Data Logger System

Introduction:

The main goal of our project was to assemble a data logger system. Unlike everyone else, we planned to execute the project by practical methods and go for a complete hardware outcome rather than just a software simulation. The tools used along the process are basic hardware devices including sensors and development board on the hardware side as well as considering applications like Arduino and LabVIEW to simulate the program.

In the modern world of technologies, everything is designed to be autonomous to increase the comfort of human race along with increased efficiency and responsibility. We opt for developing a module that will automatically put a rest to the power supply of a factory for a specific cool down period when the temperature or humidity is higher and poses a threat to the industry as well as turning it on after a specific interval of time autonomously. Recently we have observed numerous incidents happening in the industrial sector of the country resulting in compromisation of innocent lives. If a person is designated to turn off the switch of a nuclear reactor and even if he is late by no less than 30 seconds, which is not very unusual, as we know '*To err is human*', the consequences can be fatal. Whereas all of this could have been easily avoided if we had designed an autonomous system just for the serving this purpose. So, for efficient along with responsible use of technology we have come up with the idea of developing a module that will serve the purpose.

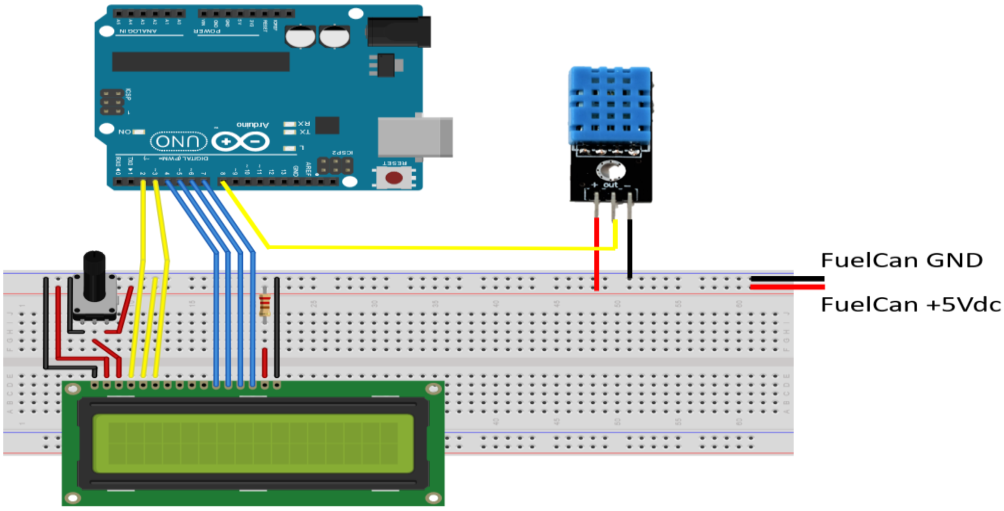
We plan to design a module that will be placed near the source of heat in a factory and if the figures exceed the limits of an admissible value, the power supply will be cut down for a specific period and automatically restarted afterwards. Not only temperature, we can consider humidity as a factor of this module as well. If there is a drastic change in the atmosphere inside the factory, the device will consider that as a malfunction and will turn off power to avoid any accidents and there will be a manual check of the machine for finding any errors.

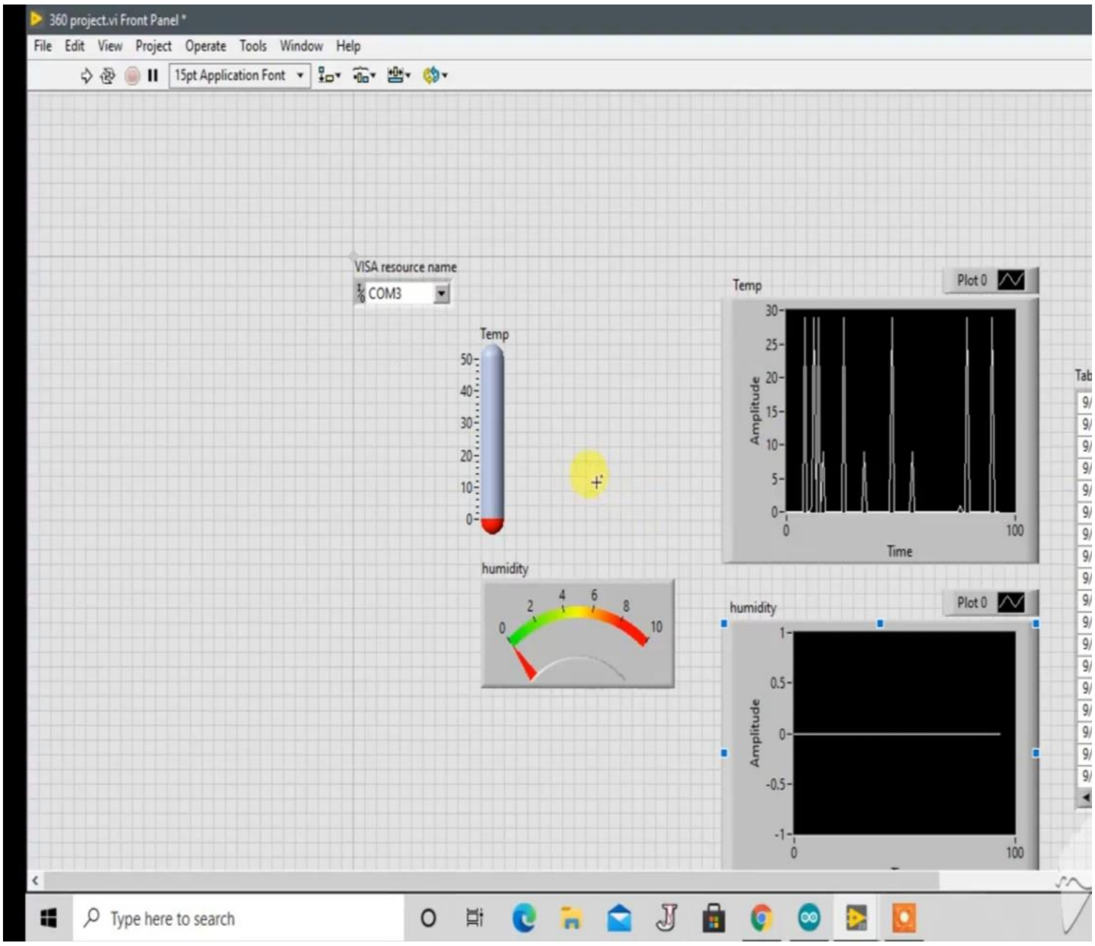
If we can monitor the temperature and humidity from close quarters and get the response through internet sitting at home, the owners can worry less about their industry. They do not have to stress about mismanagement of the employees or their time management and can easily rely on technology.

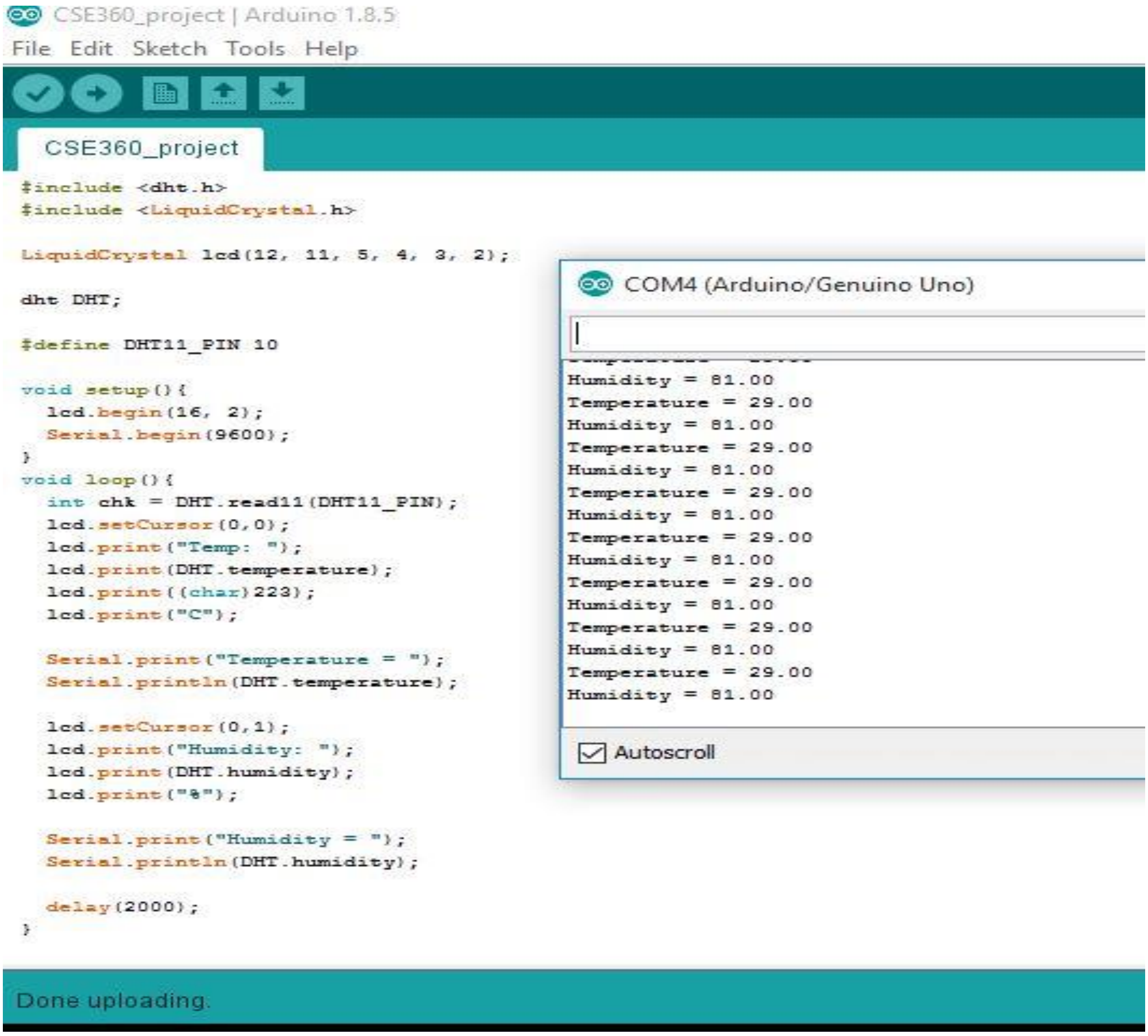
Methodology:

To visualize the project and physically implement it in day-to-day life we have used a few sensors and a development board. We plan to simulate the project with the help of LabVIEW, design the circuitry in Proteus. Other than that, we have use Arduino IDE to compile our program for the development board.

- **Product Architecture:** We have used development board Arduino UNO that uses the ATmega328 IC to operate and a DHT11 sensor more popularly known for measuring temperature and humidity of the surrounding atmosphere at close quarters.


FuelCan GND
FuelCan +5Vdc


360 project.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
VISA resource name
COM3
Temp
50
40
30
20
10
0
humidity
0
2
4
6
8
10
Temp
Plot 0
30
25
20
15
10
5
0
Amplitude
0
100
Time
humidity
Plot 0
1
0.5
0
-0.5
-1
Amplitude
0
100
Tab
Type here to search



CSE360_project | Arduino 1.8.5
File Edit Sketch Tools Help
CSE360_project
COM4 (Arduino/Genuino Uno)
☑ Autoscroll
Done uploading.

```
#include <dht.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

dht DHT;

#define DHT11_PIN 10

void setup(){
  lcd.begin(16, 2);
  Serial.begin(9600);
}
void loop(){
  int chk = DHT.read11(DHT11_PIN);
  lcd.setCursor(0,0);
  lcd.print("Temp: ");
  lcd.print(DHT.temperature);
  lcd.print((char)223);
  lcd.print("C");

  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);

  lcd.setCursor(0,1);
  lcd.print("Humidity: ");
  lcd.print(DHT.humidity);
  lcd.print("%");

  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);

  delay(2000);
}
```

```
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
Temperature = 29.00
Humidity = 81.00
```

We have connected the DHT11 data input into the digital signal pin 10 of Arduino UNO. We are using digital pins from the Arduino to receive the output neglecting the analog pins, which are considered to be less efficient in terms of digital data. We establish power supply in the DHT11 sensor via the 5V pin of the Arduino UNO and complete the circuit by connecting the ground of the sensor to the ground of the development board.

- **Interface Design and Overview:** The well-known sensor for collecting temperature and data i.e. the DHT11 sensor is used for the project because it is easily available in the market. Not only that, it is cheap and very efficient in term of giving reasonable output. It is one of the most formidably used sensors for serving this sort of purpose. It can give correct data of not only the temperature but also the humidity and that is the reason why we ae selecting that module for our project.

The programming is mostly done on Arduino IDE. It is a well-known integrated development environment for embedded system projects. It works on a language, which is often known as Modified C. The platform or operating system we are using is Windows also known as Microsoft OS. The vision of our project is to firstly implement it on a small scale. There will be a lot a bugs and errors in a new project. Therefore, after fixing those issues we will plan to go for a large scale production and support to the community.

The workflow of the project is a bit complex, as we have to integrate a number of things together. Starting from Arduino, we have to compile and upload the code on the development board. We are using DHT sensor, which has variations of DHT11 and DHT22, which serves a better purpose, but for our project, we can get away with DHT11. The serial begin of the code starts from 9600 as it is the most commonly used baud rate in a device. We have given a delay of 1000 ms to show the data at an interval of 1 second. So, ultimately our data will be automatically shown after every 1 seconds interval and updated on the any device we choose to display the output such as serial monitor or an LCD display or even another GUI.

The LabVIEW works based on G-Programming. Here we can show the output on a detailed graphical user interface or GUI where we can add different sorts of meters gauges to display the output in a user-friendly manner. We can show the output via charts or graphs as well anyway the client feels suitable.

Results:

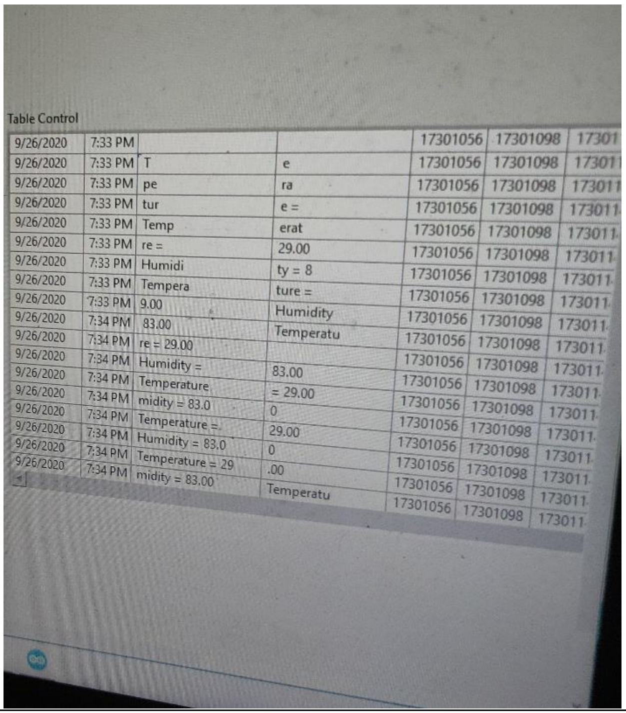
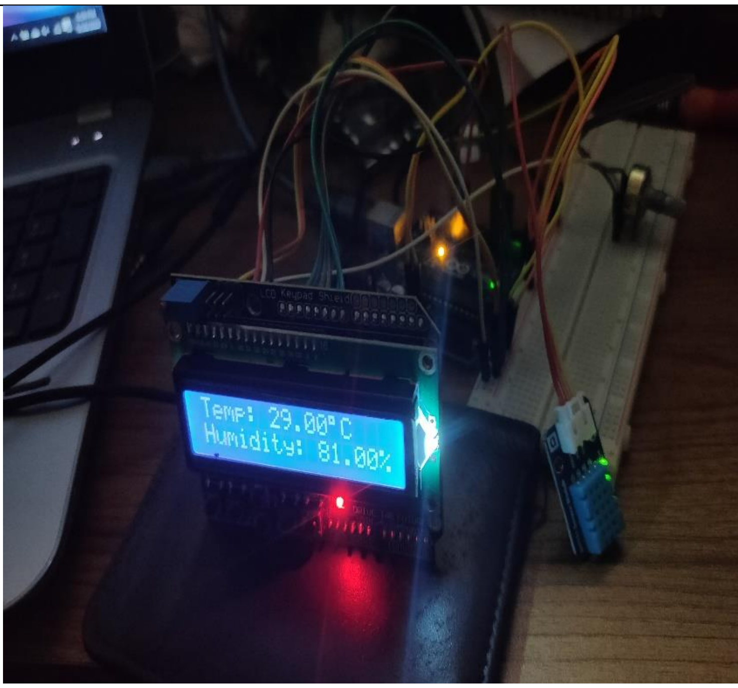


Table Control

9/26/2020	7:33 PM			17301056	17301098	17301
9/26/2020	7:33 PM	T	e	17301056	17301098	173011
9/26/2020	7:33 PM	pe	ra	17301056	17301098	173011
9/26/2020	7:33 PM	tur	e =	17301056	17301098	173011
9/26/2020	7:33 PM	Temp	erat	17301056	17301098	173011
9/26/2020	7:33 PM	re =	29.00	17301056	17301098	173011
9/26/2020	7:33 PM	Humidi	ty = 8	17301056	17301098	173011
9/26/2020	7:33 PM	Tempera	ture =	17301056	17301098	173011
9/26/2020	7:33 PM	9.00	Humidity	17301056	17301098	173011
9/26/2020	7:34 PM	83.00	Temperatu	17301056	17301098	173011
9/26/2020	7:34 PM	re = 29.00		17301056	17301098	173011
9/26/2020	7:34 PM	Humidity =	83.00	17301056	17301098	173011
9/26/2020	7:34 PM	Temperature	= 29.00	17301056	17301098	173011
9/26/2020	7:34 PM	midity = 83.0	0	17301056	17301098	173011
9/26/2020	7:34 PM	Temperature =	29.00	17301056	17301098	173011
9/26/2020	7:34 PM	Humidity = 83.0	0	17301056	17301098	173011
9/26/2020	7:34 PM	Temperature = 29	.00	17301056	17301098	173011
9/26/2020	7:34 PM	midity = 83.00	Temperatu	17301056	17301098	173011
				17301056	17301098	173011



Finally, with numerous efforts and trials we have finally managed to develop the system. The data is displayed on the serial monitor and it can also be provided in an LCD display. The results behave like standard room temperature from 28 to 30 degrees and humidity of 80% to 82%. As far as we are concerned these are the regular temperature and humidity of an average day of the region of earth we live in so we believe the results are justified.

Conclusion:

The prime purpose of our Data Logger System was to demonstrate that we can easily avoid certain accidents just by designing a simple module that will serve a small purpose of just maintaining the power supply of a factory. If the temperature of the source of heat inside an industry goes higher at a certain time, the module will automatically send it to a cool down state. It is illogical to put human labor to supervise this temperature 24/7 and maintain efficiency.

The major thing we learned are how to code on a different IDE, how to implement a hardware project, learning circuit design in Proteus as well as learning G-programming with LabVIEW. It was hard for us to integrate the entire system together because we were short of hardware resources on our computers, thus it kept malfunctioning.

We believe our system is efficient and cheap. An embedded system like ours are not available in the market as of now. However, the industries are still at a major threat of these sort of accidents. Therefore, if we the investors come forward to support our work and sponsor it we believe our project will not only be efficient but also save lives one day.

We believe a major portion of the problem will be solved by implementing our module. It is a cheap and affordable product. So small companies will not be afraid to afford it for the sake of their industry.

We faced many difficulties managing the hardware resources. Ultimately, we were able to add up all the components for the project. The integration part was hard as well since we have not done anything like this before. We faced issues like the sensor not working, not giving value on serial monitor, giving garbage value and whatnot. Nevertheless, in the end we were able to fight back all the circumstances with results.

We plan to make this product connected through the internet in near future. For that, we will use the Raspberry Pi as our server. We will use ESP8266 for taking the data. Since ESP8266 already has a built in Wi-Fi module in it so we can fetch the data to a server. After that by using the client-server protocol in Raspberry pi and other protocols like MQTT or Google's very own Firebase, we plan to give real time output to our clients.

We can apply this module in many other sectors such as dairy farms even fishing farms. We can add more sensors such as pH sensor to the module and check for a regular update of the water quality of in fisheries for suitable growth. At a certain point and time, this can be implemented in many other sectors.

We came across a lot of different new softwares and hardwares through this project. We came to know about Proteus, which is used to design circuit. Now we will not go towards complex wire connection anymore, rather we will design our own circuits and minimize the wiring. We have learned about real life hardware products like Arduino, which works like nothing short of magic. We learned about LabVIEW to make our own GUI and simulate the output which will be very helpful in future years to come for demonstrating a project.