MAE/ECE-5320 LAB 05: SERVO MOTOR CONTROL

Checklist for Lab05:

a) **A computer installed with MATLAB compatible with Arduino.**
b) **Uno or others**
c) **Jumper wires**
d) **Servo Motor in Arduino starter kit, other servo motors will also work.**
e) **Up to 2 members per group**

In the previous lab, we had experience with Simulink IO to control the hardware. The Simulink IO is an intermediate step between real-time embedded control (standalone on Arduino) and model-based simulation in Simulink.
Knowing that, in this lab, we will use a very simple example of servo control to demonstrate different approaches to interact between MATLAB and Arduino.

Section 1. Servo motor control by MATLAB script

This section is modified from this link.
https://www.mathworks.com/help/matlab/supportpkg/control-servo-motors.html

(1) Connect Arduino with servo motor.
Find out the wires of servo and connect properly to Arduino. For the servo motor in Arduino starter kit, connect power wire (red) to 5V, ground wire (black) to Arduino ground (GND) and signal wire (white) to pin 4.

(2) Set up Arduino with proper COM port number

```
a = arduino
% OR
a = arduino('COM7', 'Uno', 'Libraries', 'Servo');
% setup Arduino and program servo library into Arduino board
```

Note: if the port number is not known, then we can use ***arduinosetup*** as we did in last labs

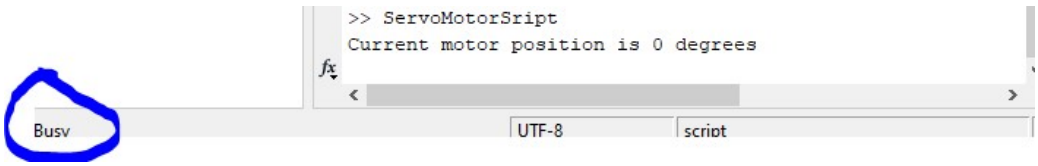
(3) Create a matlab script file and type in the following code to create a servo project

```
s = servo(a, 'D4')
```

(4) Type in or copy/paste codes for pulse signal and position reading.

```
for angle = 0:0.2:1
    writePosition(s, angle);
    current_pos = readPosition(s);
    current_pos = current_pos*180;
    fprintf('Current motor position is %d degrees\n', current_pos);
    pause(2);
end
```

The rotation of servo motor will be controlled by executing the MATLAB script. While the script is running, matlab shows its status as 'busy', that indicates interactive data communication between MATLAB and Arduino. This program is not standalone--running on the Arduino without needing to be connected to MATLAB



Exercise 1: Change servo motor resolution (step size)

The shaft position is mapped from 0-180 degree to variable angle range 0-1 values. What is the angle (degree) represented by angle step size 0.05 ?

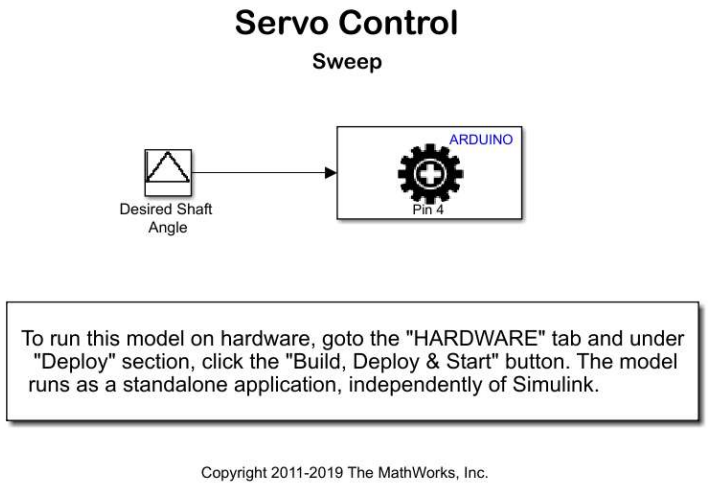
Modify the code as

```
angle = 0:0.05:1
```

Pass off with the TA or embed a movie (5-10 seconds) into your submission form showing a clip of your motor moving with your Command Window output running in the background showing the motor position sequences.

Section 2. Servo motor control by Simulink IO. (external mode)

Step 1: type *arduino_servocontrol_sweep* in MATLAB command to call the example below (if you receive an error message, click the error message's link).

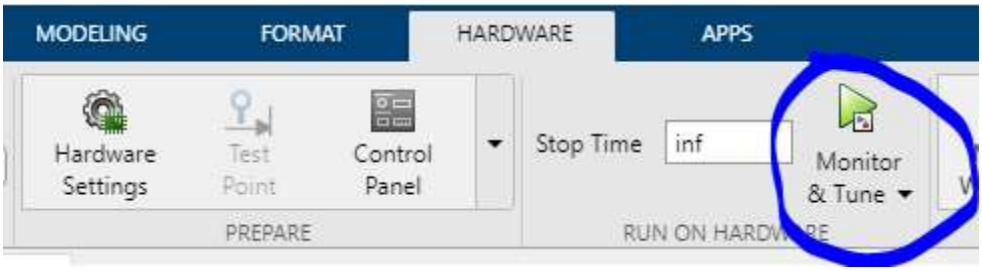


Step 2: setup hardware configuration

Select hardware based on your board. Go to model setting/solver, set ***'Fixed-step'*** for solver, and set fixed step size to ***0.001***. change stop time to inf. If using an older MATLAB version, in Hardware implementation/ target hardware setting, check ***enable connected IO***.

Step 3: add a scope to the Desired Shaft Angle (repeating sequence stair signal) block. In View => Configuration Properties = > Time, set "Time span overrun action" to "Scroll"

Step 4: click on Hardware/Monitor&tune



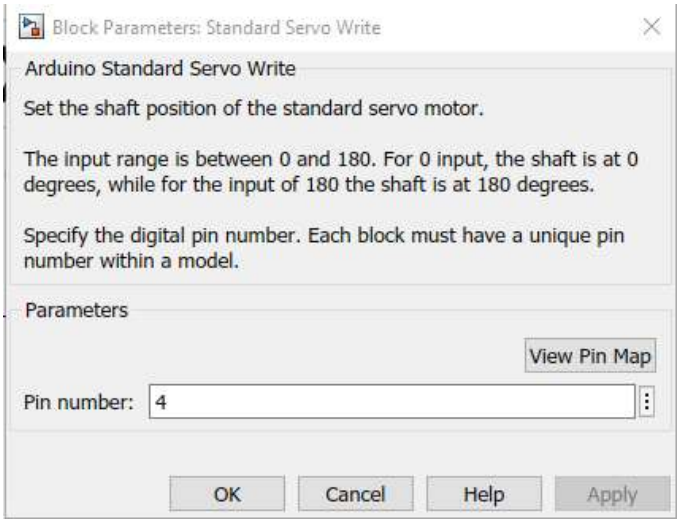
After building the model and establish connections between MATLAB and Arduino, then data communication is established.

This function enables us to monitor and tune the parameters before we permanently burn programs to Arduino board.

Use the scope's "Scale Y-axis Limits" to view the full signal amplitude.

Exercise 2: generating a repeated sequence stairs

The servo write block takes values ranging from 0 to 180.

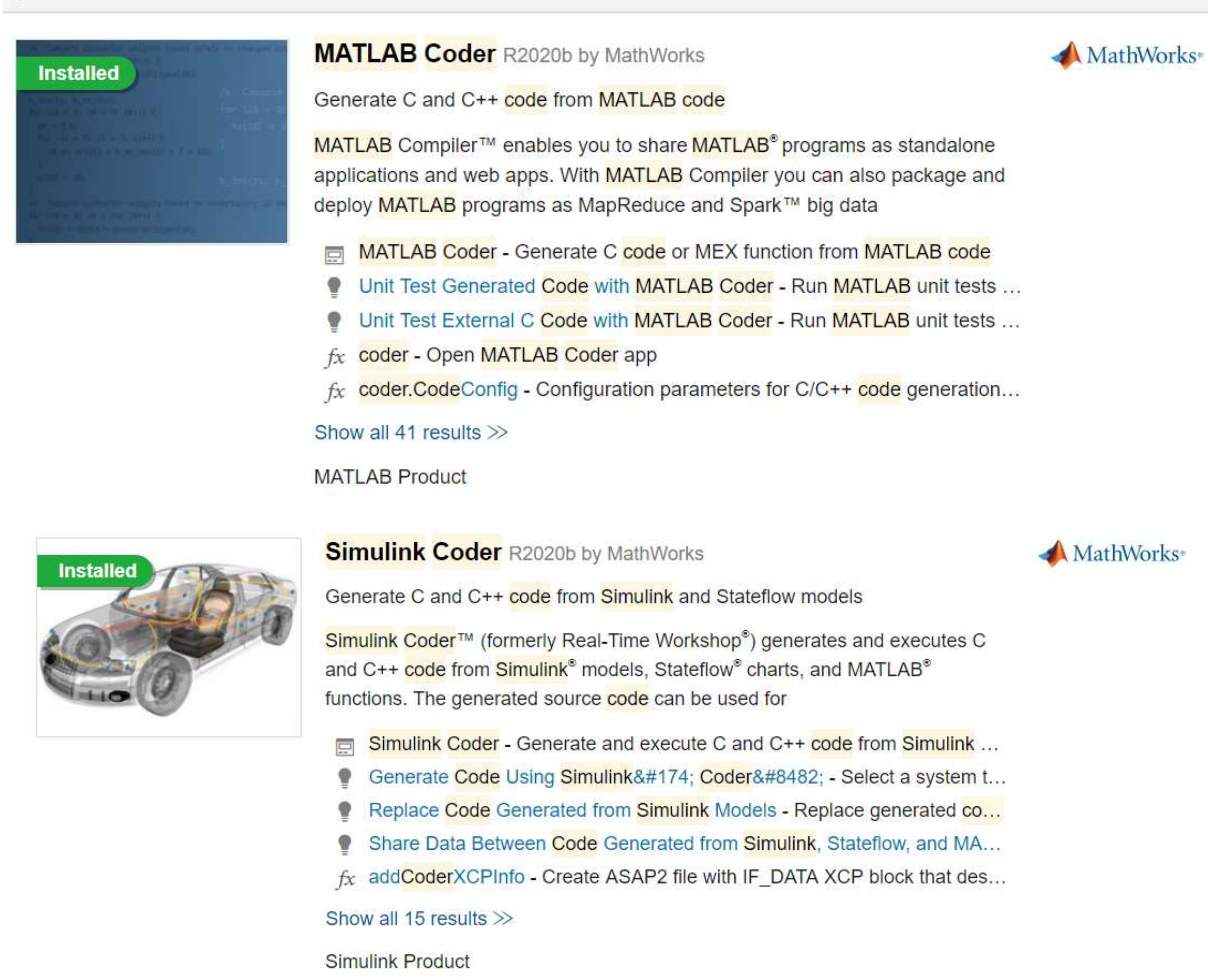


If we want the servo to rotate by following a sinusoidal function $\theta = 180^{\circ} abs(sin(t))$. What vectors of output values should be put in the Desired Shaft Angle block? To answer this question, attach a screenshot of the 'desired shaft angle' block and pass off with the TA (or embed a 5-10 sec movie) of your motor rotating with the scope block scrolling in the background.

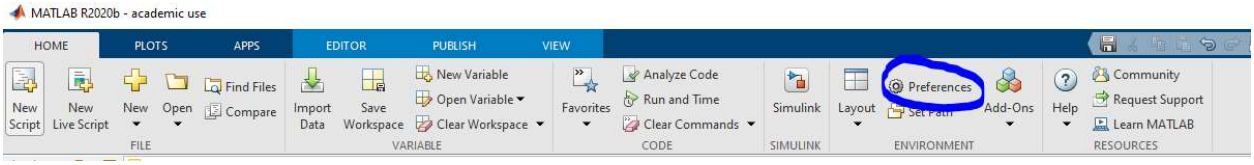
Section 3. Servo motor control by embedded codes generated from Simulink.

In the lecture, we discussed that SIMULINK can automatically generate executable and compliable codes for Arduino. This section will show the process.

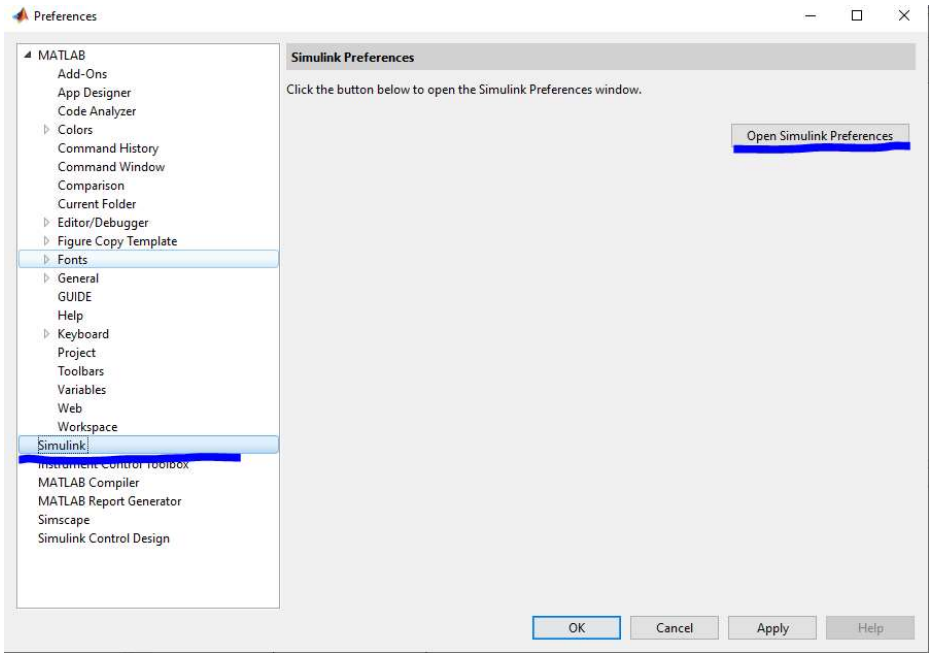
We need to install the MATLAB coder and Simulink coder packages in add-ons, so that Simulink and MATLAB will generate coding from the Simulink model.



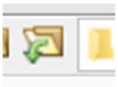
The first step is to define the folder for code generating. Click on ***Home/preference***

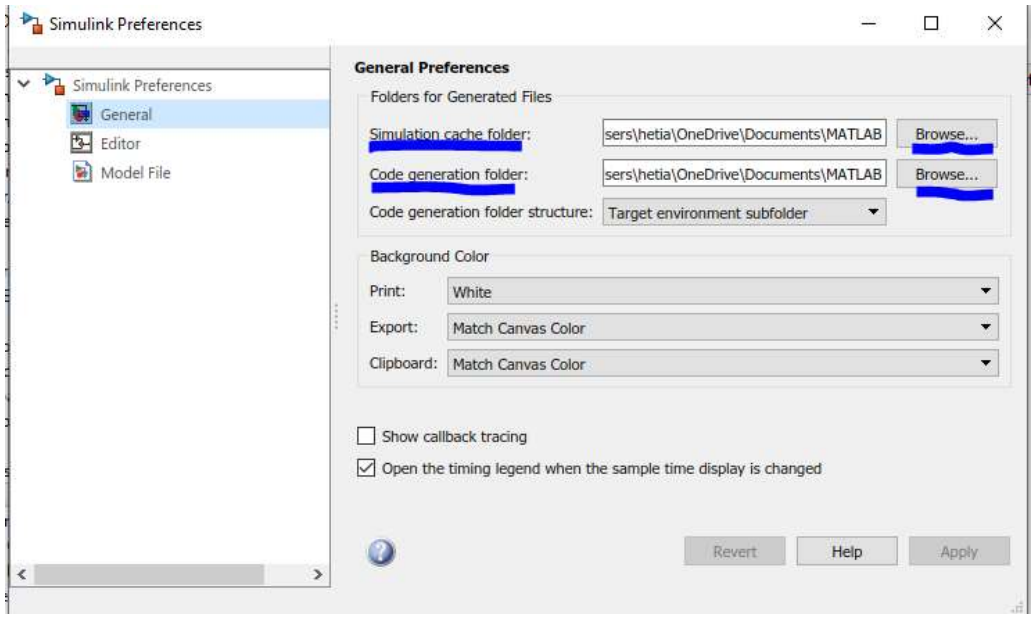


Find Simulink and open Simulink preference



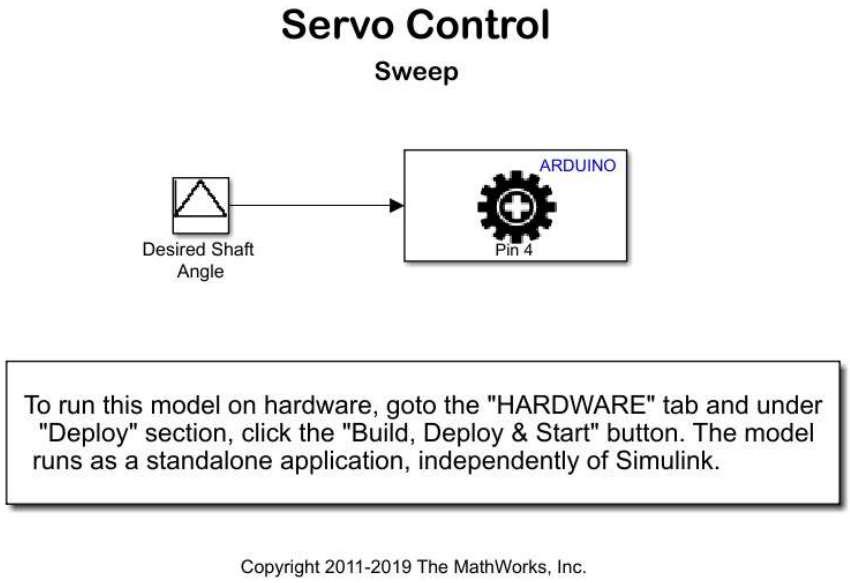
Next, we need to browse the folder path and select proper folder for simulation cache and code generation.

Select the path *C:\Users\username\OneDrive\Documents\MATLAB* for both folder (Alternative: Put the path locations in a new folder on your desktop. Set MATLAB's "current folder" as that desktop folder's location using  . Save your new "*arduino_servocontrol_sweep*" in step 1 below in this desktop folder too).



Now, we are ready to create Simulink model and then build Simulink to executable code and deploy to Arduino.

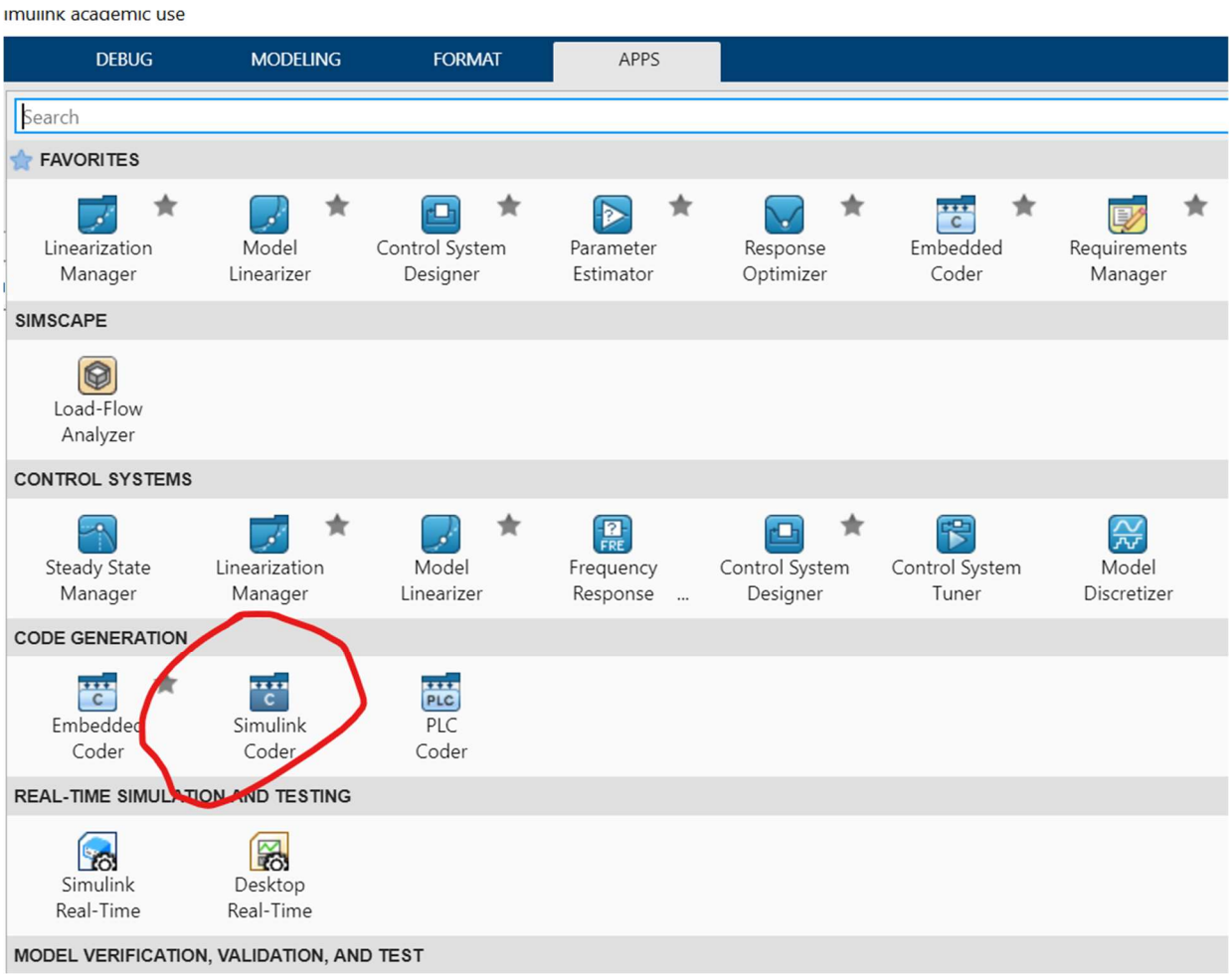
Step1: type *arduino_servocontrol_sweep* in MATLAB command to call the example model. You will see this model popped up. Change pin number in servo write block to the pin number connected with signal wire.



Step 2: setup hardware configuration

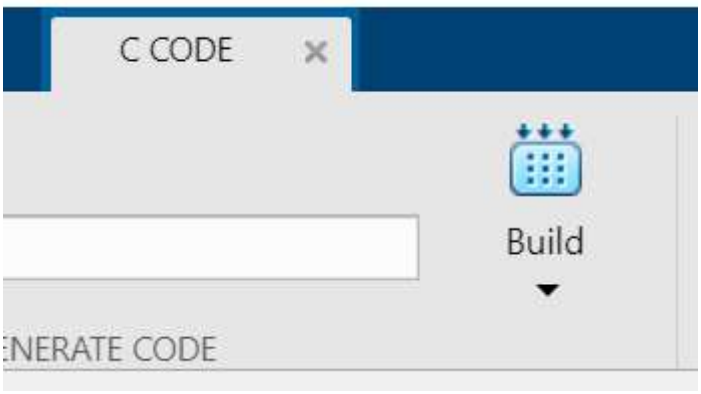
Go to model setting/solver, set ***'Fixed-step'*** for solver, and set fixed step size to ***0.001***. change stop time to inf.

Step 3: Go to Apps and select Simulink coder generation. This creates an additional "C CODE" tab next to the "APPS" tab.



Step 4: Go to the C CODE tab's setting-> hardware implementation->advanced parameters and make sure "Use Simulink coder features" is checked (To get to "Advanced Parameters", hover over the "…" that is below the "Hardware board settings" & "Target hardware resources" section).

Step 5: Click on Build in the C CODE tab, and then your Simulink model will be built and C code will be generated under the folder.



After building model and generating code, you will see a code generation report and servo will start to rotate as pulse generation block.

Look at the left-bottom corner, the status of MATLAB is 'ready', that means MATLAB is now not involved in executing programs. Arduino board itself is running embedded programs (you can't stop the motor by clicking a stop button on Simulink anymore—it runs all on its own).

Simulink generates C code instead of directly generating code for Arduino to run. But there are some ways to generate Arduino IDE codes (pde, ino)

https://www.mathworks.com/matlabcentral/answers/339523-compile-simulink-generated-code-in-arduino-ide

Exercise 3: code generation

Go to the folder path:

C:\Users\username\OneDrive\Documents\MATLAB\AtmelAVR\arduino_servocontrol_sweep (or the desktop folder if you set up Section 3 with a desktop folder)

Find out C-file named ***arduino_servocontrol_sweep.c***

Copy and Paste the C-code here.