Integrated Temperature, Light and Humidity Monitoring System for the Hospital Environment (Humidity Portion)

EE 4BI6 Capstone Project

Student: Adeel Alam [0653297]
Faculty Advisor: Dr. T. Doyle
Faculty Supervisor: Dr. J. Deen

4/23/2010

Submitted in partial fulfillment towards a Bachelor of Engineering degree
Faculty of Electrical and Computer Engineering
McMaster University, Hamilton, Ontario, Canada

Table of Contents

List of Figures 3

List of Tables 4

Abstract 5

Acknowledgements. 6

1 Introduction

1.1 Introduction. 7
1.2 Background 7
1.3 Methodology 8
1.4 Scope. 8

2 Literature Review

2.1 Literature Review. 8

3 Methodology of Design

3.1 High Level Design. 9
3.2 Description of Apparatus. 11
3.3 Microcontroller. 11
3.4 Humidity. 12

4 Experimental Design

4.1 Humidity. 17
4.2 Design for Prototype. 19
4.3 Code. 23

5 Results

5.1 Results. 23

6 Problems

6.1 Problems. 27

7 Current and Future Work

7.1 Current and Future Work. 28

8 Conclusions

8.1 Conclusions. 28

Appendix A: Specifications for Each Component. 29

Appendix B: Code. 33

Appendix C: Laws and Theory 46

References 51

Vitae. 52

List of Figures

Figure 1: Overall Experimental Design 10

Figure 2: Arduino Duemilanove 13

Figure 3: Humidity Sensor Selected 14

Figure 4: Different Sensors Considered 14

Figure 5: Spray Nozzle 14

Figure 6: Spray Bottle with Servo Motor 15

Figure 7: Ultrasonic Transducer 16

Figure 8: Output of the Humidity Sensor 17

Figure 9: BJT 18

Figure 10: Transducer with the SSR 19

Figure 11: Storage Cube 19

Figure 12: Lunch Box with the Transducer 20

Figure 13: The Lunch Box with the Pipe Apparatus 21

Figure 14: The Wood Glued from the Top 22

Figure 15: Results from having a user input at 70% 24

Figure 16: User input at 55% 25

Figure 17: Results From Having a User Input at 30% 26

Figure 18: Initial Ultrasonic Transducer 27

Figure 19: Ultrasonic Transducer 29

Figure 20: Arduino Microprocessor 30

Figure 21: 5 volt fan used for humidity control 31

Figure 22: BJT used for control of the 5 volt fans 31

Figure 23: Transformer used to provide 35 VAC to the transducer 32

Figure 24: Solid State Relay used to control the transducer, 12 volt fan and power resister 32

Figure 25: Circuit drawing for the BJT switch 47

Figure 26: Schematic of the Humidity Portion 48

List of Figures

Table 1: Comparison between the articles 9

Table 2: Output from the Humidity Sensor 23

Table 3: Output of Humidity 49

Abstract

A smart integrated temperature, light and humidity monitoring system, which actively monitors the environmental conditions, has been implemented with the use of open standard technology and household items. These materials allow the implementation to be as cost efficient as possible. The system allows for a user to input the desired conditions regarding a specific patient's temperature, lighting and humidity requirements. A microcontroller then compares the environmental conditions against the user's input requirements, and actuators changes these conditions until the desired ones have been obtained. The systems' objective is to be integrated in a hospital environment, with emphasis on the neonatal ward, where infants remain in incubators for a period of time. Results indicate the hospital monitoring system is feasible; however, ensuring an enclosed room may be difficult to obtain.

Keywords – smart monitoring device, integrated temperature, lighting and humidity, controlled environment

Acknowledgements

Recognition goes to Dr. Thomas Doyle for providing guidance and feedback throughout the course of this project.

Special thanks go to Dr. Jamal Deen allowing his ideas to be explored, as well as for supervising and guiding this project towards completion.

Recognition is given to Usman Waseem for his aid in the selection of materials and for troubleshooting the project.

Appreciation goes to Anam Fatima for providing motivation and determination during the course of the project.

Special thanks are given to Christina Tan for being a highly cooperative and patient partner for this project.

Introduction

Definition:

The proposed project is to create a device that is able to take in a specific environmental signal from a room. This measured signal, either light, humidity or temperature, would be monitored via a sensor. When a certain stimulus is added which causes a change in the overall condition of the room, sensors will detect this and actuators will try to bring back these environmental variables to their original intended value. An actuator is "a mechanism that puts something into automatic action" [dictionary.com].

Background:

A hospital stay is just one of many life experiences that a person can go through. While in the hospital, the individual may need certain environmental variables of the room, such as temperature, light or humidity, to correspond to his or her condition. This is beneficial because if these environmental factors are controlled, they can result in a speedy recovery for the patient. In reality, however, these conditions may change due to external factors, such as the weather, and hospitals lack the resources in order to have these rooms continuously monitored. This proposed system is designed to correct any deviation from the ideal environmental conditions, as determined by a doctor.

An example of where this integrated monitoring system would be beneficial is in an incubator. The daily evaporative loss from a premature baby can exceed 20% of their body mass. This can result in a high amount of heat loss. It has been found that the partial pressure of water vapour between the air and the newborn's skin is directly proportional to the exchange between the environment and neonate. Therefore, if the humidity of the environment can be controlled, the amount of evaporative loss can be reduced. [11]

The project is unique in such a way because any integrated devices that monitor and control the combination of temperature, lighting and humidity do not exist. The closest is a device that can monitor all three of these factors, but cannot control.

There were several assumptions needed for the project to perform at its peak. The room is assumed to be airtight. The only form of control of air would be through the ventilation system within the room. This allowed for better control of the humidity as “outside” air cannot enter at will. Furthermore, it was assumed that the “outside” air is very dry.

Humidity is a variable that is measured relatively. Relative humidity, the proper term, is defined as the amount of water vapour that the surrounding air can hold at that particular temperature. Therefore, a relative humidity reading of 100% means that the amount of water vapour that the air can hold has been saturated. Sensors, that measure humidity, give out outputs corresponding to the relative humidity at a particular temperature.

Methodology:

The project will consist of directly dealing with the humidity component of the room. Therefore, there will be a sensor that measures the humidity. [9] The sensor will be placed near ventilation system of the room in order to measure the overall humidity of the room. After the signal is received and conditioned, a microprocessor will compare the signal to a manually inputted value by the user. Based on this, the microprocessor will decide whether there needs to be some external input into the environment to increase the humidity. If this is the case, a signal will be sent to a transducer to generate mist to increase the overall humidity. [1]

Scope

The temperature, lighting and humidity monitoring system are to be a cost efficient system and targeted towards patients in the hospital environment. One of the main focuses would be for infants, including infants that are placed in incubators, and for the treatment of neonatal jaundice.

Literature Review

From the many different articles that were looked at, no papers that specifically mention monitoring the combination of temperature, lighting and humidity in one integrated system and have actuators to modify these settings were found. There was, however, a journal found for the

automatic detection and control of light. In the article, a light sensor in the control module detects a change in light intensity and a radio frequency module is used to change the lighting [8].

A research paper that simply monitored the three environmental conditions was looked at. However, there has been no mention about having actuators to modify these surroundings. An advantage of this sensor system is that it is wireless, based on Zigbee technology, which has low power consumption. [6]

Another research paper showed a method to effectively control humidity in an incubator. They controlled humidity based on a passive and active system. The passive system had a water reservoir where air was passed over it. The air would pick up humidity and distribute it throughout the room. The level of control was minimal. The active method used a step motor that rotated to let humidity in. Once again, this method only had control of humidity, not of all three environmental variables that this report focuses on. [11]

Paper	Environmental Focus	Notes
1	Light	
2	Light, temperature, Humidity	Researched focused on sensing, not controlling
3	Humidity	
This Project	Light, temperature, Humidity	Focus on sensing and controlling

Table 1: Comparison between the articles

Methodology of Design

High Level Design

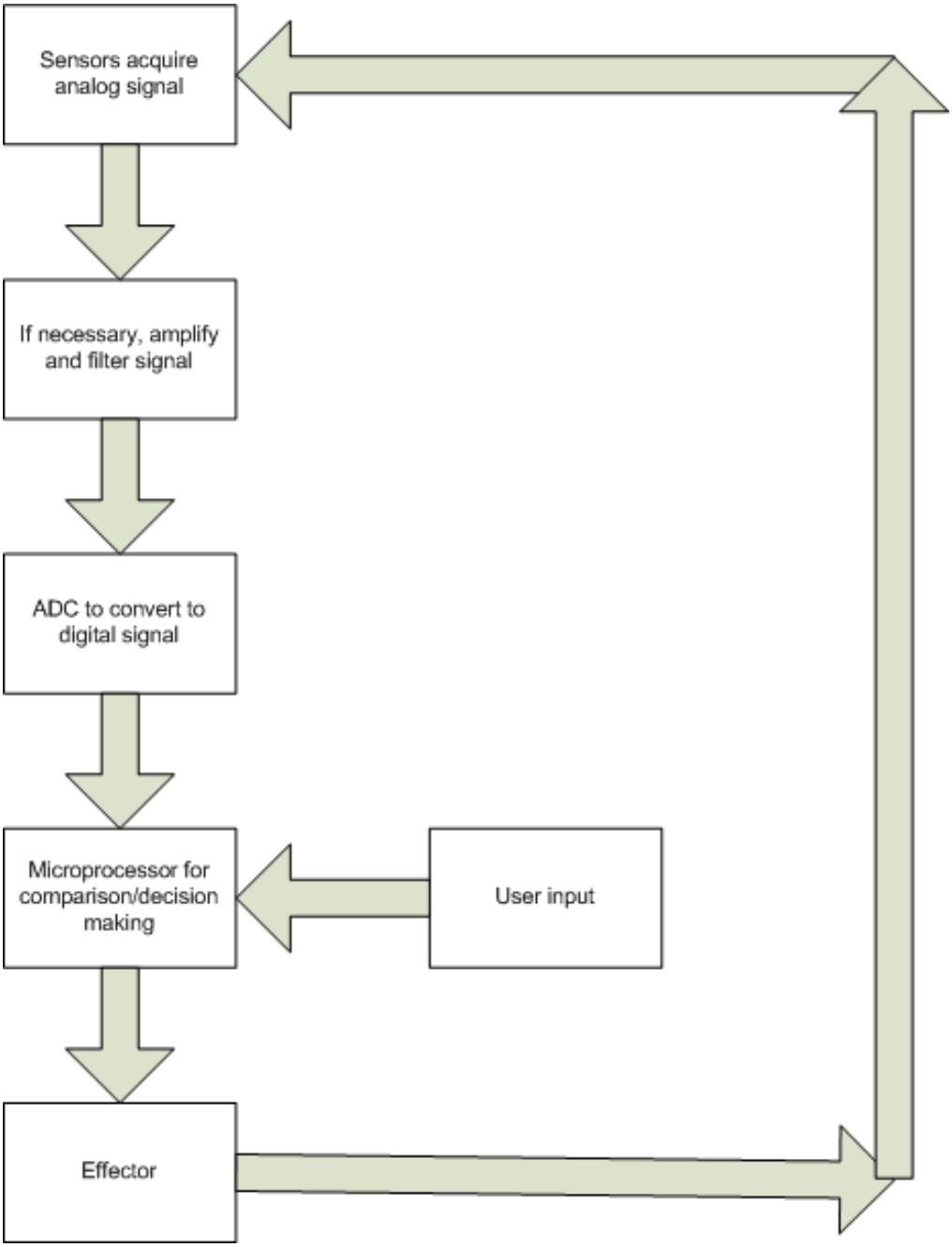


Figure 1: Overall Experimental Design

The overall design of this project followed a very systematic approach. Three environmental factors, lighting, temperature and humidity, were to be controlled. Therefore to effectively control a variable, an input is required from an external source. In the scope of this design, the inputs from the external sources were obtained via a variety of sensors. These sensors detected the surrounding environment and provided inputs to the control mechanism. [12]

In order to effectively utilize the control mechanism, the input from the sensor needs to be modified for it to be recognizable. These modifications are in the form of filtration or amplification. However, for the purposes of this project, the sensors were chosen so their output correlates directly with the control mechanism. Thus, any modifications were not required.

The control mechanism was a microcontroller. The microcontroller read the output from a sensor and made a decision in regards to whether the input had deviated from its intended value. This intended environmental setting came from a user. The user manually inputted this intended value.

After the microcontroller determined that the variable in question had deviated from its target value, a signal was sent to an actuator to start operating. This actuator functioned to either increase or decreases the effect of that environmental variable. Therefore, over time, the actuator returned the sensed value back to its intended state, behaving like a negative feedback loop. Please refer to Figure 1 for a flow diagram.

Description of Apparatus

This section entails the different components used throughout the extent of this project and the explanations as to why they were selected. For specifications of each component, please refer to Appendix A.

Microcontroller

The microcontroller selected for this design was the Arduino Duemilanove, which uses the ATMega328 processor. This processor is extremely strong and is cheap so it was ideal to purchase. It requires an input voltage range from zero to five volts, which coincided with the humidity sensor. The Arduino Duemilanove contains an on-board 10-bit ADC, which aided with the digitization of the signal acquired from the sensor. Furthermore, the temperature and light sensors used in the prototype were easily integrated with the microcontroller, as they are designed specifically for the Arduino board. The Arduino platform uses a language almost the same as C, with some variations. Therefore, having been familiarized with the C programming language, it was convenient to settle with this microcontroller. Furthermore, the Arduino is a well known board with plenty of examples and references on the web to guide this project forward. It was very simple to load the code onto the board from the computer, since the software was available for free on the Arduino website. Lastly, the Arduino had a serial port where it was possible to communicate with the computer. The USB connection from the computer went directly onto the Arduino board. Here, it possessed a USB to serial converter which made this communication possible. This was desirable as this project requires a user input. The Arduino platform made this possible. [10]

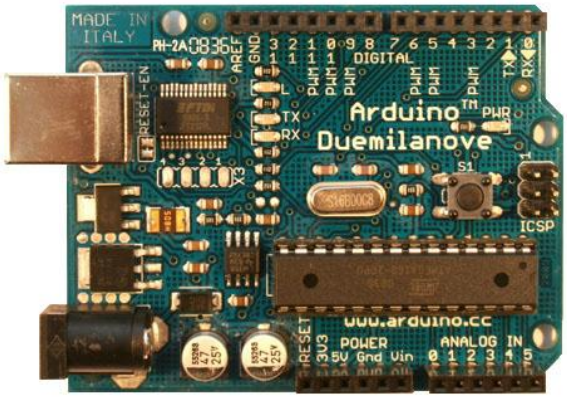


Figure 2: Arduino Duemilanove

Other microcontrollers were also considered, including the PIC series. These PICs contained 10-bit ADCs and had operating voltages ranging from 1.8 to 5.5 volts. Furthermore, to upload a code from the computer, a bootloader program along with its corresponding hardware was required to be purchased. These factors made this microcontroller unfavourable to use.

Humidity

In order to control humidity, a humidity sensor was required to effectively acquire readings from the environment. The humidity sensor chosen was the 808H5V5 from Sencera Co. LTD. Different humidity sensors were researched, including the HS1101LF from Measurement Specialties and the HIH-4602-C manufactured by Honeywell. A drawback to the sensor from Honeywell was that its response time was 50 seconds. For this application, the response time should be less than 10 seconds to effectively monitor the environment. Secondly, the output of this sensor returned a value in Farads. A voltage output was required to effectively integrate the sensor with the microcontroller. Therefore, an output in Farads was undesirable because further circuitry would be required to convert that capacitance output into a voltage output. For the sensor manufactured by Measurement Specialties, the response time was five seconds but it also returned an output measured in Farads. A second drawback to this sensor was that it only sensed the RH value from 1% to 99%, when a range from 0% to 100% was desired. The 808H5V5 had a response time of 15 seconds. However, the biggest selling point was that it provided a voltage

output in the range from zero to five volts. This was a huge advantage as it could directly be integrated with the microcontroller. [9]

Figure 3: Humidity Sensor Selected

Figure 4: Different Sensors Considered

Initially, there were three different actuators researched for the intent of increasing humidity. First and foremost, an automated mist system was looked upon. A spray nozzle which would provide different amounts of mist was considered.

Figure 5: Spray Nozzle

This idea was immediately discarded because the cost of it was a main concern. Spray nozzles are sold for over $1000, which did not abide by the project's objective to be cost efficient.

Another idea to increase the humidity was to attach a servo motor onto the back of a spray bottle. When a higher humidity level is required, the servo motor would turn at a fast enough rate to allow the spray bottle to spray mist into the air.

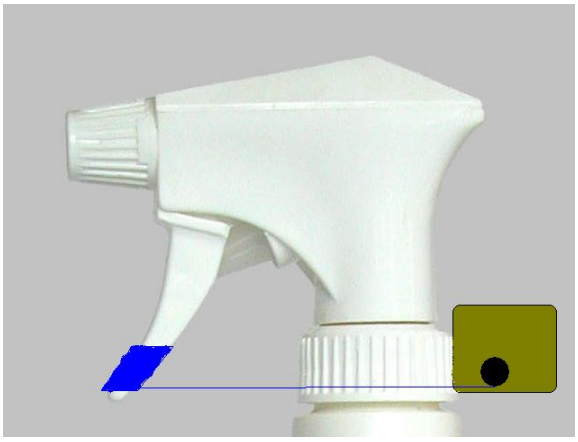


Figure 6: Spray Bottle with Servo Motor

This idea was also discarded because there must be a connection from the handle to the servo motor. If the connection were to break, it would render the whole project obsolete. Furthermore, calculations with how much tension would be required to cause mist to be released would be obligatory. This tension would have to correspond to the servo motor's torque. There is little room for error when equating the values for tension and torque; this motion also increases the likelihood of breaking the string and apparatus.

The design chosen was an ultrasonic transducer. This transducer is made from piezoelectric material. It has a disk that is slightly larger than a loonie. This disk resonates at a frequency in the order of megahertz so that when it is placed in water, mist is generated.

Figure 7: Ultrasonic Transducer

This was the most effective design as it was cost effective. Furthermore, the chances of the transducer breaking down are minimal, due to its compact design and sturdy frame. [1]

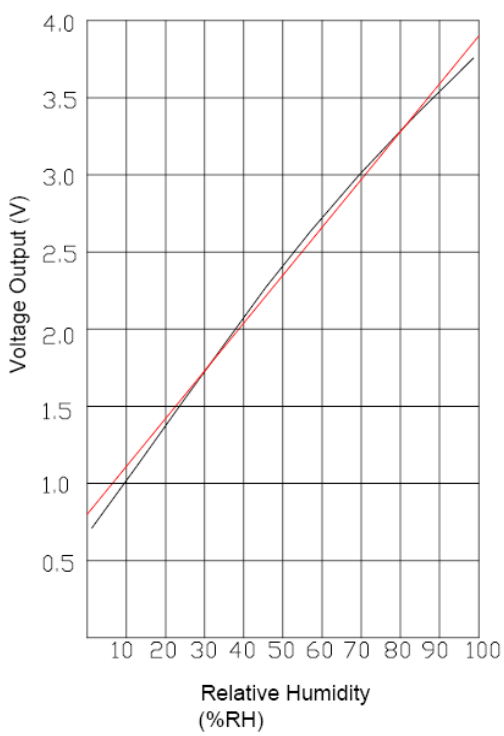
The actuators used to decrease humidity were essentially two fans. These fans were placed on opposite ends of the cubicle. The fans worked to blow the humidified air out of the container. As that air leaves, new air from outside the cubicle replaces the humidified air through a drilled hole located near the bottom right-hand side of the cube.

Experimental Design

The following section describes how each component coordinated with other components throughout the project to complete the objective that was required of this design.

Humidity

The humidity sensor needed an input of five volts. An advantage of the Arduino board is that it provides pins for both a five volt and a 3.3 volt power supply. Therefore, the power and ground pins of the Arduino were connected to the positive and negative leads of the humidity sensor. The middle lead of the humidity sensor is the output, which provides a voltage within the range of zero to five, corresponding to the relative humidity in the room. Please refer to Figure 16. The output of the humidity sensor was connected to analog pin two on the Arduino.



Legend		
Definition of each line		
Symbol	Count	Description
—	1	Approximated output
—	1	Actual Output

Figure 8: Output of the Humidity Sensor

The black line refers to the actual output of the humidity sensor. This output is almost linear; therefore the red line was used for calculation purposes because it is linear.

The Arduino has a built-in 10 bit analog to digital converter (ADC). Thus, the Arduino divides the input voltage from the humidity sensor into 2^10 = 1024 equal spacing. This corresponds to approximately 5mV. The value from analog pin two was read into the code. Based on this value and the user input, the code decided which actuator needed to turn on. The actuator turned on when the Arduino sent 5V to it, whether it were the fans or the transducer.

Even though the fans ran on 5V, they were unable to draw enough current from the Arduino's output pins to run at their maximum potential. Therefore, an external power supply or the power supply from the Arduino was required. This power supply is "on" all the time, therefore there is no sense of control. The fans only needed to be on when it is necessary to remove the humidity from the cube. Furthermore, in the case of increasing humidity, only the fan near the pipe opening needed be turned on along with the transducer. Therefore, some sort of control mechanism was needed.

This control mechanism came in the form of a bipolar junction transistor (BJT). A bipolar junction transistor operates when, if a certain voltage is provided at the base junction, current will flow from the collector to the emitter. Therefore, when the Arduino provided five volts to the base, the BJT turned on, causing the power supply of the Arduino to provide power to the fan. This power supply caused the fan to work at its maximum potential. (Refer to Appendix C) [12]

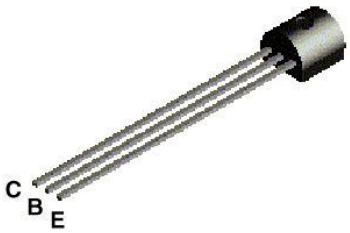


Figure 9: BJT

In order to control the transducer, a BJT was not possible because it ran on AC voltage. Therefore, a solid state relay (SSR) was used. A SSR works in a similar fashion to a BJT. It essentially functions as an electrical switch. After providing a certain voltage on one side, the electrical switch closes on another side, causing the circuit to be completed. In this case, the

Arduino provides five volts to the SSR and the switch closes. The transducer had connection holes built in for precisely this purpose. Therefore, it was easy to solder the SSR onto the switch portion of the transducer. Thus when the switch was closed, it completed the circuit of the transducer and it turned on. [10]

Figure 10: Transducer with the SSR

Design for Prototype

The project's objective was to build a prototype that would simulate a real hospital room. To build this, a small enclosed storage cube was purchased. Five pieces of wood was assembled to generate a cube with one of its faces open. To cover up the opening, Plexiglas was placed and screwed on.

Figure 11: Storage Cube

The ultrasonic transducer was set up in such a way that some sort of container could be mounted on top of it. Therefore, a plastic lunch box was purchased and a hole was drilled into it to match the size of the ultrasonic transducer. After the lunch box, or water reservoir, was placed on top of the transducer, screws were used to ensure the transducer remained attached to this lunch box. An extensive amount of silicone was placed on the gaps to guarantee the lunch box was water tight.

Figure 12: Lunch Box with the Transducer

The water reservoir would be placed outside the reservoir. Thus, a rubber pipe was used to transfer the humidity contained in the lunch box into the cube. A hole was drilled into the lunch box from the side and into the cube from the side. A connector for the pipes was screwed to each individual reservoir (water reservoir and cube). The pipe was connected and it enclosed the lunch box and the cube. Using the pipe ensured that the humidity had a direct connection from the water reservoir to the cube. Please refer to Figures 12 and 13.

The transducer needed a 36 VAC power supply. After the power supply was purchased, it had to be mounted in close proximity to the transducer itself. Therefore, the water reservoir and the power supply were placed on a piece of wood. The power supply was a transformer, so it needed

to be connected to a standard wall outlet. Thus, the leads from the primary side of the transformer needed to be connected to a plug. The leads were soldered on to that plug.

Figure 13: The Lunch Box with the Pipe Apparatus

To ensure the humidity was evenly spread amongst the inside the cube, there was a fan placed right on the opening of the drilled hole into the cube. The fan functioned to suck the humidity coming from the hole and distribute it across the room. The second fan, across the room, served a similar role. It was placed directly in front of a funnel which led out of the cube. Therefore, this fan controlled and directed air it out of the cube.

As air is leaving the cubicle, new air must be able to enter in order to maintain equilibrium. A new supply of air came from another drilled hole on the right hand side of the cube. (See Appendix C)

There was a block of wood attached to the ceiling of the cubicle. Protruding from this piece of wood, a breadboard was fastened. The humidity sensor was placed on the breadboard. There were wires from the breadboard traveling to a second breadboard on the floor in order to supply the humidity sensor with power.

Figure 14: The Wood Glued from the Top

The second breadboard was placed near the back of the cube. The Arduino's power supply powered the humidity sensor. To control the fans for the humidity, there were two BJT's on this breadboard. From the Arduino board, the power supply and ground pins were connected to the positive and negative connections on the breadboard. Therefore, from these pins, they powered the two fans for the humidity control, and the humidity sensor.

Code

The code used to program the Arduino essentially had a similar type of code for each of the environmental factors. The code was based on if statements. If the user input variable was greater than the value from the sensor, an actuator would be turned on to increase the effect of that specific variable. In contrast, if the user input variable was less than the value from the sensor, an actuator would be turned on to decrease the effect of the specific variable. Please refer to Appendix B.

Results

Each individual environmental variable was tested in the cube to observe how effective the prototype setting is.

For humidity, several cases were tested to ensure the functionality of each actuator and the humidity sensor. The humidity sensor was tested by simply viewing the readings from the humidity sensor on the serial monitor.

Time (s)	Relative Humidty (%)
0	53.43
15	53.56
30	53.65
45	53.21
60	52.95
75	53.44
90	54.44
105	54.21
120	53.95

Table 2: Output from the Humidity Sensor

The humidity was read after every 15 seconds. Fifteen seconds was selected because that was the response time for this sensor. Table 1 shows the relative consistency of the humidity readings in the cube. This ensured the functionality of the sensor as it provided consistent readings.

Secondly, the system was tested under the assumption that the user input was 70% relative humidity. Figure 15 displays these results.

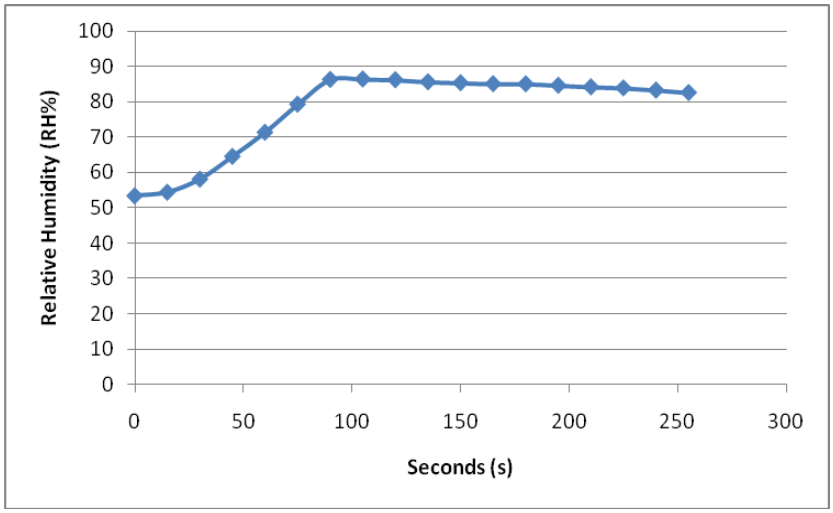


Figure 15: Results from having a user input at 70%

From Figure 15, it is evident that the actuator mechanism to increase humidity has a far greater effect than the actuator mechanism to decrease the humidity. Even though the input was set at 70%, the cube reached that value but it over shot it by a huge margin. After it did overshoot, it took considerable time to decrease the value back to its intended value of 70%. Therefore, the actuator that reduced the humidity in the cube were not as effected as orginally intended. A modification that needs to be made is that the intensity of increasing the humidity needs to be decreased as the sensor value approaches the intended user value. The sensor has a response time of 15 seconds. Thus, a reading of 67% relative humidity will not affect the transducer. Even though 67% is close to the indended value of 70%, the transducer will continue to function until it reaches 70%. However, 15 seconds later, that value can potentially shoot up to 75%. Therefore, as the value of the sensor creeps closer to the intended value, the transducer should turn off.

Next, the effect of having the user input at 55%

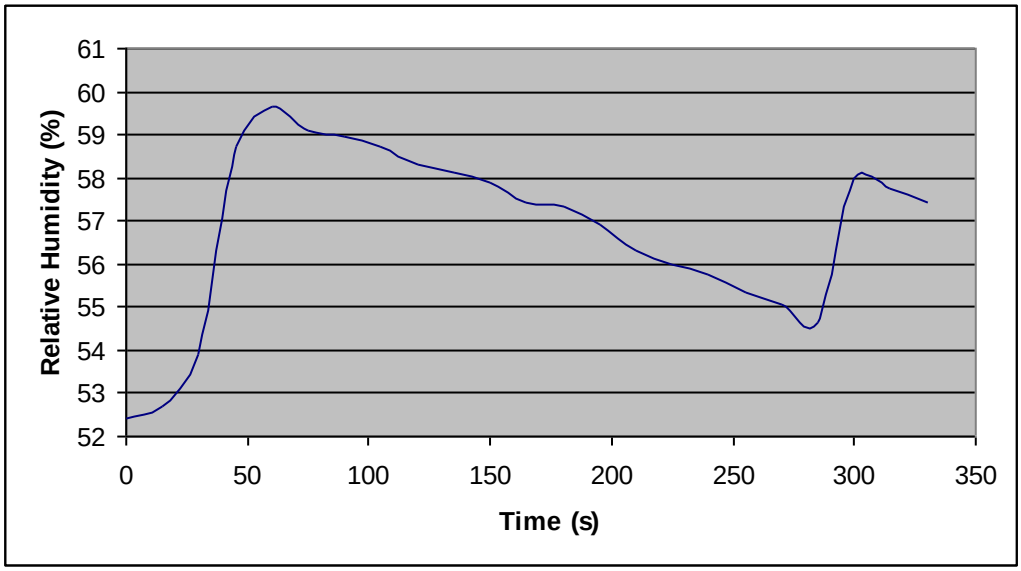


Figure 16: User input at 55%

After careful observation, it can be noted that the humidity fluctuates around the intended value of 55%. Since environmental condition will always force the away from the desired value, it is impossible to hold the value at 55%. There always needs to be room for error to take into account the non ideal factors. Furthermore, it should be noted the sharp rise when the humidity falls below the desires value. As soon as the graph fell below 55%, the transducer's effect was very pertinant. The transducer's ability to increase the humidity overshadows the fan's ability to decrease the humidity. This is noted in the steady decline of the humidity. Once again, further time needs to be spend on a way to produce dry air. This dry air will allow for greater control in the cube.

Lastly, the effect of decreasing the humidity was tested. The user inputted a value of 30% humidity. Figure 17 shows those results.

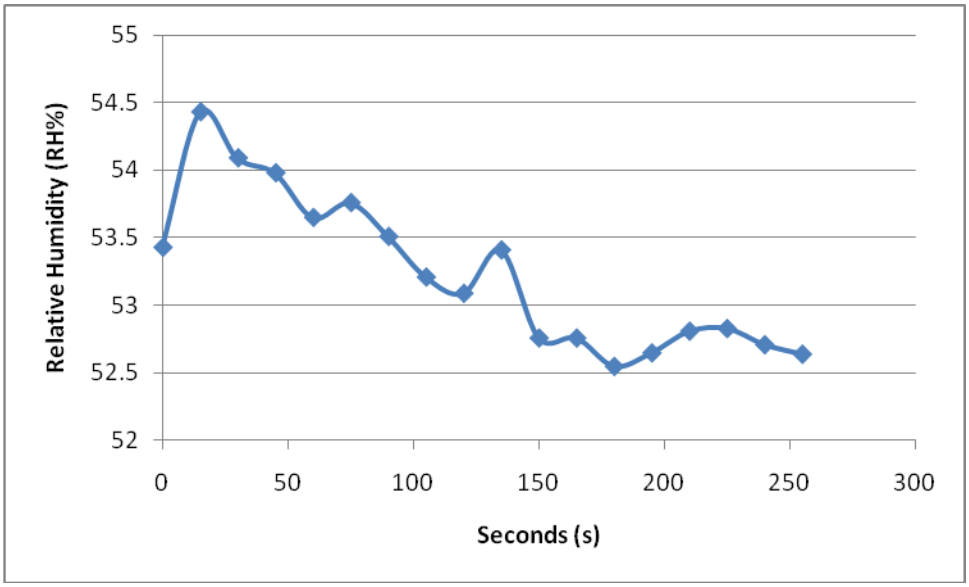


Figure 17: Results From Having a User Input at 30%

Even though the fans somewhat decreased the humidity, it did not reach the 30% value. This is because the humidity is decreased by allowing air from the outside come into the cube. Since all the air from the environment flowed into the cube, the humidity only decreased until it reached the overall humidity of the environment before saturating. This was a limitation of the project. These fans were placed under the assumption that the air outside is very dry air.

Problems

To decrease the humidity, a main assumption was that the air coming into the cubicle was significantly lower than in humidity than the air leaving. This, however, was not the case as the air entering was simply the air of the room that contained the cubicle. Therefore, the humidity level was saturated at a certain level and could not decrease any further.

The solution to this would be to find a source that can cause dry air to enter the room. The initial thought was to take the surrounding air and force it to go through a tube surrounded by a metal

heating coil. The heating coil would remove the moisture in the air to dry it. This idea is still under consideration and further research is needed.

Another issue currently under investigation how fast the humidity leaves the room. This depends on the fan that is placed on the funnel. This fan runs on five volts but is not the fastest option. The solution proposed will be to use at 12 volt fan and compare the results.

Another problem occurred with the ultrasonic transducer. Initially, the transducer purchased was different from the one being used for this project. The former one ran on 26 VDC. However, after testing this transducer, it was apparent that it will generate very little mist. It required a very big heat sink as it produced a vast amount of heat. This transducer was quickly discarded and the current transducer was implemented instead.

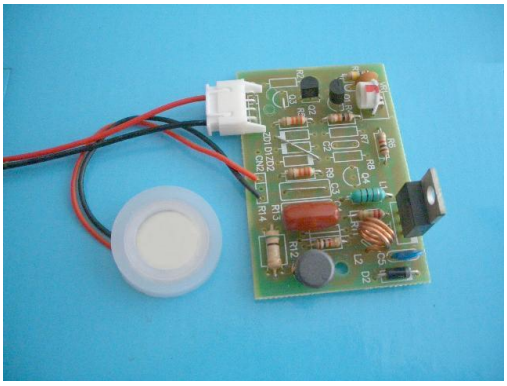


Figure 18: Initial Ultrasonic Transducer

Current and Future Work

At the moment, the code that is used for humidity is simply a couple of “if” statements. These statements can lead to the massive overshoot while testing. Therefore, the code needs to be altered to minimize this overshoot. These alterations are necessary to overcome hysteresis. It is a problem in the system that also needs to be removed from the humidity effecter. One way that is being considered is to simply stop the transducer from functioning as the desired RH percentage is approached, attempting to prevent overshooting from occurring. Another method currently being researched is to use a proportional-integral-derivative (PID) controller.

The future work for this project would be to integrate this into a room. The room can use the sensors for monitoring purposes. The actuators can be the thermostat, the lighting and the humidifiers in the rooms. This integration would make this a very unique project as it would have total control of these three factors in a room.

Lastly, projects like these can be integrated in a grocery shop. Food items can rot when they are not in an ideal environment with respect to humidity and temperature. This project can monitor the environment that the fruits and vegetables are placed in. If the ideal conditions change, these sensors can detect that and restore those conditions. This would cause less food to be discarded from these grocery stores.

Conclusions

The designed temperature, lighting and humidity monitoring system have proven to successfully acquire accurate measurements for the above mentioned environmental factors. The actuators have been able to modify these factors by changing these values to what the user has requested. Although the actuators are not 100% effective yet, necessary steps are being taken, with respect to the hardware and code, to make them effective. Other problems encountered throughout the course of this project are currently being researched and are targeted to be resolved.

Appendix A - Specifications

Specs for the Transducer

Piezo Material: SM111

Dimensions: 125mm x 70mm x 37mm

Resonant frequency f_r: 1.66 MHz±50 KHz **Electromechanical coupling coefficient K_t:**≥40%

Resonant impedance Z_m: ≤2 Ω

Static capacitance C_s: 2050pF±20%@1kHz

Voltage Input: 36Vac

Max Power Input: 25 W

Protective Layer: Stainless Steel

Mist Production: 350 cc/Hr*

Accessories: Driver board, the Piezo transducer and water level sensor.

Test Condition: 23±3 °C 40~70% R.H.

C_s => LCR meter at 1KHz 1Vrms

Figure 19: Ultrasonic Transducer

Specs for the Microprocessor

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

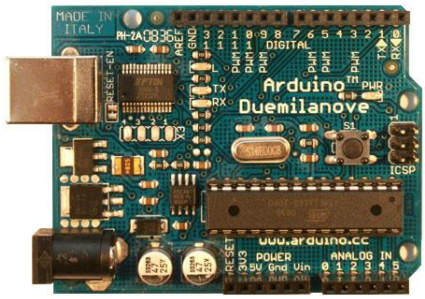


Figure 20: Arduino Microprocessor

Specs for the Fans (5 volts)

Model	1606KL-01W-B30
Rated Voltage	5 V
Operating Voltage	4.5 – 5.5 V
Current	0.120 A
Power	0.6 W
h_{FE}	40

Speed 6000 min

Figure 21: 5 volt fan used for humidity control

Specs for BJT

Rated Voltage 50 Volts

Rated Current 500 mA

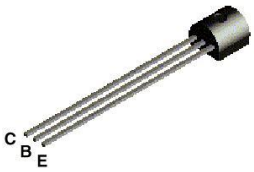


Figure 22: BJT used for control of the 5 volt fans

Specs for Transformer

Primary Voltage	115 V
Secondary Voltage	35 VCT
Secondary Current	1.5 A

Figure 23: Transformer used to provide 35 VAC to the transducer

Specs for the Relay

Output Contact On Resistance: 0.05 ohms

Maximum DC Switching Voltage: 40 VDC

Maximum AC Switching Voltage; 28 VAC

Maximum Switching Current: (AC or DC) 2.5A

Maximum Continuous Current: (AC or DC) 2.5A

Maximum Switching Speed: 20 cycles/second

Figure 24: Solid State Relay used to control the transducer, 12 volt fan and power resister

Appendix B - Code

Code Used for Temperature, Lighting and Humidity

```
//Data type declaration//

int light_pin = 0; //light sensor pin

int temp_pin = 1; //temperature sensor pin

int hum_pin = 2; //humidity sensor pin

//temperorary data type declaration//

int custom_light = 0; //store the expected illuminance

double custom_temp = 0;  //store the expected temperature

int custom_hum = 0; //store the expected humidity

int custom_data = 0;  //store the expected data update

int temp = 0;

int temp1 = 0;

int byte_i;

int serial_read;

char reading_Array[5];

void setup()

{

  //Initialize pin mode//

  pinMode(11, OUTPUT); //LEDs

  pinMode(10, OUTPUT); //heater (temp)
```

```
  pinMode(9, OUTPUT); //fan (temp)

  pinMode(8, OUTPUT); //fan for increase humidity

  pinMode(7, OUTPUT); //fan for decreasing humidity

  pinMode(6, OUTPUT); //transducer to increase humidity

  pinMode(temp_pin, INPUT);

  pinMode(light_pin, INPUT);

  pinMode(hum_pin, INPUT);

  //Initialize Serial port//

  Serial.flush();

  Serial.begin(300);

 Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");

}

// the loop() method

void loop()

{

  int light_read = analogRead(light_pin);

  int room_temp = (125*analogRead(temp_pin)) >> 8;

  double hum_reading = analogRead(hum_pin);

  double voltageValue = (5*hum_reading)/1023;

  double rh = (voltageValue - 0.788)/0.0314;

 update_data();
```

```
  //output light strength value

  temp = map(custom_light,0,7,0,1023);

  if ((temp-(1023-light_read)) <= 0){ //1023-light_read b/c light_read values are inverted

    analogWrite(11, 0);

  }

  else analogWrite(11, map((temp-(1023-light_read)),0,1023,0,255));// PWM for LED
brightness; range of 0 to 255

  //output heater and fan strength value

  if (custom_temp != 0){ //we are gonna choose a room temp range from 0 to 49..these are used
to do the below calculations

    if (custom_temp < room_temp) {

      if (room_temp-custom_temp > 5)

        analogWrite(9, 255);

      else

        analogWrite(9, 255);

        //analogWrite(9, (room_temp-custom_temp)*6.5); //closest to get 255 (39*6.5 = 254)

      analogWrite(10, 0);

    }

    else if (custom_temp > room_temp){

      if (custom_temp-room_temp > 3)

        analogWrite(10, 240);

      else if (custom_temp - room_temp > 2)
```

```
      analogWrite(10, 235);
    else
      analogWrite(10, (custom_temp-room_temp)*40);
    //analogWrite(10, (custom_temp-room_temp)*5); //multiplied by 5 so max for resistor is not reached (in case it overheats)
    analogWrite(9, 0);
  }
  else{
    analogWrite(9, 0);
    analogWrite(10, 0);
  }
}
else
{
  analogWrite(9, 0);
  analogWrite(10, 0);
}
//output transducer and fans
if (custom_hum == 0)
{
  digitalWrite(6, LOW);
```

```
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
}
else
{
 if (custom_hum > (rh + rh*0.02)) //transducer needs to turn on
  {
    if (custom_hum - rh > 8)
    {
      digitalWrite(6, HIGH);
      digitalWrite(7, LOW);   // set the LED on
      digitalWrite(8, HIGH);
    }
    else
    {
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
    }
  }
  else if (custom_hum < (rh - rh*0.02)) //fans needs to be on
```

```
  {
    if (rh - custom_hum > 3)
    {
      digitalWrite(6, LOW);
      digitalWrite(7, HIGH);   // set the LED on
      digitalWrite(8, LOW);
    }
    else
    {
      digitalWrite(6, LOW);
      digitalWrite(7, LOW);
      digitalWrite(8, LOW);
    }
  }
  else
  {
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
  }
}
```

```
  //delay(10000);


  Serial.print(room_temp);
  Serial.print(" ");
  Serial.println(rh);
}


void update_data(){
  byte_i = Serial.available();
  if (byte_i > 0){
    switch (custom_data){
      //case 0 chooses which value will be updated
      case 0: serial_read = Serial.read();
            if (Serial.available() > 0){
              Serial.println("there is no such thing. Input again.");
              Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");
            }
            else{
              if ((serial_read > 48) && (serial_read < 52)){
                custom_data = serial_read - 48;                    //update custom_data
                Serial.print("Please input a value for ");
```

```
        if (custom_data == 1)
          Serial.println("brightness (between 0 to 7):");
        else if (custom_data == 2)
          Serial.println("temperature (between 10 to 39 degC):");
        else Serial.println("humidity (between 0-100 % RH");
      }
      else{ Serial.println("there is no such thing. Input again.");
            Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");
      }
    }
    Serial.flush();
    break;
//case 1 updates the light value
case 1: serial_read = Serial.read();
    if (Serial.available() > 0){
      Serial.println("there is no such thing. Input again.");
      Serial.println("Please input a value for brightness:");
    }
    else{
      if ((serial_read > 47) && (serial_read < 56)){
        custom_light = serial_read - 48;
```

```
        custom_data = 0;
        Serial.println("Brightness value updated.");
        Serial.println(custom_light);
        Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");
      }
      else{ Serial.println("there is no such thing. Input again.");
            Serial.println("Please input a value for brightness:");
      }
    }
    Serial.flush();
    break;
//case 2 updates the temperature value
case 2:
{
    float temp_temp,super_temp;
    int flag = 0;
    int i;
    for(i = 1; (i < 5) && (Serial.available() > 0) && (!flag); i++){
      serial_read = Serial.read();
      Serial.println(serial_read);
      if ((i == 1) && (serial_read > 48) && (serial_read < 52))
```

```
    temp_temp = (serial_read - 48) * 10; //first digit of temp (1 to 3)
  else if ((i == 2) && (serial_read > 47) && (serial_read < 58))
    temp_temp = temp_temp + serial_read - 48; //second digit of temp (0 to 9) this and line above give range of 10 to 39 degC
  else if ((i == 3) && (serial_read == 46))
  {} //decimal
  else if ((i == 4) && (serial_read > 47) && (serial_read < 58)){ //point after decimal
    super_temp = serial_read;
    super_temp = (super_temp - 48) / 10;
    temp_temp = temp_temp + super_temp;
  }
  else flag = 1;
}
if (((Serial.available() == 0) && (i == 3) && (!flag)) || ((Serial.available() == 0) && (i == 5) && (!flag))) {
  custom_temp = temp_temp;
  custom_data = 0;
  Serial.println("Temperature value updated.");
  Serial.println(custom_temp);
  Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");
}
else{
```

```
      Serial.println("there is no such thing. Input again.");
      Serial.println("Please input a value for temperature:");
    }
    Serial.flush();
    break;
}
//case 3 updates the humidity value
case 3: if (Serial.available()){
      delay(5); //make sure we have all the data
      int i=0;
      while(i<5){
        reading_Array[i] = Serial.read();
        i++;
    }
    Serial.flush();


    serial_read = atoi(reading_Array);
    Serial.println(serial_read);
    if (Serial.available() > 0){
      Serial.println("there is no such thing. Input again.");
      Serial.println("Please input a value for humidity:");
```

```
        }
        else{
          if ((serial_read > 0) && (serial_read < 100)){
            custom_hum = serial_read;
            custom_data = 0;
            Serial.println("Humidity value updated.");
            Serial.println(custom_hum);
            Serial.println("Choose a thing to update(1 for light, 2 for temp, 3 for hum):");
          }
          else{ Serial.println("there is no such thing. Input again.");
                Serial.println("Please input a value for Humidity:");
          }
        }
        Serial.flush();
        break;
    }
  }
 }
}
```

The italics portion of the code relates to the humidity section. The variable hum_pin was declared to communicate with the Arduino board that the humidity data will be from input pin two. The custom_hum variable was declared to read in the user input. The read_Array variable

was declared to read in the user input variables as characters. They are converted to integers later on.

Method: void setup()

This method is a necessary method which is used in the Arduino platform. It sets the different Arduino pins as input or output. Pin two was declared as input, whereas pins six, seven and eight were declared as outputs. (Steiner & Martins, 1991)

Method: void loop()

This method is a necessary method in the Arduino platform. This method is the main method which keeps looping. For the humidity, the code was based on "if" statements. If the user inputted value was eight units greater than the sensory value, the transducer would turn on. If the user inputted value was three units less than the sensory value, the fans would turn on. If the user inputted value was anything other than the mentioned conditions, nothing would turn on. There is also a "check" if statement so that the code only functions when the user has inputted a value.

Method: void update_data()

This method is updates the light, temperature and humidity values that the user inputs. However, since the user inputted values are read in as binary, modifications need to be made to make them recognizable with the code. The humidity portion of this method functions to convert those binary digits to integers. Those binary numbers are read in and stored as individual characters. These characters are converted to integers with the "atoi" function. If the user inputted number is within the range from zero to one hundred, the variable custom_hum will store it. Otherwise, an error will appear on the serial monitor.

Appendix C – Laws and Theory

Law of Continuity

There were several holes drilled into the water reservoir from the top. These holes had a size of a small pea. The water reservoir acted as a reservoir to store all the humidity that was being generated. The main hole, which connected the water reservoir to the prototype room, was drilled to provide the humidity a route to follow. However, it was necessary to drill holes onto the top of the water reservoir. The Law of Continuity states that the mass flow rate of air entering an object must match the mass leaving the object. Therefore, the additional holes allowed for air flow to occur from the outside. As the humidity left through the bigger hole, the little hole replaced that air within the water reservoir.

BJT Calculations

To calculate the current across the base node:

$I_B = V_{Arduino} / R$

$= 5/1200$

$= 0.00416$ A

To calculate the current going across the load

$I_C = \beta * I_B$

$= 40 * 0.00416$

$= 0.166$ A

The fan required 0.12 A, as per the specs, therefore 0.166 amps theoretically sufficed. The resistance was chosen to be 1.2k to reach this current value.

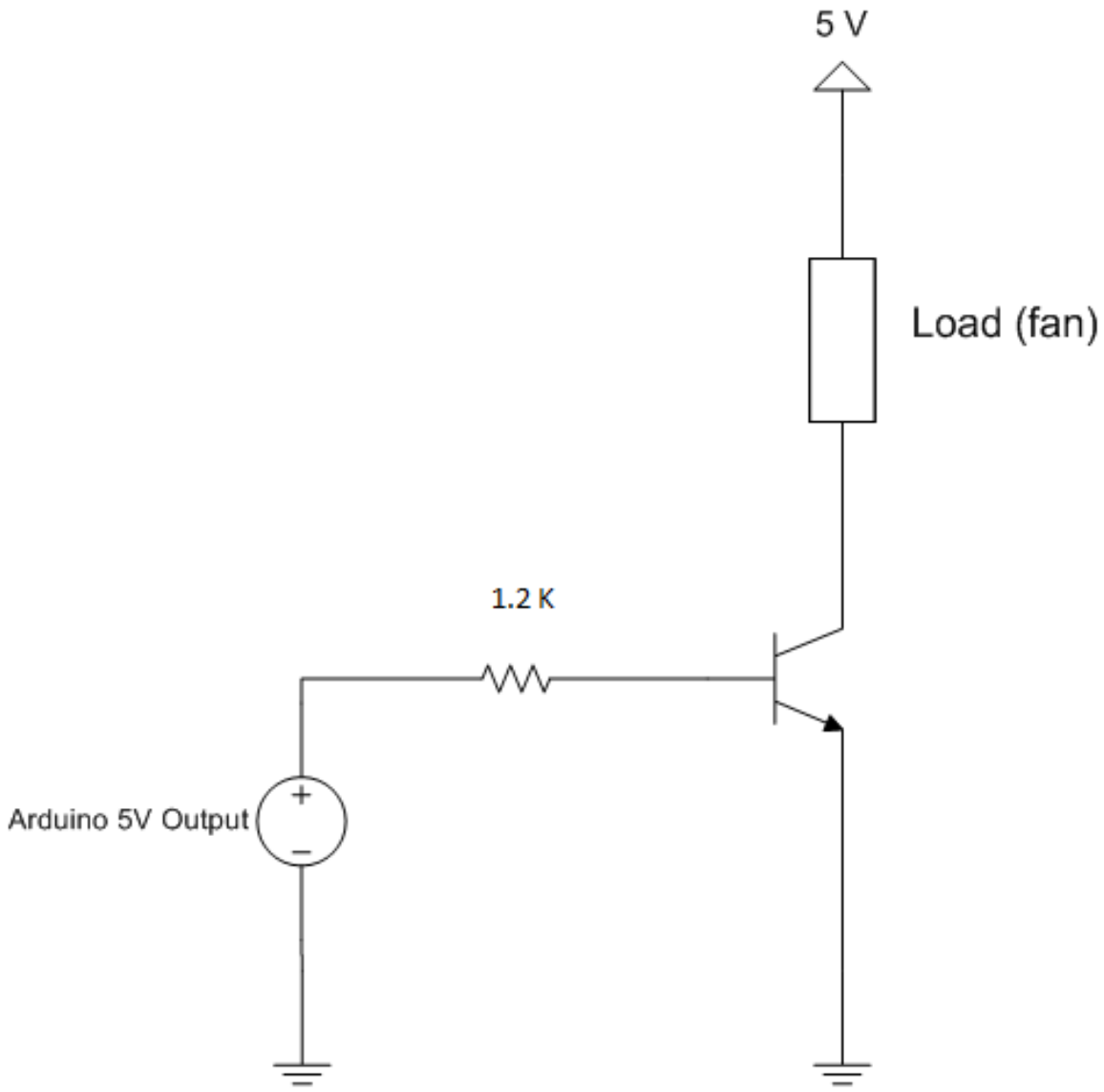


Figure 25: Circuit drawing for the BJT switch

Overall Schematic of the Circuit Design

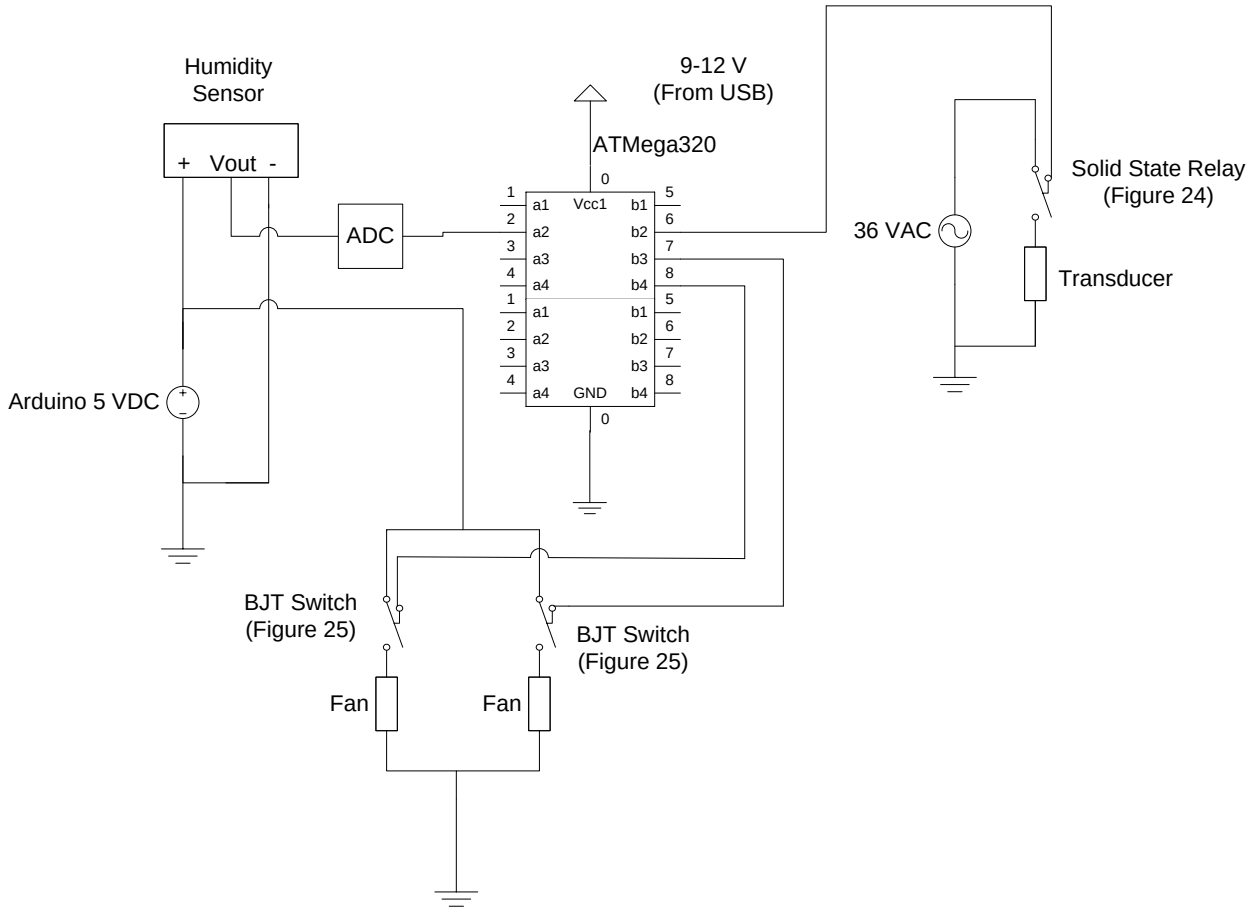


Figure 26: Schematic of the Humidity Portion

The diagram represents a high level view of how the circuit and its corresponding circuit elements are positioned. The two switches for the fans are controlled through the BJT circuit (see figure 26). The transducer is controlled by the solid state relay.

Output of Humidity

In order to compare the different The ATMega328 microprocessor has a 10 bit ADC. This splits the incoming voltage into 1023 equal digital portions. Therefore, in order to convert back to originally received analog voltage, the digital value was multiplied by 5 and divided by 1023. [10]

Analog Voltage = (Input Digital Voltage * 5) / 1023 (1)

In order to convert this voltage into its corresponding RH value, the output table was used. Please refer to table 3.

	30%RH	40%RH	50%RH	60%RH	70%RH	80%RH
808H5V5	1.73V	2.08V	2.41V	2.72V	3.01V	3.30V

Table 3: Output of Humidity

This table corresponds to Figure 16. Due the output being linear, an equation of a line can be generated. [9]

To calculate the slope of the line:

Slope = (3.3 – 1.73) / (80-30)

= 0.0314

So far, the equation looks like the following:

Output Voltage = 0.0314*RH + y-intercept.

To calculate the y-intercept:

RH = 30

Output Voltage = 1.73

With a simple calculation, it is determined that the y-intercept is 0.788. Thus, the final equation has the form:

Output Voltage = 0.0314*RH + 0.788 (2)

To calculate the RH value, equation 1 is modified to have the form:

RH = (Output Voltage – 0.788) / 0.0314 (3)

Equations 1 and 3 are implemented in the code. These equations allow the sensory value to be recognizable as an RH value. This can be easily compared to the user input value.

Alternative Design for Humidity

An alternative design for the humidity was also considered. In this design, the water reservoir was placed inside the cube prototype. Humidity would then spread out in the room directly from the water reservoir. In contrast, the current humidity model has the water reservoir outside the cube prototype. With this model, a pipe is attached from the water reservoir to the cube to effectively transfer the humidity in the water reservoir to the cube. Advantages for the former model were that humidity was provided directly in the room. It did not have to be through another medium and travel a distance, through the pipe, to get to its desired location. Disadvantages were mainly related to control. When the ultrasonic transducer turns off, there is still humidity remaining in the water reservoir. Therefore, this extra undesired humidity would contribute to increasing the overall humidity in the room. Furthermore, the water reservoir's lid would have to be absent so the water can be exposed to the surrounding air. Even when the transducer is off, the water will evaporate and cause an overall increase in the room's humidity. The latter design model's main advantage was control. The water reservoir was placed outside the cube so there was more control on how much humidity will be entering the room. The only disadvantage was that there could be trouble for all of the humidity to effectively get to the cube prototype. However, due to the fact there was a lot of humidity produced by the transducer, this was a non factor. This projects' task was to have a simple yet very efficient and controlled project. Thus, the latter design concept was implemented.

References

1. Steminc - Steiner & Martins (1991). Wide Part Number: SMUTK1660RS111. Retrieved October 25, 2009, from Steminc: http://www.steminc.com/

2. Swenson, J; *Ask A Scientist* (2005, Nov) "Law of Continuity, Bernoulli, Contradiction". Retrieved October 25, 2009, http://www.newton.dep.anl.gov/askasci/eng99/eng99439.htm

3. Liddament, M.W.; *Veetech Ltd*. Ventilation Energy and Environment. Retrieved January 02, 2010, from Veetech Ltd. http://www.veetech.org.uk/tutorial__1____calculating_flow_.htm

4. Devlin, Ann S.; Arnell, Allison B. *ENVIRONMENT AND BEHAVIOR* "Health Care Environments and Patient Outcomes", Vol. 35 No. 5, September 2003 665-694

5. UMHS Newborn Care Committee. (2005, May). Your Baby, Jaundice, and Phototherapy. Retrieved October 9, 2009, from http://www.med.umich.edu/1libr/pa/umphototherapy.htm

6. Tatsiopoulous, C; Ktena, A. "A smart Zigbee based Wireless Sensor Meter System" Retrieved March 1, 2010. [Online]

7. "Your Baby, Jaundice, and Phototherapy," *University of Michigan Health System,* May 2005. [Online]. Available: http://www.med.umich.edu/1libr/pa/umphototherapy.htm. [Accessed: Oct. 07, 2009].

8. Y.W. Bai and Y.T. Ku, "Automatic Room Light Intensity Detection and Control Using a Microprocessor and Light Sensors," *IEEE Transactions on Consumer Electronics,* vol. 54, no. 3, p. 1173, Aug. 2008. [Online]. Available: IEEE Xplore, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4637603&isnumber=4637567 . [Accessed Oct. 07, 2009].

9. Sensor Element. (2000, November). *Humidity Sensor*. Retrieved December 21, 2009 http://www.sensorelement.com/humidity/808H5V5.pdf

10. Robot Shop. (2009). *Arduino Duemilanove*. Retrieved December 4, 2009, from Robot Shop: http://www.robotshop.ca/arduino-duemilanove.html

11. Jia, Yi; Heiss, Michael; Fu, Qiuyun; Gay, Nicholas A. "A Prototype RFID Humidity Sensor for Built Environment Monitoring", DOI 10.1109/ETTandGRS.2008.35

12. Patriciu, A. Electrical Engineering 4BD4. Lecture Notes, Sept-Dec 2009

Vitae

NAME:	Adeel Alam
PLACE OF BIRTH:	Rawalpindi, Pakistan
YEAR OF BIRTH:	1988
SECONDARY EDUCATION:	John Fraser Secondary School (2002 – 2006)
HONOURS and AWARDS:	Honour Roll 2003, 2004, 2005, 2006
	Ontario Scholar 2006
	McMaster University Entrance Scholarship 2006
	Queen Elizabeth Scholarship 2006
	Dean's Honour List 2007, 2009
	University Senate Scholarship 2009