

SCADA Implementation of Industrial Temperature Automation

Haider Ali^[1] & Ahmed Ali^[2] & Riaz Ul Hassnain Syed^[3] & Ajmal Khan^[4] & Ihsanullah Khan^[5]

CIIT Abbottabad Campus, Engineering Department KPK Pakistan

Abstract

This paper concentrates upon the development of an automated setup for the measurement of temperature of any industrial environment, for example boilers, refrigerators and heaters etc. The hardware requirements are significantly curtailed using the microcontroller's modern embedded features like A/D conversion, display interface, hardware and software interrupts, and communication protocols for interfacing cellular phones and GSM modules. This truncation in hardware requirement has lead to a smarter industrial temperature automation design.

Software Tools Used

Proteus Professional ISIS 7.7, MikroC pro for PIC v4.6, Hyper Terminal, LabVIEW 2010

Keywords

SCADA (Supervisory Control And Data Acquisition), SETPOINT, FEEDBACK,

1. Introduction

There are several industrial processes including those of manufacturing, power generation, fabrication and refining that require an immediate and accurate temperature control and monitoring infrastructure. This paper concentrates upon the development of one such infrastructure, with fewer hardware details. The hardware requirement is minimized by using simpler firing angle generation hardware and data acquisition particulars.

This design incorporates a temperature sensor installed in any industrial environment, a motherboard card for deciphering the temperature sensor's FEEDBACK information and based on SETPOINT definition, issuing a timed firing pulse for actuating a TRIAC controlling the magnitude of HVAC provided to the particular industrial equipment. The SETPOINT is communicated to the motherboard via wireless channel, from the HMI. Once the coincidence between SETPOINT and FEEDBACK temperatures occur the equipment is made to operate in equilibrium. Both the SETPOINT and FEEDBACK temperatures are displayed on an LCD panel, for the convenience of operation and maintenance of the particular equipment.

2. Design Description

The SCADA system definition of this project is described below:

- Human-Machine Interface or HMI, used here is a simple cellular phone through which the operator can monitor and control the industrial environment's temperature. For providing a convenient system operation, the HMI can also be made to communicate with a computer, so that both the SETPOINT and the FEEDBACK could be issued and monitored via automated system.
- Supervisory System, used in this project is a PIC Microcontroller, which gathers temperature data from the sensor as FEEDBACK temperature, compares it with the temperature SETPOINT provided by the HMI, and issues an error signal which is used to actuate the particular industrial process.
- Communication Infrastructure used here, to transmit the SETPOINT temperature from the HMI to the Supervisory system is GSM communication protocol.

3. Hardware Assembly

The line diagram of the project is mentioned below in figure-1.

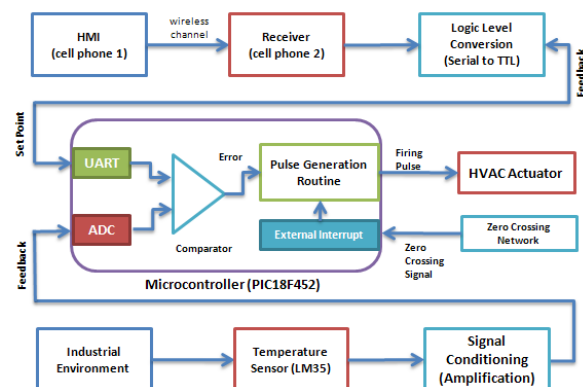


Figure 1: Line Diagram of SCADA Implementation of Industrial Temperature Automation

As from the line diagram above, it is pretty much evident that the project operation is distributed into the following major sections,

1. HMI (Cell-Phone#01): This is a handheld device provided to the user for both operating and monitoring

the temperature of the industrial environment. User sends a temperature SETPOINT, at which the industrial environment is sought to operate. The HMI can also be made to communicate with a computer in order to provide better DAQ facilities.

2. Receiver (Cell-Phone#02): Another cell phone is connected with the original hardware for receiving the SETPOINT temperature and transmitting the FEEDBACK temperature via wireless channel. The SETPOINT temperature is fed into the next block called as "Logic Level Conversion (Serial to TTL)".
3. Logic Level Conversion (Serial to TTL): In this block the binary data received by the receiver is interpreted into TTL compatible logic level. The reason for this conversion is because we cannot operate our digital circuits (microcontrollers) using the logic level protocol of serial data. The output of this stage is the SETPOINT temperature.
4. Temperature Sensor from Industrial Environment (LM35)^[3]: For obtaining the temperature information, an LM35 temperature sensor is installed in any industrial environment, such as on the head of a diesel engine or inside a refrigerator or in an incubator etc. The range of measurements that can be made by the LM35 sensor is between $-55^{\circ}\text{C} \sim 150^{\circ}\text{C}$.
5. Signal Conditioning (Amplification): Sensor's output voltage is subjected into signal conditioning unit, where it is amplified and brought to a level operational for interfacing it with a built in ADC of the PIC microcontroller.
6. Zero Crossing Network: This is a reference circuit, providing an edge triggered hardware interrupt for the microcontroller every time the input AC voltage goes through zero-magnitude point ($0^{\circ}/180^{\circ}/360^{\circ}$).
7. Microcontroller (PIC18F452): The SETPOINT and FEEDBACK temperatures are provided at the UART and ADC inputs of the microcontroller. Both the two temperatures are compared, and in case of any coincidence, a command is sent to the *Pulse Generation Routine* to produce an output firing pulse, later to be connected to the HVAC actuator. Another driving requirement for Pulse Generation Routine is to have a zero-crossing signal providing upper and lower limit to the position of the pulse with respect to the magnitude of the HVAC, accounting for Phase Angle Control.
8. HVAC Actuator^[3]: The firing pulse output from the microcontroller is fed to the HVAC actuator (capable of firing voltages as high as 600Vrms). The variable AC voltage output from the actuator can be used to drive any temperature controlling device for example, normal AC voltage operated industrial heaters, heaters of No Frost refrigerators and different space cooling compressors etc.

4. Software Description

The algorithmic representation of this project is outlined below in figure 2. The algorithm was embedded into a PIC18F452 microcontroller using the compiler called MikroC Pro for PIC v4.6^[4].

The entire algorithm is divided into following major steps,

1. Acquiring the SETPOINT temperature via HMI
2. Continuously monitoring the FEEDBACK temperature from the temperature sensor
3. Comparing the FEEDBACK with SETPOINT, and accordingly ramping up or down the HVAC output from the actuator.

So first of all the SETPOINT temperature is acquired from the HMI. The operator is required to send a temperature array (in the following format: $+xxx$ or $-xxx$) from the HMI via text message. This text message is deciphered by the microcontroller using the standard *AT Commands*. This way a 3-digit SETPOINT temperature number is extracted from it. In the flowchart below, this number is represented as SETPOINT TEMP.

Next task at hand is to read continuously the FEEDBACK temperature. The temperature reading is properly scaled down to another 3-digit FEEDBACK temperature number represented in the flowchart below as FEEDBACK TEMP.

The last step used in this algorithm is where a comparison between SETPOINT TEMP and FEEDBACK TEMP is made and this is where we wish to coincide the two temperatures. If FEEDBACK is *greater than* SETPOINT, then we must raise the FEEDBACK continuously till it matches the SETPOINT, and vice versa. The only way to raise the temperature of a particular environment (inside a refrigerator or in the vicinity of a HVAC controlled industrial heater etc), is to provide more and more voltage to the particular actuator, and vice versa is also true. The voltage output can be increased by firing the TRIAC close to 0° , and the voltage output can be reduced by firing the TRIAC close to 180° . This means we must grasp hold of the limits 0° and 180° , as well as the movement of the firing pulse between these two limits, hence accounting for the Phase Angle Control (PAC) technique.

The 0° and 180° limits can be realized using an edge triggered hardware interrupt feature of PIC18F452 microcontroller using the aid of zero crossing signal of the HVAC we wish to control. Once grabbing hold of the limits, the next task is to generate a firing pulse and continuously displace it between the two limits. The displacement/movement of the pulse (either towards 0° or 180°) depends on the comparison between

FEEDBACK and SETPOINT temperatures. If SETPOINT is greater than FEEDBACK, pulse is made to displace toward 0° , and if the SETPOINT is less than FEEDBACK, pulse is made to displace toward 180° . This pulse is coupled to the TRIAC (HVAC actuator) through an optocoupler. As a result the magnitude of the HVAC voltage is varied in order to address the coincidence issue of SETPOINT and FEEDBACK temperatures.

The firing pulse is generated and displaced using the same technique used for the generation of a PWM (Pulse Width Modulated) signal i.e. to compare a periodic and linear RAMP signal with a fixed DC signal, and every time the coincidence of magnitudes between the two signals occur, jump from the negative rail to the positive rail of the power supply. This hardware phenomenon was translated and embedded in our project's algorithm, and this is where the true essence of reduction of hardware lies.

The ramp signal can be realized in the algorithm as a continuously incrementing and resetting ramp-counter. Every time the external edge triggered interrupt occurs, the algorithm forces this counter to reset. However, during the time between two recurring external interrupts (between 0° and 180°) the counter increments continuously. The time duration between two consecutive increments of the counter is a fixed value, programmed using an internal timer interrupt of the microcontroller. In our project, the timer0 of PIC microcontroller can be configured for an increment time of 10us or 39us for a maximum count of 1000 and 256 respectively.

The fixed DC signal can also be realized into the algorithm as a constant-number. However, this number is not very much of a constant, because it is continuously made to increment or decrement (according to comparison specifications of FEEDBACK and SETPOINT).

Any instant of time at which a match is made by the instantaneous value of ramp-counter with the constant-number, the algorithm issues a firing pulse, and at that particular point, the waveform of the HVAC is notched. This notching account for reduction in the overall RMS value of the voltage supplied to the load [1].

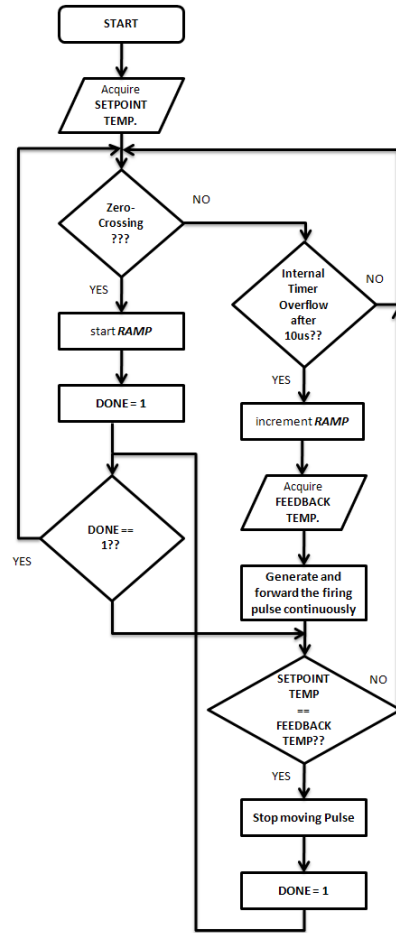


Figure 2: Algorithmic Representation of SCADA implementation of Industrial Temperature Automation

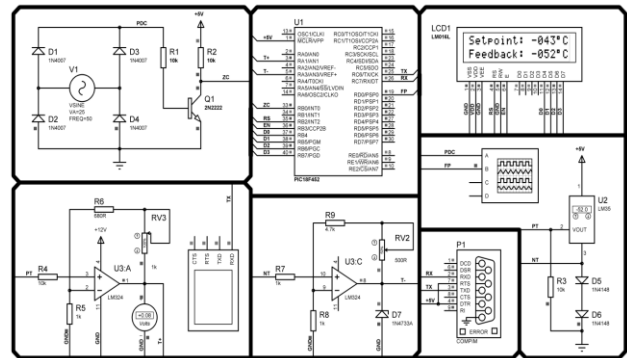


Figure 3: Simulation of SCADA Implementation of Industrial Temperature Automation

5. Results

The software simulation of this project was carried out in Proteus Professional ISIS 7.7. However, Proteus ISIS has a simulation shortcoming that it cannot perform necessary mathematical modeling of the clipping waveforms. So in order to perform the mathematical modeling, the aid of LabVIEW was employed.

In this section of the research paper, we wish to explore the error between the paper calculated values for output voltage from the HVAC actuator, and the values obtained from the mathematical model of LabVIEW. Figure 3 below shows the simulation environment of Proteus ISIS and Figure 4 shows the front panel of the LabVIEW VI.

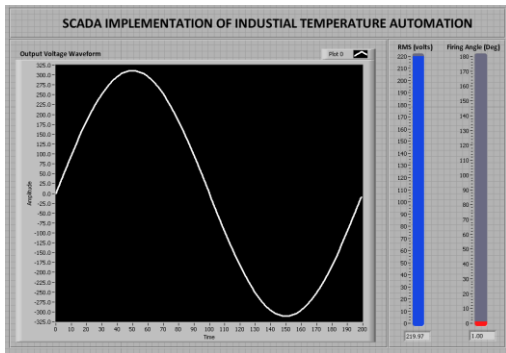


Figure 4: LabVIEW Based Front Panel for Project

As described in the Software Description section, a SETPOINT temperature is communicated from an HMI to the computer and into the simulation environment of Proteus Professional ISIS 7.7 using component named “COMPIM”. The SETPOINT information is extracted into a 3-digit number and is printed on the LCD block (as shown in figure 3 above). After that FEEDBACK is continuously read from the temperature sensor LM35, and is also displayed on the LCD. Every time a coincidence between SETPOINT and FEEDBACK occurs, a text message is sent by the microcontroller to the HMI, that FEEDBACK = SETPOINT.

The generation and displacement of firing pulse is also demonstrated on the oscilloscope. Figures 4 - 7 show the displacement of firing pulse, relative to the pulsating dc signal (a stepped down & rectified replica of the original HVAC signal), in Proteus ISIS.

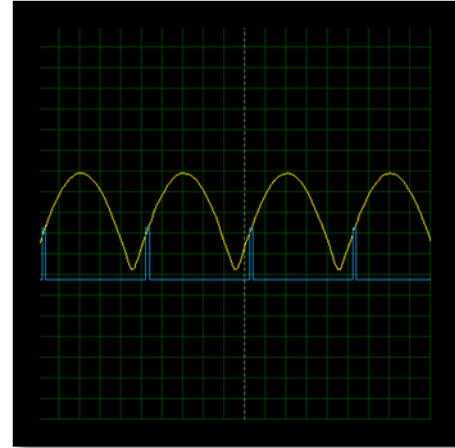


Figure 5: Firing angle at 30°

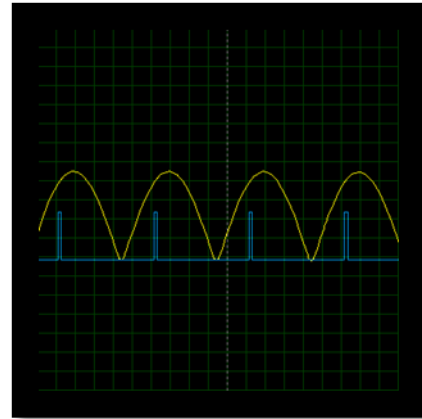


Figure 6: Firing angle at 60°

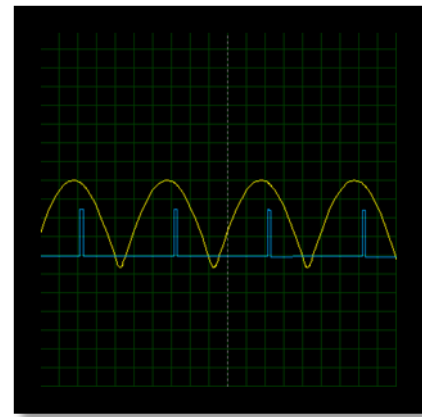


Figure 7: Firing angle at 120°

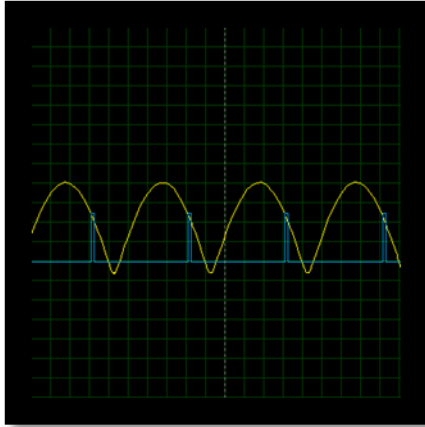


Figure 8: Firing angle at 160°

Below a couple of cases are studied from the LabVIEW mathematical model, along with the paper calculations and the error between the two is outlined at the end of every case.

The formula used for the calculation of the RMS value of the output voltage is given as,

$$V_{O(RMS)} = V_{in(RMS)} \left[\frac{1}{\pi} \left(\pi - \alpha + \frac{\sin 2\alpha}{2} \right) \right]^{\frac{1}{2}} \dots \dots (1)$$

Where,

$V_{O(RMS)}$ = RMS value of the Output Voltage

$V_{in(RMS)}$ = RMS value of the Input Voltage = 220V

α = Firing Angle (in radians) (0~180°)

According to the formula for percentage error, we have,

$$Percent\ Error = \frac{V_{O(RMS)(simulation)} - V_{O(RMS)(paper)}}{V_{O(RMS)(paper)}} \times 100\% \dots \dots \dots$$

Where,

$V_{O(RMS)(simulation)}$ = RMS Output Voltage from the mathematical model

$V_{O(RMS)(paper)}$ = RMS Output voltage calculated on paper from (1)

Case 1: $\alpha = 45^\circ$

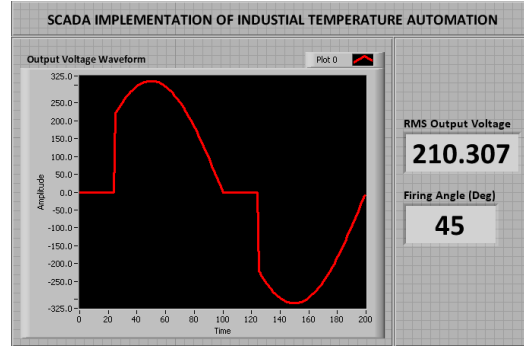


Figure 9: Firing angle at 45°

From Figure 9 we learn the simulation results as,

$$V_{O(RMS)(simulation)} = 210.307V$$

However, from formula 1, the value turns out to be,

$$V_{O(RMS)(paper)} = 209.778V$$

The percentage error is given from (2),

$$Percent\ Error = 0.25\%$$

Case 2: $\alpha = 90^\circ$

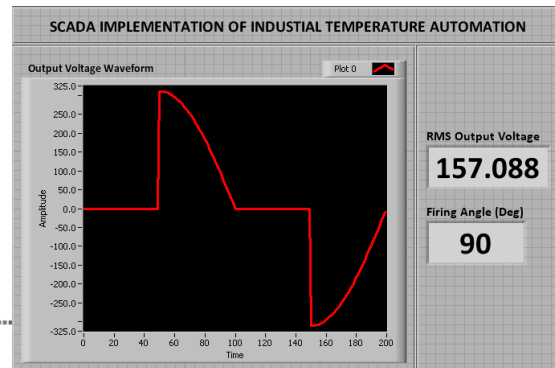


Figure 10: Firing angle at 90°

From Figure 10 we learn the simulation results as,

$$V_{O(RMS)(simulation)} = 157.088V$$

However, from formula 1, the value turns out to be,

$$V_{O(RMS)(paper)} = 155.56V$$

The percentage error is given from (2),

$$Percent\ Error = 0.98\%$$

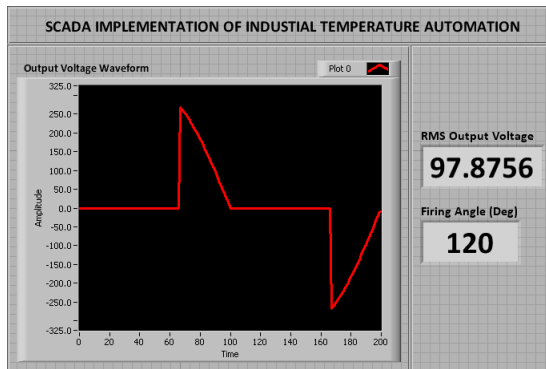
Case 3: $\alpha = 120^\circ$ 

Figure 11: Firing angle at 120°

From Figure 11 we learn the simulation results as,

$$V_{O(RMS)(simulation)} = 97.876V$$

However, from *formula1*, the value turns out to be,

$$V_{O(RMS)(paper)} = 97.25V$$

The percentage error is given from (2),

$$\text{Percent Error} = 0.63\%$$

6. Conclusions

In this research paper, our prime concern was the development of demonstrating how industrial temperature automation can be made possible using little hardware and more refined software details. The prime concern was to generate firing pulse for an HVAC controlling actuator while displacing them at the same time to vary the magnitude of the ac voltage output. The direction of displacement of the pulse was described by the SETPOINT definition from an HMI using AT commands, and FEEDBACK from the temperature sensor installed in the industrial environment. The mathematical modeling was done in LabVIEW for investigating the impact of varying firing angle on the magnitude of the ac voltage. This controlled variation of output ac voltage can be subjected to controlling temperature of the particular industrial environment.

References

- [1] Bin Hu, "High Power Converters and AC Drives," *The Institute of Electrical and Electronics Engineers, Inc.* 2006 John Wiley & Sons, 2006, ISBN-13 978-0-471-73171-9
- [2] Muhammad H. Rashid, "Power Electronics Circuits, Devices, and Applications," 3rd Edition, Pearson Prentice Hall 2006, ISBN 978-81-317-0246-8

- [3] Christopher T. Kilian, "Modern Control Technology: Components and Systems" 2nd Edition, Delmar Thomson Learning, ISBN-10: 076682358X
- [4] Dogan Ibrahim, "Advanced PIC Microcontroller Projects in C", Elsevier 2008, ISBN-978-0-7506-8611-2
- [5] John Dunlop, Demessie Girma, James Irvine, "Digital Mobile Communications and the TETRA system", John Wiley & Sons, Inc, ISBN 0-471-98792-1
- [6] Ashoke K Talukder, Hasan Ahmed, Roopa R Yavagal, "Mobile Computing - Technology, Applications and Service Creation" 2nd Edition, TATA McGraw Hill



Haider Ali, Education: Master in Electronic Systems Design Engineering. Lecturer at COMSATS Institute of Information Technology ABBOTTABAD.



Ahmad Ali, Education: Msc in Computing Institution: Manchester Metropolitan University UK.



Riaz Ul Hassnain Syed, Institution: N.W.F.P University of Engineering & Technology Peshawar, Electronics Engineering Dept. Abbottabad campus, Pakistan



Ajmal Khan, Education: MSc in Computing, Institution: Manchester Metropolitan University UK



Ihsan Ullah Khan, Department of ElectronicEngineering University of Engineering & Technology Peshawar Abbottabad Campus