TEMPERATURE AND LEVEL CONTROL OF A PROCESS STATION USING LABVIEW

[1]Anusha James,[2] Prof. K. Radhakrishnan

[1]PG student,[2] Professor
[1]Department of Electrical and Electronics Engineering,
[1]MGM College of Engineering and Technology, Ernakulam, India

Abstract : Nowadays, the need for accurate control of process variables such as temperature and level is increased in industrial areas such as automobiles, factories, homes and businesses, and also the equipment used in industries. This paper describes the design and development of a feedback control system that maintains temperature as well as the level of a Liquid in a tank. The objective of the project is to design an interactive temperature controller system using a heater coil, and the level controller using only pump, in a cost effective manner. The heater change temperature according to the error signal by implementing the Proportional Control action. The project is implemented using LabVIEW and ARDUINO based DAQ.

***IndexTerms*- Arduino UNO, LM35 sensor, HC-SR04 sensor, LabVIEW**.

I. INTRODUCTION

The demand for accurate control of process variables such as temperature and level has conquered many of industrial domains such as process heat, automotive, industrial places, etc. The job of a typical control system is to maintain something that is variable in a desired set point. The common example is the temperature control system that found in most homes. A control system helps to keeps the air in a home at a certain temperature. It does this by sensing the ambient temperature, comparing it to the desired temperature, and producing a control action that will keep the temperature at the desired temperature. This paper describes a design for the continuous monitoring and control of temperature and level of the Liquid in a tank. The system will maintain the temperature and level of a liquid tank at a desired set point. Another goal is to incorporate a PC-based data acquisition system (DAQ) along with LabVIEW, a graphics-based programming language by National Instruments. LabVIEW has been used in many control systems applications. As a combination of various technologies and computer technology, virtual instrument [1] has opened up a new era of instrument technology. The basic idea of virtual instrument is to replace the traditional electronic instrument and gradually replacing the traditional instrument with the virtual instruments to implement some functions such as the collection, analysis, display and storage of data. According to the software development platform, virtual instrument perfectly integrates the computer hardware resources and instrument hardware. The concept of this paper is to create an Interactive Automatic Temperature as well as liquid level Control System to control the temperature and the level of a liquid tank.

II. BLOCK DIAGRAM DESCRIPTION

The model block diagram of the proposed system is shown in the fig1. The block diagram describes the overall working of the proposed system. The user set water level and temperature using the set point control input of GUI in LabVIEW, then the LabVIEW compare the process variables with the desired set point. If the level of the system is less than the set point, LabVIEW generates a controller output to arduino board via UART communication and this control output triggers digital output ports of arduino from high to low. If the output turned on the optoisolators in the relay control board, the board activates and thus relay energizes and water pump turned on by getting supply through relay contact. When the water level reaches the set point LabVIEW send a turn off command to arduino thus relay de-energizes and pumping stops. The level sensor HC-SR04 sends ultrasonic sound waves to the process tank and continuously monitors the echo signal to calculate the level of water in tank by calculating the time of wave travel and the distance of water from sensor. Then LabVIEW process the parameter with the set point, if the temperature is below the set point, LabVIEW generates controller output by activating digital output of DAQ. When it reaches the set point the controller holds the temperature by ON-OFF mechanism of heating coil. The individual description about each block is given in the following paragraphs.

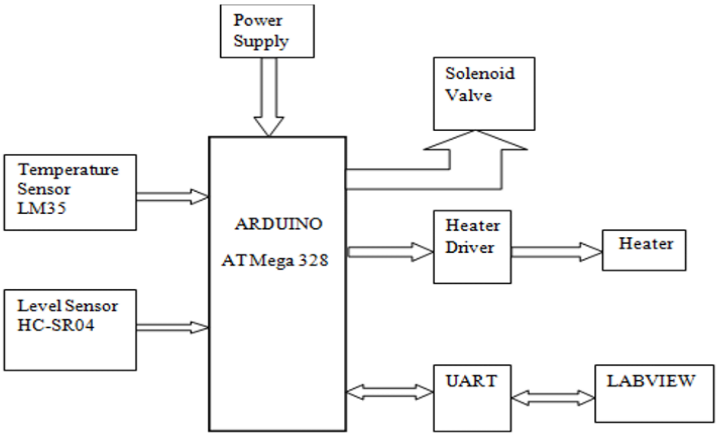


Fig.1 Block diagram of the system

2.1 LM35 temperature sensor

LM35 is a precision Integrated circuit Temperature sensor, whose output voltage varies, linearly proportional to the Celsius temperature. It is a small and cheap IC which can be used to measure temperature between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Adruino. Power the IC by applying a regulated voltage like +5V to the input pin and connected the ground pin to the ground of the Arduino board. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range[8]. The output temperature of LM35 temperature sensor decides whether to energize or de-energize the relay.

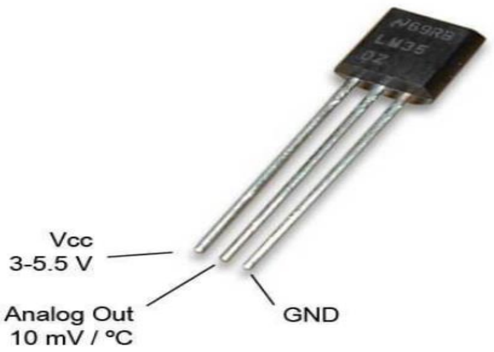


Fig.2 LM35 temperature sensor

If the temperature is 0°C, then the output voltage will also be 0V. There will be rise of 0.01V (10mV) for every degree 1 degree Celsius rise in temperature. It can range from -1V to 6V for the temperature -55^{0}C to 150^{0}C The voltage can converted into temperature using the below formulae.

V_{out} = 10mv/^{0}C × T

Where,

V_{out} is the LM35 output voltage &

T is the temperature.

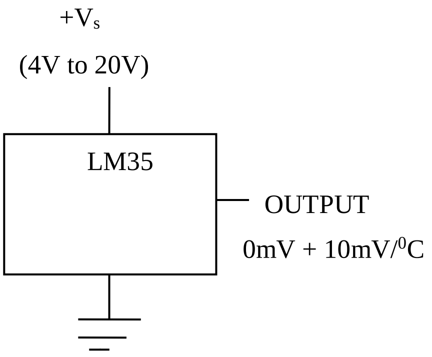


Fig.3 LM35 sensor block diagram

2.2 HC-SR04 Ultrasonic Level Sensor

The **HC-SR04 Ultrasonic sensor** is a 4 pin module, whose pin names are Vcc, Trig, Echo and Gnd respectively. It is used in many applications where measuring distance or sensing objects are required. **It** is used with both microcontroller and microprocessor platforms like Adruino, ARM, PIC, Raspberry, Pie etc. Ultrasonic ranging module HC-SR04 provides 2cm-400cm non-contact measurement function, with an accuracy of 3mm. The modules includes ultrasonic transmitters, receiver and control circuit.

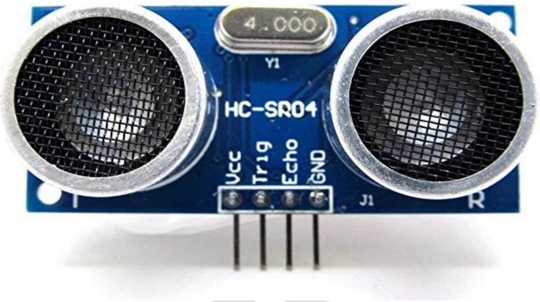


Fig.4 HC-SR04 ultrasonic level sensor

The basic principle of work:

The HC-SR04 level sensor works by the principle of “time of fight”. It emits an ultrasound of 40 kHz which travels through air. If there an obstacle is present on its path it will echo back. The distance can calculate by considering the time travel and speed of the sound.

1. To trigger the pin, 5V pulse for at least 10 µS is applied.
2. The HC-SR04 responds by transmitting a burst of 8 pulses which travels at the speed of sound and receives the echo pin.
3. If the pulse in not reflected backs then the Echo signals will timeout after 38mS and return low. This produces a 38 mS pulse that indicates no obstruction within the range of the sensor.
4. If the pulse is reflected back the Echo pin goes low when the signal is received. This produces a pulse whose width varies between 150 µS to 25 mS, depending upon the time it took for the signal to be received.
5. The width of the received pulse is used to calculate the distance to the reflected object. The sensor works with the simple a formula,

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the figure below,

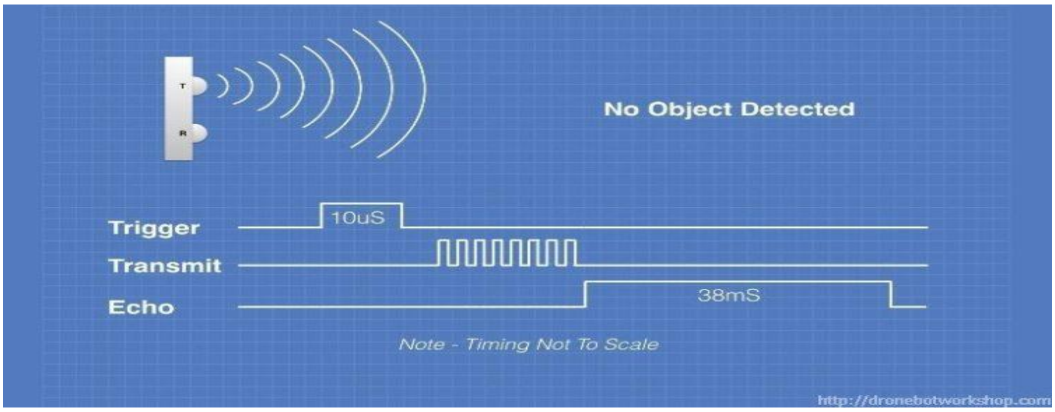


Fig.5(a) Timing Diagram Of HC-SR04 when No object

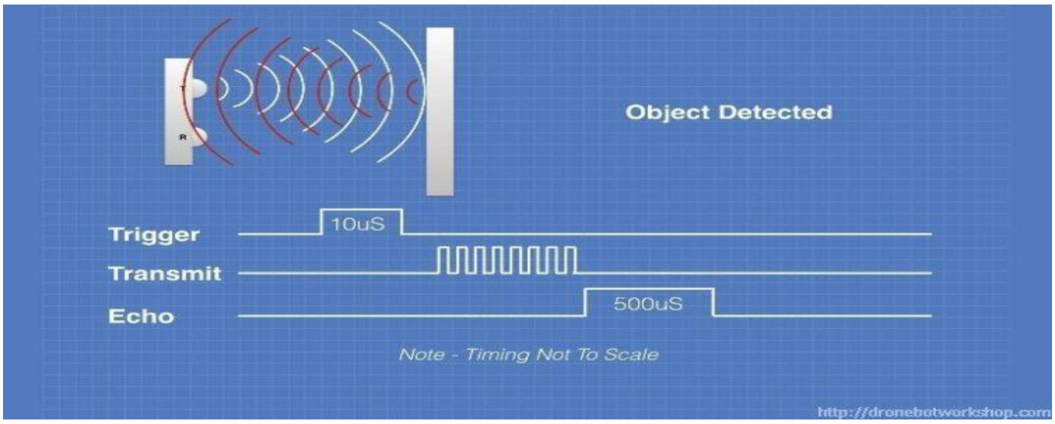


Fig.5(b) Timing Diagram Of HC-SR04 when Object detected

2.3 Arduino UNO

The Adruino UNO is an open-source microcontroller board based on 8-bit ATmega328P microcontroller and developed by Arduino.cc. Along with ATmega328P, it consist of other components such as crystal oscillator, serial communication, voltage regulator to support the microcontroller. Adruino Uno has 14 digital input/output pins (where 6 pins can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button. It can be powered by a USB cable or by an external 9V battery. The word "Uno" means one in Italian and was chosen to mark the release of Adruino Software IDE (Integrated Devlopment Environment) 1.0. The Uno board and version 1.0 of Adruino Software (IDE) is the reference versions of Adruino.

The ATmega328 on the Adruino Uno comes pre-programmed with a boot-loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the [9] original STK500 protocol. The Uno does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 programmed as a USB-to-serial converter. Adruino can be used to communicate with a computer, another Adruino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication using digital pin 0 (Rx) and digital pin 1 (Tx). The ATmega16U2 firmware uses the standard USB COM drivers, it does not require an external driver. But in case of Windows, a .inf file is required. The Adruino software includes a serial monitor which allows simple textual data to be sent to and from the Adruino board. The Rx and Tx LEDs on the Arduino blinks when data is transmitted through USB to serial chip and USB to computer. It does not flash for serial communication on pins 0 and 1. A Software Serial library allows for serial communication on any of the Uno's digital pins.

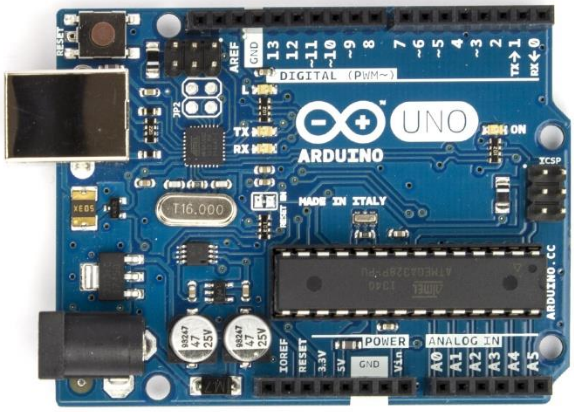


Fig.6 Arduino UNO Board

2.4 4 Channel 5V relay module with optocoupler isolation

This relay module facilitates to combine the processing power of the Arduino to devices that use higher current and voltage. It does so by providing four relays that are rated for 7A at either 28VDC or 10A at 125VAC.Each relay has a Normally Open (NO) and a Normally Closed (NC) contact. With these relays we can control appliances, motors, lights and other relays. For relay module inputs, the module is supplied with power via the pin labeled VCC and ground via the pin labeled GND. The relays are energized with low inputs to the IN1, IN2, IN3 and IN4 inputs. For relay module outputs, there are four relays and each of them provides dry contact outputs. Each relay provides a common (COM), normally open (NO) and a normally closed (NC) terminal [5].

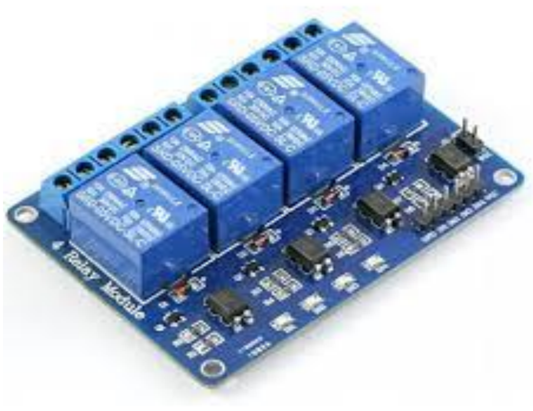


Fig.7 4 Channel 5V relay module with optocoupler isolation

2.5 LabVIEW

LabVIEW is a graphic programming tool. LabVIEW(Laboratory Virtual Instrument Engineering Workbench), is a product of National Instruments. It allows data acquisition, instrument control, data processing and data presentation. LabVIEW [5] which can run on PC under Windows, Sun SPARstations as well as on Apple Macintosh computers, uses graphical programming language (G language) departing from the traditional high level languages such as the C language, Pascal or Basic. The main advantage that LabVIEW offers is that it is a graphic language. The programs in this language have, from the user side, the same appearance than the front panel of an electronic device, with buttons, graphic screens, numerical indicators, etc. LabVIEW offers the possibility of creating the same control structures with a kind of module look, interconnected among them with cables that carry information from ones to another. In this way, those modules or sections- called in LabVIEW virtual instruments VI are the equivalent to functions in a classic programming language and the cables that interconnect would act as input or output variables of these functions. LabVIEW offers the possibility of emulating and control through software real electronic equipment as the one you can find in a laboratory. In LabVIEW, these two parts are given the names of Front Panel and Block Diagram, respectively. With the representation in the screen of the graphic visualization elements and control that will be used as interface with the user, this will observe the state of the selected inputs on the screen. The user will be able to interact with the outputs directly or with the execution of different routines that have been programmed.

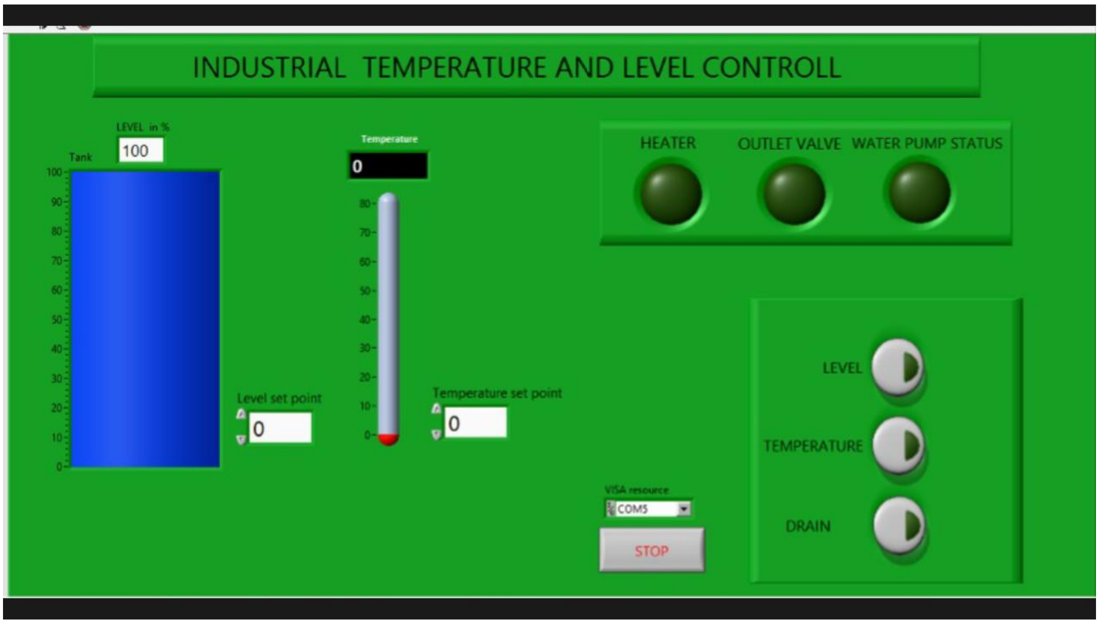


Fig. 8 Front Panel view of LabVIEW

2.6 Hardware Setup

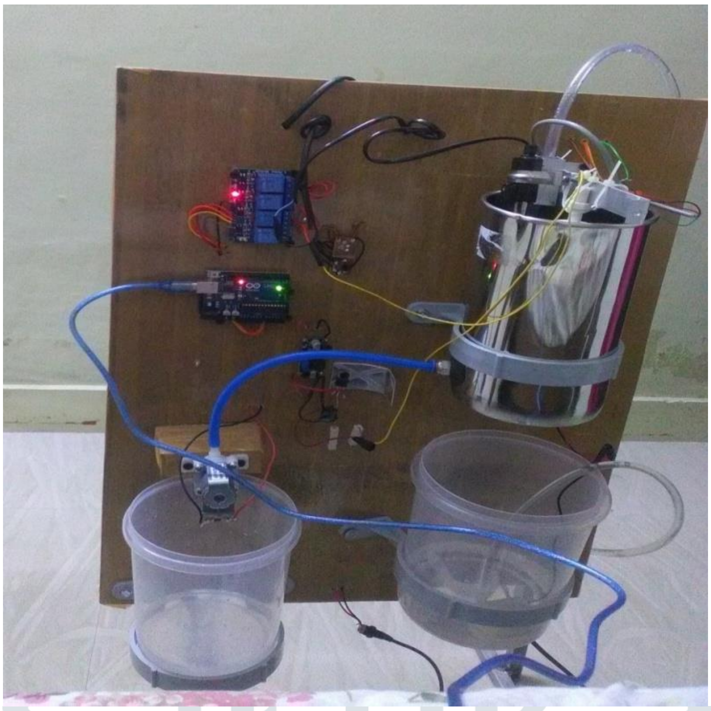
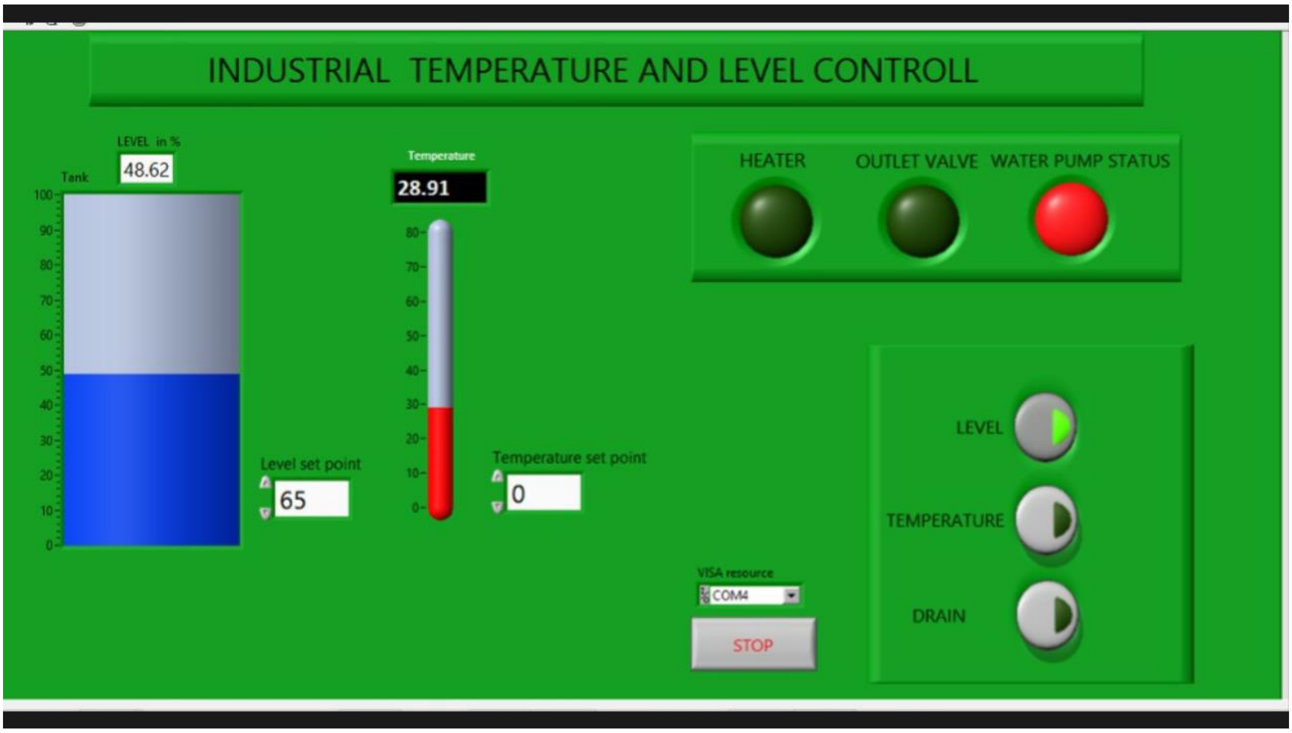


Fig.9 Hardware Setup of the proposed system

This is a very simple process to control temperature and level at desired set point using LabVIEW and Arduino where the user can set the temperature and water level using the set point control input of GUI in LabVIEW. In this project I have set the Arduino to turn ON the relay and starts water pumping when the water level is below the set point and to turn ON the heater coil when the temperature of the water is below the set point. An analog temperature sensor LM35 is used to monitor the temperature variations of the process and an ultrasonic level sensor HC-SR04 is used for the continuous monitoring of water level .The LabVIEW continuously compare the process variables with set point and make the arduino to take control action, if the temperature and level points is below the desired set point. If the level of the system goes below the set point, LabVIEW generates a controller output to Arduino board via UART communication and this control output triggers the digital output ports of Arduino from high to low. If the output turned ON, the optoisolators in the relay control board, activates and energizes the relay and starts water pumping. When the water level reaches the set point LabVIEW send a turn off command to Arduino thus de-energizes the relay and stops the water pumping. The level sensor HC-SR04 sends ultrasonic sound waves to the process tank and continuously monitor the echo signal to calculate the level of water in tank, By calculate the time of wave travel of the echo signal it calculates the level of water. The analog temperature LM35 continuously monitors the temperature of water and if it senses the temperature is below the set point, LabVIEW generates controller output by activating digital output of DAQ and turns ON the heater. When the temperature reaches the desired set point the heater turns OFF. By ON-OFF mechanism of heating coil the controller can maintain the temperature of liquid tank.

III. RESULTS

The whole program is placed in a While loop with a STOP control. The results of different conditions when temperature and level are set in different status are shown below. Simultaneous processes can be monitored and controlled in LabVIEW. In fig.8 the Complete Front Panel design of the process is shown, where the process variables such as temperature and level are at normal condition making all the final control status like heater, outlet valve and water pump status are in OFF condition.


INDUSTRIAL TEMPERATURE AND LEVEL CONTROLL

LEVEL in % 48.62

Temperature 28.91

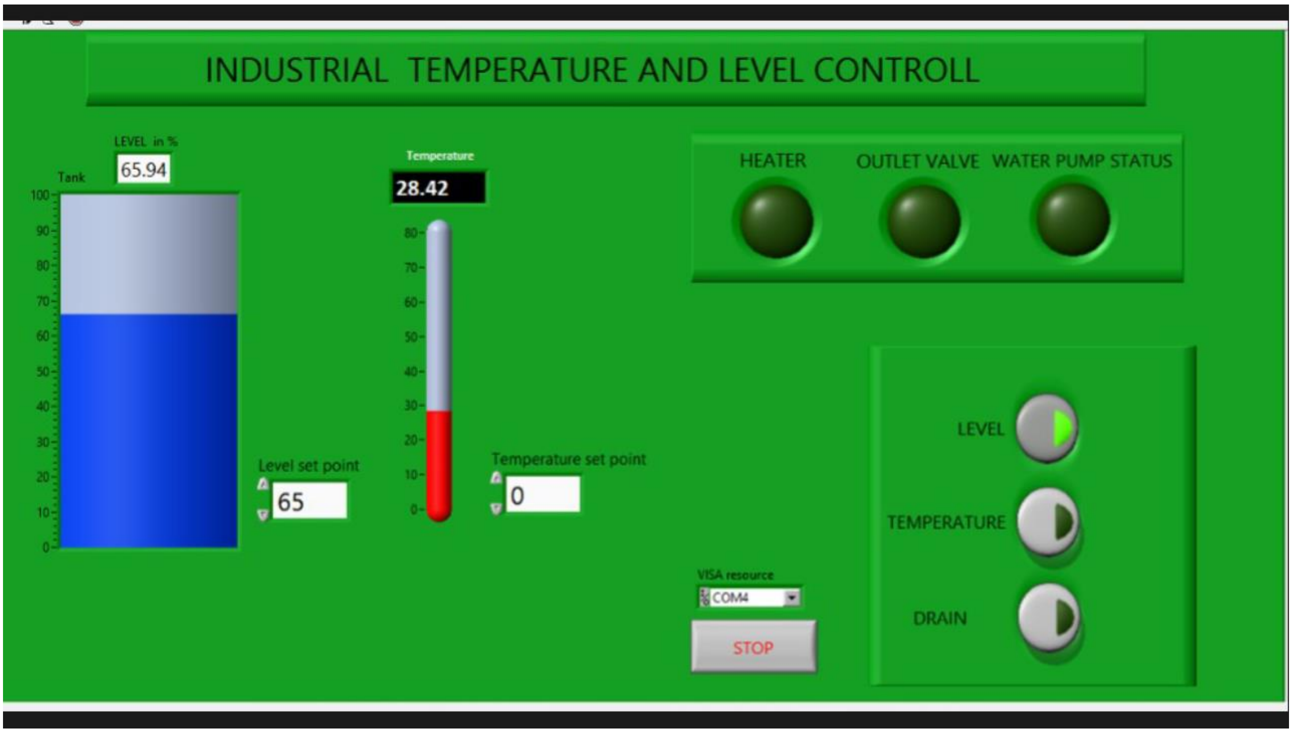
Level set point 65

Temperature set point 0

VISA resource COM4

Fig.10 Front Panel view when level is below Set Point.

Here, we set the ‘level set point’ to 65, which means the desired water level is 65. So in fig.10, as we can see the level of the water is below the set point, thus labview generates a controller output to trun ON the water pump by activating the relay and making the water pump status to turn ON. The water pump status will turn OFF when it reaches the desired set point.


INDUSTRIAL TEMPERATURE AND LEVEL CONTROLL

LEVEL in % 65.94

Temperature 28.42

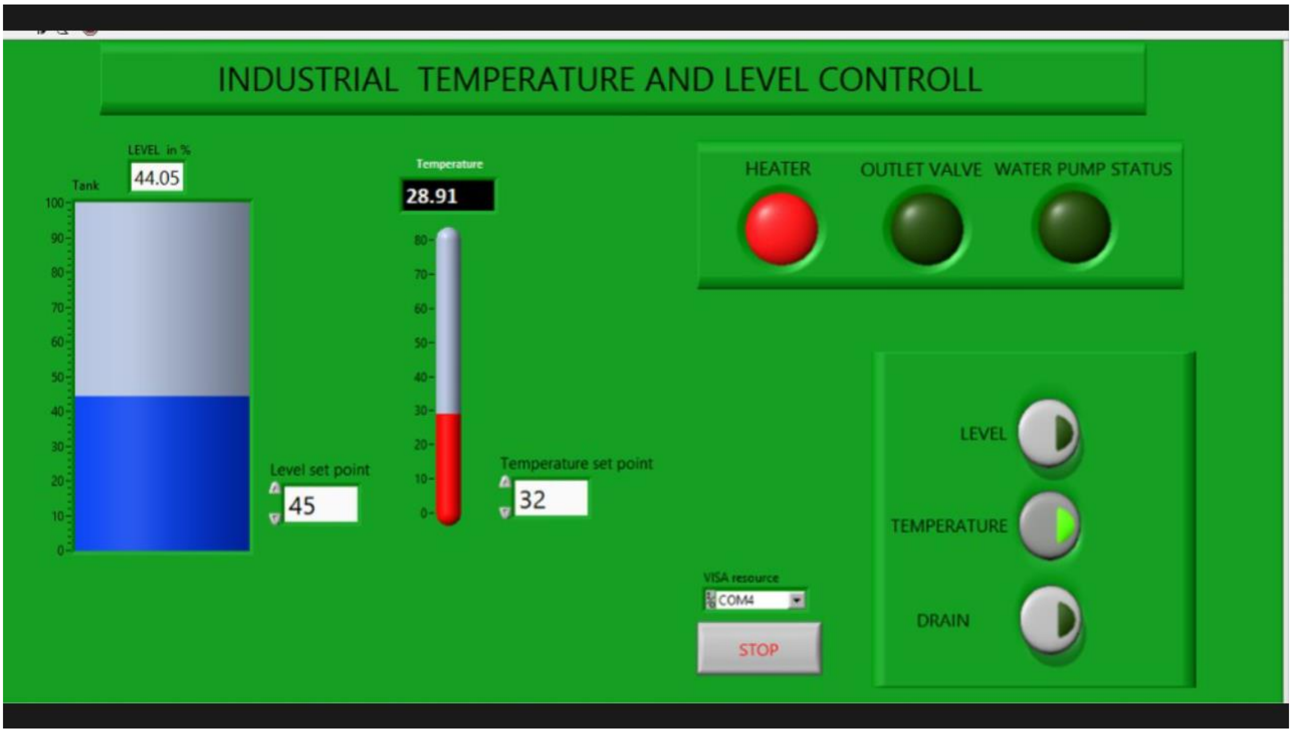
Level set point 65

Temperature set point 0

VISA resource COM4

Fig.11 Front Panel view when level reach the set point.

Here In fig.11 the desired water level is set to ‘65’.when the level of water reaches the desired set point, the LabVIEW sends a turn OFF command to Arduino and makes the ‘water pump status’ to turn OFF.



INDUSTRIAL TEMPERATURE AND LEVEL CONTROLL

LEVEL in % 44.05

Temperature 28.91

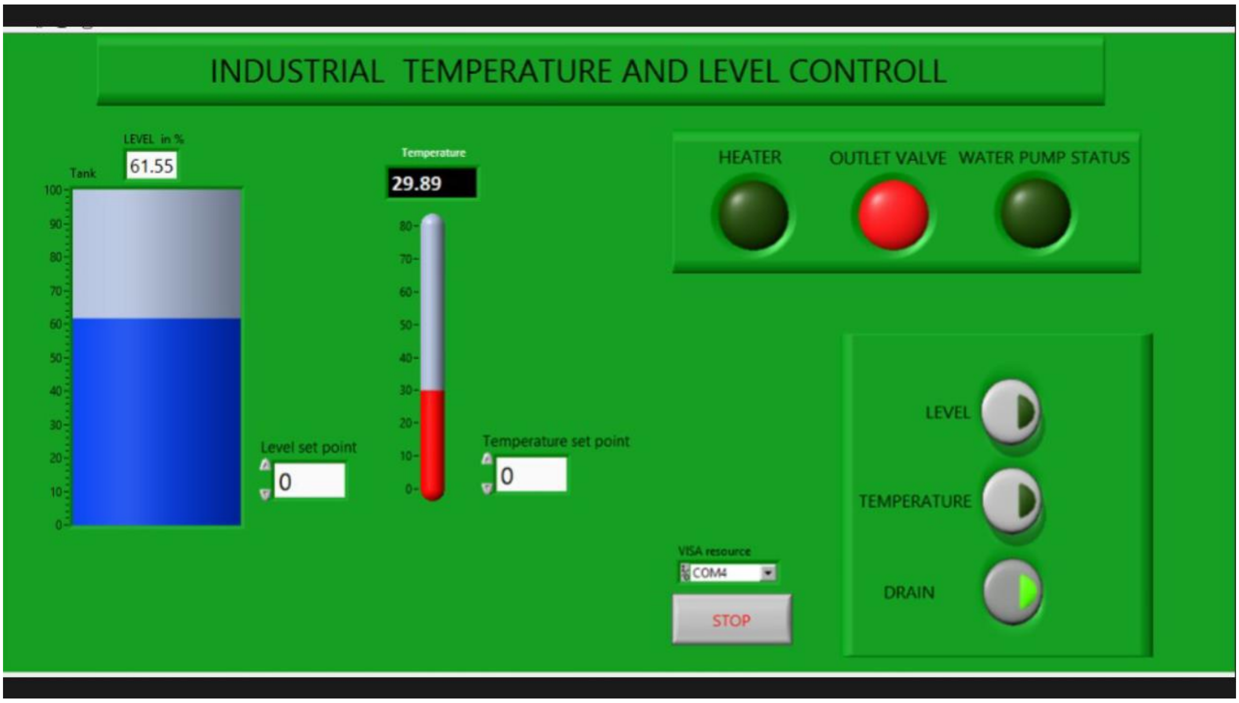
Level set point 45

Temperature set point 32

VISA resource COM4

Fig.12 Front Panel view when temperature is below set point.

Here In fig.12, we set the 'level set point' to 45 and 'temperature set point' to 32. The temperature of water in liquid tank is below the set point, thus the labview generates a controller output to trun ON the Heater and making the heater status to turns ON. The heater will turn OFF when it reaches the desired temperature. Since the water level is at the desired set point the water pump status is OFF.



INDUSTRIAL TEMPERATURE AND LEVEL CONTROLL

LEVEL in % 61.55

Temperature 29.89

Level set point 0

Temperature set point 0

VISA resource COM4

Fig.13 front panel view when water is draining.

Here in fig.13 we set the water to drain and making the 'outlet valve' status to turn ON. The outlet valve status turns OFF when water draining is complete

IV. Conclusion and Future Scope

In this paper, we designed and implemented an efficient method for monitoring and controlling of temperature and level of a liquid tank using LabVIEW with the help of an Adriano UNO. The output was verified by setting the level and temperature at different set points and it found that the LED turn on and off when the temperature and level varies from desired set points. The main advantage of the system is that, it is cost effective way to control a plant using arguing UNO. This can be used in industry as well as in home. The Virtual Instrumentation can completely replace the traditional controllers. Using technologically advanced virtual instrument technology to replace traditional measuring and testing technology is not only feasible, but also better and more stable. Again, this system can be fuzzified to make it more efficient and increase controllability and can be used in the field of Artificial Intelligence.

REFERENCES

[1] M.Viyayaraghavan and K. Madumathi. Implementation of PID Temperature Control Using LABVIEW, International Journal of Research in Advent Technology, Vol.2, No.11, November2014 E-ISSN: 2321-9637 115.

[2]Vaibhav M. Davande and Pradeep C. Dhanawade. Real Time Temperature Monitoring Using LABVIEW and Adruino, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 3, March 2016.

[3]Lizy Abraham. LabVIEW based Modeling and Analysis of Temperature Sensors. First International Conference on Computational Systems and Communications (ICCSC) | 17-18 December 2014.

[4]Sayan Kumar Swar and Susmita Das. Automatic Liquid Level & Temperature Monitoring and Controlling using LABVIEW and ARDUINO, IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 11, Issue 2 Ver.II (Mar. – Apr. 2016), PP 56-60.

[5] https://en.wikipedia.org/wiki/LabVIEW

[6] https://hacktronics.co.in/4-channel-relay-module/4-channel-5v-relay-module-with-optocoupler-isolation

[7] https://components101.com/ultrasonic-sensor-working-pinout-datasheet

[8]https://wiki.eprolabs.com/index.php?title=Temperature_Sensor_LM35

[9] https://en.wikipedia.org/wiki/Arduino_Uno

[10] https://iamzxlee.wordpress.com/2013/07/06/ultrasonic-sensor-hc-sr04/