```
;*****************************************************
; Copyright 2015 Charles Kerr
;*****************************************************
;                Comment out both for WS2811
;*****************************************************
 radix dec

;#define _WS2812
;#define _WS2801

;
;                    Comment out if not running SSD
;#define _SSD

;StartAddress equ 513
PixelCount equ 170
StartAddress equ 1
;PixelCount equ 319

;*****************************************************
;           PIC16F1455/PIC16F1825 WS2801/2811/2812
;
;*****************************************************
#ifdef __16F1455
 list  p=16F1455
#include P16F1455.INC
#else
 list p=16F1825
#include P16F1825.INC
#endif
```

```
31
32  errorlevel -302
33  errorlevel -303
34
35
36 ;*****************************************************
37 ;
38 ;                          Macros and equates
39 ;*****************************************************
40 FrameGood equ 0
41 StartCode equ 1
42 DataValid equ 2
43 DataReady equ 7
44 TimeFlag  equ 6
45
46 TimeValue equ 24
47 PgmBit equ 3
48
49 #ifdef __16F1825
50 #ifdef _SSD
51 #undefine _SSD
52 #endif
53 #endif
54
55 #ifdef  _WS2801
56 SPIBAUD equ   5
57 #else
58 SPIBAUD equ  1
59 #endif
60
```

```
61 MAXCHANL equ 1
62 #ifdef _SSD
63 MAXCHANH equ 0xc
64 #else
65 MAXCHANH equ 0x2
66 #endif
67
68 #ifdef __16F1455
69 READFLASH macro FADR,RADR
70     BCF     INTCON,GIE
71
72     BANKSEL PMADRL
73     MOVLW   low (FADR)
74     MOVWF   PMADRL
75     MOVLW   high (FADR)
76     MOVWF   PMADRH
77
78     BCF     PMCON1,CFGS
79     BSF     PMCON1,RD
80     NOP
81     NOP
82     MOVF    PMDATL,W
83     MOVWF   (RADR)
84     MOVF    PMDATH,W
85     MOVWF   (RADR+1)
86     BSF     INTCON,GIE
87  endm
88 #else
89 READFLASH macro FADR,RADR
90     BCF     INTCON,GIE
```

```
 91
 92      BANKSEL EEADRL
 93      MOVLW   low (FADR)
 94      MOVWF   EEADRL
 95      MOVLW   high (FADR)
 96      MOVWF   EEADRH
 97
 98      BCF     EECON1,CFGS
 99      BSF     EECON1,EEPGD
100      BSF     EECON1,RD
101      NOP
102      NOP
103      MOVF    EEDATL,W
104      MOVWF   (RADR)
105      MOVF    EEDATH,W
106      MOVWF   (RADR+1)
107      BSF     INTCON,GIE
108   endm
109  #endif
110  CHECKSADDR  macro
111      movf    CurIndexl,W
112      subwf   Adrl,W
113      movf    CurIndexh,W
114      subwfb  Adrh,W
115      btfsc   STATUS,C
116      bsf     FLAGS,DataValid
117   endm
118
119  CHECKEND    macro
120      movlw   1
```

```
121     addwf   CurIndexl,F
122     clrw
123     addwfc  CurIndexh,F
124     movf    CurIndexl,W
125     subwf   EndAdrl,W
126     clrw
127     movf    CurIndexh,W
128     subwfb  EndAdrh,W
129     btfsc   STATUS,C
130     goto    $+3
131     clrf    FLAGS
132     bsf     FLAGS,DataReady
133  endm
134 #ifdef   _WS2801
135 SENDPIXEL macro
136     movwf   SSP1BUF
137     btfss   SSP1STAT,BF
138     goto    $-1
139     movf    SSP1BUF,W
140  endm
141 #else
142 SENDBIT   macro  bit
143     movlw   0xc0
144     btfsc   Holder1,bit
145 #ifdef __16F1455
146     movlw   0xF0
147 #else
148     movlw   0xF8
149 #endif
150     movwf   SSP1BUF
```

```
151     btfss   SSP1STAT,BF
152     goto    $-1
153     movf    SSP1BUF,W
154  endm
155 SENDPIXEL macro
156     movwf   Holder1
157     SENDBIT 7
158     SENDBIT 6
159     SENDBIT 5
160     SENDBIT 4
161     SENDBIT 3
162     SENDBIT 2
163     SENDBIT 1
164     SENDBIT 0
165  endm
166 #endif
167 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
168 ;
169 ;                                   Configuration words
170 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
171 ;
172 #ifdef __16F1455
173  __CONFIG _CONFIG1, _FOSC_INTOSC&_WDTE_OFF&_PWRTE_
174  __CONFIG _CONFIG2, _WRT_OFF&_PLLMULT_3x &_PLLEN_E
175 #else
176  __CONFIG _CONFIG1, _FOSC_INTOSC&_WDTE_OFF&_PWRTE_
177  __CONFIG _CONFIG2, _WRT_OFF&_PLLEN_ON & _LVP_OFF
178 #endif
179 ;**********************************************************
180 ;   Vector Segment
```

```
181 ;*****************************************************************
182     org 0x0
183     pagesel Main
184     goto    Main
185     NOP
186     NOP
187     pagesel ISR
188     goto    ISR
189 ;*****************************************************************
190 ;
191 ;                   Flash Storage section
192 ;*****************************************************************
193     org 32
194 SAddr:       dw StartAddress
195 SCount:      dw PixelCount
196 SType:       dw 1
197
198 ;*****************************************************************
199 ;
200 ;                   RAM Storage
201 ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
202  CBLOCK       0x70
203     FLAGS
204     Adrl
205     Adrh
206     EndAdrl
207     EndAdrh
208     Templ
209     Temph
210     CurIndexl
```

```
211     CurIndexh
212     WriteIndexl
213     WriteIndexh
214     PixelCountl
215     PixelCounth
216     Timer
217     Holder
218     Holder1
219  ENDC
220
221    org 100
222 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
223 ;                Setup Oscillator
224 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
225 CnfOSC:
226
227         BANKSEL OSCCON      ; We need the bank wit
228 #ifdef  __16F1455
229         movlw   0xFC        ;PLLEN enabled, IRC=48
230 #else
231         movlw   0xF0        ;PLLEN enabled, IRC=32
232 #endif
233         movwf   OSCCON      ; move w into OSSCON r
234         BANKSEL CLKRCON
235         movlw   0x0
236         movwf   CLKRCON
237
238         BANKSEL OSCSTAT
239 #ifdef __16F1455
240
```

```
241         btfss   OSCSTAT,PLLRDY
242 #else
243         btfss   OSCSTAT,PLLR
244 #endif
245         goto $-1
246 ;
247 ;           We are now running at 48/32 Mhz
248         return
249
250 ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
251 ;               Confgure mappable pins
252 ;               Setup the Port Pins
253 ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
254 CnfIO:
255 ;
256 ;           Disable analog settings;
257 ;
258         BANKSEL ANSELA
259         clrf    ANSELA
260         BANKSEL ANSELC
261         clrf    ANSELC
262 ;
263 ;           Configure the mappable pins
264 ;
265 #ifdef __16F1455
266         BANKSEL APFCON
267         clrf    APFCON
268 #else
269         BANKSEL APFCON0
270         clrf    APFCON0
```

```
271         clrf    APFCON1
272 #endif
273 ;
274 ;           Configure the PORTA setup
275 ;
276         BANKSEL PORTA
277         clrf    PORTA
278         BANKSEL LATA
279         clrf    LATA
280         BANKSEL TRISA
281
282         movlw   0x38            ; RA3 is input or
283         movwf   TRISA
284
285 ;
286 ;           Configure the PORTC setup
287         BANKSEL PORTC
288         clrf    PORTC
289         BANKSEL LATC
290         clrf    LATC
291         BANKSEL TRISC
292
293         movlw   0x30            ; RC5 is input fo
294         movwf   TRISC
295 ;
296 ;           Configure the PullUps
297 ;
298         BANKSEL OPTION_REG
299         bcf     OPTION_REG,7
300         BANKSEL WPUA
```

```
301         clrf        WPUA
302         ;bsf        WPUA,3          ; RA 3 pullup
303         bsf         WPUA,PgmBit
304         return
305
306 #ifdef __16F1455
307 ;++++++++++++++++++++++++++++++++++++++++++++++++++++
308 ;               Confgure MSSP (master serial port
309 ;++++++++++++++++++++++++++++++++++++++++++++++++++++
310 CnfMSSP:
311 ;           Configure the SPI module
312         BANKSEL SSP1CON1          ; SPI registers
313         movlw   0xa
314         movwf SSP1CON1             ; Disable the S
315 ;           Baud Rate reg value is (where Fbaud i
316 ;           regvalue = ((Fosc/Fbaud)/4)-1
317 ;           Fbaud=Fosc/(4*(regvalue+1))
318                                   ; 3 for baud rate
319         movlw SPIBAUD               ; Move 3 into
320         BANKSEL SSP1ADD
321         movwf SSP1ADD             ;
322 ;
323 ;                                   SMP=0 Input data
324                                  ;CKE=0 Transmit on lo
325         BANKSEL SSP1STAT
326         clrf SSP1STAT             ;
327 #ifdef   _WS2801
328         bsf  SSP1STAT,CKE
329 #else
330         bcf  SSP1STAT,CKE
```

```
331 #endif
332 ;
333
334         BANKSEL SSP1CON1
335         bsf   SSP1CON1,SSPEN    ; Enable serialpo
336
337         return
338 #else
339 ;++++++++++++++++++++++++++++++++++++++++++++++++++++
340 ;               Confgure MSSP (master serial port
341 ;++++++++++++++++++++++++++++++++++++++++++++++++++++
342 CnfMSSP:
343 ;           Configure the SPI module
344         BANKSEL SSP1CON1          ; SPI registers
345 #ifdef  _WS2801
346         movlw   0x1
347 #else
348         movlw   0x0
349 #endif
350         movwf SSP1CON1              ; Disable the S
351 ;           Baud Rate reg value is (where Fbaud i
352 ;           regvalue = ((Fosc/Fbaud)/4)-1
353 ;           Fbaud=Fosc/(4*(regvalue+1))
354                                   ; 3 for baud rate
355         movlw SPIBAUD               ; Move 3 into
356         BANKSEL SSP1ADD
357         movwf SSP1ADD             ;
358 ;
359 ;                                   SMP=0 Input data
360                                 ;CKE=0 Transmit on lo
```

```
361         BANKSEL SSP1STAT
362         clrf SSP1STAT           ;
363 #ifdef   _WS2801
364         bsf  SSP1STAT,CKE
365 #else
366         bcf  SSP1STAT,CKE
367 #endif
368 ;
369
370         BANKSEL SSP1CON1
371         bsf   SSP1CON1,SSPEN    ; Enable serialpo
372
373         return
374 #endif
375 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
376 ;               Configure UART
377 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
378 CnfUART:
379         BANKSEL SPBRGL           ; We one are goi
380         clrf    SPBRGH
381 ;
382 ;           Baud rate first
383 ;
384
385                                  ; We want Async,
386                                  ; With that, the
387                                  ; Fbaud = (Fosc(6
388                                  ; SPBRG = (Fosc/(
389                                  ; Fbaud 250Khz SP
390 #ifdef __16F1455
```

```
391         movlw   2                   ; Loading
392
393 #else
394         movlw   1
395 #endif
396         movwf   SPBRGL              ;
397         clrf    BAUDCON
398         clrf    TXSTA
399 #ifdef _SSD
400         bsf     TXSTA,BRGH
401 #endif
402
403         BANKSEL PIE1                ;
404         bsf   PIE1,RCIE             ;
405         bsf INTCON,PEIE
406         BANKSEL RCSTA
407         clrf RCSTA
408         movlw 0x10
409         movwf RCSTA
410
411         return
412
413 ;*****************************************************
414 ;               CnfTimer1
415 ;*****************************************************
416 CnfTimer1:
417     BANKSEL         T1CON
418     movlw           0x30
419     movwf           T1CON
420     movlw           TimeValue
```

```
421     movwf           Timer
422     clrf            TMR1L
423     clrf            TMR1H
424     BANKSEL         PIE1
425     bsf             PIE1,TMR1IE
426     return
427 #ifdef __16F1455
428 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
429 ;
430 ;                   Erase Stored Data
431 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
432 EraseFlash:
433     BANKSEL PMADRL
434     movlw   low SAddr       ; Load the address of
435     movwf   PMADRL
436     movlw   high SAddr
437     movwf   PMADRH
438     bcf     PMCON1, CFGS
439     bsf     PMCON1,FREE
440     bsf     PMCON1,WREN
441
442     movlw   0x55            ; unlock sequence
443     movwf   PMCON2
444     movlw   0xAA
445     movwf   PMCON2
446     bsf     PMCON1,WR
447     nop
448     nop
449     bcf     PMCON1,WREN
450
```

```
451     return
452
453 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
454 ;
455 ;               Write Flash
456 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
457 WriteFlash:
458     call        EraseFlash
459     bcf         INTCON,GIE
460     BANKSEL     PMADRH
461     movlw       high SAddr
462     movwf       PMADRH
463     movlw       low SAddr
464     movwf       PMADRL
465
466     bcf         PMCON1,CFGS
467     bsf         PMCON1,WREN
468     bsf         PMCON1,LWLO
469
470     movf        Adrl,W
471     movwf       PMDATL
472     movf        Adrh,W
473     movwf       PMDATH
474
475     movlw       0x55
476     movwf       PMCON2
477     movlw       0xAA
478     movwf       PMCON2
479     bsf         PMCON1,WR
480     nop
```

```
481     nop
482
483     incf        PMADRL,F
484     movf        PixelCountl,W
485     movwf       PMDATL
486     movf        PixelCounth,W
487     movwf       PMDATH
488
489     movlw       0x55
490     movwf       PMCON2
491     movlw       0xAA
492     movwf       PMCON2
493     bsf         PMCON1,WR
494     nop
495     nop
496     incf        PMADRL,F
497     movlw       29
498     movwf       Templ
499 Floop:
500     clrw
501     movwf       PMDATL
502     movwf       PMDATH
503     movlw       0x55
504     movwf       PMCON2
505     movlw       0xAA
506     movwf       PMCON2
507     bsf         PMCON1,WR
508     nop
509     nop
510     incf        PMADRL,F
```

```
511     decf        Templ,F
512     btfss       STATUS,Z
513     goto        Floop
514
515     bcf         PMCON1,LWLO
516     movlw       0x55
517     movwf       PMCON2
518     movlw       0xAA
519     movwf       PMCON2
520     bsf         PMCON1,WR
521     nop
522     nop
523     bcf         PMCON1,WREN
524     bsf         INTCON,GIE
525     return
526 #else
527 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
528 ;
529 ;                   Erase Stored Data
530 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++
531 EraseFlash:
532     BANKSEL EEADRL
533     movlw   low SAddr       ; Load the address of
534     movwf   EEADRL
535     movlw   high SAddr
536     movwf   EEADRH
537     bsf     EECON1,EEPGD    ; setup erase configu
538     bcf     EECON1, CFGS
539     bsf     EECON1,FREE
540     bsf     EECON1,WREN
```

```
541
542     movlw   0x55            ; unlock sequence
543     movwf   EECON2
544     movlw   0xAA
545     movwf   EECON2
546     bsf     EECON1,WR
547     nop
548     nop
549     bcf     EECON1,WREN
550
551     return
552 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
553 ;
554 ;               Write Flash
555 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
556 WriteFlash:
557     call        EraseFlash
558     bcf         INTCON,GIE
559     BANKSEL     EEADRH
560     movlw       high SAddr
561     movwf       EEADRH
562     movlw       low SAddr
563     movwf       EEADRL
564
565     bsf         EECON1,EEPGD
566     bcf         EECON1,CFGS
567     bsf         EECON1,WREN
568     bsf         EECON1,LWLO
569
570     movf        Adrl,W
```

```
571     movwf       EEDATL
572     movf        Adrh,W
573     movwf       EEDATH
574
575     movlw       0x55
576     movwf       EECON2
577     movlw       0xAA
578     movwf       EECON2
579     bsf         EECON1,WR
580     nop
581     nop
582
583     incf        EEADRL,F
584     movf        PixelCountl,W
585     movwf       EEDATL
586     movf        PixelCounth,W
587     movwf       EEDATH
588
589     movlw       0x55
590     movwf       EECON2
591     movlw       0xAA
592     movwf       EECON2
593     bsf         EECON1,WR
594     nop
595     nop
596     incf        EEADRL,F
597     movlw       29
598     movwf       Templ
599 Floop:
600     clrw
```

```
601     movwf       EEDATL
602     movwf       EEDATH
603     movlw       0x55
604     movwf       EECON2
605     movlw       0xAA
606     movwf       EECON2
607     bsf         EECON1,WR
608     nop
609     nop
610     incf        EEADRL,F
611     decf        Templ,F
612     btfss       STATUS,Z
613     goto        Floop
614
615     bcf         EECON1,LWLO
616     movlw       0x55
617     movwf       EECON2
618     movlw       0xAA
619     movwf       EECON2
620     bsf         EECON1,WR
621     nop
622     nop
623     bcf         EECON1,WREN
624     bsf         INTCON,GIE
625     return
626 #endif
627 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
628 ;
629 ;                   Read flash
630 ;
```

```
;++++++++++++++++++++++++++++++++++++++++++++++++++
ReadFlash:
    READFLASH   SAddr,Adrl
    READFLASH   SCount,Templ
    movf        Templ,W
    movwf       PixelCountl
    movf        Temph,W
    movwf       PixelCounth

    movf        Adrl,W
    movwf       EndAdrl
    movf        Adrh,W
    movwf       EndAdrh

    movf        Templ,W
    addwf       EndAdrl,F
    movf        Temph,W
    addwfc      EndAdrh,F

    movf        Templ,W
    addwf       EndAdrl,F
    movf        Temph,W
    addwfc      EndAdrh,F

    movf        Templ,W
    addwf       EndAdrl,F
    movf        Temph,W
    addwfc      EndAdrh,F

    movlw       1
```

```
661     subwf       EndAdrl,F
662     clrw
663     subwfb      EndAdrh,F
664 ;
665 ;                   Ensure that address isn't gre
666     movlw       MAXCHANL
667     subwf       EndAdrl,W
668     movlw       MAXCHANH
669     subwfb      EndAdrh,W
670     btfsc       STATUS,C
671     goto        RADDR
672     goto        ERead
673 RADDR:
674     movlw       MAXCHANL
675     movwf       EndAdrl
676     movlw       MAXCHANH
677     movwf       EndAdrh
678     movlw       1
679     subwf       EndAdrl,F
680     clrw
681     subwfb      EndAdrh,F
682
683 ERead:
684
685 ;                           Check our start addre
686     movlw       MAXCHANL
687     subwf       Adrl,W
688     movlw       MAXCHANH
689     subwfb      Adrh,W
690     btfsc       STATUS,C
```

```
691     goto        SADDR
692     goto        FRead
693 SADDR:
694     movlw      MAXCHANL
695     movwf       Adrl
696     movlw       MAXCHANH
697     movwf       Adrh
698     movlw       1
699     subwf       Adrl,F
700     clrw
701     subwfb      Adrh,F
702 FRead:
703     return
704
705 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
706 ;                   PgmOn
707 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
708 PgmOn:
709     BANKSEL     SSP1BUF
710 #ifdef      _WS2812
711     clrw
712     SENDPIXEL
713     movlw   0xff
714     SENDPIXEL
715 #else
716     movlw   0xff
717     BANKSEL SSP1BUF
718     SENDPIXEL
719     clrw
720     SENDPIXEL
```

```
721 #endif
722     clrw
723     SENDPIXEL
724     return
725 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
726 ;                   PgmGood
727 ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
728 PgmGood:
729     BANKSEL SSP1BUF
730 #ifdef  _WS2812
731     movlw   0xFF
732     SENDPIXEL
733     clrw
734     SENDPIXEL
735 #else
736     clrw
737     SENDPIXEL
738     movlw   0xFF
739     SENDPIXEL
740 #endif
741     clrw
742     SENDPIXEL
743     return
744 ;************************************************************************
745 ;                   Get Data
746 ;************************************************************************
747 ClrRam:
748     movlw   0x20
749     BANKSEL FSR0H
750     movwf   FSR0H
```

```
751     clrf    FSR0L
752     movlw   3
753     movwf   Temph
754     movlw   0x20
755     movwf   Templ
756     movlw   0x0
757     movwi   FSR0++
758     movlw   1
759     subwf   Templ,F
760     clrw
761     subwfb  Temph,F
762     btfsc   STATUS,C
763     goto    $-7
764     return
765
766 ;*****************************************************
767 ;                Configure Timer2
768 ;*****************************************************
769 CnfTmr2:
770     BANKSEL T2CON
771     movlw   0x7b
772     movwf   T2CON
773     BANKSEL PR2
774     movlw   0xFF
775     movwf   PR2
776     clrf    TMR2
777     return
778 ;*****************************************************
779 ;                Delay
780 ;*****************************************************
```

```
781 Delay:
782     BANKSEL T2CON
783     bsf     T2CON,TMR2ON
784     BANKSEL PIR1
785     btfss   PIR1,TMR2IF
786     goto    $-1
787     bcf     PIR1,TMR2IF
788     BANKSEL T2CON
789     bcf     T2CON,TMR2ON
790     return
791
792
793 ;*****************************************************
794 ;                 Main routine
795 ;*****************************************************
796 Main:
797     call    CnfOSC
798     call    ClrRam
799     call    CnfIO
800     call    CnfMSSP
801     call    CnfUART
802     call    CnfTmr2
803     call    Delay
804     call    CnfTimer1
805     bsf     INTCON,GIE
806
807     clrf    FLAGS
808     BANKSEL PORTA
809     btfss   PORTA,PgmBit
810     goto    PgmMode
```

```
811     call    ReadFlash
812     bsf     INTCON,GIE
813     clrf    FLAGS
814     BANKSEL T1CON
815     bsf     T1CON,TMR1ON
816     BANKSEL RCSTA
817     bsf     RCSTA,SPEN
818     BANKSEL SSP1BUF
819 MainLoop:
820     btfsc   FLAGS,TimeFlag
821     goto    SendClear
822     btfss   FLAGS,DataReady
823     goto    $-3
824     movf    PixelCountl,W
825     movwf   Templ
826     movf    PixelCounth,W
827     movwf   Temph
828     movlw   0x20
829     movwf   FSR1H
830     clrf    FSR1L
831 SLoop:
832 #ifdef   _WS2812
833     moviw   FSR1++
834     movwf   Holder
835     moviw   FSR1++
836     SENDPIXEL
837     movf    Holder,W
838     SENDPIXEL
839 #else
840     moviw   FSR1++
```

```
841     SENDPIXEL
842     moviw   FSR1++
843     SENDPIXEL
844 #endif
845     moviw   FSR1++
846     SENDPIXEL
847     movlw   1
848     subwf   Templ,F
849     clrw
850     subwfb  Temph,F
851     movf    Templ,W
852     iorwf   Temph,W
853     btfss   STATUS,Z
854     goto    SLoop
855     bcf     FLAGS,DataReady
856     goto    MainLoop
857 SendClear:
858     bcf     FLAGS,TimeFlag
859     movf    PixelCountl,W
860     movwf   Templ
861     movf    PixelCounth,W
862     movwf   Temph
863 ClearLoop:
864     clrw
865     SENDPIXEL
866     clrw
867     SENDPIXEL
868     clrw
869     SENDPIXEL
870     movlw   1
```

```
871     subwf   Templ,F
872     clrw
873     subwfb  Temph,F
874     movf    Templ,W
875     iorwf   Temph,W
876     btfss   STATUS,Z
877     goto    ClearLoop
878     goto    MainLoop
879
880
881 PgmMode:
882     movlw   1
883     movwf   Adrl
884     clrf    Adrh
885     movlw   5
886     movwf   EndAdrl
887     clrf    EndAdrh
888     bsf     INTCON,GIE
889     call    PgmOn
890     BANKSEL RCSTA
891     clrf    FLAGS
892     bsf     RCSTA,SPEN
893
894 PgmLoop:
895     btfss   FLAGS,DataReady
896     goto    PgmLoop
897     bcf     RCSTA,SPEN
898     bcf     FLAGS,DataReady
899     movlw   0x20
900     movwf   FSR0H
```

```
901     clrf    FSR0L
902     moviw   FSR0++
903     xorlw   0xAA
904     btfss   STATUS,Z
905     goto    PgmLoop-2
906
907     BANKSEL RCSTA
908     bcf     RCSTA,SPEN
909
910     moviw   FSR0++
911     movwf   Adrl
912     moviw   FSR0++
913     movwf   Adrh
914     moviw   FSR0++
915     movwf   PixelCountl
916     moviw   FSR0++
917     movwf   PixelCounth
918
919     movlw       MAXCHANL
920     subwf       Adrl,W
921     movlw       MAXCHANH
922     subwfb      Adrh,W
923     btfsc       STATUS,C
924     goto        Main
925
926     movf    Adrl,W
927     movwf   Templ
928     movf    Adrh,W
929     iorwf   Templ,W
930     btfsc   STATUS,Z
```

```
931     goto    Main
932     call    WriteFlash
933     call    PgmGood
934     goto    $
935
936 ;*****************************************************
937 ;                Interrupt vector
938 ;*****************************************************
939 ISR:
940     BANKSEL PIR1
941     btfsc   PIR1,TMR1IF
942     goto    Timeout
943     btfss   PIR1,RCIF
944     goto    BadInterrupt
945     bcf     PIR1,RCIF
946     ; Check for a framing error
947     BANKSEL RCSTA
948     btfsc   RCSTA,FERR          ; Check for a framm
949     goto    FrameError
950     ; Should we look for a start code?
951     btfss   FLAGS,FrameGood
952     goto    FlushRead
953     btfss   FLAGS,StartCode
954     goto    CheckStart
955     btfss   FLAGS,DataValid
956     goto    NotValid
957     ; Data is valid
958     movlw   0x20
959     movwf   FSR0H
960     clrf    FSR0L
```

```
961     movf    WriteIndexl,W
962     addwf   FSR0L,F
963     movf    WriteIndexh,W
964     addwfc  FSR0H,F
965     movf    RCREG,W
966     movwi   FSR0++
967
968
969 ;***************CheckEnd
970     movlw   1
971     addwf   CurIndexl,F
972     clrw
973     addwfc  CurIndexh,F
974     movlw   1
975     addwf   WriteIndexl,F
976     clrw
977     addwfc  WriteIndexh,F
978     movf    CurIndexl,W
979     subwf   EndAdrl,W
980     clrw
981     movf    CurIndexh,W
982     subwfb  EndAdrh,W
983     btfsc   STATUS,C
984     goto    $+3
985     clrf    FLAGS
986     bsf     FLAGS,DataReady
987 ;*****************************************
988
989
990
```

```
991     goto    ISRend
992
993 NotValid:
994     movlw   1
995     addwf   CurIndexl,F
996     clrw
997     addwfc  CurIndexh,F
998 ;************CheckSadr
999     movf    Adrl,W
1000    subwf   CurIndexl,W
1001    movf    Adrh,W
1002    subwfb  CurIndexh,W
1003    btfsc   STATUS,C
1004    ;bsf     FLAGS,DataValid
1005    goto    SetDataR
1006 ;*****************************
1007    goto    FlushRead
1008 CheckStart:
1009    movf    RCREG,W
1010    btfss   STATUS,Z
1011    goto    BadStart
1012    bsf     FLAGS,StartCode
1013    movlw   1
1014    movwf   CurIndexl
1015    clrf    CurIndexh
1016
1017 ;************CheckSadr
1018    movf    Adrl,W
1019    subwf   CurIndexl,W
1020    movf    Adrh,W
```

```
1021     subwfb  CurIndexh,W
1022     btfsc   STATUS,C
1023     goto    SetData
1024     ;bsf     FLAGS,DataValid
1025
1026 ;*****************************
1027
1028
1029
1030
1031     goto    ISRend
1032 SetDataR:
1033     movf    RCREG,W
1034 SetData:
1035     bsf     FLAGS,DataValid
1036     clrf    WriteIndexl
1037     clrf    WriteIndexh
1038     goto    ISRend
1039 BadStart:
1040     movlw   0x80
1041     andwf   FLAGS,F
1042     goto    ISRend
1043 FrameError:
1044     movlw   TimeValue
1045     movwf   Timer
1046     btfsc   FLAGS,DataValid
1047     bsf     FLAGS,DataReady
1048     movlw   0x80
1049     andwf   FLAGS,F
1050     bsf     FLAGS,FrameGood
```

```
1051     ;clrf    WriteIndexl
1052     ;clrf    WriteIndexh
1053
1054     goto    FlushRead
1055 Timeout:
1056     bcf     PIR1,TMR1IF
1057     decfsz  Timer,F
1058     goto    ISRend
1059     movlw   TimeValue
1060     movwf   Timer
1061     bsf     FLAGS,TimeFlag
1062     goto    ISRend
1063
1064 FlushRead:
1065     movf    RCREG,W
1066 BadInterrupt:
1067 ISRend:
1068     retfie
1069
1070     end
```