**Date Submitted: 10.13.2018**

**Task 00: *No submission***

**Youtube Link: N/A**

**------------------------------------------------------------------------------------**

**Task 01:** Change the PWM duty cycle to make the servo motor to do a loop of a complete sweep

from 0 to 180 deg.

**Youtube Link**: <https://youtu.be/kWquLqR_93A>

Modified Code:

ui8Adjust adjusting this variable to between 30 and 125 was the necessary modification from the give code to create a signal with a Pulse Width between

8ms ~2.2ms which generated the 180° sweep.

**if**(ROM\_GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_4)==0x00)

{

ui8Adjust--;

**if** (ui8Adjust < 30)

{

ui8Adjust = 30;*//1.82µs \* 30 ~.55ms 0°*

}

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_0, ui8Adjust \* ui32Load / 1000);

}*//enf if1*

**if**(ROM\_GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_0)==0x00)

{

ui8Adjust++;

**if** (ui8Adjust > 125)

{

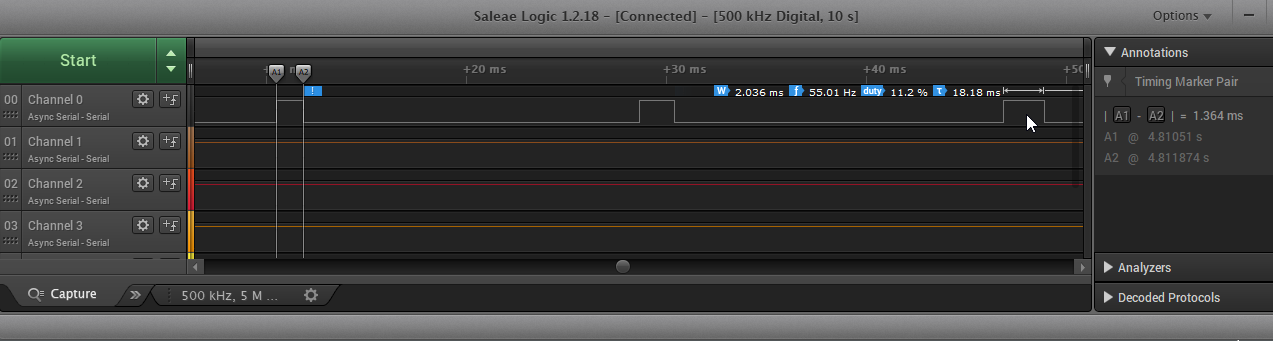
ui8Adjust = 125;*//1.82µs \* 125 ~2.28ms 180°*

}

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_0, ui8Adjust \* ui32Load / 1000);

}*//end if2*

ROM\_SysCtlDelay(13000);

****

**Task 02:**

**Youtube Link: <https://youtu.be/7Ew1TgyXyko>**

Modified Added /Code:

*//Constants for Min and Max*

*//Values*

#define PWM\_FREQUENCY 55

#define MN\_Bright 100 *//10% Task 2&3*

#define MX\_Bright 900 *//90% Task 2&3*

*//Values*

ROM\_GPIOPinTypePWM(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3); *//Values Pins set to output*

ROM\_GPIOPinConfigure(GPIO\_PF1\_M1PWM5);*//R Task 2*

*//To move between 10% and 90%*

**for**(; R <= MX\_Bright; R++)

{

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, R \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//end for 1*

**for**(; R >= MN\_Bright; R--)

{

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, R \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//end for2*

**Task 03:**

**Youtube Link:** **<https://youtu.be/NZHVQFW60hE>**

Modified Added /Code:

#define MN\_Bright 100 *//10% Task 2&3*

#define MX\_Bright 900 *//90% Task 2&3*

ROM\_GPIOPinConfigure(GPIO\_PF1\_M1PWM5);*//R Task 2*

ROM\_GPIOPinConfigure(GPIO\_PF2\_M1PWM6);*//G Task 2&3*

ROM\_GPIOPinConfigure(GPIO\_PF3\_M1PWM7);*//B Task 2&3*

*//R&G get incremented once and in bottom loop B goes from 10-90% then the middle loop*

*//increases to 90% and finally the 1st loop, in the second set of for loops this happens in reverse*

**for**(; R <= MX\_Bright; R++)

{

**for**(; G <= MX\_Bright; G++)

{

**for**(; B <= MX\_Bright; B++)

{

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, B \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//end*

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, G \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, R \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}

**for**(; R >= MN\_Bright; R--)

{

**for**(; B >= MN\_Bright && R == MX\_Bright; B--)

{

**for**(; G >= MN\_Bright && B == MX\_Bright && R == MX\_Bright; G--)

{

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, G \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//end*

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, B \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//end 2nd for*

ROM\_PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, R \* ui32Load / 1000);

ROM\_SysCtlDelay(12000);

}*//1st for*