**Assignment 9:  
Pulse Width Modulation and Servos**

**Cal Poly CPE 329-01**

**Spring 2017**

Brendan Baronia

Melinda Ong

Professor Gerfen, Jeffrey B.

**May 31, 2017**

Project Demonstration:

<https://youtu.be/bVOOmIL9UEs>

Code:

***main.c:***

#include "msp.h"  
#include "keypad.h"  
#include <stdio.h>  
  
#define CYCLETIMER 100  
#define MAXANGLE 180  
  
void processKey();  
void portRemap();  
void delayDuration(int n);  
  
static int angle;  
  
void main(void)  
{  
 WDTCTL = WDTPW | WDTHOLD; /\*Stop watchdog timer\*/  
  
 CS->KEY = CS\_KEY\_VAL; // unlock CS registers  
 CS->CTL0 = 0; // clear register CTL0  
 CS->CTL0=CS\_CTL0\_DCORSEL\_0; /\*1.5 MHz\*/  
  
 CS->CTL1 = CS\_CTL1\_SELA\_2 | CS\_CTL1\_SELS\_3 | CS\_CTL1\_SELM\_3; // select clock sources  
 CS->KEY = 0; // lock the CS registers  
  
  
 P2->SEL0 |= 0x80; /\*P2.7 is TA0.4 out\*/  
 P2->SEL1 &= ~0x80;  
 P2->DIR |= 0x80;  
  
 portRemap();  
  
 TIMER\_A0->CCR[0] = 30000 - 1; /\*period\*/  
 TIMER\_A0->CCR[4] = 0; /\*PWM duty cycle\*/  
 TIMER\_A0->CCTL[4] = 0xE0; /\*reset/set mode\*/  
 TIMER\_A0->CTL = 0x214; /\*SMCLK, count up, clear TA0R register\*/  
  
  
 //TIMER\_A0->CTL = TIMER\_A\_CTL\_SSEL\_\_SMCLK | // SMCLK, continuous mode  
 // TIMER\_A\_CTL\_MC\_\_CONTINUOUS;  
  
 keypad\_init(); /\*initialize keypad pins\*/  
 angle = 0; /\*initialize angle\*/  
  
 while (1) {  
 processKey(); /\*get input from keypad\*/  
 TIMER\_A0->CCR[4] = 1100 + 2300 \* angle / MAXANGLE; /\*PWM duty cycle\*/  
 printf("%d\n", angle);  
 }  
}  
  
void processKey() {  
 int firstKey, secondKey = 0;  
  
 firstKey = keypad\_getkey();  
 delayDuration(100); /\*software delay\*/  
  
 /\*check for numeric input\*/  
 if (firstKey > 0 && firstKey != 10 && firstKey != 12) {  
  
 while (secondKey == 0) {  
 secondKey = keypad\_getkey();  
 delayDuration(100); /\*software delay\*/  
 }  
  
 if (firstKey == 11) /\*convert if 0 pressed\*/  
 firstKey = 0;  
 if (secondKey == 11) /\*convert if 0 pressed\*/  
 secondKey = 0;  
  
 angle = firstKey \* 100 + secondKey \* 10; /\*set angle\*/  
  
 if (angle > 180)  
 angle = 180;  
 }  
 /\*check for \* input and valid angle\*/  
 else if (firstKey == 10 && angle > 0)  
 angle -= 10;  
 /\*check for # input and valid angle\*/  
 else if (firstKey == 12 && angle < 180)  
 angle += 10;  
}  
  
void portRemap() {  
 PMAP->KEYID = 0x2D52; /\*unlock PMAP\*/  
  
 P2MAP->PMAP\_REGISTER1 = 34; /\*remap P2.1 to 23 (TPM0.4)\*/  
 P2->DIR |= 2;  
 P2->SEL0 |= 2;  
 P2->SEL1 &= !2;  
  
 PMAP->CTL = 1; /\*lock PMAP\*/  
 PMAP->KEYID = 0;  
}  
  
/\* delay milliseconds when system clock is at 12 MHz \*/  
void delayDuration(int n) {  
 int i, j;  
  
 for (j = 0; j < n; j++)  
 for (i = 300; i > 0; i--); /\* Delay \*/  
}