Steve Guerrero

1/31/2018

ECPE 155

Pre-Lab 2

This pre lab is based off the initialization and configuration example in the Tiva C series datasheet (section 20.4) in sequential order. This is for a 25kHz frequency with a 25% duty cycle on pin 0 and 75% duty cycle for pin 1.

Using module 0 and 1 on Port B for this example below:

|  |  |
| --- | --- |
| Register name | Register Value |
| PWM Run-Clock-Gate-Control [SYSCTL\_RCGCPWM] | |= (3 << 0) |
| GPIO Alt-Func-Select  GPIO\_PORTB [GPIO\_AFSEL] | |= ( 1 << 6 ) | ( 1 << 7 ) |
| GPIO Port Control  GPIO\_PORTB [GPIO\_PCTL] | |= ( 4 << (0 \* 4) ) | ( 4<< (1 \* 4) ) |
| GPIO Direction  GPIO\_PORTB [GPIO\_DIR] | |= ( 1 << 6 ) | ( 1 << 7 ) |
| GPIO Run-Clock-Gate-Control [SYSCTL\_RCGCGPIO] | |= ( 1 << 0 ) | ( 1 << 1 ) |
| PWM0CTL (disable for configuration) | = 0x0000.0000 |
| PWM0GENA | = 0x0000.008C |
| PWM0GENB | = 0x0000.080C |
| PWM0LOAD | = 0x0000.018F |
| PWM0CMPA | = 0x0000.012B |
| PWM0CMPB | = 0x0000.0063 |
| PWM0CTL (enable after configuration) | = 0x0000.0001 |
| PWMENABLE | = 0x0000.0003 |

Some code I wrote that I hope works with a few modifications;

void init\_PWM\_servo(void)

{ // Pulsed Width Modulation (PWM)

//--------- INITIALIZATION AND CONFIGURATION OF PWM0 in Port B ---------

// STEP(1)

// Enable the RCGCPWM register in SYSCTL

SYSCTL[SYSCTL\_RCGCPWM] |= SYSCTL\_RCGCPWM\_M0;

SYSCTL[SYSCTL\_RCGCPWM] |= SYSCTL\_RCGCPWM\_M0;

// STEP(2)

// Enable PORT B gate clocks. The second write is only necessary

// as a delay so that the clock is stable before writing to the GPIO

// registers. If we don't delay, we may get a fault.

SYSCTL[SYSCTL\_RCGCGPIO] |= SYSCTL\_RCGCGPIO\_PORTB;

SYSCTL[SYSCTL\_RCGCGPIO] |= SYSCTL\_RCGCGPIO\_PORTB;

// STEP(3)

// Enable the appropriate pins for their alternative function using GPIOAFSEL register

// PB(6) = PWM\_M0

GPIO\_PORTB[GPIO\_AFSEL] |= GPIO\_PIN\_6; // Sets Port G to function as I2C peripheral

// STEP(4)

// Configure the PMC fields in the GPIOPCTL register. Assign PWM signal to pins

GPIO\_PORTB[GPIO\_PCTL] &= ~(0Xf << (0 \* 4)); // mask off bits to set

GPIO\_PORTB[GPIO\_PCTL] |= (4 << (0 \* 4)); // corresponding bits: 0x.0100

GPIO\_PORTB[GPIO\_DIR] |= GPIO\_PIN\_6; // Sets Outward direction of Port B Pin 6

GPIO\_PORTB[GPIO\_DEN] |= GPIO\_PIN\_6; // Enables Port B Pin 6

// STEP(5)

// PWM Clock Config

PWM0\_CC\_R |= (1 << 8); // Enables use of PWM clock divider

PWM0\_CC\_R |= (0x5 << 0); // Divides SYS CLOCK (120MHz) by 64 = 1.875MHz

// STEP(6)

// PWM Signal Config

PWM0\_0\_CTL\_R | = 0x00000000; // Disables the PWM signal for configuration

PWM0\_0\_GENA\_R = 0x8c; // PWM Generation A Control

PWM0\_0\_LOAD\_R = 0x0927c; // PWM Load value for a 20ms period = 50Hz freq signal

// Calc ->(1.875MHz / 50Hz) = 37,500 or (0x927C)

PWM0\_0\_CMPA\_R = 0x00008b29; // for duty cycle (1 - (PWM0CMPA / PWM0LOAD))

// (1 - (35625 / 37500)) = 5% duty cycle

PWM0\_0\_CTL\_R |= (1 << 0); // Enables the PWM signal

PWM0\_ENABLE\_R = 0x00000001; // Enables PWM Periph

}

void servo\_step(uint32\_t \*step)

{ // Modifies the PWM signal between 5-10% duty cycle

// Steps from -90 deg @ 5% to 90 deg @ 10%

// Step angle resolution = 1.8 deg per step

// (1 - (35625 / 37500)) = 5%

// (1 - (33750 / 37500)) = 10%

uint32\_t data;

data = ((\*step \* 19) + 33750);

PWM0\_0\_CTL\_R = data;

}