LED (PWM) Exercise

1.0 Learning Outcomes……1
2.0 Hardware Set-Up……1
3.0 Simulink Set-Up and PWM Testing……3
4.0 Additional Exercises……4
5.0 Concluding Remarks……4
Recommended Citation……4

1.0 Learning Outcomes

After completing this exercise, you should be able to demonstrate the Learning Outcomes listed in Table 1. At the end of the exercise, revisit the Learning Outcomes in Table 1 and confirm that each has been achieved.

Table 1: *Learning Outcomes from the LED (PWM) Exercise*

ID	Learning Outcome	Achieved
1.	Use the Simulink Support Package for Arduino Hardware to develop and deploy a simple PWM-based control algorithm.	☐
2.	Implement Pulse Width Modulation (PWM) to control LED brightness using an Arduino PWM-capable output.	☐

2.0 Hardware Set-Up

This exercise uses the same hardware configuration as the previous LED (Digital) Exercise. However, rather than simply switching the LED on and off, the LED is connected to a PWM-capable Arduino output pin so that its brightness can be varied using 8-bit PWM resolution, providing 256 discrete output levels.

Note: While the ACE-Box can be used for this exercise, it is not required here. Only the individual components listed below are needed.

Required hardware

- ATmega328P microcontroller
- Breadboard

- Breadboard jumper wires
  - 2 x male-to-male wires (10cm)
- USB Type-B cable
- LED
- 220Ω resistor

Hardware Assembly Steps

1. Connect digital pin 9 on the Arduino to a chosen column on the breadboard using a jumper wire.
2. Insert a 220Ω resistor with one end in the same column as the wire from pin 9 and the other end in a different row.
3. Insert the LED such that:
   - The long leg (anode) connects to the free end of the resistor.
   - The short leg (cathode) connects to an Arduino GND pin using a jumper wire.

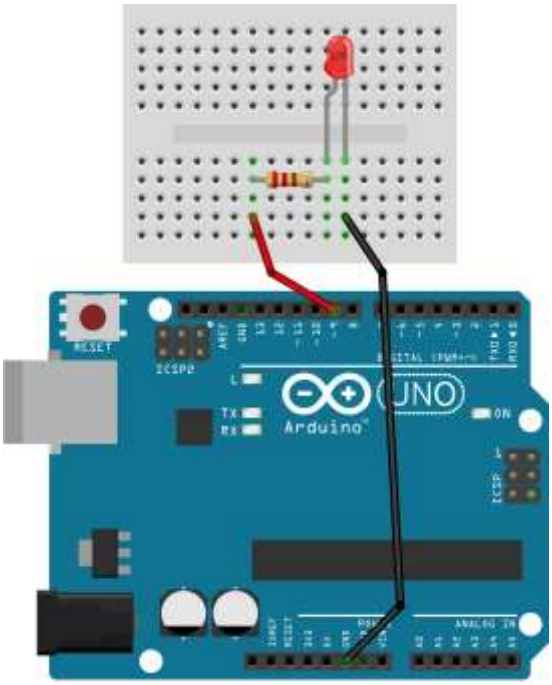


Figure 1: *Hardware Set-Up for LED (PWM) Exercise*

3.0 Simulink Set-Up and PWM Testing

In this part of the exercise, you will develop a Simulink model to control the Arduino PWM output and vary the LED brightness. A sine wave signal is used to vary the PWM command between 0 and 255, corresponding to approximately 0% to 100% duty cycle, as shown in Figure 2.

Simulink Model Construction Steps

1. Add a Sine Wave block (from *Sources*) and connect it to an Abs (absolute value) block (from *Math Operations*).

2. Add a Slider Gain block from Math Operations and configure it to adjust the maximum PWM command.
   - The Slider Gain can be used to limit the maximum PWM value applied to the LED.
   - For example, an approximate average output equivalent to **2.3 V** from a **5 V** PWM signal corresponds to:

$$\frac{2.3}{5}\ x\ 255 \approx 117$$

   - For this exercise, initially set the maximum value to 255, corresponding to a PWM duty cycle of approximately 100%.
3. Add a PWM block (from *Common* in the Simulink Support Package for Arduino Hardware) and connect three Scope blocks (from *Sinks*) to observe relevant signals.
4. Deploy the model using the same code-generation procedure as the previous exercise. Once deployed, observe the LED brightness variation and explore the effect of adjusting the Slider Gain.

For a reminder of the Code Generation process, refer back to the previous LED (Digital) Exercise.

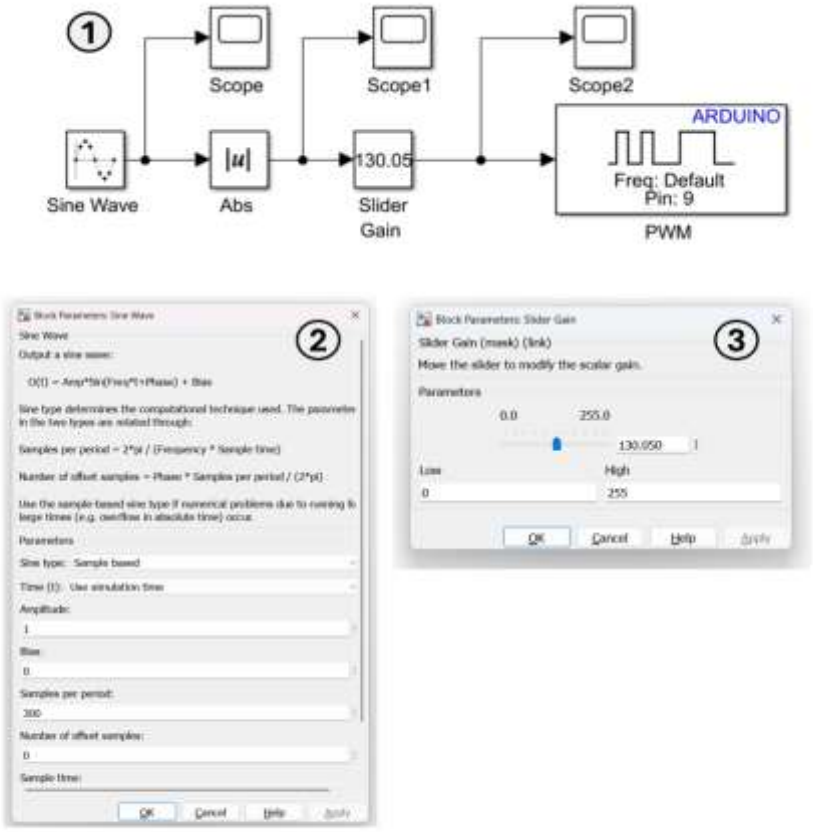


Figure 2: *Simulink Set-Up for LED (PWM) Exercise*

4.0 Additional Exercises

1. Modify the input signal used to drive the PWM block, for example by changing the sine wave frequency or amplitude. Observe and describe how these changes affect the rate and range of the LED brightness variation.
2. Define a new maximum PWM output requirement below 255. Update the Simulink model to satisfy this requirement and observe the effect on the LED brightness when deployed to the Arduino. As part of this exercise, investigate the use of a **Saturation** block in Simulink to constrain the PWM command to the specified range.

5.0 Concluding Remarks

This exercise has extended the previous digital LED control task by introducing Pulse Width Modulation (PWM) as a method of achieving variable output behaviour using a digital signal. By varying the PWM command, you have demonstrated how LED brightness can be adjusted across multiple levels rather than simply switched on or off.

Through the development and deployment of the Simulink model, you have gained practical experience in PWM control, signal scaling and automatic code generation for embedded hardware. The exercise has also demonstrated how changes made within a Simulink model can directly influence the behaviour of a physical system.

Before completing the exercise, return to the Learning Outcomes in Table 1 and confirm that you are confident you can demonstrate each one. If so, tick the corresponding Achieved box.

Recommended Citation

If you have used this practical exercise in your teaching, learning or research, please cite it using the reference below, adapting the format as required by your preferred referencing style.

> Pickering, J. E. (2026). LED (PWM) Exercise. ACE-Lab. Teaching Notes: ACE-Lab Practical Exercises, Version 1.0. https://zenodo.org/records/23098233