

```

LOC OBJECT CODE      LINE SOURCE TEXT
VALUE

00001          LIST      N=83,C=110,B=4,P=16F18313
00002          ERRORLEVEL -302 ;ignore banking messages
00003  title     "----- Servo Control Assembler Program, (C) LWH brainware 2020"
00004  subtitle  "----- Main Program"
00005 ;=====
00006 ;- Servo Control Program
00007 ;- (C) LWH brainware 2020
00008 ;=====
00009 ;- 200531 lwh created
00010 ;-----
00011  #include   "P16F18313.inc"
00001          LIST
00002
00003 ;=====
00004 ; Build date : May 15 2016
00005 ; MPASM PIC16F18313 processor include
00006 ;
00007 ; (c) Copyright 1999-2016 Microchip Technology, All rights reserved
00008 ;=====
00009
02267          LIST
00012 ;#define TESTSIM
00013 ;-----
00014 ;- General Addresses and Registers for PIC16F182x
00015 ;-----
00016 ;- Addresses
00017 ;-----
00000100      00018 CODE_START EQU 0x100 ; start of program space
00000000      00019 RESET_ADR EQU 0x000 ; reset vector
00000004      00020 INTER_ADR EQU 0x004 ; interrupt vector
00021 ; last common RAM address declared so far (0070..007F)
00000070      00022 LASTCOMMON SET 0x70 ; initialize to 0070, updated later
00023 ;-----
00024 ;- Memory registers
00025 ;-----
00000070      00026 TMP EQU LASTCOMMON ; tmp register in common RAM
00000071      00027 TMPH EQU TMP + 1 ; tmp high byte
00000072      00028 SERVOSTATUS EQU TMPH + 1 ; status bits
00000073      00029 LASTCOMMON SET SERVOSTATUS + 1 ; update next available RAM address
00030 #if (LASTCOMMON > H'7F')
00031 error "Running out of common memory!"
00032 #endif
00033 ;-----
00034 ;- Servo status bits
00035 ;-----
00000000      00036 SStatRun EQU 0 ; bit 0: 1 = servo is running
00000001      00037 SStatTeach EQU 1 ; bit 1: 1 = teach mode, 0 = normal
00000002      00038 SStatInfo EQU 2 ; last info state before blinking
00039 ;SStatSetup EQU 7 ; bit 7: 1 = setup ready (mainly PWM)
00040 ;-----
00041 ;- User ID words 0..3
00042 ;-
00043 UserID code 0x8000
8000 0100      00044 dw 0x0100 ; Servo Control
8001 3FFF      00045 dw 0x3FFF ; not used
8002 0000      00046 dw 0x0000 ; board version
8003 0100      00047 dw 0x0100 ; version 1.0
00048 ;-----
00049 ;- Configuration bits
00050 ;- - external oscillator off
00051 ;- - use internal oscillator with 1MHz
00052 ;- - no clock output on pin OSC2
00053 ;- - clock switching by software allowed
00054 ;- - Fail-Safe Clock Monitor is disabled
8007 1FEC      00055 __config _CONFIG1, _FEXTOSC_OFF & _RSTOSC_HFINT1 & _CLKOUTEN_OFF & _CSWEN_O
N & _FCMEN_OFF & H'3FFF'
00056 ;- - debugging off unless managed by debugger tool
00057 ;- - Stack Overflow or Underflow will cause a Reset
00058 ;- - PPS can be used repeatedly
00059 ;- - Brown-out reset voltage is low (2.45V)
00060 ;- - Brown-out will cause a reset
00061 ;- - watchdog timer is off
00062 ;- - power-up timer is on
00063 ;- - RA3 is !MCLR
8008 37F1      00064 __config _CONFIG2, _DEBUG_OFF & _STVREN_ON & _PPS1WAY_OFF & _BORV_LOW & _BO
REN_ON & _WDTE_OFF & _PWRTE_ON & _MCLRE_ON & H'3FFF'
00065 ;- - low voltage programming off

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Main Program

LOC	OBJECT	CODE	LINE	SOURCE	TEXT
	VALUE				

```

00066 ; - write-protection off (for now)
8009 1FFF 00067 __config _CONFIG3, _LVP_OFF & _WRT_OFF & H'3FFF'
00068 ; - EEPROM code protection off
00069 ; - Program code protection off (for now)
800A 3FFF 00070 __config _CONFIG4, _CPD_OFF & _CP_OFF & H'3FFF'
00071
00072     ORG RESET_ADR                ; reset vector
00073 ;-----
00074 ; - Program
00075 ;-----
0000 2??? 00076     GOTO    init                ; start with initialization
00077
00078     ORG INTER_ADR                ; interrupt vector
00079 ;-----
00080 ; - Interrupt handling
00081 ;-----
0004 00082 intINT:    ; check general interrupt
0004 1010 00083     BCF     PIR0, INTF            ; clear interrupt (not used here)
00084 ;-----
0005 00085 intTMR2:    ; check timer 2 interrupt (PWM update)
0005 0020 00086     banksel PIR1
0006 1C91 00087     BTFSS  PIR1, TMR2IF          ; check if timer2 interrupt set
0007 2??? 00088     GOTO    intButton           ; if not, check next
00089     ; handle timer2 interrupt
0008 1091 00090     BCF     PIR1, TMR2IF          ; first, clear interrupt
0009 2??? 00091     CALL    onUpdate
00092 ;-----
000A 00093 intButton:  ; check button state changed
000A 0020 00094     banksel PIR0
000B 1E10 00095     BTFSS  PIR0, IOCIF          ; check if interrupt on change is set
000C 2??? 00096     GOTO    intTmr0            ; if not, check next
00097     ; handle button interrupts
000D 1210 00098     BCF     PIR0, IOCIF          ; first, clear interrupt flag
000E 2??? 00099     CALL    onButton
00100 ;-----
000F 00101 intTmr0:    ; check timer 0 interrupt (button debouncing and downtime counter)
000F 0020 00102     banksel PIR0
0010 1E90 00103     BTFSS  PIR0, TMR0IF          ; check if timer0 interrupt is set
0011 2??? 00104     GOTO    intTmr1            ; if not, check next
00105     ; handle timer0 interrupts
0012 1290 00106     BCF     PIR0, TMR0IF          ; first, clear interrupt flag
0013 2??? 00107     CALL    onButtonPoll
00108 ;-----
0014 00109 intTmr1:    ; check timer 1 interrupt (blink)
0014 0020 00110     banksel PIR1
0015 1C11 00111     BTFSS  PIR1, TMR1IF          ; check if timer1 interrupt is set
0016 2??? 00112     GOTO    intDone            ; if not, check next
00113     ; handle timer1 interrupts
0017 1011 00114     BCF     PIR1, TMR1IF          ; first, clear interrupt flag
0018 2??? 00115     CALL    onBlink
00116 ;-----
0019 00117 intDone:    ; all interrupts checked and handled
0019 0009 00118     RETFIE                       ; return from interrupt
00119
00120     ORG CODE_START
00121 ;-----
00122 ; - Controller Initialization
00123 ;-----
00124 ; - internal oscillator setup to 4MHz
0100 00125 init:
00126 ;     BCF     SERVOSTATUS, SStatSetup ; starting setup
0100 0032 00127     banksel OSCCON1
0101 3060 00128     MOVLW  B'01100000'
00129     ;     x|||++++---    postscaler set to 1:1
00130     ;     +++-----    HF internal oscillator, 1MHz (just in case)
0102 0099 00131     MOVWF  OSCCON1
0103 3003 00132     MOVLW  B'00000111'
00133     ;     xxx++++---    HFINTOSC set to 4MHZ
0104 009F 00134     MOVWF  OSCFRQ
0105 3000 00135     MOVLW  0x00
0106 009E 00136     MOVWF  OSCTUNE                ; fix fix osctune bug (from the errata doc)
0107 171D 00137     BSF     OSCEN, HFOEN            ; enable oscillator
00138     ; now there may be a 2µs delay until oscillator has changed
00139     ; wait until oscillator is ready
00140 #ifndef TESTSIM
00141     BTFSS  OSCSTAT1, HFOR          ; is HFINTOSC ready?
00142     GOTO    $-1                    ; if not, test again

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Main Program

LOC	OBJECT	CODE	LINE	SOURCE	TEXT
	VALUE				

```

00143 #endif
0108 0021 00144 banksel PIE0
0109 1010 00145 BCF PIE0, INTE ; disable INT pin
00146 ;- initialize and start PWM, buttons and info return output
010A 2??? 00147 CALL setupInfoPin
010B 2??? 00148 CALL setupButtons
00149 ; if teach button pressed (0), reset EEPROM to factory settings
010C 0020 00150 banksel PORTA
010D 1C0C 00151 BTFSS PORTA, BUTTONTCH
010E 2??? 00152 CALL eepromReset
010F 2??? 00153 CALL setupPWM
00154 ;- lock peripheral setup (?)
0110 138B 00155 BCF INTCON, GIE ; disable interrupts
0111 003C 00156 banksel PPSLOCK
00157 ; required sequence:
0112 3055 00158 MOVLW 0x55
0113 008F 00159 MOVWF PPSLOCK
0114 30AA 00160 MOVLW 0xAA
0115 008F 00161 MOVWF PPSLOCK
0116 140F 00162 BSF PPSLOCK, PPSLOCKED ; disable further writes
0117 0020 00163 banksel INTCON
0118 178B 00164 BSF INTCON, GIE ; reenale interrupts
0119 170B 00165 BSF INTCON, PEIE ; enable peripheral interrupts
00166 #ifdef TESTSIM
00167 ; call eepromTest
00168 #endif
00169 ; CALL testPWM
011A 1872 00170 BTFSC SERVOSTATUS, SStatRun
011B 2??? 00171 GOTO $-1 ; wait until motor has settled
011C 2??? 00172 CALL setBlinkFast
011D 3002 00173 MOVLW .2
00174 ; if teach button pressed (0), blink 5 times to indicate reset
011E 0020 00175 banksel PORTA
011F 1C0C 00176 BTFSS PORTA, BUTTONTCH
0120 3005 00177 MOVLW .5
0121 2??? 00178 CALL pinInfoBlink ; blink 2 times
00179 ;-----
0122 0000 00180 idle: NOP
00181 ;TODO everything else
0123 2??? 00182 GOTO idle
00183 ;-----
00184 ;- End main program
00185 ;-----
00186 include "Servo_PWM.inc"
00001 subtitle "----- PWM Section"

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- PWM Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00002 PAGE
00003 ;=====
00004 ;- Servo PWM include file
00005 ;- (C) LWH brainware 2020
00006 ;=====
00007 ;- 20-05-31 lwh created
00008 ;-----
00009 ;- PWM duty cycle values are always stored as 10 bits in 2 Bytes,
00010 ;- where positive values are forward, negative backwards.
00011 ;- bits 14..10 are ignored: Sxxxxx98 76543210
00012 ;- PWM frequency should be something around 50Hz, with a DC of 3..13%
00013 ;- Here: Frequency is 61Hz with DC setting of 30...170 -> only 1byte req.
00014 ;-
00015 ;- from top of servo motor:
00016 ;- min = CLW 0 CCW = max
00017 ;- <->
00018 ;- left center right
00019 ;-----
00020 ;- memory registers
00021 ;-----
00022 ;- Common RAM
00023 TARGPMDCL EQU LASTCOMMON ; target PWM duty cycle, LSB (7..0)
00024 LASTPMDCL EQU TARGPMDCL + 1 ; last PWM duty cycle, LSB (7...0)
00025 HOLDCTR EQU LASTPMDCL + 1 ; count down until DC off
00026 LASTCOMMON SET HOLDCTR + 1 ; update next available RAM address
00027 #if (LASTCOMMON > H'7F')
00028 error "Running out of common memory!"
00029 #endif
00030 ;-----
00031 ;- Constants
00032 #define PWMDCmin .30
00033 #define PWMDCmax .170
00034 #define PWMDCcen .94 ; about 1.5ms at 61Hz. fix, not editable
00035 #define UpdDefault .1 ; DC update speed for 4s per full motion
00036 #define UpdSlow .6 ; DC update speed for 15s per full motion
00037 #define Countdown .10 ; number of updates until DC is switched off
00038 #define PWMPort .2 ; use port A2 for PWM output
00039 ;-----
00040 ;- Configure/Remap ports:
00041 ;- RA2 is PWM5 output
0124 0042 setupPWM:
0124 3003 00043 MOVLW LOW EPWMTARG
0125 00F9 00044 MOVWF EEPROMADR
0126 2??? 00045 CALL eepromRead
0127 00F3 00046 MOVWF TARGPMDCL ; set last stored target DC
0128 1472 00047 BSF SERVOSTATUS, SStatRun ; set status to running
00048 ; set output port
0129 0020 00049 banksel PORTA
012A 110C 00050 BCF PORTA, PWMPort ; clear output
012B 0022 00051 banksel LATA
012C 110C 00052 BCF LATA, PWMPort ; clear output latch
012D 0023 00053 banksel ANSELA
012E 110C 00054 BCF ANSELA, PWMPort ; set as digital i/o
012F 0021 00055 banksel TRISA
0130 150C 00056 BSF TRISA, PWMPort ; disable output driver (1)
00057 ; configure pwm
0131 002C 00058 banksel PWM5CON
0132 1399 00059 BCF PWM5CON, PWM5EN ; disable PWM during setup
0133 1219 00060 BCF PWM5CON, PWM5POL ; set output polarity to high
0134 0020 00061 banksel PR2
0135 30FF 00062 MOVLW 0xFF
0136 009E 00063 MOVWF PR2 ; set PWM period
0137 3002 00064 MOVLW LOW EPWMCUR
0138 00F9 00065 MOVWF EEPROMADR
0139 2??? 00066 CALL eepromRead ; get current (last stored) DC
013A 2??? 00067 CALL setCurrentDC
013B 1091 00068 BCF PIR1, TMR2IF ; clear timer interrupt
013C 0020 00069 banksel T2CON
013D 3007 00070 MOVLW B'0000111'
00071 ; x|||||+--- prescaler 1:64
00072 ; |||+----- timer2 on
00073 ; ++++----- no postscaler for now
013E 009F 00074 MOVWF T2CON
013F 0020 00075 banksel PIR1
0140 1C91 00076 BTFSS PIR1, TMR2IF ; wait until timer interrupt flag
0141 2??? 00077 GOTO $-1
0142 0021 00078 banksel TRISA

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- PWM Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

0143 110C      00079      BCF      TRISA, PWMPort      ; enable output driver (0)
                                00080      ; map pwm output
0144 003D      00081      banksel  RA2PPS
0145 3002      00082      MOVLW   B'0000010'
                                00083      ;          xxx+++++--- RA2 source is PWM5
0146 0092      00084      MOVWF   RA2PPS
0147 002C      00085      banksel  PWM5CON
0148 1799      00086      BSF     PWM5CON, PWM5EN      ; enable PWM after setup
0149 3001      00087      MOVLW   UpdDefault          ; set default speed
                                00088      ;-----
                                00089      ;- sets the pwm update speed to the value in W:
                                00090      ;- approx (W+1)*2sec per full motion, min W=0, max W=15
                                00091      ;- enables timer2 interrupts
014A      00092      setSpeed:
014A 0021      00093      banksel  PIE1
014B 1091      00094      BCF     PIE1, TMR2IE        ; disable interrupt for now
014C 00F0      00095      MOVWF   TMP                 ; keep the speed
014D 35F0      00096      LSLF   TMP, F
014E 35F0      00097      LSLF   TMP, F
014F 35F0      00098      LSLF   TMP, F                ; and move it to the proper bit position
0150 0020      00099      banksel  T2CON
0151 081F      00100      MOVF   T2CON, W             ; get current setting
0152 3907      00101      ANDLW  B'00000111'         ; clear postscaler setting
0153 0470      00102      IORWF  TMP, W               ; apply new setting
0154 009F      00103      MOVWF  T2CON                ; and write back
                                00104      ; enable interrupt
0155      00105      enableUpd:
0155 0020      00106      banksel  PIR1
0156 1091      00107      BCF     PIR1, TMR2IF        ; clear any lingering interrupt
0157 0021      00108      banksel  PIE1
0158 1491      00109      BSF     PIE1, TMR2IE        ; enable interrupt
0159 0008      00110      RETURN
                                00111      ;-----
                                00112      ;- sets the new target DC to the value in W
015A      00113      setTarget:
015A 00F3      00114      MOVWF  TARGPMDCL
                                00115      ; store to eeprom
015B 3003      00116      MOVLW  LOW EPWMTARG
015C 00F9      00117      MOVWF  EEPROMADR
015D 0873      00118      MOVF   TARGPMDCL, W
015E 2???      00119      CALL   eepromWrite
                                00120      ; reactivate current DC
015F 3002      00121      MOVLW  LOW EPWMCUR
0160 00F9      00122      MOVWF  EEPROMADR
0161 2???      00123      CALL   eepromRead          ; get current (last stored) DC
0162 2???      00124      CALL   setCurrentDC
0163 1472      00125      BSF     SERVOSTATUS, SStatRun ; set status to running
0164 30FF      00126      MOVLW  .255
0165 2???      00127      CALL   pinInfoBlink        ; blink until target reached (max 32s)
0166 2???      00128      GOTO   enableUpd           ; just in case
                                00129      ;-----
                                00130      ;- sets the current DC to the value in W
0167      00131      ;- 20-06-14 lwh created and tested
0167      00132      setCurrentDC:
0167 00F4      00133      MOVWF  LASTPMDCL           ; keep value
0168 002C      00134      banksel  PWM5DCH
0169 00F1      00135      MOVWF  TMPH
016A 01F0      00136      CLRF   TMP                 ; TMP = 76543210 -----
016B 1003      00137      BCF     STATUS, C           ; clear carry
016C 0CF1      00138      RRF    TMPH, F              ;
016D 0CF0      00139      RRF    TMP, F               ; TMP = -7654321 0-----
016E 0CF1      00140      RRF    TMPH, F              ;
016F 0CF0      00141      RRF    TMP, F               ; TMP = --765432 10-----
0170 0871      00142      MOVF   TMPH, W
0171 0098      00143      MOVWF  PWM5DCH
0172 0870      00144      MOVF   TMP, W
0173 0097      00145      MOVWF  PWM5DCL
0174 300A      00146      MOVLW  Countdown
0175 00F5      00147      MOVWF  HOLDCTR             ; reset countdown
0176 0008      00148      RETURN
                                00149      ;-----
                                00150      ;- gets the current DC from the PWM register into W
0177      00151      ;- 20-06-14 lwh created and tested
0177      00152      getCurrentDC:
0177 002C      00153      banksel  PWM5DCH
0178 0818      00154      MOVF   PWM5DCH, W
0179 00F1      00155      MOVWF  TMPH

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- PWM Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

017A 0817      00156      MOVF    PWM5DCL, W
017B 00F0      00157      MOVWF   TMP                ; TMP = --765432 10-----
017C 0DF0      00158      RLF    TMP, F
017D 0DF1      00159      RLF    TMPH, F            ; TMP = -7654321 0-----C
017E 0DF0      00160      RLF    TMP, F
017F 0D71      00161      RLF    TMPH, W            ; TMP = 76543210 -----CC
0180 00F4      00162      MOVWF  LASTPWMDCCL        ; keep value
0181 0008      00163      RETURN
00164
00165 ;-----
00166 ;- updates the duty cycle to the next pulse width.
00167 ;- called on timer2 interrupt
0182      00168 onUpdate:
0182 2???      00169      CALL   getCurrentDC        ; get current DC
0183 0809      00170      MOVF   WREG, W            ; set zero status
0184 1903      00171      BTFSC  STATUS, Z          ; if 0, we are already finished
0185 0008      00172      RETURN
0186 0874      00173      MOVF   LASTPWMDCCL, W
0187 0273      00174      SUBWF  TARGPWMDCCL, W     ; check direction
0188 1903      00175      BTFSC  STATUS, Z          ; check if equal
0189 2???      00176      GOTO   targReached
018A 1C03      00177      BTFSS  STATUS, C          ; check direction
018B 2???      00178      GOTO   down                ; current > target
00179
00180 ; - - - - -
00181 ; current < target, so increment
018C 0A74      00181      INCF   LASTPWMDCCL, W
018D 2???      00182      CALL   setCurrentDC
018E 0008      00183      RETURN
00184
00185 ; - - - - -
00186 ; current > target, so decrement
018F 0374      00186 down:  DECF   LASTPWMDCCL, W
0190 2???      00187      CALL   setCurrentDC
0191 0008      00188      RETURN
00189
00190 ; - - - - -
00191 ; we are done, so store the current DC and, if countdown done, switch off
0192      00191 targReached:
0192 2???      00192      CALL   stopBlink          ; stop blinking: target reached
00193
00194 ; store current duty cycle
0193 3002      00194      MOVLW  LOW EPWMCUR
0194 00F9      00195      MOVWF  EEPROMADR
0195 0873      00196      MOVF   TARGPWMDCCL, W
0196 2???      00197      CALL   eepromWrite
0197 2???      00198      CALL   eepromWait        ; wait until written
0198 0BF5      00199      DECFSZ HOLDCTR, F        ; check countdown
0199 0008      00200      RETURN                    ; if not expired, leave DC as is
019A      00201 finishPWM:
00202
00203 ; if expired, stop PWM by setting DC to 0 and set info output
019A 002C      00203      banksel PWM5DCL
019B 0197      00204      CLR    PWM5DCL
019C 0198      00205      CLR    PWM5DCH            ; clear duty cycle
019D 1072      00206      BCF    SERVOSTATUS, SStatRun ; clear status from running
00207
00208 ; check current DC against center DC
019E 0873      00208      MOVF   TARGPWMDCCL, W     ; target (current) DC in W
019F 3C5E      00209      SUBLW  PWMDCcen            ; center - current
01A0 1C03      00210      BTFSS  STATUS, C
01A1 2???      00211      GOTO   pinInfoOn          ; if current > center, report H
01A2 2???      00212      GOTO   pinInfoOff         ; else report L
00213 ; RETURN ; just in case
00214 ;-----
00215 ;- when stopping the motion (e.g. by button), store the current DC
01A3      00216 stopPWM:
01A3 3002      00217      MOVLW  LOW EPWMCUR
01A4 00F9      00218      MOVWF  EEPROMADR
01A5 2???      00219      CALL   getCurrentDC
01A6 2???      00220      CALL   eepromWrite
01A7 2???      00221      CALL   finishPWM          ; and do the actual stop
01A8 2???      00222      CALL   eepromWait        ; wait until written
01A9 0008      00223      RETURN
00224 ;-----
00225 ;- test the PWM (and the servo) by moving to and fro (max, not stored!)
00226 ;- 20-05-16 tested, works. limits for some motors too wide
01AA      00227 testPWM:
01AA 08F5      00228      MOVF   HOLDCTR, F
01AB 1D03      00229      BTFSS  STATUS, Z          ; check countdown
01AC 2???      00230      GOTO   $-2                ; if not expired, wait
00231
00232 ; - - - - -
00233 ; countdown expired, so position is reached

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- PWM Section

```

LOC OBJECT CODE      LINE SOURCE TEXT
VALUE
01AD  30AA           00233      MOVLW  PWMDCmax
01AE  2???           00234      CALL   setTarget
01AF  08F5           00235      MOVF   HOLDCTR, F
01B0  1D03           00236      BTFSS  STATUS, Z           ; check countdown
01B1  2???           00237      GOTO   $-2                 ; if not expired, wait
                                00238      ; -----
                                00239      ; countdown expired, so position is reached
01B2  301E           00240      MOVLW  PWMDCmin
01B3  2???           00241      CALL   setTarget
01B4  2???           00242      GOTO   testPWM
                                00243      ;=====
                                00187      include "Servo_Buttons.inc"
                                00001      subtitle  "----- Button Section"
    
```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Button Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00002 PAGE
00003 ;=====
00004 ;- Servo Buttons include file
00005 ;- (C) LWH brainware 2020
00006 ;=====
00007 ;- 20-06-10 lwh created
00008 ;- 20-06-23 LWH functions improved, short press introduced
00009 ;-----
00010 ;- buttons are used for teaching and final action.
00011 ;- left corresponds to lowest DC, right to highest
00012 ;- Buttons are active 0 and can be pressed short, normal and long
00013 ;-
00014 ;- See Servo_pic_pinout.ods:Keypress for function matrix
00015 ;-
00016 ;-
00017 ;- button center (teach only): short: go to center long: enter teach
00018 ;- button left and right: short: go to end
00019 ;- when in teach short: increase/decrease DC
00020 ;- long: save current position and exit teach
00021 ;- all buttons are debounced by timer:
00022 ;- left and right if not in teach:
00023 ;- act after stable (some timer cycles) to allow control by switch
00024 ;- center button and others in teach:
00025 ;- act after release, then decide whether short or long keypress.
00026 ;-
00027 ;- button recognition is a combination of interrupts (IOC) and polling:
00028 ;- When a button is pressed, an interrupt on the negative flank is thrown:
00029 ;-
00030 ;- This starts a timer. When the timer fires,
00031 ;- if the button is still pressed if the button was released (or a glitch)
00032 ;- - the IOC is changed to rising - the timer is stopped
00033 ;- - a time counter is incremented
00034 ;- When a button is released and the interrupt on positive flank is thrown:
00035 ;- - the timer is stopped
00036 ;- - the IOC is changed to falling to act on the next keypress
00037 ;- - depending on the time the key was pressed and the teach state, the
00038 ;- keypress is evaluated
00039 ;-----
00040 ;- memory registers
00041 ;-----
00042 ;- Common RAM
00000076 00043 BUTTONCTR EQU LASTCOMMON ; target PWM duty cycle, LSB (7..0)
00000077 00044 LASTBUTTONS EQU BUTTONCTR + 1 ; last button state
00000078 00045 LASTCOMMON SET LASTBUTTONS + 1 ; update next available RAM address
00046 #if (LASTCOMMON > H'7F')
00047 error "Running out of common memory!"
00048 #endif
00049 ;-----
00050 ;- Constants
00000005 00051 MINCOUNTS EQU .5 ; minimum number of 20ms intervals for button
00000032 00052 SHORTCOUNTS EQU .50 ; number of 20ms intervals between short and normal
00000096 00053 LONGCOUNTS EQU .150 ; number of 20ms intervals to count as long
0000009B 00054 TMR1PERIOD EQU .155 ; for a timer period of approx 20ms
00000031 00055 BUTTONMASK EQU B'00110001' ; 1s for button inputs
FFFFFFCE 00056 BUTTONNMASK EQU ~ BUTTONMASK ; 0s to clear button pins
00000004 00057 BUTTONLFT EQU .4 ; port left button (cclockwise, lowest DC)
00000005 00058 BUTTONRGT EQU .5 ; port right button (clockwise, highest DC)
00000000 00059 BUTTONTCH EQU .0 ; port teach button
00060 ;-----
00061 ;- Configure/Remap ports:
00062 ;- left: RA4, right: RA5, teach: RA0
01B5 00063 setupButtons:
01B5 10F2 00064 BCF SERVOSTATUS, SStatTeach ; clear teach mode (just in case)
01B6 3031 00065 MOVLW BUTTONMASK ; flag button inputs as such
01B7 0021 00066 banksel TRISA
01B8 048C 00067 IORWF TRISA, F ; set pins to input
01B9 0024 00068 banksel WPUA
01BA 048C 00069 IORWF WPUA, F ; enable pull-ups
01BB 30CE 00070 MOVLW BUTTONNMASK
01BC 0023 00071 banksel ANSELA
01BD 058C 00072 ANDWF ANSELA, F ; clear analog bits for digital inputs
00073 ; set timer0 to poll buttons every 5 ms
01BE 0020 00074 banksel T0CON0
01BF 3000 00075 MOVLW B'00000000'
00076 ; |xr|++++--- postscaler 1:1
00077 ; | +----- 8 bit timer only

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Button Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00078 ; +----- timer disabled for now
01C0 0097 00079 MOVWF T0CON0
01C1 0020 00080 banksel T0CON1
01C2 3057 00081 MOVLW B'01010111'
00082 ; |||++++--- prescaler 1:128 for period * 128us
00083 ; |||+----- no synchronization (no external clock)
00084 ; +++----- clock source command clock (Fosc/4)
01C3 0098 00085 MOVWF T0CON1
01C4 0020 00086 banksel TMR0H
01C5 309B 00087 MOVLW TMR1PERIOD
01C6 0096 00088 MOVWF TMR0H ; set period
00089 ; enable button down and timer interrupt
01C7 2??? 00090 CALL enableBnDnInt
01C8 00091 enableBnInt:
01C8 0020 00092 banksel PIR0
01C9 1290 00093 BCF PIR0, TMR0IF ; clear any lingering interrupts
01CA 0021 00094 banksel PIE0
01CB 1690 00095 BSF PIE0, TMR0IE
01CC 0008 00096 RETURN
00097 ;-----
00098 ;- enable interrupt on button down (negative flank)
00099 ;- make sure to enable peripheral interrupts in main program
01CD 00100 enableBnDnInt:
01CD 3031 00101 MOVLW BUTTONMASK ; flag button inputs as such
01CE 0027 00102 banksel IOCAP
01CF 0492 00103 IORWF IOCAN, F ; set interrupt on negative flank
01D0 30CE 00104 MOVLW BUTTONNMASK
01D1 0591 00105 ANDWF IOCAP, F ; clear interrupt on positive flank
01D2 0021 00106 banksel PIE0
01D3 1610 00107 BSF PIE0, IOCIE ; enable interrupt on change
01D4 0008 00108 RETURN
00109 ;-----
00110 ;- enable interrupt on button up (positive flank)
00111 ;- make sure to enable peripheral interrupts in main program
01D5 00112 enableBnUpInt:
01D5 3031 00113 MOVLW BUTTONMASK ; flag button inputs as such
01D6 0027 00114 banksel IOCAP
01D7 0491 00115 IORWF IOCAP, F ; set interrupt on positive flank
01D8 30CE 00116 MOVLW BUTTONNMASK
01D9 0592 00117 ANDWF IOCAN, F ; clear interrupt on negative flank
01DA 0021 00118 banksel PIE0
01DB 1610 00119 BSF PIE0, IOCIE ; enable interrupt on change
01DC 0008 00120 RETURN
00121 ;-----
00122 ;- enter teach mode: slower, full limits, long and short keypress
01DD 00123 enterTeachMode:
01DD 14F2 00124 BSF SERVOSTATUS, SStatTeach ; set teach mode
01DE 3006 00125 MOVLW UpdSlow
01DF 2??? 00126 CALL setSpeed ; set slow speed
01E0 3002 00127 MOVLW 2
01E1 2??? 00128 CALL pinInfoBlink ; blink 2 times for feedback
01E2 0008 00129 RETURN
00130 ;-----
00131 ;- leave teach mode: normal speed, stored limits
01E3 00132 leaveTeachMode:
01E3 3001 00133 MOVLW UpdDefault
01E4 2??? 00134 CALL setSpeed ; set normal speed
01E5 10F2 00135 BCF SERVOSTATUS, SStatTeach ; clear teach mode
01E6 3003 00136 MOVLW 3
01E7 2??? 00137 CALL pinInfoBlink ; blink 3 times for feedback
01E8 0008 00138 RETURN
00139 ;-----
00140 ;- react on button changed interrupt
01E9 00141 onButton:
00142 ;Check whether it is down or up (all flanks are set similar)
01E9 0027 00143 banksel IOCAP
01EA 1A11 00144 BTFSC IOCAP, BUTTONLFT ; is positive flank requested?
01EB 2??? 00145 GOTO onButtonUp ; if yes, a button was released
00146 ; - - - - -
00147 ; if no, a button was pressed
01EC 00148 onButtonDown:
01EC 0027 00149 banksel IOCAF
01ED 0913 00150 COMF IOCAF, W ; get inverted IOC source bits (0=int)
00151 ; ANDLW BUTTONMASK
01EE 00F7 00152 MOVWF LASTBUTTONS ; remember interrupt source (0 = pressed)
01EF 0193 00153 CLRf IOCAF ; reset IOC bits
00154 ; (re)start the timer

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Button Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

01F0 01F6      00155      CLRFBUTTONCTR      ; (re)start button down time counter
01F1 0020      00156      bankselTMR0H
01F2 309B      00157      MOVLWTMR1PERIOD
01F3 0096      00158      MOVWFTMR0H      ; (re)set period
01F4 0020      00159      bankselT0CON0
01F5 1797      00160      BSFT0CON0, T0EN      ; (re)start timer
01F6 0008      00161      RETURN
00162      ; -----
00163      ; a button was released
01F7      00164      onButtonUp:
01F7 0027      00165      bankselIOCAF
01F8 0193      00166      CLRFIGIOCAF      ; reset IOC bits
01F9 2???      00167      CALLenableBnDnInt      ; set for bn down
01FA 0020      00168      bankselT0CON0
01FB 1397      00169      BCF T0CON0, T0EN      ; stop timer
00170      ; check button down time
01FC 3005      00171      MOVLWMINCOUNTS
01FD 0276      00172      SUBWFBUTTONCTR, W      ; check for min down time (debounce)
01FE 1C03      00173      BTFSSSTATUS, C
01FF 0008      00174      RETURN      ; if C=0, counter < W, bouncing
00175      ; -----
00176      ; else bouncing finished, pin stable for some ms
00177      ; should only occur when in teach mode or teach button pressed
00178      ; check key down duration
0200 3096      00179      MOVLWLONGCOUNTS
0201 0276      00180      SUBWFBUTTONCTR, W      ; check for long down time
0202 1803      00181      BTFSCSTATUS, C
0203 2???      00182      GOTOlongButton      ; if C=1, counter > W, long time
00183      ; else check further
0204 3032      00184      MOVLWSHORTCOUNTS
0205 0276      00185      SUBWFBUTTONCTR, W      ; check for medium down time
0206 1803      00186      BTFSCSTATUS, C
0207 2???      00187      GOTOmediumButton      ; if C=1, counter > W, medium time
00188      ; else short time
0208 2???      00189      GOTOshortButton
0209 0008      00190      RETURN      ; just in case
00191      ; -----
00192      ;- react on button timer interrupt, which occurs first 20ms after downward
00193      ;- flank, then each 20ms
020A      00194      onButtonPoll:
020A 0020      00195      bankselPORTA
020B 090C      00196      COMFPORTA, W      ; get inverted port status, 1 is btn pressed
020C 3931      00197      ANDLWBUTTONMASK      ; clear unwanted bits
020D 1903      00198      BTFSCSTATUS, Z      ; check if 0
020E 2???      00199      GOTOnoBnPressed      ; if 0, no button currently pressed
00200      ; else accept button and wait until released
020F 080C      00201      MOVFPORTA, W      ; get port status again (not inverted)
0210 3931      00202      ANDLWBUTTONMASK      ; clear unwanted bits
0211 00F7      00203      MOVWFLASTBUTTONS      ; keep button status: 0 = pressed
00204      ; increment counter
0212 0AF6      00205      INCFBUTTONCTR, F      ; increment time counter
0213 1903      00206      BTFSCSTATUS, Z      ; check if 0
0214 03F6      00207      DECFBUTTONCTR, F      ; if 0 (overflow), set back to useful value
00208      ; check if just became stable
0215 3005      00209      MOVLWMINCOUNTS
0216 0276      00210      SUBWFBUTTONCTR, W
0217 1D03      00211      BTFSSSTATUS, Z
0218 0008      00212      RETURN      ; if not, that's it
00213      ; -----
00214      ; if yes, a button was just pressed, so act accordingly.
00215      ; first check whether in teach mode or not
0219 18F2      00216      BTFSCSERVOSTATUS, SStatTeach
021A 2???      00217      GOTObuttonTeaching      ; if in teach, act further
00218      ; if not in teach, check whether teach button was pressed (0!)
021B 1C77      00219      BTFSSLASTBUTTONS, BUTTONTCH
021C 2???      00220      GOTObuttonTeaching      ; if yes, act further
00221      ; if direction button outside teach, just move
021D 0020      00222      bankselT0CON0
021E 1397      00223      BCF T0CON0, T0EN      ; stop timer
021F 1E77      00224      BTFSSLASTBUTTONS, BUTTONLFT
0220 2???      00225      GOTOmediumLeft      ; if left button, move left
0221 2???      00226      GOTOmediumRight      ; else move right
00227      ; -----
00228      ;- we are in teach mode or maybe entering
0222      00229      buttonTeaching:
0222 1872      00230      BTFSCSERVOSTATUS, SStatRun
0223 2???      00231      CALLstopPWM      ; if running, stop

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Button Section

LOC	OBJECT	CODE	LINE	SOURCE	TEXT
				VALUE	

```

0224 2???      00232      CALL    enableBnUpInt      ; set for interrupt when button released
0225 0008      00233      RETURN
                                00234      ;-----
                                00235      ; if no button pressed (or was released since), wait for next
0226          00236  noBnPressed:
0226 0020      00237      banksel T0CON0
0227 1397      00238      BCF    T0CON0, T0EN      ; stop timer
0228 2???      00239      CALL    enableBnDnInt     ; set for interrupt when button pressed
0229 0008      00240      RETURN
                                00241      ;-----
                                00242      ;- react on short button press
022A          00243  shortButton:
022A 1C77      00244      BTFSS  LASTBUTTONS, BUTTONTCH ; has teach been pressed short?
022B 2???      00245      GOTO   shortTeach
022C 1E77      00246      BTFSS  LASTBUTTONS, BUTTONLFT ; has left been pressed short?
022D 2???      00247      GOTO   shortLeft
022E 1EF7      00248      BTFSS  LASTBUTTONS, BUTTONRGT ; has right been pressed short?
022F 2???      00249      GOTO   shortRight
0230 0008      00250      RETURN      ; no button, probably glitch?
                                00251      ;-----
                                00252      ; teach button has been pressed
                                00253      ; move to center position
0231          00254  shortTeach:
0231          00255  mediumTeach:
0231 305E      00256      MOVLW  PWMDCcen
0232 2???      00257      GOTO   setTarget      ; move to center position
0233 0008      00258      RETURN
                                00259      ;-----
                                00260      ; left button has been pressed short
                                00261      ; move one step left
0234          00262  shortLeft:
0234 3002      00263      MOVLW  LOW EPWMCUR
0235 00F9      00264      MOVWF  EEPROMADR
0236 2???      00265      CALL    eepromRead      ; get last stored DC (from stop)
0237 0309      00266      DECF   WREG, W
0238 2???      00267      GOTO   setTarget      ; move to DC - 1
0239 0008      00268      RETURN
                                00269      ;-----
                                00270      ; right button has been pressed short
                                00271      ; move one step right
023A          00272  shortRight:
023A 3002      00273      MOVLW  LOW EPWMCUR
023B 00F9      00274      MOVWF  EEPROMADR
023C 2???      00275      CALL    eepromRead      ; get last stored DC (from stop)
023D 0A09      00276      INCF   WREG, W
023E 2???      00277      GOTO   setTarget      ; move to DC + 1
023F 0008      00278      RETURN
                                00279      ;-----
                                00280      ;- react on medium button press
                                00281      ;- normally moving
0240          00282  mediumButton:
0240 1C77      00283      BTFSS  LASTBUTTONS, BUTTONTCH ; has teach been pressed long?
0241 2???      00284      GOTO   mediumTeach
0242 1E77      00285      BTFSS  LASTBUTTONS, BUTTONLFT ; has left been pressed long?
0243 2???      00286      GOTO   mediumLeft
0244 1EF7      00287      BTFSS  LASTBUTTONS, BUTTONRGT ; has right been pressed long?
0245 2???      00288      GOTO   mediumRight
0246 0008      00289      RETURN      ; no button, probably glitch?
                                00290      ;-----
                                00291      ; left button has been pressed moderately
                                00292      ; move towards left (min) position, dependent on teach or no
0247          00293  mediumLeft:
0247 3000      00294      MOVLW  LOW EPWMCLW
0248 00F9      00295      MOVWF  EEPROMADR
0249 2???      00296      CALL    eepromRead      ; get min DC from eeprom
024A 18F2      00297      BTFSC  SERVOSTATUS, SStatTeach
024B 301E      00298      MOVLW  PWMDCmin      ; if in teach, use minimum
024C 2???      00299      GOTO   setTarget
024D 0008      00300      RETURN
                                00301      ;-----
                                00302      ; right button has been pressed moderately
                                00303      ; move towards right (max) position, dependent on teach or no
024E          00304  mediumRight:
024E 3001      00305      MOVLW  LOW EPWMCCW
024F 00F9      00306      MOVWF  EEPROMADR
0250 2???      00307      CALL    eepromRead      ; get max DC from eeprom
0251 18F2      00308      BTFSC  SERVOSTATUS, SStatTeach

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Button Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

0252 30AA      00309      MOVLW    PWMDCmax          ; if in teach, use maximum
0253 2???      00310      GOTO     setTarget
0254 0008      00311      RETURN
00312 ;-----
00313 ;- react on long button press
0255          00314 longButton:
0255 1C77      00315      BTFSS   LASTBUTTONS, BUTTONTCH ; has teach been pressed long?
0256 2???      00316      GOTO     longTeach
0257 1E77      00317      BTFSS   LASTBUTTONS, BUTTONLFT ; has left been pressed long?
0258 2???      00318      GOTO     longLeft
0259 1EF7      00319      BTFSS   LASTBUTTONS, BUTTONRGT ; has right been pressed long?
025A 2???      00320      GOTO     longRight
025B 0008      00321      RETURN          ; no button, probably glitch?
00322 ;-----
00323 ; teach button has been pressed long
00324 ; enter/leave teach mode
025C          00325 longTeach:
025C 1CF2      00326      BTFSS   SERVOSTATUS, SStatTeach ; in teach mode?
025D 2???      00327      GOTO     enterTeachMode          ; if not, enter teach mode
00328          ; if yes, leave teach mode
025E 2???      00329      GOTO     leaveTeachMode
025F 0008      00330      RETURN
00331 ;-----
00332 ; left button has been pressed long
00333 ; if in teach, store left position, leave teach mode, speed to normal
00334 ; if not in teach?
0260          00335 longLeft:
0260 18F2      00336      BTFSC   SERVOSTATUS, SStatTeach
0261 2???      00337      GOTO     storeLeft          ; if in teach mode, store and leave
0262 0008      00338      RETURN          ; if not in teach, nothing to do yet
0263          00339 storeLeft:
0263 3002      00340      MOVLW   LOW EPWMCUR
0264 00F9      00341      MOVWF   EEPROMADR
0265 2???      00342      CALL    eepromRead          ; get last stored DC (from stop)
0266 00F4      00343      MOVWF   LASTPMDCL
0267 3000      00344      MOVLW   LOW EPWMCLW          ; set eeprom address for lowest DC
0268 2???      00345      GOTO     storeAndExit
00346 ;-----
00347 ; right button has been pressed long
00348 ; if in teach, store right position, leave teach mode, speed to normal
00349 ; if not in teach?
0269          00350 longRight:
0269 18F2      00351      BTFSC   SERVOSTATUS, SStatTeach
026A 2???      00352      GOTO     storeRight          ; if in teach mode, store and leave
026B 0008      00353      RETURN          ; if not in teach, nothing to do yet
026C          00354 storeRight:
026C 3002      00355      MOVLW   LOW EPWMCUR
026D 00F9      00356      MOVWF   EEPROMADR
026E 2???      00357      CALL    eepromRead          ; get last stored DC (from stop)
026F 00F4      00358      MOVWF   LASTPMDCL
0270 3001      00359      MOVLW   LOW EPWMCCW          ; set eeprom address for highest DC
00360 ;-----
00361 ;- store last pwm duty cycle in EEPROM location given in W
0271          00362 storeAndExit:
0271 00F9      00363      MOVWF   EEPROMADR          ; set eeprom address from W
0272 0874      00364      MOVF    LASTPMDCL, W          ; get last duty cycle (before stop)
0273 2???      00365      CALL    eepromWrite          ; and write value to eeprom
0274 2???      00366      CALL    eepromWait          ; wait until written
0275 2???      00367      CALL    leaveTeachMode
0276 0008      00368      RETURN
00369 ;=====
00188      include "Servo_Info.inc"
00001      subtitle "----- Info Return Section"

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Info **Return** SectionLOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00002 PAGE
00003 ;=====
00004 ;- Servo Info include file
00005 ;- (C) LWH brainware 2020
00006 ;=====
00007 ;- 20-06-15 lwh created
00008 ;-----
00009 ;- Single output for reporting motor position:
00010 ;- if at maximum DC, output is H (LED on)
00011 ;- if at minimum DC, output is L (LED off)
00012 ;- if moving, output blinks
00013 ;-----
00014 ;- memory registers
00015 ;-----
00016 ;- Common RAM
00017 BLINKCTR EQU LASTCOMMON ; clock divider counter for blinking
00018 LASTCOMMON SET BLINKCTR + 1 ; update next available RAM address
00019 #if (LASTCOMMON > H'7F')
00020 error "Running out of common memory!"
00021 #endif
00022 ;-----
00023 ;- Constants
00024 INFOPin EQU 1 ; port bit for info output
00025 ;-----
00026 ;- timer 1 as blink timer, blink frequency 4Hz
00027 sTmr1C4Hz EQU B'00110000'
00028 ; |||||x+--- timer off
00029 ; |||||++---- no oscillator, no external
00030 ; ||++----- prescaler 8
00031 ; ++----- clock source Fosc/4
00032 sTmr1C8Hz EQU B'00100000'
00033 ; |||||x+--- timer off
00034 ; |||||++---- no oscillator, no external
00035 ; ||++----- prescaler 4
00036 ; ++----- clock source Fosc/4
00037 sTmr1GCon EQU B'00000000'
00038 ; +----- no gate control, free run
00039 T1HZ2VAL EQU -D'15624' & H'FFFF'
00040 ;-----
00041 ;- Configure/Remap port:
00042 ;- info: RA1
00043 setupInfoPin:
00044 ; set output port
00045 banksel PORTA
00046 BCF PORTA, INFOPin ; clear output
00047 banksel LATA
00048 BCF LATA, INFOPin ; clear output latch
00049 banksel ANSELA
00050 BCF ANSELA, INFOPin ; set as digital i/o
00051 banksel TRISA
00052 BCF TRISA, INFOPin ; enable output driver (0)
00053 banksel RA1PPS
00054 MOVLW B'00000000'
00055 ; xxx+++++--- RA1 source is LATA1
00056 MOVWF RA1PPS
00057 ; setup timer
00058 banksel T1CON
00059 MOVLW sTmr1C4Hz
00060 MOVWF T1CON
00061 banksel T1GCON
00062 MOVLW sTmr1GCon
00063 MOVWF T1GCON
00064 banksel PIE1
00065 BCF PIE1, TMR1GIE ; no gate interrupt
00066 banksel PIR1
00067 BCF PIR1, TMR1GIF ; clear gate interrupt bit
00068 banksel T1CON
00069 BCF T1CON, TMR1ON ; make sure timer is off
00070 RETURN
00071 ;-----
00072 ;- set blink timing to 4Hz (slow)
00073 setBlinkSlow
00074 banksel T1CON
00075 BSF T1CON, TICKPS0
00076 RETURN
00077 ;-----
00078 ;- set blink timing to 8Hz (fast)

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Info **Return** SectionLOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

0292          00079 setBlinkFast
0292  0020          00080      banksel T1CON
0293  121B          00081      BCF      T1CON, TICKPS0
0294  0008          00082      RETURN
0294          00083 ;-----
0294          00084 ;- (re)start blinking, switching output on
0295          00085 startBlink:
0295  0020          00086      banksel T1CON
0296  101B          00087      BCF      T1CON, TMR10N      ; timer off during setup
0296          00088      ; write timer value to timer
0297  0020          00089      banksel TMR1H
0298  30C2          00090      MOVLW  high      T1HZ2VAL
0299  009A          00091      MOVWF  TMR1H
029A  30F8          00092      MOVLW  low       T1HZ2VAL
029B  0099          00093      MOVWF  TMR1L
029C  0020          00094      banksel PIR1
029D  1011          00095      BCF      PIR1, TMR1IF      ; clear interrupt bit
029E  0021          00096      banksel PIE1
029F  1411          00097      BSF      PIE1, TMR1IE      ; (re)enable interrupt
02A0  0020          00098      banksel T1CON
02A1  141B          00099      BSF      T1CON, TMR10N      ; restart timer
02A2  0008          00100      RETURN
02A2          00101 ;-----
02A2          00102 ;- stop blinking, switching output off
02A3          00103 stopBlink:
02A3  0020          00104      banksel T1CON
02A4  101B          00105      BCF      T1CON, TMR10N      ; timer off
02A5  161B          00106      BSF      T1CON, TICKPS0      ; fall back to slow (4Hz)
02A6  2???          00107      CALL    pinInfoOff          ; should already be off
02A7  1972          00108      BTFSC  SERVOSTATUS, SStatInfo
02A8  2???          00109      CALL    pinInfoOn          ; if on before, switch on again
02A9  0008          00110      RETURN
02A9          00111 ;-----
02AA          00112 ;- perform blink. called on timer interrupt
02AA          00113 onBlink:
02AA  03F8          00114      DECF   BLINKCTR, F          ; decrement counter
02AB  1903          00115      BTFSC  STATUS, Z           ; check for zero
02AC  2???          00116      GOTO   stopBlink          ; if counter 0, stop
02AC          00117      ; else blink
02AD  2???          00118      CALL    pinInfoToggle
02AE  2???          00119      GOTO   startBlink          ; restart timer
02AE          00120      ; RETURN
02AE          00121 ;-----
02AF          00122 ;- set pin on (H)
02AF          00123 pinInfoOn:
02AF  0020          00124      banksel PORTA
02B0  148C          00125      BSF     PORTA, INFOPin      ; set output
02B1  0008          00126      RETURN
02B1          00127 ;-----
02B2          00128 ;- set pin off (L)
02B2          00129 pinInfoOff:
02B2  0020          00130      banksel PORTA
02B3  108C          00131      BCF     PORTA, INFOPin      ; set output
02B4  0008          00132      RETURN
02B4          00133 ;-----
02B5          00134 ;- toggle pin now, bypassing counter
02B5          00135 pinInfoToggle:
02B5  0020          00136      banksel PORTA
02B6  1C8C          00137      BTFSS  PORTA, INFOPin      ; is pin currently on?
02B7  2???          00138      GOTO   pinInfoOn          ; if not, switch it on
02B8  2???          00139      GOTO   pinInfoOff         ; if yes, switch it off
02B8          00140      ; RETURN                  ; just in case
02B8          00141 ;-----
02B9          00142 ;- blink the number of times given in the w register
02B9          00143 ;-
02B9          00144 ;- XXX_|_|_|_|_|_XXX for W = 3
02B9          00145 ;-   6 5 4 3 2 1 0
02B9          00146 pinInfoBlink:
02B9  00F8          00147      MOVWF  BLINKCTR          ; store blink count
02BA  35F8          00148      LSLF   BLINKCTR, F          ; * 2 ( 1 on, 1 off)
02BB  1172          00149      BCF     SERVOSTATUS, SStatInfo ; clear lastinfo pin
02BC  0020          00150      banksel PORTA
02BD  188C          00151      BTFSC  PORTA, INFOPin      ; is pin currently on?
02BE  1572          00152      BSF     SERVOSTATUS, SStatInfo ; if yes, set lastinfo pin
02BF  2???          00153      CALL    pinInfoOff          ; start with off phase
02C0  2???          00154      GOTO   startBlink
02C0          00155      ; RETURN

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- Info **Return** Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```
00156 ;-----  
00157 ;- switches the info pin on and off  
00158 ;- expected behaviour: on 2 instruction cycles, off 3 instruction cycles  
00159 ;- may jitter due to interrupts  
02C1 00160 testInfoPin:  
02C1 0020 00161     banksel PORTA  
02C2 00162 loopPin:  
02C2 148C 00163     BSF     PORTA, INFOPin    ; set output      1cy  
02C3 0000 00164     NOP                               ;                  1cy  
02C4 108C 00165     BCF     PORTA, INFOPin    ; clear output   1cy  
02C5 2??? 00166     GOTO    loopPin          ;                  2cy  
02C6 0008 00167     RETURN                   ; will never return  
00168 ;=====
```

```
00189 ; this must be the last, as it addresses the EEPROMs!
```

```
00190 include "Servo_EEPROM.inc"
```

```
00001 subtitle "----- EEPROM Section"
```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- EEPROM Section

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

```

00002 PAGE
00003 ;=====
00004 ;- Servo EEPROM include file
00005 ;- (C) LWH brainware 2020
00006 ;=====
00007 ;- 20-05-31 lwh created
00008 ;-----
00009 ;- memory registers
00010 ;-----
00011 ;- Common RAM
00012 EEPROMADR EQU LASTCOMMON ; the EEPROM address for the next R/W
00013 LASTCOMMON SET EEPROMADR+1 ; update next available RAM address
00014 #if (LASTCOMMON > H'7F')
00015 error "Running out of common memory!"
00016 #endif
00017 ;-----
00018 ;- EEPROM locations
00019 EEPROMSTART EQU 7000h
00020 ; here, NVMADRH is always 70h, so use "LOW address" for NVMADRL
00021 EPWMCLW EQU EEPROMSTART + 0 ; DC clockwise position (lowest DC)
00022 EPWMCCW EQU EPWMCLW + 1 ; DC counterclockwise position (highest DC)
00023 EPWMCUR EQU EPWMCCW + 1 ; current DC (save over power off?)
00024 EPWMTARG EQU EPWMCUR + 1 ; target DC (set when motion requested)
00025
00026
00027 ;-----
00028 ;- write the value in W into the EEPROM address given in EEPROMADR
00029 ;- tested, takes approx. 5ms!
00030 eepromWrite
00031 banksel NVMCON1
00032 BTFSC NVMCON1, WR ; check whether WR is cleared
00033 RETURN ; if not, last write not finished,
00034 ; currently no more writing possible
00035 ; if write possible, do so
00036 banksel PIR2
00037 BCF PIR2, NVMIF ; clear interrupt flag
00038 banksel NVMDAT
00039 MOVWF NVMDATL
00040 BSF NVMCON1, NVMREGS ; switch to EEPROM
00041 BSF NVMCON1, WREN ; enable write/erase
00042 MOVLW HIGH EEPROMSTART
00043 MOVWF NVMADRH ; high address byte
00044 MOVF EEPROMADR, W
00045 MOVWF NVMADR ; low address byte
00046 ; NVM unlock and write sequence
00047 BCF INTCON, GIE ; avoid interrupting the sequence
00048 MOVLW 0x55
00049 MOVWF NVMCON2
00050 MOVLW 0xAA
00051 MOVWF NVMCON2
00052 BSF NVMCON1, WR ; initialize sequence by setting WR bit
00053 BSF INTCON, GIE ; reenable interrupts
00054 ; when ready, NVMCON1:WR=0, PIR2:NVMIF=1 (interrupt if enabled)
00055 RETURN
00056 ;-----
00057 ;- wait until the last eeprom write is complete
00058 eepromWait
00059 banksel NVMCON1
00060 BTFSC NVMCON1, WR ; wait until WR is cleared
00061 GOTO $-1 ; if not, wait more
00062 RETURN
00063 ;-----
00064 ;- read the value from the EEPROM address given in EEPROMADR
00065 ;- uses indirect read via FSR0 registers, tested, ca. 9us
00066 eepromRead
00067 MOVLW HIGH EEPROMSTART
00068 MOVWF FSR0H ; high address byte (always 70h)
00069 MOVF EEPROMADR, W
00070 MOVWF FSR0L ; low address byte (EEPROM address)
00071 MOVF INDF0, W ; get contents
00072 RETURN
00073 ;-----
00074 ;- reset the values in the EEPROM to the default values
00075 ;- called only if restate to factory settings is required
00076 eepromReset
00077 MOVLW LOW EPWMCLW ; select clockwise position preset
00078 MOVWF EEPROMADR

```

----- Servo Control Assembler Program, (C) LWH brainware 2020

----- EEPROM Section

```

LOC OBJECT CODE      LINE SOURCE TEXT
VALUE
02E8 301E            00079      MOVLW  PWMDCmin          ; and set to minimum
02E9 2???            00080      CALL   eepromWrite
02EA 2???            00081      CALL   eepromWait
02EB 3001            00082      MOVLW  LOW EPWMCCW      ; select counterclockwise position preset
02EC 00F9            00083      MOVWF  EEPROMADR
02ED 30AA            00084      MOVLW  PWMDCmax          ; and set to maximum
02EE 2???            00085      CALL   eepromWrite
02EF 2???            00086      CALL   eepromWait
02F0 3002            00087      MOVLW  LOW EPWMCUR      ; select current position
02F1 00F9            00088      MOVWF  EEPROMADR
02F2 305E            00089      MOVLW  PWMDCcen          ; and set to center
02F3 2???            00090      CALL   eepromWrite
02F4 2???            00091      CALL   eepromWait
02F5 3003            00092      MOVLW  LOW EPWMTARG     ; select target position
02F6 00F9            00093      MOVWF  EEPROMADR
02F7 305E            00094      MOVLW  PWMDCcen          ; and set to center
02F8 2???            00095      CALL   eepromWrite
02F9 2???            00096      CALL   eepromWait
00097      ;-- Does not work during setup (interrupts not active)!
02FA 2???            00098      CALL   setBlinkFast
02FB 3005            00099      MOVLW  5
02FC 2???            00100      CALL   pinInfoBlink     ; blink 5 times for feedback
02FD 0008            00101      RETURN
00102      ;-----
02FE            00103      eepromTest
02FE 3000            00104      MOVLW  LOW EPWMCLW
02FF 00F9            00105      MOVWF  EEPROMADR
0300 3017            00106      MOVLW  0x17
0301 2???            00107      CALL   eepromWrite
0302 3001            00108      MOVLW  LOW EPWMCCW
0303 00F9            00109      MOVWF  EEPROMADR
0304 3088            00110      MOVLW  0x88
0305 2???            00111      CALL   eepromWrite
0306 3000            00112      MOVLW  LOW EPWMCLW
0307 00F9            00113      MOVWF  EEPROMADR
0308 2???            00114      CALL   eepromRead
0309 3001            00115      MOVLW  LOW EPWMCCW
030A 00F9            00116      MOVWF  EEPROMADR
030B 2???            00117      CALL   eepromRead
030C 0008            00118      RETURN
00119      ;-----
00120      ;- initialize EEPROM locations (only on programming)
00121      ;- MUST BE AT END OF PROGRAM!
00122      CODE          0xF000          ; The absolute address F000h is mapped to
00123                                     ; the 0000 location of EE data memory for
00124                                     ; PIC18 devices.
00125      ;          CLW          CCW          current          target
F000 001E 00AA 005E 00126      de          PWMDCmin, PWMDCmax, PWMDCcen, PWMDCcen
005E
00127      ;=====
00191      subtitle      ""
00192      ;=====
1FEC 37F1 1FFF 3FFF 00193      END          ; end of program code
    
```