Tactile Button Switch Exercise

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 containing an tactile button switch
2. Understand the functions of the Arduino Uno and the digital pins for use with a simple circuit (e.g., using PWM to control an LED in combination with a tactile push-button switch)

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 switch that can contract the state of the LED being on or off via physical force on the tactile button switch, 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
- Tactile button switch

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

1. Insert the tactile button switch so that its legs are positioned across the centre gap of the breadboard, ensuring each pair of legs sits on opposite sides where no internal connection exists.
2. Use a male-to-male jumper wire to connect Pin 9 on the Arduino to one leg of the tactile switch (choose a column that connects to only that leg).
3. Insert a 220Ohm resistor so that one end is in the same column as the opposite leg of the tactile switch (i.e., the leg that is not connected to Pin 9). Place the other end of the resistor in a separate row.
4. Connect the long leg (anode) of the LED to the free end of the resistor. Finally, complete the circuit by running a male-to-male jumper wire from the LED's short leg (cathode) to a GND pin on the Arduino.

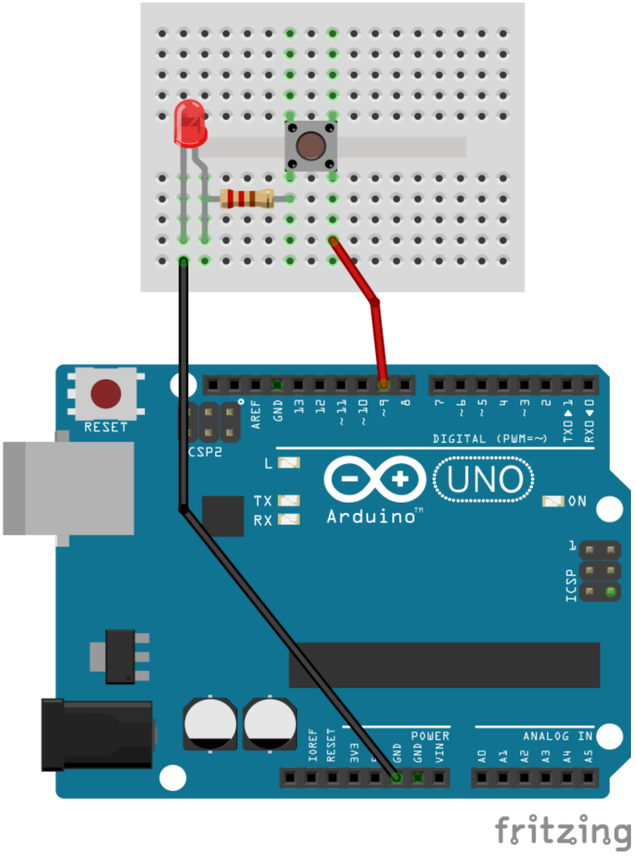


Figure 1: *Hardware Set-Up for Tactile Button Switch Exercise*

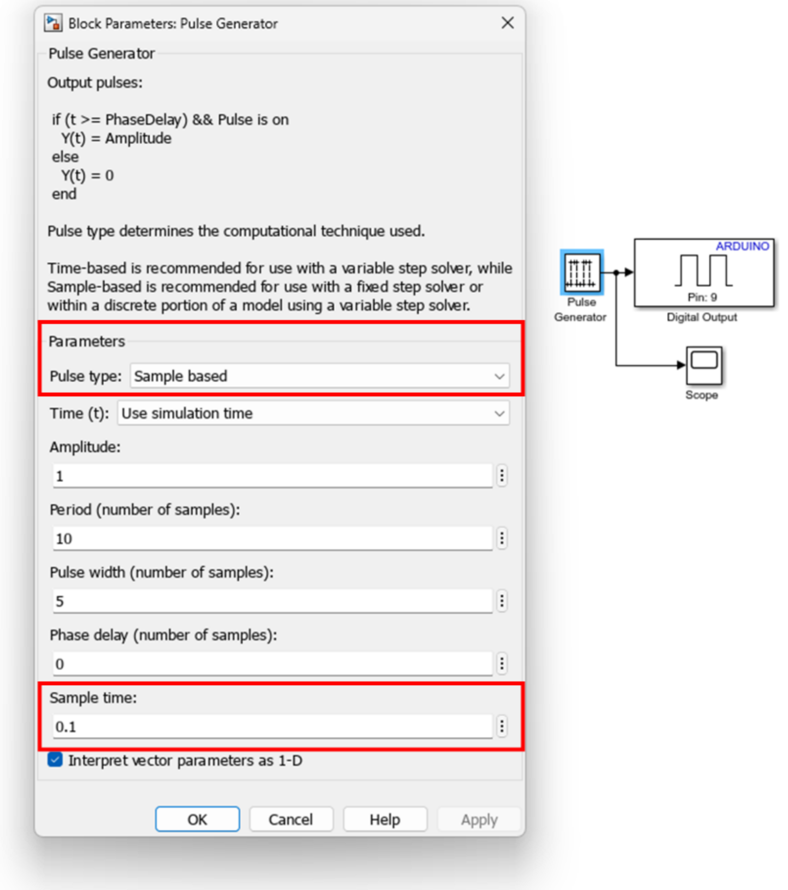
3.0 Simulink Set Up and Tactile Button Switch Testing

As a tactile button switch has been added to the hardware (i.e., there are no additional inputs/outputs), the Simulink setup is the same as the LED (PWM) and on/off power switch exercises, see Figure 2. However for completeness, the full set-up is explained again - with a demonstration provided via the video recording below.

The steps to set up the model are as follows (again, these are the same as the LED (PWM) and on/off button switch exercises) setup:

1. Use a Sine Wave block (from Sources) as the input and connect it to an Abs (absolute value) block (from Math Operations), as shown in (2).
2. Connect a Slider Gain block (from Math Operations). You can specify the lower (Low) and upper (High) limits of the gain to constrain the output voltage. For example, to limit the voltage to 2.3 V: 256 × 2.3 = 117.76 ≈ 117. In this case, the upper limit (High) for the Slider Gain would be set to 117. However, for this exercise, set the upper limit to 255 (equivalent to 5 V on the Arduino Uno) (3).
3. Add a PWM block (from Common in the Simulink Support Package for Arduino Hardware) and connect three Scopes (from Sinks).
4. Then run the steps from the previous exercise to perform the code generation (a recap link is provided below), to create an LED that is varying in brightness! Explore the effect of the gain value within the Slider Gain. Watch the video below for further help!

Recall how to perform code-generation by visiting HERE.



Block Parameters: Pulse Generator

Pulse Generator

Output pulses:

```
if (t >= PhaseDelay) && Pulse is on
  Y(t) = Amplitude
else
  Y(t) = 0
end
```

Pulse type determines the computational technique used.

Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or within a discrete portion of a model using a variable step solver.

Parameters

Pulse type: Sample based

Time (t): Use simulation time

Amplitude: 1

Period (number of samples): 10

Pulse width (number of samples): 5

Phase delay (number of samples): 0

Sample time: 0.1

☑ Interpret vector parameters as 1-D

Figure 2: *Simulink Set-Up for Tactile Button Switch Exercise*

Key Properties to Modify

1. **Sine Wave (Sine type)** → Set to Sample-based
2. **Sine Wave (Sample per period)** → Set to Sample-based
3. **Sine Wave (Samples per period)** → Set to 300
4. **Sine Wave (Sample time)** → Set to 0.01 seconds
5. **Slider Gain (High)** → Set to 255

All other properties may remain as their default values.

⚠ If you encounter any errors, click the link HERE for troubleshooting help.