Documentation on

Optimal Room Condition Controller

Team - 15

Team Members:

1. Muhtasim Abid - 170042035
2. Maliha Mehzabin Zoyee - 170042048
3. Sadat Bin Sharfuddin - 170042068

22th September, 2021

Introduction

The Optimal Room Condition Controller is a controller system that aims to maintain an optimal condition in our rooms which includes maintaining optimal temperature, luminosity, humidity and CO2 concentration.

The parameters we considered for maintaining optimality are:

- Perfect room temperature lies between 18 and 22 °C (according to WHO organisation).
- Optimal humidity level for health and eye comfort is minimum 40% (40% - 60%).
- Concentration of CO2 should be smaller than 1,000 ppm.
- Level of the luminosity for eye comfort should be 400 - 600 lux.

Motivation for this project based on its Social Impact:

The things that motivated us to consider this project was,

- Surrounding environment plays an important role in the productivity of a person. Optimal room condition controller can help maintain a comfortable environment for a person, in turn, increasing their productivity.
- It helps a person become healthy by maintaining a suitable condition. For example : Controlling luminosity level can help keep the eye healthy.

Some recent works related to our topic:

Most of the works related to our topic are partial implementations of the features our project has. One work was on the monitoring of the optimal room conditions, it had no feature to control them while others worked on one or two of the features like temperature and humidity, or luminosity control etc. The work we proposed is to monitor and also to control the optimal room conditions, namely, temperature, humidity, CO2 concentration and luminosity.

Features:

- Continuously monitor and show the current temperature, humidity, luminosity, and CO2 concentration of the room.
- Automatically control temperature, humidity, luminosity and CO2 concentration to bring them back to optimal conditions if they go out of optimal range.

- Added a fan to control if temperature goes above optimal condition.
- Added heater to control if temperature goes below optimal condition.
- Added fan to control if the CO2 concentration gets out of optimal range.
- Added fan to control if humidity goes above optimal condition.
- For light, if the luminosity increases from the optimal range we decrease the intensity of light, and if the luminosity decreases from the optimal range, we increase the intensity of light.
- Added notification to LCD if anything is not within the optimal condition.
- Showed readings of all the sensors in the LCD.

Sensor and Equipments:

1. **Arduino Uno :** The Arduino Uno is an open-source microcontroller board designed by Arduino.cc and based on the Microchip ATmega328P microprocessor. The board has a number of digital and analog input/output (I/O) pins that can be used to connect to different expansion boards (shields) and other circuits.

2. **DHT11 :** The DHT11 is a basic digital temperature and humidity sensor with a modest price tag. It measures the ambient air with a capacitive humidity sensor and a thermistor and outputs a digital signal on the data pin (no analog input pins needed).

3. **MQ2 :** The MQ2 Sensor is a gas detection module that can detect Hydrogen, LPG, Smoke, CO, and Alcohol. It has a great sensitivity and responds quickly.

4. **ORP-12 :** The Light Dependent Resistor (LDR), also known as the Photoresistor, is a type of resistor that has no polarity, allowing it to be connected in either direction. They can be used on a breadboard. The LDR symbol is similar to Resistor, but it contains inward arrows, as illustrated in the LDR pinout diagram above. The light signals are shown by the arrows.

5. **LCD :** An LCD screen is an electrical display module that generates a visible image using liquid crystal. The 162 LCD display is a fairly basic module that can be found in many DIY projects and circuits.

6. **Fan DC:** This is used when temperature is increased and gas is increased too.

7. **Oven :** This one is used as a heater to heat up the room when the temperature drops down.

Workflow (with observation having experimental setup) :

In our project, we used Arduino Uno, and we are taking our room's environmental readings of temperature and humidity, luminosity and gas (CO2) using the Temperature and Humidity sensor (DHT 11), LDR Light sensor (ORP-12) and Gas sensor (MQ2) respectively. And based on the readings, we are adjusting the optimality of the conditions of the room by using and controlling a Fan (DC Motor), a Heater, and an LED light. The LCD display is used for showing all the readings and notifications.

We control the optimality of the room conditions in the following way:

- If the temperature goes above 22° C, the fan is turned on and an appropriate message is shown on the LCD display.
- If the temperature goes below 18° C, the heater is turned on and an appropriate message is shown on the LCD display.
- If the amount of light in the room goes above 600 lux, intensity of the LED bulb is decreased and an appropriate message is shown on the LCD display.
- If the amount of light in the room goes below 400 lux, intensity of the LED bulb is increased and an appropriate message is shown on the LCD display.
- If the humidity of the room goes above 60%, the fan is turned on and an appropriate message is shown on the LCD display.
- If the humidity of the room goes below 40%, only an appropriate message is shown on the LCD display. (As there was no way to bring humidity back up through simulation)
- If gas is detected in the room, the fan is turned on and an appropriate message is shown on the LCD display. (We used a generic gas sensor as there's no CO2 gas sensor available for simulation in Proteus)

The diagram of the workflow is shown below:

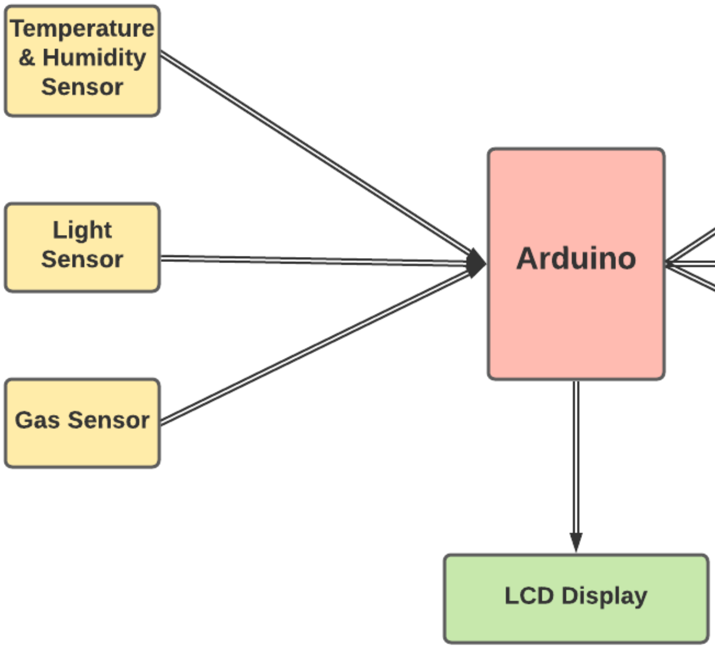

Fig: Workflow

User Manual:

1. Download the Proteus project file (optimalRoomController.pdsprj) and the compiled hex file (optimalRoomController.ino.hex) for Arduino (Uploaded in GitHub).
2. Right click on the Arduino Uno and select the compiled hex file (optimalRoomController.ino.hex) in the ‘Program Files’ field.
3. Download the MQ2 gas sensor library file (Uploaded in GitHub).
4. Right click on the MQ2 gas sensor and select the gas sensor hex file (GasSensorTEP.HEX) in the ‘Program Files’ field.
5. Start the simulation.
6. Watch the video (Link attached) for understanding how the project functions.

Links:

1. The Video Link : ***Optimal Room Condition Controller***

2. The GitHub LInk : ***Optimal Room Condition Controller***