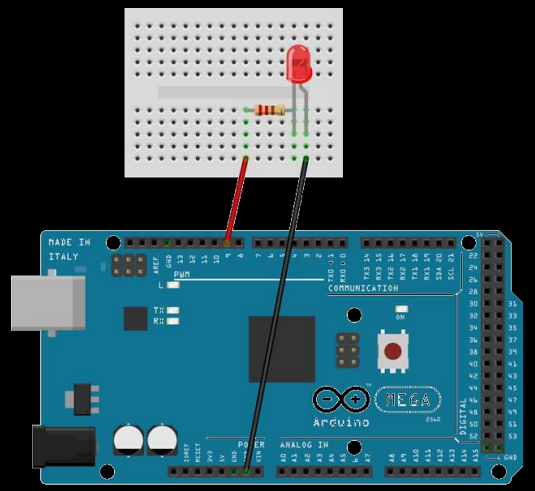
Contents

Simulink for Code Generation and Deployment: LED Exercise

Key Learning Points

After this Lecture, 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/Mega digital pins for use with a simple circuit
3. Understand the operation of pulse width modulation (PWM) and saturation limit



1.1 Installing Simulink Support Package for Arduino Hardware

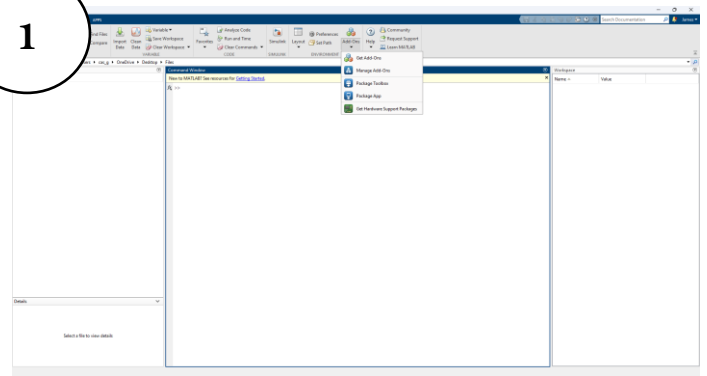
- **Undertake the following steps:**

1) Click on '*Add-Ons*' on the Toolbar and then '*Get Hardware Support Package*'

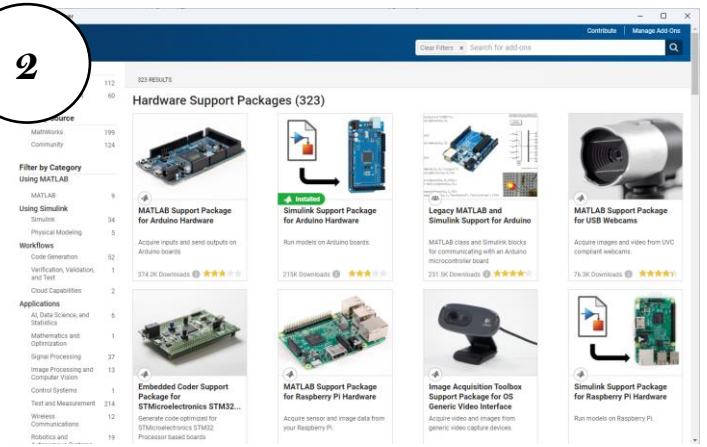
2) Search for the '*Simulink Support Package for Arduino Hardware*' and click on it

3) Then click on '*Install*' and then follow the steps-through

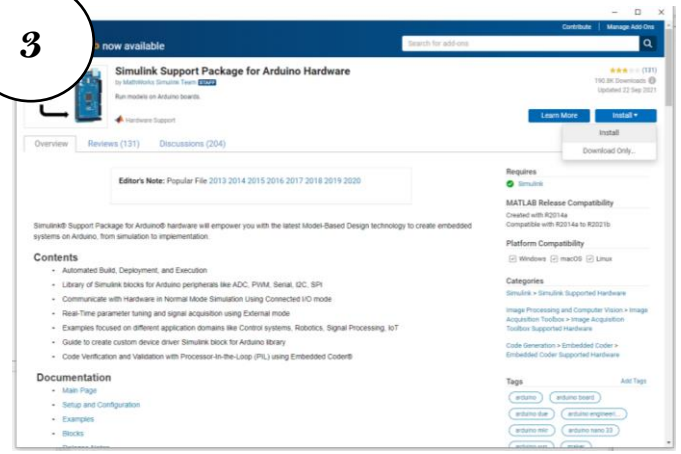
1



2



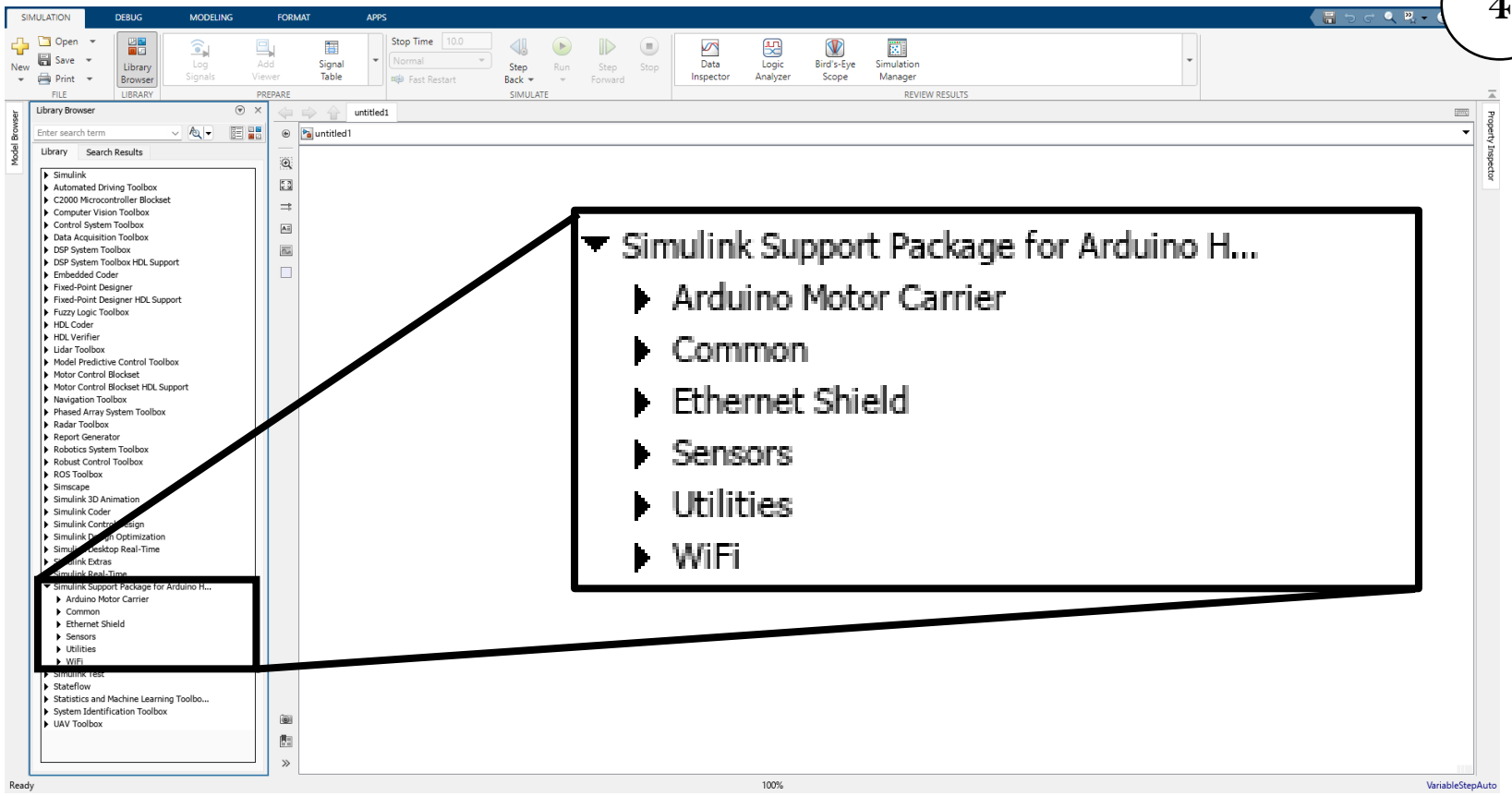
3



1.1 Installing Simulink Support Package for Arduino Hardware

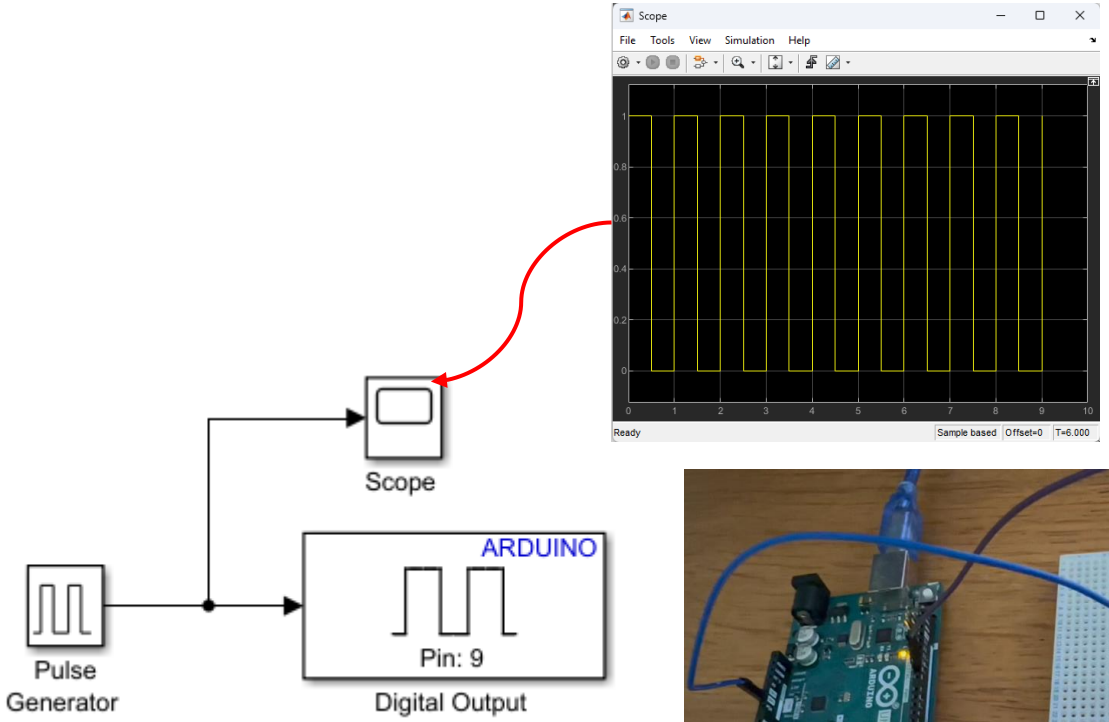
- **Undertake the following steps:**
  4) After opening a '*Blank Model*' within Simulink, open the '*Library Browser*' – the '*Simulink Support Package for Arduino Hardware*' is located here containing the blocks that will be used with the Arduino Uno

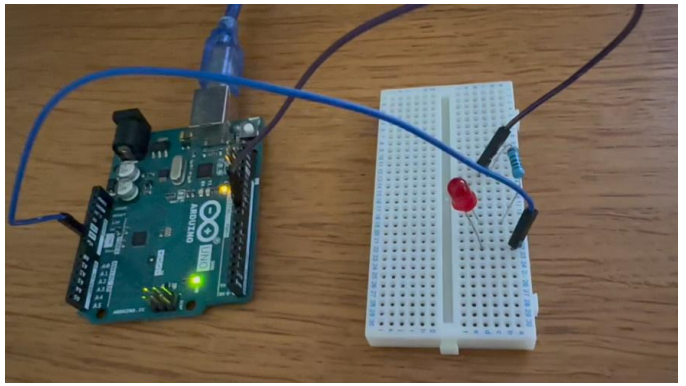
4

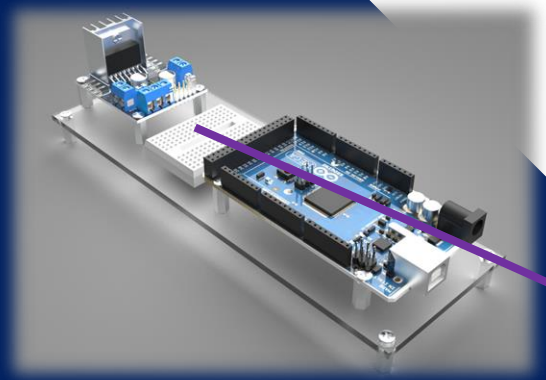


1.2 LED Exercises

- The exercise involves connecting an LED to an Arduino output pin and implemented changes to the state of the LED
- In the examples, MATLAB and Simulink Version 2023a is used
- **Important:** this initial exercise uses a mixture of Arduino Uno and Arduino Mega 2560. This is to show that either of the boards can be used (and you will likely use different boards!).





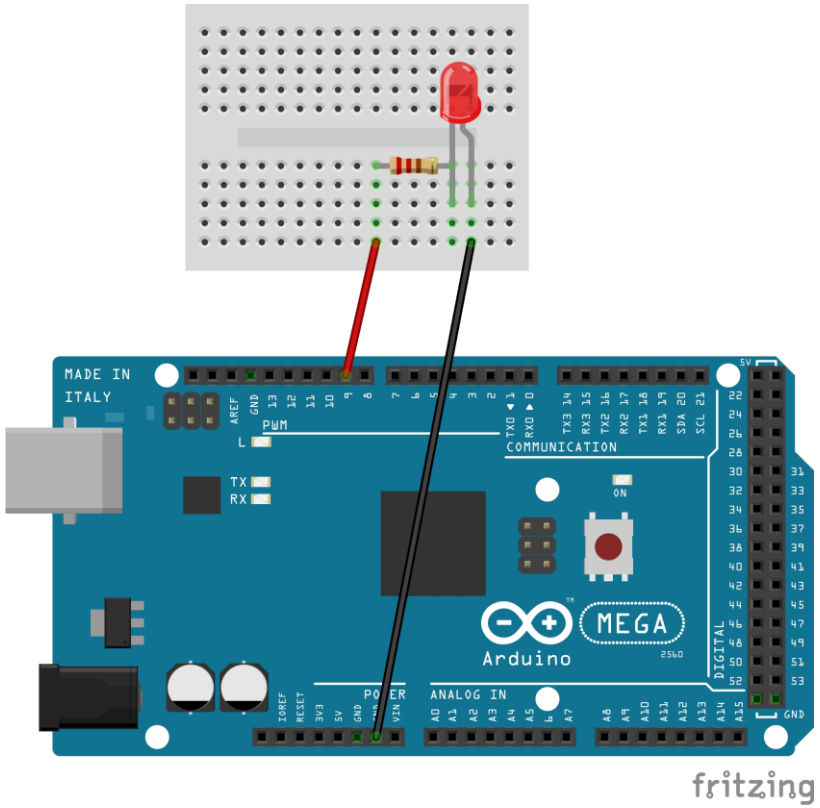


1.2.1 Hardware

- Required hardware for this exercise:
  i. Supported Arduino Mega 2560 board
  ii. USB cable
  iii. Small breadboard
  iv. LED
  v. 220 Ohm resistor
  vi. 2 x male-male breadboard wires
- A breadboard is a device used to make simple electrical connections – each column of the breadboard is connected with a break in the centre

- The steps for this part of the exercise are as follows:
  i. Attach one end of the 220 Ohm resistor to output pin 9 on the Arduino board, using the small breadboard and male-male breadboard wires
  ii. Attach the long leg (positive) of the LED to the resistor

Breadboard found on control-lab-in-a-box (CLB) rig



- To determine the required resistor, Equation (1-1) is used:

$$R = \frac{V_s - V_{LED}}{I_{LED}} \qquad (1\text{-}1)$$

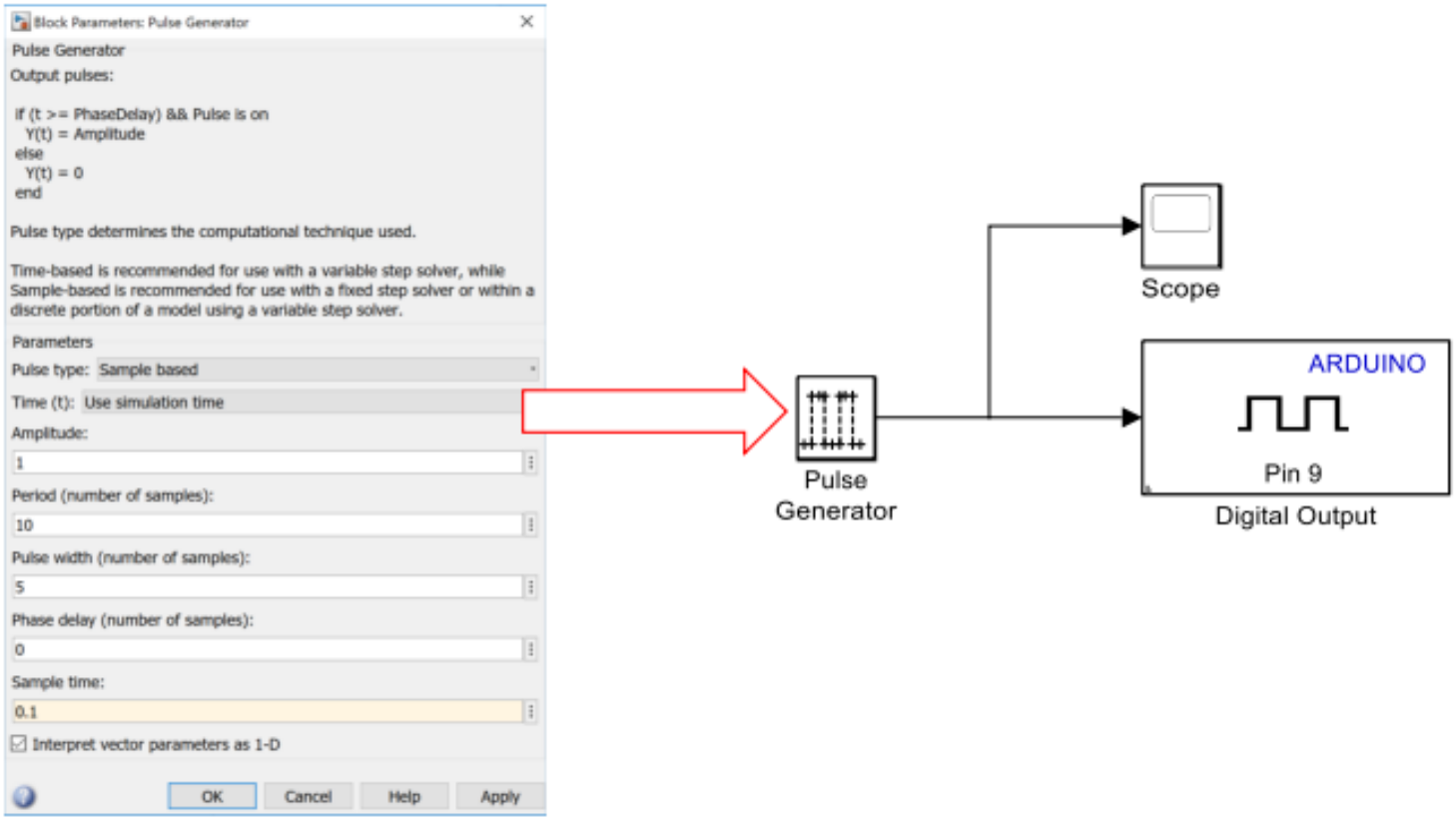
where V_s is the source voltage (Volts/V), V_{LED} is the voltage drop across the LED (Volts/V) and I_{LED} is the current through the LED (Amps/A) and R is the resistance (Ohms/Ω)

- Using Equation (1-1), $R = \frac{5.00-2.90}{0.01} = 210\Omega$
- Therefore a 220 Ohm resistor is used

1.2.2 Algorithm Design – On/Off

- In this part of the exercise a Simulink model will be developed to change the state of the Arduino digital output to being either on or off

- The steps for this part of the exercise are as follows:
  i. Set-up the pulse generator as shown below (found in 'Sources') and connect this to a digital output block (found under 'Common' in the 'Simulink Support Package for Arduino Hardware'), with a scope connected (found in 'Sinks')
  ii. Amplitude of 1 implied a value of $5V$ will be supplied to Pin 9 (by default the 'Digital Output' block will be Pin 9)



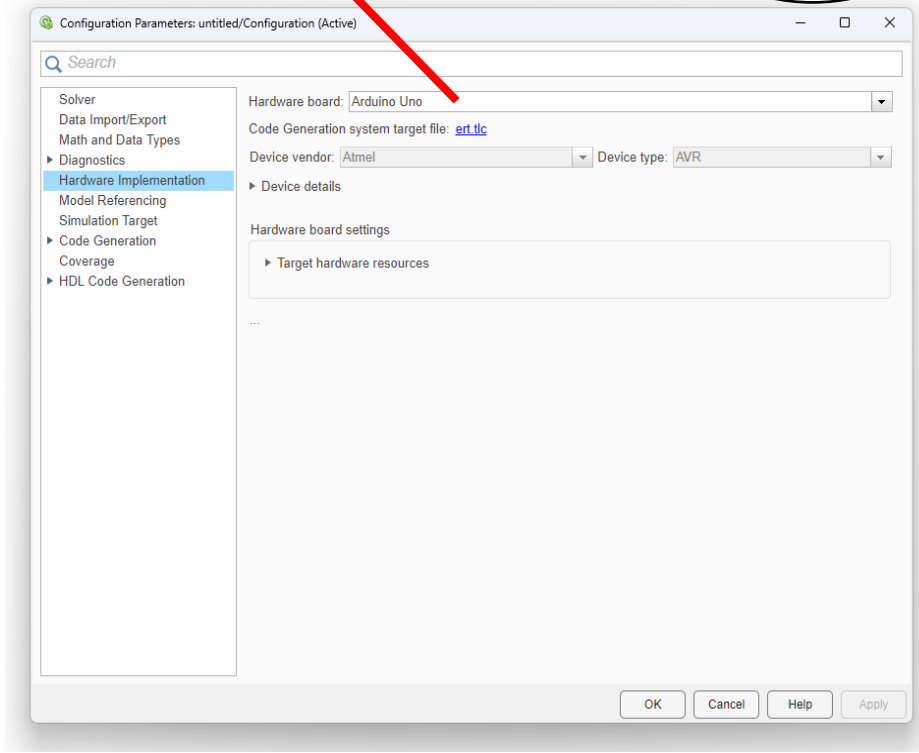
1.2.2 Algorithm Design – On/Off

- In this part of the exercise, the Simulink model will be configured and run on the supported Arduino Uno/Mega board
- **Important:** make sure MATLAB is operating/saving in a suitable working directory before clicking 'Run'

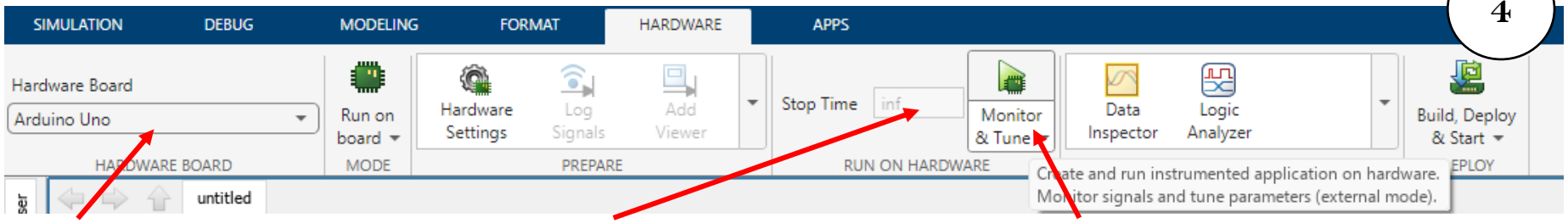
1. Connect the Arduino board to your computer with a USB cable
2. When on your Simulink Window, click on '*Modelling*' and then '*Model Settings*' to open '*Configuration Parameters*'
3. Select the '*Hardware Implementation*' panel and within the '*Hardware Board*' tab, select the '*Arduino Uno*' (or Arduino Mega 2560 – depending on the board being used) hardware from the '*Hardware Board*' list, then click '*Apply*' and then '*OK*'
4. Now in your Simulink model, the 'Hardware' tab will appear (Note: your Simulink model should be developed before undertaking this step). You should see the '*Arduino Board*'. Now change the simulation stop time to '*inf*' (infinity) and then click '*Monitor & Tune*'. The Simulink model (algorithm design) will now be deployed (converted to C code) to the connected Arduino hardware.
5. Look at the LED attached to pin 9. The LED should blink once time every second.
6. See the next slide if you're experiencing errors

Arduino Mega 2560 can also be selected here

2 & 3



4

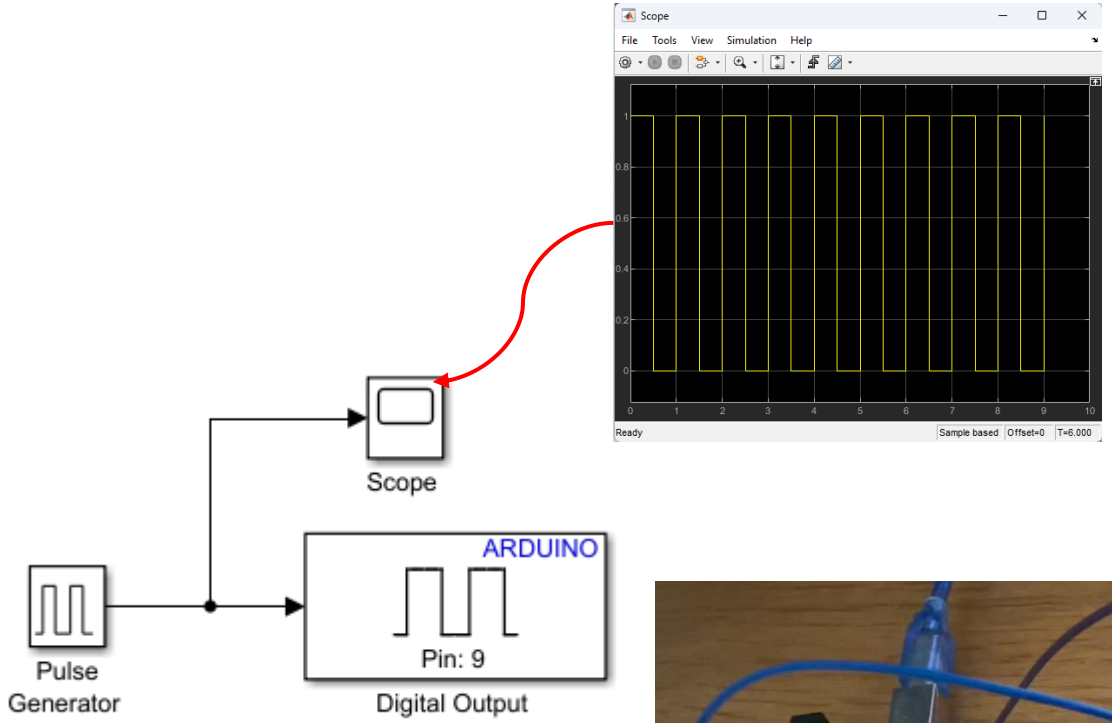


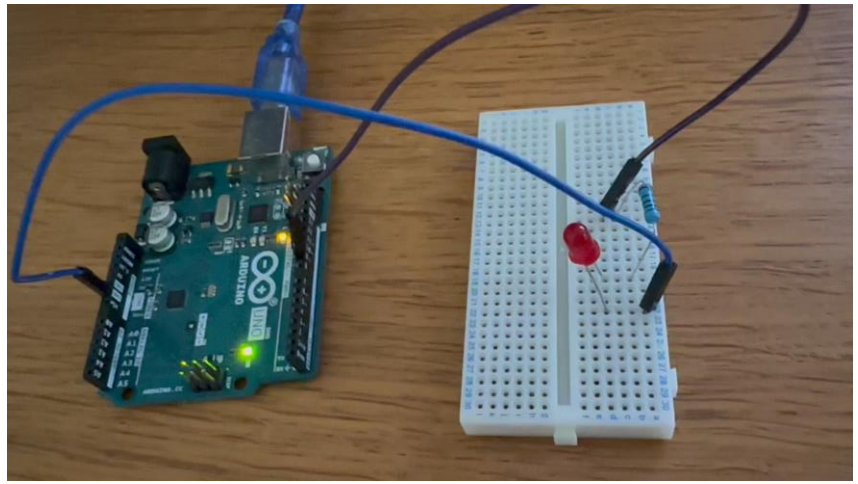
Linked board

Simulation stop time (i.e., infinity')

Embed code

1.2.2 Algorithm Design – On/Off

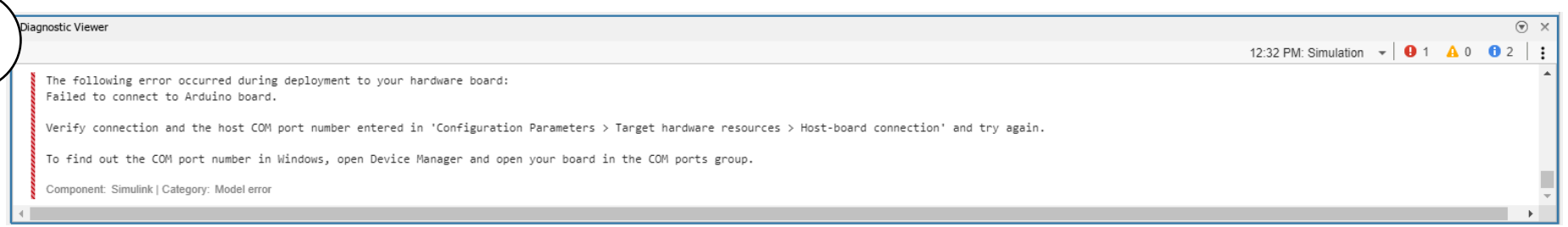




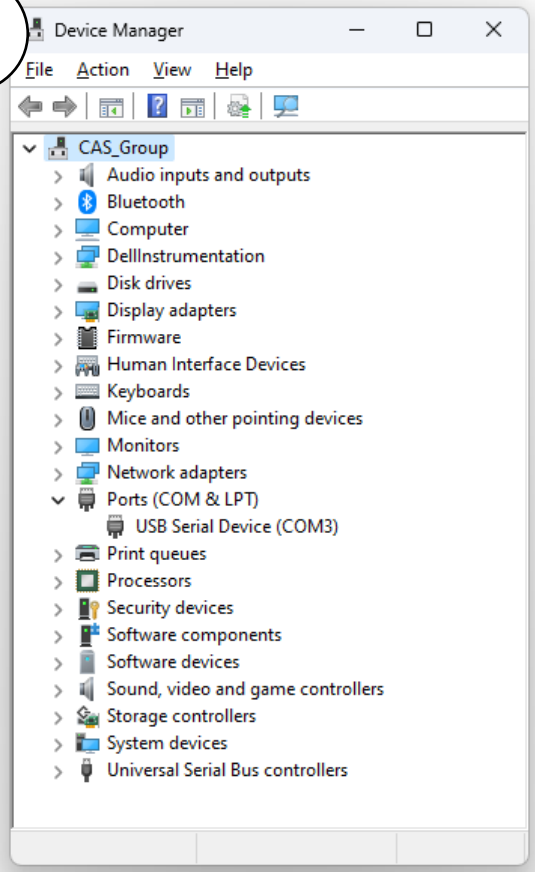
1.2.3 Error Messages

- Common error 1: board not connected to the correct COM
  1. Error message – failed to connect to Arduino board
  2. To potentially solve this, check the Port used within the Device Manger
  3. Then alter the 'Host Com Port' applied within the 'Configuration Parameters' by selecting 'Manually Specify' within 'Set host COM port'

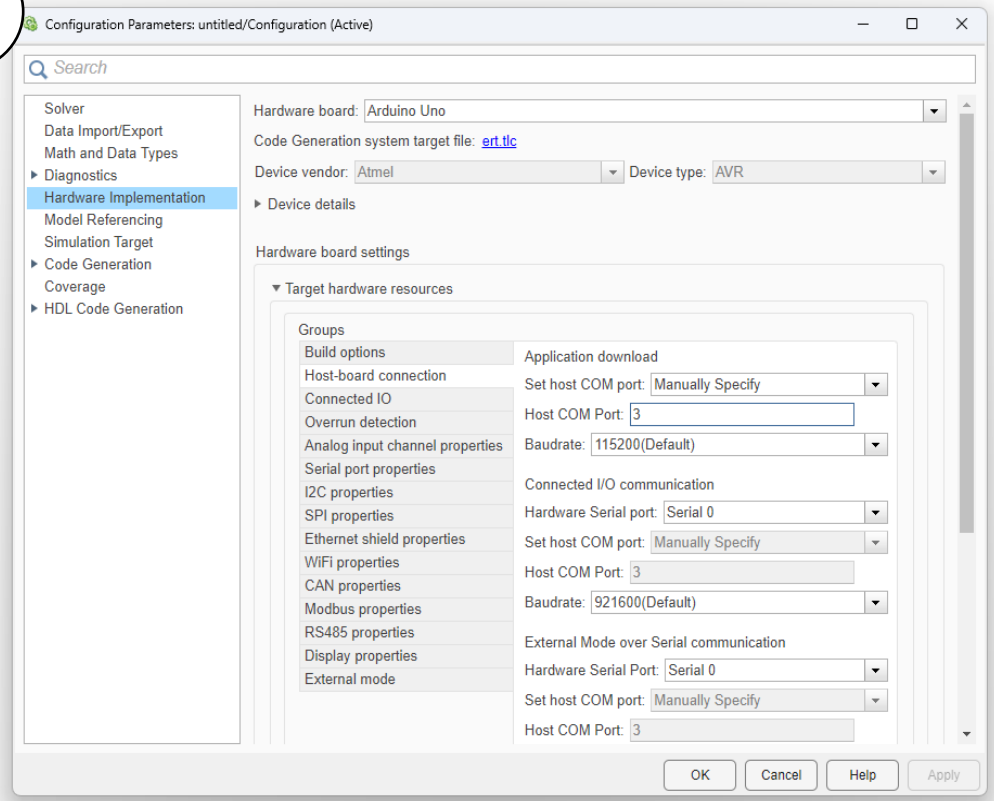
1



2



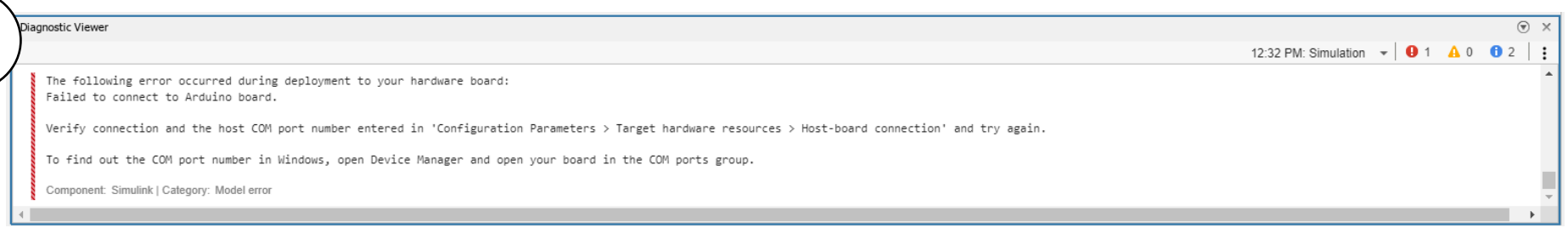
3



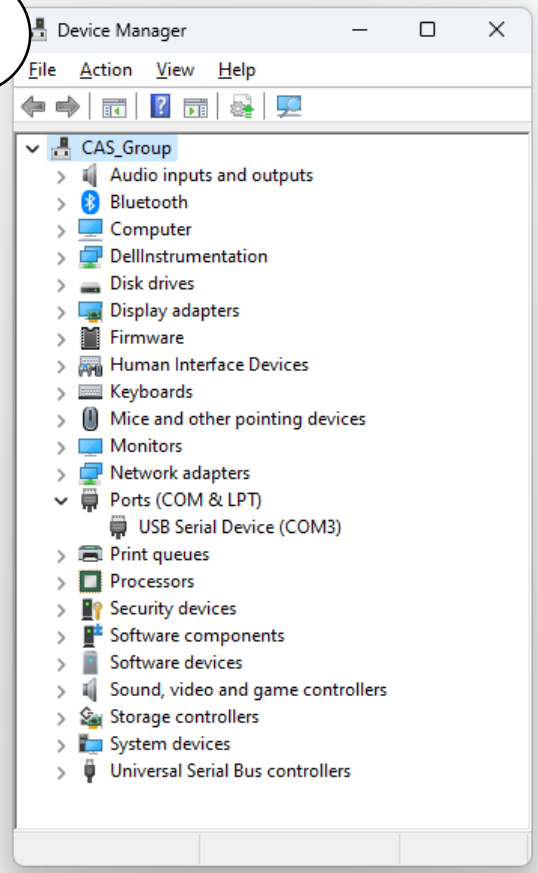
1.2.3 Error Messages

- Common error 1: need to update drivers
  1. Error message – failed to connect to Arduino board
  2. To potentially solve this, within the Device Manager, locate the USB port for the Arudino board and click on it
  3. With the 'Driver' tab, click on 'Update Driver'

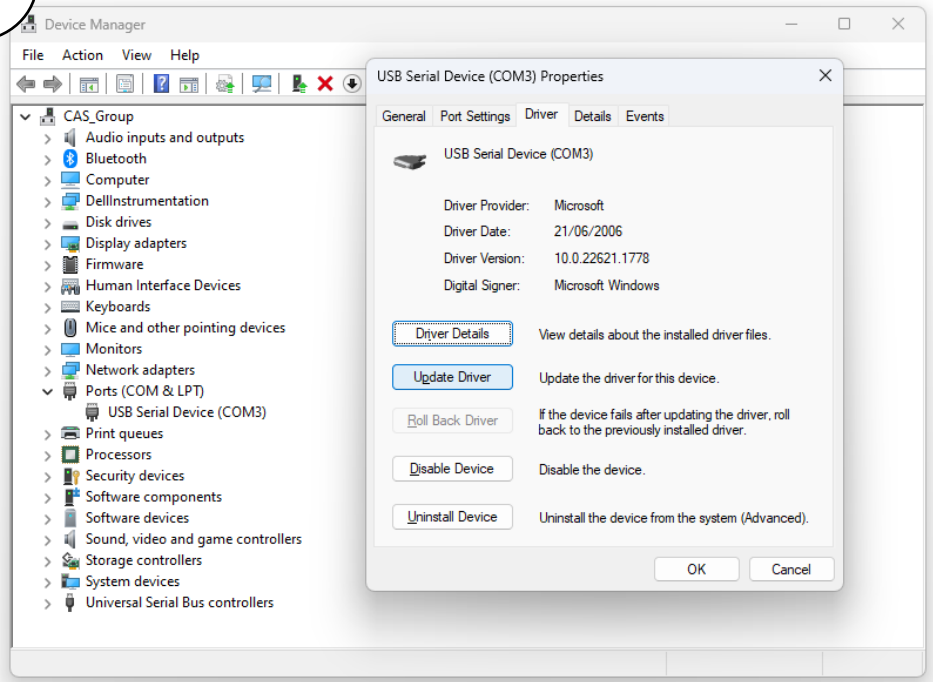
1



2

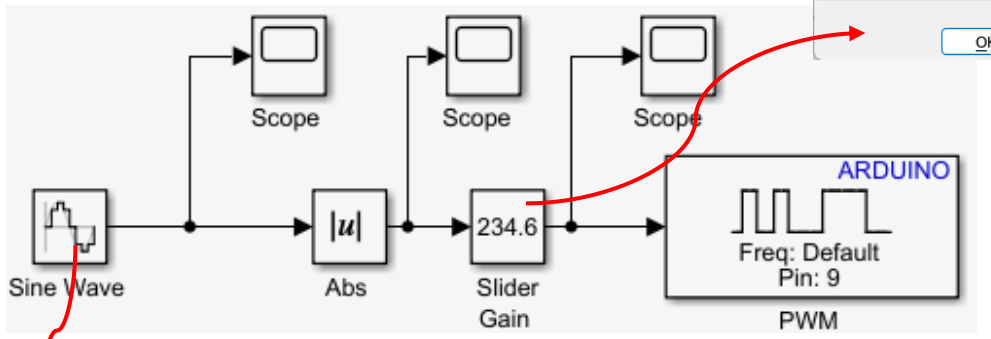


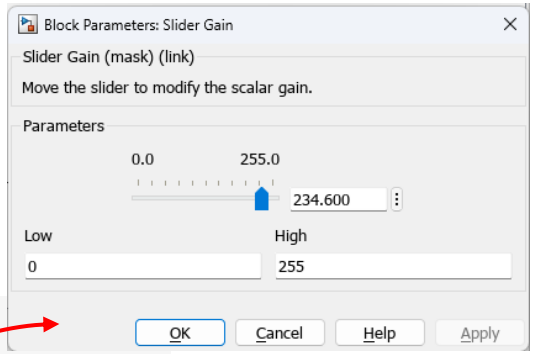
3

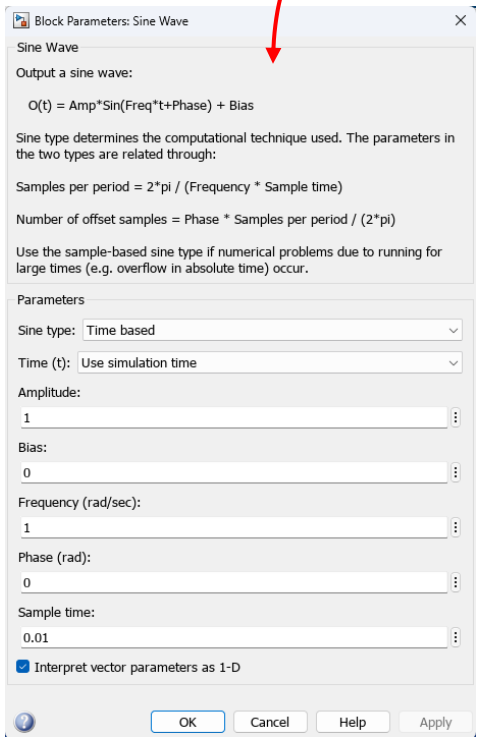


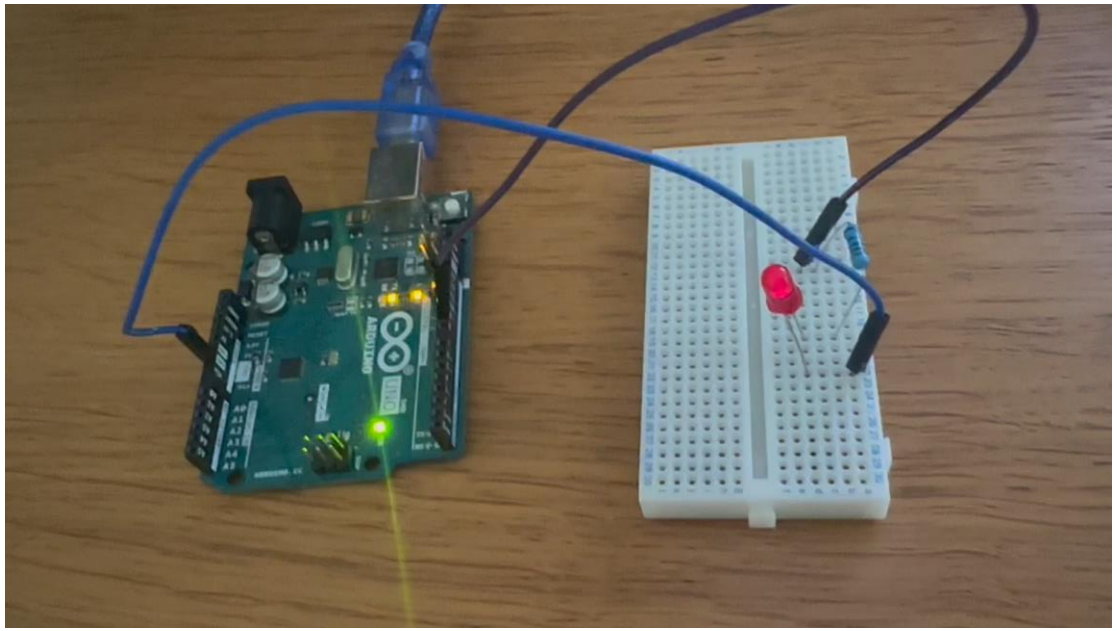
1.2.4 Algorithm Design – PWM

- A sine wave is used for the input (found in '*Sources*') and connected to an '*Abs*' (i.e., absolute) block (found in 'Math Operations')
- Connect a 'Slider Gain' block (found in 'Math Operations') and the 'PWM' block (found under 'Common') in the 'Simulink Support Package for Arduino Hardware'), with various scopes connected (found in 'Sinks')



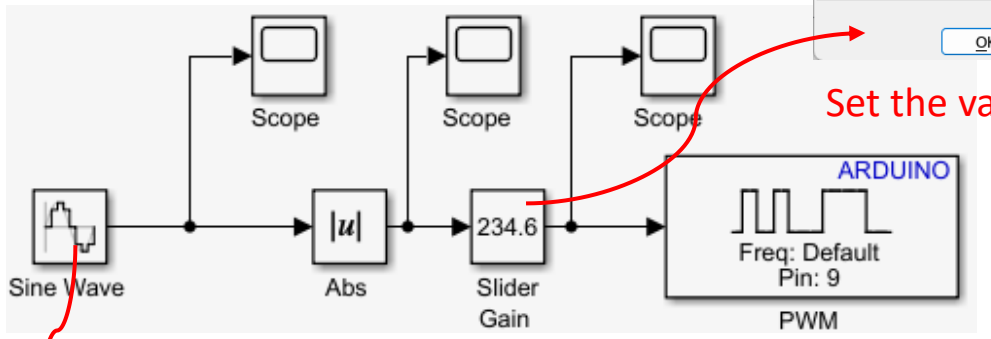


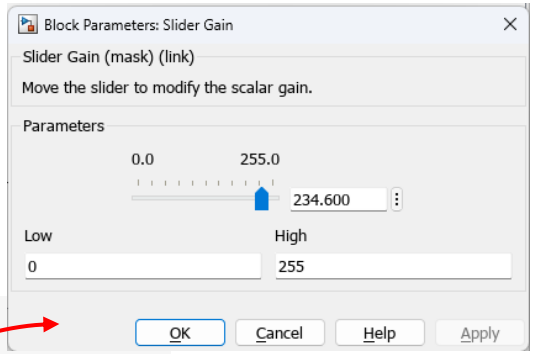




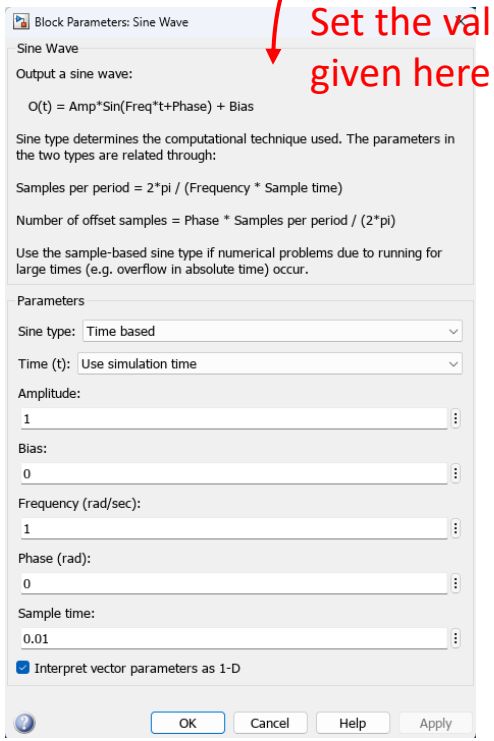
1.2.4 Algorithm Design – PWM

- A sine wave is used for the input (found in '*Sources*') and connected to an '*Abs*' (i.e., absolute) block (found in 'Math Operations')
- Connect a 'Slider Gain' block (found in 'Math Operations') and the 'PWM' block (found under 'Common') in the 'Simulink Support Package for Arduino Hardware'), with various scopes connected (found in 'Sinks')





Set the values as given here



Set the values as given here

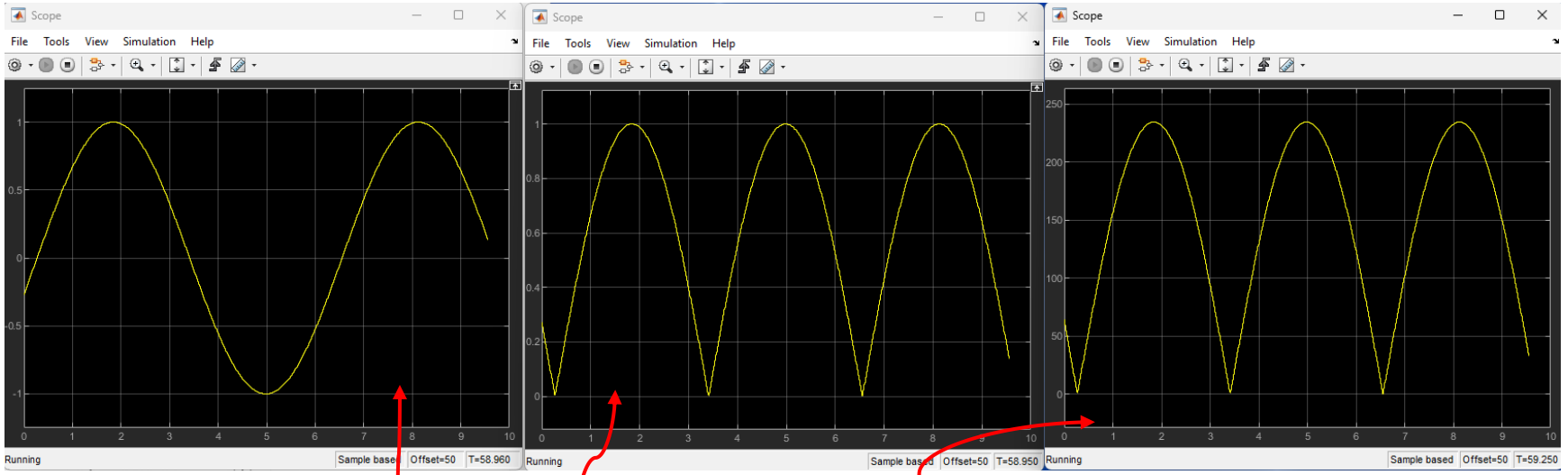
- Lower (Low) and upper (High) limits can be applied to the Slider Gain, e.g., to limit the voltage
- In this example, a limit of $2.3V$ could be applied to the Slide Gain, i.e.,

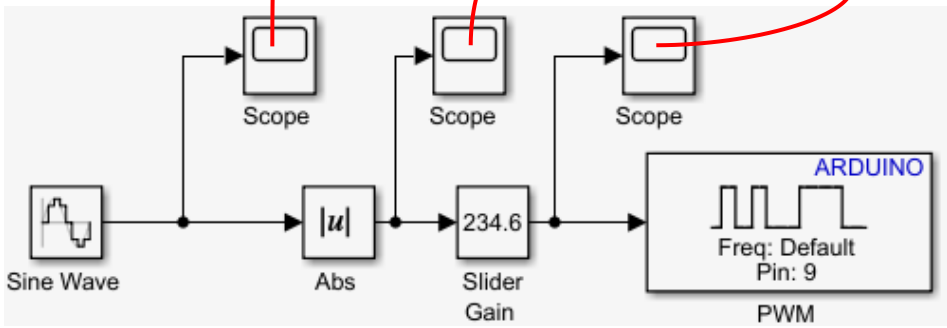
$$\left(\frac{256}{5}\right)2.3 = 117.76$$

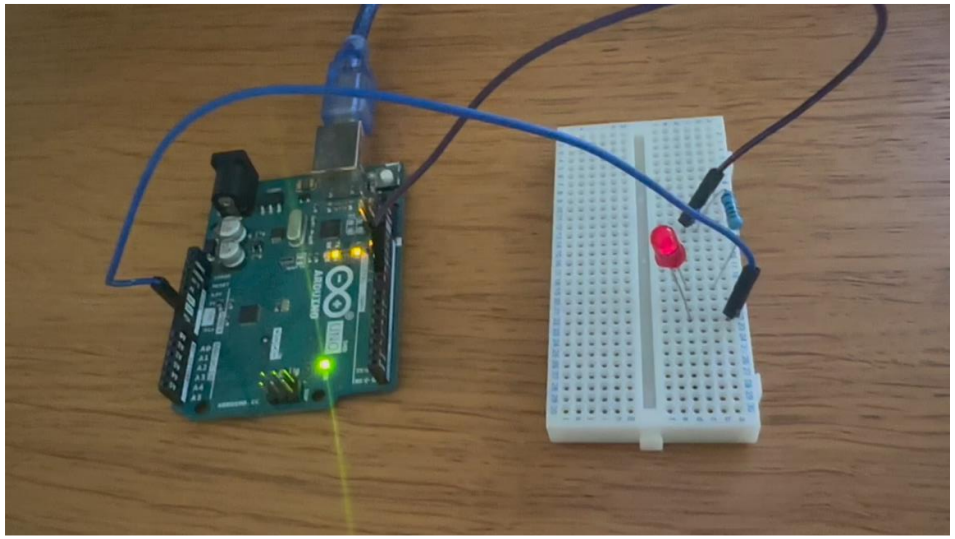
- Hence, the upper limit (High) applied to the saturation block would be 117

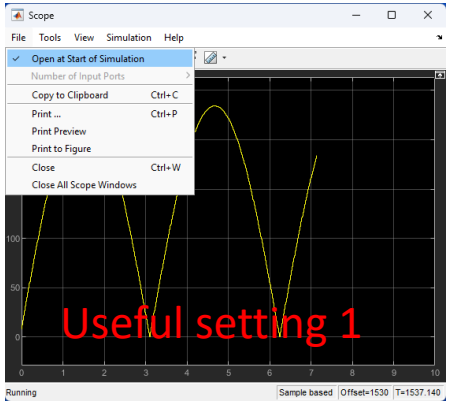
1.2.4 Algorithm Design – PWM

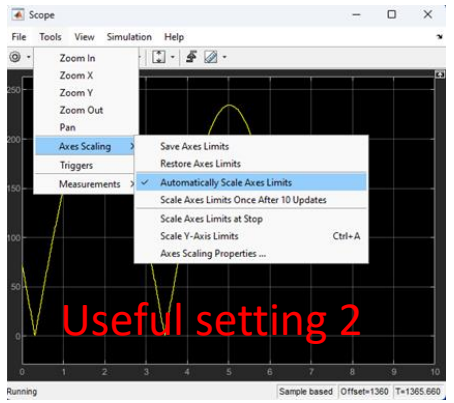
- A sine wave is used for the input (found in '*Sources*') and connected to an '*Abs*' (i.e., absolute) block (found in 'Math Operations')
- Connect a 'Slider Gain' block (found in 'Math Operations') and the 'PWM' block (found under 'Common') in the 'Simulink Support Package for Arduino Hardware'), with various scopes connected (found in 'Sinks')







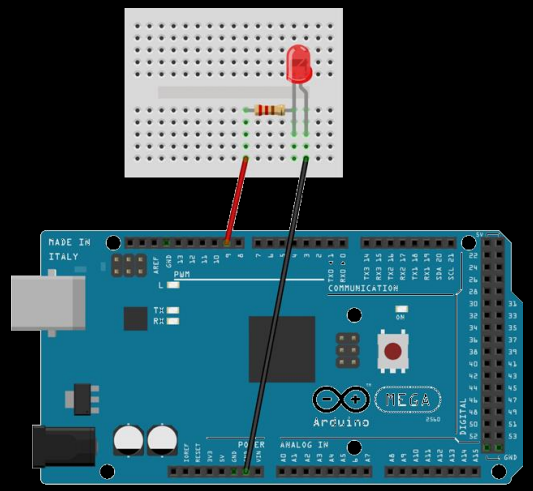




Contents

Simulink for Autocode Generation: LED Exercise

1.3 Summary



- Simulink has been used along with an Arduino Uno/Mega to develop the algorithm design for a simple circuit (resistor and LED) hardware set-up
- Initial example detailed the on/off operation of the LED
- The use of pulse width modulation (PWM) has then been applied for varying brightness of the LED
- A saturation limit has been applied within the algorithm design to limit the voltage