DC Motor Step Response

Part 1: Motor Step Response

Objectives:

- Use the DRV8833 motor driver and obtain the motor step response
  - In External mode
  - Directly through the serial port
- Understand the logic needed to control the magnitude and direction of a motor
- Observe the different effects of different logic schemes to drive a motor

Background Information:

The DRV8833 motor driver chip converts a lower power signal (from the microcontroller: 5v, 40mA) to a high power signal (9v, 1.5A) to drive the motor. It can be thought of as a switch or relay to the driver supply voltage (9v in this case). Since the supply voltage is fixed this would result in a fixed motor speed. To regulate the speed a PWM signal is used to quickly switch the output on and off so that the average output voltage can be controlled.

The term motor "driver" is also commonly called "amplifier" or "chopper". The DRV8833 driver is capable of providing 2.7-10.8v at 1.5A RMS on each channel.

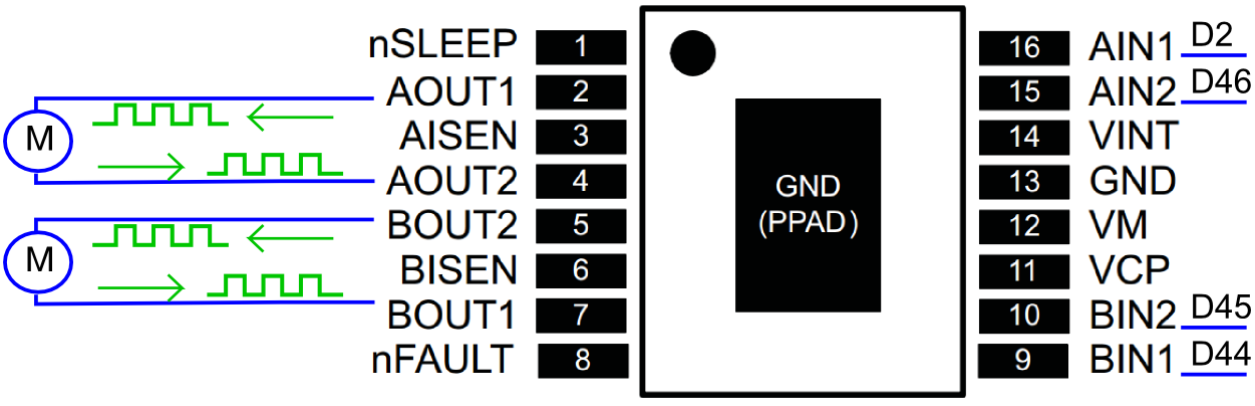
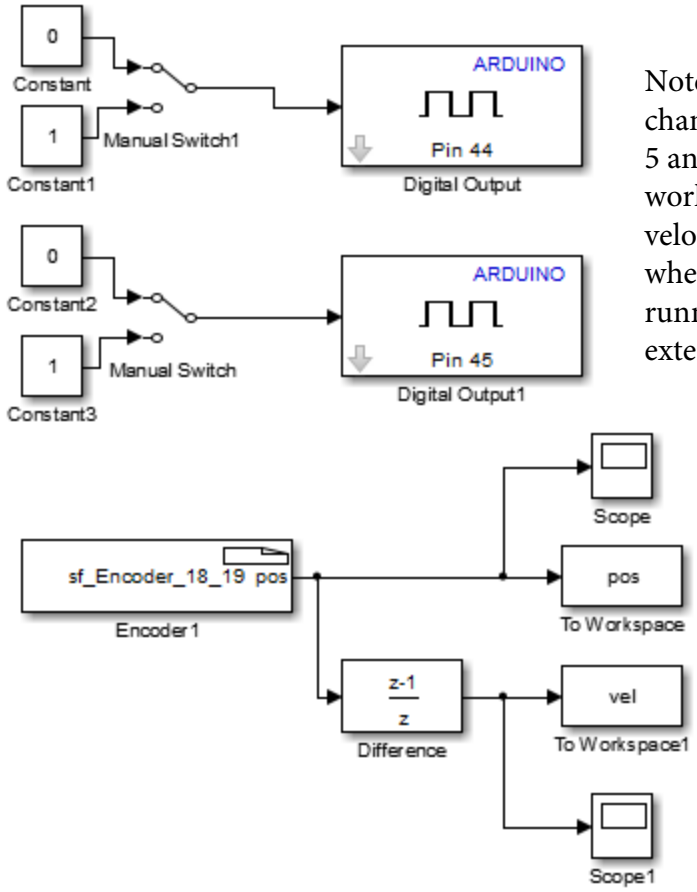


Figure 1: DRV8833 pinout/wiring diagram

Simulink Model

Build and run the following simulink diagram.



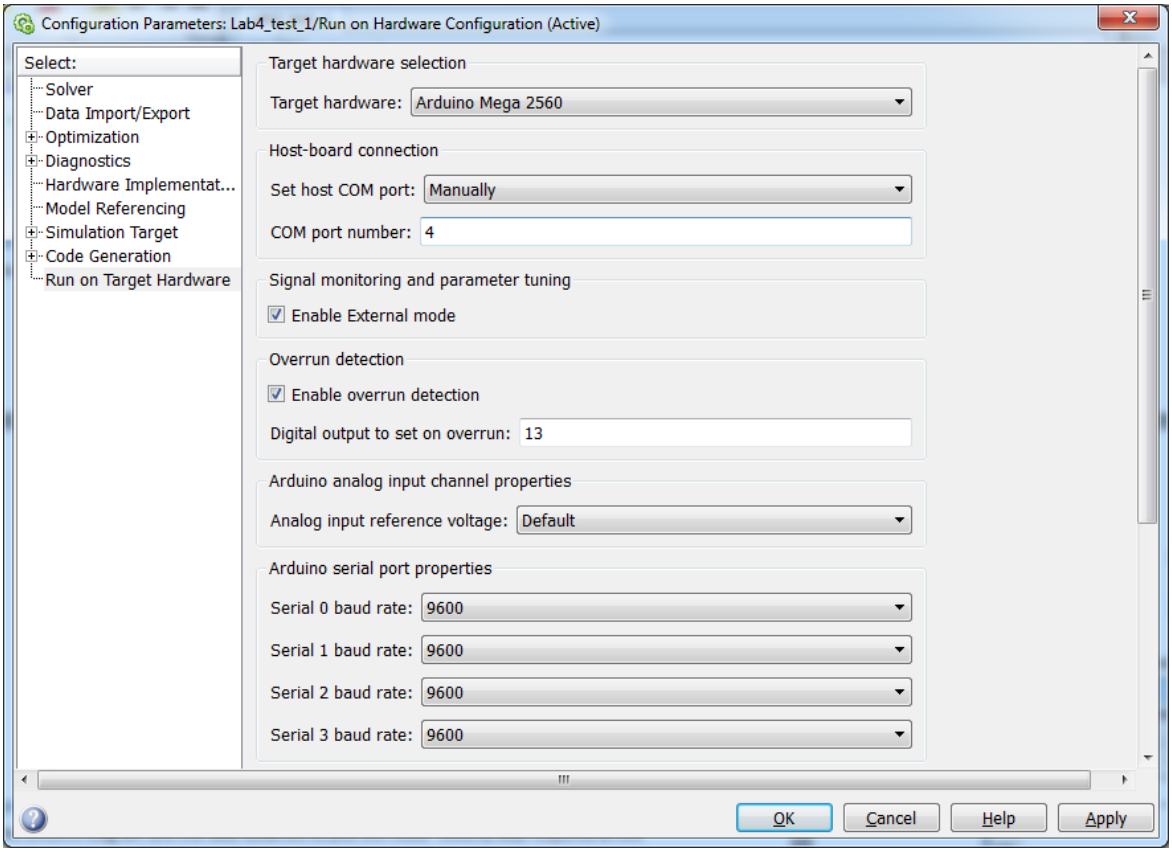
Note 1/20/14: Be sure to change the pins 44 and 45 to 5 and 2 and it should start working. Change the velocity to zero to make the wheels stop. It will keep on running after you stop external mode.

- Run the model in external mode to log the data
  - Sampling rate of .03 seconds
- Observe the response of the system
  - toggle the manual switches
  - toggle the On/Off switch on your MinSegMega board – this will determine the voltage source of the driver chip (VM):
    - OFF - USB voltage (~4.5 volts)
    - ON – Battery voltage (~9v fully charged)
- Stop/disconnect from the system, save the data then plot the results

```
%save ext_dat pos vel   % save the data
load ext_dat            % load the data
figure, plot(pos)
figure, plot(vel)
```

The velocity data obtained in external mode will be noisy. The primary reason seems to be that the time between samples is not actually fixed/constant. In the configuration parameters

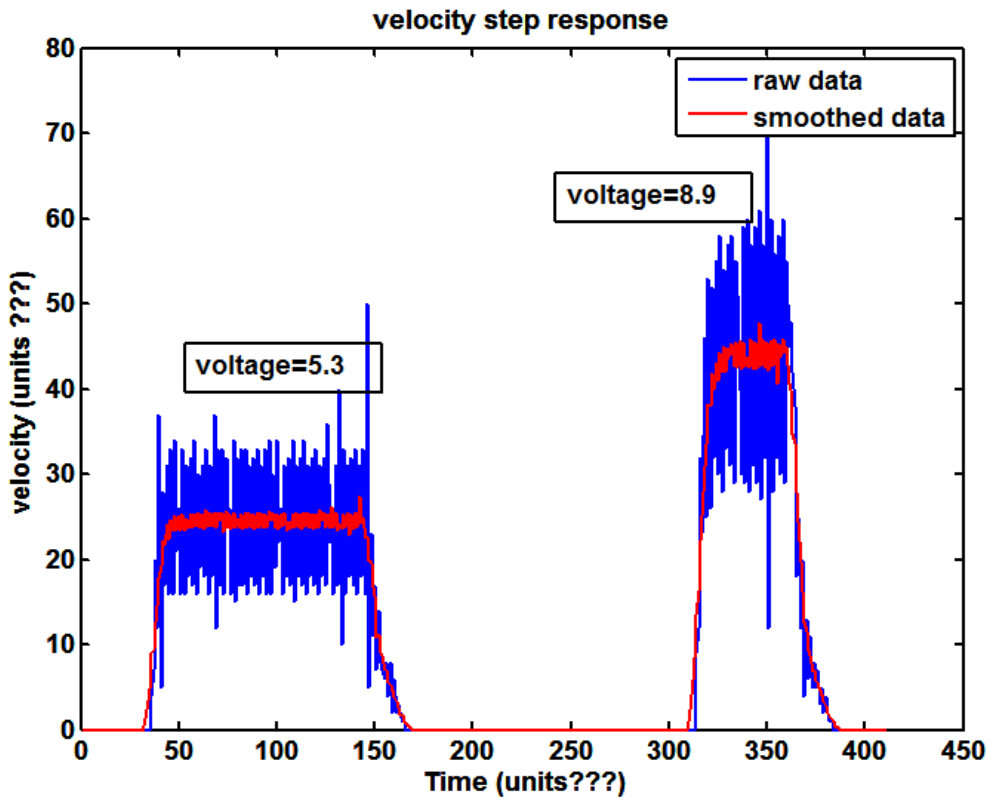
if you check the box “Enable overrun detection” then the digital pin you specify will be set high if the controller cannot execute your code in the sample time you specify. Check this box, select pin 13, and connect to the device. You will notice that the LED on pin 13 is always on. This indicates that in external mode the microprocessor has trouble meeting the sample time. This implies the time between each sample is not exactly 30 milliseconds. Any calculation assuming a fixed sample time, such as velocity, will then contain some error (noise) due to the sample time not being constant. Note that this noise is from the measurement system -not actual noise present in the signal or system.



- The actual motor velocity will not be fluctuating like the “noisy” data indicates. To make the data more useful use the “`smooth`” command in Matlab. Type “`help smooth`” to obtain details on the smoothing method
  - `vel_smooth=smooth(double(Vel), 10)`
    - This function will use a moving average over 10 samples to smooth the velocity data
    - double – this converts the int16 values in Vel to doubles. If this is not done the function will complain

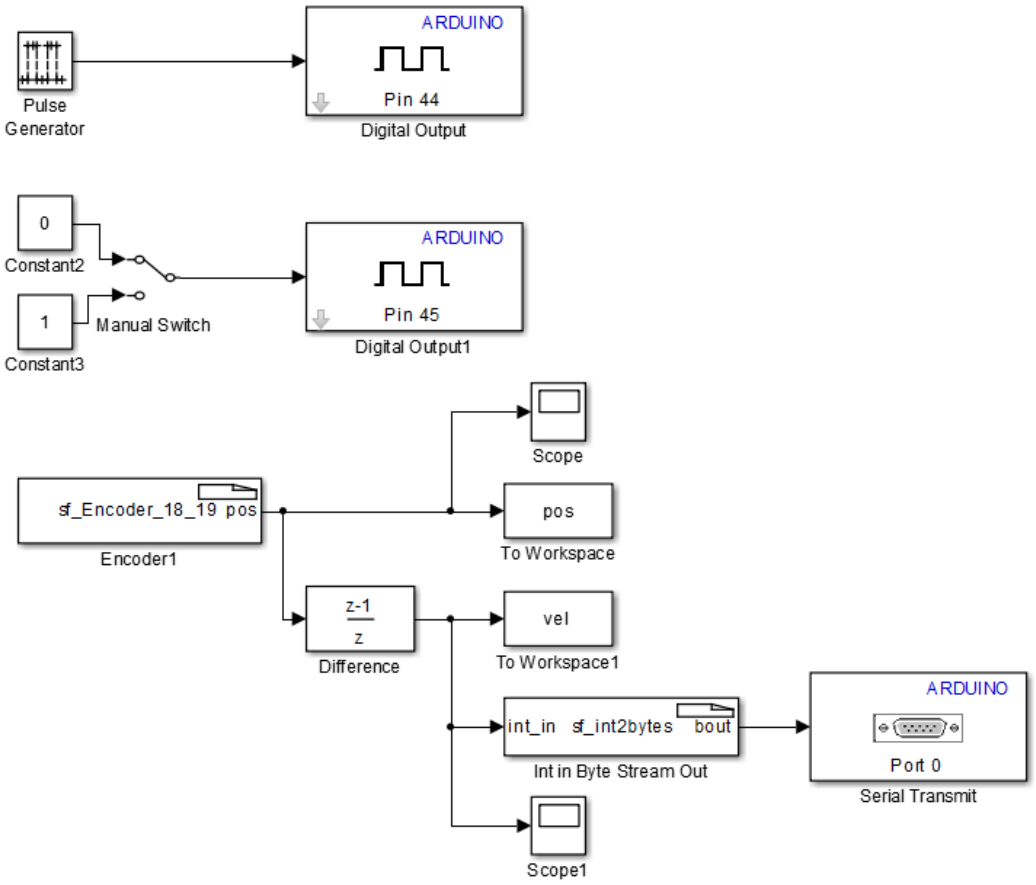
Questions:

- Provide a plot that contains the velocity step response at two different voltages. Provide the raw data and a smoothed response. Plot the velocity in RPM and time in seconds. An example plot is shown below for an unknown motor with unknown units:

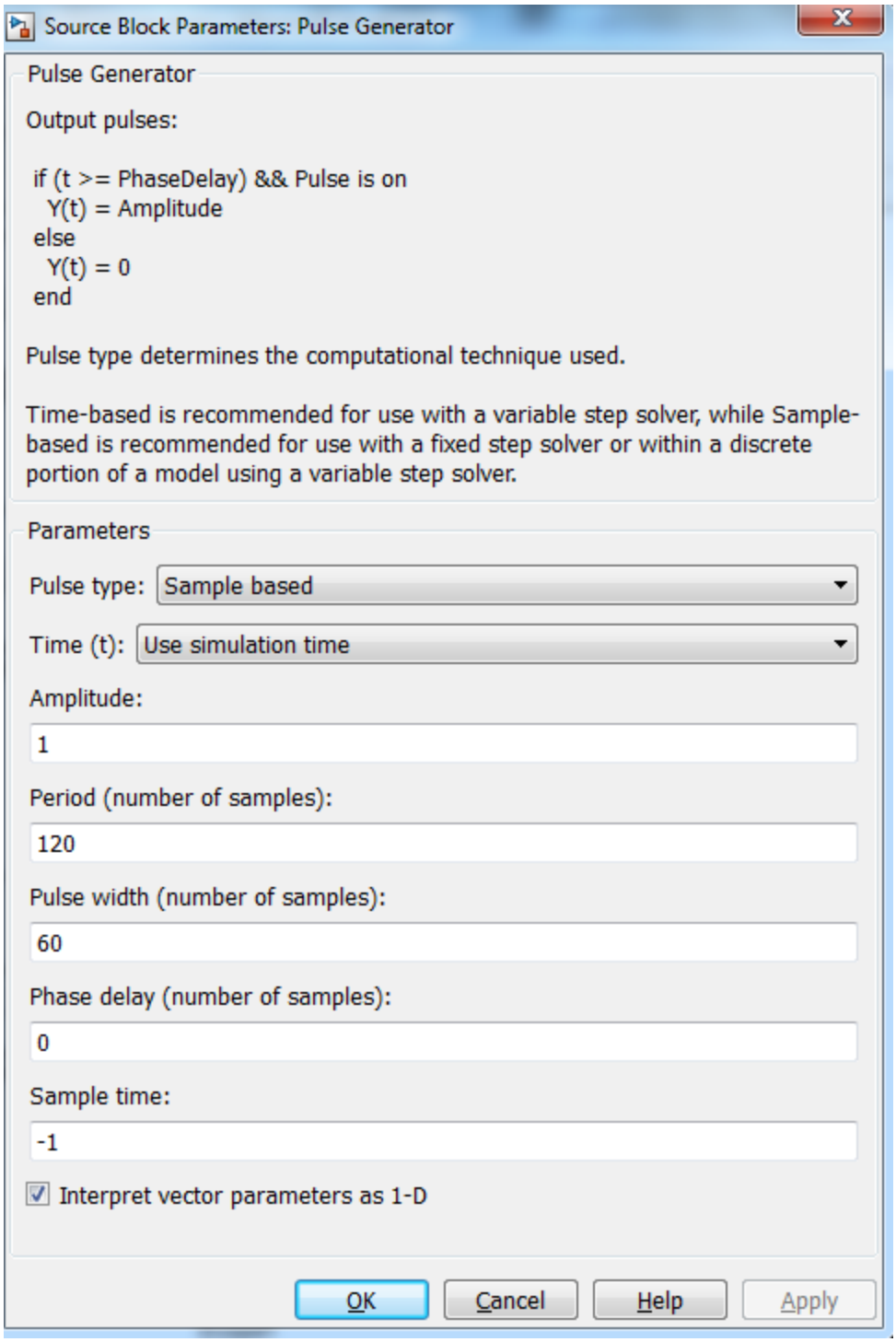


Part 2: Step Response Data from Serial Port

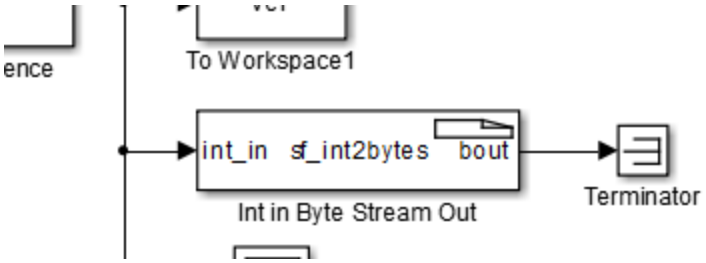
Obtain same velocity information directly from the serial port:



- Be sure external mode is not checked (in the configuration parameters)
- Toggle the ON/OFF switch on the MinSegMega board to obtain the step response at USB voltage and at battery voltage
- The switched cannot be toggled so use a pulse generator block:



- Use RealTerm to capture the data to a file
- Also this model in external mode to obtain the data in the pos and vel variables
    - When the system is running toggle the ON/OFF button to get the response at ~9V in addition to the USB voltage
    - Temporally change the “Serial Transmit” block to a terminator so external mode can be used.



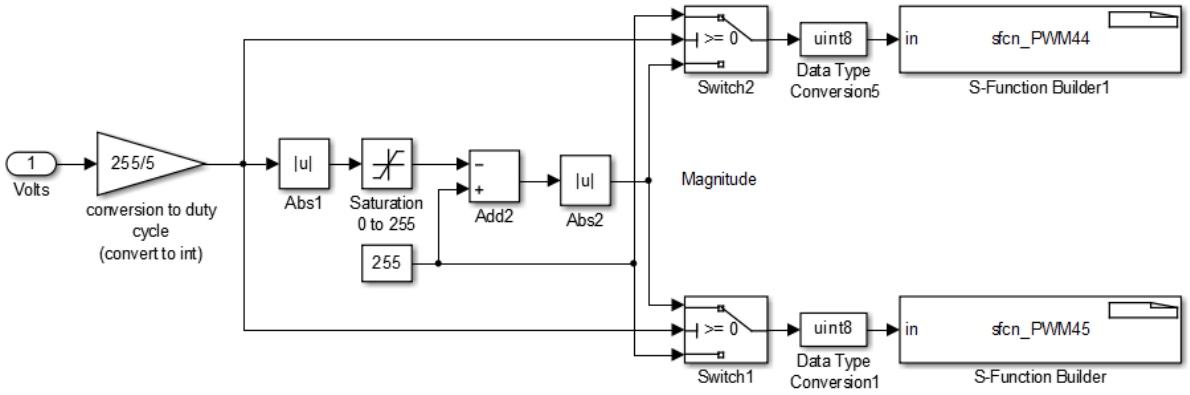
- Plot this data with the data from external mode on the same plot:

```
%save ext_dat pos vel    % save the data
load ext_dat             % load the data
figure, plot(pos)
figure, plot(vel)

% plot serial data:
fid = fopen('dat.bin','r', 'b')
dat = fread(fid, inf, 'int16')
hold on
plot(dat, 'r')

legend('external mode', 'direct serial data')
```

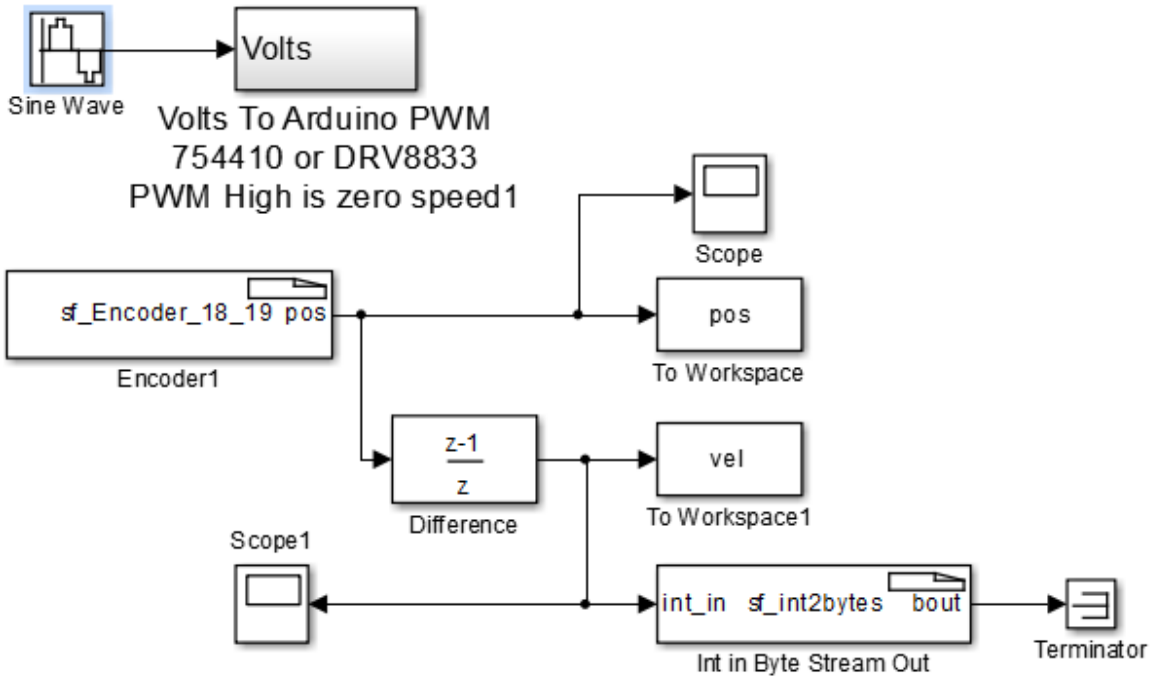
Part 3: Motor Logic: Direction and Magnitude

To regulate the speed a PWM signal is used to quickly switch the output on and off so that the average output voltage can be controlled. In addition the correct switching logic needs to be implemented so that the motor can change direction based on the sign of the input.

The subsystem block below is used to correctly output the correct magnitude and logic of the PWM if the only input is a scalar input voltage (which could be negative). In this example a high signal is sent to both outputs to create a zero speed.

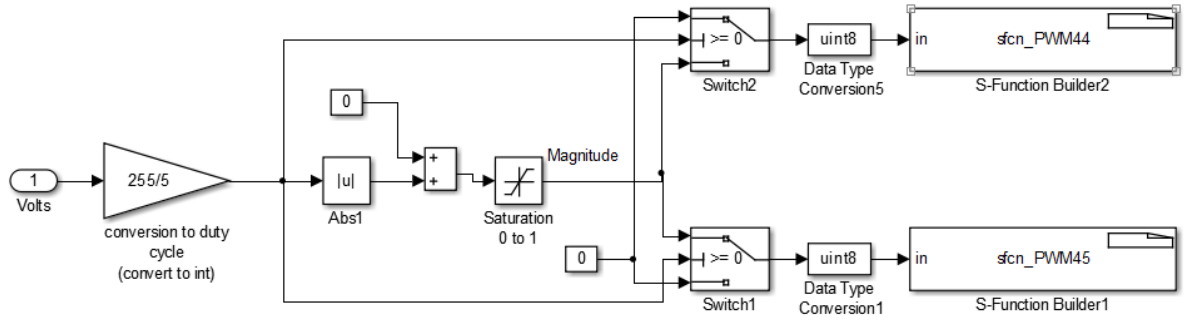


Use this subsystem block to generate a sinusoidal motor response:

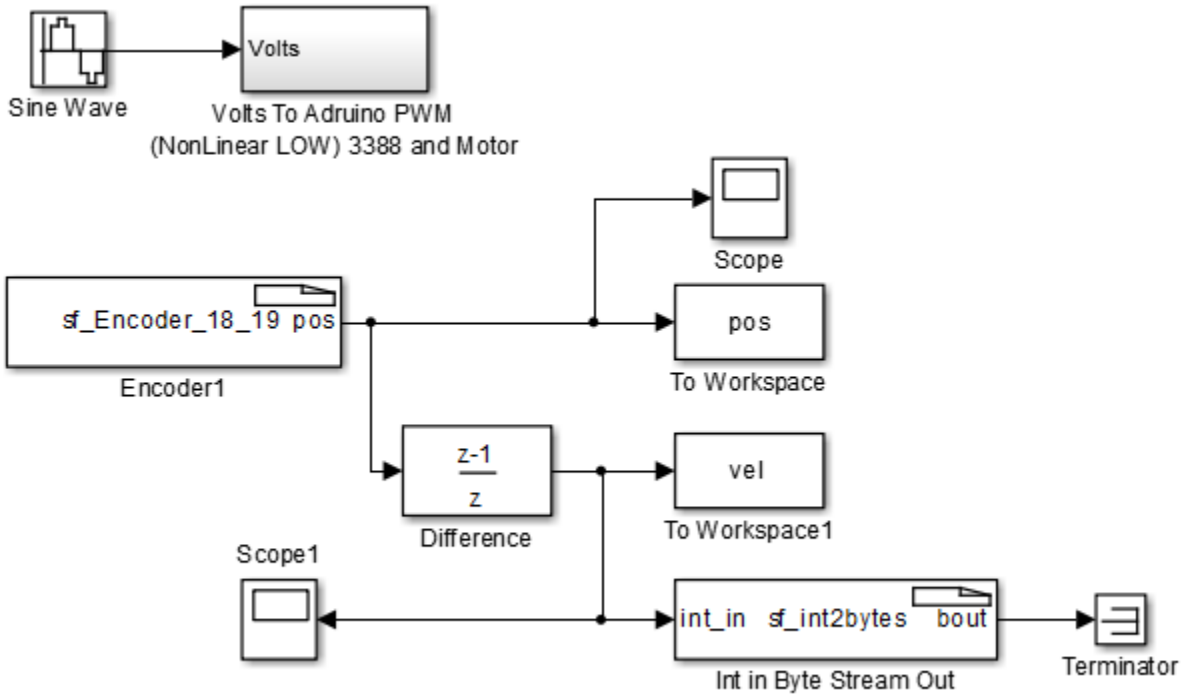


Provide a plot of the velocity in external mode and by obtaining data directly with the serial port.

A different logic can be implemented so that a low signal sent to both outputs corresponds to a zero speed:



Use this subsystem block to generate a sinusoidal motor response:



Plot this response with the previous response from the logic high implementation.