LED (Digital) Exercise

1.0 Learning Outcomes……..……..……..…1
2.0 Hardware Set-Up……..……..….1
3.0 Simulink Set-Up……...…..…..2
4.0 Running Simulink Code Generation……..……..…….4
5.0 Exercises……..……..……..……5
6.0 Concluding Remarks……..……..……..……..5
Recommended Citation……...5

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 (Digital) Exercise*

ID	Learning Outcome	Achieved
1.	Use the Simulink Support Package for Arduino Hardware to develop and deploy a simple control algorithm.	☐
2.	Configure and use an Arduino Uno digital output pin to control an LED.	☐

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/off, as shown in Figure 1.

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
    a. 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 male-to-male 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) is connected to the free end of the resistor.
    - The short leg (cathode) is connected to an Arduino GND pin using a jumper wire.

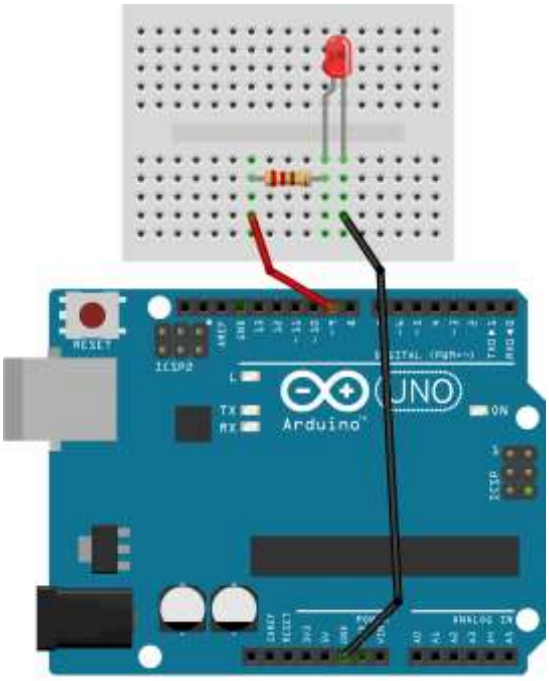


Figure 1: *Hardware Set-Up for LED (Digital) Exercise*

3.0 Simulink Set-Up

In this part of the exercise, you will develop a Simulink model to control the Arduino digital output pin and turn the LED on/off. A Pulse Generator block is used to alternate the output between 0 (off) and 255 (on), as shown in Figure 2.

Simulink Model Construction Steps

1. From Simulink → Sources, add a Pulse Generator block to the model.
2. From Simulink Support Package for Arduino Hardware → Common, add a Digital Output block.

3. Connect the output of the Pulse Generator block to the input of the Digital Output block.
4. Open the Pulse Generator block parameters and configure: Pulse Type: *Sample based* and Sample time: 0.1 seconds.
5. Open the Digital Output block parameters and set: Pin number: 9.
6. From Simulink → Sinks, add a Scope block.
7. Branch the signal from the Pulse Generator output and connect it to the Scope input to monitor the generated waveform.

Key Block Parameters for Pulse Generator to Modify

- Pulse type → Sample-based
- Sample time → 0.1 s

All other parameters may remain at their default values.

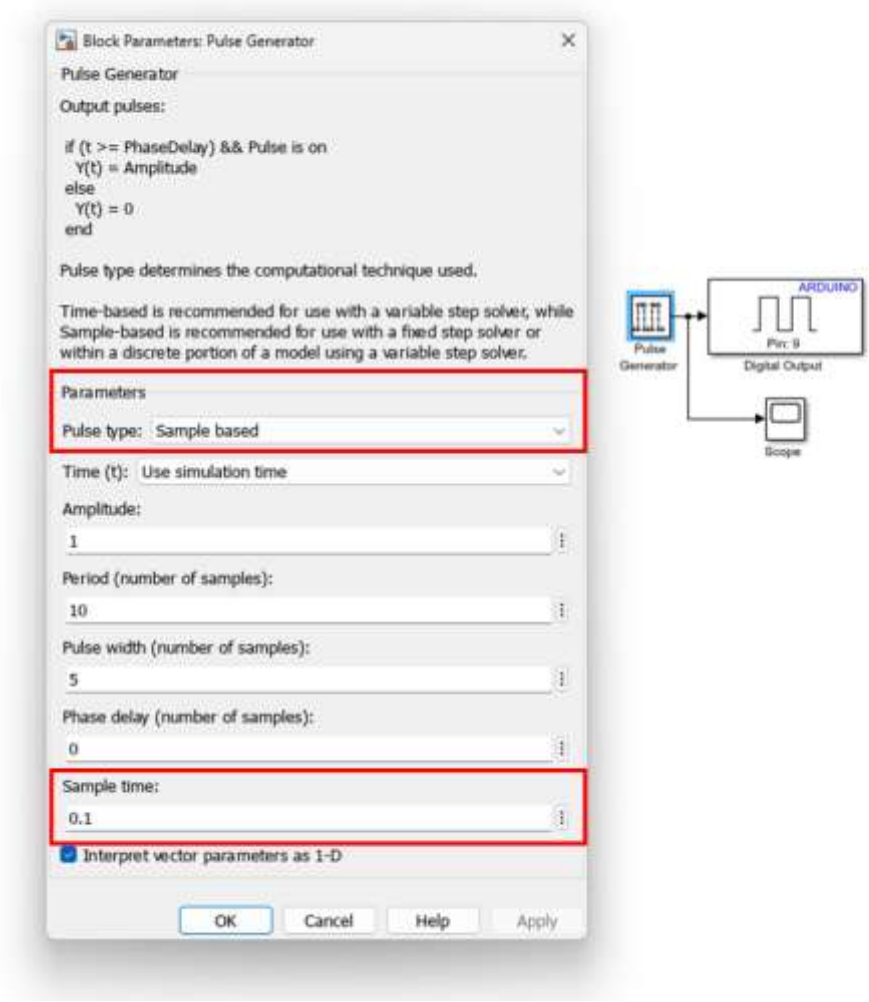


Figure 2: *Simulink Set-Up for LED (Digital Exercise)*

4.0 Running Simulink Code Generation

In this part of the exercise, you will configure the Simulink model for code generation and deploy it to the Arduino Uno, as shown in Figure 3.

1. Connect the Arduino Uno to your computer using a USB cable.
2. In Simulink, open Model Settings by selecting *Modelling → Model Settings*.
3. Select Hardware Implementation from the left-hand menu and choose Arduino Uno as the hardware board. Click Apply and OK.
4. Confirm that the Hardware tab appears in Simulink.
5. Ensure MATLAB is operating in a suitable working directory before running the model.
6. Set the simulation stop time to inf (infinity) and click Monitor & Tune to deploy the model.

The Simulink model will now be compiled into C code and executed on the ATmega328P microcontroller.

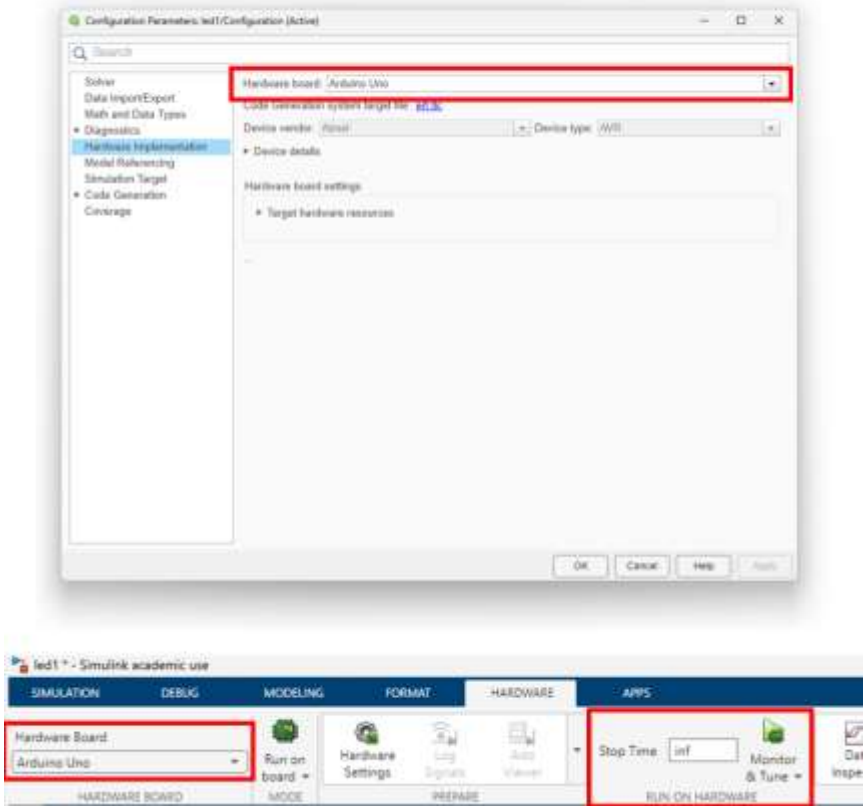


Figure 3: *Running Simulink Code Generation*

5.0 Exercises

1. Define new values for the LED on/off timing and sample time. Update the relevant Pulse Generator parameters in the Simulink model, deploy the revised model to the Arduino Uno, and verify that the LED operates according to the new timing requirements.
2. Extend the system by adding a second LED connected to a different Arduino digital output pin.
    - 2.1 Define a new set of requirements so that the two LEDs operate out of sequence with one another.
    - 2.2 Modify the Simulink model to implement the required phase-shifted behaviour.
    - 2.3 Deploy the updated model and verify that the two LEDs operate according to the defined requirements.

6.0 Concluding Remarks

This exercise has demonstrated the complete workflow for developing, configuring and deploying a simple Simulink model to an ATmega328P microcontroller. By controlling an LED through an Arduino digital output, you have gained practical experience in moving from a Simulink model to executable behaviour on physical hardware.

The exercise has introduced key concepts including digital input/output (I/O), sample-based timing and automatic C code generation using the Simulink Support Package for Arduino Hardware. It has also demonstrated how Simulink block diagrams can be deployed directly to embedded hardware without the need to manually write the underlying C code.

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 (Digital) Exercise. ACE-Lab. Teaching Notes: ACE-Lab Practical Exercises, Version 1.0. https://zenodo.org/records/23098135