chimneys and with some control, they can be made to keep a comfortable environment inside a house and use solar energy for both heating and cooling a room.

Energy demand is continuously increasing with the rapid rise of both population and new technologies. To reduce the usage of fossil fuels, one of the best ways is to reduce their usages in buildings. This is because buildings are responsible for over 40% energy usage in the whole world, mainly for heating, cooling, providing electricity and air conditioning. Furthermore, renewable energy is environmental friendly with much less greenhouse gas emission. Therefore, many renewable energy systems have been utilized in buildings, such as solar-based, ground source-based energy systems, and daylighting system.

Solar houses are based on solar chimneys. Under the fact that studies have usually ignored the influences of room configuration, wall solar chimney under both cooling and heating modes were analysed theoretically. In this project it will be attempted to obtain a physical model based on previously mentioned theoretical models.

1.2 Objectives

The objective of the project is to apply the theory behind the Control and Sensor Technology course. By applying the methods that were given in the course it is possible to design a controller that can control the heaters in the chimney so we can achieve desired temperature in the house. We will try to design a controller for 2 modes - heating and cooling of house. Since the main objective is to design a model of solar house, proving the concept of solar chimney is made parallel with the project.

1.3 Outline of the report

As an outline of this project report, it will be shown how to design a controller for temperature in the house, which means that it will be shown how to construct a dynamic model based on which a controller is created. Analysis of control, simulation, obtaining system stability should also be demonstrated. In the experimental part, testing of the operation of the solar house during cooling and heating will be shown and compared with expected results obtained in theoretical way. Use of sensors for sensing will be vital part of project. Generally, it will be shown how to use all the theory that is learned during the course.

2 | Problem formulation

Project aims to design the control and sensing system for a solar house. System must be tested by testing/experiment made on physical model.

List of sub-problems is established as follows:

- Design, production, and implementation of physical model
  - Creating the model of the house with 3D printing or wood
- Creating control system – control of 2 heaters with Arduino, with purpose of creating two possible desired states, either cooling or heating of the house
  - Establish control goals
  - Identify variables to be controlled
  - Establish system configuration
  - Obtain a model
- Choosing the sensors
  - Temperature sensor setup
  - Air flow sensor setup
- Testing followed by validating and interpreting the data
  - Measuring the outside and inside temperature before and after the cooling/heating system is applied

2.1 Limitation

Instead of using sunlight and a full-scale room, model made from 3D printing/thick cardboard/wooden structure for the building and a metal plate with a power resistor to simulate the thermal energy from the sun will be used.

3 | Dynamic Model

3.1 Introduction

One of the most challenging parts in most of control projects is to derive the dynamic model of the system, which therefore can be used to derive the control system.

3.2 Dynamic Model Derivation

3.2.1 Definition of inputs and outputs

In order to design the dynamic model for system it is necessary to define inputs and outputs of the system. As stated in the requirements the final temperature must be within the range of approximately 5°C to referenced temperature in the room. As input the referenced temperature is defined, where output temperature is sensed by sensors.

3.2.2 Thermal System

It is well known that to derive the transfer function for thermal system it is necessary to understand the principle how it works. The project includes thermal system since it involves transfer of heat between one place to another. The heat is transferred from solar energy to the air in the room. There are three different types of energy transfers 3.1:

- Conduction – transfer of heat energy from plate chimney to air in chimney [10] [11]
- Convection – transfer of heat through the air by molecular motion
- Radiation – transfer of energy between sun and heating plate in chimney

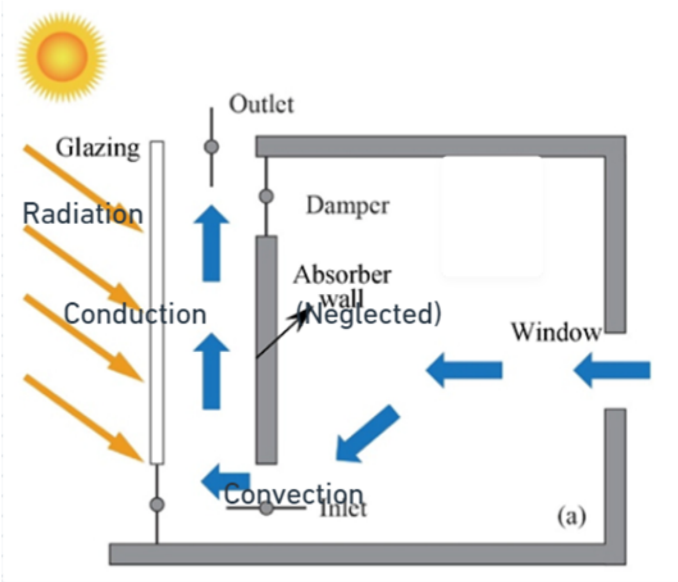


Figure 3.1: Thermal heat transfer used in the project

One way of defining the thermal model of heating/cooling system of the house is to deduce it from thermal resistance and thermal capacitance and treat heat flow in the same way as electric current flow [9]. In this case temperature change can be defined as voltage change and heat flow as current. Therefore, the thermal circuit can be defined for the project as it can be seen in 3.2.

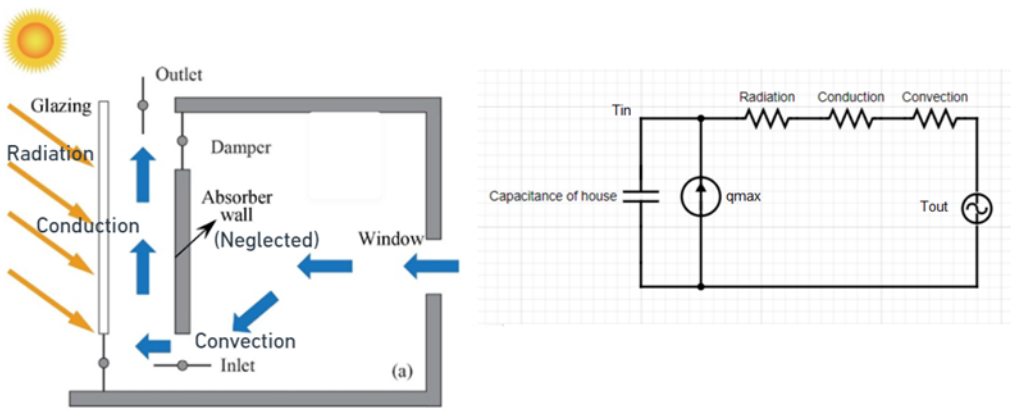


Figure 3.2: Thermal circuit for the system

3.3 Dynamic Analyses

One designing the heating system for the house must define the energy balance for the house system. In the simplest way, the equilibrium equation would look like:

$$Energystored + EnergyLoss = HeatInput \tag{3.1}$$

Considering the house with capacitance of C and conductance between inside and outside of Ut, where Tin is inside temperature and Tout is outside temperature it is possible to derive the final equation [8].

$$C * \frac{dTin}{dt} + Ut * (Tin - Tout) = q \tag{3.2}$$

where q = auxiliary heating or cooling.

Transforming the equation into Laplace, the outcome is achieved:

$$s * C * T_{in}(s) + Ut * T_{in}(s) - Ut * T_{out}(s) = q(s) \tag{3.3}$$

Rearranging stated below equation is achieved:

$$\Delta T_{in}(s) = \frac{Ut}{(C * s + Ut)} * T_{out}(s) + \frac{1}{(C * s + Ut)} * q(s) \tag{3.4}$$

Equation involves two transfer functions which include as first effect of change of Tout to Tin and second effect of change of q to the Tin.

The thermal system can be treated in the same way as RC circuit since it defines thermal capacitance and thermal resistance (C and R). Therefore, it is possible to define a time constant for the system in similar way.

$$\tau = R * C = \frac{C}{Ut} \tag{3.5}$$

Creating:

$$\Delta T_{in}(s) = \frac{1}{(\tau * s + 1)} * T_{out}(s) + \frac{R}{(\tau * s + 1)} * q(s) \tag{3.6}$$

The next step is to define the equation for both thermal capacitance and resistance. From theory thermal capacitance is stated as:

$$C * \frac{d(T_{in} - T_{out})}{dt} = q_{in} - q_{out} \tag{3.7}$$

The rate of change of internal energy of the system is equal to the rate of change of temperature multiplied with capacitance. Applying Laplace transform:

$$C = \frac{q_{in}(s) - q_{out}(s)}{s * (T_{in}(s) - T_{out}(s))} \tag{3.8}$$

Formula can be also rewritten in the form of:

$$C = \frac{(R * q_{max}(s) - \Delta T(s)}{\Delta T(s))} \tag{3.9}$$

where q_{max} = maximum heat distributed to the system from chimney.
$\Delta T(s)$=temperature difference between temperature out and in.

Thermal resistance from the other side is defined as ratio of change in temperature to change in the heat flow.

$$R = \frac{d(T_{in} - T_{out})}{d(q_{in} - q_{out})} \tag{3.10}$$

However, resistance can be simplified and adjusted for different ways of heat flow. Identifying the systems types of flow, the resistance can be summarized:

$$MainR = RadiationR + ConvectionR + ConductionR \tag{3.11}$$

Which can give and output in the form:

$$R = \frac{1}{4 * K_r * \Delta T_{rad}^3} + \frac{1}{H * A} + \frac{1}{K_c} \tag{3.12}$$

where K_r = radiation coefficient.
ΔT_{rad}=effective temperature difference of the emitter and receiver.
A = area of chimney's plate.
H = Convection coefficient.
K_c=conduction coefficient

Thermal resistance can be treated as constant as long as effective temperature difference is not changed during the time.

4 Control Design, Analysis, and Simulation

4.1 Introduction

In order to control the system it is mandatory to establish a transfer function and its model. Since the dynamic model is defined, it makes it possible to estimate appropriate transfer function and design specifications related to it.

4.2 Design Analysis

Dynamic model obtained in the previous chapter can be right now used with its physical properties of the system to extract transfer function of the plant.

The project mostly focuses on the impact of the distributed heat to the temperature in the house, therefore neglecting the relation between temperature outside and inside of house (no conduction through the walls of the house) equation can be simplified and rewritten.

$$T_{in}(s) = \frac{R}{RC * s + 1} * q(s) \tag{4.1}$$

System can include on/off cycle, cycling between 0 and q_{max}, determining the response of the system to temperature swings. This can be done by applying the sine wave as input, then by adjustment of frequency, one can adjust frequency of response of the system.

Auxiliary heating or cooling depends on the frequency of the system to work. Therefore, considering cycles and frequency it can be estimated by the equation:

$$q_{aux}(t) = \frac{q_{max}}{2} * sin(\frac{2 * \pi}{P} * t) + \frac{q_{max}}{2} \tag{4.2}$$

Disturbance is another factor that has to be considered, most of the time it is obtained from experimental data according to particular case. The main reason of the thermal disturbance is the heat loss due to conduction through the walls. However, since the experimental set-up of the project does not represent the real case scenario, data would not be correct, therefore disturbance must be estimated based on internet research. To simplify it, it is considered to be type 0 transfer function, which would be represented as:

$$Td(s) = \frac{1}{30 * s + 1} \tag{4.3}$$

Feedback of the system includes temperature sensor (or multiple sensors) which measures temperature in the room. It is important part of the control system due to its measurements temperature can be adjusted. That is why it is important to define its transfer function. Most of time for simplification first order transfer function is used to define model. In practice, it is not recommended to use models of order greater than 4 because usually it does not improve the accuracy of approximation of real sensor dynamic parameters and involves amplification of noise when correction systems or algorithms are applied. The simplest transfer function of sensor can be represented by:

$$G(s) = \frac{1}{N_t * s + 1} \tag{4.4}$$

Where N_t can be defined as time constant and is calculated with the use of equation defined by parameters of the manufactured sensor.

$$N_t = \frac{m * c_p}{\alpha * F} \tag{4.5}$$

Where m is defined as cylindrical mass of the sensor, c_p as specific heat, α as heat transfer coefficient and F as heat transfer function.

4.3 Simplified Control Design

The next step is control design of the system. Previously described idea of the model must be analyzed and therefore adjusted with usage of the controller, depending on final outcome it could be either P, PI, PID or lead/lag controller.

Due to complexity of the system, it is convenient to simplify the transfer function of it to minimum. This is due to experimental limitations and various environments in which thermal system had to work (no disturbance taken into consideration). Therefore, some fixed values were taken to establish transfer function of the plant and also estimated and simplified transfer function of sensor was considered 4.1.

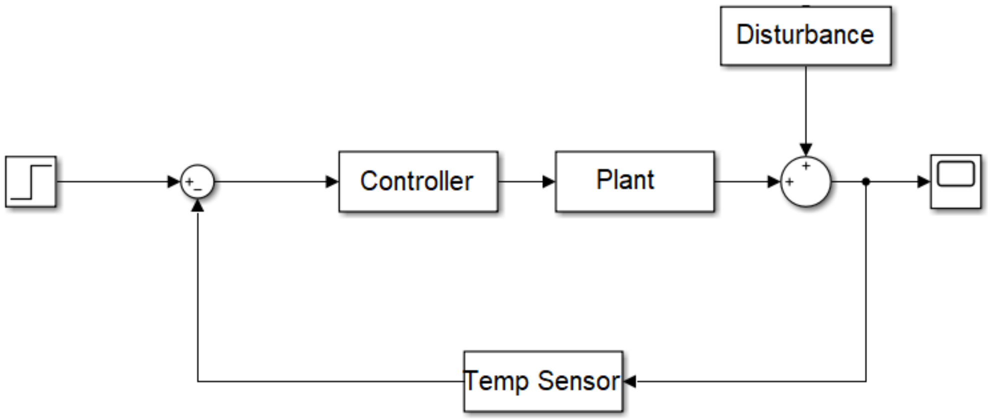


Figure 4.1: Block diagram

System contains closed-loop transfer function, since it needs feedback for adjustment of fluctuation of the temperature.

After block diagram simplification transfer function can be presented as 4.2:

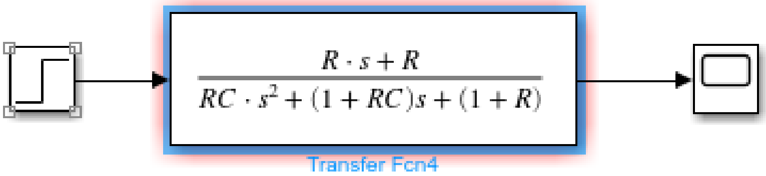


Figure 4.2: Simplified transfer function

System has 2 poles and 1 zero. Transfer function can be considered as type 0. With stated variables transfer function can be analyzed in the matlab as well as adjusted with controller 4.3.

```
s=tf('s');
Kr=0.9;
H=0.9;
A=4;
Kc=0.9;
deltaT=10;
P=1;
Trad=40;
qmax=700
R=1/(4*Kr*Trad^3)+1/(H*A)+1/Kc;
C=(R*qmax-deltaT)/deltaT;
G1=R/(R*C*s+1)
H=1/(s+1);
sys2=feedback(G1,H)
controlSystemDesigner(sys2);
```

Figure 4.3: Simplified transfer function

As shown below system is stable, since all of the poles are on the left hand side. What is more it has infinite gain margin and phase margin, meaning that the loop is stable. There is also no overshoot, meaning that the system is probably overdamped. However, system has very huge steady state error, more than 60%,that is why there is a need for controller 4.4.

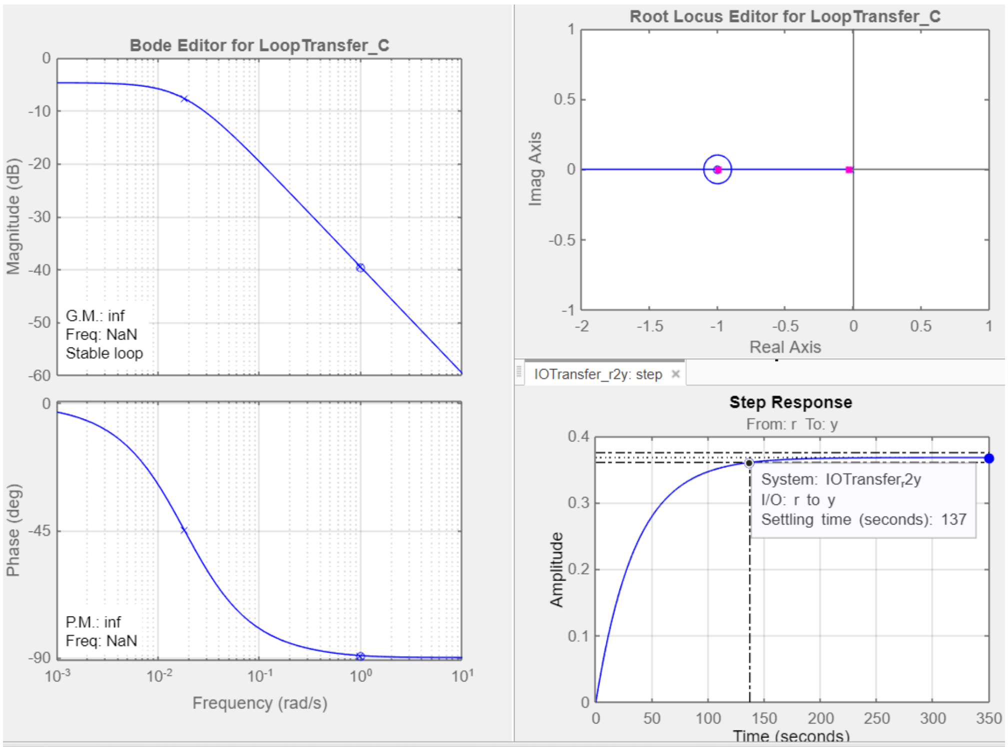


Figure 4.4: Properties of simplified response

PID can be set up in multiple ways, however Ziegler-Nichols method is very often good choice for starting the design. Method worked perfectly, however it produced overshoot, which could be removed by manual adjustment. In case of solar chimney project PI controller is good choice 4.5.

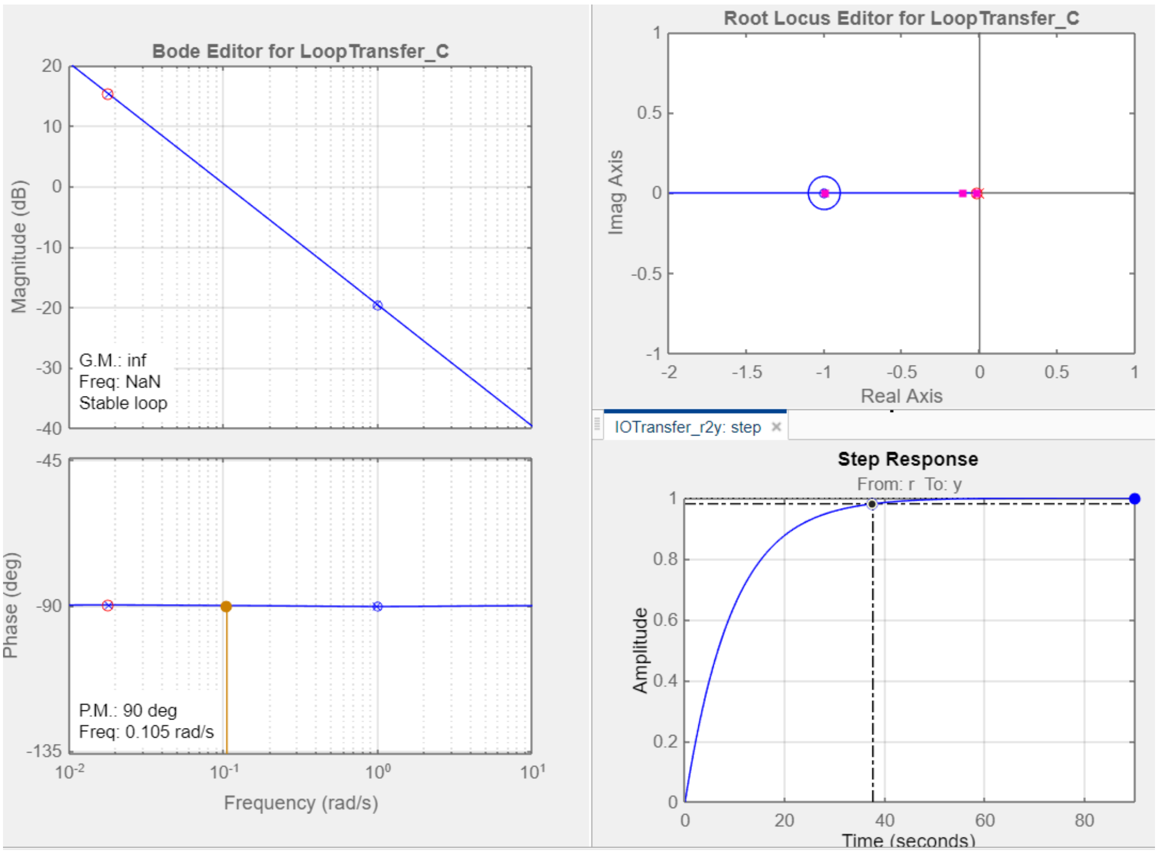


Figure 4.5: Response with PI controller Kp-10.04 , Ki-0.1794

As it can be seen at the figure above, phase margin was reduced, however it is still huge, additionally settling time was reduced and system has faster response time.

4.4 Advanced Control Design

In most cases the simplified model could not work, therefore it seems convenient to check it with more advanced input type as well as disturbances. That is why in this case more advanced transfer function block diagram has to be created as in figure 4.6.

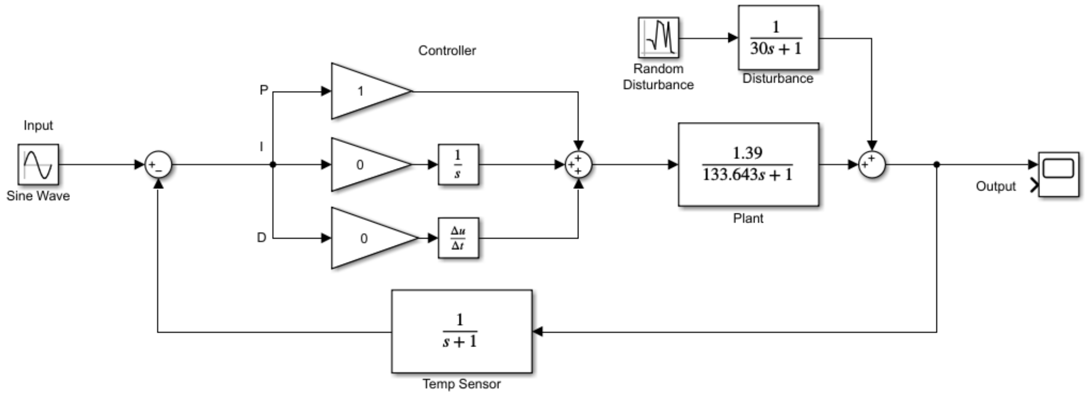


Figure 4.6: Block diagram of the control

Depending on variables and input type, system might seem unstable, however proper adjustment of the controller, mainly proportional gain may reduce problem as in figure 4.7.

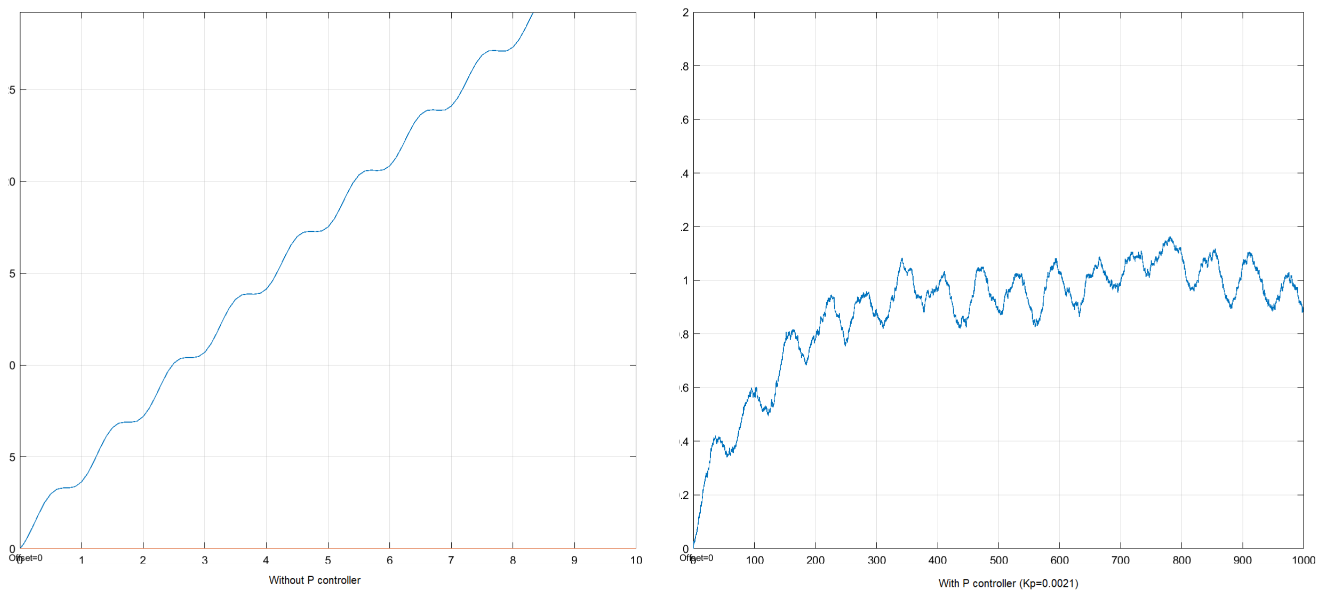


Figure 4.7: Comparison of response without P and with P controller

On the left side on the figure above it can be seen that system might be unstable (value of response crosses 1 and goes to infinity), this is due to gain problem, which is too big for the system. That is why proportional control can adjust it. When applying proper value of proportional gain, system seems oscillating around value 1. Fluctuations are due to random disturbances and also comes from on/off mode of the system (sin wave as input, in this case frequency is 0.1). The final response is very close to the output value, mainly by small steady-state error, since functions oscillates around value of 1. There is also no overshoot, meaning that system is either overdamped or close to damped state.

In general system can be controlled only by the P controlled, since integral and derivative components do not impact the response in major way. However, they reduce disturbance and also create "smoother" fluctuations, meaning that system is more stable and more error-resistant, which can be seen in figure 4.8.

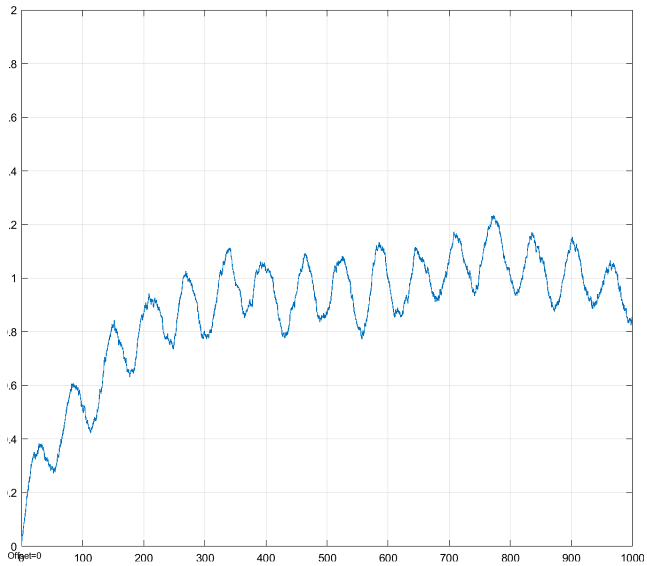


Figure 4.8: Response with PID controller Kp-0.0021, Ki-10^{-8},Kd$-$0.025

4.5 Summary

Control design for solar chimney can be done in multiple ways. System though is stable in both cases (simplified and complex version). The best choice as controller is PI, because it provides stability and reduction of steady state error. However, P controller also can be used maintaining less accurate response, but still acceptable.

5 | Controller Setup

Unlike the previous section, this section will focus on physical setup of the controller. Each component will be described in detail together with the the idea behind it. In the end, complete assembly will be presented.

5.1 PID Controller

PID controller [15] stands for proportional–integral–derivative controller. PID control uses closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible. It is widely used as a temperature controller. As the name say, it consists of three parts (proportional–integral–derivative). It is based on the control of the input (voltage) in order to manipulate output (temperature). After initial heating, temperature is observed and at the instant the temperature rises higher than the setpoint temperature, controller will lower input voltage with the aim of lower temperature. On the contrary, as soon as temperature falls below setpoint value, controller will conduct more power to the heater. The described procedure is the proportional part of the controller since the current will be proportional to the error (error = setpoint – measured value). In some cases, using P-controller alone is sufficient, but there is a good chance that the system will never be stable because it requires an error to generate the proportional response. To compensate for the mentioned side effects, PD-controller is introduced. The derivative part will react to the speed of temperature change - The more rapid the change, the greater the controlling or damping effect. In the end, by adding term I (integral) which accounts for past values of the errors and integrates them over time, PID controller as in figure 5.1 is completed. Crucial part of any PID controller is feedback system.

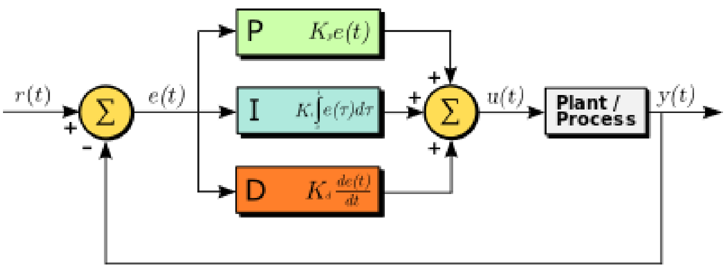


Figure 5.1: PID Diagram

5.2 Feedback system

Feedback system has main objective is to obtain, and transfer measured output value in order to be compared to setpoint value. In the case of temperature controller as used in this project, special kind of thermometer has to be used.

5.2.1 K type thermocouple + MAX6675

Type K Thermocouple 5.2 provides widest operating temperature range. It consist of positive leg which is non-magnetic and negative leg which is magnetic. In K Type Thermocouple traditional base metal is used due to which it can work at high temperature and can provide widest operating temperature range.

Figure 5.2: K Type Sensor

The MAX6675 5.3 performs cold-junction compensation and digitizes the signal from a type-K thermocouple. This converter resolves temperatures to 0.25°C, allows readings as high as +1024°C, and exhibits thermocouple accuracy of 8LSBs for temperatures ranging from 0°C to +700°C. Together this two elements form a feedback system of a controller.

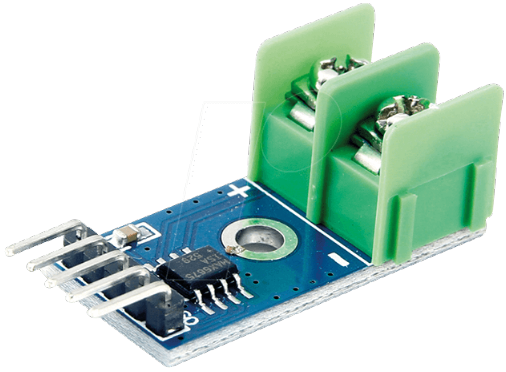


Figure 5.3: MAX6675

5.3 Control

As a controller, Arduino was used. Arduino is a single-board microcontroller. Besides the physical part of PID controller there is also a program made in Arduino IDE. Program can be found in Appendix A. First, most important physical elements will be described followed by a short program description.

5.3.1 MOSFET

MOSFET [14] stands for metal–oxide–semiconductor field-effect transistor. MOSFET transistor is a semiconductor device that is widely used for switching purposes and for the amplification of electronic signals in electronic devices. In our project MOSFET was used for PWM [16] (pulse-width modulation). PWM is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts, in other word, MOSFET will control the voltage that goes to heating element. MOSFET used is IRF830 (Figure 5.4 and Appendix E), it is N-channel MOSFET.

A N-Channel MOSFET is a type of MOSFET in which the channel of the MOSFET is composed of most electrons as current carriers. When the MOSFET is activated and is on, the majority of the current flowing are electrons moving through the channel. MOSFET has three pins, marked as G (ground), S (source) and D (drain) as in 5.4. Voltage is controlled between source and drain pins with PWM being achieved through dissipation of heat at the drain.

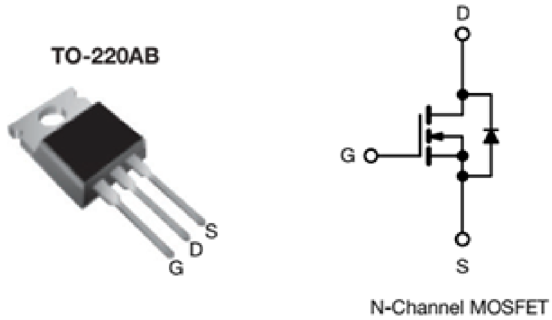


Figure 5.4: IRF830 MOSFET

5.3.2 S8050

S8050 is a NPN transistor hence the collector and emitter will be left open (Reverse biased) when the base pin is held at ground and will be closed (Forward biased) when a signal is provided to base pin. It has a maximum gain value of 400, this value determines the amplification capacity of the transistor normally S8050. In the project it is used as BJT to activate the N channel gate, so the MOSFET is activated with a LOW value in this case. It has three pins, marked 1 (collector), 2 (base), 3 (emitter) as in figure 5.5.

Figure 5.5: S8050 NPN Transistor

5.4 Final schematic

Figure 5.6 represents schematic used in a final assembly of the controller. Once all components are connected properly it is very simple to recognize closed-loop system, with each component playing crucial role in the loop.

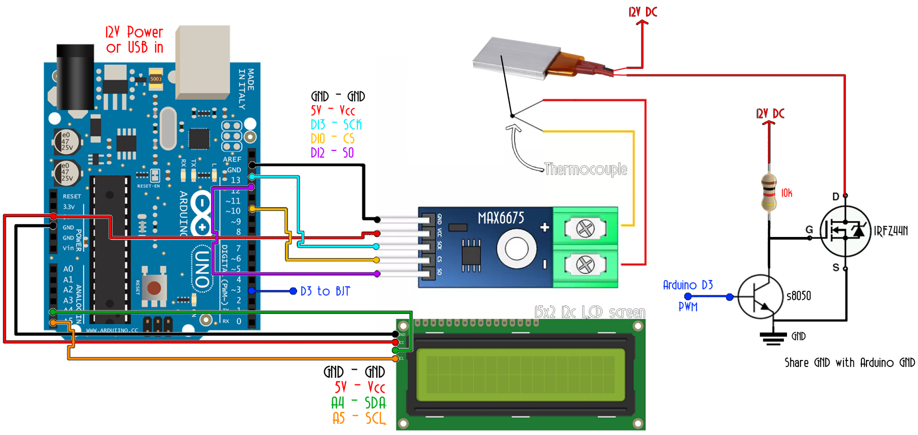


Figure 5.6: Controller wiring [9]

5.5 Arduino Code

The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. Code in A starts with standard routines in Arduino code, such as including libraries and deciding pin connections. Following, variables are given initial values, inside this, most important value is given, named ‘set-temperature’ (see Figure 5.7). ‘Set-temperature’ refers to setpoint value, described before in the project, based on which error is calculated and controller will act upon.

```
//Variables
float temperature_read = 0.0;
float set_temperature = Temperature Outside + 5;
float PID_error = 0;
float previous_error = 0;
float elapsedTime, Time, timePrev;
int PID_value = 0;
```

Figure 5.7: Code segment: Setpoint

Furthermore, in beginning it is possible to find initial Kp, Ki and Kd values, which are calculated in the ‘Control Design, Analysis and Simulation’ section. Main loop of the code starts with reading the temperature of room. Subsequently, error is calculated as “Error = Setpoint Temperature – Measured Temperature”. Next, P, I and D values are calculated with multiplication of the error with PID constants from the beginning. Only notable difference here is that for D value real time has to be obtained to calculate speed change rate. Finally, PWM modulation is conducted by writing PWM signal to the MOSFET on digital pin D3.

```
void loop() {
 // First we read the real value of temperature
  temperature_read = readThermocouple();
  //Next we calculate the error between the setpoint and the real value
    PID_error = set_temperature - temperature_read;
  //Calculate the P value
  PID_p = kp * PID_error;
```

Figure 5.8: Code segment: Error

6 The Model for Experimental Setup and Sensing

6.1 Introduction

The Model for Experimental Setup and Sensing section consists of three parts. First part describes the model of house which will be used for experimental setup. The main goal was to make a model of a house with a solar chimney that could prove the theoretical operation of a solar house. Due to limitations, the solar house was made on a 3D printer in a scaled version, paying attention to the concept of the rooms.

The second part describes in detail the sensors used on the solar house model and also the methods and reason for their use.

Third part shows how the physical model for experimental setup is obtained.

6.2 The Model for Experimental Setup

The experimental setup model consists of 5 parts. The first part is a 3D print of the house model, which was created according to the theoretical model obtained as part of the task description, can be seen in 6.1 figure. Also with the 3D printed elements for opening and closing the inlets on house, depending on whether the cooling method or the heating method should be used. The second part are the stepper motors used to control the components for opening and closing the inlets, which is controlled by the Arduino Mega. In order to imitate solar heating, two heating elements were used as the next part, to rise the temperature in the chimney. Vital part of model was use of sensors, also controlled with Arduino Mega, which enable the monitoring of the operation of the solar house, and through which the proof and verification of the validity of the procedure is obtained. As a last and most important part was Arduino Mega, which controls all the important parts of model for experimental setup.

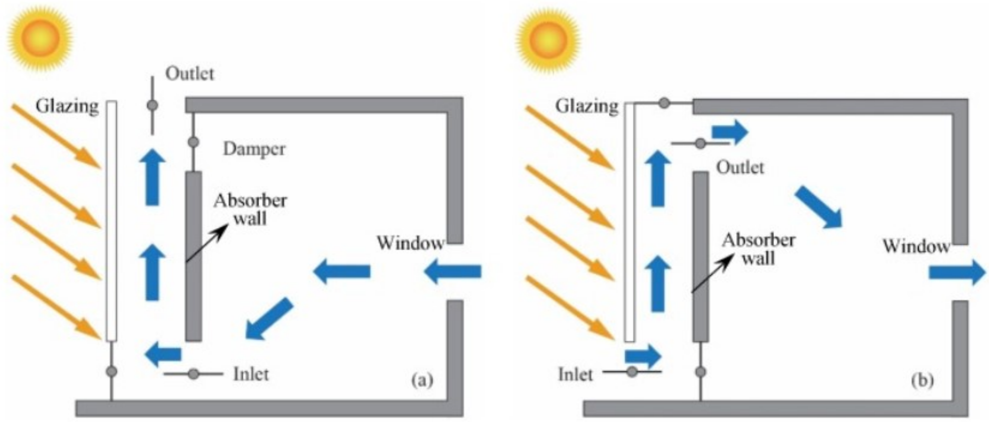


Figure 6.1: Typical wall solar chimney under: (a) cooling mode (b) heating mode [13]

6.2.1 The Model of House

Solar chimney [13] performance is dependent on the airflow rate and its temperature, where theoretical models were developed to predict the performance of two typical types, including fresh-air cooling and fresh-air heating through chimney cavity and room. It is known that the room configuration shows considerable influences on solar chimney performance. Different from the cooling mode, airflow rate under heating mode was found not only dependent on cavity height, but also the opening height of the room. To heat a typical room, fresh-air heating through the cavity shows the highest airflow rate but with the lowest temperature, which can be applied to regularly occupied building under cool weather conditions. Fresh-air heating through the room shows an opposite way, which is suitable for regularly occupied buildings under cold weather conditions. In order of trying to satisfy both methods of heating and cooling as efficient as possible, the 3D printing model shown in figure 6.2 was made.

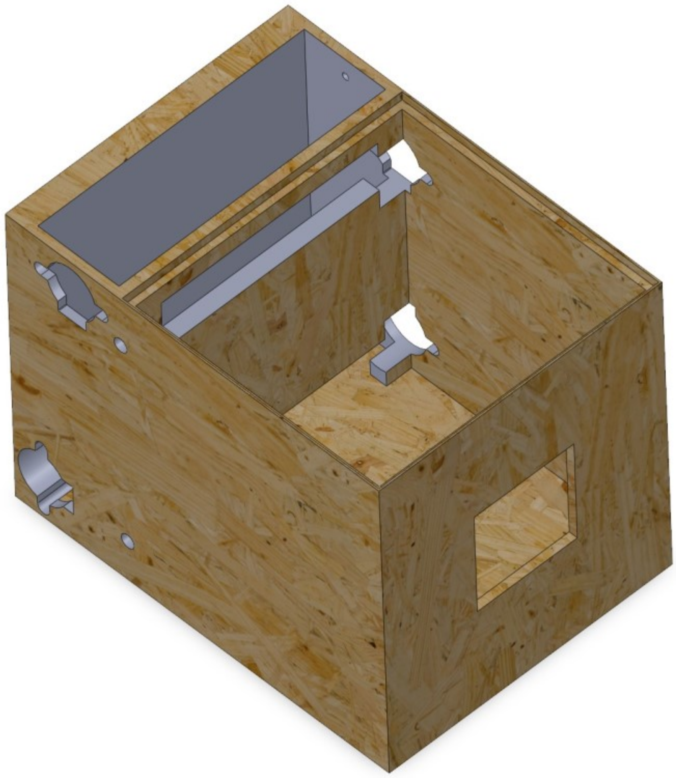


Figure 6.2: The Model of House

Since it is needed, the already mentioned two modes of operation (heating and cooling), the inlets in the house are made according to the theoretical model shown in the figure 6.1. In order to control the opening and closing components of inlets, they are made together with holes for attaching stepper motors that will control each inlet separately so that desired mode of operation of the solar house can be achieved.

A very important part of the house is the free opening,that is the window which is one of the vital parts of the heating and cooling process, because through it the house is supplied with fresh air from the outside and enables the desired operation. All this gathered together can be seen on figure 6.3

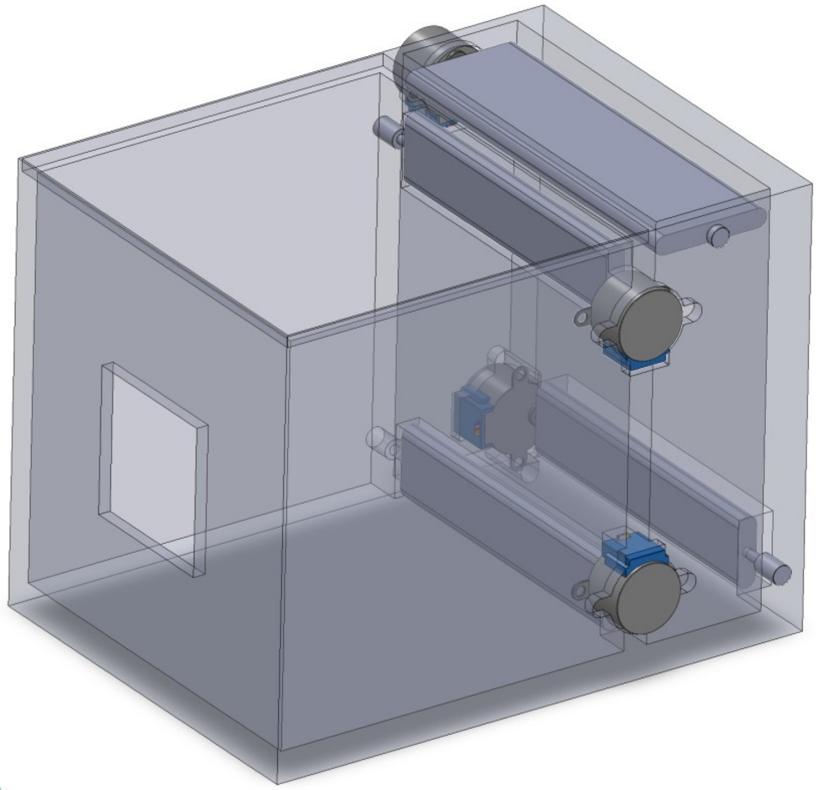


Figure 6.3: The Transparent Model of House

6.2.2 Stepper motor

Stepper motors [4], due to their unique design, can be controlled to a high degree of accuracy without any feedback mechanisms. The shaft of a stepper, mounted with a series of magnets, is controlled by a series of electromagnetic coils that are charged positively and negatively in a specific sequence, precisely moving it forward or backward in small "steps". There are two types of steppers, Unipolars and Bipolars, and it is very important to know which type you are working with.

In our case, stepper motors are needed to control the inlets on the house, made of a light 3D print, which determines mode of operation of the solar house that is used. The requirement is that the two components of the inlets are rotated 90 degrees at the same time, thus letting air flow through them, 2 for heating and 2 for cooling, operating at different moments.This is not a very demanding task and does not require a powerful stepper motor, as the moments are very small, so we opted for the simplest unipolar 28BYJ-48-5V Stepper Motor, compatible with Arduino, which can be seen in figure 6.4.

The model of house has 4 inlets important for performance so there is a need for 4 Stepper Motors 28BYJ-48.

Figure 6.4: 28BYJ-48 Stepper Motor [6]

6.2.3 PTC Heating Plate

Solar chimney is basically a solar air heater, which is vertically or horizontally embedded as a part of wall or roof. Chimney cavity normally consists of a glazing wall for solar penetration and an absorption wall for heat absorption, as shown in 6.1. The air in the chimney cavity is heated under the penetration of solar radiation through the glazing wall and they rise under the thermal buoyancy to enhance the natural ventilation of buildings. Solar chimney is used to promote natural ventilation by taking away indoor heat (cooling mode) or bringing in hot air (heating mode).

Since a scaled-down version of the solar house was made, the way of function of the solar chimney had to be adjusted. In order to imitate the heating of the chimney by sunlight through the glazing wall, two PTC heaters were used to heat the air in the chimney so that the process of heating and cooling the house could be achieved. Two AC/DC 12V thermostatic aluminum ceramic heating plates were used to obtain desired temperature inside of chimney.

6.2.4 Arduino Mega 2560 Rev3

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM out-

Figure 6.5: PTC Heating Element AC-DC 12V Aluminum Shell Ceramic Heating Plate [3]

puts), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

For the need to control the components of the solar house Arduino Mega 2560, shown in figure 6.6, was best suited. It was a needed for controller that has more input/outputs than an Arduino Uno because it needs to control 4 stepper motors as well as all the sensors that will be explained in the next section.

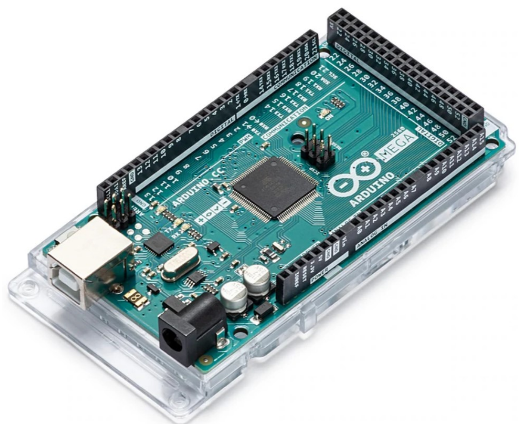


Figure 6.6: Arduino Mega 2560 Rev3 [5]

6.3 Sensor and Sensing

The sensors were one of the most important parts of the solar house model because they are used to monitor temperature inside and outside, which is the foundation around this whole process was made, and thus be able to confirm whether the model is well made and whether we have achieved the process for which it was intended. It is also possible to monitor the flow of air moving through the house during the heating/cooling process.

6.3.1 DHT11–Temperature and Humidity Sensor

The DHT11, which can be seen in figure 6.7, is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers. The sensor can measure temperature from 0°C to 50°C and humidity from 20 to 90% with an accuracy of ±1°C and ±1%.

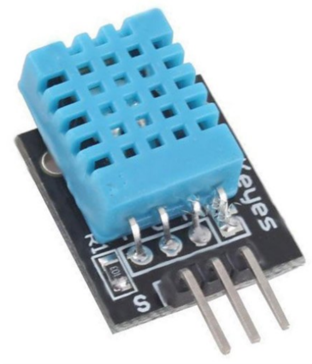


Figure 6.7: DHT11–Temperature and Humidity Sensor [7]

In order to monitor the progress of both processes in the house, it was necessary to use several DHT11 sensors in different places, to be able to confirm that the processes are going properly. The main goal was to achieve a difference between the outside temperature and the temperature in the room of 5°C, both in heating mode and cooling mode. One sensor was placed in the chimney so it can monitor the temperature to which the air inside is heating up, in order to try to keep it within some limits that are realistic in real life. Given the theoretical model of the solar house and knowledge of thermodynamics, it can be concluded that during heating mode, warm air from the chimney should flow through the upper part of the room to the window, so one sensor is placed in the upper part of the room. The cooling mode should also work on the same principle, but this time cold air should flow from the lower part of the room to the chimney, so one sensor is placed in the lower part of the room. As a comparative temperature is the outdoor temperature, so it was necessary to install a sensor to measure outside, according to which we will determine the success of the process.

6.3.2 Wind Sensor Rev.P

The Wind Sensor Rev.P, shown in figure 6.8 is anemometer with an analog output. It is a thermal anemometer based on a traditional technique for measuring wind speed called the "hot-wire" technique. This involves heating an element to a constant temperature and then measuring the electrical power that is required to maintain the heated element at temperature as the wind changes. The wind velocity is proportional to the heat (i.e. power) applied to the sensor.

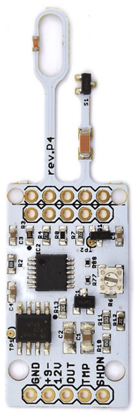


Figure 6.8: Wind Sensor Rev.P [12]

Considering the law of conservation of mass, from the presented model it can be concluded that the only common point for both heating and cooling modes is the window, through which warm air exits during heating or during cooling fresh air enters from the outside. For this reason, the flow of air entering and leaving can be estimated on it. With the help of a wind sensor, the speed of air flow through the window can be measured and since the surface of the window is known, the air flow can be calculated. Thus, it can be proven that the model works because the air flow is the most important for changing the temperature inside the room.

6.4 The Final Model for Experimental Setup

According to the parts explained in the previous sections, the Experimental Setup Model was assembled and designed to look as on the figure 6.9

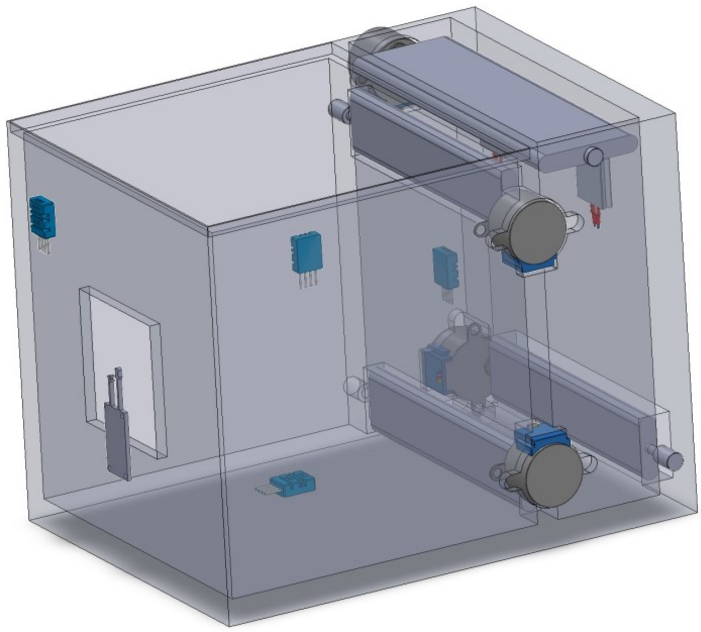


Figure 6.9: The Final Model for Experimental Setup

The physical model of house was made on 3D printer, all components were added so the real scaled physical model was obtained how it can be seen on figure 6.10. Upper part of house (roof), was made of Plexiglas so that inside of house can be seen.

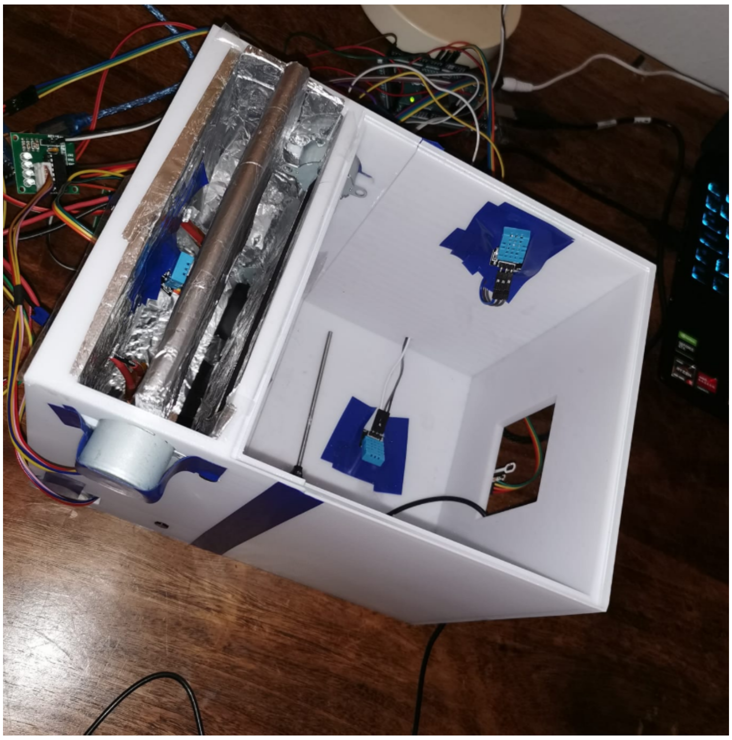


Figure 6.10: The Physical Model for Experimental Setup

6.5 Summary

As mentioned earlier, solar houses were mostly treated in the theoretical part. Thus, according to a theoretical model that would support the process of heating and cooling, this experimental model was realized, on the basis of which functionality will be proven. The model of the house itself is very important because it is necessary to take into account the configuration of the room so that air flow is possible at all, which is the basis for achieving the desired procedure. Each part has a very important role in the proving the process, but perhaps the most important part would be the temperature and flow sensors with which feasibility of process is proven.

7 Testing and Discussion

7.1 Introduction

After completion of physical setup, testing of the theory followed. Basic idea behind is to create real-life conditions of the house and observe the changes, primarily – in temperature. Following chapter will explain in details most important parts of testing procedure, such as requirements, scope of testing and test strategy according to the scope. Before testing two different goals are set. First, to prove that the concept of solar chimney is viable and second to justify controller design and usage in the project. In proving the concept, two possible modes are cooling and heating. Based on the inlets and outlets states desired modes are achieved.

7.2 Testing procedure

As described in the 'Experimental Setup and Sensing' chapter, physical model must be assembled before beginning of the testing. Arduino components must be connected, and Arduino code must be ready. Before testing started, one concept had to be proved, that is that the flow between inlets and outlets is happening, because if the flow is not occurring, testing the concept and controller would not make sense since the difference in the temperature of the room could never occur. Flow was proven by two different tactics. First was to prepare the testing conditions for cooling as described later and create smoke in front of the window of the room, if there was a flow, smoke should go inside the room, and the again outside of the house trough the chimney. Test was successful as it can be seen in the figure 7.1 and in video [2]. Another way was to use the wind sensor at the window. First, value of wind speed was obtained when all inlets/outlets were closed, after Inlet and Outlet 1 were opened, once wind speed increased significantly, test was marked as successful.

7.2.1 Heating mode

Heating mode will be described first. At the beginning of testing all inlets/outlets are closed, this is done in order to achieve starting temperature in chimney faster. Heating elements, which are placed inside the chimney, are energized, and start to warm up gradually. Once the temperature sensors inside the chimney read the value of 50°C, Inlet 1 and Outlet 1 are opening,

Figure 7.1: Smoke Test

allowing the air to go from Inlet 1 to Outlet 1 inside the room because of the pressure difference. K type thermocouple is simultaneously measuring the temperature inside the room. In this case, as a setpoint for the controller, 'outside temperature + 5 °C' is taken. Until the final temperature difference between outside and inside temperature of 5°C is achieved, controller will conduct 100 percent of voltage to the heater. Test is successful if temperature difference is achieved in a reasonable time. Complete procedure is also made in form of flow chart 7.2, where steps follow chronological order of testing.

7.2.2 Cooling mode

Since the majority of the steps from Cooling mode resembles the steps in the already described Heating mode, only the main differences will be mentioned in this section. First and most important difference in the testing procedure is obtaining the setpoint. In the cooling mode setpoint = outside temperature – 5°C. Furthermore, after achieving the starting temperature inside the chimney, Inlet 2 and Outlet 2 are opened instead of Inlet 1 and Outlet 1. Also, notable difference is that in the cooling mode cooling was enhanced by placing the ice-cold can in front of the window.

Figure 7.4 shows the complete assembly of setup.

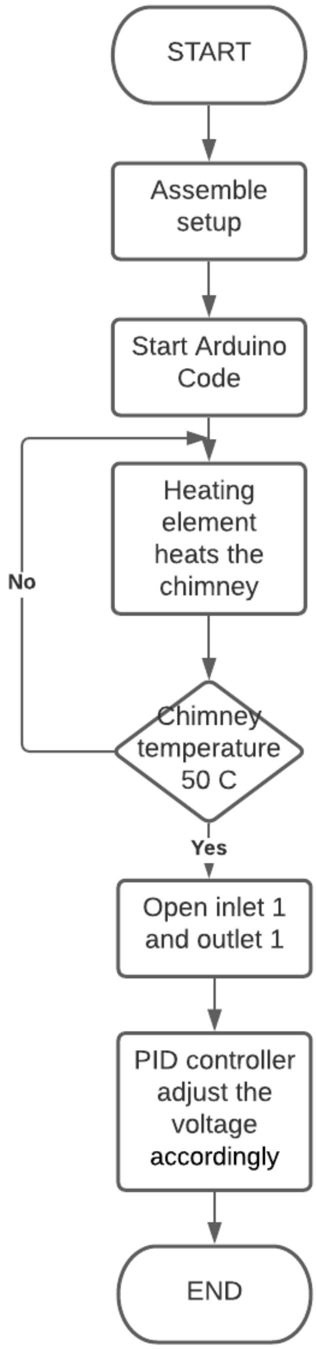


Figure 7.2: Process flow chart

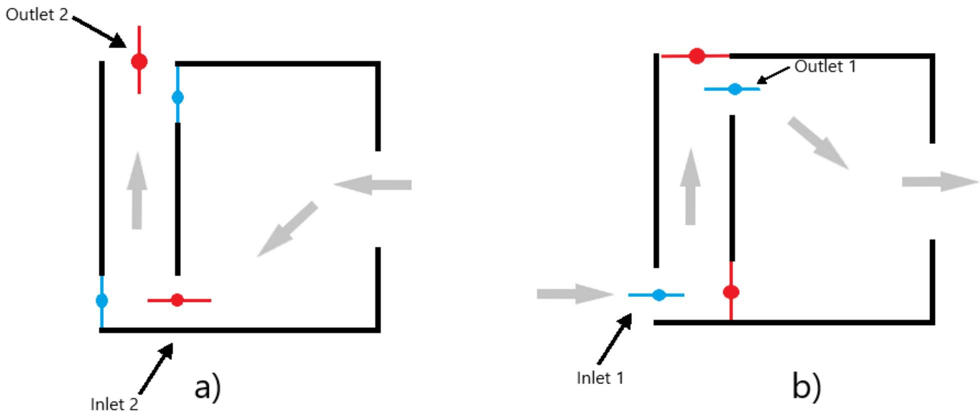


Figure 7.3: Inlets and Outlets a) Cooling mode b) Heating mode

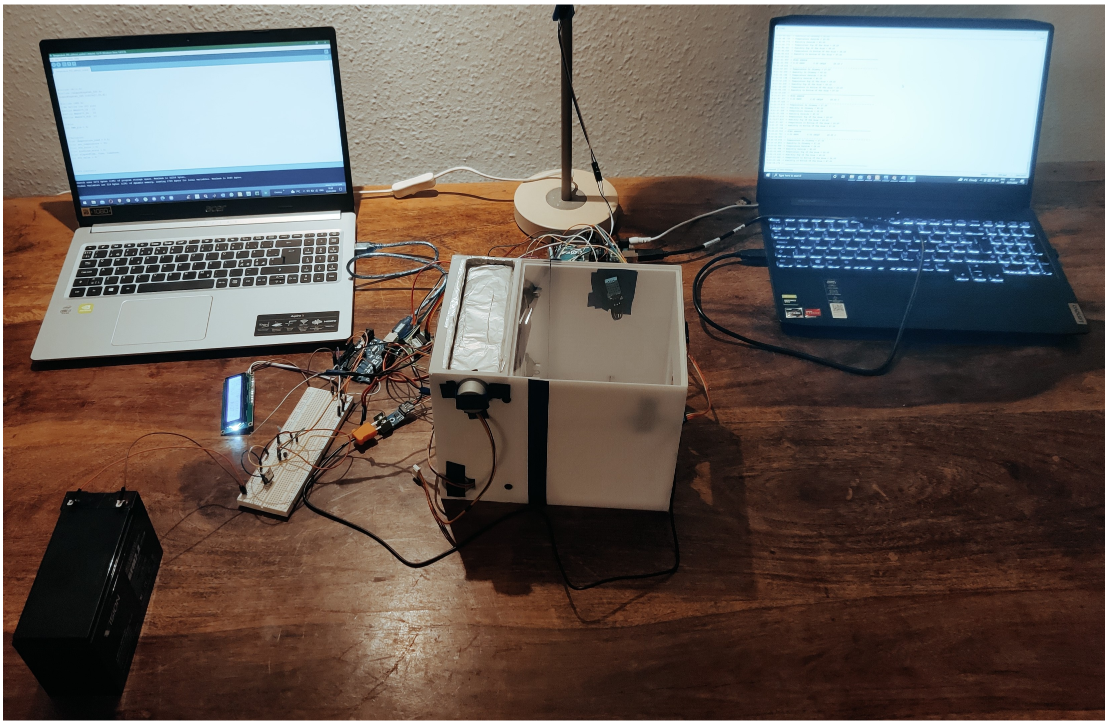


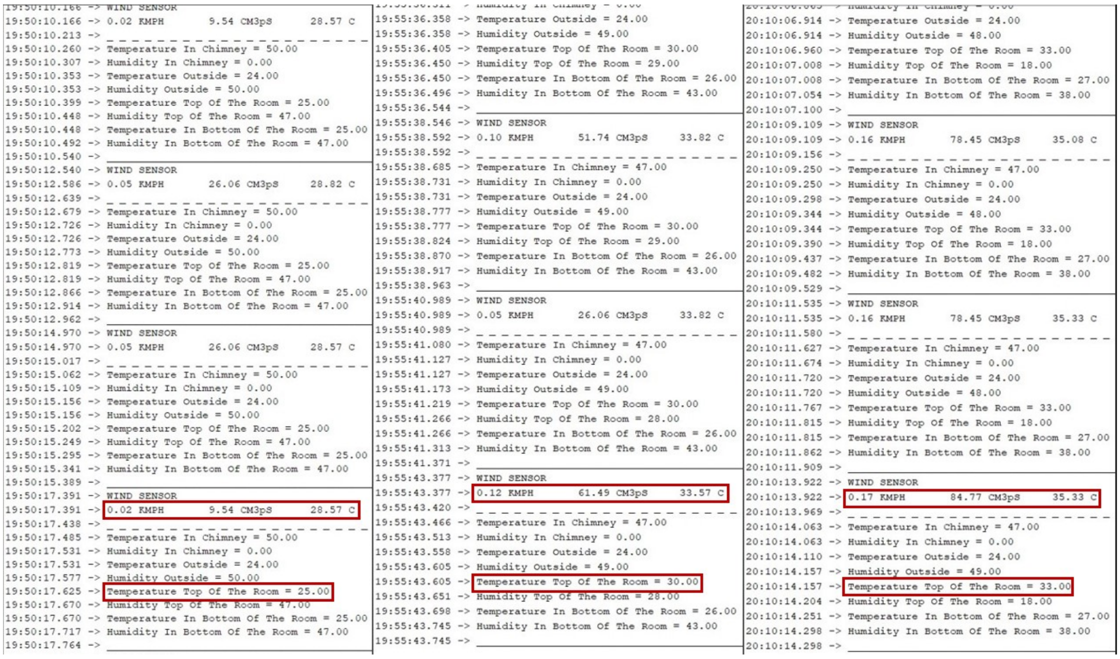
Figure 7.4: Complete Setup

7.3 Testing results and analysis

7.3.1 Heating mode results

Temperature difference results

Initially, proving of the solar chimney concept had to be done. After starting two Arduino codes, controller conducted almost 100 % of the voltage to the heating element. Starting temperature of 50°C inside the chimney was achieved after approximately 10 minutes, while all inlets and outlets were closed. After opening inlets for heating mode temperature inside the room started rising notably, which can be seen from the figure 7.5. Figure 7.5 is composed of three different segments of the test results. On the left side of the picture initial phase of the test can be seen with temperature of 25°C and almost no air-flow through the window. Middle part of the picture shows the conditions of the room after approximately 5 minutes, where it can be seen that temperature rises 5°C and also the air-flow increased significantly. After approximately 20 minutes of testing, final value of 33°C were obtained, which is less than setpoint value, but that is because of DHT11 sensors accuracy, where wind sensor show the value much closer to K type thermocouple, and thus setpoint temperature. It can also be seen that airflow is constantly higher then compared to state of closed inlets.



```
19:50:10.166 -> WIND SENSOR
19:50:10.166 -> 0.02 KMPH       9.54 CM3pS      28.57 C
19:50:10.213 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:50:10.260 -> Temperature In Chimney = 50.00
19:50:10.307 -> Humidity In Chimney = 0.00
19:50:10.353 -> Temperature Outside = 24.00
19:50:10.353 -> Humidity Outside = 50.00
19:50:10.399 -> Temperature Top Of The Room = 25.00
19:50:10.448 -> Humidity Top Of The Room = 47.00
19:50:10.448 -> Temperature In Bottom Of The Room = 25.00
19:50:10.492 -> Humidity In Bottom Of The Room = 47.00
19:50:10.540 -> 
19:50:12.540 -> WIND SENSOR
19:50:12.586 -> 0.05 KMPH       26.06 CM3pS     28.82 C
19:50:12.639 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:50:12.679 -> Temperature In Chimney = 50.00
19:50:12.726 -> Humidity In Chimney = 0.00
19:50:12.726 -> Temperature Outside = 24.00
19:50:12.773 -> Humidity Outside = 50.00
19:50:12.819 -> Temperature Top Of The Room = 25.00
19:50:12.819 -> Humidity Top Of The Room = 47.00
19:50:12.866 -> Temperature In Bottom Of The Room = 25.00
19:50:12.914 -> Humidity In Bottom Of The Room = 47.00
19:50:12.962 -> 
19:50:14.970 -> WIND SENSOR
19:50:14.970 -> 0.05 KMPH       26.06 CM3pS     28.57 C
19:50:15.017 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:50:15.062 -> Temperature In Chimney = 50.00
19:50:15.109 -> Humidity In Chimney = 0.00
19:50:15.156 -> Temperature Outside = 24.00
19:50:15.156 -> Humidity Outside = 50.00
19:50:15.202 -> Temperature Top Of The Room = 25.00
19:50:15.249 -> Humidity Top Of The Room = 47.00
19:50:15.295 -> Temperature In Bottom Of The Room = 25.00
19:50:15.341 -> Humidity In Bottom Of The Room = 47.00
19:50:15.389 -> 
19:50:17.391 -> WIND SENSOR
19:50:17.391 -> 0.02 KMPH       9.54 CM3pS      28.57 C
19:50:17.438 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:50:17.485 -> Temperature In Chimney = 50.00
19:50:17.531 -> Humidity In Chimney = 0.00
19:50:17.531 -> Temperature Outside = 24.00
19:50:17.577 -> Humidity Outside = 50.00
19:50:17.625 -> Temperature Top Of The Room = 25.00
19:50:17.670 -> Humidity Top Of The Room = 47.00
19:50:17.670 -> Temperature In Bottom Of The Room = 25.00
19:50:17.717 -> Humidity In Bottom Of The Room = 47.00
19:50:17.764 -> 
```

```
19:55:36.358 -> Temperature Outside = 24.00
19:55:36.358 -> Humidity Outside = 49.00
19:55:36.405 -> Temperature Top Of The Room = 30.00
19:55:36.450 -> Humidity Top Of The Room = 29.00
19:55:36.450 -> Temperature In Bottom Of The Room = 26.00
19:55:36.496 -> Humidity In Bottom Of The Room = 43.00
19:55:36.544 -> 
19:55:38.546 -> WIND SENSOR
19:55:38.592 -> 0.10 KMPH       51.74 CM3pS     33.82 C
19:55:38.592 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:55:38.685 -> Temperature In Chimney = 47.00
19:55:38.731 -> Humidity In Chimney = 0.00
19:55:38.731 -> Temperature Outside = 24.00
19:55:38.777 -> Humidity Outside = 49.00
19:55:38.777 -> Temperature Top Of The Room = 30.00
19:55:38.824 -> Humidity Top Of The Room = 29.00
19:55:38.870 -> Temperature In Bottom Of The Room = 26.00
19:55:38.917 -> Humidity In Bottom Of The Room = 43.00
19:55:38.963 -> 
19:55:40.989 -> WIND SENSOR
19:55:40.989 -> 0.05 KMPH       26.06 CM3pS     33.82 C
19:55:40.989 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:55:41.080 -> Temperature In Chimney = 47.00
19:55:41.127 -> Humidity In Chimney = 0.00
19:55:41.127 -> Temperature Outside = 24.00
19:55:41.173 -> Humidity Outside = 49.00
19:55:41.219 -> Temperature Top Of The Room = 30.00
19:55:41.266 -> Humidity Top Of The Room = 28.00
19:55:41.266 -> Temperature In Bottom Of The Room = 26.00
19:55:41.313 -> Humidity In Bottom Of The Room = 43.00
19:55:41.371 -> 
19:55:43.377 -> WIND SENSOR
19:55:43.377 -> 0.12 KMPH       61.49 CM3pS     33.57 C
19:55:43.420 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
19:55:43.466 -> Temperature In Chimney = 47.00
19:55:43.513 -> Humidity In Chimney = 0.00
19:55:43.558 -> Temperature Outside = 24.00
19:55:43.605 -> Humidity Outside = 49.00
19:55:43.605 -> Temperature Top Of The Room = 30.00
19:55:43.651 -> Humidity Top Of The Room = 28.00
19:55:43.698 -> Temperature In Bottom Of The Room = 26.00
19:55:43.745 -> Humidity In Bottom Of The Room = 43.00
19:55:43.745 -> 
```

```
20:10:06.914 -> Temperature Outside = 24.00
20:10:06.914 -> Humidity Outside = 48.00
20:10:06.960 -> Temperature Top Of The Room = 33.00
20:10:07.008 -> Humidity Top Of The Room = 18.00
20:10:07.008 -> Temperature In Bottom Of The Room = 27.00
20:10:07.054 -> Humidity In Bottom Of The Room = 38.00
20:10:07.100 -> 
20:10:09.109 -> WIND SENSOR
20:10:09.109 -> 0.16 KMPH       78.45 CM3pS     35.08 C
20:10:09.156 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
20:10:09.250 -> Temperature In Chimney = 47.00
20:10:09.250 -> Humidity In Chimney = 0.00
20:10:09.298 -> Temperature Outside = 24.00
20:10:09.344 -> Humidity Outside = 48.00
20:10:09.344 -> Temperature Top Of The Room = 33.00
20:10:09.390 -> Humidity Top Of The Room = 18.00
20:10:09.437 -> Temperature In Bottom Of The Room = 27.00
20:10:09.482 -> Humidity In Bottom Of The Room = 38.00
20:10:09.529 -> 
20:10:11.535 -> WIND SENSOR
20:10:11.535 -> 0.16 KMPH       78.45 CM3pS     35.33 C
20:10:11.580 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
20:10:11.627 -> Temperature In Chimney = 47.00
20:10:11.674 -> Humidity In Chimney = 0.00
20:10:11.720 -> Temperature Outside = 24.00
20:10:11.720 -> Humidity Outside = 48.00
20:10:11.767 -> Temperature Top Of The Room = 33.00
20:10:11.815 -> Humidity Top Of The Room = 18.00
20:10:11.815 -> Temperature In Bottom Of The Room = 27.00
20:10:11.862 -> Humidity In Bottom Of The Room = 38.00
20:10:11.909 -> 
20:10:13.922 -> WIND SENSOR
20:10:13.922 -> 0.17 KMPH       84.77 CM3pS     35.33 C
20:10:13.969 -> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
20:10:14.063 -> Temperature In Chimney = 47.00
20:10:14.063 -> Humidity In Chimney = 0.00
20:10:14.110 -> Temperature Outside = 24.00
20:10:14.157 -> Humidity Outside = 49.00
20:10:14.157 -> Temperature Top Of The Room = 33.00
20:10:14.204 -> Humidity Top Of The Room = 18.00
20:10:14.251 -> Temperature In Bottom Of The Room = 27.00
20:10:14.298 -> Humidity In Bottom Of The Room = 38.00
20:10:14.298 -> 
```

Figure 7.5: Heating mode results

Controller results

PID Controller in Heating mode was great success. After achieving the desired temperature of 35°C (5°C higer than outside of the house), PID controller was adjusting the voltage value accordingly. There is a video as a proof, where it can be seen that oscillations occur between 34.5°C and 35.5°C, that is inside 1°C difference, thus inside very tight tolerances. That can be seen in figure 7.6 and video [1].

Figure 7.6: LCD of PID with 34.7°C

7.3.2 Cooling mode results

Temperature difference results

Proving the solar chimney concept for the cooling mode went surprisingly faster than heating mode, final temperature was achieved after only 14 minutes as it can be seen in figure 7.7. One reason for this can be, enhancing the cooling mode by placing two ice cold cans in front of the window. Again, figure consists of three parts, initial, middle and final. Same conclusion as in heating mode can be taken, including rise of the air flow with opening of the inlets. One important difference here is, that wind sensor was not so consistent, that is because of proximity to the cold cans, that is why wind sensor results for temperature should be taken with care.

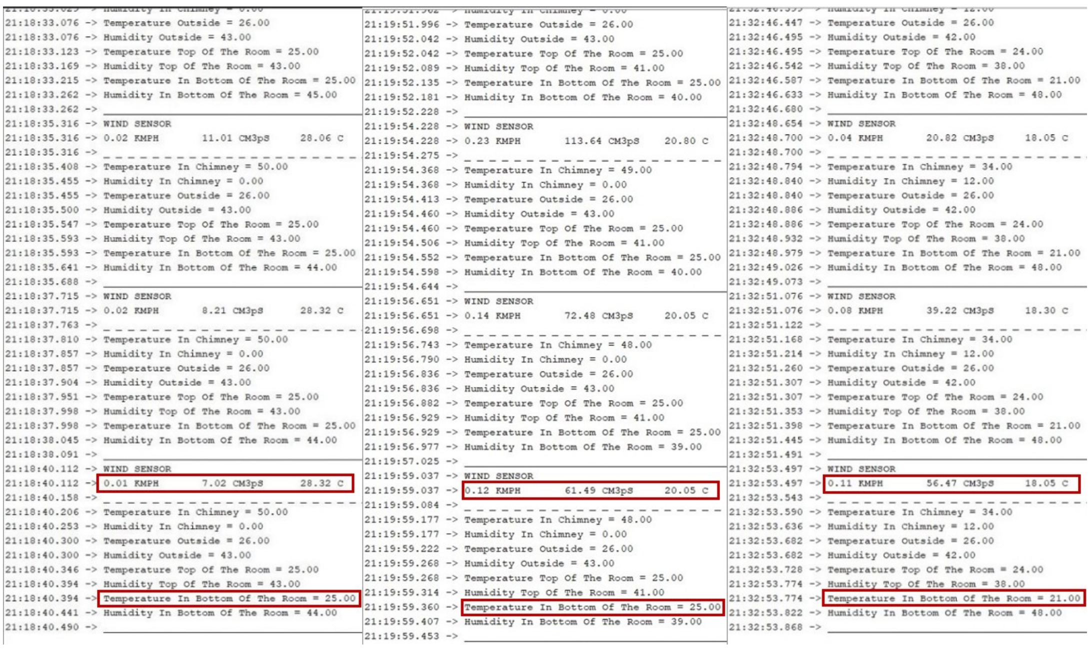


Figure 7.7: Cooling mode results

Controller results

Before testing the Cooling mode, some changes in Arduino code for PID controller had to be done. Most important one was to change the way the signal gets modified, instead of cutting the voltage when temperature goes above setpoint, reversed had to be done. So, modified PID value was added to minimum PWM value instead of subtracting it from maximum PWM value. Unfortunately, PID controller was not so succesful as in heating mode, because the measured temperature was always 2°C higher then setpoint. Beliefs of the group are that much more detailed adjusting of the PID values had to be done in order to get more precise results, although the figure 7.8 shows the best result of 22.1°C, but that was just for a brief moment.

Figure 7.8: LCD of PID with 22.1°C

7.4 Summary

Finally, all planned tests from problem formulation were conducted. Heating and cooling mode were tested, with heating mode producing more successful results. That was expected from the beginning, since cooling mode is slightly more complicated, although cooling mode obtained result faster due to contribution from ice cold cans in front of the window. All in all, goal of temperature difference with respect to outside temperature was achieved. Similarly, controller in heating mode was far more accurate than the cooling mode.

8 | Conclusion

Both goals from the beginning of the project, proving of the concept and subsequently designing the controller, were achieved. Final report was divided into four main parts, starting with dynamic model from which transfer function was later obtained that was used in designing PID controller code in Arduino. Finally, building the physical 3D model of the house.

All mentioned sections were combined in the testing chapter. Results of the test confirmed initial theoretical model of solar chimney. Controller was designed in order to maintain the setpoint value of temperature in the room. PID controller used is based on the PID values in control design section.

Many methods learned during semester were used while designing controller as for example of obtainig dynamic model, Ziegler- Nichols method, which had worked perfectly, and lastly tuning of PID controller.

In the end, group agreed that for future improvements one could use more accurate sensors compared to those used with Arduino, because also during the testing it was clear to see difference in results from various sensors.

Finally, group is very satisfied with the project outcome and course content learned during the semester, because in the end we were able to combine theoretical knowledge with real-life application and successfully designed the controller that can be used.

Bibliography

[1] *LCD PID Control Video.* https://aarhusuniversitet-my.sharepoint.com/:v:/g/personal/au703474_uni_au_dk/EXC4gdFgTfFJnyMmb-Lm8GcBNE4Me9GrsKdS_BUodsbpQg?e=J24nzr, 2022. Downloaded: 14-01-2022.

[2] *Smoke test video.* https://aarhusuniversitet-my.sharepoint.com/:v:/g/personal/au703474_uni_au_dk/ESPJFQlF6ARNj6pR_oc9c2sBI02xbcZ7iHFqitrzbS4vNg?e=Wvial1, 2022. Downloaded: 14-01-2022.

[3] AliExpress. *AC/DC 12V Thermostatic Aluminum Shell Ceramic Heating Plate Sheet Constant Temperature Heater Thermostatic Heating Plates.* https://www.aliexpress.com/item/32813489122.html, 2022. Downloaded: 10-01-2022.

[4] Arduino. *Arduino and Stepper Motor Configurations.* https://docs.arduino.cc/learn/electronics/stepper-motors, 2022. Downloaded: 10-01-2022.

[5] Arduino. *Arduino Mega 2560 Rev3.* https://docs.arduino.cc/hardware/mega-2560, 2022. Downloaded: 10-01-2022.

[6] Components101. *28BYJ-48-5V Stepper Motor.* https://components101.com/motors/28byj-48-stepper-motor, 2021. Downloaded: 11-01-2022.

[7] Components101. *DHT11–Temperature and Humidity Sensor.* https://components101.com/sensors/dht11-temperature-sensor, 2021. Downloaded: 11-01-2022.

[8] electricalengineeringinfo. *Transfer Function Mathematical Model of Thermal System.* https://www.electricalengineeringinfo.com/2017/05/transfer-function-mathematical-model-thermal-system.html, 2021. Downloaded: 13-01-2022.

[9] ElectroNoobs. *Temperature PID controller - Arduino.* http://electronoobs.com/eng_arduino_tut24.php, 2022. Downloaded: 13-01-2022.

[10] learnthermo. *Radiation Heating and Convective Cooling of a Flat Plate.* https://www.learnthermo.com/examples/example-problem.php?ch=4&lesson=B&problem=1, 2022. Downloaded: 13-01-2022.

[11] learnthermo. *Radiation Heating and Convective Cooling of a Flat Plate.* https://www.learnthermo.com/examples/ch04/p-4b-1.php, 2022. Downloaded: 13-01-2022.

[12] Modern Device. *Wind Sensor Rev.P.* https://moderndevice.com/product/wind-sensor-rev-p/, 2022. Downloaded: 10-01-2022.

[13] Long Shi. Theoretical models for wall solar chimney under cooling and heating modes considering room configuration. *Energy*, 165:925–938, 2018. Downloaded: 10-01-2022.

[14] Wikipedia. *MOSFET.* https://en.wikipedia.org/wiki/MOSFET, 2022. Downloaded: 13-01-2022.

[15] Wikipedia. *PID Controller.* https://en.wikipedia.org/wiki/PID_controller, 2022. Downloaded: 11-01-2022.

[16] Wikipedia. *PWM.* https://en.wikipedia.org/wiki/Pulse-width_modulation, 2022. Downloaded: 13-01-2022.

A Arduino Controller Code

Introduction: In this appendix the Arduino code for PID Controller used is included.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

#include <SPI.h>
//We define the SPI pins
#define MAX6675_CS   10
#define MAX6675_SO   12
#define MAX6675_SCK  13

//Pins
int PWM_pin = 3;

//Variables
float temperature_read = 0.0;
float set_temperature = Temperature Outside + 5;
float PID_error = 0;
float previous_error = 0;
float elapsedTime, Time, timePrev;
int PID_value = 0;

//PID constants
int kp = 10.04;   int ki = 0.1794;   int kd = 1.8;
int PID_p = 0;    int PID_i = 0;    int PID_d = 0;

void setup() {
  pinMode(PWM_pin,OUTPUT);
  TCCR2B = TCCR2B & B11111000 | 0x03;    // pin 3 and 11 PWM frequency of 980.39 Hz
  Time = millis();
  lcd.init();
  lcd.backlight();
}

void loop() {
 // First we read the real value of temperature
```

```
temperature_read = readThermocouple();
//Next we calculate the error between the setpoint and the real value
  PID_error = set_temperature - temperature_read;
//Calculate the P value
PID_p = kp * PID_error;
//Calculate the I value in a range on +-3
if(-3 < PID_error <3)
{
  PID_i = PID_i + (ki * PID_error);
}

//For derivative we need real time to calculate speed change rate
timePrev = Time;                            // the previous time is stored before the actual time read
Time = millis();                            // actual time read
elapsedTime = (Time - timePrev) / 1000;
//Now we can calculate the D calue
PID_d = kd*((PID_error - previous_error)/elapsedTime);
//Final total PID value is the sum of P + I + D
PID_value = PID_p + PID_i + PID_d;

//We define PWM range between 0 and 255
if(PID_value < 0)
{    PID_value = 0;    }
if(PID_value > 255)
{    PID_value = 255;  }
//Now we can write the PWM signal to the mosfet on digital pin D3
analogWrite(PWM_pin,255-PID_value);
previous_error = PID_error;     //Remember to store the previous error for next loop.

delay(300);
//lcd.clear();

lcd.setCursor(0,0);
```

```
  lcd.print("PID␣TEMP␣control");
  lcd.setCursor(0,1);
  lcd.print("S:");
  lcd.setCursor(2,1);
  lcd.print(set_temperature,1);
  lcd.setCursor(9,1);
  lcd.print("R:");
  lcd.setCursor(11,1);
  lcd.print(temperature_read,1);
}

double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);

  digitalWrite(MAX6675_CS, LOW);
  delay(1);

  // Read in 16 bits,
  //  15    = 0 always
  //  14..2 = 0.25 degree counts MSB First
  //  2     = 1 if thermocouple is open circuit
  //  1..0  = uninteresting status

  v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
  v <<= 8;
  v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

  digitalWrite(MAX6675_CS, HIGH);
  if (v & 0x4)
  {
    // Bit 2 indicates if the thermocouple is
       disconnected
    return NAN;
  }

  // The lower three bits (0,1,2) are discarded status
     bits
```

```
  v >>= 3;

  // The remaining bits are the number of 0.25 degree
     (C) counts
  return v*0.25;
}
```

B Arduino Code for sensing the Temperature and Air Flow

Introduction: In this appendix the Arduino code for sensing the Temperature and Air Flow used is included.

```
#include <dht.h>
dht DHT;

#define SensorInTopOfTheRoom (13)
#define SensorInChimney (37)
#define SensorOutside (29)
#define SensorInBottomOfTheRoom (25)

const int OutPin  = A0;   // wind sensor analog pin hooked up to Wind P sensor "OUT" pin
const int TempPin = A2;   // temp sensor analog pin hooked up to Wind P sensor "TMP" pin

void setup() {
    Serial.begin(9600);
}

void loop() {

    int windADunits = analogRead(OutPin);

    float windKMH =  pow((((float)windADunits - 264.0) / 85.6814), 3.36814)*1.6;
    Serial.println("WIND␣SENSOR");
    Serial.print(windKMH);
    Serial.print("␣KMPH\t");

    float AirFlow = (((float)windKMH/0.036)*36)/ 2.0 ;
    Serial.print(AirFlow);
    Serial.print("␣CM3pS\t"); // Air flow through window in cube centimeter per second

```

```
    int tempRawAD = analogRead(TempPin);
    float tempC = ((((float)tempRawAD * 5.0) / 1024.0)
        - 0.400) / .0195;
    Serial.print(tempC);
    Serial.println(" C");
    Serial.println("_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 
       _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ ");

    int B=DHT.read11(SensorInChimney);
    Serial.print("Temperature In Chimney = ");
    Serial.println(DHT.temperature);
    Serial.print("Humidity In Chimney = ");
    Serial.println(DHT.humidity);
    int C=DHT.read11(SensorOutside);
    Serial.print("Temperature Outside = ");
    Serial.println(DHT.temperature);
    Serial.print("Humidity Outside = ");
    Serial.println(DHT.humidity);
    int A=DHT.read11(SensorInTopOfTheRoom);
    Serial.print("Temperature Top Of The Room = ");
    Serial.println(DHT.temperature);
    Serial.print("Humidity Top Of The Room = ");
    Serial.println(DHT.humidity);
    int D=DHT.read11(SensorInBottomOfTheRoom);
    Serial.print("Temperature In Bottom Of The Room = 
       ");
    Serial.println(DHT.temperature);
    Serial.print("Humidity In Bottom Of The Room = ");
    Serial.println(DHT.humidity);

    Serial.println("
       ____________________________________________________________
       ");
    delay(2000);
}
```

C Arduino Code for opening and closing Inlets

Introduction: In this appendix the Arduino code for opening and closing Inlets used is shown.

```
#include <Stepper.h>
int stepsPerRevolution=512;
int motSpeed=50;
int dt=500;
int k=2; // 1 - for opening inlets ''Heating mode'', 2 - for opening inlets ''Cooling mode''
         // 3 - for closing inlets ''Heating mode'', 4 - for closing inlets ''Cooling mode''

Stepper myStepper1(stepsPerRevolution, 8,10,9,11);
Stepper myStepper2(stepsPerRevolution, 4,6,5,7);

boolean i=true;

void setup() {

Serial.begin(9600);
myStepper1.setSpeed(motSpeed);
myStepper2.setSpeed(motSpeed);
}

void loop() {

if(i==true) {
  if (k==1){
    myStepper1.step(stepsPerRevolution);} //opening inlets ''Heating mode''
  if (k==2){
    myStepper2.step(stepsPerRevolution);} //opening inlets ''Cooling mode''
  if (k==3){
    myStepper1.step(-stepsPerRevolution);} //closing inlets ''Heating mode''
  if (k==4){
```

```
    mySteppper2.step(-stepsPerRevolution);} //closing inlets ''Cooling mode''
  }
delay(dt);
i=false;
}
```

D | PTC Heating Element

Introduction: In this appendix Specifications of PTC Heating Element AC DC 12V Aluminum Shell Ceramic Heating Plate is shown.

AC/DC 12V Thermostatic Aluminum Shell Ceramic Heating Plate Sheet Constant Temperature Heater Thermostatic Heating Plates

Description:

This item shell is made of aluminum, eco-friendly, durable and sturdy to use. With insulator on surface, it is safe to use and can be dry heated for a long time, high reliability. It adopts high temperature wire as leading wire, durable and is also able to heat both solids and gases, convenient. Simple structure, easy to install and long service life over 3 years. Applicable for hair curler devices, water boiler, yogurt maker, chocolate extrusion, coffee maker.

Feature:

Made of aluminum shell, eco-friendly, durable and sturdy to use.
With insulator on surface, safe to use and can be dry heated for a long time, high reliability.
Adopts high temperature wire as leading wire, durable. Able to heat both solids and gases, convenient.
Simple structure, easy to install and long service life over 3 years.
Applicable for hair curler devices, water boiler, yogurt maker, chocolate extrusion, coffee maker.

Specification:

Condition: 100% Brand New
Shell Material: Aluminum
Power: (optional)
80℃:2W-5W
110℃:3W-10W
220℃:5W-28W
Voltage: AC/DC 12V
Surface Dry Heating Temperature: 80℃, 110℃, 220℃(optional) ±10℃
Leading Wire Length: Approx. 15cm / 5.9inch
Product Size: Approx. 35 * 21 * 5mm / 1.4 * 0.8 * 0.2inch
Product Weight: Approx. 14g~15g

Package Included:

1 x PTC Heating Plate

E | IRF830 MOSFET

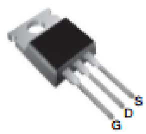
Introduction: In this appendix Data sheet of IRF830 MOSFET is shown.

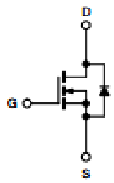
VISHAY.
www.vishay.com

IRF830
Vishay Siliconix

Power MOSFET

TO-220AB





N-Channel MOSFET

FEATURES

- Dynamic dV/dt rating
- Repetitive avalanche rated
- Fast switching
- Ease of paralleling
- Simple drive requirements
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

RoHS*
Available

Note
* This datasheet provides information about parts that are RoHS-compliant and / or parts that are non RoHS-compliant. For example, parts with lead (Pb) terminations are not RoHS-compliant. Please see the information / tables in this datasheet for details

PRODUCT SUMMARY		
V_{DS} (V)	500	
$R_{DS(on)}$ (Ω)	V_{GS} = 10 V	1.5
Q_g max. (nC)	38	
Q_{gs} (nC)	5.0	
Q_{gd} (nC)	22	
Configuration	Single	

DESCRIPTION

Third generation power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness.

The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.

ORDERING INFORMATION	
Package	TO-220AB
Lead (Pb)-free	IRF830PbF
Lead (Pb)-free and halogen-free	IRF830PbF-BE3

ABSOLUTE MAXIMUM RATINGS (T_C = 25 °C, unless otherwise noted)

PARAMETER			SYMBOL	LIMIT	UNIT
Drain-source voltage			V_{DS}	500	V
Gate-source voltage			V_{GS}	± 20	
Continuous drain current	V_{GS} at 10 V	T_C = 25 °C	I_D	4.5	A
		T_C = 100 °C		2.9	
Pulsed drain current [a]			I_{DM}	18	
Linear derating factor				0.59	W/°C
Single pulse avalanche energy [b]			E_{AS}	280	mJ
Repetitive avalanche current [a]			I_{AR}	4.5	A
Repetitive avalanche energy [a]			E_{AR}	7.4	mJ
Maximum power dissipation	T_C = 25 °C		P_D	74	W
Peak diode recovery dV/dt [c]			dV/dt	3.5	V/ns
Operating junction and storage temperature range			T_J, T_{stg}	-55 to +150	°C
Soldering recommendations (peak temperature) [d]	For 10 s			300	
Mounting torque	6-32 or M3 screw			10	lbf · in
				1.1	N · m

Notes

a. Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11)
b. V_{DD} = 50 V, starting T_J = 25 °C, L = 24 mH, R_g = 25 Ω, I_{AS} = 4.5 A (see fig. 12)
c. $I_{SD} \leq$ 4.5 A, dI/dt $\leq$ 75 A/μs, $V_{DD} \leq V_{DS}$, $T_J \leq$ 150 °C
d. 1.6 mm from case

S21-0852-Rev. D, 16-Aug-2021 1 Document Number: 91063

For technical questions, contact: hvm@vishay.com

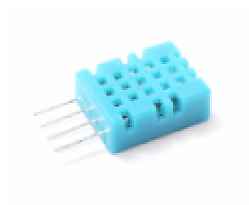
THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

F | DHT11–Temperature and Humidity Sensor

Introduction: In this appendix Data sheet of DHT11–Temperature and Humidity Sensor, according to [7], is shown.

AOSONG

Your specialist in innovating humidity & temperature sensors



Digital relative humidity & temperature sensor DHT11

1. Feature & Application:

*Good precision
*Resistive type
*Full range temperature compensated
*Relative humidity and temperature measurement
*Calibrated digital signal
*Outstanding long-term stability
*Extra components not needed
*Long transmission distance, up to 100 meters
*Low power consumption
*4 pins packaged and fully interchangeable

2. Description:

DHT11 output calibrated digital signal. It applys exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT11	
Power supply	3.3-5.5V DC	
Output signal	digital signal via Aosong 1-wire bus	
Sensing element	Polymer humidity resistor	
Operating range	humidity 20-90%RH;	temperature 0~50Celsius
Accuracy	**humidity +-5%RH;**	temperature +-2Celsius
Resolution or sensitivity	humidity 1%RH;	temperature 1Celsius
Repeatability	humidity +-2%RH;	temperature +-1Celsius
Humidity hysteresis	+-1%RH	
Long-term Stability	+-1%RH/year	
Interchangeability	fully interchangeable	

4.Dimensions: (unit----mm)

http://www.aosong.com
Thomas Liu (Sales Manager)
Email: thomasliu198518@aliyun.com , sales@aosong.com