Assignment – ESE S4 Embedded Systems Software – week 5

Positioning an RC Servo at a commanded angle

For this week's lab, we will configure a channel on a hardware timer module as a Pulse Width Modulator (PWM). We will use this PWM to generate a waveform to position an RC servo. The setpoint for the servo position will change at regular time intervals determined by using the SysTick timer module.

Portions of this lab document come from Zhu

- Lab_07_Servo_PWM_Output

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

Due: ***Monday*** 1 March 2021
Submit your code listing 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
- Labelled scope shots showing the RC servo drive waveform including measured pulse width for three positions:
  - Centred
  - 45 degree clockwise from centre position
  - 45 degree counter-clockwise from centre position

Demo when by video or arranged zoom meeting once completed.

References:

Chapter 15 General-purpose Timers. Esp. 15.3

The basic steps of the lab are:

1. 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.

2. Write software that will initialize all the hardware and then loop continuously changing the commanded position to the RC servo over a range of +/- 45 degrees. One transit from one limit to the other should take 2 seconds; the servo should pause at the limit for 1 second, and then transit to the other limit over 2 seconds where it will also pause for 1 second.

3. Report the position of the servo by printing the output angle and pulse width to TeraTerm using the UARTprintf() function.

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.
- Functions using SysTick to wait for specified periods of time as required for your code to meet the specified sweep times of the RC Servo.

- A main() to do the following:

    - Call functions to initialize the GPIO ports, UART module, and timer module (PWM).

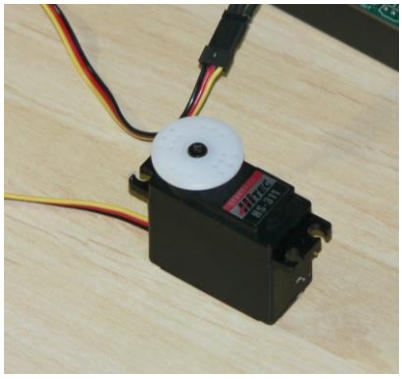
    - Write a fixed message to the display with headings of RC Servo Angle (degrees), and Command Pulse Width (usec)

    - Loop forever doing the following:
        - Calculate the corresponding position to command the RC servo to move to and write the appropriate value to the PWM channel
        - Update TeraTerm with the current values as specified above.
        - Use the SysTick based functions to ensure the required timing of the RC servo motion.

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

RC Servo Motor

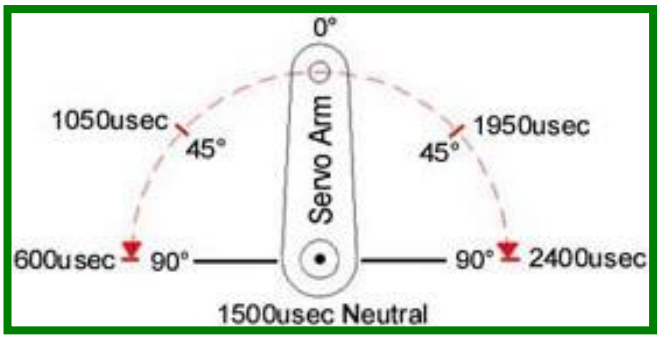
An RC servo motor is a small actuator used primarily for radio-controlled cars and airplanes. The position of the output "horn" is controlled by an integral position control system that includes position feedback and an H-bridge capable of driving an internal PM-DC motor to move the horn to a commanded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

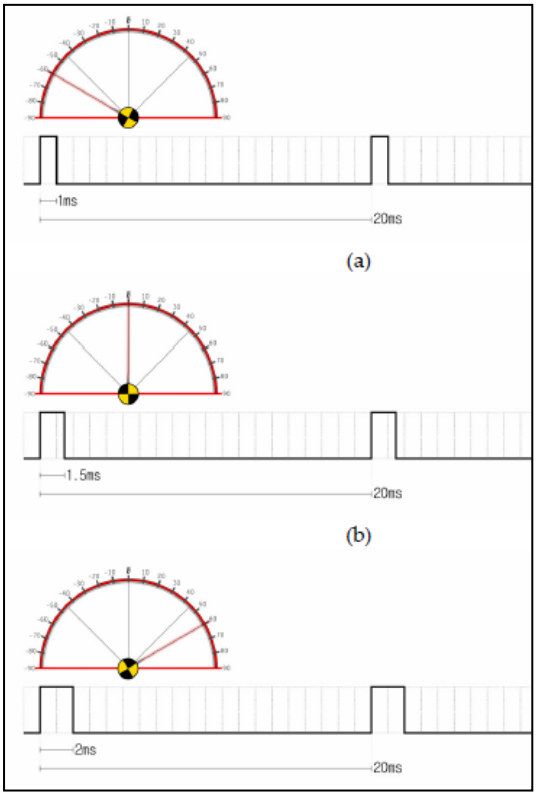


3 Wires

+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:





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.***