International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

LabVIEW Based Temperature Controller using Arduino

V. Prakasam, Vikas Kumar Tiwari, K.R.AnudeepLaxmiKanth

Abstract—The main aim of this paper provides to existing two types of temperature control system that are automatic temperature control system and manual control system using LABVIEW and Arduino. In this paper, we are designated LABVIEW based temperature controller using Arduino. In this paper, we are operated 5 volt cooling fan when the temperature goes to above the fixed point. Here, we will use LM35 temperature sensor and we are using three different LEDs or single RGB LED to indicate the range of temperature. The design and working operation will classified into three different categories that are first one is when the temperature is above the mark (set) point, in this case the Red LED will glows, second one is when the temperature exists between higher mark point and lower mark (set) point, in this case the Blue LED will glows and third one is when the temperature set the below set point, in this case the Green LED glows. The temperature reading also stored in Microsoft excel sheet. Here we used LABVIEW 2017 software version for designing temperature controller system. The remaining software's also used for designing temperature controller system but the LabView is the simplest way of remaining all software's.

Index Terms: LABVIEW, Arduino, temperature, LM35, RGB, and LED.

I. INTRODUCTION

The principle of division of labour is noticeable with technology to contribution man in the character of tasks with slight struggle by control systems. Normal control system processes the controlled mutable, associates that dimension with the usual point or position, and if there is a variance between the two, variations its output signal to the manipulated variable in order to remove the error. This still gives huge benefits as great quantity of strain and complications which can be dull are overcome using a pertinent applied system. Therefore, from the principle of the control systems, this focused on the domain of Temperature Control system. Ideal plant temperature appropriate for both manufacture and storage of goods, is generally harmfully pretentious by too high or too low temperature which are unwieldy to adjust and control by conventional systems, hence the necessity for computerize and effective system growth to attain the task is will be required.

In various fields there is a necessity of observing particular physical constraints such as temperature, pressure, humidity, flow rate, etc. With respect to the amounts, spreading and noticed frequency of watched entities, there are dissimilar observing methods to get detention the measurements. It's a decision of the management whether to have a monitoring based on manual or automated basis. There are two types of temperature sensors that are Positive Temperature Coefficient (PTC) of Resistivity and Negative Temperature Coefficient (NTC) of Resistivity. The significance of thermistor sensor is a negative temperature coefficient of resistivity (NTC) type. So as temperature increases its resistance goes on decreasing. This sensor is a passive type of transducer so it wants an exterior power supply for its occupied.

Temperature control is significant for parting and return processes, and temperature must be preserved within restrictions to safeguard safe and reliable task of process apparatus. The temperature sensor is endangered from the process resources to check intervention with correct sensing and to remove impairment to the sensor. Thus, designated physically dense, chemically resistant barrier happens between the process and sensor.

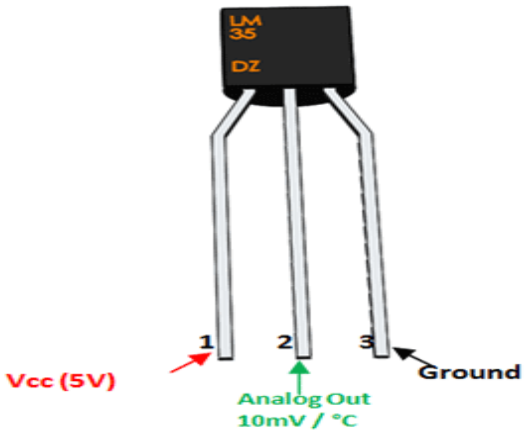


Figure 1: LM35 Temperature sensor.

The LM35 succession are precision united circuit temperature strategies with a voltage of output directly proportional to centigrade temperature. The figure 1 shows the LM35 temperature sensor, the LM35 does not need any peripheral adjustment or decoration to deliver classic accuracies of ±1/4 °C at room temperature and ±3/4 °C over a full -55 °C to 150 °C temperature choice.

Manuscript published on November 30, 2019.
* Correspondence Author
V. Prakasam*, Department of ECE,Vignan Institute of Technology and Science, Deshmukhi, Hyderabad,Telangana, India.
Vikas Kumar Tiwari, Department of ECE,Vignan Institute of Technology and Science, Deshmukhi, Hyderabad,Telangana, India.
K.R.AnudeepLaxmiKanth, Department of ECE,Vignan Institute of Technology and Science, Deshmukhi, Hyderabad,Telangana, India.

There are 3 pins available in LM35 temperature sensor that are left pin (pin1) is V_{cc}, middle pin (pin2) is connects to the analog pin A0 and right pin (pin3) is ground.

The proposed work involves both hardware as well as software as monitors.

Hardware: Arduino Uno Board, LM35 Temperature Sensor, DC fan (5V), RGB LED or three separate LEDs.

Software: LabVIEW installed Laptop, LIFA, Arduino Software, MakerHub LabVIEW Packages.

The executed system, LM35 sensor provides the sensed temperature which is associated to LabView through MakerHub open LINX. Based on the sensed temperature LabView displays the deviations in it on the graph chart. As the temperature go above above the set point LabView provides instructions to the Arduino Uno board and goes the LED as well as the LabViews front panels Boolean LED signal come to be on.

II. HARDWARE DESIGN

The block diagram of proposed system shown in figure 2. The proposed system implementation consists of LabVIEW installed Laptop, Arduino Uno board, LM35 Sensor, three LEDs or RGB LED, DC 5V fan and connecting wires.

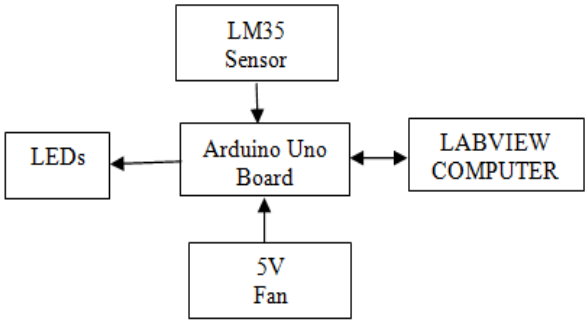


Figure 2: Proposed system block diagram

LM35 Sensor: in the proposed system, LM35 pin1 is connected to bread board power supply and then connected to Arduino uno 5 v supply. LM35 pin2 is connected to analog A0 port and LM35 pin3 is connected to Arduino uno ground port.

Arduino Uno Board:

Figure 3: Arduino Uno Board

The Arduino Uno is a microcontroller board based on the ATmega328. The Arduino Uno consists of

- Input/output pins: 14 (of which PWM outpt pins: 6, Analog Input pins: 6).
- Ceramic resonator: 16 MHz
- USB connection
- Power jack
- ICSP header, and a reset button.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-9V
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	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 1: Arduino Uno parameters.

LED or RGB LED: LED is abbreviated as light-emitting diode which is a two-terminal semiconductor light basis. LED act as a working principle of p–n junction diode, which produces light when stimulated. When a proper voltage is useful to the indications, electrons are capable to recombine with electron holes inside the device, discharging energy in the form of photons.

The RGB LED is a 5mm RGB LED, RGB means Red Green Blue. RGB LED has 4 pins. There three output one for each color with a common cathode. This LED is a pulse width modulation by mixing the various colors at various bright ness we get 257x257x257 different colors.

Here, the Red, Green and Blue LEDs one end is connected to Arduino uno pin 8, pin10 and pin12 and other end is connected to bread board ground. Bread board ground is connected to Arduino uno ground.

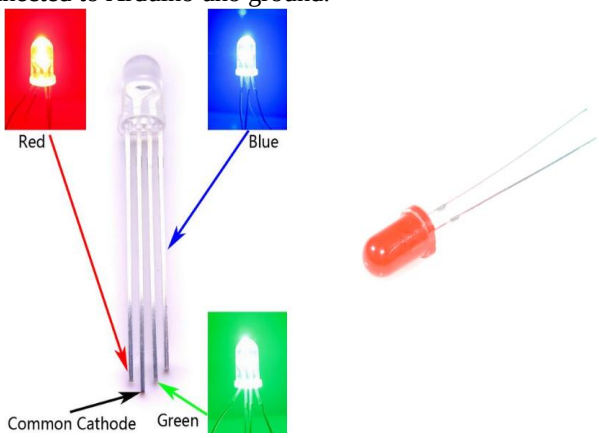


Figure 4: RGB LED and Simple LED

Cooling Fan: Here we are using DC cooling fan which has 5 volts. This fan is used to show the system output. This fan has two terminals that are positive and negative. The positive terminal is connected to Arduino board pin3 and negative terminal is connected to bread board ground. Once it getting input signal from the Arduino uno board microcontroller, it provide the air flow as the output.

The fan terminal is supported and joined to one of the LED logic pin design in the labview, which mutually start automatically at increase in temperature more than the set point.

Figure 5: DC 5 volt cooling fan.

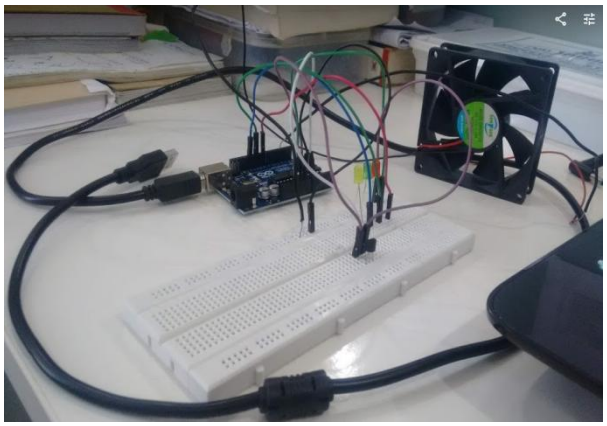


Figure 6: Hardware setup.

III. LABVIEW SOFTWARE DESIGN

In software section, the first step is LabVIEW programing VI to model the temperature from the building circuit divider and transform it into a temperature analysis in physical time. LIFA stands for LabView interface For Arduino (or LINX) which is the software that supports us interface LabView and the Physical Arduino Hardware design.

The front panel design permits us to control and observer the progression. It contains of software inputs and outputs that look like the physical controls of the hardware structure such as LEDs, sliders, buttons, speed controller and temperature graphic charts. Figure 7 shows the screenshot of the front panel design. The block diagram panel permits to design proposed system logic, which is the graphical platform that displays the data flow of the temperature control process.

First select the LINX MakerHub open, LINX serial port is connected to Arduino Uno. Next select the whie loop, whit in the loop select and drag the TMP35 sensor, the output LINX Resource connected to input to TMP35 sensor.

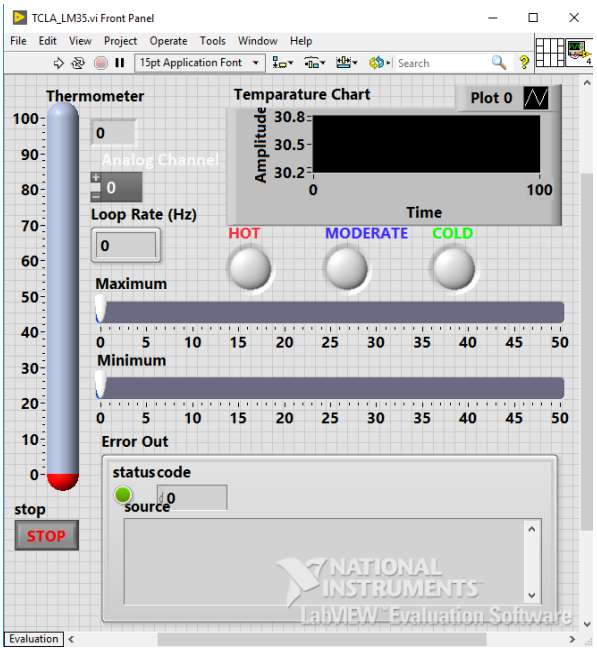


Figure 7: Temperature controller Front panel

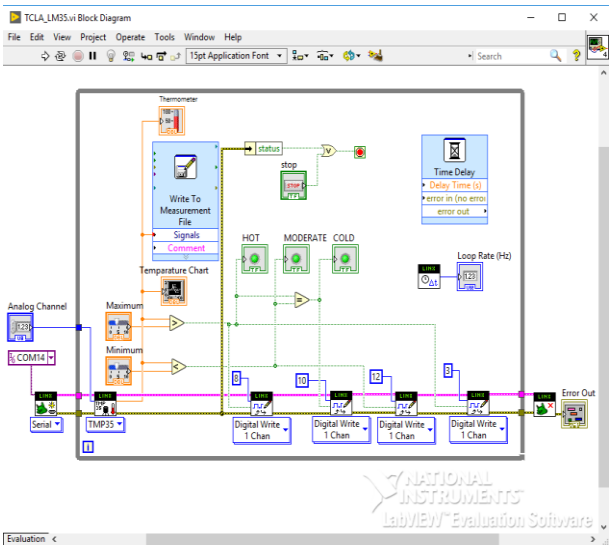


Figure 8: Temperature controller Block diagram panel

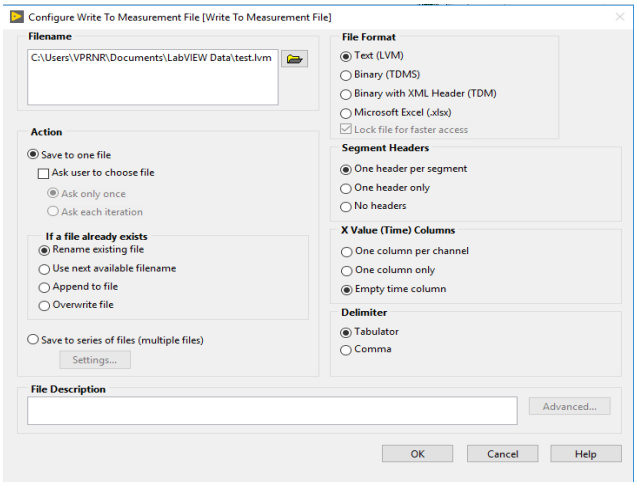


Figure 9: Write to measurement file functional palette.

The output of sensor is connected to the 'Write To Measurement File' labview block which store the various temperature reading result that is being exported into Microsoft Excel. The connection process of proposed system as shown in figure 8. The overall setup that is hardware and software design setup shown in figure 10.

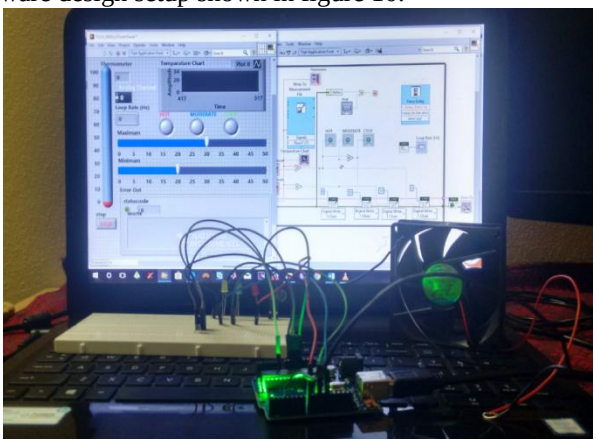


Figure 10: Temperature controller Hardware and Software implementation setup.

IV. ARDUINO INTERFACE WITH LABVIEW (LIFA AND LINX)

How to Interface Arduino with Labview:-

After front panel and block diagram panel design. In order to interface the Arduino uno microcontroller port, we must install LINX device driver. LabView communicate with the Arduino uno by its LIFA or LINX. The value found from the temperature sensor is showed in LabView part. LabView section modifies the communication by retrieving the temperature sensor of the embedded program using the install drive driver. The below process steps are followed by driver installation process.

Step 1: first download LIFA and MakerHub interface from VI package manager.

Step 2: LIFA don't install directly. If completed so an error will occur, like "VIPM could not continue".

Step 3: Open LabVIEW select Tools, next click on option and then click on VI server.

Step 4: After progress with on VI Server move down to check Machine Access address and then add manually machine access list. Make sure that you added "127.0.01", "localhost", "*" in the Machine access list, then click on ok.

Step 5: Now open downloaded VI Package Manager.

Step 6: Search for LabVIEW Interface for Arduino and double click on it to install it.

Step 7: Click on continue and after installation is, completed click on finish.

After completion of LIFA, the loop rate is selected by 2 Hz and then select the maximum value 35 and minimum value is 25 which is shown in figure 11. Before running LabView program, go to Tools → MakerHub → LINX → LINX Firmware wizard. In LINX Firmware wizard select the Device Family Arduino, Device Type is Arduino Uno, Firmware Upload Method is Serial/USB and then click Next button. Next select the Arduino uno port (COM13) then click the Next button, after that click the Next button. Finally click the Finish button. This process is shown in figure 12. Otherwise it display the below error out.

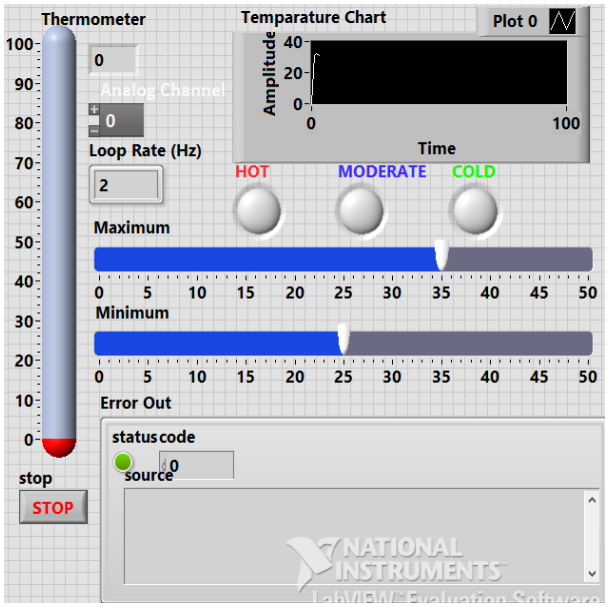


Figure 11: Temperature Controller setup values.

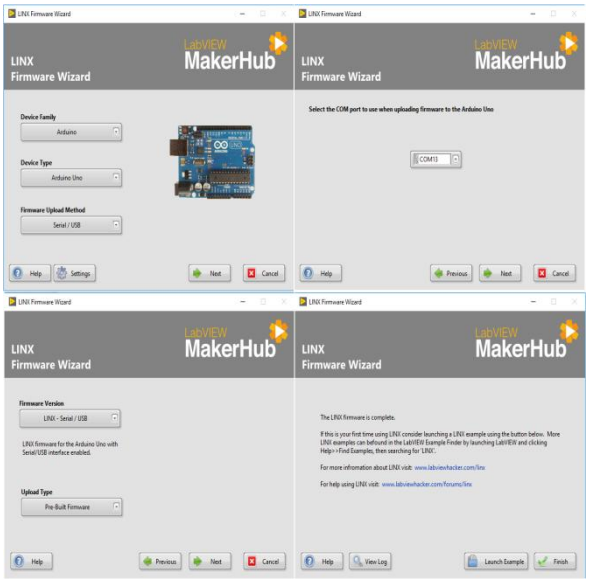
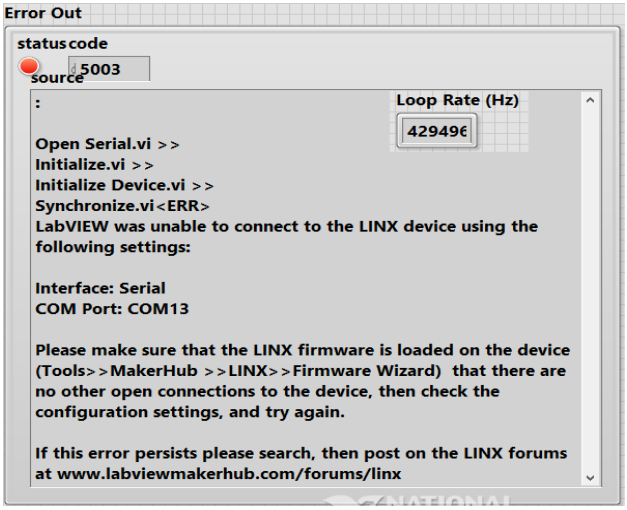


Figure 12: LINX MakerHub Process.



Retrieval Number: D5230118419/2019©BEIESP
DOI:10.35940/ijrte.D5230.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V. RESULTS AND DISCUSSION

After effective design and implementation of proposed system and examination run, the below list of figures and tables show the temperature variable range. The figure 13 shows the output product displays when the factories situation temperature is changing at 34^0C, although the temperature range condition is between 25^0C to 35^0C, so the cooling fan system unit does not run. The Green LED is blinking on front panel and Hardware setup which is shown in figure 13.

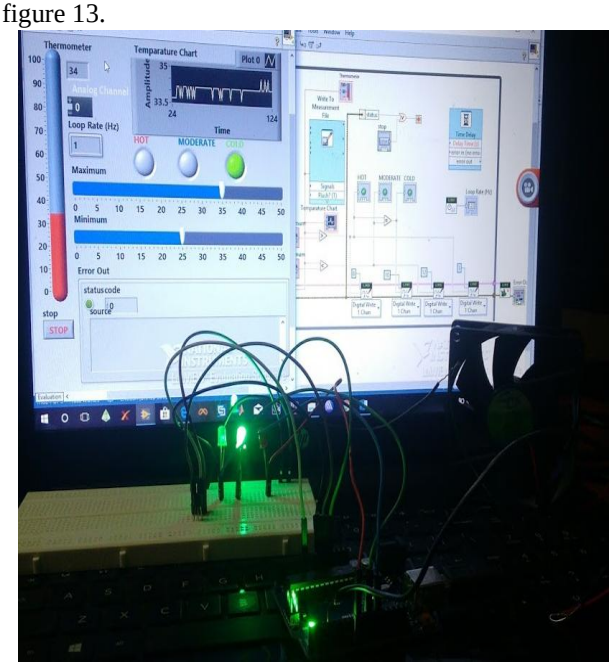


Figure 13: cold temperature

Here, we set the maximum temperature 35^0C, if it exceeds Red LED will glows on LabView front panel and hardware portion. Now, we are going to close my temperature sensor, the Thermometer reading exceeds the maximum temperature, we can also see that the fan is started, which is shown in figure 14. Figure 14 shows the output result when the factories setting temperature is fluctuating at 38^0C, while the requirement of temperature range is 25^0C to 35^0C.

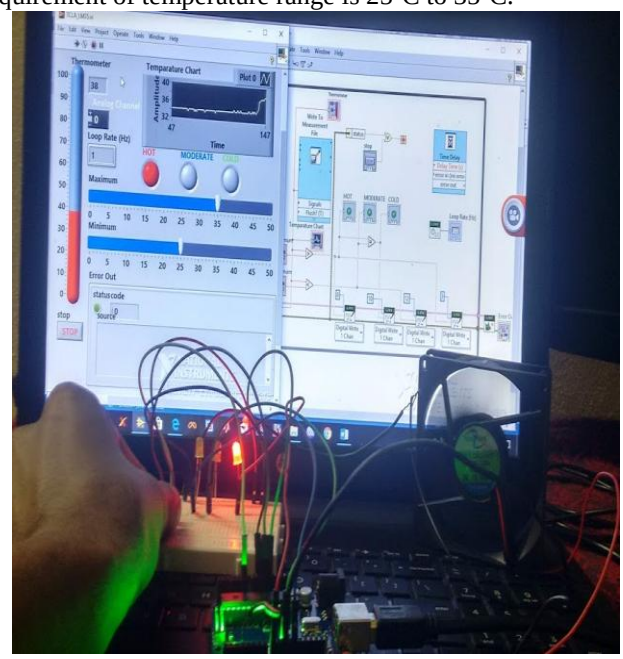


Figure 14: hot temperature

Now, we are removing fingers from my temperature sensor, the temperature is gradually reducing and cooling fan also turned OFF.

Figure 15 shows the output result, when the factories setting temperature is changing at 35^0C, while the requirement range of temperature is between 35^0C to 40^0C, so the cooling fan system unit does not run in this case. The Blue LED is blinking on LabView front panel and Hardware setup which is shown in figure 15.

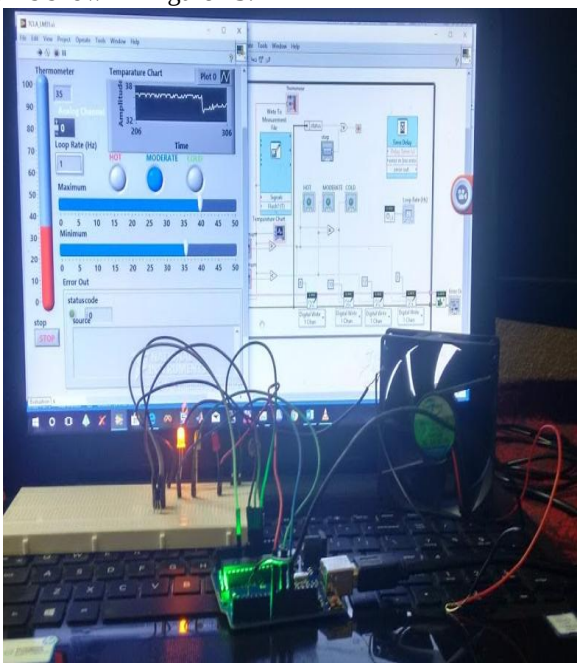


Figure 15: moderate temperature

Table 2: Data obtain during test

S.NO	DATE	TIME	TEMP ^{0}C
1	19-10-2019	19:37:06.543	33.691406
2	19-10-2019	19:37:14.120	34.179687
3	19-10-2019	19:37:31.509	33.691406
4	19-10-2019	19:37:43.501	34.179687
5	19-10-2019	19:39:43.582	37.109375
6	19-10-2019	19:39:54.588	38.574219
7	19-10-2019	19:40:33.585	36.621094
8	19-10-2019	19:42:34.280	35.15625
9	19-10-2019	19:46:00.989	35.644531

	A	B
1	Date and Time	Temp in 0C
2	19-10-2019 19:37:06.543	33.691406
3	19-10-2019 19:37:09.151	33.691406
4	19-10-2019 19:37:10.502	33.691406
5	19-10-2019 19:37:11.502	33.691406
6	19-10-2019 19:37:14.120	34.179687
7	19-10-2019 19:37:16.012	34.179687
8	19-10-2019 19:37:17.453	34.179687
9	19-10-2019 19:37:18.809	34.179687
10	19-10-2019 19:37:20.329	34.179687
11	19-10-2019 19:37:21.966	34.179687
12	19-10-2019 19:37:22.970	34.179687
13	19-10-2019 19:37:24.540	34.179687
14	19-10-2019 19:37:25.542	34.179687
15	19-10-2019 19:37:26.971	34.179687
16	19-10-2019 19:37:30.510	34.179687
17	19-10-2019 19:37:31.509	33.691406
18	19-10-2019 19:37:33.385	33.691406
145	19-10-2019 19:39:45.584	37.597656
146	19-10-2019 19:39:46.584	37.597656
147	19-10-2019 19:39:47.583	37.597656
148	19-10-2019 19:39:48.587	38.085937
149	19-10-2019 19:39:49.586	38.085937
150	19-10-2019 19:39:50.585	38.574219
151	19-10-2019 19:39:51.584	38.574219
152	19-10-2019 19:39:52.589	38.574219
153	19-10-2019 19:39:53.588	38.574219
154	19-10-2019 19:39:54.588	38.574219
155	19-10-2019 19:39:55.586	38.085937
156	19-10-2019 19:39:56.585	38.085937
157	19-10-2019 19:39:57.586	38.085937
158	19-10-2019 19:39:58.589	37.597656
159	19-10-2019 19:39:59.588	38.085937
160	19-10-2019 19:40:00.587	37.597656
505	19-10-2019 19:45:47.988	34.667969
506	19-10-2019 19:45:48.989	35.644531
507	19-10-2019 19:45:49.988	35.15625
508	19-10-2019 19:45:50.987	35.644531
509	19-10-2019 19:45:51.986	35.15625
510	19-10-2019 19:45:52.985	35.644531
511	19-10-2019 19:45:53.984	34.667969
512	19-10-2019 19:45:54.984	35.644531
513	19-10-2019 19:45:55.988	35.15625
514	19-10-2019 19:45:56.986	35.644531
515	19-10-2019 19:45:57.988	35.15625
516	19-10-2019 19:45:58.991	35.644531
517	19-10-2019 19:45:59.989	35.15625
518	19-10-2019 19:46:00.989	35.644531
519	19-10-2019 19:46:01.988	35.15625

Figure 16: Data Storage in Excel Sheet.

VI. CONCLUSION

The main intension of this paper is regulate the factory environment temperature automatically and manually and to store and display the on LabView.

1. The output product displays when the factories situation temperature is changing at 34^0C, although the temperature range condition is between 25^0C to 35^0C, so the cooling fan system unit does not run. The Green LED is blinking on LabView front panel and Hardware setup.

2. The output result when the factories setting temperature is fluctuating at 38^0C, while the requirement of temperature range is 25^0C to 35^0C. So the cooling fan system unit started automatically. The Red LED is blinking on LabView front panel and Hardware setup.

3. the output result when the factories setting temperature is changing at 35^0C, while the requirement range of temperature is between 35^0C to 40^0C, so the cooling fan system unit does not run in this case. The Blue LED is blinking on LabView front panel and Hardware setup.

We also recommend upcoming design to include other structures such as air condition and weather station controller and advance sensing unit.

REFERENCES

1. Jim Baker.How to call win32 Dynamic link Libaries(DLLs)from LabVIEW. National Instruments Notes 088. www.ni.com
2. Based on NImyDAQ Sound Processing System, College of Electronics and Information Engineering,Changchun University,Changchun,china.
3. Using LabVIEW to Measure Temperature with a Thermistor, C. Briscoe and W. Dufee, University of Minnesota.
4. P.A.Harsha Vardhini, Y.Murali Mohan Babu, A.Krishna Veni, "Industry Parameters Monitoring and Controlling system based on Embedded Web server", International Journal of Emerging Technologies and Innovative Research, Vol.6, Issue.2, page no. pp.80-84, February 2019.
5. Real Time Temperature Monitoring Using LABVIEW and Arduino, Vaibhav M. Davande1, Pradeep C. Dhanawade2, Vinayak B. Sutar3, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 3, March 2016.
6. Rajesh, S. K. (2013), ''Industrial Temperature Monitoring and Control System through Ethernet LAN ''International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 2 Issue 6 Page No. 1988-1991.
7. K. Chenna Kesava Reddy, P.Venkatrao, "Real Time Field Monitoring and Controlling System", International Journal of Recent Technology and Engineering (IJRTE), Volume-2, Issue-4, pp.-98-100, September 2013.
8. N.Koteswaramma, P.A.Harsha Vardhini, "Implementation of Arduino based Object Detection System", International Journal of Modern Electronics and Communication Engineering (IJMECE), pp.2018-211, Vol. 7, Issue.3, May 2019.
9. K.Murali Chandra Babu, P.A.Harsha Vardhini, N.Koteswaramma, "Design and Implementation of Arduino based Riders Safe Guard 2.0", International Journal of Innovative Technology and Exploring Engineering (IJITEE), pp.3078-3083, Vol.9 , Issue.1, Nov 2019.
10. V.Prakasam, P.Sandeep, P.A.Harsha Vardhini, "Snappy and Video Stream Edge Detection using LabView", International Journal of Mechanical and Production Engineering Research and Development (IJMPERD), pp.161-168, Vol. 8, Spl Issue 2, Nov 2018.