```
; SERVO.ASM - Programmed by Maarten Hofman
; 310806: Started programming
; 020906: Continued programming
; 030906: Debugged the programming interface
; 020608: Potentiometer version
; 030608: Enable correct ANSEL values
; 042119: Will Collier: corrected delay time between control pulses

; Hardware description
; RA0: Potentiometer
; RC0: Servo output

 LIST R=DEC
 INCLUDE "p16f688.inc"

 CBLOCK 0x020
   IROMaddress : 1
   IROMvalue : 1
   Checksum : 1
   Checksumnew : 1
   t0 : 1
      Pulselength : 1
   Servo1pulse : 2
      ServoDelay : 1
   Command : 1
   Loop : 1
 ENDC

 __CONFIG _CP_OFF & _WDT_OFF & _INTRC_OSC_NOCLKOUT & _PWRTE_ON & _MCLRE_OFF

 PAGE

 org 0
  goto Programstart
 org 4
  ; interrupt routines
 org 256
Programstart
; Start of port initialisation
  nop
  movlw 0x07                    ; Turn comparators off
  movwf CMCON0
  bsf STATUS,RP0
    movlw    0x01
    movwf    TRISA ^ 0x080      ; Make PortA all input
;  movlw B'01110000'            ; 8 MHz bitmask
;  iorwf OSCCON^0x80,f     ; Update to increase frequency
  movlw    0x10       ; A2D Clock Fosc/8
  movwf    ADCON1 ^ 0x080
  movlw    0x01       ; we want all Port A pins Analoga
  movwf    ANSEL ^ 0x080
;  clrf TRISA ^ 0x080       ; PORTA: output
  clrf TRISC ^ 0x080        ; PORTC: output
  bcf STATUS,RP0
```

```
  movlw    0x01
  movwf    ADCON0    ; configure A2D for Channel 0 (RA0), Left justified, and
turn on the A2D module

; End of port initialisation


; Test
Test
  call ReadPotentiometer
  movlw 0
  movwf Servo1pulse
  movf ADRESH,w
  movwf Servo1pulse+1
  movlw 1
  movwf ServoDelay
  call DoThisNow
  goto Test

ReadPotentiometer
  nop          ; wait 5uS for A2D amp to settle and capacitor to charge.
  nop          ; wait 1uS
  nop          ; wait 1uS
  nop          ; wait 1uS
  nop          ; wait 1uS
  bsf    ADCON0,GO    ; start conversion
  btfsc    ADCON0,GO    ; this bit will change to zero when the conversion is
complete
  goto    $-1
  return

; Correct pulse values at least 1-400, perhaps use 9 bit values?

DoThisNow
  ; Servo 1
  bsf PORTC,0              ; Start pulse servo 1
  movlw 243                ; First wait for 1ms 4*255=1024us, but 243 yields
1.00us
  movwf Pulselength
  call Servopulsewait    ; generate 1ms wait pulse
ServoPulse1 not used
;*********************
  ;**********************
  movf Servo1pulse+1,w   ;Setup second servo control Pulse
  movwf ServoTOnSecond  ;Preserve second Pulse Time
  movwf Pulselength
  call Servopulsewait   ;Generate second servo control Pulse
  bcf PORTC,0              ; End pulse servo 1
  call Dlay5               ; Then wait for at least 20ms
  call Dlay5
  call Dlay5

  ; Delay count
  decfsz ServoDelay,f    ; Wait for the requested number of cycles
```

```
  goto DoThisNow
  return

Servopulsewait
  movf Pulselength,f
  btfsc STATUS,Z
  return
SPWloop
  nop                       ; Make sure we have 4us/value
  decfsz Pulselength,f    ; Start counting
  goto SPWloop    ; Loop
  return                    ; Done!

; Uses t0
Dlay5                           ;  Delay 5 msecs
  movlw  4                      ;  Set up the Delay
  movwf  t0
  movlw  256 - 0x0E8 ;this could be just 18
  addlw  1
  btfsc  STATUS, Z
  decfsz t0,f
  goto $ - 3
  return

Dlay3  ; delay 3ms
  movlw 3
  movwf 10
  movlw 170
  addlw 1
  btfsc STATUS,Z
  decfsz t0, f
  goto $-3
  Return

end
```