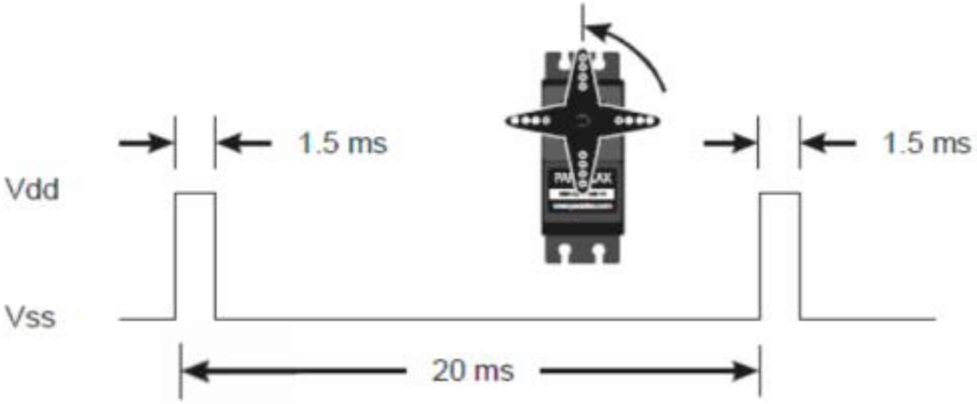
Rolf Anderson
Alec Meyer
Adam LaRocque

Lab 8 – Part 1



Moving the servo involves using a 16 + 8 bit timer to manipulate a PWM output on Port B pin 5. The pulsewidth of the high PWM signal determines where the servo moves to. If the pulsewidth is 1ms, the servo will move to 0 degrees. If it is 2ms, it will move to 180 degrees. Thus there is a linear relationship between the pulsewidth of the high output, and the degree of the servo position. The timer facilitates this by counting down and having an adjustable match value to determine when to set the output low. The period of this timer cycle is 20ms, this is not variable. The timer will be counting down from this to 0, so this will be the value in the load registers. Since the timers count clock cycles rather than seconds, we have to convert 20ms into clock cycles on the 16mhz processor. This turns out to be 320000 (.02 * 1.6E7). Using basic algebra we can find a formula to determine the match register values.

$$match = 320000 - \left(\frac{degrees}{180} + 1\right) * (\frac{320000}{20})$$

$$match = 320000 - (88.88 * degrees + 160000)$$

$$0 \leq degrees \leq 180$$

From when the counter begins at the load value of 320000 (equivalent to 20ms or .02s), until when it reaches the match value (for example, 0 degrees would give a match value of 304000) the output PWM signal on PB5 will be high. It will be low from the match value until the timer reaches 0. This timer runs continuously to facilitate easy and quick changes to the servo position.

Registers used:

- RCGCGPIO // turn on clock to pb5
- PORTB_AFSEL // turn on alternate function for pb5
- PORTB_PCTL // select t1ccp1 as alternate function
- PORTB_DIR // set pb5 as output
- PORTB_DEN // enable it as digital (pwm)
- RCGCTIMER // turn on clock to timer1
- TIMER1_CTL // disable t1b to configure it, set PWM to be non-inverted, re-enable
- TIMER1_CFG // configure t1 to be 16bit
- TIMER1_TBMR // configure t1b to be periodic, PWM, count down, disable capturemode
- TIMER1_TBILR // least significant 16 bits of the load interval value
- TIMER1_TBPR // most significant 8 bits of load interval value
- TIMER1_TBMATCHR // least sig 16 bits of match value
- TIMER1_TBPMR //most sig 8 bits of match value

Note: after servo_init, the only registers that change are the TBMATCHR and TBPMR.