

LED (Digital) Exercise

| | | |
|-----|---------------------------------------|---|
| 1.0 | Learning outputs..... | 1 |
| 2.0 | Hardware Set Up..... | 1 |
| 3.0 | Simulink Set Up..... | 2 |
| 4.0 | Running Simulink Code Generation..... | 3 |

1.0 Learning Outcomes

After completing this Exercise, you will be able to:

1. Understand the use of the Simulink Arduino support package for algorithm design for a simple circuit, e.g., for an LED
2. Understand the functions of the Arduino Uno and the digital pins for use with a simple circuit, i.e., LED blinking on/off

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

2.0 Hardware Set Up

The exercise involves connecting an LED to a digital Arduino output pin and controlling its state by switching it on and off, see Figure 1.

Note that while the ACE-Box can be used for all the exercises, it is not required and only the individual components are needed.

Required hardware for this exercise:

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

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

1. Use a male-to-male jumper wire to connect Pin 9 on the Arduino to a chosen column on the breadboard.

2. Insert a 220Ohm resistor so that one end is in the same column as the wire from Pin 9. Place the other end of the resistor in a different row.
3. Insert the LED so that its long leg (anode) is connected to the free end of the resistor. Then complete the circuit by connecting the LED's short leg (cathode) to a GND pin on the Arduino using a male-to-male jumper wire.

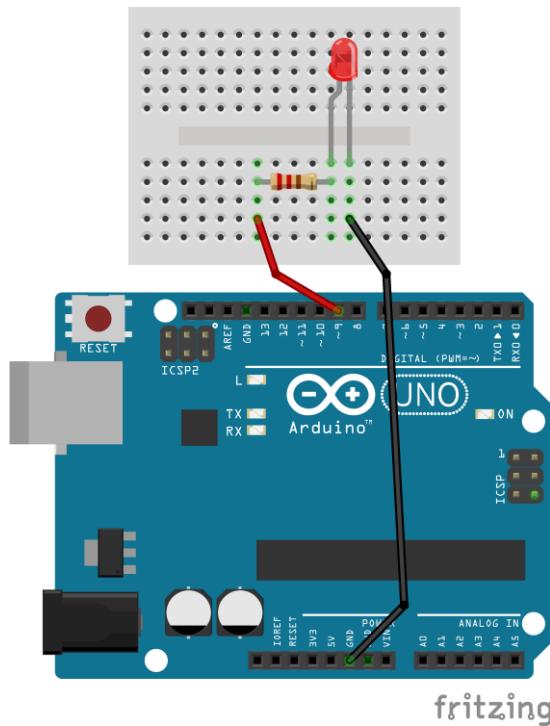


Figure 1: Hardware Set-Up for LED (Digital Exercise)

3.0 Simulink Set Up

In this part of the exercise, you will develop a Simulink model to control the Arduino digital pin output (i.e., to turn an LED on or off). A pulse generator will be used to change the alternate the value between zero (off) and 255 (fully on), see Figure 2.

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

1. Use a male-to-male jumper wire to connect Pin 9 on the Arduino to a chosen column on the breadboard.
2. Insert a 220 Ohm resistor so that one end is in the same column as the wire from Pin 9. Place the other end of the resistor in a different row.
3. Insert the LED so that its long leg (anode) is connected to the free end of the resistor. Then complete the circuit by connecting the LED's short leg (cathode) to a GND pin on the Arduino using a male-to-male jumper wire.

Key Properties to Modify

1. **Pulse type** → Set to Sample-based
2. **Sample time** → Set to 0.1 seconds

All other properties may remain as their default values.

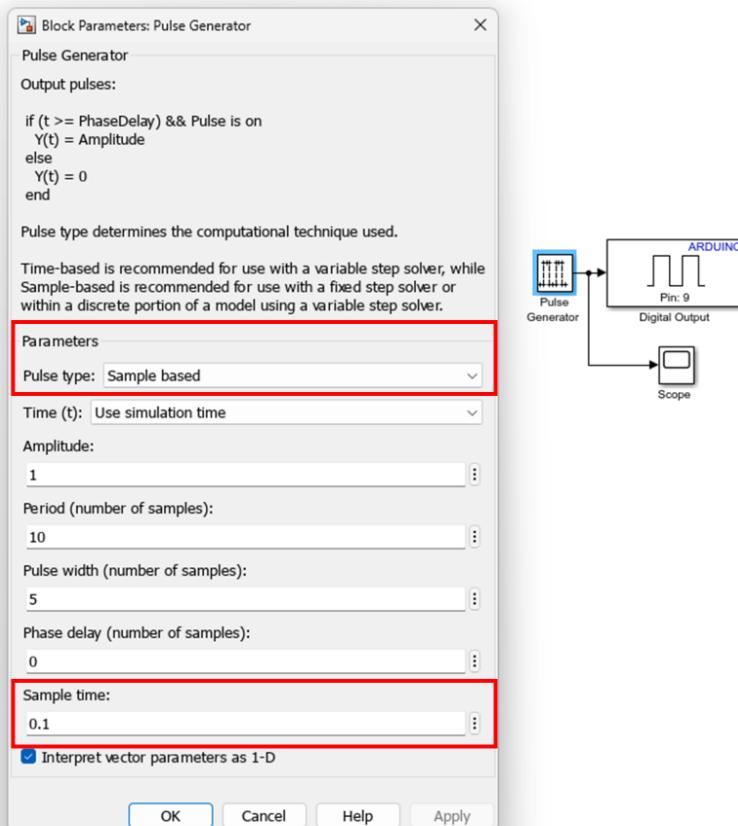


Figure 2: Simulink Set-Up for LED (Digital Exercise)

4.0 Running Simulink Code Generation

In this part of the exercise, you will configure and run a Simulink model to perform code generation on a supported Arduino Uno board. Using the following steps and as shown in the illustration, and referring to Figure 3, undertake the following:

1. Connect the Arduino Uno to your computer using a USB cable.
2. In your Simulink window, open Model Settings by selecting Modelling → Model Settings. This will bring up the Configuration Parameters dialog.
3. From the left-hand menu, select Hardware Implementation. Under Hardware board, choose Arduino Uno from the list, then click Apply and OK. Note that other boards can be used beyond that of the Arduino Uno.

4. A new Hardware tab will now appear in Simulink. (Ensure your Simulink model is already developed before reaching this step.)
5. Important: make sure MATLAB is operating/saving in a suitable working directory before clicking 'Run'
6. Confirm that the Arduino Uno is displayed, change the simulation stop time to inf (infinity), and then click Monitor & Tune. The Simulink model will now be deployed to the Arduino, compiled into C code, and executed on the hardware.

Key Properties to Modify

1. **Pulse type** → Set to Sample-based
2. **Sample time** → Set to 0.1 seconds

All other properties may remain as their default values.

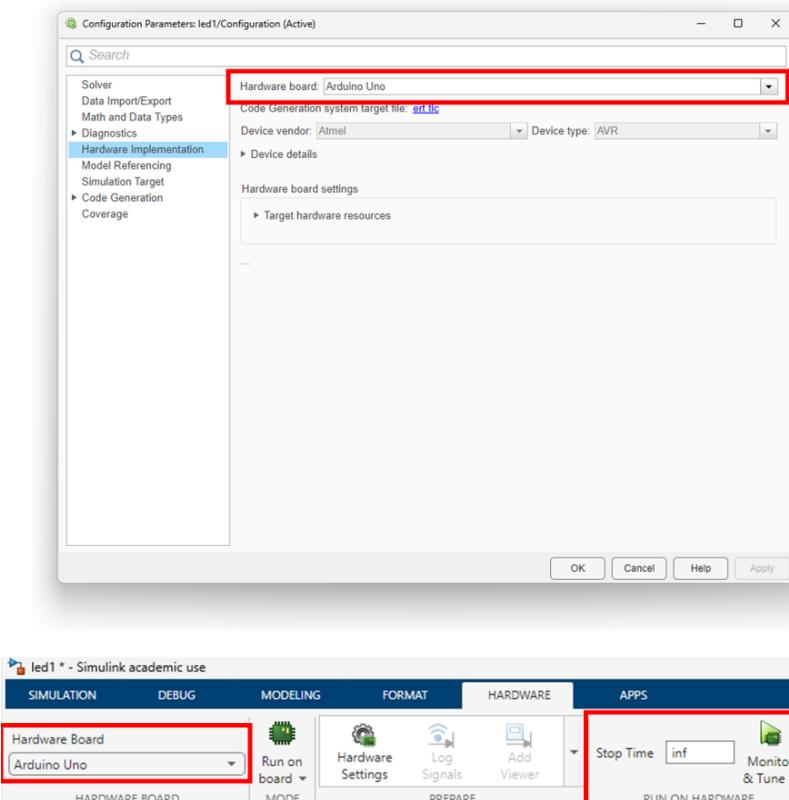


Figure 3: Running Simulink Code Generation

⚠ If you encounter any errors, click the link [HERE](#) for troubleshooting help.