

Slide Potentiometer

1.0	Learning Outputs.....	1
2.0	Hardware Set Up.....	1
3.0	Simulink Set Up and Slide Potentiometer Testing.....	2
4.0	Additional Exercises.....	3
5.0	Concluding Remarks.....	3

1.0 Learning Outcomes

After completing this Exercise, you will be able to:

1. Understand how to map the analogue input reading from the slide potentiometer to a suitable range and use it to control the LED brightness via PWM.

After completing this Exercise, it is recommended that you refer back to the Learning Outcomes.

2.0 Hardware Set Up

This exercise involves connecting an LED to a PWM output pin on the Arduino and controlling its brightness using the 10-bit analogue input from the slide potentiometer. The potentiometer provides 1024 possible input levels, which are mapped directly to 1024 brightness settings for the LED, see Figure 1.

Required hardware for this exercise:

- Arduino Uno board (supported by Simulink)
- USB Cable Type A to B
- Breadboard
- LED
- 220 Ohm resistor
- Slide potentiometer
- 5 x male-male breadboard wires

Set-up the hardware as shown and following these steps:

1. Place the LED on the breadboard, connect its cathode to GND, and connect its anode through a 220 Ohm resistor to Pin 9.

2. Place the slide potentiometer on the breadboard, connect its outer pins to 5V and GND, and connect its middle pin (wiper) to A0. Note that potentiometers do vary in type and it might be worth undertaking some background reading to understand exactly how the 3 pins map.

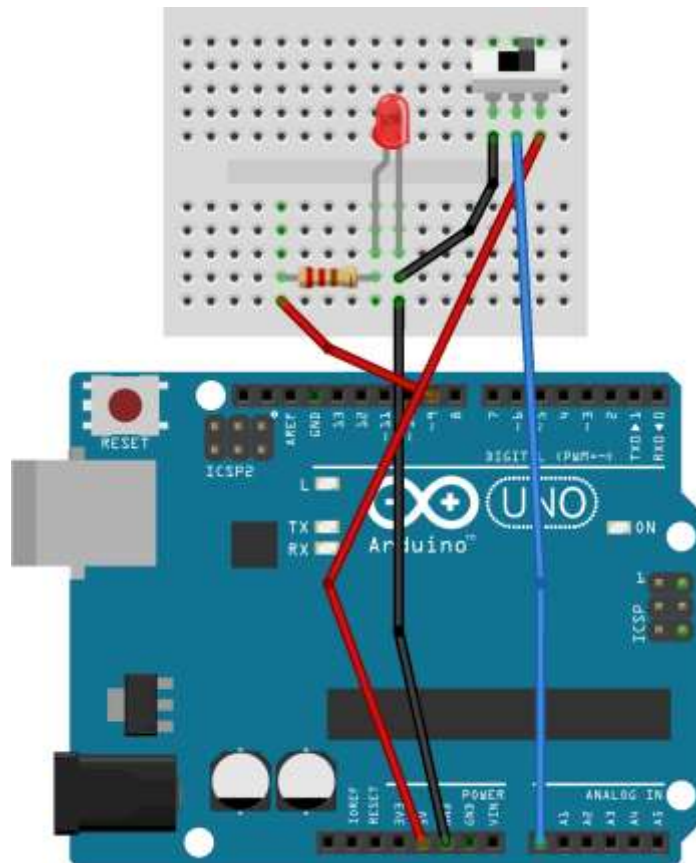


Figure 1: Hardware Set-Up for Slide Potentiometer Exercise

3.0 Simulink Set Up and Slide Potentiometer Testing

In this part of this initial exercise, you will develop a Simulink model to control the brightness of an LED using PWM, based on the analogue reading from a slide potentiometer, i.e., as the analogue reading of the slide potentiometer increases, so will the brightness of the LED. The steps for this exercise are very similar to those used for the LDR (light dependent resistor) exercise.

The steps to set up the model are as follows:

1. Add an Analog Input block and changing the pin number of A0 and sample time to 0.01. This reads the sensor voltage and outputs a value between 0 and 1023.
2. Insert a Gain block with value 255/1023. The gain block scales the ADC reading into the 0–255 range needed for PWM duty cycle.

3. Connect the scaled signal to a PWM Output block and change the Pin to 9. The PWM block uses the 0–255 value to generate a proportional PWM signal on the Arduino.
4. Connect one scope block to the signal after the gain block. This will enable the raw sensor reading to be viewed in real-time as the model runs, and also show the scaled PWM value, such that you can verify the conversion.
5. Deploy the model to the Arduino. Once uploaded, adjust the slide potentiometer position to see the PWM output change, and the brightness of the LED.

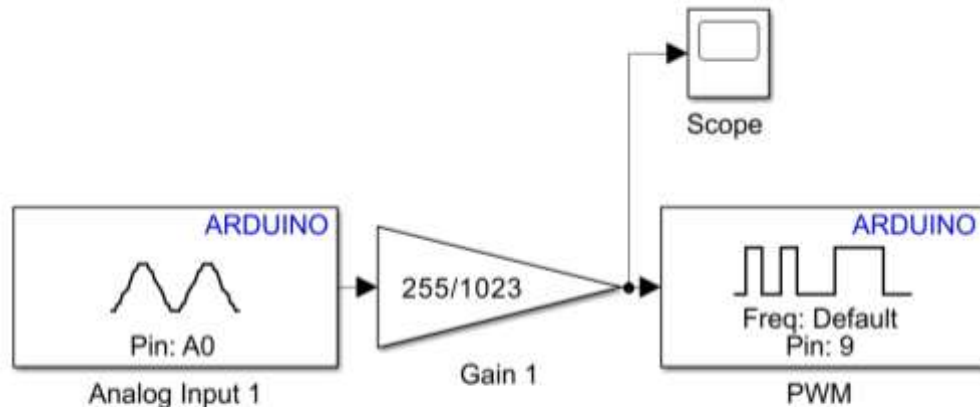


Figure 2: *Simulink Set-Up for Slide Potentiometer Exercise*

4.0 Additional Exercises

1. Modify the model so that increasing the potentiometer position decreases LED brightness.
2. Introduce a small deadband around mid-position so that small movements do not change brightness.
3. Add a second LED that activates when the potentiometer exceeds 75% of its range.

5.0 Concluding Remarks

This exercise demonstrates analogue signal acquisition, scaling, and PWM-based actuator control using Simulink and Arduino hardware. It reinforces the practical relationship between sensor resolution (10-bit ADC) and actuator resolution (8-bit PWM).

You have implemented a complete signal-conditioning chain: acquisition → scaling → actuation → verification. This mapping process is fundamental in

embedded control systems, where sensor signals must be appropriately transformed before driving actuators.

Recommended citation:

Pickering, J.E. (2026). Slide Potentiometer. ACE-Lab Teaching Notes. Available at: www.ace-lab.co.uk.