1. /\*
2. \* File:   huan2251\_lab4\_main\_v001.c
3. \* Author: ericl
4. \*
5. \* Created on February 27, 2023, 2:18 PM
6. \*/
8. #include <stdio.h>
9. #include <stdlib.h>
10. #include "xc.h"
11. #include "huan2251\_lab4\_asimLib.h"
12. // CW1: FLASH CONFIGURATION WORD 1 (see PIC24 Family Reference Manual 24.1)
13. #pragma config ICS = PGx1          // Comm Channel Select (Emulator EMUC1/EMUD1 pins are shared with PGC1/PGD1)
14. #pragma config FWDTEN = OFF        // Watchdog Timer Enable (Watchdog Timer is disabled)
15. #pragma config GWRP = OFF          // General Code Segment Write Protect (Writes to program memory are allowed)
16. #pragma config GCP = OFF           // General Code Segment Code Protect (Code protection is disabled)
17. #pragma config JTAGEN = OFF        // JTAG Port Enable (JTAG port is disabled)

20. // CW2: FLASH CONFIGURATION WORD 2 (see PIC24 Family Reference Manual 24.1)
21. #pragma config I2C1SEL = PRI       // I2C1 Pin Location Select (Use default SCL1/SDA1 pins)
22. #pragma config IOL1WAY = OFF       // IOLOCK Protection (IOLOCK may be changed via unlocking seq)
23. #pragma config OSCIOFNC = ON       // Primary Oscillator I/O Function (CLKO/RC15 functions as I/O pin)
24. #pragma config FCKSM = CSECME      // Clock Switching and Monitor (Clock switching is enabled,
25. // Fail-Safe Clock Monitor is enabled)
26. #pragma config FNOSC = FRCPLL      // Oscillator Select (Fast RC Oscillator with PLL module (FRCPLL))
27. **void** setup(**void**)
28. {
29. CLKDIVbits.RCDIV = 0;  //Set RCDIV=1:1 (default 2:1) 32MHz or FCY/2=16M
30. // Execute once code goes here
31. AD1PCFG = 0x9fff;            //sets all pins to digital I/O
33. }
35. **void** delay(**int** delay\_in\_ms){
36. **int** i = 0;
37. **for** (i = 0; i < delay\_in\_ms; i++){
38. wait\_1ms();
39. }
40. }
41. /\*
42. \*
43. \*
44. \*/
45. **void** initServo() {
46. //use B6 as a pwm pin, pin should be an ouput
47. TRISBbits.TRISB6 = 0;
49. //map output compare to pin b6
50. \_\_builtin\_write\_OSCCONL(OSCCON & 0xBF);
51. RPOR3bits.RP6R = 18;
52. RPOR3bits.RP7R = 19;
53. \_\_builtin\_write\_OSCCONL(OSCCON | 0x40);
55. OC1CON = 0;
56. OC1CONbits.OCTSEL = 0b1;
57. OC1CONbits.OCM = 0b110;
58. OC1R = 1234;
60. OC2CON = 0;
61. OC2R = 1234;
62. OC2RS = 1234;
63. OC2CONbits.OCM = 0b110;
64. OC2CONbits.OCTSEL = 1;
66. T3CON = 0;
67. T3CONbits.TCKPS0 = 1;
68. T3CONbits.TCKPS1 = 0;
69. T3CONbits.TON = 1;
70. TMR3 = 0;
71. PR3 = 40000 - 1;
73. T3CONbits.TON = 1;
74. IFS0bits.T3IF = 0;

77. }
79. **void** setServo(**int** Val){
80. OC1RS = Val; //OC1RS shadow register
82. }
84. **void** initPushButton(**void**){
86. TRISBbits.TRISB8 = 1;
88. T2CON = 0;
89. PR2 = 0xffff;
90. TMR2 = 0;
91. IFS0bits.T2IF = 0;
92. IEC0bits.T2IE = 1;
93. IPC1bits.T2IP = 3;
94. T2CONbits.TCKPS = 0b11;
96. CNPU2bits.CN22PUE = 1;
98. \_\_builtin\_write\_OSCCONL(OSCCON & 0xbf);
99. RPINR7bits.IC1R = 8;
100. \_\_builtin\_write\_OSCCONL(OSCCON | 0x40);
102. IC1CON = 0;
103. IC1CONbits.ICTMR = 1;
104. IEC0bits.IC1IE = 1;
105. IC1CONbits.ICM = 0b010;
106. IC1CONbits.ICI = 0b00;
107. IPC0bits.IC1IP = 4;
108. \_IC1IF = 0;
109. \_T2IF = 0;
111. T2CONbits.TON = 1; //timer 2 enabled
112. }
114. **volatile** unsigned **int** overflow = 0;
115. **void** \_\_attribute\_\_((interrupt, auto\_psv)) \_T2Interrupt(**void**) {
116. overflow++;
117. \_T2IF = 0;
118. }
120. **int** main(**void**) {
121. setup();
122. initServo();
123. initPushButton();
125. **while**(1) {
126. //        setServo(3600); //25%
127. //        delay(1000);
128. //        setServo(2400); //75%
129. //        delay(1000);
130. setServo(10000);
131. delay(1000);
132. setServo(30000);
133. delay(1000);
134. }
135. **return** 0;
136. }