Experiment No. 03

3.1 Experiment Name

Study of fast and phase correct PWM generation using microcontroller.

3.2 Objectives

- To get a better understanding of Pulse Width Modulation (PWM) technique
- To understand the generation of PWM signal using an Arduino Uno
- To get familiar with the timer/counter and observe output waveform using oscilloscope

3.3 Theory

Pulse Width Modulation (PWM) is a technique used to generate an analog signal using a digital signal. PWM is employed in microcontroller applications to regulate motor speed, dim LEDs, and adjust the power supply to loads, among other tasks. The duty cycle of the PWM signal is controlled by changing the time that the output pin is high. This method, while simple to implement, lacks resolution and may not be suitable for high-frequency applications. Thus, modified techniques like fast PWM and phase-correct PWM provide a better solution to these traditional limitations.

In the fast PWM technique, the microcontroller rapidly alternates between high and low states, resulting in a high-frequency PWM signal. To be more precise, in fast PWM mode, the counter counts up to its maximum value and then resets to 0. The output at the output pin toggles every time the timer value equals the value of the OCR register. As a result, it offers more precise control over the duty cycle, ensuring high-speed modulation. In contrast, in phase-correct PWM mode, the counter starts counting down from its maximum upon reaching it before resetting back to 0. This cycle repeats. Here also, the output at the output pin toggles every time the timer value equals the value of the OCR register, resulting in a symmetric waveform. This helps minimize errors in certain control scenarios.

3.4 Apparatus

- ❖ Arduino Uno (for ATmega328 microcontroller)
- ❖ Oscilloscope
- ❖ Jumper wires
- ❖ Laptop
- ❖ Power supply

3.5 Connection diagram

3.5.1 Experimental connection

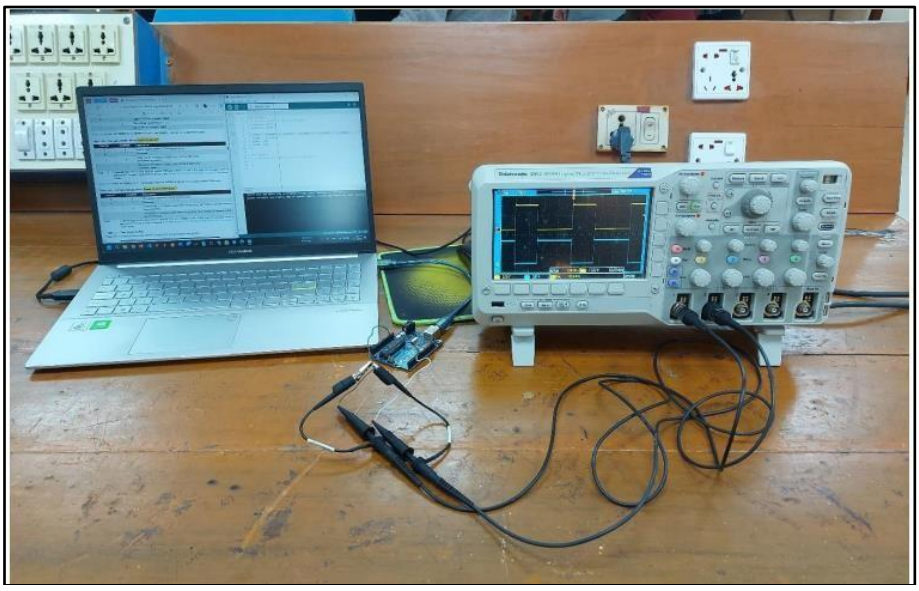

Figure 3.1: Experimental setup

3.6 Waveform

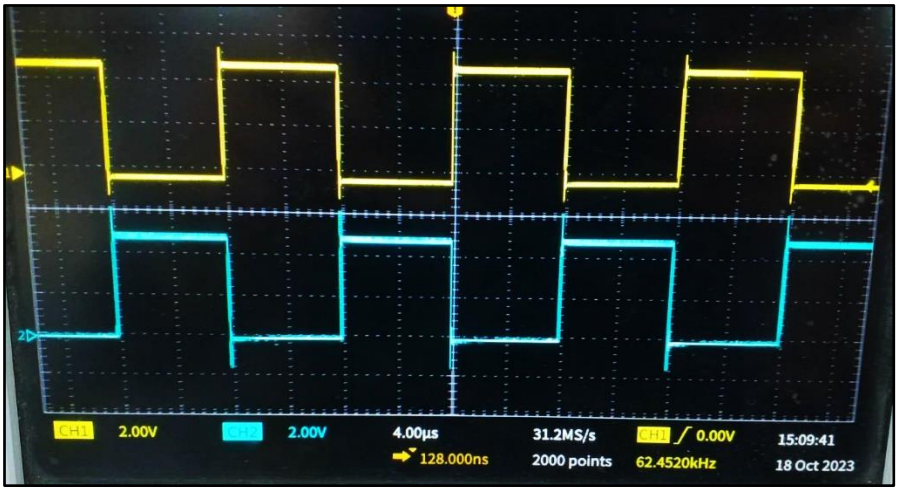

Figure 3.2: Fast PWM waveform

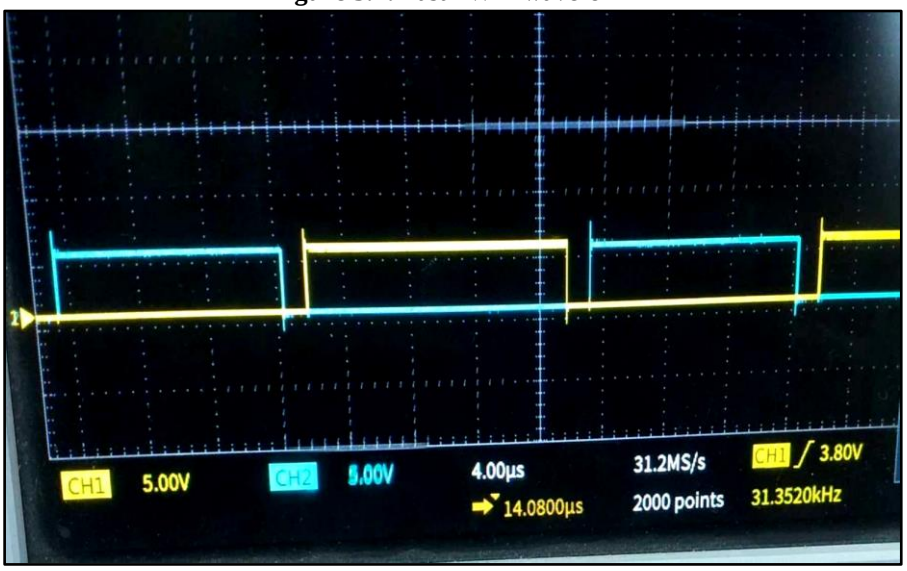

Figure 3.3: Phase Correct PWM waveform

3.7 Discussion & Conclusion

In this experiment, we successfully carried out the generation of a PWM signal using an Arduino Uno. Our approach involved meticulous programming of the Arduino Uno to align with the experimental requirements.

For the generation of fast PWM, we strategically configured digital pins 6 and 5 as the outputs for the PWM signal. Additionally, we tailored the settings of Timer 0 to operate in fast PWM mode. This entailed activating the WGM00 and WGM01 bits to enable fast PWM. Further customization was achieved by configuring COM0A1, COM0B1, and COM0B0 to dictate the behavior of the PWM outputs on pins 6 and 5. To fine-tune the duty cycle, we set the values in the Output Compare Registers (OCR0A and OCR0B) to 127 and 130, respectively. Conversely, in the case of phase-correct PWM generation, achieving our desired waveform was simplified. We only needed to set the values in the Output Compare Registers (OCR0A and OCR0B) to 127 and 150, respectively. These modifications effectively realized the desired waveform characteristics. The experiment conclusively demonstrated the successful generation of PWM signals with the Arduino Uno, showcasing the adaptability and precision achievable through careful configuration. The obtained results affirm the effectiveness of our programmed settings in producing the intended waveforms, marking the experiment as a success.