Assignment – ESE S4 Embedded Systems Software – week 5

Positioning an RC Servo at a commanded angle

In this week's lab, we will combine multiple peripherals to perform a task. A timer configured as a Pulse Width Modulator (PWM) will generate a waveform to position an RC servo. The setpoint for the servo position will be derived from an analog voltage read from a potentiometer attached to an Analog-to-Digital Convertor (ADC).

Portions of this lab document come from Zhu

- Lab_07_Servo_PWM_Output
- Lab_10_ADC

As always, when building circuits and when working with active circuits, wear safety glasses.

Due: ***Thursday*** 27 February 2020
Submit your code listing, schematic, and required scope screen shots to the eConestoga dropbox by 11:59 pm

Submit lab report to dropbox including:

- Your commented c source code and header files
- Schematic sketch showing the connections from the Discovery board to the potentiometer and RC servo
- Labelled scope shots showing the RC servo drive waveform including measured pulse width, and the measured analog input voltage for three positions, including measured pulse width:
  - Centred
  - 45 degree clockwise from centre position
  - 45 degree counter-clockwise from centre position

Demo when completed or during inter-class period during the week

References:

Chapter 15 General-purpose Timers. Esp. 15.3
Chapter 20 Analog-to-Digital Convertor. Esp. 20.8, 20.9, fig. 20-12

The basic steps of the lab are:

1. Connect an external LCD module to your Discovery Board as covered in previous labs. Use your Keil project and code from the LCD lab as a starting point.

2. Connect a RC servo to the STM32 timer module output pin (see below) and write software to generate position control waveforms as outlined in class.

3. Connect a 10k potentiometer to an analog (ADC) input (see below) so that moving the pot over its full range will change the voltage applied to the ADC input within safe limits (0 V to 3 V)

4. Write software that will initialize all the hardware and then loop continuously reading the potentiometer voltage, positioning the RC servo at a position determined from ADC value, and updating the LCD display with current values

You will need to write the following:

- A function to initialize the timer channel to generate a servo positioning waveform. A function to set the on-time of the servo position waveform to a value between 600 us and 2400 us.

- A function to initialize the ADC channel as outlined below. A function to read a single sample of that channel.

- You will need to modify your LCD code from lab 4 to use PE10 instead of PE8 for the Register Select (RS) line since PE8 will be used as the timer output to drive the servo.

- A main() to do the following:

    - Call functions to initialize the GPIO ports, LCD module, timer module (PWM), and ADC

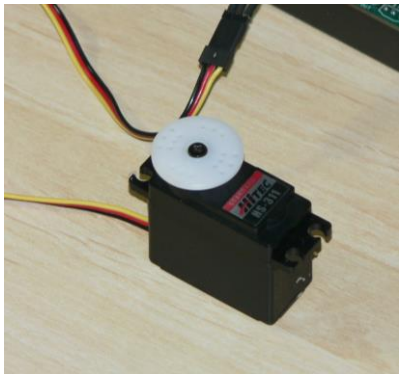
    - Write a fixed message to the display with headings for the analog input voltage (raw & mV) and commanded RC servo position (degrees)

    - Loop forever doing the following:
        - Read the 12-bit ADC value
        - Calculate the corresponding position to command the RC servo to move to and write the appropriate value to the PWM channel
        - Update the LCD with the current values of the raw ADC reading, the ADC reading in mV, and the servo output angle (+/- 90 deg).

To receive maximum credit, use good coding style, appropriately separate your code into modules, and comment as outlined for previous labs.

RC Servo Motor

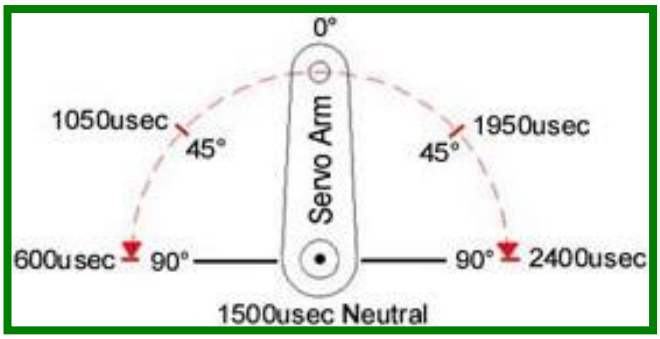
An RC servo motor is a small actuator used primarily for radio-controlled cars and airplanes. The position of the output "horn" is controlled by an integral position control system that includes position feedback and an H-bridge capable of driving an internal PM-DC motor to move the horn to a commanded position.

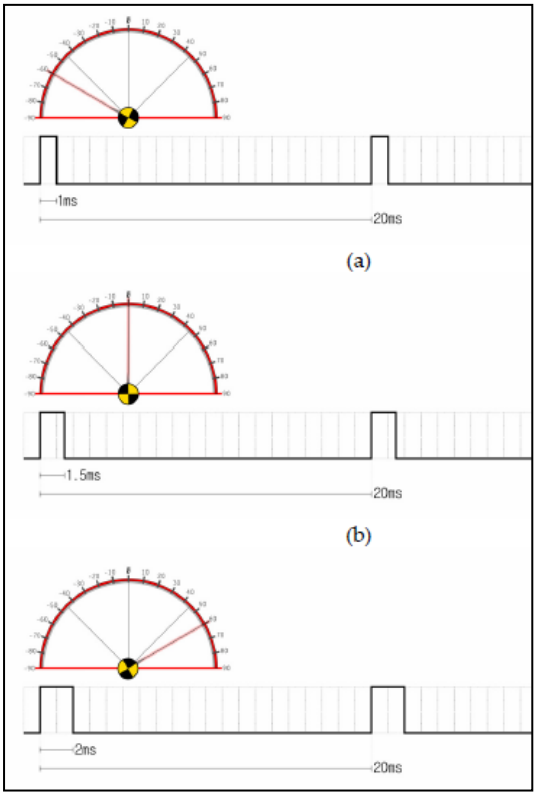


3 Wires

+5V power, 250 mA (red)
Gnd (black)
Control Signal (yellow)

In this lab, we will use HiTec HS-311 servo motor. It can rotate approximately 180 degrees, with 90 degrees in each direction from centre. The position is controlled by the width of a pulse that must repeat every 20 ms as outlined in the below:





Connect the RC servo control signal to PE8 of the Discovery Board – the same GPIO pin as the green LED. Connect the ground line to a ground pin on the Discovery board too. The RC Servo should be powered from your bench supply set to 5V (300 mA) since it draws significant current and may cause the processor on your Discovery board to reset.

Configure the timer driving the PWM output with a Counter Frequency of 1 MHz, and a PWM Frequency of 50 Hz (20 ms period). Adjust the CCR1 register value to set the output pulse width of the signal to the RC servo. Use PWM Mode 1 (output high until CCR value reached) and upcount. ***Refer to textbook section 15.3.2 and example 15-4.***

Potentiometer and ADC connection

A potentiometer (pot) is a three-terminal variable resistor. It uses a sliding contact and works as an adjustable voltage divider. When two outer terminals connected to Vcc and the ground respectively, the center terminal generates a voltage that varies from 0 to Vcc, depending on the position of the sliding contact. You have a potentiometer in your STM kit. **Connect the ends of the resistor as shown and connect the wiper to pin PA1 of the Discovery board**. The potentiometer voltage can be read on ADC channel 1, input #6.

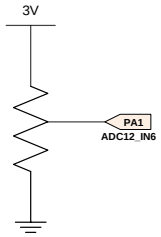


Figure 1. Measure the voltage from a potentiometer-based divider (Vcc = 3V)

Configure the ADC input as outlined in figure 20-12 of the textbook. (No need to configure pin PB2 to drive the red LED for this lab.) ***Use the software trigger of a single read of channel 6 (PA1) as outlined at the end of figure 20-12.***