

```

1 ;=====
2 ;
3 ;   Filename:      NerfCannonLights.asm
4 ;   Date:         4/4/2015
5 ;   File Version:  1.0d1
6 ;
7 ;   Author:       David M. Flynn
8 ;   Company:      Oxford V.U.E., Inc.
9 ;   E-Mail:       dflynn@oxfordvue.com
10 ;   Web Site:     http://www.oxfordvue.com/
11 ;
12 ;=====
13 ;   NerfCannonLights receives a pulse from the gun controller and
14 ;   flashes the lights down the barrel.
15 ;
16 ;
17 ;   History:
18 ;
19 ; 1.0d1    4/4/2015    First code. Copied from StepperTest.
20 ;
21 ;=====
22 ; Options
23 ;
24 ;=====
25 ;
26 ; What happens next:
27 ;   At power up the system LED will blink and the barrel lights cycle once.
28 ;
29 ;
30 ;=====
31 ;
32 ;   Pin 1 (RA2/AN2) Address A2 (output)
33 ;   Pin 2 (RA3/AN3) Enable Servos 0..7 (active low output)
34 ;   Pin 3 (RA4/AN4) Enable Servos 8..15 (active low output)
35 ;   Pin 4 (RA5/MCLR*) N.C.
36 ;   Pin 5 (GND) Ground
37 ;   Pin 6 (RB0) SW3/LED3 (Active Low Input/Output)
38 ;   Pin 7 (RB1/AN11/SDA1) I2C Data
39 ;   Pin 8 (RB2/AN10/RX) SW2/LED2 (Active Low Input/Output)
40 ;   Pin 9 (RB3/CCP1) Pulse output for Servos 0..7
41 ;
42 ;   Pin 10 (RB4/AN8/SLC1) I2C Clock
43 ;   Pin 11 (RB5/AN7) SW1/LED1 (Active Low Input/Output)(System LED)
44 ;   Pin 12 (RB6/AN5/CCP2) N.C.
45 ;   Pin 13 (RB7/AN6) N.C.
46 ;   Pin 14 (Vcc) +5 volts
47 ;   Pin 15 (RA6) N.C.
48 ;   Pin 16 (RA7/CCP2) Pulse output for Servos 0..7
49 ;   Pin 17 (RA0) Address A0 (output)
50 ;   Pin 18 (RA1) Address A1 (output)
51 ;
52 ;=====
53 ;
54 ;
55 ;               list                p=16f1847,r=hex,W=0      ; list directive to def
56 ;               nolist
57 ;               include                p16f1847.inc          ; processor specific va
58 ;               list
59 ;
60 ;               __CONFIG __CONFIG1,_FOSC_INTOSC & _WDTE_OFF & _MCLRE_OFF & _IESO_OFF
61 ;
62 ;
63 ; INTOSC oscillator: I/O function on CLKIN pin
64 ; WDT disabled

```

```
65 ; PWRT disabled
66 ; MCLR/VPP pin function is digital input
67 ; Program memory code protection is disabled
68 ; Data memory code protection is disabled
69 ; Brown-out Reset enabled
70 ; CLKOUT function is disabled. I/O or oscillator function on the CLKOUT pin
71 ; Internal/External Switchover mode is disabled
72 ; Fail-Safe Clock Monitor is enabled
73 ;
74         __CONFIG __CONFIG2, _WRT_OFF & _PLLEN_OFF & _LVP_OFF
75 ;
76 ; Write protection off
77 ; 4x PLL disabled
78 ; Stack Overflow or Underflow will cause a Reset
79 ; Brown-out Reset Voltage (Vbor), low trip point selected.
80 ; Low-voltage programming enabled
81 ;
82 ; '__CONFIG' directive is used to embed configuration data within .asm file.
83 ; The labels following the directive are located in the respective .inc file.
84 ; See respective data sheet for additional information on configuration word.
85 ;
86             constant                oldCode=0
87             constant                userS232=0
88 ;
89 #Define          _C                STATUS,C
90 #Define          _Z                STATUS,Z
91 ;
92 CCPCON_Clr      EQU                b'00001001'           ;Clear output on match
93 CCPCON_Set      EQU                b'00001000'           ;Set output on match
94 kMidPulseWidth  EQU                d'3000'               ;1500uS
95 kMinPulseWidth  EQU                d'1800'               ;900uS
96 kMaxPulseWidth  EQU                d'4200'               ;2100uS
97 kServoDwellTime EQU                d'50000'              ;2.5mS/Channel
98 ;=====
99             nolist
100            include                F1847_Macros.inc
101            list
102 ;
103 ;     Port A bits
104 PortADDRBits    EQU                b'01100000'
105 PortAValue      EQU                b'00011000'
106 ;
107 #Define          Servo_A0          LATA,0                ;Output
108 #Define          Servo_A1          LATA,1                ;Output
109 #Define          Servo_A2          LATA,2                ;Output
110 #Define          Enable0_7         LATA,3                ;Output
111 #Define          Enable8_15        LATA,4                ;Output
112 #Define          RA5_In            PORTA,5                ;unused
113 #Define          RA6_In            PORTA,6                ;unused
114 #Define          RA7_Out           PORTA,7                ;CCP2 Output
115 ;
116 Servo_AddrDataMask EQU            0xF8
117 ;
118 ;
119 ;     Port B bits
120 PortBDDRBits    EQU                b'11110111'           ;LEDs Out Others In
121 PortBValue      EQU                b'00000000'
122 ;
123 #Define          SW3_In            PORTB,0                ;SW3/LED3
124 #Define          RB1_In            PORTB,1                ;I2C Data
125 #Define          SW2_In            PORTB,2                ;SW2/LED2
126 #Define          RB3_In            PORTB,2                ;CCP1 Output
127 #Define          RB4_In            PORTB,4                ;I2C Clock
128 #Define          SW1_In            PORTB,5                ;SW1/LED1
```

```
129 #Define          RB6_In          PORTB,6          ;N.C.
130 #Define          RB7_In          PORTB,7          ;N.C.
131 LED1_Bit         EQU             5                ;LED1 (Active Low Output
132 LED2_Bit         EQU             2                ;LED2 (Active Low Output
133 LED3_Bit         EQU             0                ;LED3 (Active Low Output
134 #Define          LED1_Trис       TRISB,LED1_Bit   ;LED1 (Active Low Output
135 #Define          LED2_Trис       TRISB,LED2_Bit   ;LED2 (Active Low Output
136 #Define          LED3_Trис       TRISB,LED3_Bit   ;LED3 (Active Low Output
137 ;
138 ;
139 ;=====
140 ;=====
141 ;
142 ;Constants
143 All_In           EQU             0xFF             ;
144 All_Out          EQU             0x00             ;
145 ;
146 TMR0Val          EQU             0xB2             ;0xB2=100Hz, 0.000128s/
147 LEDTIME          EQU             d'100'          ;1.00 seconds
148 LEDErrorTime     EQU             d'10'           ;
149 kWDTTime         EQU             d'200'          ;2 seconds
150 ;
151 T1CON_Val        EQU             b'00000001'      ;PreScale=1,Fosc/4,Time
152 TMR1L_Val        EQU             0x3C             ; -2500 = 2.5 mS, 400 s
153 TMR1H_Val        EQU             0xF6             ;
154 ;
155 ;TMR1L_Val        EQU             0x1E             ; -1250 = 1.25 mS, 800
156 ;TMR1H_Val        EQU             0xFB             ;
157 ;
158 ;TMR1L_Val        EQU             0x8F             ; -625 = 0.625 mS, 1600
159 ;TMR1H_Val        EQU             0xFD             ;
160 ;
161 TXSTA_Value       EQU             b'00100000'      ;8 bit, TX enabled, Asy
162 RSTA_Value        EQU             b'10010000'      ;RX enabled, 8 bit, Con
163 ; 8MHz clock low speed (BRGH=0,BRG16=1)
164 Baud_300          EQU             d'1666'         ;0.299, -0.02%
165 Baud_1200         EQU             d'416'          ;1.199, -0.08%
166 Baud_2400         EQU             d'207'          ;2.404, +0.16%
167 Baud_9600         EQU             d'51'           ;9.615, +0.16%
168 BaudRate          EQU             Baud_9600
169 ;
170 ;
171 DebounceTime      EQU             d'10'
172 ;
173 ;=====
174 ;***** VARIABLE DEFINITIONS
175 ; there are 256 bytes of ram, Bank0 0x20..0x7F, Bank1 0xA0..0xEF, Bank2 0x120..0x16F
176 ; there are 256 bytes of EEPROM starting at 0x00 the EEPROM is not mapped into memory but
177 ; accessed through the EEADR and EEDATA registers
178 ;=====
179 ; Bank0 Ram 020h-06Fh 80 Bytes
180 ;
181 Bank0_Vars        udata          0x20
182 ;
183 LED_Time          RES            1
184 lastc             RES            1                ;part of tickcount timm
185 tickcount         RES            1                ;Timer tick count
186 ;
187 StatLED_Time      RES            1
188 Stat_Count        RES            1
189 ;
190 if users232
191 TXByte            RES            1                ;Next byte to send
192 RXByte            RES            1                ;Last byte received
```

```
193 WorkingRXByte      RES      1
194 RS232Flags         RES      1
195 #Define            DataSentFlag      RS232Flags,0
196 #Define            DataReceivedFlag  RS232Flags,1
197 endif
198 ;
199 ;
200 ;
201 EEAddrTemp          RES      1      ;EEProm address to read
202 EEDataTemp          RES      1      ;Data to be written to E
203 ;
204 ;
205 Timer1Lo            RES      1      ;1st 16 bit timer
206 Timer1Hi            RES      1      ; one second RX timeiou
207 ;
208 Timer2Lo            RES      1      ;2nd 16 bit timer
209 Timer2Hi            RES      1      ;
210 ;
211 Timer3Lo            RES      1      ;3rd 16 bit timer
212 Timer3Hi            RES      1      ;GP wait timer
213 ;
214 Timer4Lo            RES      1      ;4th 16 bit timer
215 Timer4Hi            RES      1      ; debounce timer
216 ;
217 ;
218 SysFlags            RES      1
219 #Define            SW1_Flag      SysFlags,0
220 #Define            SW2_Flag      SysFlags,1
221 #Define            SW3_Flag      SysFlags,2
222 #Define            LED2_Flag     SysFlags,3
223 #Define            LED3_Flag     SysFlags,4
224 #Define            ServoOff0_7   SysFlags,5      ;Set to disable all ser
225 #Define            ServoOff8_15  SysFlags,5      ;Set to disable all ser
226 ;
227 ;#Define            FirstRAMParam      MinSpdLo
228 ;#Define            LastRAMParam      SysFlags
229 ;
230 ;
231 ;=====
232 ; Bank2 Ram 120h-16Fh 80 Bytes
233 ;
234 ; I2C Stuff is here
235 ;Note: only upper 7 bits of address are used
236 I2C_ADDRESS          EQU      0x30      ; Slave address
237 RX_ELEMENTS          EQU      .32      ; number of allowable a
238 TX_ELEMENTS          EQU      .8       ; Status nibble for eac
239 I2C_TX_Init_Val      EQU      0xAA     ; value to load into tr
240 I2C_RX_Init_Val      EQU      0xAA     ; value to load into re
241 ;
242 Bank2_Vars            udata          0x120
243 I2C_ARRAY_TX          res            RX_ELEMENTS      ; array to transmit to
244 I2C_ARRAY_RX          res            TX_ELEMENTS      ; array to receive from
245 ;
246 ;=====
247 ; Bank4 Ram 1A0h-1EFh 80 Bytes
248 ;
249 Bank3_Vars            udata          0x1A0
250 ServoMaxSpeed0_7      res            .8      ;0=no Accel, 1..255 cou
251 ServoMaxSpeed8_15     res            .8
252 ServoAccelValue0_7    res            .8      ;1..8 counts/20mS squar
253 ServoAccelValue8_15   res            .8
254 ServoCurSpeed0_7     res            .8      ;0=Stopped, MSb=Directi
255 ServoCurSpeed8_15    res            .8
256 ServoActive0_15       res            .8
```

```
257 ;
258 ;=====
259 ; Bank4 Ram 220h-26Fh 80 Bytes
260 ;
261 Bank4_Vars          udata          0x220
262 CMDServoIDX0        res            1
263 CMDServoIDX1        res            1
264 CMDSigTime0_7       res            .16          ;Commanded position
265 CMDSigTime8_15      res            .16
266 MinTime0_7          res            .16          ;Minimum pulse time (9
267 MinTime8_15         res            .16          ;Minimum pulse time (9
268 ;
269 ;=====
270 ; Bank5 Ram 2A0h-2EFh 80 Bytes
271 ;
272 Bank5_Vars          udata          0x2A0
273 ServoIDX0           res            1          ;Index 0..7
274 ServoIDX1           res            1
275 ServoFlags          res            .8          ;4 bits per servo
276 ValueSentFlag0_7    EQU            0
277 ValueSentFlag8_15   EQU            4
278 CalcdDwell          res            1          ;scratch var
279 CalcdDwellH         res            1
280 SigOutTime0_7       res            .16          ;Current position
281 DwellTime0_7        res            .16          ;Next dwell time
282 SigOutTime8_15      res            .16
283 DwellTime8_15       res            .16
284 ;
285 ;=====
286 ; Bank6 Ram 320h-26Fh 80 Bytes
287 ;
288 Bank6_Vars          udata          0x320
289 MaxTime0_7          res            .16          ;Maximum pulse time (2
290 MaxTime8_15         res            .16          ;Maximum pulse time (2
291 ;
292 ;=====
293 ; Common Ram 70-7F same for all banks
294 ; except for ISR_W_Temp these are used for paramiter passing and temp vars
295 ;=====
296 ;
297 ;          cblock          0x70
298 Param70
299 Param71
300 Param72
301 Param73
302 Param74
303 Param75
304 Param76
305 Param77
306 Param78
307 Param79
308 Param7A
309 Param7B
310 Param7C
311 Param7D
312 Param7E
313 Param7F
314 endc
315 ;
316 #Define INDEX_I2C          Param70          ;I2C Data Pointer
317 #Define GFlags            Param71
318 #Define I2C_TXLocked      Param71,0        ; Set/cleared by ISR,
319 #Define I2C_RXLocked      Param71,1        ; Set/cleared by ISR,
320 #Define I2C_NewRXData     Param71,2        ; Set by ISR, The new
```

```
321 ;
322 ;=====
323 ;Conditions
324 HasISR          EQU          0x80          ;used to enable interu
325 ;
326 ;=====
327 ;=====
328 ; ID Locations
329 ;                ORG          0x2000
330 ;                DE          '1','.', '0','0'
331 ;
332 ;=====
333 ; EEPROM locations (NV-RAM) 0x00..0x7F (offsets)
334                cblock          0x0000
335 ;
336 nvMinSpdLo;      RES          1          ;0x1E; -1250 = 1.25 mS
337 nvMinSpdHi;      RES          1          ;0xFB
338 nvMaxSpdLo;      RES          1          ;0x8F; -625 = 0.625 mS
339 nvMaxSpdHi;      RES          1          ;0xFD
340 ;
341 nvSysFlags;      RES          1
342                endc
343 ;
344 #Define          nvFirstParamByte      nvMinSpdLo
345 #Define          nvLastParamByte       nvSysFlags
346 ;
347 ;
348 ;=====
349 ;=====
350 ;
351 ;
352                ORG          0x000          ; processor reset vect
353                CLRF          STATUS
354                CLRF          PCLATH
355                goto          start          ; go to beginning of p
356 ;
357 ;=====
358 ; Interrupt Service Routine
359 ;
360 ; we loop through the interupt service routing every 0.008192 seconds
361 ;
362 ;
363                ORG          0x004          ; interrupt vector loc
364                CLRF          BSR          ; bank0
365 ;
366 ;
367                btfss          INTCON,T0IF
368                goto          SystemBlink_end
369 ;
370                movlw          TMR0Val          ;256x39+16 cycles (10,
371                addwf          TMR0,F          ; reload TMR0 with -40
372                bcf          INTCON,T0IF          ; reset interupt flag
373 ;-----
374 ; These routines run 100 times per second
375 ;-----
376 ;Decrement timers until they are zero
377 ;
378                call          DecTimer1          ;if timer 1 is not zer
379                call          DecTimer2
380                call          DecTimer3
381                call          DecTimer4
382 ;
383 ;-----
384 ; blink LEDs
```

```
385          MOVLW          LOW TRISB
386          MOVWF          FSR0L
387          MOVLW          HIGH TRISB
388          MOVWF          FSR0H
389      ; All LEDs off
390          BSF             INDF0,LED1_Bit
391          BSF             INDF0,LED2_Bit
392          BSF             INDF0,LED3_Bit
393      ;
394      ; Read SW's
395          BCF             SW1_Flag
396          BCF             SW2_Flag
397          BCF             SW3_Flag
398          BTFSS           SW1_In
399          BSF             SW1_Flag
400          BTFSS           SW2_In
401          BSF             SW2_Flag
402          BTFSS           SW3_In
403          BSF             SW3_Flag
404      ; Dec LED time
405          DECFSZ          tickcount,F          ;Is it time?
406          GOTO            SystemBlink_end      ; No, not yet
407      ;
408          MOVF            LED_Time,F
409          MOVWF           tickcount
410      ; Flash LEDs
411          BCF             INDF0,LED1_Bit
412          BTFSC           LED2_Flag
413          BCF             INDF0,LED2_Bit
414          BTFSC           LED3_Flag
415          BCF             INDF0,LED3_Bit
416      ;
417      ;
418      SystemBlink_end
419      ;
420      ;=====
421      IRQ_Servol          MOVLB          0          ;Bank0
422                          BTFSS          PIR1,CCP1IF
423                          GOTO          IRQ_Servol_End
424      ;
425                          BTFSS          ServoOff0_7          ;Are we sending a puls
426                          GOTO          IRQ_Servol_1          ; Yes
427      ;
428      ;Oops, how did we get here???
429                          MOVLB          0x05
430                          CLRF           CCP1CON
431                          GOTO          IRQ_Servol_X
432      ;
433      IRQ_Servol_1        MOVLB          0x05
434                          BTFSC          CCP1CON,CCP1M0          ;Set output on match?
435                          GOTO          IRQ_Servol_OL          ; No
436      ; An output just went high
437      ;
438          LSLF            ServoIDX0,W
439          ADDLW           LOW SigOutTime0_7
440          MOVWF          FSR0L
441          MOVLW          HIGH SigOutTime0_7
442          MOVWF          FSR0H
443      ;
444          MOVIW           FSR0++          ;Put the pulse into th
445          ADDWF           CCPR1L,F
446          MOVIW           FSR0--
447          ADDWFC          CCPR1H,F
448          MOVLW           CCPCON_Clr          ;Clear output on match
```

```
449      MOVWF      CCP1CON      ;CCP1 clr on match
450 ;Calculate dwell time
451      MOVLW      LOW kServoDwellTime
452      MOVWF      CalcdDwell
453      MOVLW      HIGH kServoDwellTime
454      MOVWF      CalcdDwellH
455      MOVIW      FSR0++      ;SigOutTime0_7
456      SUBWF      CalcdDwell,F
457      MOVIW      FSR0++      ;SigOutTime0_7+1
458      SUBWFB     CalcdDwellH,F
459 ;
460      LSLF      ServoIDX0,W
461      ADDLW      LOW DwellTime0_7
462      MOVWF      FSR0L
463      MOVF      CalcdDwell,W
464      MOVWI      FSR0++
465      MOVF      CalcdDwellH,W
466      MOVWI      FSR0++
467 ;Set Value Sent Flag
468      MOVF      ServoIDX0,W
469      ADDLW      LOW ServoFlags
470      MOVWF      FSR0L
471      BSF      INDF0,ValueSentFlag0_7
472 ;
473      GOTO      IRQ_Servo1_X
474 ;
475 ; output went low so this cycle is done
476 IRQ_Servo1_OL      LSLF      ServoIDX0,W
477      ADDLW      LOW DwellTime0_7
478      MOVWF      FSR0L
479      MOVLW      HIGH DwellTime0_7
480      MOVWF      FSR0H
481 ;
482      MOVIW      FSR0++
483      ADDWF      CCPR1L,F
484      MOVIW      FSR0++
485      ADDWFC     CCPR1H,F
486 ;
487      MOVLW      CCPCON_Set      ;Set output on match
488      MOVWF      CCP1CON
489      INCF      ServoIDX0,F
490      MOVLW      0x07
491      ANDWF      ServoIDX0,F
492 ;
493 IRQ_Servo1_X      MOVLB      0
494      BCF      PIR1,CCP1IF
495 IRQ_Servo1_End:
496 ;=====
497 IRQ_Servo2      MOVLB      0      ;Bank0
498      BTFSS     PIR2,CCP2IF
499      GOTO      IRQ_Servo2_End
500 ;
501      BTFSS     ServoOff8_15      ;Are we sending a puls
502      GOTO      IRQ_Servo2_1      ; Yes
503 ;
504 ;Oops, how did we get here???
505      MOVLB      0x05
506      CLRF      CCP2CON
507      GOTO      IRQ_Servo2_X
508 ;
509 IRQ_Servo2_1      MOVLB      0x05
510      BTFSC     CCP2CON,CCP2M0      ;Set output on match?
511      GOTO      IRQ_Servo2_OL      ; No
512 ; An output just went high
```



```
513 ;
514             LSLF             ServoIDX1,W
515             ADDLW            LOW SigOutTime8_15
516             MOVWF            FSR0L
517             MOVLW            HIGH SigOutTime8_15
518             MOVWF            FSR0H
519 ;
520             MOVIW            FSR0++                ;Put the pulse into th
521             ADDWF            CCPR2L,F
522             MOVIW            FSR0--
523             ADDWFC            CCPR2H,F
524             MOVLW            CCPCON_Clr            ;Clear output on match
525             MOVWF            CCP2CON                ;CCP1 clr on match
526 ;Calculate dwell time
527             MOVLW            LOW kServoDwellTime
528             MOVWF            CalcdDwell
529             MOVLW            HIGH kServoDwellTime
530             MOVWF            CalcdDwellH
531             MOVIW            FSR0++                ;SigOutTime0_7
532             SUBWF            CalcdDwell,F
533             MOVIW            FSR0++                ;SigOutTime0_7+1
534             SUBWFB            CalcdDwellH,F
535 ;
536             LSLF             ServoIDX1,W
537             ADDLW            LOW DwellTime8_15
538             MOVWF            FSR0L
539             MOVF             CalcdDwell,W
540             MOVWI            FSR0++
541             MOVF             CalcdDwellH,W
542             MOVWI            FSR0++
543 ;Set Value Sent Flag
544             MOVF             ServoIDX1,W
545             ADDLW            LOW ServoFlags
546             MOVWF            FSR0L
547             BSF              INDF0,ValueSentFlag8_15
548 ;
549             GOTO             IRQ_Servo2_X
550 ;
551 ; output went low so this cycle is done
552 IRQ_Servo2_OL             LSLF             ServoIDX1,W
553             ADDLW            LOW DwellTime8_15
554             MOVWF            FSR0L
555             MOVLW            HIGH DwellTime8_15
556             MOVWF            FSR0H
557 ;
558             MOVIW            FSR0++
559             ADDWF            CCPR2L,F
560             MOVIW            FSR0++
561             ADDWFC            CCPR2H,F
562 ;
563             MOVLW            CCPCON_Set            ;Set output on match
564             MOVWF            CCP2CON
565             INCF             ServoIDX1,F
566             MOVLW            0x07
567             ANDWF            ServoIDX1,F
568 ;
569 IRQ_Servo2_X             MOVLB            0
570             BCF              PIR2,CCP2IF
571 IRQ_Servo2_End:
572 ;=====
573 ;-----
574 ; I2C Com
575 IRQ_4             MOVLB            0x00
576             btfs             PIR1,SSP1IF            ; Is this a SSP interr
```

```
577          goto          IRQ_4_End          ; if not, bus collision
578          banksel       SSP1STAT
579          btfsc         SSP1STAT,R_NOT_W    ; is it a master read:
580          goto          I2C_READ            ; if so go here
581          goto          I2C_WRITE          ; if not, go here
582 I2C_READ_Return:
583 I2C_WRITE_Return      movlb          0x00
584                      bcf             PIR1,SSP1IF      ; clear the SSP interrupt
585 IRQ_4_End
586 ;-----
587 ; I2C Bus Collision
588 IRQ_5                MOVLB          0x00
589                      btfss         PIR2,BCL1IF
590                      goto          IRQ_5_End
591                      banksel       SSPBUF
592                      clrf          SSPBUF            ; clear the SSP buffer
593                      movlb         0x00             ;banksel PIR2
594                      bcf           PIR2,BCL1IF      ; clear the SSP interrupt
595                      banksel       SSPCON1
596                      bsf           SSPCON1,CKP      ; release clock stretch
597                      movlb         0x00
598 ;
599 IRQ_5_End:
600 ;
601 ;-----
602 ;
603                      retfie                ; return from interrupt
604 ;
605 ;-----
606 ;-----
607 ;-----
608 ;
609                      include        F1847_Common.inc
610                      include        I2C_SLAVE.inc
611 ;
612 ;-----
613 ;
614 start                MOVLB          0x01            ; select bank 1
615                      bsf           OPTION_REG,NOT_WPUEN ; disable pullups on pins
616                      bcf           OPTION_REG,TMR0CS    ; TMR0 clock Fosc/4
617                      bcf           OPTION_REG,PSA        ; prescaler assigned to TMR0
618                      bsf           OPTION_REG,PS0        ;111 8mhz/4/256=7812.5
619                      bsf           OPTION_REG,PS1        ;101 8mhz/4/64=31250hz
620                      bsf           OPTION_REG,PS2
621 ;
622                      MOVLB          0x01            ; bank 1
623                      MOVLW         b'01110000'        ; 8 MHz
624                      MOVWF         OSCCON
625                      movlw         b'00010111'        ; WDT prescaler 1:65535
626                      movwf         WDTCON
627 ;
628                      MOVLB          0x03            ; bank 3
629                      CLRF          ANSELB             ;Digital I/O
630 ;
631 ; setup timer 1 for 1uS/count
632 ;
633                      MOVLB          0x00            ; bank 0
634                      bcf           T1CON,TMR1CS0      ; Fosc/4 = 2Mhz
635                      bcf           T1CON,TMR1CS1
636                      bsf           T1CON,T1CKPS0      ; prescale /2
637                      bcf           T1CON,T1CKPS1
638                      bsf           T1CON,NOT_T1SYNC    ;not sync'ed
639                      bsf           T1CON,TMR1ON        ;always on
640                      bcf           T1GCON,TMR1GE
```

```
641 ;
642 ;
643             MOVLB                0x00                ;Bank 0
644 ; setup data ports
645             movlw                PortBValue
646             movwf                PORTB                ;init port B
647             movlw                PortAValue
648             movwf                PORTA
649             MOVLB                0x01                ; bank 1
650             movlw                PortADDRBits
651             movwf                TRISA
652             movlw                PortBDDRBits        ;setup for programmer
653             movwf                TRISB
654 ;
655             if users232
656 ; setup serial I/O
657             MOVLW                TXSTA_Value
658             MOVWF                TXSTA
659             MOVLW                BaudRate
660             MOVWF                SPBRG
661             MOVLB                0x00                ; bank 0
662             MOVLW                RCSTA_Value
663             MOVWF                RCSTA
664             endif
665 ;
666             CLRWDT
667 ; clear memory to zero
668             CALL                 ClearRam
669 ;-----
670 ; Setup CCP1 & CCP2 for compare
671 ;
672             MOVLW                LEDTIME
673             MOVWF                LED_Time
674 ;
675             CLRWDT
676             MOVLB                0x00
677             call                 Init_I2C                ;setup I2C
678 ;
679             bsf                  INTCON,PEIE            ; enable periferal int
680             bsf                  INTCON,T0IE            ; enable TMR0 interrupt
681             bsf                  INTCON,GIE             ; enable interrupts
682 ;
683 ;=====
684 ;=====
685 ; Main Loop
686 ;
687 ;=====
688 MainLoop    CLRWDT
689 ;
690             CALL                 I2C_DataInturp
691 ;
692             CALL                 I2C_DataSender
693 ;
694             goto                 MainLoop
695 ;
696 ;=====
697 ;=====
698 ; Parse the incoming data and put it where it belongs
699 ; Even byte: data type nibble, 4 MSb
700 ; Odd byte: data
701 kServoPosCmd    EQU                0x80                ;Position Command CMD5
702 kServoMaxSpd    EQU                0x90                ;LSB is ServoMaxSpeed
703 kServoAccel      EQU                0xA0                ;LSB is ServoAccelValu
704 kServoON         EQU                0xB0                ;Set ServoActive
```

```
705 kServoOFF          EQU          0xC0          ;Clr ServoActive
706 kServoMinTime      EQU          0xD0          ;Minimum pulse time (9
707 kServoMaxTime      EQU          0xE0          ;Maximum pulse time (2
708 ;
709 ;
710 I2C_DataInturp      BTFSC          I2C_RXLocked
711                     RETURN
712                     BTFSS          I2C_NewRXData      ;Data is new?
713                     RETURN          ; No
714                     BCF          I2C_NewRXData
715                     CLRF          Param79          ;offset
716 I2C_DataInturp_L1   LOADFSR0      I2C_ARRAY_RX,Param79
717                     MOVIW          FSR0++
718                     MOVWF          Param78
719                     ANDLW          0xF0
720                     MOVWF          Param7A
721                     MOVLW          0x0F
722                     ANDWF          Param78,F
723 ; *** kServoPosCmd ***
724                     MOVF          Param7A,W
725                     SUBLW          kServoPosCmd
726                     SKPZ
727                     GOTO          I2C_DataInturp_1
728                     LOADFSR1      CMDSigTime0_7,Param79
729 I2C_DI_Mov2         MOVF          Param78,W
730                     MOVWI          FSR1++
731                     MOVIW          FSR0++
732                     MOVWI          FSR1++
733                     GOTO          I2C_DataInturp_Next
734 ; *** kServoMaxSpd ***
735 I2C_DataInturp_1    MOVF          Param7A,W
736                     SUBLW          kServoMaxSpd
737                     SKPZ
738                     GOTO          I2C_DataInturp_2
739                     LSRF          Param79,W
740                     LOADFSR1      ServoMaxSpeed0_7,WREG
741 I2C_DI_Mov1         MOVIW          FSR0++
742                     MOVWI          FSR1++
743                     GOTO          I2C_DataInturp_Next
744 ; *** kServoAccel ***
745 I2C_DataInturp_2    MOVF          Param7A,W
746                     SUBLW          kServoAccel
747                     SKPZ
748                     GOTO          I2C_DataInturp_3
749                     LSRF          Param79,W
750                     LOADFSR1      ServoAccelValue0_7,WREG
751                     GOTO          I2C_DI_Mov1
752 ; *** kServoON ***
753 I2C_DataInturp_3    MOVF          Param7A,W
754                     SUBLW          kServoON
755                     SKPZ
756                     GOTO          I2C_DataInturp_4
757
758                     GOTO          I2C_DataInturp_Next
759 ; *** kServoOFF ***
760 I2C_DataInturp_4    MOVF          Param7A,W
761                     SUBLW          kServoOFF
762                     SKPZ
763                     GOTO          I2C_DataInturp_5
764
765                     GOTO          I2C_DataInturp_Next
766 ; *** kServoMinTime ***
767 I2C_DataInturp_5    MOVF          Param7A,W
768                     SUBLW          kServoMinTime
```

```

769          SKPZ
770          GOTO I2C_DataInturp_6
771          LOADFSR1 MinTime0_7,Param79
772          GOTO I2C_DI_Mov2
773 ; *** kServoMaxTime ***
774 I2C_DataInturp_6 MOVF Param7A,W
775                 SUBLW kServoMaxTime
776                 SKPZ
777                 LOADFSR1 MaxTime0_7,Param79
778                 GOTO I2C_DI_Mov2
779 ;
780 I2C_DataInturp_7:
781 I2C_DataInturp_Next INCF Param79,F
782                   INCF Param79,F
783                   MOVLW .32
784                   SUBWF Param79,W
785                   SKPZ
786                   GOTO I2C_DataInturp_L1
787                   MOVLB 0x00
788                   RETURN
789 ;
790 ;=====
791 ;
792 I2C_DataSender    BTFSC I2C_TXLocked
793                 RETURN
794 ;
795                 CLRF Param78
796 ;               BTFSS SW1BtnBit
797 ;               BSF Param78,0
798 ;
799                 CLRF Param79 ;offset
800                 LOADFSR0 I2C_ARRAY_TX,Param79
801                 MOVF Param78,W
802                 MOVWF INDF0
803 ;
804                 RETURN
805 ;=====
806 ;
807 ;=====
808 ; 10mS Delay
809 ; 8MHz = 500uS/instruction
810 ;
811 Light_Delay_10mS MOVLW .10
812                 MOVWF Param78
813 ;
814 Delay_W_mS       CLRF Param77
815 Light_Delay_L1   NOP
816                 NOP
817                 NOP
818                 NOP
819                 NOP
820                 DECFSZ Param77,F
821                 GOTO Light_Delay_L1 ;4uS/Loop
822                 DECFSZ Param78,F
823                 GOTO Light_Delay_L1 ;1025uS/Loop
824 ;
825 ;=====
826 ; Set CCP1/2 to go high in 0x100 clocks
827 ;
828 StartServos      MOVLB 0 ;bank 0
829                 BTFSS ServoOff0_7
830                 RETURN
831                 BCF ServoOff0_7
832 ;

```

```
833 ; CALL SetMiddlePosition
834 ; CALL Copy7CToSig
835 ;
836 MOVLW 0x00 ;start in 0x100 clocks
837 MOVWF TMR1L
838 MOVLW 0xFF
839 MOVWF TMR1H
840 ;
841 MOVLB 0x05
842 CLRF CCPR1H
843 CLRF CCPR1L
844 MOVLW CCPCON_Set
845 MOVWF CCP1CON ;go high on match
846 MOVLB 0x00 ;Bank 0
847 RETURN
848 ;
849 ; Don't disable interrupts if you don't need to...
850 SetMiddlePosition MOVLW LOW kMidPulseWidth
851 MOVWF Param7C
852 MOVLW HIGH kMidPulseWidth
853 MOVWF Param7D
854 Return
855 ;
856 ;=====
857 ; ClampInt(Param7D:Param7C,kMinPulseWidth,kMaxPulseWidth)
858 ;
859 ; Entry: Param7D:Param7C
860 ; Exit: Param7D:Param7C=ClampInt(Param7D:Param7C,kMinPulseWidth,kMaxPulseWidth)
861 ;
862 ClampInt MOVLW high kMaxPulseWidth
863 SUBWF Param7D,W ;7D-kMaxPulseWidth
864 SKPNB ;7D<Max?
865 GOTO ClampInt_1 ; Yes
866 SKPZ ;7D=Max?
867 GOTO ClampInt_tooHigh ; No, its greater.
868 MOVLW low kMaxPulseWidth ; Yes, MSB was equal c
869 SUBWF Param7C,W ;7C-kMaxPulseWidth
870 SKPNZ ;=kMaxPulseWidth
871 RETURN ;Yes
872 SKPB ;7C<Max?
873 GOTO ClampInt_tooHigh ; No
874 RETURN ; Yes
875 ;
876 ClampInt_1 MOVLW high kMinPulseWidth
877 SUBWF Param7D,W ;7D-kMinPulseWidth
878 SKPNB ;7D<Min?
879 GOTO ClampInt_tooLow ; Yes
880 SKPZ ;=Min?
881 RETURN ; No, 7D>kMinPulseWidth
882 MOVLW low kMinPulseWidth ; Yes, MSB is a match
883 SUBWF Param7C,W ;7C-kMinPulseWidth
884 SKPB ;7C>=Min?
885 RETURN ; Yes
886 ;
887 ClampInt_tooLow MOVLW low kMinPulseWidth
888 MOVWF Param7C
889 MOVLW high kMinPulseWidth
890 MOVWF Param7D
891 RETURN
892 ;
893 ClampInt_tooHigh MOVLW low kMaxPulseWidth
894 MOVWF Param7C
895 MOVLW high kMaxPulseWidth
896 MOVWF Param7D
```

```
897             RETURN
898 ;
899             if oldCode
900 ;=====
901 ;=====
902 ;
903 MoveTo78      MOVWF      FSR0L
904               MOVF       INDF0,W
905               MOVWF      Param78
906               INCF       FSR0L,F
907               MOVF       INDF0,W
908               MOVWF      Param79
909               RETURN
910 ;
911 ;=====
912 ;
913 MoveTo7C      MOVWF      FSR0L
914               MOVF       INDF0,W
915               MOVWF      Param7C
916               INCF       FSR0L,F
917               MOVF       INDF0,W
918               MOVWF      Param7D
919               RETURN
920 ;
921 ;=====
922 ;
923 Move78To7C    MOVF       Param78,W
924               MOVWF      Param7C
925               MOVF       Param79,W
926               MOVWF      Param7D
927               RETURN
928 ;
929 ;=====
930 ;
931 MoveFrom7C    MOVWF      FSR0L
932               MOVF       Param7C,W
933               MOVWF      INDF0
934               INCF       FSR0L,F
935               MOVF       Param7D,W
936               MOVWF      INDF0
937               RETURN
938 ;
939 ;=====
940 ; Less or Equal
941 ;
942 ; Entry: Param7D:Param7C, Param79:Param78
943 ; Exit: Param77:0=Param7D:Param7C<=Param79:Param78
944 ;
945 Param7D_LE_Param79 CLRF      Param77           ;default to >
946                   MOVF      Param79,W
947                   SUBWF     Param7D,W
948                   SKPNB
949                   GOTO      SetTrue
950                   SKPZ
951                   RETURN
952                   MOVF      Param78,W
953                   SUBWF     Param7C,W
954                   SKPNB
955                   GOTO      SetTrue
956                   SKPZ
957                   RETURN
958 ;
959 SetTrue       BSF          Param77,0
960               RETURN
```

[illegible]



```
1025 ;  
1026 ;  
1027 ;  
1028 ;  
1029          END  
1030 ;  
1031
```