Experiment 11: MATLAB with Arduino
Part1: Introduction to ARDUINO/SIMULINK

Lab Objectives

- Install and verify Arduino software package for Simulink using a digital output to light a LED
- Communicate with the target board (Arduino) using external mode by changing the brightness of an LED with PWM

Part 1: Arduino Toolbox Installation for Simulink

Objective:

- Install and verify Arduino software package for Simulink using a digital output to light a LED

Simulink Arduino Library Installation:

- Open MATLAB2013a(b)/MATLAB 2014a and type "targetinstaller" in the MATLAB command window.
- The target installer window will appear. Select "Install from Internet".
- Select "Arduino (Arduino Uno/Nano/Mega 2560 for Matlab 2014a)" and click "Next"
- When prompted, log in using your MathWorks credentials. If you do not have an account you will need to create one.
- Follow the onscreen prompts to install the library.
- Once the installation processis finished, you can exit out by clicking on"Finish".

Arduino Mega 2560 Drivers' Setup:

- Connect your Arduino Mega board to the computer using the USB cable. Your computer will attempt to search for the necessary drivers online.
    - If your computer cannot find the right drivers do the following:
        - Go to Device Manager and look for your Arduino Mega board (Arduino Mega 2560). Your Arduino board will be under Ports (COM & LPT). Otherwise, it may be listed as "Unidentified Device" under 'Other Devices'.
            - If you find an unidentified device, unplug the Arduino to see this device is removed from the list to verify that this unidentified device is your Arduino board. Plug you Arduino board back in.
        - Right click on Arduino Mega (or "Unidentified Device") and select "Update Driver and Software"
        - Select "Browse my computer for driver software"
        - Set the location to :
          Matlab 2014a C:\MATLAB\SupportPackages\R2014a\arduino-1.0.5\drivers.
          Matlab 2013a C:\MATLAB\SupportPackages\R2013a\arduino-1.0\drivers.

- You will receive a prompt letting you know Windows cannot verify the publisher of the driver, select "Install this driver software anyway."
- Windows will then install the necessary drivers for the Arduino Mega from this folder. A message will appear when the driver software has installed.

o If you are using windows 8 the driver installation might have a problem because the driver file is not digitally signed. See http://mytechblog.com/tutorials/arduino/install-arduino-drivers-on-windows-8/

The Arduino drivers can also be installed from the Arduino website (www.arduino.cc).

How do I find the COM port for my Arduino Board:

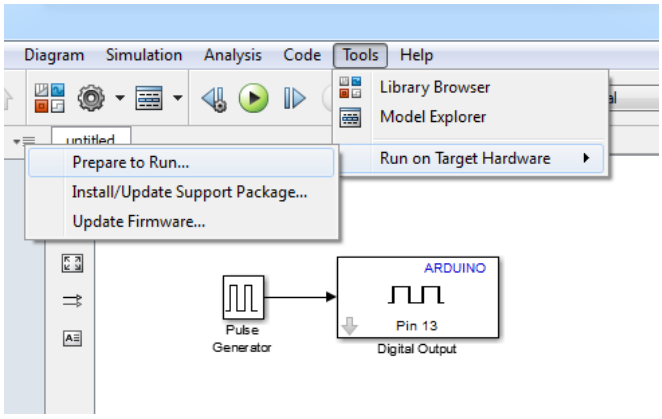
- Go to Device Manager and look for your Arduino Mega board (Arduino Mega 2560). Your Arduino board will be under Ports (COM & LPT).

Simulink Setup:

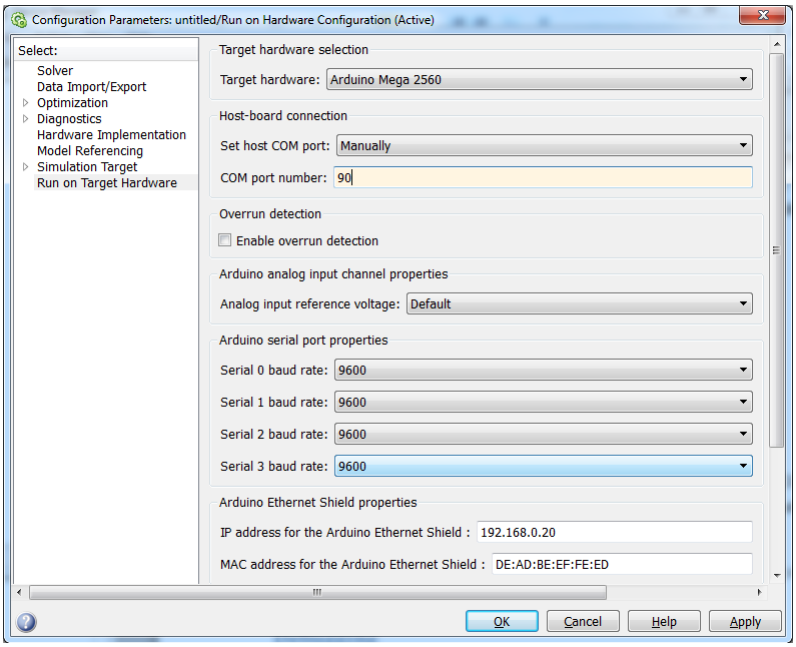
In order to ensure that your computer can properly communicate with the Arduino board, build and run the following Simulink diagram using the steps below:

1) Open MATLAB 2013a/2014a then open a new Simulink model.
2) Find and assemble the required blocks.
   - Pulse Generator is under View->Library Browser->Simulink->Sources
   - Digital Output is under View->Library Browser->Simulink Support Package for Arduino Hardware (2013b/2014a: View->Library Browser->Simulink Support Package for Arduino Hardware->Common)
3) Change the output pin to 13, which is connected to the onboard LED. This is done by double clicking on the Digital Output block.
4) Double click on the Pulse Generator to change the amplitude to 1, the Period to 10, and the pulse width to 50%.

5) Under the tools menu go to ‘Run on Target Hardware’ and click ‘Prepare to Run’. In the window that pops up you will need to select Arduino Mega 2560 in the drop down for Target Hardware.

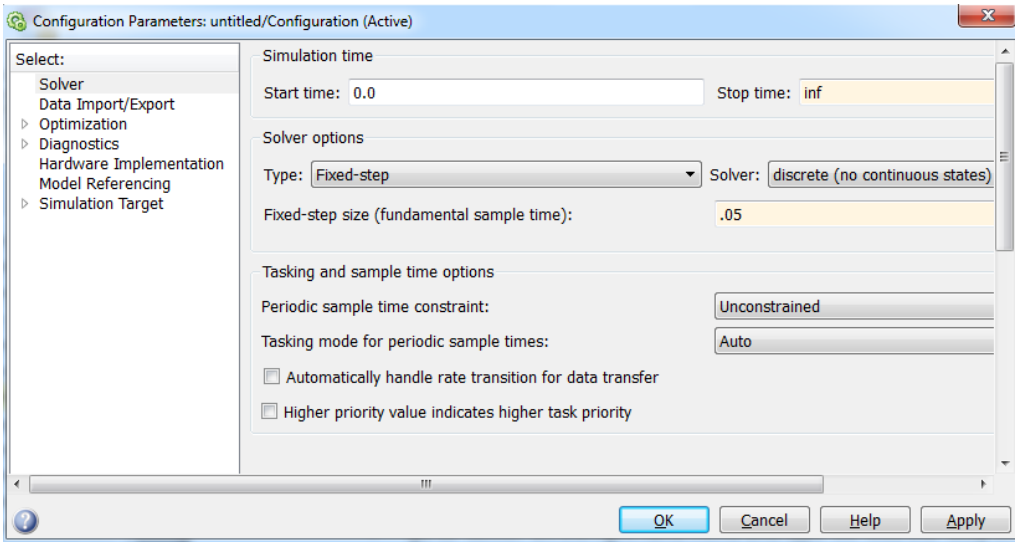


- “Set host COM port: “ should be set to “Manually” and specify the COM port your board is connected to, clink “Apply” and “Ok”. Automatically does not work reliably on some computers.

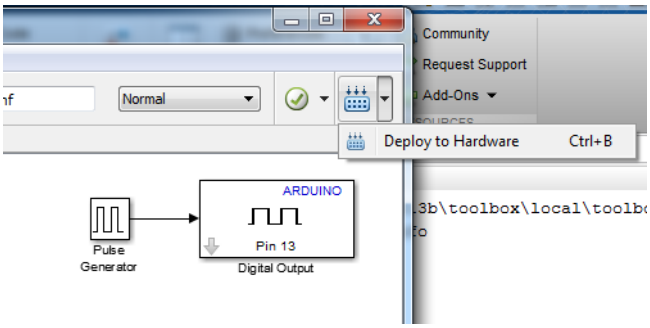


- Right click on the background and select ‘Model Configuration Parameters’, or  .
- go to ‘Solver’ on the right hand menu and type ‘inf’ into the ‘Stop time’ spot so that the model will keep running. Make sure the solver options are set to “Fixed-step” and “discrete (no continuous states)”. The “Fixed-step size” should be 0.05. This is how fast in seconds the

control loop will execute. In this case every 50 milliseconds it will execute the Simulink code.



6) Compile and download the code to the board:
   - **2013a:**Go to 'Tools-Run on Target Hardware-Run'.
   - **2013b/2014a:** Clink "Deploy to Hardware Button" or the keystroke Ctrl+B:



At this point the on board LED connected to pin 13 should start blinking on and off. If not, check the debugging section below.

Debugging

- If an error occurs indicating that there is no driver, follow the directions on the screen.
- If MATLAB cannot find the board find COM port number from the device manager and in 'Configuration Parameters' set the port manually to the correct number.
- Some errors appear on the MATLAB workspace screen - if something is not working, check here.

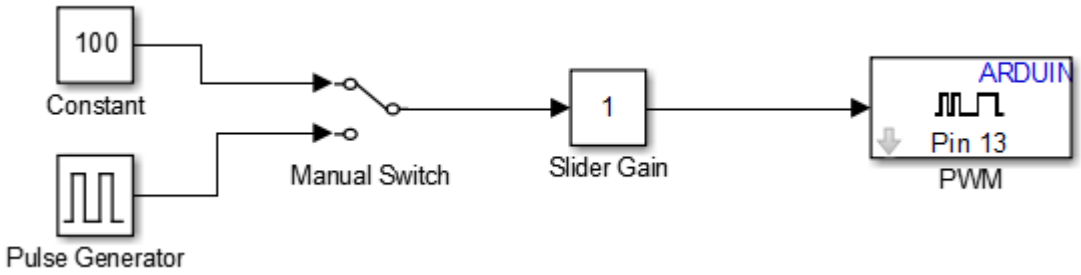
Part 2: Communicating with External Mode

Objective

- Communicate with the target board (Arduino) using external mode by changing the brightness of an LED with PWM

Simulink

Modify your Simulink diagram to the following



- Pin 13 can also be set as a PWM instead of just an on/off digital output – replace the Digital Output block with the PWM block
- The "Slider Gain" can be found in View->Library Browser->Simulink-> Math Operations. Change the high value to 255 (the max value of the PWM)
- The "Manual Switch" can be found in View->Library Browser->Simulink ->Signal Routing
- The "Constant" can be found in View->Library Browser->Simulink ->"Commonly Used Blocks"

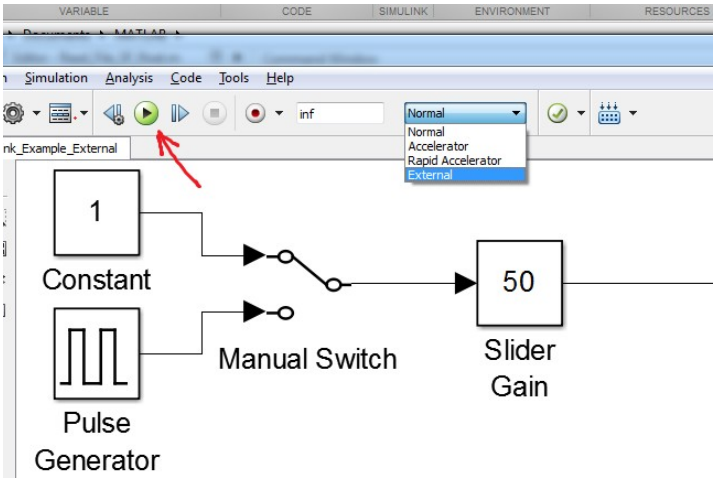
External Mode

- External mode allows bi-directional communication to/from the application board to the PC.
- When external mode is selected it downloads additional code to the board to allow this communication so this does increase the program size.
- When you use external mode you can change parameters while the system is running
    - However this has an impact on the performance:Due to the communication bandwidth, the fastest you can run in this mode and still meet your control loop time is approximately 30 milliseconds
- The values change on the Slider Gain and the Manual Switch can be changed while the program is running.
- The code can be run much faster (in the orders of1 or 2 milliseconds) if external mode is not used.

Enabling External Mode:

- **2013a:**"Configuration Parameters" under the "Run on Target Hardware" tab make sure to enable external mode is enabled.

- **2013b/2014a:** Select “External” from the drop down menu, then press the “Play” button



Run the code and experiment with the setup. Observe the effects on the Arduino LED when changing the value of the Slider Gain, Manual Switch, and the Pulse Generator.

- While running the simulation, you can change the position of the switch by double clicking on the switch
- You can change the gain of the Slider Gain while running the simulation by double clicking it.
- Adjusting the Pulse Generator, experiment with all three parameters. Which increases brightness? What is the approximate range over which you can observe a difference? [Note: These parameters will be addressed in a later lab on PWM]

Stopping the Simulation:

When you are running the simulation in external mode you are connected to the device and are sending data and receiving data. When this is occurring do NOT use the stop button to stop the simulation:

- 2013a:
  - Always use the “Connect to Target” button to connect to the board. If you use the “Run on Target Hardware” menu then it will automatically download your code, connect to the board and run your code.

  - Use “Disconnect from Target” if to make changes to the code or to unplug the USB cable. Do not unplug the USB cable before disconnecting from the target. Doing so will leave the serial port open on the computer and you will not be able to connect to the

device without restarting Matlab or logging off your machine (to ensure the serial port gets closed).

- 2013b/2014a:
    - Use the Play button to download and run the code in external mode
    - Use the square Stop button to stop

Debugging – cannot connect to device or cannot download code

- If you cannot connect to the device or there is an error when trying to download the code usually this means external mode was enabled and the simulation was stopped with the stop button or the USB cable was unplugged while being connected to the device. In this case you have a couple things to try that may successfully close the serial port
    - Try the following commands at the Matlab command line (find connected instruments and close them):
        - newobjs=instrfindall
        - fclose(newobjs)
        - delete(newobjs)
    - Reset the device with the reset button or disconnect/connect the serial cable (this sometimes works)
    - Close and restart Matlab (this usually works)
    - Log off your machine, log back in, then restart Matlab (this almost always works)
- Make sure the COM port is correct in the "Model Configuration Parameters"

Checkpoint:

To complete Lab 1 you will need to show the ability to control the blink of the LED in one program with both the 'manual switch' and the 'slider gain'.

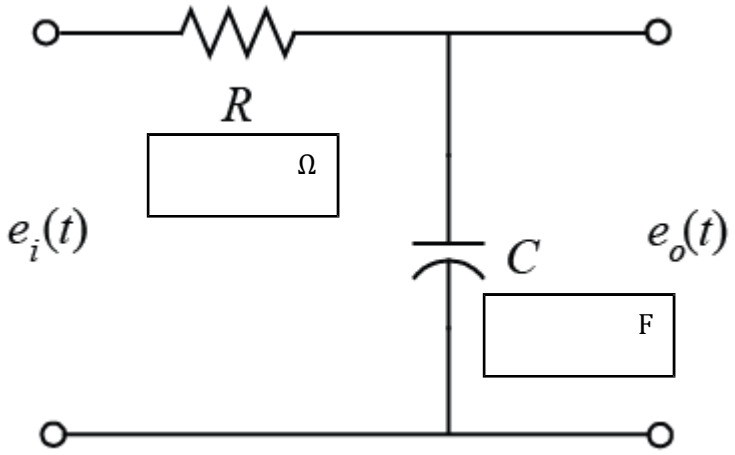
Part2: RC Circuit / First Order System

The purpose of this part is to demonstrate how to model a simple electrical system. Specifically, a first principles approach based on the underlying physics of the circuit and a black-box approach based on recorded data will be employed. The associated experiment is employed to demonstrate the black-box approach, as well as to demonstrate the accuracy of the resulting models. This activity also provides a physical example of the common class of first-order systems.

I. Mathematical Modeling of System

First we will employ our understanding of the underlying physics of the RC circuit to derive the structure of the system model. We will term this process "modeling from first principles.

1) Write the value of resistor and Capacitor for the circuit



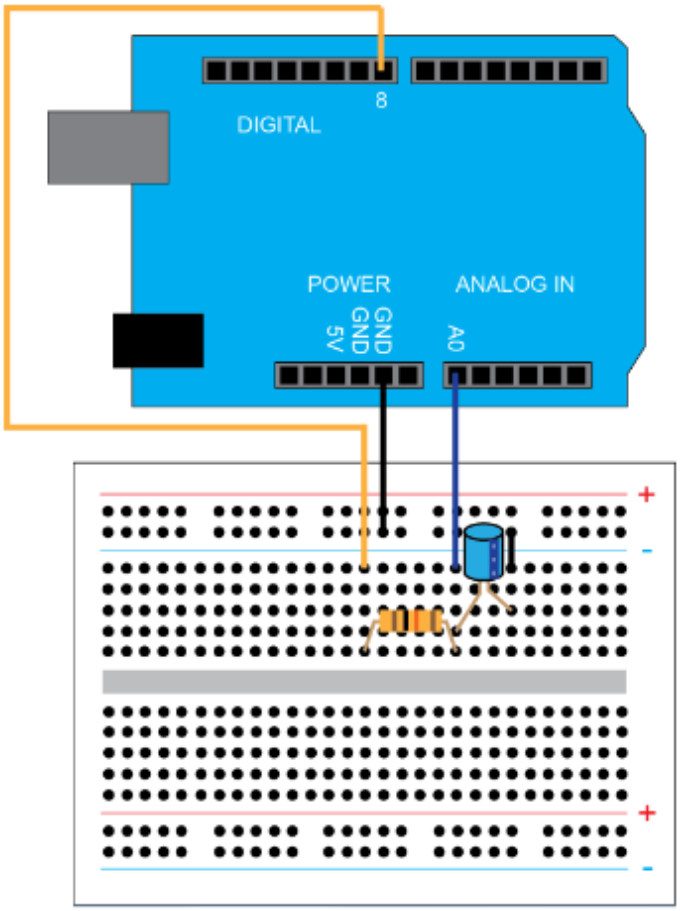
2) Find the transfer function $G(s) = \frac{E_o(s)}{E_i(s)}$, and determine the value of gain and time constant

II. System Identification of System (Open Loop Response)

In this part we will record the output voltage of the RC circuit for a step in input voltage. Based on the resulting time response of the output voltage, we will fit a model to the data. This approach is sometimes referred to as black-box modeling or data-driven modeling. After we have generated such a model, we will compare it to the first-principles derived model we created previously.

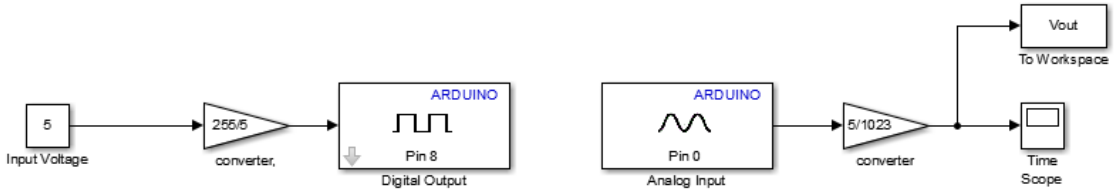
1) Hardware setup

The Arduino board is employed to receive the input command from Simulink and to apply **the input voltage to the circuit (via a Digital Output**). The board also acquires the **output voltage data from the circuit (via an Analog Input)** and communicates the data to Simulink.



2) **Software Setup**

We will employ Simulink to read the data from the board and to plot the data in real time. In particular, we will employ the Simulink Support Package for Arduino Hardware from the MathWorks. The Simulink model we will use is shown below



The Arduino Analog Read block reads the output voltage data via the Analog Input A0 on the board. Double-clicking on the block allows us to set the Pin to 0 from the drop-down menu. We also will set the Sample Time to "0.1". This is 10 times faster than the circuit's time constant and hence is sufficiently fast. The other blocks in the model can also be set to have a Sample Time of "0.1" (or left as "-1"). In the downloadable model, the sample time is set to the variable Ts which needs to be defined in the MATLAB workspace by typing Ts = 0.1 before the model can be run. The Gain block on the Analog Input is included to convert the data into units of Volts (by multiplying the data by 5/1023). This conversion can be understood by recognizing that the Arduino Board employs a 10-bit analog-to-digital converter, which means (for the default) that an Analog Input channel reads a voltage between 0 and 5 V and slices that range into $2^{10} = 1024$ pieces. Therefore, 0 corresponds to 0 V and 1023 corresponds to 5 V.

3) **Parameter identification**

$$\frac{ك}{1}$$

1) Determine the value of parameter

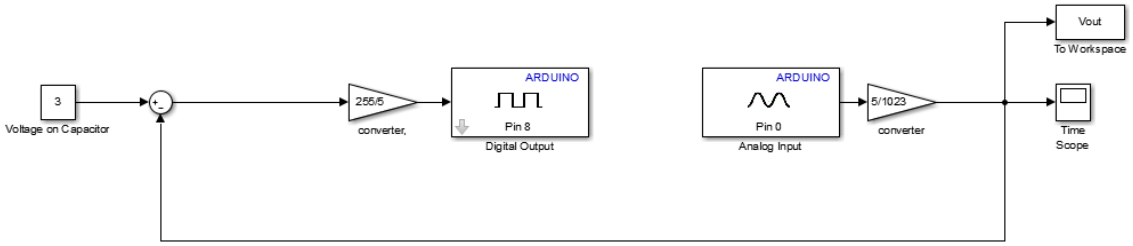
Parameters	Value
Gain (K)	
Time Constant (	

2) Write the Estimated transfer function.

3) Compare the value with the value in section 2 of Part I, write your comment??

III. Closed Loop Response

The following figure shows the Simulink block to read the closed loop response of RC circuit



1) Determine the step specifications of closed loop system

Step Specifications	Value
Overshoot (%)	
Settling time (sec)	
Steady state value of output ب00 (V)	
Steady State Error (◌00	

IV. Design PI Controller

Build discrete PI controller using the estimated model and test the controller experimently

Controller Parameter	
ك	
ك	
Closed Loop Specifications after the addition of PI controller	
Step Specifications	**Value**
Overshoot (%)	
Settling time (sec)	
Steady state value of output ب00 (V)	
Steady State Error (◌00	

V. Design PID Controller

Build discrete PID controller using the estimated model and test the controller experimently

Controller Parameter	
ค	
ค	
ค๓	
Closed Loop Specifications after the addition of PID controller	
Step Specifications	**Value**
Overshoot (%)	
Settling time (sec)	
Steady state value of output ๒๐ (V)	
Steady State Error (๐	

**** Note: Print Screen All Figures**

1) The Experimental Response with Simulation Response of **open loop system**
2) The Experimental Response with Simulation Response of **closed loop system**
3) The Experimental Response with Simulation Response of **PI controller**
4) The Experimental Response with Simulation Response of **PID controller**