Lab 1: Introduction to MATLAB Simulink and the ArduinoIO block-set

1 BACKGROUND

In Simulink, it is very straightforward to represent and then simulate a mathematical model representing a physical system. Models are represented graphically in Simulink as block diagrams. A wide array of blocks are available to the user in provided libraries for representing various phenomena and models in a range of formats. One of the primary advantages of employing Simulink (and simulation in general) for the analysis of dynamic systems is that it allows us to quickly analyze the response of complicated systems that may be difficult to analyze analytically.

In this lab you will become familiar with the Simulink graphical environment and block library. You will gain some experience with commonly used blocks as well as the add-on blocks from the ArduinoIO library.

Using the ArduinoIO library with MATLAB and Simulink, you will learn to collect analog data from an Arduino over the serial link. This data can then be made available in the MATLAB workspace for analysis, plotting and calculations.

2 EXPERIMENT

In this introductory lab, you will practice collecting analog voltage readings from the Arduino in tethered mode. The analog voltage will be generated with a simple potentiometer circuit.

2.1 SETTING UP YOUR EXPERIMENT ENVIRONMENT | ARDUINO

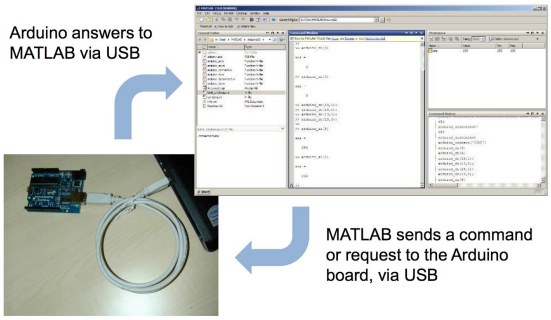
1. Identify which serial port the Arduino is using to communicate

2. Using the Arduino IDE, upload the ArduinoIO motor_v2.pde sketch to the Arduino, this sketch provides functionality for analog and digital IO as well as encoder, servo and motor shield support.

3. The ArduinIO support package should already be installed on the lab PC's so now you are ready to start MATLAB and communicate with the Arduino

2.2 Interfacing the Arduino with MATLAB

Figure 2.1: Matlab acts like a host to the tethered Arduino



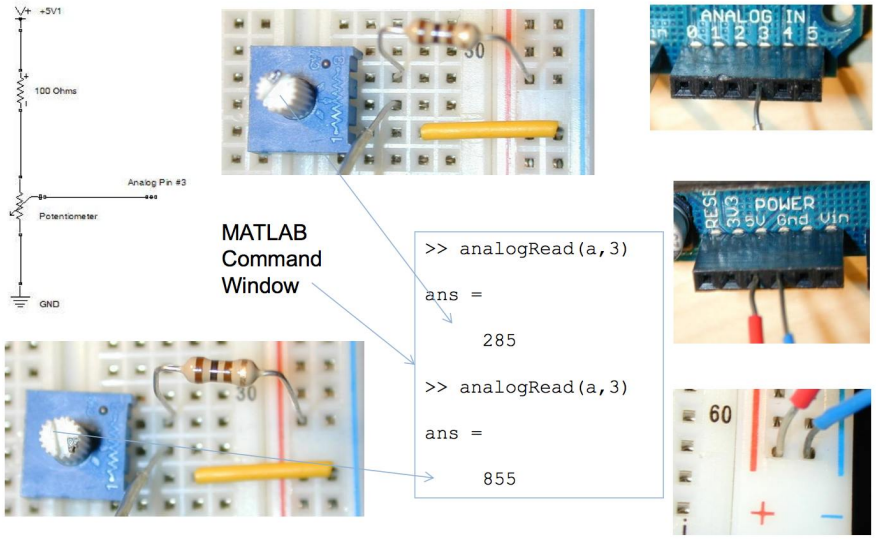
1. Use the command `a=arduino('port')`, with the right COM port as a string input argument, to connect MATLAB with the board and create an Arduino object in the workspace.
   `Example:`
   ```
   » a=arduino('COM5');
   » a=arduino('/dev/ttyS101');
   ```

2. Use the command `pinMode(a,pin,str)` to get or set the mode of a specified pin:
   `Example:`
   ```
   » pinMode(a,11,'output')
   » pinMode(a,10,'input')
   » a.pinMode(10,'input')
   » val=pinMode(a,10)
   » pinMode(a,5)
   » pinMode(a)
   ```

3. Use the command `digitalWrite(a,pin,val)` with the pin as first argument and the value (0 or 1) as second argument:
   `Examples:`

```
» digitalWrite(a,13,1); - sets pin #13 high
» a.digitalWrite(13,1); - sets pin #13 high
» digitalWrite(a,13,0); - sets pin #13 low
```

Figure 2.2: Using a potentiometer

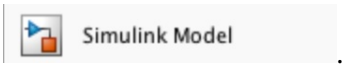


4. Use the command `val=analogRead(a,pin)` with the pin as an integer argument. The returned argument ranges from 0 to 1023
   `Examples:`
   ```
   » val=analogRead(a,0); - reads analog pin #0
   » val=a.analogRead(5); - reads analog pin #5
   ```

5. Use the command `delete(a)` to disconnect the MATLAB session from the Arduino board. This deletes the connection to the serial port and frees it for other applications.
   `Examples:`
   ```
   » delete(a);
   » a.delete;
   ```

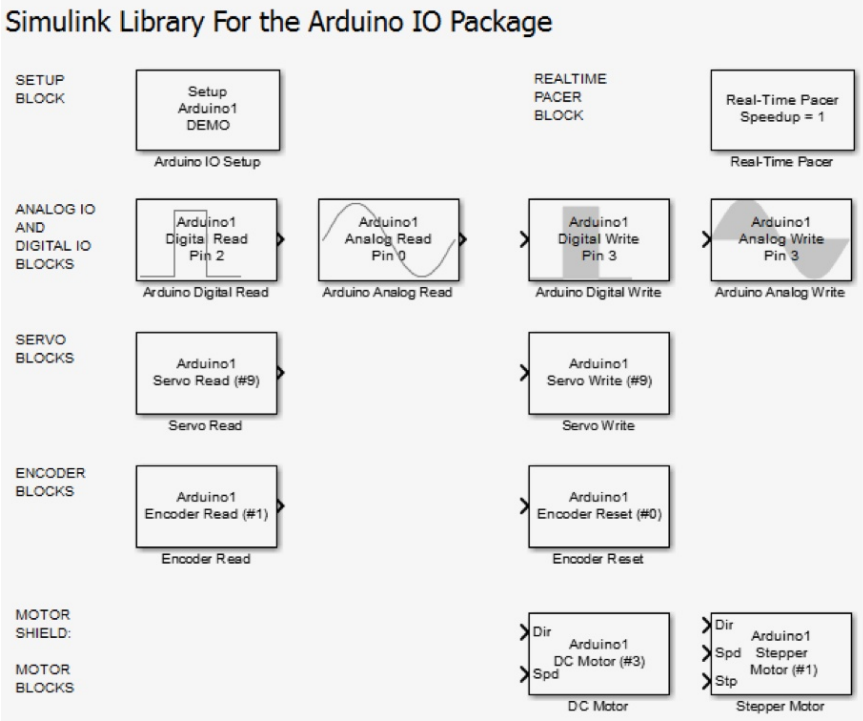
2.3 Setting up experiment environment | Simulink

1. In the MATLAB command window, type `simulink` or press new 

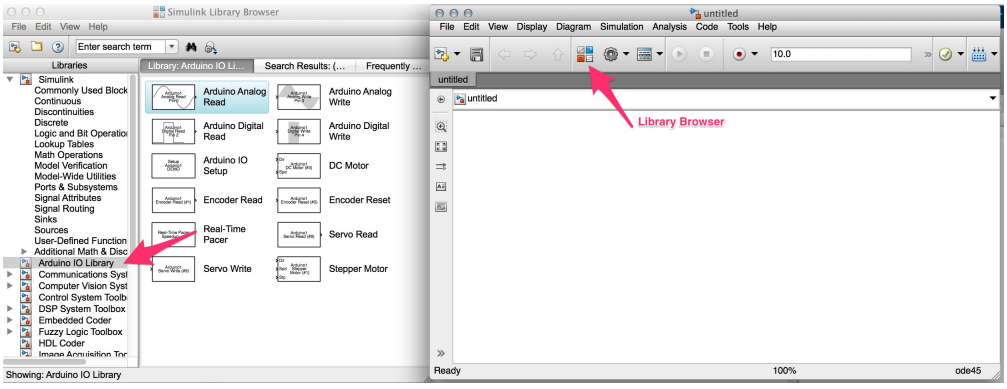
, followed by Simulink Model 

.

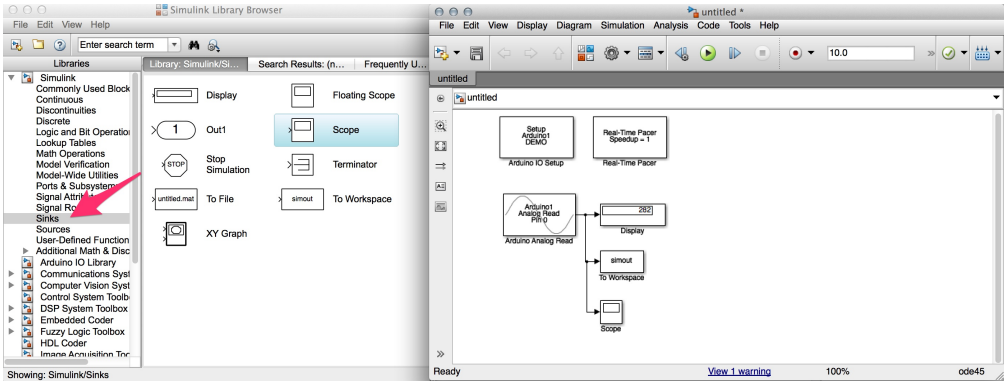
Figure 2.3: The ArduinoIO Simulink library



2. Once the Simulink design window has opened, open the Library Browser to find the ArduinoIO Library



3. You can drag and drop blocks from the Library Browser to the diagram. Create the following:



Here, the two free floating blocks help set up the environment. The "Setup" block establishes parameters for communicating with Arduino while the "Real-Time Pacer" block ensures that Simulink runs in real time (seconds).

The "Analog Read" block captures data from Arduino's pin 0 and sends it to three blocks from the "Sinks" Library in Simulink. These blocks display the data in a simple graph, numerical display and save it the workspace as a variable "simout". The parameters of each of these blocks are configurable by double-clicking the blocks.

4. Use the potentiometer circuit to generate some variable analog signals. Capture them in the MATLAB workspace.

3 DELIVERABLES

For this first experiment, use both Simulink and Matlab script to capture analog voltage from the potentiometer circuit.

1. While turning the potentiometer clockwise from lock, capture analog voltages in Volts (V) for the following conditions:
   a) 5 Seconds at 1 Hz
   b) 5 Seconds at 10 Hz
   c) 5 Seconds at 100 Hz
2. Plot each dataset individually and on the same plot, maintaining time scale.
3. In your report, describe the process to interface the Arduino with MATLAB using the ArduinoIO library, include both the MATLAB .m script and a screenshot of your Simulink model to complete the above task.