Generation of firing pulse for TRIAC using PIC 16F877A Microcontroller

[1]Qamar Ul Islam, and [2]Farhad Ilahi Bakhsh
[1]Department of Electrical and Renewable Energy Engineering, Baba Ghulam Shah Badshah University, Rajouri-185234, J & K
[2]Department of Electrical Engineering, National Institute of Technology Srinagar, Hazratbal, Srinagar, J & K

***Abstract*—Mostly the TRIAC is used in phase angle controllers. In phase angle control, a gate pulse is applied to the gate terminal of TRIAC between one zero crossing to the next zero crossing. Without the gate pulse the TRIAC is OFF and no current flows through it. When the gate signal is given to the TRIAC it turns ON and the voltage is supplied to the load. In the paper, firing pulses for TRIAC have been generated using microcontroller. The microcontroller used is PIC 16F877A. Depending upon the delay time between the zero-crossing and the applied gate pulse to the TRIAC, the load voltage is controlled. Hence, the power flow to the load is controlled by delaying the firing angle of the TRIAC.**

Index Terms—*Firing pulse, TRIAC, PIC 16F877A microcontroller, MPLAB SIM software, opto-isolator (MOC 3021), zero crossing, phase angle control.*

I. INTRODUCTION

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode. MPLAB IDE is a Windows Operating System (OS) software program that runs on a PC to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment, or IDE, because it provides a single integrated "environment" to develop code for embedded microcontrollers.

II. PROGRAMMING OF MICROCONTROLLER

Generating the hex code from MPLAB IDE software involves following step.

1. Open MPLAB IDE and click project.
2. Goto project wizard, click to open it.
3. Click next and select PIC 16F877A from select a device, click next.
4. In active tool suite select microchip MPASM, click next.
5. Browse the computer to determine the place where the project files will be saved.
6. Click next, existing files can be added later, click finish.
7. Now open a new file and start writing the assembly program.
8. Save it and add it in source file.

Build the program. Hex code will automatically be generated at the predefined position as shown in Fig. 1.

The major advantage of using microcontroller for generating triggering pulses is the ease with which it can be programmed to generate a number of relative switching delay pulses for control circuit. The pulses generated from microcontroller are given to the inverter through opto-isolator (MOC 3021) for isolating it from the power circuit. One opto-isolator is required for a thyristor. The board developed for PIC16F877A is shown in Fig. 2.

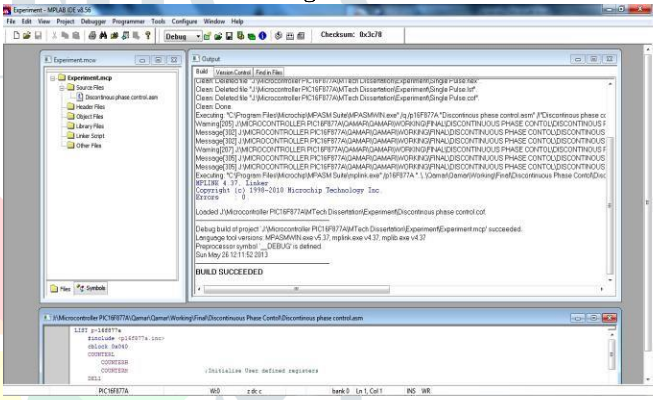

Fig. 1 MPLAB IDE window.

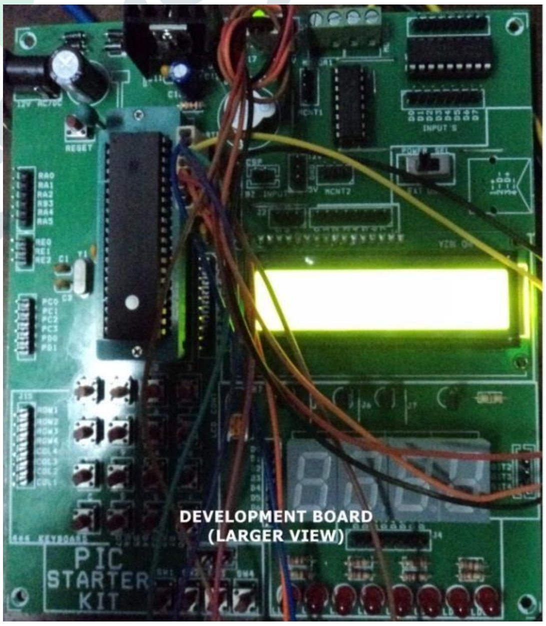

Fig. 2 Development Board of PIC16F877A.

III. Zero Crossing Detection

Zero crossing detection is very important, especially in power control circuits employing thyristors [1]. For a constant delay trigger pulse, synchronized signal has to be generated with respect to zero crossing instant of the mains voltage. The PIC 16F877A detects the zero crossing using the RB0/INT external interrupt function. This is explained how the zero cross is detected and how the PIC acts upon detection as shown in Fig. 3.

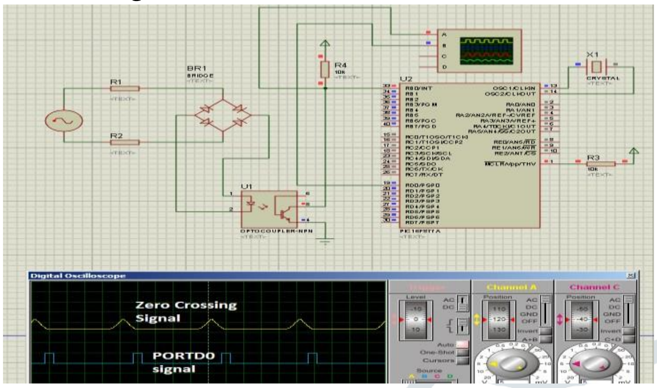

Fig. 3 Schematic zero crossing signal and RD0 signals [2]

During each half cycle, the LED of optocoupler is on and so RB0 is low. When the LED of optocoupler is off as the AC wave is about to cross "zero", RB0 goes high. The transition from low to high on RB0 is a rising edge. When the opt coupler LED is then again turned on as the AC wave crossed the "zero" level (zero crossing), RB0 goes low. This transition from high to low is a falling edge. And upon a falling edge, an interrupt is generated.

When an interrupt is generated, the ZC flag is set (see appendix). In the main code, ZC flag is always checked (this is called polling a flag). When ZC flag is set (zero crossing occurred), a 1ms pulse is generated on RD0 (PORTD0) and then the ZC flag is cleared. For example, here 1ms pulse is produced. The ZC flag is then again continuously polled to check for next interrupt.

IV. Phase Angle Control

Phase angle control method is applied to ac circuits using TRIAC. The power flow to the load is controlled by delaying the triggering angle in each half-cycle of the TRIAC [1]. Here the zero-crossing is first checked. After zero-crossing occurs, a small delay is present before the triac is triggered. (Here, it is 2ms) So, the triac is triggered after 2ms when the zero-crossing occurs. Then the gate signal is generated for 250µs. It is enough time to ensure that the triac has turned on. Even though the gating signal is removed, the triac stays on until the next zero-crossing as it is a latching device. The circuit setup is shown in Fig. 4.

The driver in the circuit is an optocoupler, MOC3021. This is a random phase optically isolated triac output driver. When the LED is turned on, the triac in the MOC3021 turns on and drives the triac in the circuit. It is a “random phase” driver meaning that it can be driven on at any time during the drive signal, as is required for phase angle control. There are other drivers that only allow drive at the zero-crossing instant. For guaranteeing that the triac is latched, the LED side of the MOC3021 must be driven with at least 15mA current. The maximum current rating for the LED is 60mA. The peak current rating for the triac is 1A. We have stayed within these limits in the design. The output waveform is shown in Fig. 5. In Fig. 5, the input AC supply is shown by green signal, the gate pulse is shown by pink signal and the output ac after phase control is shown by yellow signal.

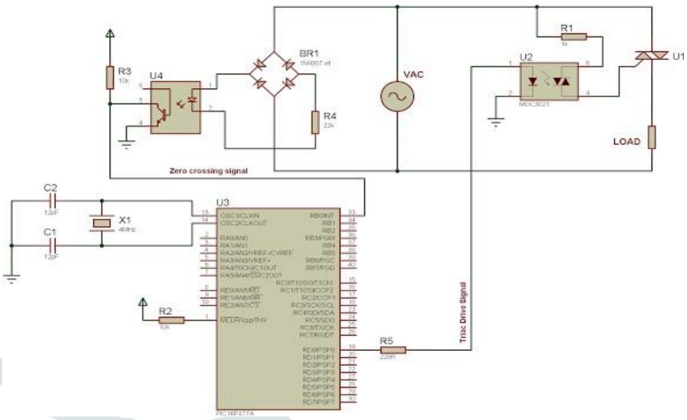

Fig. 4 Circuit Diagram for phase angle control [3]

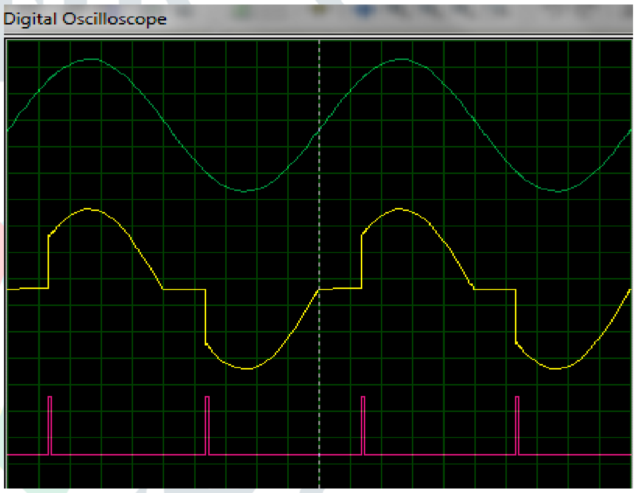

Fig. 5 TRIAC firing with 2 ms delay.

This shows that the simulation work for the generation of gate pulses for TRIAC and its application in phase angle control. The same can be achieved experimentally as shown in Fig. 6.

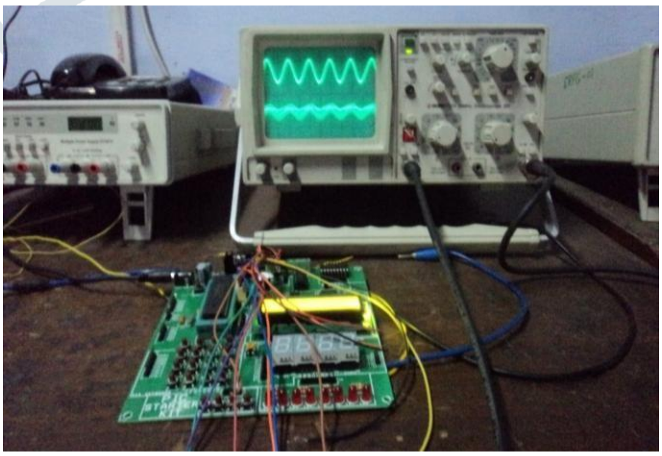

Fig. 6 Experiment Setup.

CONCLUSION

The paper presents the generation of gate pulse for TRIAC using PIC 16F877A microcontroller in details. The generated gate pulse is applied to the gate terminal of TRIAC for phase angle control. For this, simulation as well as experimental results has been obtained. The delay time between the zero-crossing and the applied gate pulse to the TRIAC controls the load voltage. Thus, the power flow to the load is controlled by phase angle control.

REFERENCES

[1] Twinkle Wahane, Kajal Khobragade, Rahul Argelwar, "Application of Phase Angle Control of Triac for Speed Control of Induction Motor," Journal of Network Communications and Emerging Technologies (JNCET), Vol. 6, No. 4, April, 2016.

[2] http://tahmidmc.blogspot.com/2012/10/zero-crossing-detection-with-pic16f877a.html

[3] http://tahmidmc.blogspot.com/2013/06/power-control-with-thyristor-phase.html

[4] A. Sarwar, M. S. J. Asghar, and F. I. Bakhsh, "Microcontroller Based Novel Dc-to-Ac Grid Connected Inverter Topology," Int. J. on Electrical and Power Engineering, USA, Vol. 02, No. 03, pp. 14-18, Nov 2011.