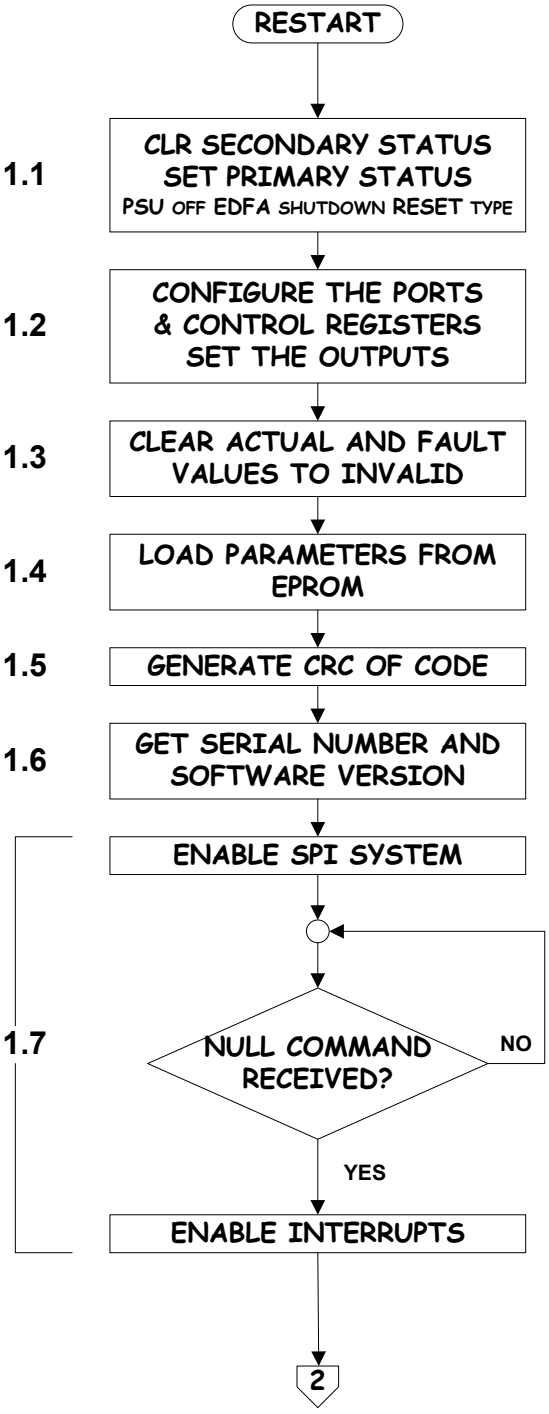
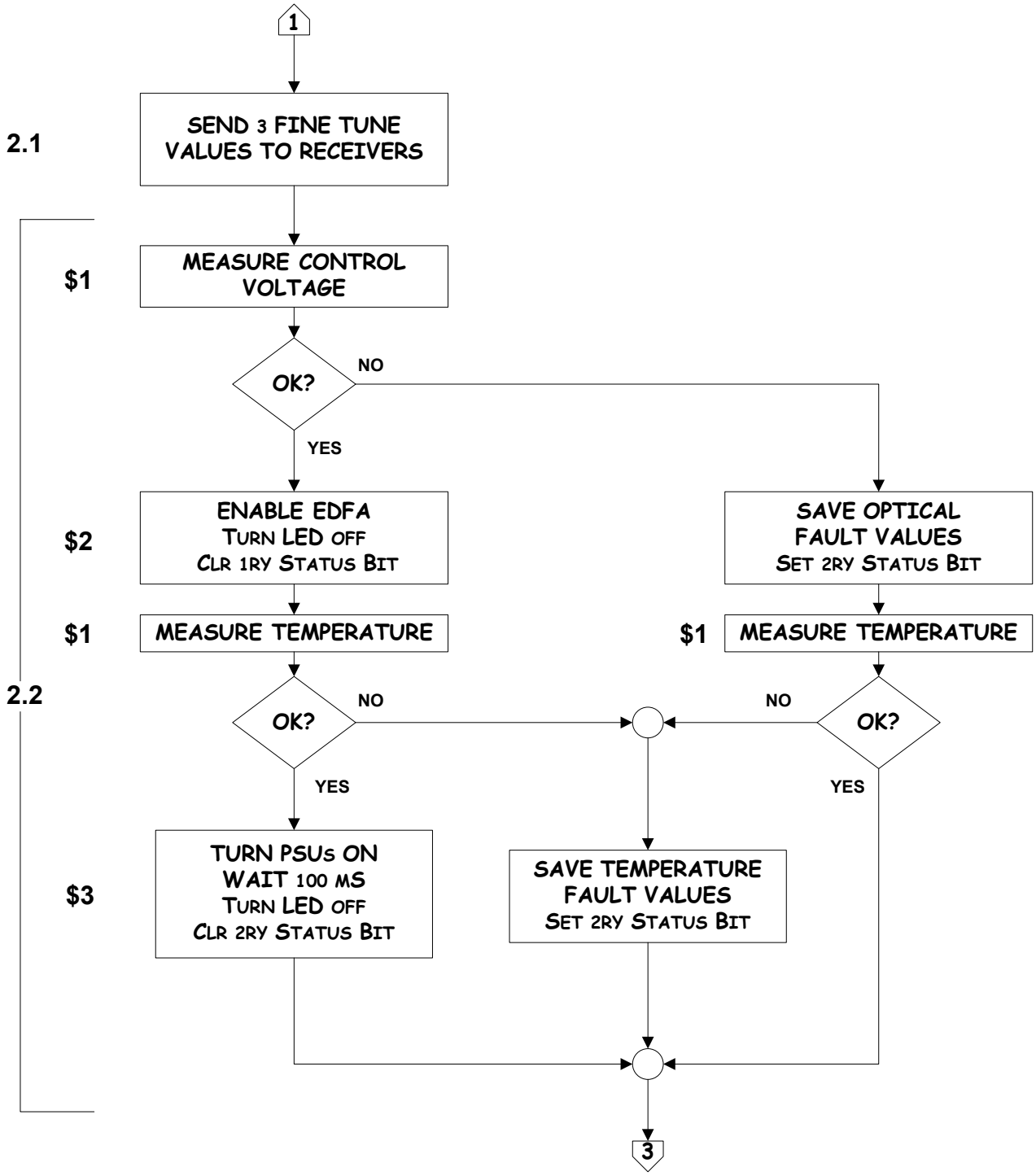
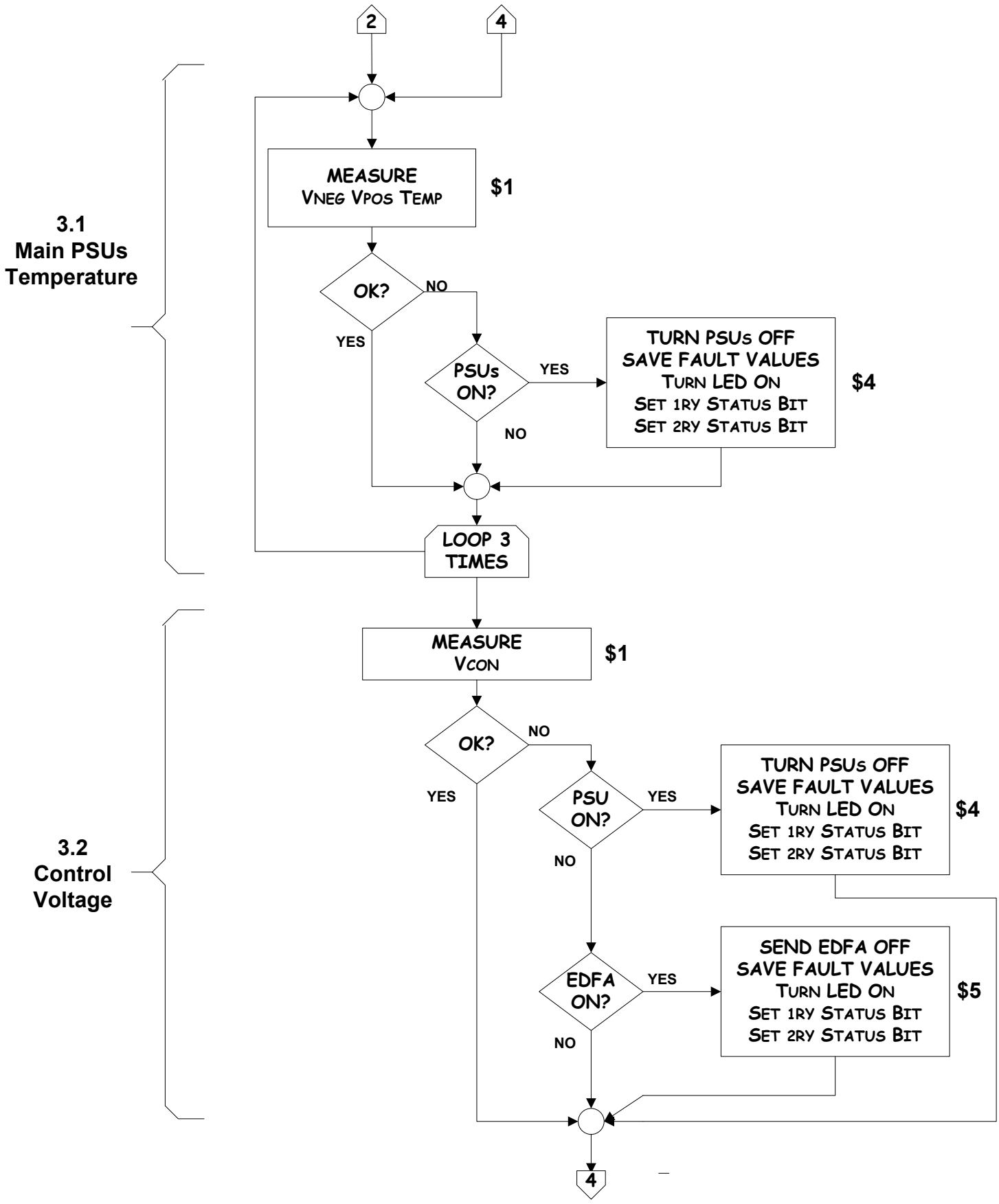
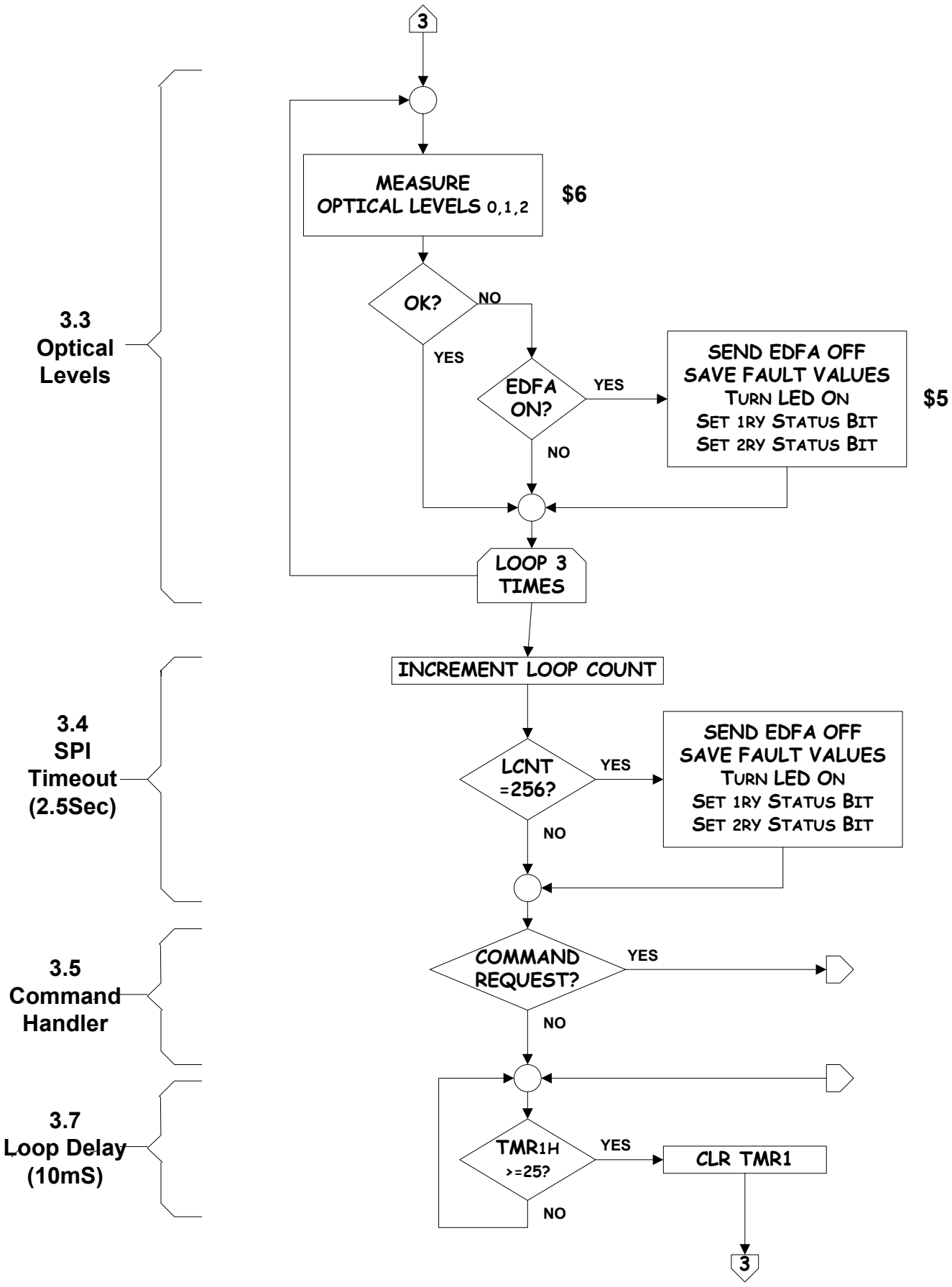
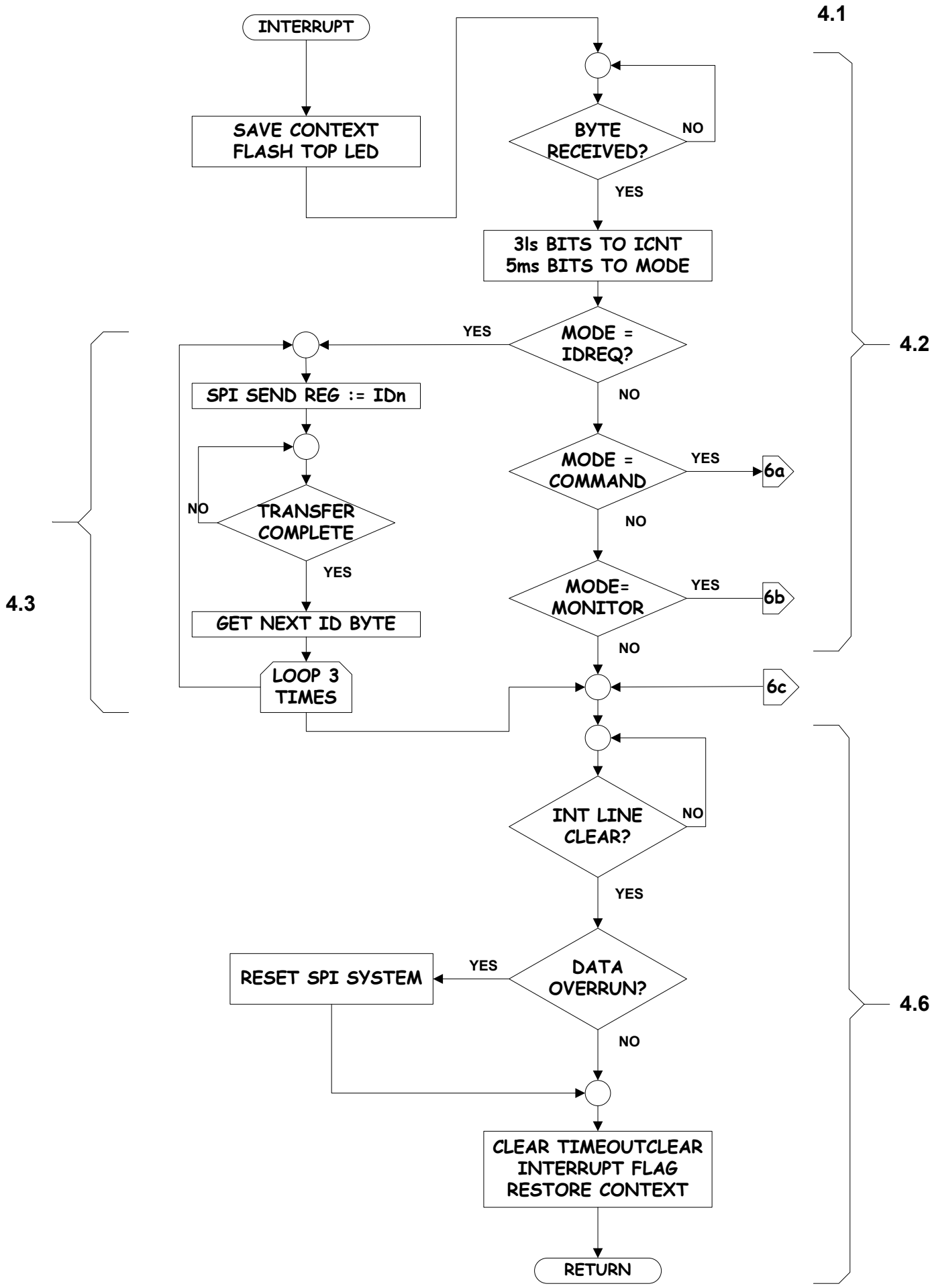
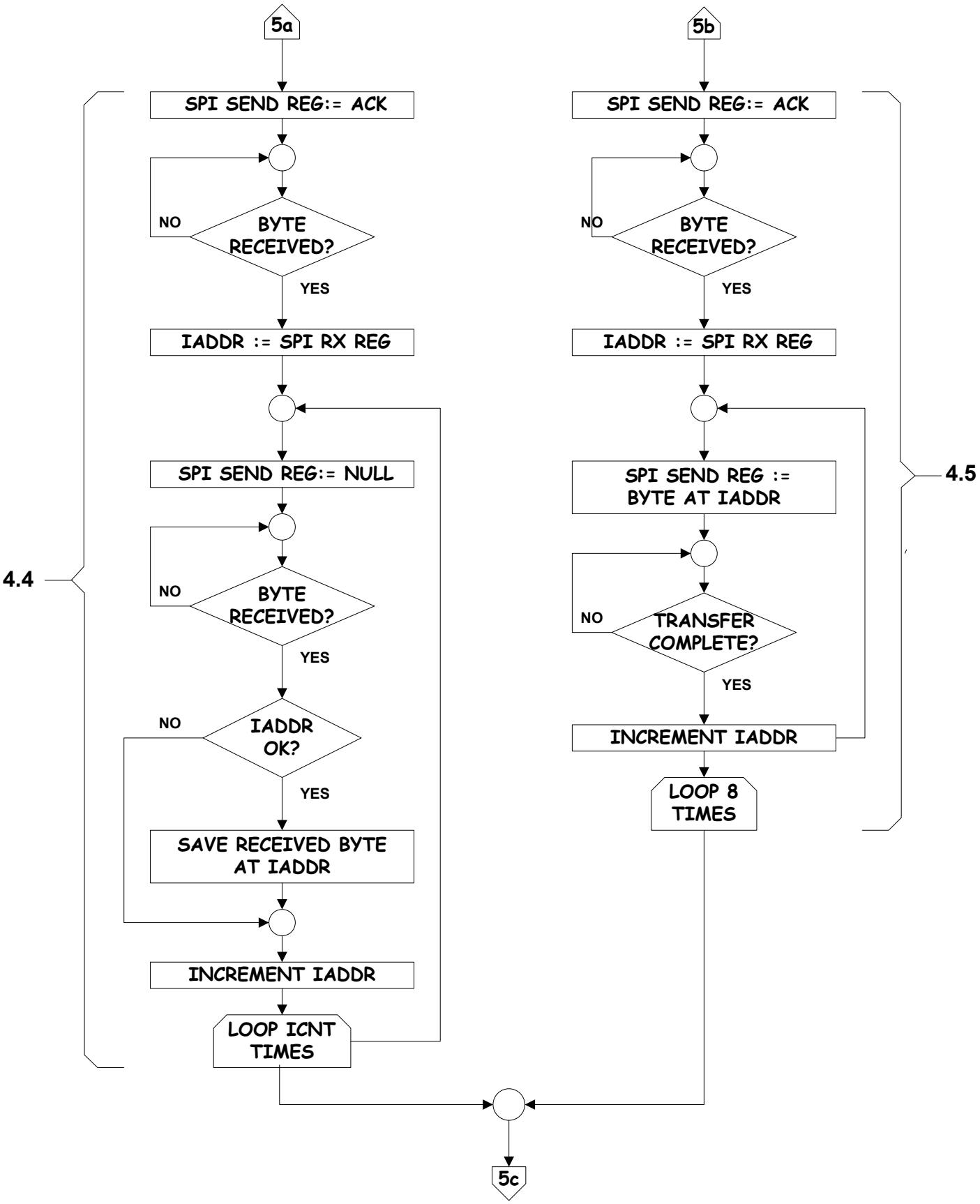

RESTART
1.1
CLR SECONDARY STATUS
SET PRIMARY STATUS
PSU OFF EDFA SHUTDOWN RESET TYPE
1.2
CONFIGURE THE PORTS
& CONTROL REGISTERS
SET THE OUTPUTS
1.3
CLEAR ACTUAL AND FAULT
VALUES TO INVALID
1.4
LOAD PARAMETERS FROM
EPROM
1.5
GENERATE CRC OF CODE
1.6
GET SERIAL NUMBER AND
SOFTWARE VERSION
ENABLE SPI SYSTEM
1.7
NULL COMMAND
RECEIVED?
NO
YES
ENABLE INTERRUPTS
2


1
2.1
SEND 3 FINE TUNE VALUES TO RECEIVERS
$1
MEASURE CONTROL VOLTAGE
OK?
NO
YES
2.2
$2
ENABLE EDFA
TURN LED OFF
CLR 1RY STATUS BIT
SAVE OPTICAL FAULT VALUES
SET 2RY STATUS BIT
$1
MEASURE TEMPERATURE
$1
MEASURE TEMPERATURE
OK?
NO
NO
OK?
YES
YES
$3
TURN PSUs ON
WAIT 100 mS
TURN LED OFF
CLR 2RY STATUS BIT
SAVE TEMPERATURE FAULT VALUES
SET 2RY STATUS BIT
3


2
4
MEASURE
VNEG VPOS TEMP
$1
3.1
Main PSUs
Temperature
OK?
NO
YES
PSUs
ON?
YES
NO
TURN PSUs OFF
SAVE FAULT VALUES
TURN LED ON
SET 1RY STATUS BIT
SET 2RY STATUS BIT
$4
LOOP 3
TIMES
MEASURE
VCON
$1
3.2
Control
Voltage
OK?
NO
YES
PSU
ON?
YES
NO
TURN PSUs OFF
SAVE FAULT VALUES
TURN LED ON
SET 1RY STATUS BIT
SET 2RY STATUS BIT
$4
EDFA
ON?
YES
NO
SEND EDFA OFF
SAVE FAULT VALUES
TURN LED ON
SET 1RY STATUS BIT
SET 2RY STATUS BIT
$5
4


3
MEASURE OPTICAL LEVELS 0,1,2
$6
OK?
NO
YES
EDFA ON?
YES
NO
SEND EDFA OFF
SAVE FAULT VALUES
TURN LED ON
SET 1RY STATUS BIT
SET 2RY STATUS BIT
$5
LOOP 3 TIMES
3.3 Optical Levels
INCREMENT LOOP COUNT
LCNT =256?
YES
NO
SEND EDFA OFF
SAVE FAULT VALUES
TURN LED ON
SET 1RY STATUS BIT
SET 2RY STATUS BIT
3.4 SPI Timeout (2.5Sec)
COMMAND REQUEST?
YES
NO
3.5 Command Handler
TMR1H >=25?
YES
NO
CLR TMR1
3.7 Loop Delay (10mS)
3


INTERRUPT
SAVE CONTEXT
FLASH TOP LED
BYTE RECEIVED?
NO
YES
3ls BITS TO ICNT
5ms BITS TO MODE
MODE = IDREQ?
YES
NO
SPI SEND REG := IDn
MODE = COMMAND
YES
6a
NO
TRANSFER COMPLETE
NO
YES
MODE= MONITOR
YES
6b
GET NEXT ID BYTE
NO
LOOP 3 TIMES
6c
INT LINE CLEAR?
NO
YES
RESET SPI SYSTEM
YES
DATA OVERRUN?
NO
CLEAR TIMEOUTCLEAR INTERRUPT FLAG
RESTORE CONTEXT
RETURN
4.1
4.2
4.3
4.6


5a
SPI SEND REG:= ACK
NO
BYTE RECEIVED?
YES
IADDR := SPI RX REG
SPI SEND REG:= NULL
NO
BYTE RECEIVED?
YES
NO
IADDR OK?
YES
SAVE RECEIVED BYTE AT IADDR
INCREMENT IADDR
LOOP ICNT TIMES
4.4
5b
SPI SEND REG:= ACK
NO
BYTE RECEIVED?
YES
IADDR := SPI RX REG
SPI SEND REG := BYTE AT IADDR
NO
TRANSFER COMPLETE?
YES
INCREMENT IADDR
LOOP 8 TIMES
4.5
5c

```
                     00001 ;===============================================================================
                     00002 ; EMULATOR should be defined when controlled by the laptop                     ;
                     00003 ;                 undefined running as a mezzanine to the digital board      ;
                     00004 ; TEST should be defined when using the test board for initial debugging      ;
                     00005 ;              undefined when using the real board                           ;
                     00006 ;                                                                            ;
                     00007  #DEFINE EMULATOR
                     00008 ; #DEFINE TEST
                     00009 ;
                     00010 ;===============================================================================
                     00011 ; Number of 256 word blocks to include in the CRC                ;
                     00012 ;                                        ;
                     00013 #DEFINE PROGBLOCKS 8
                     00014 ;
                     00015 ;===============================================================================
                     00016 ; This is the program version number  Should be the same as IDLOCS    ;
                     00017 ;                                                                            ;
                     00018         #DEFINE I_VN_M          0X04
                     00019         #DEFINE I_VN_L          0X01
                     00020 ;
                     00021 ;===============================================================================
                     00022
                     00023         LIST    P=16F877,N=0
                     00024     TITLE   " ALMA Optical Receiver microcontroller
                     00025
                     00026         INCLUDE         <D16F877.INC>   ;
                     00001 ; P16F877.INC  Standard Header File, Version 1.00    Microchip Technology Inc
                     00372         LIST
                     00027
                     00028 ;__CONFIG _CP_OFF & _DEBUG_OFF & _WRT_ENABLE_OFF & _LVP_ON & _BODEN_ON & _PWRTE_OFF & _WDT_ON &_XT_OSC

2007   3D32          00029  __CONFIG _CP_OFF & _DEBUG_OFF & _WRT_ENABLE_OFF & _LVP_OFF & _BODEN_OFF & _PWRTE_ON & _WDT_OFF & _HS_OSC

2000   0000 0004 0000 00030         __IDLOCS    0X0401
       0001
                     00031
                     00032         INCLUDE         <DCBMACRO.INC>
                     00001 ; DCBMACRO.INC   Simple macros.  Version 1.10    dcb
                     00337     LIST
                     00033 ;_____________________________________________________________________________
                     00034 ; NOTE ON PAGES AND BANKS
                     00035 ;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                     00036 ; All code resides on page zero so paging is not an issue;
                     00037 ; All used RAM is in bank zero so bank switching is only an issue for accessing
                     00038 ; Special Function Registers.  All modules and subroutines which access outside
                     00039 ; bank zero must return to bank zero on completion.
                     00040 ; Interrupt can occur when any bank is selected.  However the context registers
                     00041 ; (W,FSR & Status) are mapped across all banks and are saved to RAM above $70
                     00042 ; which is also mapped across all banks
                     00043
                     00044
```

```
00045 ; This is the number of bytes in a block read.  It must be 8 in the final system
00046  IFDEF EMULATOR
00047     #DEFINE BLOCKLEN    32
00048  ELSE
00049     #DEFINE BLOCKLEN    8
00050  ENDIF
00051
00052 ; These are the memory locations which are read by the M&C bus
00053 ; They are all in bank zero and are arranged in groups of eight
00054
00055     #DEFINE MONITOR_POINTS_BASE 0X20
00056     #DEFINE _STATUS1        0X20    ; Primary status
00057     #DEFINE _STATUS2        0X21    ; Secondary status
00058
00059     #DEFINE EXTERNAL_VALUES_BASE    0X22
00060     #DEFINE _OP0VAL_M       0X22    ; Current value - Optical Rx 0
00061     #DEFINE _OP0VAL_L       0X23
00062     #DEFINE _OP1VAL_M       0X24    ; Current value - Optical Rx 1
00063     #DEFINE _OP1VAL_L       0X25
00064     #DEFINE _OP2VAL_M       0X26    ; Current value - Optical Rx 2
00065     #DEFINE _OP2VAL_L       0X27
00066
00067     #DEFINE INTERNAL_VALUES_BASE    0X28
00068     #DEFINE _VNVAL_M        0X28    ; Current value - Negative PSU
00069     #DEFINE _VNVAL_L        0X29
00070     #DEFINE _VPVAL_M        0X2A    ; Current value - Positive PSU
00071     #DEFINE _VPVAL_L        0X2B
00072     #DEFINE _VCVAL_M        0X2C    ; Current value - Control PSU
00073     #DEFINE _VCVAL_L        0X2D
00074     #DEFINE _TMPVAL_M       0X2E    ; Current value - Temperature
00075     #DEFINE _TMPVAL_L       0X2F    ;
00076
00077     #DEFINE EXTERNAL_FAULTS_BASE    0X32
00078     #DEFINE _OP0FLT_M       0X32    ; Fault value - Optical Rx 0
00079     #DEFINE _OP0FLT_L       0X33    ;
00080     #DEFINE _OP1FLT_M       0X34    ; Fault value - Optical Rx 1
00081     #DEFINE _OP1FLT_L       0X35    ;
00082     #DEFINE _OP2FLT_M       0X36    ; Fault value - Optical Rx 2
00083     #DEFINE _OP2FLT_L       0X37    ;
00084
00085     #DEFINE INTERNAL_FAULT_BASE     0X38
00086     #DEFINE _VNFLT_M        0X38    ; Fault value - Negative PSU
00087     #DEFINE _VNFLT_L        0X39
00088     #DEFINE _VPFLT_M        0X3A    ; Fault value - Positive PSU
00089     #DEFINE _VPFLT_L        0X3B
00090     #DEFINE _VCFLT_M        0X3C    ; Fault value - Control PSU
00091     #DEFINE _VCFLT_L        0X3D
00092     #DEFINE _TMPFLT_M       0X3E    ; Fault value - Temperature
00093     #DEFINE _TMPFLT_L       0X3F    ;
00094
00095     #DEFINE _COMMAND    0X40
00096
00097     #DEFINE EEPROM_START    0X42
```

```
00098     #DEFINE EXTERNAL_LIMITS_BASE    0X42
00099     #DEFINE _OP1ULIM_M      0X42   ; Upper limit - Optical Rx 1
00100     #DEFINE _OP0LLIM        0X43   ; Lower Limit - Optical Rx 0
00101     #DEFINE _OP0ULIM_M      0X44   ; Upper limit - Optical Rx 0
00102     #DEFINE _OP1LLIM        0X45   ; Lower Limit - Optical Rx 1
00103     #DEFINE _OP2ULIM_M      0X46   ; Upper limit - Optical Rx 2
00104     #DEFINE _OP2LLIM        0X47   ; Lower Limit - Optical Rx 2
00105
00106     #DEFINE INTERNAL_LIMITS_BASE    0X48
00107     #DEFINE _VNULIM         0X48    ; Upper limit - Negative PSU
00108     #DEFINE _VNLLIM         0X49    ; Lower limit - Negative PSU
00109     #DEFINE _VPULIM         0X4A    ; Upper limit - Positive PSU
00110     #DEFINE _VPLLIM         0X4B    ; Lower limit - Positive PSU
00111     #DEFINE _VCULIM         0X4C    ; Upper limit - Control PSU
00112     #DEFINE _VCLLIM         0X4D    ; Lower limit - Control PSU
00113     #DEFINE _TMPULIM        0X4E    ; Upper limit - Temperature
00114     #DEFINE _TMPLLIM        0X4F    ; Lower limit - Temperature
00115
00116     #DEFINE FINE_TUNE_BASE          0X50
00117     #DEFINE _OP0FT_M        0X50    ; Fine Tune   - Optical Rx 0
00118     #DEFINE _OP0FT_L        0X51
00119     #DEFINE _OP1FT_M        0X52    ; Fine Tune   - Optical Rx 1
00120     #DEFINE _OP1FT_L        0X53
00121     #DEFINE _OP2FT_M        0X54    ; Fine Tune   - Optical Rx 2
00122     #DEFINE _OP2FT_L        0X55
00123
00124     #DEFINE CALIBRATION_BASE        0X56
00125     #DEFINE _OP0CAL1_M      0X56    ; 0dB cal     - Optical Rx 0
00126     #DEFINE _OP0CAL1_L      0X57
00127     #DEFINE _OP0CAL0        0X58    ; Zero cal    - Optical Rx 0
00128     #DEFINE _OP1CAL1_M      0X59    ; 0dB cal     - Optical Rx 1
00129     #DEFINE _OP1CAL1_L      0X5A
00130     #DEFINE _OP1CAL0        0X5B    ; Zero cal    - Optical Rx 1
00131     #DEFINE _OP2CAL1_M      0X5C    ; 0dB cal     - Optical Rx 2
00132     #DEFINE _OP2CAL1_L      0X5D
00133     #DEFINE _OP2CAL0        0X5E    ; Zero cal    - Optical Rx 2
00134
00135     #DEFINE _ID0            0X60    ; Identity Bytes
00136     #DEFINE _ID1            0X61
00137     #DEFINE _ID2            0X62
00138     #DEFINE _CHECKSUM       0X63
00139
00140
00141     #DEFINE _SN_M           0X64    ; Board Serial Number
00142     #DEFINE _SN_L           0X65
00143     #DEFINE _VN_M           0X66    ; Program version number
00144     #DEFINE _VN_L           0X67
00145
00146     #DEFINE _CRC_M          0X68    ; CRC of program
00147     #DEFINE _CRC_L      0X69
00148
00149
00150 ;================================================================================
```

```
00151 ; These are registers and bits used  by the program - mapped across all banks
00152
00153         #DEFINE RESERVED    0X70        ; for ICD
00154
00155     #DEFINE SCNT            0X71            ; Usually a counter in subroutines
00156     #DEFINE LCNT        0X72            ; Usually a counter in program
00157     #DEFINE SHFT_M      0X73
00158     #DEFINE SHFT_L      0X74
00159
00160     #DEFINE TEMP0       0X75
00161     #DEFINE TEMP1       0X76
00162
00163     #DEFINE CAL0        0X77        ; Used in calibration routine
00164     #DEFINE CAL1        0X78        ;
00165     #DEFINE CAL2        0X79        ;
00166     #DEFINE CCNT        0X7A        ;
00167
00168     #DEFINE TCNT        0X7B        ; Time out counter
00169
00170     #DEFINE ICNT        0X7C        ; Counter in the interrupt routine
00171     #DEFINE WSAVE       0X7D        ; Save W in interrupt
00172     #DEFINE SSAVE       0X7E        ; Save Status in interrupt
00173     #DEFINE FSAVE       0X7F        ; Save FSR in interrupt
00174
00175 ;=============================================================================
00176 ; These are names for the bits in the ports
00177 ;
00178     #DEFINE ANALOG0     PORTA,0 ;Analog inputs (multiplexed)        a
00179     #DEFINE ANALOG1     PORTA,1 ;|                                  a
00180     #DEFINE ASEL0       PORTE,0 ;Analog multiplexer selects         o
00181     #DEFINE ASEL1       PORTE,1 ;|                                  o
00182
00183     #DEFINE ZCSIN       PORTB,0 ;Copy of A5 for interrupt
00184     #DEFINE ZCSN        PORTA,5 ;Hardware SPI bus (to AMBSI)        i
00185     #DEFINE ZCLK        PORTC,3 ;|                                  o
00186     #DEFINE ZDI         PORTC,4 ;|                                  i
00187     #DEFINE ZDO         PORTC,5 ;|                                  o
00188
00189     #DEFINE DCLK        PORTC,0 ;Software SPI bus (ADC and DAC)     o
00190     #DEFINE DDI         PORTC,1 ;|                                  i
00191     #DEFINE DDO         PORTC,2 ;|                                  o
00192     #DEFINE ACSN        PORTC,6 ;|                                  o
00193     #DEFINE DCSN        PORTC,7 ;|                                  o
00194
00195     #DEFINE VPENN       PORTB,1 ;Positive PSU enable                o
00196     #DEFINE VNENN       PORTB,2 ;Negative PSU enable                o
00197     #DEFINE EDFASD      PORTB,4 ;EDFA shut down line                o
00198
00199     #DEFINE SNREQN      PORTB,5 ;Request serial number              o
00200     #DEFINE SNIN        PORTE,2 ;Read serial  number                i
00201
00202     #DEFINE LEDSN       PORTD   ; LEDs                              o
00203     #DEFINE EDFALEDN    PORTD,0 ;
```

```
00204     #DEFINE PSULEDN     PORTD,1 ;
00205
00206 ;=============================================================================
00207 ; These are constants for the standard PSU shutdown modes.  Values 0-3 are
00208 ; also the analog multiplexer control codes
00209
00210     #DEFINE PSU_VN      0   ; Vneg fault
00211     #DEFINE PSU_VP      1   ; Vpos fault
00212     #DEFINE PSU_VC      2   ; Vcon fault
00213     #DEFINE PSU_TMP     3   ; Temp fault
00214     #DEFINE PSU_MC      4   ; M&C command
00215
00216 ; These are constants for the EDFA shutdown modes
00217     #DEFINE EDF_R0      0   ; Receiver 0 fault
00218     #DEFINE EDF_R1      1   ; Receiver 1 fault
00219
00220     #DEFINE EDF_R2      2   ; Receiver 2 fault
00221     #DEFINE EDF_VC      3   ; Control supply fault
00222     #DEFINE EDF_MC      4   ; M&C command
00223     #DEFINE EDF_TO      5   ; Time out
00224
00225 ; These are constants for the secondary status codes for PSU no start
00226
00227     #DEFINE ST2P_VCX    0X42    ; Control voltage at startup
00228     #DEFINE ST2P_TMPX   0X01    ; Temperature at startup
00229
00230 ; These are individual bits in the primary status word
00231
00232     #DEFINE ST1_EDFA    _STATUS1,7      ; Optical shutdown commanded
00233     #DEFINE ST1_PSU     _STATUS1,6      ; The main PSUs are off
00234     #DEFINE ST1_NOSN    _STATUS1,5      ; Set if serial number is invalid
00235     #DEFINE ST1_EECSF   _STATUS1,4  t when EEPROM reads bad checksum
00236
00237
00238 ;=============================================================================
00239 ; These are constants for the M&C command codes.
00240
00241     #DEFINE CMD_START   0X00
00242
00243 ; These are constants for the M&C protocol control codes.
00244
00245     #DEFINE MC_IDR      0XB0
00246     #DEFINE MC_CMD      0XA8
00247     #DEFINE MC_MON      0XA0
00248     #DEFINE MC_ACK      0X59
00249
00250 ;=============================================================================
00251 ; These are initialization values for configuration registers and ports
00252
00253     #DEFINE I_TRISA         0XFF    ; Port A is all inputs
00254     #DEFINE I_TRISB         0X01    ; Port B0 is int input
00255     #DEFINE I_TRISC         0X1A    ; PortC bits 1,3,4 are inputs
00256     #DEFINE I_TRISD         0X00    ; PortD all outputs
```

```
00257     #DEFINE I_TRISE         0X04    ; PortE bit 2 is input
00258
00259
00260     #DEFINE I_PORTB         0XFF    ; PSUENn, EDFAn, SNREQn off
00261     #DEFINE I_PORTC         0XC0    ; CSn, DOUTs,CLKs off
00262     #DEFINE I_PORTD         0XFC    ; PSULEDn, EDFALEDn on. Others off
00263     #DEFINE I_PORTE         0XF0    ; Analog Multiplexer at zero
00264
00265     #DEFINE I_ADCON0        0X80    ; ADC - Fo/32; chan 0; all off
00266     #DEFINE I_ADCON1        0X85   ; ADC - right justified; mode 5
00267
00268     #DEFINE I_SSPSTAT       0X00    ; Falling edge clock. BF cleared
00269     #DEFINE I_SSPCON        0X04    ; Fault bits cleared. SPI disabled
00270                           ; Clock idle low.  SPI mode 4
00271
00272     #DEFINE I_INTCON        0X10    ; External interrupts enabled
00273
00274     #DEFINE I_OPTION        0X0F    ; Pull ups enabled ; RB0 falling edge
00275                                       ; Post scale WDT by 128 (896-4224 mS)
00276     #DEFINE I_T1CON     0X31    ; Prescaler=8 ; External oscillator off
00277                       ; Internal clock ; enabled
```

```
                    00278 ;================================================================================
0000                00279     ORG 0
0000   2900         00280     GOTO    START
                    00281
0004                00282     ORG 0X004
                    00283     INCLUDE INTS.INC

                    00001         PAGE
                    00002         SUBTITLE " Interrupt handler "
                    00003
                    00004 ; This section handles the interrupt which can only be caused by the
                    00005 ; rising edge of the chip select line at the start of an SPI transfer
                    00006 ;
                    00007 ;################################################################################;
                    00008 ;                                                                                ;
                    00009 ; This macro is to send and receive a byte on the SPI                            ;
                    00010 ; W contains the byte to send and is replaced by the received byte               :
                    00011 ;________________________________________________________________________________;
                    00012
                    00013 SPIT    MACRO
                    00014         LOCAL   SPIT0
                    00015
                    00016         MOVWF   SSPBUF
                    00017         BANK1
                    00018 SPIT0   BTFSS   SSPSTAT,BF
                    00019         GOTO    SPIT0
                    00020         BANK0
                    00021         MOVFW   SSPBUF
                    00022
                    00023         ENDM;
                    00024
                    00025 ;################################################################################;
                    00026 ;                                                                                ;
                    00027 ; First section saves the W,STATUS and FSR registers and clears the flag         ;
                    00028 ; and flashes the top LED                                                        ;
                    00029 ;________________________________________________________________________________;
                    00030
                    00031
0004   00FD         00032         MOVWF   WSAVE                   ; Save the context
0005   0E03         00033         SWAPF   STATUS,W
0006   00FE         00034         MOVWF   SSAVE
0007   0804         00035         MOVF    FSR,W
0008   00FF         00036         MOVWF   FSAVE
                    00037
                    00038         BANK00
0009   1283             M     BCF STATUS,RP0
000A   1303             M     BCF STATUS,RP1
000B   3080         00039         MOVLW   0X80
000C   0688         00040         XORWF   PORTD,F
                    00041
                    00042 ;================================================================================;
                    00043 ;                                                                                ;
                    00044 ; Wait for a byte to be received.  Save the bottom three bits which might be ;
```

```
                    00045 ; the count.  Then branch on the top five bits to the appropriate transfer   ;
                    00046 ; type.  If unrecognized then do nothing                                     ;
                    00047 ;____________________________________________________________________________;
                    00048
                    00049         BANK1
000D   1683             M     BSF STATUS,RP0
000E   1C14         00050 M7_L0   BTFSS   SSPSTAT,BF              ; Wait for byte to be received
000F   280E         00051         GOTO    M7_L0
                    00052
                    00053         BANK0
0010   1283             M     BCF STATUS,RP0
                    00054         ANDLF   0X07,SSPBUF,W           ; Extract the bottom three bit
0011   3007             M     MOVLW   0X07
0012   0513             M     ANDWF   SSPBUF,W
0013   00FC         00055         MOVWF   ICNT                    ; and save in ICNT
                    00056         ANDLF   0XF8,SSPBUF,F           ; and the top five bits
0014   30F8             M     MOVLW   0XF8
0015   0593             M     ANDWF   SSPBUF,F
                    00057
                    00058         SKPNEL  SSPBUF,MC_IDR
0016   30B0             M     MOVLW   0XB0    ; w := k
0017   0213             M     SUBWF   SSPBUF,W    ; w := reg1-k
0018   1903             M     SKPNZ           ;
0019   2823         00059         GOTO    M7_ID
                    00060
                    00061         SKPNEL  SSPBUF,MC_CMD           ; Or command (write to PIC)
001A   30A8             M     MOVLW   0XA8    ; w := k
001B   0213             M     SUBWF   SSPBUF,W    ; w := reg1-k
001C   1903             M     SKPNZ           ;
001D   2832         00062         GOTO    M7_CD
                    00063
                    00064         SKPNEL  SSPBUF,MC_MON           ; Or monitor (read from PIC)
001E   30A0             M     MOVLW   0XA0    ; w := k
001F   0213             M     SUBWF   SSPBUF,W    ; w := reg1-k
0020   1903             M     SKPNZ           ;
0021   284B         00065         GOTO    M7_MN
                    00066
0022   285F         00067         GOTO    M7_NXT                  ; If none of these goto end
                    00068
                    00069
                    00070 ;============================================================================;
                    00071 ;                                                                            ;
                    00072 ; An ID transfer.  Return the three ID bytes in response to the next        ;
                    00073 ; three transfers.  The received data can be ignored                         ;
                    00074 ;____________________________________________________________________________;
                    00075
                    00076 M7_ID   MOVLF   _ID0,FSR                ; FSR points at first ID byte
0023   3060             M     MOVLW   0X60
0024   0084             M     MOVWF   FSR
                    00077         MOVLF   3,ICNT                  ; Three ID bytes to send
0025   3003             M     MOVLW   3
0026   00FC             M     MOVWF   0X7C
                    00078
```

```
0027   0800           00079 M7_ID1  MOVFW   INDF                    ; ID byte to SSPBUF wait until gone
                      00080         SPIT                            ; Not concerned with rx data
  0000                    M         LOCAL   SPIT0
                          M
0028   0093               M         MOVWF   SSPBUF
                          M         BANK1
0029   1683               M     BSF STATUS,RP0
002A   1C14               M SPIT0   BTFSS   SSPSTAT,BF
002B   282A               M         GOTO    SPIT0
                          M         BANK0
002C   1283               M     BCF STATUS,RP0
002D   0813               M         MOVFW   SSPBUF
                          M
                      00081
002E   0A84           00082         INCF    FSR
002F   0BFC           00083         DECFSZ  ICNT
0030   2827           00084         GOTO    M7_ID1
                      00085
0031   285F           00086         GOTO    M7_NXT
                      00087 ;=============================================================================;
                      00088 ;                                                                             ;
                      00089 ; A command transfer.  The count-1 has been stored in ICNT and the start     ;
                      00090 ; address offset from the base is the next byte received.  ICNT bytes of data;
                      00091 ; are then received and written to incrementing addresses.  If the address is;
                      00092 ; not in the range 0x40-0x5F the write is inhibited                           ;
                      00093 ;_____________________________________________________________________________;
                      00094
0032   3059           00095 M7_CD   MOVLW   MC_ACK                  ; Send an acknowledge and wait for
                      00096         SPIT                            ; receipt of address in W
  0000                    M         LOCAL   SPIT0
                          M
0033   0093               M         MOVWF   SSPBUF
                          M         BANK1
0034   1683               M     BSF STATUS,RP0
0035   1C14               M SPIT0   BTFSS   SSPSTAT,BF
0036   2835               M         GOTO    SPIT0
                          M         BANK0
0037   1283               M     BCF STATUS,RP0
0038   0813               M         MOVFW   SSPBUF
                          M
                      00097
                      00098 ;       ANDLW   0x7F                    ; only seven bit addresses
                      00099 ;       ADDLF   MONITOR_POINTS_BASE,FSR,F       ; referring to base
                      00100 ;       NOP
                      00101 ;       NOP
0039   0084           00102         MOVWF   FSR                     ; FSR points at the address
003A   0AFC           00103         INCF    ICNT,F
                      00104
003B   3000           00105 M7_CD1  MOVLW   0                       ; Null data to SSPBUF wait until gone
                      00106         SPIT                            ; W now contains received byte
  0000                    M         LOCAL   SPIT0
                          M
003C   0093               M         MOVWF   SSPBUF
```

```
                         M         BANK1
003D   1683              M     BSF STATUS,RP0
003E   1C14              M SPIT0   BTFSS   SSPSTAT,BF
003F   283E              M         GOTO    SPIT0
                         M         BANK0
0040   1283              M     BCF STATUS,RP0
0041   0813              M         MOVFW   SSPBUF
                         M
                       00107
0042   1F04            00108         BTFSS   FSR,6                   ; Only permitted to write to address
0043   2847            00109         GOTO    M7_CD2                  ; 0x4- and 0x5-.  That is bit 6 set
0044   1A84            00110         BTFSC   FSR,5                   ; and bit 5 clear
0045   2847            00111         GOTO    M7_CD2
                       00112
0046   0080            00113         MOVWF   INDF                    ; Save byte in addressed location
                       00114
0047   0A84            00115 M7_CD2  INCF    FSR                     ; Transfer size was defined
0048   0BFC            00116         DECFSZ  ICNT                    ; in low three bits of control byte
0049   283B            00117         GOTO    M7_CD1
                       00118
004A   285F            00119         GOTO    M7_NXT
                       00120
                       00121 ;=================================================================================;
                       00122 ;                                                                                 ;
                       00123 ; A monitor transfer.  The count is fixed at forty bytes and the start            ;
                       00124 ; address offset from the base is the next byte received. 40 bytes of data        ;
                       00125 ; are then received and written to incrementing addresses.                        ;
                       00126 ;_________________________________________________________________________________;
                       00127
                       00128
004B   3059            00129 M7_MN   MOVLW   MC_ACK                   ; Send an acknowledge and wait for
                       00130         SPIT                            ; receipt of address
  0000                     M         LOCAL   SPIT0
                           M
004C   0093                M         MOVWF   SSPBUF
                           M         BANK1
004D   1683                M     BSF STATUS,RP0
004E   1C14                M SPIT0   BTFSS   SSPSTAT,BF
004F   284E                M         GOTO    SPIT0
                           M         BANK0
0050   1283                M     BCF STATUS,RP0
0051   0813                M         MOVFW   SSPBUF
                           M
                       00131
0052   0084            00132         MOVWF   FSR
                       00133         MOVLF   BLOCKLEN,ICNT                   ; TESTMODE all monitor transfers are 36 bytes
0053   3020                M     MOVLW   32
0054   00FC                M     MOVWF   0X7C
                       00134
0055   0800            00135 M7_MN1  MOVFW   INDF                    ; Data byte to SSPBUF wait until gone
                       00136         SPIT                            ; Not concerned with rx data
  0000                     M         LOCAL   SPIT0
                           M
```

```
0056   0093              M         MOVWF   SSPBUF
                         M         BANK1
0057   1683              M     BSF STATUS,RP0
0058   1C14              M SPIT0   BTFSS   SSPSTAT,BF
0059   2858              M         GOTO    SPIT0
                         M         BANK0
005A   1283              M     BCF STATUS,RP0
005B   0813              M         MOVFW   SSPBUF
                         M
                      00137
005C   0A84           00138         INCF    FSR
005D   0BFC           00139         DECFSZ  ICNT
005E   2855           00140         GOTO    M7_MN1
                      00141
                      00142 ;=================================================================================;
                      00143 ;                                                                                 ;
                      00144 ; Clear the timeout counter so program can measure time between interrupts        ;
                      00145 ; Wait until the transfer has ended as signaled by chip select going high         ;
                      00146 ; Restore the saved values of W,STATUS,FSR and return                             ;
                      00147 ;_________________________________________________________________________________;

                      00148 ;
005F   1C06           00149 M7_NXT  BTFSS   ZCSIN                   ; Wait till end of transfer
0060   285F           00150         GOTO    M7_NXT
                      00151
0061   1F14           00152         BTFSS   SSPCON,SSPOV            ; Test for data overrun
0062   2867           00153         GOTO    M7_NOR                  ; no overrun
                      00154
                      00155         MOVLF   I_SSPCON,SSPCON         ; Reset the SP system
0063   3004              M     MOVLW   0X04
0064   0094              M     MOVWF   SSPCON
0065   0813           00156         MOVFW   SSPBUF                  ; Clear the overrun data
0066   1694           00157         BSF     SSPCON,SSPEN            ; Reenable the SPI
                      00158
0067   01FB           00159 M7_NOR  CLRF    TCNT                    ; Clear the timeout counter
0068   108B           00160         BCF     INTCON,INTF             ; Clear the interrupt flag
                      00161
0069   087F           00162         MOVF    FSAVE,W                 ; Restore context
006A   0084           00163         MOVWF   FSR
006B   0E7E           00164         SWAPF   SSAVE,W
006C   0083           00165         MOVWF   STATUS
006D   0EFD           00166         SWAPF   WSAVE,F
006E   0E7D           00167         SWAPF   WSAVE,W
                      00168
006F   0009           00169         RETFIE                          ; and return
                      00170
                      00171
                      00172
                      00284
0100                  00285         ORG     0X100
0100                  00286 START
                      00287     INCLUDE INITIAL.INC
```

```
                     00001         PAGE
                     00002         SUBTITLE " Initial code up to contact"
                     00003
                     00004 ; This MODULE contains the code that runs after a restart until receiving
                     00005 ; a NULL request from the SPI
                     00006
                     00007 ;### SECTION 1-1 ##################################################################;
                     00008 ;                                                                                  ;
                     00009 ; Sets the Status words according to the reset mode                               ;
                     00010 ;__________________________________________________________________________________;
                     00011
                     00012         BANK00                  ;
0100   1283              M     BCF STATUS,RP0
0101   1303              M     BCF STATUS,RP1
                     00013
0102   01C0          00014         CLRF    _COMMAND
0103   01A1          00015         CLRF    _STATUS2        ; Clear the secondary status word
                     00016
                     00017         MOVLF   0XF1,_STATUS1   ; Set reset status in primary  word to 1
0104   30F1              M     MOVLW   0XF1
0105   00A0              M     MOVWF   0X20
                     00018                                 ; PSU and EDFA and NOSN to ones
                     00019         BANK1
0106   1683              M     BSF STATUS,RP0
0107   1C8E          00020         BTFSS   PCON,PORN       ; PCON is in bank 1, STATUS is mapped
0108   2912          00021         GOTO    M11_POR
0109   1C0E          00022         BTFSS   PCON,BORN
010A   2910          00023         GOTO    M11_BOR
010B   1E03          00024         BTFSS   STATUS,TON
010C   290E          00025         GOTO    M11_WDT
010D   2914          00026         GOTO    M11_MCR
                     00027
                     00028 M11_WDT BANK0
010E   1283              M     BCF STATUS,RP0
010F   0AA0          00029         INCF    _STATUS1,F
                     00030 M11_BOR BANK0
0110   1283              M     BCF STATUS,RP0
0111   0AA0          00031         INCF    _STATUS1,F
                     00032 M11_POR BANK0
0112   1283              M     BCF STATUS,RP0
0113   0AA0          00033         INCF    _STATUS1,F
                     00034
                     00035 M11_MCR BANK1                   ; PCON is in bank 1, STATUS is mapped
0114   1683              M     BSF STATUS,RP0
0115   148E          00036         BSF     PCON,PORN
0116   140E          00037         BSF     PCON,BORN
0117   1603          00038         BSF     STATUS,TON
                     00039
                     00040 ;=== SECTION 1-2 ==========================================================;
                     00041 ;                                                                           ;
                     00042 ; Sets the configuration registers and sets all output pins to their        ;
                     00043 ; inactive values                                                           ;
                     00044 ;___________________________________________________________________________;
```

```
                     00045
                     00046          BANK0
0118   1283              M      BCF STATUS,RP0
                     00047
                     00048          MOVLF   I_PORTB,PORTB
0119   30FF              M      MOVLW   0XFF
011A   0086              M      MOVWF   PORTB
                     00049          MOVLF   I_PORTC,PORTC
011B   30C0              M      MOVLW   0XC0
011C   0087              M      MOVWF   PORTC
                     00050          MOVLF   I_PORTD,PORTD
011D   30FC              M      MOVLW   0XFC
011E   0088              M      MOVWF   PORTD
                     00051          MOVLF   I_PORTE,PORTE
011F   30F0              M      MOVLW   0XF0
0120   0089              M      MOVWF   PORTE
                     00052
                     00053          MOVLF   I_INTCON,INTCON
0121   3010              M      MOVLW   0X10
0122   008B              M      MOVWF   INTCON
                     00054          MOVLF   I_ADCON0,ADCON0
0123   3080              M      MOVLW   0X80
0124   009F              M      MOVWF   ADCON0
                     00055          MOVLF   I_SSPCON,SSPCON
0125   3004              M      MOVLW   0X04
0126   0094              M      MOVWF   SSPCON
                     00056          MOVLF   I_T1CON,T1CON
0127   3031              M      MOVLW   0X31
0128   0090              M      MOVWF   T1CON
                     00057
                     00058          BANK1
0129   1683              M      BSF STATUS,RP0
                     00059
                     00060          MOVLF   I_OPTION,OPTION_REG
012A   300F              M      MOVLW   0X0F
012B   0081              M      MOVWF   OPTION_REG
                     00061          MOVLF   I_ADCON1,ADCON1
012C   3085              M      MOVLW   0X85
012D   009F              M      MOVWF   ADCON1
                     00062          MOVLF   I_SSPSTAT,SSPSTAT
012E   3000              M      MOVLW   0X00
012F   0094              M      MOVWF   SSPSTAT
                     00063
                     00064          MOVLF   I_TRISA,TRISA
0130   30FF              M      MOVLW   0XFF
0131   0085              M      MOVWF   TRISA
                     00065          MOVLF   I_TRISB,TRISB
0132   3001              M      MOVLW   0X01
0133   0086              M      MOVWF   TRISB
                     00066          MOVLF   I_TRISC,TRISC
0134   301A              M      MOVLW   0X1A
0135   0087              M      MOVWF   TRISC
                     00067          MOVLF   I_TRISD,TRISD
```

```
0136   3000              M     MOVLW   0X00
0137   0088              M     MOVWF   TRISD
                       00068         MOVLF   I_TRISE,TRISE
0138   3004              M     MOVLW   0X04
0139   0089              M     MOVWF   TRISE
                       00069
013A   0000            00070         NOP
                       00071
                       00072 ;=== SECTION 1-4 ================================================================;
                       00073 ;                                                                                 ;
                       00074 ; Clears actual and fault value locations TO INVALID                              ;
                       00075 ;_________________________________________________________________________________;
                       00076
                       00077         MOVLF   30,SCNT
013B   301E              M     MOVLW   30
013C   00F1              M     MOVWF   0X71
                       00078         MOVLF   EXTERNAL_VALUES_BASE,FSR
013D   3022              M     MOVLW   0X22
013E   0084              M     MOVWF   FSR
                       00079
013F   30FF            00080         MOVLW   0XFF
0140   0080            00081 M14_1   MOVWF   INDF
0141   0A84            00082         INCF    FSR,F
0142   0BF1            00083         DECFSZ  SCNT
0143   2940            00084         GOTO    M14_1
                       00085
0144   14A8            00086         BSF     _VNVAL_M,1
                       00087
                       00088 ;=== SECTION 1-4 ================================================================;
                       00089 ;                                                                                 ;
                       00090 ; Copies parameters and limits from the EEPROM into the M&C registers.            ;
                       00091 ;_________________________________________________________________________________;
                       00092
0145   26AB            00093         CALL    EEPROM_READ
0146   0000            00094         NOP
                       00095 ;
                       00096 ;=== SECTION 1-4 ================================================================;
                       00097 ;                                                                                 ;
                       00098 ; Generates CRC of Code                                          .                ;
                       00099 ;_________________________________________________________________________________;
                       00100
0147   3008            00101         MOVLW   PROGBLOCKS
0148   25AE            00102         CALL    CRC
0149   0000            00103         NOP
                       00104
                       00105 ;=== SECTION 1-3 ================================================================
                       00106 ;                                                                                 ;
                       00107 ; loads the version number anT1d gets the serial number from the  SN pin.      ;
                       00108 ; Time out checks are made to ensure that a serial number is being read but ;
                       00109 ; do not check that activity is within spec.                                      ;
                       00110 ;_________________________________________________________________________________;
                       00111
014A   0000            00112         NOP
```

```
                     00113
                     00114          BANK0
014B   1283              M      BCF STATUS,RP0
                     00115          MOVLF   I_VN_M,_VN_M    ; Put version number in M&C reg
014C   3004              M      MOVLW   0X04
014D   00E6              M      MOVWF   0X66
                     00116          MOVLF   I_VN_L,_VN_L
014E   3001              M      MOVLW   0X01
014F   00E7              M      MOVWF   0X67
                     00117
                     00118
0150   01E4          00119          CLRF    _SN_M
0151   01E5          00120          CLRF    _SN_L
                     00121
                     00122          MOVLF   8,SCNT
0152   3008              M      MOVLW   8
0153   00F1              M      MOVWF   0X71
0154   1286          00123          BCF     SNREQN          ; Request the serial number by resetting the SN-PIC
0155   0BF1          00124 M13_RST  DECFSZ  SCNT,F          ; for 24 cycles (4.8uS)
0156   2955          00125          GOTO    M13_RST
0157   1686          00126          BSF     SNREQN
                     00127
                     00128          MOVLF   4,LCNT          ; Allow an extra 1mS for the SN-PIC to send the first low
0158   3004              M      MOVLW   4
0159   00F2              M      MOVWF   0X72
015A   01F1          00129 S13_FL0  CLRF    SCNT
015B   1D09          00130 S13_FL1  BTFSS   SNIN
015C   2961          00131          GOTO    S13_FL2
015D   0BF1          00132          DECFSZ  SCNT,F
015E   295B          00133          GOTO    S13_FL1
015F   0BF2          00134          DECFSZ  LCNT,F
0160   295A          00135          GOTO    S13_FL0
                     00136
                     00137 S13_FL2  MOVLF   0X10,LCNT
0161   3010              M      MOVLW   0X10
0162   00F2              M      MOVWF   0X72
                     00138
                     00139 S13_LOP  MOVLF   100,SCNT
0163   3064              M      MOVLW   100
0164   00F1              M      MOVWF   0X71
0165   1D09          00140 S13_L1   BTFSS   SNIN            ; Wait for pin to go low on subsequent bits
0166   296A          00141          GOTO    S13_B0          ; Fail if not seen within 100 loops
0167   0BF1          00142          DECFSZ  SCNT,F          ; of 5 cycles ~ 100uS
0168   2965          00143          GOTO    S13_L1
0169   297F          00144          GOTO    S13_NOS
                     00145
                     00146 S13_B0   MOVLF   50,SCNT         ; Wait 50*5 cycles = 50uS to get to
016A   3032              M      MOVLW   50
016B   00F1              M      MOVWF   0X71
016C   0000          00147 S13_B1   NOP
016D   0000          00148          NOP
016E   0BF1          00149          DECFSZ  SCNT,F          ; centre of bit cell
016F   296C          00150          GOTO    S13_B1
```

```
                    00151
0170   1003         00152         BCF     STATUS,C        ; Copy the bit value into carry
0171   1909         00153         BTFSC   SNIN
0172   1403         00154         BSF     STATUS,C
                    00155
0173   0DE5         00156         RLF     _SN_L,F         ; Double length left shift
0174   0DE4         00157         RLF     _SN_M,F         ; picking up carry
                    00158
                    00159
                    00160         MOVLF   50,SCNT
0175   3032             M     MOVLW   50
0176   00F1             M     MOVWF   0X71
0177   1909         00161 S13_H1  BTFSC   SNIN            ; Wait for pin to go high
0178   297C         00162         GOTO    S13_NXT         ; Fail if not seen within 100 loops
0179   0BF1         00163         DECFSZ  SCNT,F          ; of 5 cycles = ~ 100uS
017A   2977         00164         GOTO    S13_H1
017B   297F         00165         GOTO    S13_NOS
                    00166
017C   0BF2         00167 S13_NXT DECFSZ  LCNT,F
017D   2963         00168         GOTO    S13_LOP
                    00169
017E   12A0         00170         BCF     ST1_NOSN
                    00171
017F   0000         00172 S13_NOS NOP
                    00173
                    00174
                    00175
                    00176
                    00177 ;=== SECTION 1-5 ================================================================
                    00178 ;                                                                               ;
                    00179 ; Initialise the SPI system and wait for a NULL transfer to be sent.            ;
                    00180 ;_______________________________________________________________________________;
                    00181
                    00182         BANK0
0180   1283             M     BCF STATUS,RP0
0181   1694         00183         BSF     SSPCON,SSPEN            ; Enable SSI
                    00184
                    00185 S15_1   MOVLF   BLOCKLEN,LCNT           ; Null transfer is 2 more than
0182   3020             M     MOVLW   32
0183   00F2             M     MOVWF   0X72
                    00186         ADDLF   2,LCNT,F                        ; longest block (10 bytes in final)
0184   3002             M     MOVLW   2
0185   07F2             M     ADDWF   0X72,F
                    00187
0186   3059         00188 S15_3   MOVLW   MC_ACK                  ; Send anything
                    00189         SPIT                            ; Wait for received data
  0000                  M         LOCAL   SPIT0
                        M
0187   0093             M         MOVWF   SSPBUF
                        M         BANK1
0188   1683             M     BSF STATUS,RP0
0189   1C14             M SPIT0   BTFSS   SSPSTAT,BF
018A   2989             M         GOTO    SPIT0
```

```
                         M         BANK0
018B   1283              M     BCF STATUS,RP0
018C   0813              M         MOVFW   SSPBUF
                         M
018D   3800           00190         IORLW   0                       ; To test W
018E   1D03           00191         SKPZ
018F   2982           00192         GOTO    S15_1                    ; If not zero then start again
0190   0BF2           00193         DECFSZ  LCNT
0191   2986           00194         GOTO    S15_3
                      00195
0192   01FB           00196         CLRF    TCNT                    ; Clear the time out counter
0193   1C06           00197 S15_2   BTFSS   ZCSIN
0194   2993           00198         GOTO    S15_2                   ; Wait until CS/ clears
0195   0000           00199         NOP
0196   0000           00200         NOP
0197   108B           00201         BCF     INTCON,INTF             ; clear the external interrupt flag
0198   0000           00202         NOP
0199   0000           00203         NOP
019A   178B           00204         BSF     INTCON,GIE              ; then enable interrupts
                      00205
                      00206
                      00207
                      00288     INCLUDE STARTUP.INC

                      00001         PAGE
                      00002         SUBTITLE " Startup PSUs and EDFA"
                      00003 ;
                      00004 ; This MODULE performs initial measurements before running the main loop
                      00005 ; First set up the DAC converters in the Receiver control loop.
                      00006 ; Then check the control voltage and temperature and if is safe to  turn on
                      00007 ; the powers supplies and enable the EDFA dos so
                      00008 ;
                      00009 ;################################################################################;
                      00010 ;                                                                               ;
                      00011 ; This macro sends the tweak values to the three external DAC channels          ;
                      00012 ;_______________________________________________________________________________;
                      00013
                      00014 TWEAK   MACRO
                      00015         LOCAL   TWK0
                      00016
                      00017         CLRF    LCNT            ;
                      00018
                      00019 TWK0    MOVFW   LCNT            ; W contains channel number
                      00020         CALL    XDAC
                      00021
                      00022         INCF    LCNT,F          ; 3 units plus initiate command
                      00023         SKPGEL  LCNT,4
                      00024         GOTO    TWK0
                      00025
                      00026         ENDM
                      00027
                      00028 ;=== SECTION 2-1 ========================================================;
                      00029 ;                                                                           ;
```

```
                    00030 ; Send the tweak values to the three external DAC channels                 ;
                    00031 ;_____________________________________________________________________________;
                    00032
                    00033         TWEAK
  0000                  M         LOCAL   TWK0
                        M
019B   01F2             M         CLRF    LCNT            ;
                        M
019C   0872             M TWK0    MOVFW   LCNT            ; W contains channel number
019D   25D8             M         CALL    XDAC
                        M
019E   0AF2             M         INCF    LCNT,F          ; 3 units plus initiate command
                        M         SKPGEL  LCNT,4
019F   3004             M     MOVLW   4       ; w := k
01A0   0272             M     SUBWF   0X72,W      ; w := reg1-k
01A1   1C03             M     SKPC            ; carry for +0 : reg1 >= k
01A2   299C             M         GOTO    TWK0
                        M
01A3   0000         00034         NOP
                    00035
                    00036
                    00037 ;=== SECTION 2-2 ============================================================;
                    00038 ;                                                                             ;
                    00039 ; Measure the control voltage and the temperature.                             ;
                    00040 ; If both are correct turn on the PSUs.                                       ;
                    00041 ; It is safe to release the EDFA if only the control voltage is correct       ;
                    00042 ;_____________________________________________________________________________;
                    00043
01A4   3002         00044         MOVLW   PSU_VC          ; Measure the control voltage
01A5   263D         00045         CALL    IADC
01A6   1903         00046         SKPNZ
01A7   29AF         00047         GOTO    M22_VF          ; If out of limit goto volt fault
                    00048
01A8   26E9         00049         CALL    EDFA_ON         ; Release the shutdown line
                    00050                                 ; Turn off the LED  Clear the status bit
                    00051
01A9   3003         00052         MOVLW   PSU_TMP         ; Measure the temperature
01AA   263D         00053         CALL    IADC
01AB   1903         00054         SKPNZ
01AC   29B9         00055         GOTO    M22_TF          ; If out of limit goto temp fault
                    00056
                    00057
01AD   26EF         00058         CALL    PSU_ON          ; Turn the PSUs on  delay 100mS
01AE   29BF         00059         GOTO    M22_NXT         ; Turn off the LED ; Clear the status bits
                    00060
                    00061 ;--- Control Voltage Fault -------------------------------------------------
                    00062
                    00063 M22_VF  MOVFF   _VCVAL_M,_VCFLT_M       ; Save failing value
01AF   082C             M     MOVF    0X2C,W
01B0   00BC             M     MOVWF   0X3C
                    00064         MOVFF   _VCVAL_L,_VCFLT_L
01B1   082D             M     MOVF    0X2D,W
01B2   00BD             M     MOVWF   0X3D
```

```
                    00065         IORLF   ST2P_VCX,_STATUS2,F     ; Update Secondary status
01B3   3042             M     MOVLW   0X42
01B4   04A1             M     IORWF   0X21,F
                    00066
01B5   3003         00067         MOVLW   PSU_TMP         ; Measure the temperature
01B6   263D         00068         CALL    IADC
01B7   1D03         00069         SKPZ                    ; If out of limit skip to temp fault
01B8   29BF         00070         GOTO    M22_NXT
                    00071
                    00072 ;--- Temperature Fault ----------------------------------------------------
                    00073
                    00074 M22_TF  MOVFF   _TMPVAL_M, _TMPFLT_M    ; Save failing values
01B9   082E             M     MOVF    0X2E,W
01BA   00BE             M     MOVWF   0X3E
                    00075         MOVFF   _TMPVAL_L, _TMPFLT_L    ;
01BB   082F             M     MOVF    0X2F,W
01BC   00BF             M     MOVWF   0X3F
                    00076         IORLF   ST2P_TMPX,_STATUS2,F    ; Update Secondary status
01BD   3001             M     MOVLW   0X01
01BE   04A1             M     IORWF   0X21,F
                    00077
                    00078 ;-------------------------------------------------------------------------
01BF   0000         00079 M22_NXT NOP
                    00080
                    00081
                    00289
01C0   0000         00290 PATCH   NOP
01C1   0000         00291     NOP
                    00292 ;   GOTO PATCH
                    00293
                    00294         INCLUDE LOOP.INC

                    00001         PAGE
                    00002         SUBTITLE        "Main Loop"
                    00003
                    00004 ; This MODULE is the main loop.  It monitors the voltages, temperature and
                    00005 ; optical levels.  And handles commands sent by the M&C bus
                    00006
01C2                00007 MAIN_LOOP
01C2   0000         00008         NOP
                    00009
                    00010 ;=== SECTION 3-1 ============================================================;
                    00011 ; Measures Vpos,Vneg and Temp.  If one of them fails and the PSUs are on the ;
                    00012 ; PSU fault routine is called.   The loop count is some what contrived as    ;
                    00013 ; (due to a tragic hardware error) it must take values 0,1,3.  This is done  ;
                    00014 ; by doubling and adding 1 (from carry), finishing when the count is 7       ;
                    00015 ; ___________________________________________________________________________;
                    00016
01C3   01F2         00017 M31_L0  CLRF    LCNT
                    00018
01C4   0872         00019 M31_L1  MOVFW   LCNT            ; W must contain channel number
01C5   263D         00020         CALL    IADC            ; make measurement
01C6   1D03         00021         SKPZ
```

```
01C7   29CA          00022         GOTO    M31_L2          ; if no fault goto next loop
                     00023
01C8   0872          00024         MOVFW   LCNT            ; W must contain channel number
01C9   2678          00025         CALL    PFLT            ; for call to shut down routine
                     00026
01CA   1403          00027 M31_L2  BSF     STATUS,C        ; 2*LCNT+1
01CB   0DF2          00028         RLF     LCNT
01CC   1D72          00029         BTFSS   LCNT,2
01CD   29C4          00030         GOTO    M31_L1
                     00031
                     00032 ;=== SECTION 3-2 ========================================================;
                     00033 ; Measures Vcon.  If it fails and the PSUs are on the PSU fault routine is   ;
                     00034 ; called.   If the EDFA is enabled the optical fault routine is called       ;
                     00035 ; ____________________________________________________________________________;
                     00036
01CE   3002          00037         MOVLW   PSU_VC          ; W must contain channel number
01CF   263D          00038         CALL    IADC            ; make measurement
01D0   1D03          00039         SKPZ
01D1   29D6          00040         GOTO    M32_NXT         ; If no fault skip to end of section
                     00041
01D2   3002          00042         MOVLW   PSU_VC          ; W must contain channel number
01D3   2678          00043         CALL    PFLT            ; for call to PSU fault routine
                     00044
01D4   3003          00045         MOVLW   EDF_VC          ; W contains special code
01D5   2692          00046         CALL    EFLT            ; for call to optical fault routine
                     00047
01D6                 00048 M32_NXT
01D6   0000          00049         NOP
                     00050
                     00051 ;=== SECTION 3-3 ========================================================;
                     00052 ; Measures the three optical levels.  If one of them fails and the EDFA is   ;
                     00053 ; enabled the optical fault routine is called                                ;
                     00054 ; ____________________________________________________________________________;
                     00055
01D7   01F2          00056         CLRF    LCNT
                     00057
01D8   0872          00058 M33_L1  MOVFW   LCNT            ; W must contain channel number
01D9   25F7          00059         CALL    XADC            ; make measurement
01DA   1D03          00060         SKPZ
01DB   29DE          00061         GOTO    M33_L2          ; If no fault go to next loop
                     00062
01DC   0872          00063         MOVFW   LCNT            ; W must contain channel number
01DD   2692          00064         CALL    EFLT            ; for all the shut down routine
                     00065
01DE   0AF2          00066 M33_L2  INCF    LCNT
                     00067         SKPGEL  LCNT,3
01DF   3003              M     MOVLW   3       ; w := k
01E0   0272              M     SUBWF   0X72,W      ; w := reg1-k
01E1   1C03              M     SKPC            ; carry for +0 : reg1 >= k
01E2   29D8          00068         GOTO    M33_L1
                     00069
                     00070 ;=== SECTION 3-4 ========================================================;
                     00071 ; Check how many loops have been executed since an SPI interrupt reset TCNT  ;
```

```
                     00072 ; If TCNT reaches 256 call for an EDFA shutdown.  As the loop time is set to ;
                     00073 ; about 10mS this is a timeout of about 2.5 seconds                         ;
                     00074 ; _____________________________________________________________________________;
                     00075
                     00076  IFNDEF EMULATOR
                     00077         INCFSZ  TCNT            ; Increment loop counter
                     00078         GOTO    M34_NXT
                     00079         MOVLW   EDF_TO          ; SPI timeout
                     00080         CALL    EFLT            ; shut down EDFA
                     00081 M34_NXT
                     00082  ENDIF
                     00083 ;=== SECTION 3-5 ===============================================================;
                     00084 ; Check whether a command has been received or is active and deal with it    ;
                     00085 ; _____________________________________________________________________________;
                     00086
01E3   2400          00087         CALL    CMD
                     00088
                     00089 ;=== SECTION 3-6 ===============================================================;
                     00090 ; Delay until ~10mS has elapsed.  With maximum prescale of eight the LS byte ;
                     00091 ; Timer 1 overflows in ~400uS so a count of 25 in the MS byte is right       ;
                     00092 ; _____________________________________________________________________________;
                     00093
                     00094 M36_1   SKPGEL  TMR1H,25
01E4   3019              M     MOVLW   25      ; w := k
01E5   020F              M     SUBWF   TMR1H,W     ; w := reg1-k
01E6   1C03              M     SKPC            ; carry for +0 : reg1 >= k
01E7   29E4          00095         GOTO    M36_1
01E8   018E          00096         CLRF    TMR1L
01E9   018F          00097         CLRF    TMR1H
01EA   29C2          00098         GOTO    MAIN_LOOP
                     00099
                     00100 ;-------------------------------------------------------------------------------
                     00295
0400                 00296     ORG 0X400
                     00297     INCLUDE COMMANDS.INC

                     00001         PAGE
                     00002         SUBTITLE        "Command Handler Subroutine"
                     00003
                     00004 ; This SUBROUTINE examines the command register and branches to the appropriate
                     00005 ; command, if any.  Note that commands 2-7, the calibrate commands modify
                     00006 ; themselves to 10-15 after the first pass
                     00007 ;
                     00008 ; The command register is cleared on successful completion of a command or
                     00009 ; set to a number > $80 if it fails
                     00010
                     00011 ;#############################################################################;
                     00012 ; This macro clears the calibration variables CAL0,1,2 CCNT and sets the flag;
                     00013 ; in  _CMHD                                                                  ;
                     00014 ;_____________________________________________________________________________;
                     00015
                     00016 CALINIT MACRO
                     00017
```

```
00018         CLRF    CCNT
00019         CLRF    CAL0
00020         CLRF    CAL1
00021         CLRF    CAL2
00022         BSF     _COMMAND,3
00023
00024         ENDM
00025
00026 ;=============================================================================;
00027 ; This rather complex macro is used in the calibration loop.  The parameter  ;
00028 ; is set to zero to indicate a 0mW cal or to 1 to indicate a 1mW cal         ;
00029 ;                                                                            ;
00030 ; It accumulates the current value of an optical level into the three global ;
00031 ; registers CAL2,CAL1,CAL0 and increments the calibration count CCNT.        ;
00032 ;                                                                            ;
00033 ; When,after 256 loops,CCNT returns to zero the calibration is complete     ;
00034 ; the accumulated value is divided by 256 (by discarding CAL0 and rounding). ;
00035 ; If the MS part of the result is  reasonable  the new calibration is        ;
00036 ; written to RES_L and RES_M (RES_M only for the 1mW CAL) and _COMMAND set   ;
00037 ; to zero                                                                    ;
00038 ;                                                                            ;
00039 ; For the 1mW cal reasonable means >0 and <=$0F.  For the 0mW cal it means   ;
00040 ; =0.  If a fail occurs on the zero test _COMMAND is set to &FF.  On the     ;
00041 ; $0F test it is set to $FE.  In neither case is the existing calibration    ;
00042 ; altered                                                                    ;
00043 ;____________________________________________________________________________;
```

```
00044
00045 CALIBRATE       MACRO   VAL_M,VAL_L,LIMIT,RES_M,RES_L
00046         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
00047
00048         MOVFW   VAL_L           ; Two + three byte addition
00049         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
00050         MOVFW   CAL1            ;
00051
00052         BTFSC   STATUS,C
00053         INCFSZ  VAL_M
00054         ADDWF   VAL_M,W
00055         MOVWF   CAL1
00056
00057         BTFSC   STATUS,C
00058         INCF    CAL2,F
00059
00060
00061         INCFSZ  CCNT            ; accumulate for 256 counts then
00062         RETURN                  ; wind up
00063
00064         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
00065         GOTO    CAL_A           ;
00066         INCF    CAL1            ; by incrementing CAL1
00067         SKPNZ                   ; If that overflows to zero
00068         INCF    CAL2            ; then increment CAL2
00069
00070                                 ; Now check reasonable
00071 CAL_A   MOVLW   LIMIT
00072         TSTW
00073         SKPZ                    ; Skip if zero cal
00074         GOTO    CAL_1
00075
00076 CAL_0   TSTF    CAL2            ; Zero cal test
00077         SKPZ                    ;
00078         GOTO    CAL_X           ; Fail high if CAL2 is not zero
00079         MOVFF   CAL1,RES_L      ; save one byte
00080         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
00081         RETURN
00082
00083 CAL_1   TSTF    CAL2            ; 1mW cal test
00084         SKPNZ
00085         GOTO    CAL_Y           ; Fail low if CAL2 is zero
00086
00087         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
00088         GOTO    CAL_X           ; Fail high
00089
00090         MOVFF   CAL2,RES_M      ; otherwise save both bytes
00091         MOVFF   CAL1,RES_L
00092         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
00093         RETURN
00094
00095 CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
```

```
                    00096         RETURN                  ; and finish
                    00097
                    00098 CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
                    00099         RETURN                  ; and finish
                    00100
                    00101         ENDM
                    00102
                    00103 ;#################################################################################;
                    00104 ; Examine the command word and branch accordingly.  The clumsy coding shows  ;
                    00105 ; my apprehension about using computed gotos                                  ;
                    00106 ;_____________________________________________________________________________;
                    00107
                    00108
0400   1BC0         00109 CMD     BTFSC   _COMMAND,7      ; If _COMMAND is negative a failed
0401   0008         00110         RETURN                  ; command has not yet been acknowledged
                    00111
0402   0840         00112         MOVF    _COMMAND,W      ; Has a new command been received
0403   1903         00113         SKPNZ
0404   0008         00114         RETURN                  ; No command
                    00115
                    00116         DECW
0405   3EFF             M     ADDLW   0XFF
0406   1903         00117         SKPNZ
0407   2C4A         00118         GOTO    CMD_1           ; Re-tweak
                    00119         DECW
0408   3EFF             M     ADDLW   0XFF
0409   1903         00120         SKPNZ
040A   2C7C         00121         GOTO    CMD_2           ; Initiate calibrate Rx0 at zero input
                    00122         DECW
040B   3EFF             M     ADDLW   0XFF
040C   1903         00123         SKPNZ
040D   2CAF         00124         GOTO    CMD_3           ; Initiate calibrate Rx1 at zero input
                    00125         DECW
040E   3EFF             M     ADDLW   0XFF
040F   1903         00126         SKPNZ
0410   2CE2         00127         GOTO    CMD_4           ; Initiate calibrate Rx2 at zero input
                    00128         DECW
0411   3EFF             M     ADDLW   0XFF
0412   1903         00129         SKPNZ
0413   2D15         00130         GOTO    CMD_5           ; Initiate calibrate Rx0 at 1mW  input
                    00131         DECW
0414   3EFF             M     ADDLW   0XFF
0415   1903         00132         SKPNZ
0416   2D48         00133         GOTO    CMD_6           ; Initiate calibrate Rx1 at 1mW  input
                    00134         DECW
0417   3EFF             M     ADDLW   0XFF
0418   1903         00135         SKPNZ
0419   2D7B         00136         GOTO    CMD_7           ; Initiate calibrate Rx2 at 1mWo input
                    00137         DECW
041A   3EFF             M     ADDLW   0XFF
041B   1903         00138         SKPNZ
041C   2C54         00139         GOTO    CMD_8           ; Reload parameters form EEPROM
                    00140         DECW
```

```
041D   3EFF              M     ADDLW   0XFF
041E   1903           00141         SKPNZ
041F   2C5A           00142         GOTO    CMD_9           ; Save parameters in EEPROM
                      00143         DECW
0420   3EFF              M     ADDLW   0XFF
0421   1903           00144         SKPNZ
0422   2C81           00145         GOTO    CMD_10          ; Continue calibrate Rx0 at zero input
                      00146         DECW
0423   3EFF              M     ADDLW   0XFF
0424   1903           00147         SKPNZ
0425   2CB4           00148         GOTO    CMD_11          ; Continue calibrate Rx1 at zero input
                      00149         DECW
0426   3EFF              M     ADDLW   0XFF
0427   1903           00150         SKPNZ
0428   2CE7           00151         GOTO    CMD_12          ; Continue calibrate Rx2 at zero input
                      00152         DECW
0429   3EFF              M     ADDLW   0XFF
042A   1903           00153         SKPNZ
042B   2D1A           00154         GOTO    CMD_13          ; Continue calibrate Rx0 at 1mW  input
                      00155         DECW
042C   3EFF              M     ADDLW   0XFF
042D   1903           00156         SKPNZ
042E   2D4D           00157         GOTO    CMD_14          ; Continue calibrate Rx1 at 1mW  input
                      00158         DECW
042F   3EFF              M     ADDLW   0XFF
0430   1903           00159         SKPNZ
0431   2D80           00160         GOTO    CMD_15          ; Continue calibrate Rx2 at 1mW  input
                      00161         DECW
0432   3EFF              M     ADDLW   0XFF
0433   1903           00162         SKPNZ
0434   2C60           00163         GOTO    CMD_16          ; Turn PSUs off
                      00164         DECW
0435   3EFF              M     ADDLW   0XFF
0436   1903           00165         SKPNZ
0437   2C64           00166         GOTO    CMD_17          ; Turn PSUs on
                      00167         DECW
0438   3EFF              M     ADDLW   0XFF
0439   1903           00168         SKPNZ
043A   2C67           00169         GOTO    CMD_18          ; Assert EDFA shutdown
                      00170         DECW
043B   3EFF              M     ADDLW   0XFF
043C   1903           00171         SKPNZ
043D   2C6B           00172         GOTO    CMD_19          ; Deassert EDFA shutdown
                      00173         DECW
043E   3EFF              M     ADDLW   0XFF
043F   1903           00174         SKPNZ
0440   2C6E           00175         GOTO    CMD_20          ; Blank the LEDS
                      00176         DECW
0441   3EFF              M     ADDLW   0XFF
0442   1903           00177         SKPNZ
0443   2C72           00178         GOTO    CMD_21          ; Enable the LEDS
                      00179         DECW
0444   3EFF              M     ADDLW   0XFF
```

```
0445   1903           00180         SKPNZ
0446   2C76           00181         GOTO    CMD_22          ; Generate CRC
                      00182
                      00183         MOVLF   0XFF,_COMMAND   ; Illegal command
0447   30FF               M     MOVLW   0XFF
0448   00C0               M     MOVWF   0X40
0449   0008           00184         RETURN
                      00185
                      00186
                      00187 ;=================================================================================;
                      00188 ; Simple commands                                                                  ;
                      00189 ;_________________________________________________________________________________;
                      00190
                      00191 CMD_1   TWEAK                   ;  Retweak
  0000                    M         LOCAL   TWK0
                          M
044A   01F2               M         CLRF    LCNT            ;
                          M
044B   0872               M TWK0    MOVFW   LCNT            ; W contains channel number
044C   25D8               M         CALL    XDAC
                          M
044D   0AF2               M         INCF    LCNT,F          ; 3 units plus initiate command
                          M         SKPGEL  LCNT,4
044E   3004               M     MOVLW   4       ; w := k
044F   0272               M     SUBWF   0X72,W      ; w := reg1-k
0450   1C03               M     SKPC            ; carry for +0 : reg1 >= k
0451   2C4B               M         GOTO    TWK0
                          M
0452   01C0           00192         CLRF    _COMMAND
0453   0008           00193         RETURN
                      00194
                      00195 CMD_8   MOVLF   36,SCNT         ; number of byte to read
0454   3024               M     MOVLW   36
0455   00F1               M     MOVWF   0X71
0456   3042           00196         MOVLW   EEPROM_START    ; first address to indirect register
0457   26AB           00197         CALL    EEPROM_READ
0458   01C0           00198         CLRF    _COMMAND
0459   0008           00199         RETURN
                      00200
                      00201 CMD_9   MOVLF   36,SCNT         ; number of byte to read
045A   3024               M     MOVLW   36
045B   00F1               M     MOVWF   0X71
045C   3042           00202         MOVLW   EEPROM_START    ; first address to indirect register
045D   26C6           00203         CALL    EEPROM_WRITE
045E   01C0           00204         CLRF    _COMMAND
045F   0008           00205         RETURN
                      00206
0460   3004           00207 CMD_16  MOVLW   PSU_MC          ;  Turn off power supplies
0461   2678           00208         CALL    PFLT
0462   01C0           00209         CLRF    _COMMAND
0463   0008           00210         RETURN
                      00211
0464   26EF           00212 CMD_17  CALL    PSU_ON          ;  Turn on power supplies
```

```
0465   01C0           00213         CLRF    _COMMAND
0466   0008           00214         RETURN
                      00215
0467   3004           00216 CMD_18  MOVLW   EDF_MC          ;  EDFA shut down
0468   2692           00217         CALL    EFLT
0469   01C0           00218         CLRF    _COMMAND
046A   0008           00219         RETURN
                      00220
046B   26E9           00221 CMD_19  CALL    EDFA_ON         ;  EDFA start up
046C   01C0           00222         CLRF    _COMMAND
046D   0008           00223         RETURN
                      00224
046E   30FF           00225 CMD_20  MOVLW   0XFF            ; LEDS all off
046F   2710           00226         CALL    LEDSINH
0470   01C0           00227         CLRF    _COMMAND
0471   0008           00228         RETURN
                      00229
0472   3000           00230 CMD_21  MOVLW   0X00            ; LEDS all on
0473   2710           00231         CALL    LEDSINH
0474   01C0           00232         CLRF    _COMMAND
0475   0008           00233         RETURN
                      00234
                      00235
0476   25AE           00236 CMD_22  CALL    CRC             ; Generate CRC
0477   01C0           00237         CLRF    _COMMAND
0478   0008           00238         RETURN
                      00239
                      00240         MOVLF   0XFF,_COMMAND   ; Illegal command
0479   30FF               M     MOVLW   0XFF
047A   00C0               M     MOVWF   0X40
047B   0008           00241         RETURN
                      00242
                      00243
                      00244
                      00245 ;=========================================================================;
                      00246 ; Calibration commands.  On the first pass these modify their command code   ;
                      00247 ; so that initialization occurs only once.  The return occurs within the     ;
                      00248 ; macro CALIBRATE                                                           ;
                      00249 ;_________________________________________________________________________;
                      00250
                      00251 CMD_2   CALINIT
                          M
047C   01FA               M         CLRF    CCNT
047D   01F7               M         CLRF    CAL0
047E   01F8               M         CLRF    CAL1
047F   01F9               M         CLRF    CAL2
0480   15C0               M         BSF     _COMMAND,3
                          M
                      00252 CMD_10  CALIBRATE       _OP0VAL_M,_OP0VAL_L,0,_OP0CAL0,_OP0CAL0
  0000                    M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                          M
0481   0823               M         MOVFW   0X23            ; Two + three byte addition
0482   07F7               M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
```

```
0483   0878             M         MOVFW   CAL1            ;
                        M
0484   1803             M         BTFSC   STATUS,C
0485   0FA2             M         INCFSZ  0X22
0486   0722             M         ADDWF   0X22,W
0487   00F8             M         MOVWF   CAL1
                        M
0488   1803             M         BTFSC   STATUS,C
0489   0AF9             M         INCF    CAL2,F
                        M
                        M
048A   0FFA             M         INCFSZ  CCNT            ; accumulate for 256 counts then
048B   0008             M         RETURN                  ; wind up
                        M
048C   1FF7             M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
048D   2C91             M         GOTO    CAL_A           ;
048E   0AF8             M         INCF    CAL1            ; by incrementing CAL1
048F   1903             M         SKPNZ                   ; If that overflows to zero
0490   0AF9             M         INCF    CAL2            ; then increment CAL2
                        M
                        M                                 ; Now check reasonable
0491   3000             M CAL_A   MOVLW   0
                        M         TSTW
0492   3800             M     IORLW   0       ;
0493   1D03             M         SKPZ                    ; Skip if zero cal
0494   2C9C             M         GOTO    CAL_1
                        M
0495   08F9             M CAL_0   TSTF    CAL2            ; Zero cal test
0496   1D03             M         SKPZ                    ;
0497   2CA9             M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                        M         MOVFF   CAL1,0X58       ; save one byte
0498   0878             M     MOVF    0X78,W
0499   00D8             M     MOVWF   0X58
049A   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
049B   0008             M         RETURN
                        M
049C   08F9             M CAL_1   TSTF    CAL2            ; 1mW cal test
049D   1903             M         SKPNZ
049E   2CAC             M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                        M
                        M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
049F   0879             M     MOVF    0X79,W      ; w := reg1
04A0   3C0E             M     SUBLW   0X0E    ; w := k-reg1
04A1   1C03             M     SKPC            ; carry for +0 : reg1 <= k.
04A2   2CA9             M         GOTO    CAL_X           ; Fail high
                        M
                        M         MOVFF   CAL2,0X58       ; otherwise save both bytes
04A3   0879             M     MOVF    0X79,W
04A4   00D8             M     MOVWF   0X58
                        M         MOVFF   CAL1,0X58
04A5   0878             M     MOVF    0X78,W
04A6   00D8             M     MOVWF   0X58
04A7   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
```

```
04A8   0008              M         RETURN
                         M
                         M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
04A9   30FE              M     MOVLW   0XFE
04AA   00C0              M     MOVWF   0X40
04AB   0008              M         RETURN                  ; and finish
                         M
                         M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
04AC   30FD              M     MOVLW   0XFD
04AD   00C0              M     MOVWF   0X40
04AE   0008              M         RETURN                  ; and finish
                         M
                      00253
                      00254 CMD_3   CALINIT
                         M
04AF   01FA              M         CLRF    CCNT
04B0   01F7              M         CLRF    CAL0
04B1   01F8              M         CLRF    CAL1
04B2   01F9              M         CLRF    CAL2
04B3   15C0              M         BSF     _COMMAND,3
                         M
                      00255 CMD_11  CALIBRATE       _OP1VAL_M,_OP1VAL_L,0,_OP1CAL0,_OP1CAL0
  0000                   M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                         M
04B4   0825              M         MOVFW   0X25            ; Two + three byte addition
04B5   07F7              M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
04B6   0878              M         MOVFW   CAL1            ;
                         M
04B7   1803              M         BTFSC   STATUS,C
04B8   0FA4              M         INCFSZ  0X24
04B9   0724              M         ADDWF   0X24,W
04BA   00F8              M         MOVWF   CAL1
                         M
04BB   1803              M         BTFSC   STATUS,C
04BC   0AF9              M         INCF    CAL2,F
                         M
                         M
04BD   0FFA              M         INCFSZ  CCNT            ; accumulate for 256 counts then
04BE   0008              M         RETURN                  ; wind up
                         M
04BF   1FF7              M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
04C0   2CC4              M         GOTO    CAL_A           ;
04C1   0AF8              M         INCF    CAL1            ; by incrementing CAL1
04C2   1903              M         SKPNZ                   ; If that overflows to zero
04C3   0AF9              M         INCF    CAL2            ; then increment CAL2
                         M
                         M                                 ; Now check reasonable
04C4   3000              M CAL_A   MOVLW   0
                         M         TSTW
04C5   3800              M     IORLW   0       ;
04C6   1D03              M         SKPZ                    ; Skip if zero cal
04C7   2CCF              M         GOTO    CAL_1
                         M
```

```
04C8   08F9              M CAL_0   TSTF    CAL2            ; Zero cal test
04C9   1D03              M         SKPZ                    ;
04CA   2CDC              M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                         M         MOVFF   CAL1,0X5B       ; save one byte
04CB   0878              M     MOVF    0X78,W
04CC   00DB              M     MOVWF   0X5B
04CD   01C0              M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
04CE   0008              M         RETURN
                         M
04CF   08F9              M CAL_1   TSTF    CAL2            ; 1mW cal test
04D0   1903              M         SKPNZ
04D1   2CDF              M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                         M
                         M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
04D2   0879              M     MOVF    0X79,W      ; w := reg1
04D3   3C0E              M     SUBLW   0X0E    ; w := k-reg1
04D4   1C03              M     SKPC            ; carry for +0 : reg1 <= k.
04D5   2CDC              M         GOTO    CAL_X           ; Fail high
                         M
                         M         MOVFF   CAL2,0X5B       ; otherwise save both bytes
04D6   0879              M     MOVF    0X79,W
04D7   00DB              M     MOVWF   0X5B
                         M         MOVFF   CAL1,0X5B
04D8   0878              M     MOVF    0X78,W
04D9   00DB              M     MOVWF   0X5B
04DA   01C0              M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
04DB   0008              M         RETURN
                         M
                         M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
04DC   30FE              M     MOVLW   0XFE
04DD   00C0              M     MOVWF   0X40
04DE   0008              M         RETURN                  ; and finish
                         M
                         M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
04DF   30FD              M     MOVLW   0XFD
04E0   00C0              M     MOVWF   0X40
04E1   0008              M         RETURN                  ; and finish
                         M
                    00256
                    00257 CMD_4   CALINIT
                         M
04E2   01FA              M         CLRF    CCNT
04E3   01F7              M         CLRF    CAL0
04E4   01F8              M         CLRF    CAL1
04E5   01F9              M         CLRF    CAL2
04E6   15C0              M         BSF     _COMMAND,3
                         M
                    00258 CMD_12  CALIBRATE       _OP2VAL_M,_OP2VAL_L,0,_OP2CAL0,_OP2CAL0
  0000                   M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                         M
04E7   0827              M         MOVFW   0X27            ; Two + three byte addition
04E8   07F7              M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
04E9   0878              M         MOVFW   CAL1            ;
```

```
                        M
04EA   1803             M         BTFSC   STATUS,C
04EB   0FA6             M         INCFSZ  0X26
04EC   0726             M         ADDWF   0X26,W
04ED   00F8             M         MOVWF   CAL1
                        M
04EE   1803             M         BTFSC   STATUS,C
04EF   0AF9             M         INCF    CAL2,F
                        M
                        M
04F0   0FFA             M         INCFSZ  CCNT            ; accumulate for 256 counts then
04F1   0008             M         RETURN                  ; wind up
                        M
04F2   1FF7             M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
04F3   2CF7             M         GOTO    CAL_A           ;
04F4   0AF8             M         INCF    CAL1            ; by incrementing CAL1
04F5   1903             M         SKPNZ                   ; If that overflows to zero
04F6   0AF9             M         INCF    CAL2            ; then increment CAL2
                        M
                        M                                 ; Now check reasonable
04F7   3000             M CAL_A   MOVLW   0
                        M         TSTW
04F8   3800             M     IORLW   0       ;
04F9   1D03             M         SKPZ                    ; Skip if zero cal
04FA   2D02             M         GOTO    CAL_1
                        M
04FB   08F9             M CAL_0   TSTF    CAL2            ; Zero cal test
04FC   1D03             M         SKPZ                    ;
04FD   2D0F             M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                        M         MOVFF   CAL1,0X5E       ; save one byte
04FE   0878             M     MOVF    0X78,W
04FF   00DE             M     MOVWF   0X5E
0500   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
0501   0008             M         RETURN
                        M
0502   08F9             M CAL_1   TSTF    CAL2            ; 1mW cal test
0503   1903             M         SKPNZ
0504   2D12             M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                        M
                        M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
0505   0879             M     MOVF    0X79,W      ; w := reg1
0506   3C0E             M     SUBLW   0X0E    ; w := k-reg1
0507   1C03             M     SKPC            ; carry for +0 : reg1 <= k.
0508   2D0F             M         GOTO    CAL_X           ; Fail high
                        M
                        M         MOVFF   CAL2,0X5E       ; otherwise save both bytes
0509   0879             M     MOVF    0X79,W
050A   00DE             M     MOVWF   0X5E
                        M         MOVFF   CAL1,0X5E
050B   0878             M     MOVF    0X78,W
050C   00DE             M     MOVWF   0X5E
050D   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
050E   0008             M         RETURN
```

```
                          M
                          M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
050F   30FE               M     MOVLW   0XFE
0510   00C0               M     MOVWF   0X40
0511   0008               M         RETURN                  ; and finish
                          M
                          M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
0512   30FD               M     MOVLW   0XFD
0513   00C0               M     MOVWF   0X40
0514   0008               M         RETURN                  ; and finish
                          M
                        00259
                        00260 CMD_5   CALINIT
                          M
0515   01FA               M         CLRF    CCNT
0516   01F7               M         CLRF    CAL0
0517   01F8               M         CLRF    CAL1
0518   01F9               M         CLRF    CAL2
0519   15C0               M         BSF     _COMMAND,3
                          M
                        00261 CMD_13  CALIBRATE       _OP0VAL_M,_OP0VAL_L,1,_OP0CAL1_M,_OP0CAL1_L
  0000                    M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                          M
051A   0823               M         MOVFW   0X23            ; Two + three byte addition
051B   07F7               M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
051C   0878               M         MOVFW   CAL1            ;
                          M
051D   1803               M         BTFSC   STATUS,C
051E   0FA2               M         INCFSZ  0X22
051F   0722               M         ADDWF   0X22,W
0520   00F8               M         MOVWF   CAL1
                          M
0521   1803               M         BTFSC   STATUS,C
0522   0AF9               M         INCF    CAL2,F
                          M
                          M
0523   0FFA               M         INCFSZ  CCNT            ; accumulate for 256 counts then
0524   0008               M         RETURN                  ; wind up
                          M
0525   1FF7               M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
0526   2D2A               M         GOTO    CAL_A           ;
0527   0AF8               M         INCF    CAL1            ; by incrementing CAL1
0528   1903               M         SKPNZ                   ; If that overflows to zero
0529   0AF9               M         INCF    CAL2            ; then increment CAL2
                          M
                          M                                 ; Now check reasonable
052A   3001               M CAL_A   MOVLW   1
                          M         TSTW
052B   3800               M     IORLW   0       ;
052C   1D03               M         SKPZ                    ; Skip if zero cal
052D   2D35               M         GOTO    CAL_1
                          M
052E   08F9               M CAL_0   TSTF    CAL2            ; Zero cal test
```

```
052F   1D03              M         SKPZ                    ;
0530   2D42              M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                         M         MOVFF   CAL1,0X57       ; save one byte
0531   0878              M     MOVF    0X78,W
0532   00D7              M     MOVWF   0X57
0533   01C0              M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
0534   0008              M         RETURN
                         M
0535   08F9              M CAL_1   TSTF    CAL2            ; 1mW cal test
0536   1903              M         SKPNZ
0537   2D45              M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                         M
                         M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
0538   0879              M     MOVF    0X79,W      ; w := reg1
0539   3C0E              M     SUBLW   0X0E    ; w := k-reg1
053A   1C03              M     SKPC            ; carry for +0 : reg1 <= k.
053B   2D42              M         GOTO    CAL_X           ; Fail high
                         M
                         M         MOVFF   CAL2,0X56       ; otherwise save both bytes
053C   0879              M     MOVF    0X79,W
053D   00D6              M     MOVWF   0X56
                         M         MOVFF   CAL1,0X57
053E   0878              M     MOVF    0X78,W
053F   00D7              M     MOVWF   0X57
0540   01C0              M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
0541   0008              M         RETURN
                         M
                         M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
0542   30FE              M     MOVLW   0XFE
0543   00C0              M     MOVWF   0X40
0544   0008              M         RETURN                  ; and finish
                         M
                         M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
0545   30FD              M     MOVLW   0XFD
0546   00C0              M     MOVWF   0X40
0547   0008              M         RETURN                  ; and finish
                         M
                      00262
                      00263 CMD_6   CALINIT
                         M
0548   01FA              M         CLRF    CCNT
0549   01F7              M         CLRF    CAL0
054A   01F8              M         CLRF    CAL1
054B   01F9              M         CLRF    CAL2
054C   15C0              M         BSF     _COMMAND,3
                         M
                      00264 CMD_14  CALIBRATE       _OP1VAL_M,_OP1VAL_L,1,_OP1CAL1_M,_OP1CAL1_L
  0000                   M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                         M
054D   0825              M         MOVFW   0X25            ; Two + three byte addition
054E   07F7              M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
054F   0878              M         MOVFW   CAL1            ;
                         M
```

```
0550   1803             M         BTFSC   STATUS,C
0551   0FA4             M         INCFSZ  0X24
0552   0724             M         ADDWF   0X24,W
0553   00F8             M         MOVWF   CAL1
                        M
0554   1803             M         BTFSC   STATUS,C
0555   0AF9             M         INCF    CAL2,F
                        M
                        M
0556   0FFA             M         INCFSZ  CCNT            ; accumulate for 256 counts then
0557   0008             M         RETURN                  ; wind up
                        M
0558   1FF7             M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
0559   2D5D             M         GOTO    CAL_A           ;
055A   0AF8             M         INCF    CAL1            ; by incrementing CAL1
055B   1903             M         SKPNZ                   ; If that overflows to zero
055C   0AF9             M         INCF    CAL2            ; then increment CAL2
                        M
                        M                                 ; Now check reasonable
055D   3001             M CAL_A   MOVLW   1
                        M         TSTW
055E   3800             M     IORLW   0       ;
055F   1D03             M         SKPZ                    ; Skip if zero cal
0560   2D68             M         GOTO    CAL_1
                        M
0561   08F9             M CAL_0   TSTF    CAL2            ; Zero cal test
0562   1D03             M         SKPZ                    ;
0563   2D75             M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                        M         MOVFF   CAL1,0X5A       ; save one byte
0564   0878             M     MOVF    0X78,W
0565   00DA             M     MOVWF   0X5A
0566   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
0567   0008             M         RETURN
                        M
0568   08F9             M CAL_1   TSTF    CAL2            ; 1mW cal test
0569   1903             M         SKPNZ
056A   2D78             M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                        M
                        M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
056B   0879             M     MOVF    0X79,W      ; w := reg1
056C   3C0E             M     SUBLW   0X0E    ; w := k-reg1
056D   1C03             M     SKPC            ; carry for +0 : reg1 <= k.
056E   2D75             M         GOTO    CAL_X           ; Fail high
                        M
                        M         MOVFF   CAL2,0X59       ; otherwise save both bytes
056F   0879             M     MOVF    0X79,W
0570   00D9             M     MOVWF   0X59
                        M         MOVFF   CAL1,0X5A
0571   0878             M     MOVF    0X78,W
0572   00DA             M     MOVWF   0X5A
0573   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
0574   0008             M         RETURN
                        M
```

```
                        M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
0575   30FE             M     MOVLW   0XFE
0576   00C0             M     MOVWF   0X40
0577   0008             M         RETURN                  ; and finish
                        M
                        M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
0578   30FD             M     MOVLW   0XFD
0579   00C0             M     MOVWF   0X40
057A   0008             M         RETURN                  ; and finish
                        M
                      00265
                      00266 CMD_7   CALINIT
                        M
057B   01FA             M         CLRF    CCNT
057C   01F7             M         CLRF    CAL0
057D   01F8             M         CLRF    CAL1
057E   01F9             M         CLRF    CAL2
057F   15C0             M         BSF     _COMMAND,3
                        M
                      00267 CMD_15  CALIBRATE       _OP2VAL_M,_OP2VAL_L,1,_OP2CAL1_M,_OP2CAL1_L
  0000                  M         LOCAL   CAL_0,CAL_1,CAL_A,CAL_X,CAL_Y
                        M
0580   0827             M         MOVFW   0X27            ; Two + three byte addition
0581   07F7             M         ADDWF   CAL0,F          ; results being accumulated in CAL0,1,2
0582   0878             M         MOVFW   CAL1            ;
                        M
0583   1803             M         BTFSC   STATUS,C
0584   0FA6             M         INCFSZ  0X26
0585   0726             M         ADDWF   0X26,W
0586   00F8             M         MOVWF   CAL1
                        M
0587   1803             M         BTFSC   STATUS,C
0588   0AF9             M         INCF    CAL2,F
                        M
                        M
0589   0FFA             M         INCFSZ  CCNT            ; accumulate for 256 counts then
058A   0008             M         RETURN                  ; wind up
                        M
058B   1FF7             M         BTFSS   CAL0,7          ; If top bit of CAL0 is set then round up
058C   2D90             M         GOTO    CAL_A           ;
058D   0AF8             M         INCF    CAL1            ; by incrementing CAL1
058E   1903             M         SKPNZ                   ; If that overflows to zero
058F   0AF9             M         INCF    CAL2            ; then increment CAL2
                        M
                        M                                 ; Now check reasonable
0590   3001             M CAL_A   MOVLW   1
                        M         TSTW
0591   3800             M     IORLW   0       ;
0592   1D03             M         SKPZ                    ; Skip if zero cal
0593   2D9B             M         GOTO    CAL_1
                        M
0594   08F9             M CAL_0   TSTF    CAL2            ; Zero cal test
0595   1D03             M         SKPZ                    ;
```

```
0596   2DA8             M         GOTO    CAL_X           ; Fail high if CAL2 is not zero
                        M         MOVFF   CAL1,0X5D       ; save one byte
0597   0878             M     MOVF    0X78,W
0598   00DD             M     MOVWF   0X5D
0599   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
059A   0008             M         RETURN
                        M
059B   08F9             M CAL_1   TSTF    CAL2            ; 1mW cal test
059C   1903             M         SKPNZ
059D   2DAB             M         GOTO    CAL_Y           ; Fail low if CAL2 is zero
                        M
                        M         SKPLEL  CAL2,0X0E       ; OK if CAL2 < 15
059E   0879             M     MOVF    0X79,W      ; w := reg1
059F   3C0E             M     SUBLW   0X0E    ; w := k-reg1
05A0   1C03             M     SKPC            ; carry for +0 : reg1 <= k.
05A1   2DA8             M         GOTO    CAL_X           ; Fail high
                        M
                        M         MOVFF   CAL2,0X5C       ; otherwise save both bytes
05A2   0879             M     MOVF    0X79,W
05A3   00DC             M     MOVWF   0X5C
                        M         MOVFF   CAL1,0X5D
05A4   0878             M     MOVF    0X78,W
05A5   00DD             M     MOVWF   0X5D
05A6   01C0             M         CLRF    _COMMAND        ; set _COMMAND to 0 and finish
05A7   0008             M         RETURN
                        M
                        M CAL_X   MOVLF   0XFE,_COMMAND   ; set _COMMAND to 254 for too high
05A8   30FE             M     MOVLW   0XFE
05A9   00C0             M     MOVWF   0X40
05AA   0008             M         RETURN                  ; and finish
                        M
                        M CAL_Y   MOVLF   0XFD,_COMMAND   ; set _COMMAND to 253 for too low
05AB   30FD             M     MOVLW   0XFD
05AC   00C0             M     MOVWF   0X40
05AD   0008             M         RETURN                  ; and finish
                        M
                    00268
                    00269
                    00270
                    00298         INCLUDE SUBS.INC

                    00001         PAGE
                    00002         SUBTITLE        "Other Subroutines"
                    00003
                    00004 ;=================================================================================
                    00005 ; Subroutine to calculate the CRC of PROGBLOCKS blocks  of code                    ;
                    00006 ;                                                                                  ;
                    00007 ; Uses the polynomial x^16 + x^12 + x^5 + 1 with starting value 0xFF               ;
                    00008 ; Words are extended to 16 bits                                                    ;
                    00009 ;                                                                                  ;
                    00010 ; On Exit   SHFT_M  undefined                                                      ;
                    00011 ;           SHFT_L  undefined                                                      ;
                    00012 ;           SCNT    zero                                                           ;
```

```
                     00013 ;           LCNT    zero                                                  ;
                     00014 ;           ICNT    zero                                                  ;
                     00015 ;           FSR     undefined                                             ;
                     00016 ;_________________________________________________________________________;
                     00017
                     00018         #DEFINE HASH_M  0X10
                     00019         #DEFINE HASH_L  0X12
                     00020
                     00021 CRC     BANK02
05AE   1283              M     BCF STATUS,RP0
05AF   1703              M     BSF STATUS,RP1
                     00022
05B0   018F          00023         CLRF    EEADRH          ; Upper byte of address
                     00024         MOVLF   PROGBLOCKS,SCNT ; Number of blocks
05B1   3008              M     MOVLW   8
05B2   00F1              M     MOVWF   0X71
                     00025
05B3   30FF          00026         MOVLW   0XFF            ; initialize the accumulator
05B4   00F3          00027         MOVWF   SHFT_M
05B5   00F4          00028         MOVWF   SHFT_L
                     00029
05B6   018D          00030 CRC1    CLRF    EEADR           ; Lower byte of address
05B7   01F2          00031         CLRF    LCNT            ; Counts words in block
                     00032
05B8   1683          00033 CRC2    BSF     STATUS,RP0      ; Bank 2->3
05B9   178C          00034         BSF     EECON1,EEPGD    ; select program memory
05BA   140C          00035         BSF     EECON1,RD       ; fire a read
05BB   0000          00036         NOP
05BC   0000          00037         NOP
05BD   1283          00038         BCF     STATUS,RP0      ; Bank 3->2
                     00039
                     00040         MOVLF   16,ICNT         ; bit counter
05BE   3010              M     MOVLW   16
05BF   00FC              M     MOVWF   0X7C
05C0   0C8C          00041 CRC3    RRF     EEDATA          ; 4 byte shift
05C1   0C8E          00042         RRF     EEDATH          ; of data into
05C2   0CF4          00043         RRF     SHFT_L          ; into accumulator
05C3   0CF3          00044         RRF     SHFT_M
                     00045
05C4   1C03          00046         BTFSS   STATUS,C        ; if a carry out
05C5   2DCA          00047         GOTO    CRC4            ; then hash the acc
                     00048         XORLF   HASH_M,SHFT_M,F
05C6   3010              M     MOVLW   0X10
05C7   06F3              M     XORWF   0X73,F
                     00049         XORLF   HASH_L,SHFT_L,F
05C8   3012              M     MOVLW   0X12
05C9   06F4              M     XORWF   0X74,F
                     00050
05CA   0BFC          00051 CRC4    DECFSZ  ICNT
05CB   2DC0          00052         GOTO    CRC3
                     00053
05CC   0A8D          00054         INCF    EEADR
05CD   0BF2          00055         DECFSZ  LCNT
```

```
05CE   2DB8          00056         GOTO    CRC2
                     00057
05CF   0A8F          00058         INCF    EEADRH
05D0   0BF1          00059         DECFSZ  SCNT
                     00060
                     00061         BANK00
05D1   1283              M     BCF STATUS,RP0
05D2   1303              M     BCF STATUS,RP1
                     00062         MOVFF   SHFT_M,_CRC_M
05D3   0873              M     MOVF    0X73,W
05D4   00E8              M     MOVWF   0X68
                     00063         MOVFF   SHFT_L,_CRC_L
05D5   0874              M     MOVF    0X74,W
05D6   00E9              M     MOVWF   0X69
                     00064
05D7   0008          00065         RETURN
                     00066
                     00067
                     00068 ;================================================================================
                     00069 ; Subroutine to send 12 bit fine tune data to the external DAC                    ;
                     00070 ;                                                                                 ;
                     00071 ; On Entry  W contains the DAC number (0-2 or 3 for the initiate command)         ;
                     00072 ;                                                                                 ;
                     00073 ; On Exit   SHFT_M  undefined                                                     ;
                     00074 ;           SHFT_L  undefined                                                     ;
                     00075 ;           SCNT    zero                                                          ;
                     00076 ;           FSR     undefined                                                     ;
                     00077 ;_________________________________________________________________________________;
                     00078
05D8   3903          00079 XDAC    ANDLW   0X03
05D9   0084          00080         MOVWF   FSR             ; Move W to FSR and multiply by two
05DA   1003          00081         BCF     STATUS,C        ; so that it is the offset from base
05DB   0D84          00082         RLF     FSR,F           ;
                     00083
05DC   2716          00084         CALL    XDLUT           ; converts DAC number to control code in W
05DD   00F3          00085         MOVWF   SHFT_M          ; and save
                     00086
05DE   3050          00087         MOVLW   FINE_TUNE_BASE  ; Add offset to base
05DF   0784          00088         ADDWF   FSR,F           ; to get address of MS Byte
                     00089
05E0   0800          00090         MOVF    INDF,W          ; get MS byte to W
05E1   390F          00091         ANDLW   0X0F            ; mask off permitted four bits
05E2   04F3          00092         IORWF   SHFT_M,F        ; add the control code
                     00093
05E3   0A84          00094         INCF    FSR,F           ; get the LS BYTE
05E4   0800          00095         MOVF    INDF,W
05E5   00F4          00096         MOVWF   SHFT_L
                     00097
                     00098         MOVLF   16,SCNT         ; Loop counter
05E6   3010              M     MOVLW   16
05E7   00F1              M     MOVWF   0X71
05E8   1387          00099         BCF     DCSN            ; enable chip select
                     00100
```

```
05E9   0DF4          00101 XDAC_1  RLF     SHFT_L,F        ; Get top bit into carry
05EA   0DF3          00102         RLF     SHFT_M,F
                     00103
05EB   1803          00104         BTFSC   STATUS,C        ; and copy to output pin
05EC   1507          00105         BSF     DDO
05ED   1C03          00106         BTFSS   STATUS,C
05EE   1107          00107         BCF     DDO
                     00108
05EF   0000          00109         NOP                     ;Pulse the clock 400nS
05F0   1407          00110         BSF     DCLK
05F1   0000          00111         NOP
05F2   1007          00112         BCF     DCLK
                     00113
05F3   0BF1          00114         DECFSZ  SCNT,F          ; Next Bit
05F4   2DE9          00115         GOTO    XDAC_1
05F5   1787          00116         BSF     DCSN            ; Disable chip select
                     00117
05F6   0008          00118         RETURN                  ; and finish
                     00119
                     00120 ;=========================================================================;
                     00121 ; Subroutine to get twelve bit optical level data from the external ADC     ;
                     00122 ; and check it against the limits.                                          ;
                     00123 ;                                                                           ;
                     00124 ; On entry  W contains the ADC number (0-2)                                 ;
                     00125 ;                                                                           ;
                     00126 ; On exit   SHFT_M undefined                                                ;
                     00127 ;           SHFT_L undefined                                                ;
                     00128 ;           SCNT   zero                                                     ;
                     00129 ;           FSR    undefined                                                ;
                     00130 ;           STATUS,Z is clear if value was in limits, set otherwise         ;
                     00131 ;_____________________________________________________________________________;
                     00132
05F7   3903          00133 XADC    ANDLW   0X03
05F8   00F5          00134         MOVWF   TEMP0            ; Save W for limit check
                     00135
05F9   0084          00136         MOVWF   FSR             ; Move W to FSR and multiply by two
05FA   1003          00137         BCF     STATUS,C        ; so that it is the offset from base
05FB   0D84          00138         RLF     FSR,F           ;
                     00139
05FC   2720          00140         CALL    XALUT           ; Converts ADC number to command code in W
05FD   00F3          00141         MOVWF   SHFT_M          ; and move to register for sending
                     00142
05FE   3022          00143         MOVLW   EXTERNAL_VALUES_BASE    ; Add offset to base
05FF   0784          00144         ADDWF   FSR,F           ; to get address of MS Byte
                     00145
                     00146         MOVLF   8,SCNT          ; Set up loop to send command byte
0600   3008              M     MOVLW   8
0601   00F1              M     MOVWF   0X71
0602   1307          00147         BCF     ACSN            ; Assert chip select
                     00148
0603   0DF3          00149 XADC_1  RLF     SHFT_M,F        ; Get top bit into carry
                     00150
0604   1803          00151         BTFSC   STATUS,C        ; and copy to output pin
```

```
0605   1507           00152         BSF     DDO
0606   1C03           00153         BTFSS   STATUS,C
0607   1107           00154         BCF     DDO
                      00155
0608   0000           00156         NOP                     ; Pulse the clock
0609   1407           00157         BSF     DCLK
060A   0000           00158         NOP
060B   1007           00159         BCF     DCLK
                      00160
060C   0BF1           00161         DECFSZ  SCNT,F          ; Next Bit
060D   2E03           00162         GOTO    XADC_1
                      00163
                      00164         MOVLF   13,SCNT         ; prepare to acquire 12 bits in 13 clocks
060E   300D               M     MOVLW   13
060F   00F1               M     MOVWF   0X71
0610   01F3           00165         CLRF    SHFT_M          ; first clock will acquire zero
0611   01F4           00166         CLRF    SHFT_L          ; from pre cleared carry
0612   1003           00167         BCF     STATUS,C
                      00168
0613   0DF4           00169 XADC_2  RLF     SHFT_L,F        ; shift up, picking up new bit from carry
0614   0DF3           00170         RLF     SHFT_M,F        ; first time round will be zero
                      00171
0615   1407           00172         BSF     DCLK            ; Pulse the clock to get next bit
0616   0000           00173         NOP
0617   1007           00174         BCF     DCLK
0618   0000           00175         NOP
                      00176
0619   1887           00177         BTFSC   DDI             ; get the bit from the input pin
061A   1403           00178         BSF     STATUS,C        ; into carry.
061B   1C87           00179         BTFSS   DDI             ; last time round no data acquired
061C   1003           00180         BCF     STATUS,C
                      00181
061D   0BF1           00182         DECFSZ  SCNT,F          ; Next Bit
061E   2E13           00183         GOTO    XADC_2
061F   1707           00184         BSF     ACSN            ; Disable chip select
                      00185
                      00186                                 ; FSR still points to MS byte
0620   1780           00187         BSF     INDF,7          ; Set changing flag in MS byte
0621   0A84           00188         INCF    FSR,F           ; Move to LS byte
                      00189         MOVFF   SHFT_L,INDF     ; and store the value
0622   0874               M     MOVF    0X74,W
0623   0080               M     MOVWF   INDF
0624   0384           00190         DECF    FSR,F           ; Back to MS byte
                      00191         MOVFF   SHFT_M,INDF     ; and store the value (which clears the flag)
0625   0873               M     MOVF    0X73,W
0626   0080               M     MOVWF   INDF
                      00192
                      00193
                      00194 ;--------------------------------------------------------------------------------
                      00195 ; At this point the result is in SHFT_M (ms 4 bits) and SHFT_L (lower 8 bits)
                      00196 ; The upper limit test is made only on the ms 4 bits
                      00197 ; The lower limit test is made on the ls 8 bits if the ms 4 bits are zero
                      00198
```

```
                     00199         MOVFF   TEMP0,FSR              ; Recover the channel number
0627   0875              M     MOVF    0X75,W
0628   0084              M     MOVWF   FSR
0629   1003          00200         BCF     STATUS,C               ; to FSR and multiply by two
062A   0D84          00201         RLF     FSR,F                  ; so that it is the offset from base
                     00202
062B   3042          00203         MOVLW   EXTERNAL_LIMITS_BASE   ; Add offset to base of limits
062C   0784          00204         ADDWF   FSR,F                  ; to get address of upper limit
                     00205
062D   0800          00206         MOVFW   INDF                   ; Get the upper limit to W
                     00207         SKPLTW  SHFT_M                 ; If value < upper limit OK
062E   0273              M     SUBWF   0X73,W      ; w := reg1-w
062F   1803              M     SKPNC           ; carry/ for - : reg1 < w
0630   2E3B          00208         GOTO    XADC_4                 ; If value >= upper limit goto fault
                     00209
0631   08F3          00210         TSTF    SHFT_M                 ; If ms value> 0 OK-above lower limit
0632   1D03          00211         SKPZ                           ; ms value is zero - more tests needed
0633   2E39          00212         GOTO    XADC_3                 ; > zero so finish OK
                     00213
0634   0A84          00214         INCF    FSR,F                  ; FSR points to lower limit
0635   0800          00215         MOVFW   INDF                   ; get lower limit to W
                     00216         SKPGEW  SHFT_L                 ; If value >= lower limit OK
0636   0274              M     SUBWF   0X74,W      ; w := reg1-w
0637   1C03              M     SKPC            ; carry for +0 : reg1 >= w
0638   2E3B          00217         GOTO    XADC_4                 ; If value < lower limit then fault
                     00218
0639   1103          00219 XADC_3  BCF     STATUS, Z              ; No fault
063A   0008          00220         RETURN                         ; Return with Z clear
                     00221
063B   1503          00222 XADC_4  BSF     STATUS,Z               ; Fault
063C   0008          00223         RETURN                         ; Return with Z set
                     00224
                     00225 ;=========================================================================;
                     00226 ; Subroutine to get ten bit voltage or temperature data from the Internal ADC ;
                     00227 ; and check against limits                                                  ;
                     00228 ;                                                                          ;
                     00229 ; On entry  W contains the channel select code(0-3)                        ;
                     00230 ;                                                                          ;
                     00231 ; On exit   SCNT   zero                                                    ;
                     00232 ;           FSR    undefined                                               ;
                     00233 ;           STATUS,Z is clear if value was in limits, set otherwise       ;
                     00234 ;__________________________________________________________________________;
                     00235
                     00236
                     00237 ;*** The initial code has a wretched patch to swap bits 0 and 1 of W to
                     00238 ;*** compensate for  a wiring error on the inputs to the analog mux
                     00239 ;*** which should be corrected on the next build of the board
                     00240 ;*** When patch is removed ember to retain MOVWF TEMP0
                     00241
                     00242
                     00243 IADC    BANK0
063D   1283              M     BCF STATUS,RP0
                     00244
```

```
063E   00F5           00245         MOVWF   TEMP0                   ; Save W for limit check
                      00246
                      00247 ;*      BCF     PORTE,0                 ; Patch removed
                      00248 ;*      BCF     PORTE,1                 ;
                      00249 ;*      BTFSC   TEMP0,0                 ;
                      00250 ;*      BSF     PORTE,1                 ;
                      00251 ;*      BTFSC   TEMP0,1                 ;
                      00252 ;*      BSF     PORTE,0                 ;
                      00253
063F   1009           00254         BCF     PORTE,0                 ;
0640   1875           00255         BTFSC   TEMP0,0                 ;
0641   1409           00256         BSF     PORTE,0                 ;
0642   1089           00257         BCF     PORTE,1                 ;
0643   18F5           00258         BTFSC   TEMP0,1                 ;
0644   1489           00259         BSF     PORTE,1                 ;
                      00260
0645   0084           00261         MOVWF   FSR                     ; Move W to FSR and multiply by two
0646   1003           00262         BCF     STATUS,C                ; so that it is the offset from base
0647   0D84           00263         RLF     FSR,F                   ;
                      00264
0648   3028           00265         MOVLW   INTERNAL_VALUES_BASE    ; Add offset to base of read values
0649   0784           00266         ADDWF   FSR,F                   ; to get address of MS Byte
                      00267
064A   141F           00268         BSF     ADCON0,ADON             ; turn on ADC
                      00269
                      00270         MOVLF   0X22,SCNT               ; wait Tacq = 20uS
064B   3022               M     MOVLW   0X22
064C   00F1               M     MOVWF   0X71
064D   0BF1           00271 IADC_1  DECFSZ  SCNT,F
064E   2E4D           00272         GOTO    IADC_1
                      00273
064F   151F           00274         BSF     ADCON0,GO               ; Start conversion
0650   191F           00275 IADC_2  BTFSC   ADCON0,GO               ; wait till finished
0651   2E50           00276         GOTO    IADC_2
                      00277
0652   101F           00278         BCF     ADCON0,ADON             ; Turn off again
                      00279
0653   1780           00280         BSF     INDF,7                  ; FSR is pointing to MS byte
                      00281                                         ; Set flag bit
                      00282         BANK1
0654   1683               M     BSF STATUS,RP0
0655   0A84           00283         INCF    FSR,F                   ; Advance FSR to LS byte
                      00284         MOVFF   ADRESL,INDF             ; and store the value
0656   081E               M     MOVF    ADRESL,W
0657   0080               M     MOVWF   INDF
0658   00F4           00285         MOVWF   SHFT_L
                      00286
                      00287         BANK0
0659   1283               M     BCF STATUS,RP0
065A   0384           00288         DECF    FSR,F                   ; Back to the MS byte
                      00289         MOVFF   ADRESH,INDF             ; store the value (which clears flag)
065B   081E               M     MOVF    ADRESH,W
065C   0080               M     MOVWF   INDF
```

```
065D   00F3           00290         MOVWF   SHFT_M
                      00291
065E   08F5           00292         TSTF    TEMP0
065F   1903           00293         BTFSC   STATUS,Z
0660   0000           00294         NOP                             ; PATCH TO FIND W=0
                      00295
                      00296 ;--------------------------------------------------------------------------------
                      00297 ; At this point the result is in SHFT_M (ms 2 bits) and SHFT_L (ls 8 bits)
                      00298 ; The test is only made on the ms 8 bits
                      00299
0661   0CF3           00300         RRF     SHFT_M,F                ; Shift right two places
0662   0CF4           00301         RRF     SHFT_L,F                ; so ms 8 bits are in SHFT_L
0663   0CF3           00302         RRF     SHFT_M,F
0664   0CF4           00303         RRF     SHFT_L,F
                      00304
                      00305         MOVFF   TEMP0,FSR               ; Recover the channel number
0665   0875               M     MOVF    0X75,W
0666   0084               M     MOVWF   FSR
0667   1003           00306         BCF     STATUS,C                ; to FSR and multiply by two
0668   0D84           00307         RLF     FSR,F                   ; so that it is the offset from base
                      00308
0669   3048           00309         MOVLW   INTERNAL_LIMITS_BASE    ; Add offset to base of limits
066A   0784           00310         ADDWF   FSR,F                   ; to get address of upper limit
                      00311
066B   0800           00312         MOVFW   INDF                    ; Get the upper limit to W
                      00313         SKPLTW  SHFT_L                  ; If value < upper limit then no fault
066C   0274               M     SUBWF   0X74,W      ; w := reg1-w
066D   1803               M     SKPNC           ; carry/ for - : reg1 < w
066E   2E76           00314         GOTO    IADC_4                  ; If value >= upper limit goto fault
                      00315
066F   0A84           00316         INCF    FSR,F                   ; FSR points to lower limit
0670   0800           00317         MOVFW   INDF                    ; get lower limit to W
                      00318         SKPGEW  SHFT_L                  ; If value >= lower limit OK
0671   0274               M     SUBWF   0X74,W      ; w := reg1-w
0672   1C03               M     SKPC            ; carry for +0 : reg1 >= w
0673   2E76           00319         GOTO    IADC_4                  ; If value < lower limit then fault
                      00320
0674   1103           00321         BCF     STATUS, Z               ; No fault
0675   0008           00322         RETURN                          ; Return with Z clear
                      00323
0676   1503           00324 IADC_4  BSF     STATUS,Z                ; Fault
0677   0008           00325         RETURN                          ; Return with Z set
                      00326
                      00327 ;================================================================================;
                      00328 ; Subroutine called on a voltage or temperature fault after start up or for  ;
                      00329 ; an M&C commanded PSU shutdown. IF the PSUs are not already shut down       ;
                      00330 ; saves the values, shut down the PSUs and flag  the status word            ;
                      00331 ;                                                                           ;
                      00332 ; An alternate entry point IFLT_X skips the PSU on check                    ;
                      00333 ;                                                                           ;
                      00334 ; On entry  W contains the reason for the call                              ;
                      00335 ;       0 - Vneg fault          translates to status code $04               ;
                      00336 ;       1 - Vpos fault          translates to status code $05               ;
```

```
                    00337 ;       2 - Vcon fault          translates to status code $06               ;
                    00338 ;       3 - Temp fault          translates to status code $07               ;
                    00339 ;       4 - M&C command         translates to status code $08               ;
                    00340 ;                                                                           ;
                    00341 ; On exit W contains 0X00 if the routine was executed                       ;
                    00342 ;                or 0xFF if an immediate exit was taken                     ;
                    00343 ;___________________________________________________________________________;
                    00344
0678   1B20         00345 PFLT    BTFSC   ST1_PSU         ; If PSU already shut down
0679   34FF         00346         RETLW   0XFF            ; return at once
                    00347
067A   3E04         00348 PFLT_X  ADDLW   4               ; Convert channel number to fault code in W
067B   390F         00349         ANDLW   0X0F
067C   04A1         00350         IORWF   _STATUS2,F      ; Update the secondary status word
                    00351
                    00352         MOVFF   _VNVAL_M,_VNFLT_M       ; Copy the current readings
067D   0828             M     MOVF    0X28,W
067E   00B8             M     MOVWF   0X38
                    00353         MOVFF   _VNVAL_L,_VNFLT_L       ; to the fault locations
067F   0829             M     MOVF    0X29,W
0680   00B9             M     MOVWF   0X39
                    00354         MOVFF   _VPVAL_M,_VPFLT_M
0681   082A             M     MOVF    0X2A,W
0682   00BA             M     MOVWF   0X3A
                    00355         MOVFF   _VPVAL_L,_VPFLT_L
0683   082B             M     MOVF    0X2B,W
0684   00BB             M     MOVWF   0X3B
                    00356         MOVFF   _TMPVAL_M, _TMPFLT_M
0685   082E             M     MOVF    0X2E,W
0686   00BE             M     MOVWF   0X3E
                    00357         MOVFF   _TMPVAL_L, _TMPFLT_L
0687   082F             M     MOVF    0X2F,W
0688   00BF             M     MOVWF   0X3F
                    00358         MOVFF   _VCVAL_M,_VCFLT_M
0689   082C             M     MOVF    0X2C,W
068A   00BC             M     MOVWF   0X3C
                    00359         MOVFF   _VCVAL_L,_VCFLT_L
068B   082D             M     MOVF    0X2D,W
068C   00BD             M     MOVWF   0X3D
                    00360
068D   1486         00361         BSF     VPENN           ; Turn of the two PSUs
068E   1506         00362         BSF     VNENN
068F   1088         00363         BCF     PSULEDN         ; Turn on the fault LED
                    00364
0690   1720         00365         BSF     ST1_PSU         ; Update the primary status word
                    00366
0691   0008         00367         RETURN
                    00368
```

```
                     00369 ;===============================================================================;
                     00370 ; Subroutine called on an optical fault or for an M&C commanded EDFA shutdown;
                     00371 ; If EDFA not already shut down -  save the values, issue an EDFA shutdown   ;
                     00372 ; command and flag the status word.                                          ;
                     00373 ;                                                                           ;
                     00374 ; An alternate entry point XFLT_X skips the EDFA check                       ;
                     00375 ;                                                                           ;
                     00376 ; On entry  W contains the reason for the call                              ;
                     00377 ;       0 - Rx0 fault           translates to status code $10               ;
                     00378 ;       1 - Rx1 fault           translates to status code $20               ;
                     00379 ;       2 - Rx2 fault           translates to status code $30               ;
                     00380 ;       3 - Vcon Fault          translates to status code $40               ;
                     00381 ;       4 - M&C command         translates to status code $50               ;
                     00382 ;       5 - SPI timeout         translates to status code $60               ;
                     00383 ;                                                                           ;
                     00384 ; On exit W contains 0x00 if the routine was executed                       ;
                     00385 ;                 or 0xFF if an immediate return was taken                  ;
                     00386 ;___________________________________________________________________________;
                     00387
                     00388
                     00389
0692   1BA0          00390 EFLT    BTFSC   ST1_EDFA        ; If EDFA already shut down
0693   34FF          00391         RETLW   0XFF            ; return at once
                     00392
0694   3E01          00393 XFLT_X  ADDLW   1               ; Convert channel number to fault code in W
0695   00F5          00394         MOVWF   TEMP0
0696   0DF5          00395         RLF     TEMP0
0697   0DF5          00396         RLF     TEMP0
0698   0DF5          00397         RLF     TEMP0
0699   0D75          00398         RLF     TEMP0,W
                     00399
069A   04A1          00400         IORWF   _STATUS2,F      ; Update the secondary status word
                     00401
                     00402         MOVFF   _OP0VAL_M,_OP0FLT_M     ; Copy the current readings
069B   0822              M     MOVF    0X22,W
069C   00B2              M     MOVWF   0X32
                     00403         MOVFF   _OP0VAL_L,_OP0FLT_L     ; to the fault locations
069D   0823              M     MOVF    0X23,W
069E   00B3              M     MOVWF   0X33
                     00404         MOVFF   _OP1VAL_M,_OP1FLT_M
069F   0824              M     MOVF    0X24,W
06A0   00B4              M     MOVWF   0X34
                     00405         MOVFF   _OP1VAL_L,_OP1FLT_L
06A1   0825              M     MOVF    0X25,W
06A2   00B5              M     MOVWF   0X35
                     00406         MOVFF   _OP2VAL_M,_OP2FLT_M
06A3   0826              M     MOVF    0X26,W
06A4   00B6              M     MOVWF   0X36
                     00407         MOVFF   _OP2VAL_L,_OP2FLT_L
06A5   0827              M     MOVF    0X27,W
06A6   00B7              M     MOVWF   0X37
                     00408
```

```
06A7   1606           00409         BSF     EDFASD          ; Turn on the EDFA shut down line
06A8   1008           00410         BCF     EDFALEDN        ; Turn on the  fault LED
                      00411
06A9   17A0           00412         BSF     ST1_EDFA        ; Update the primary status word
                      00413
06AA   0008           00414         RETURN
                      00415
                      00416 ;=================================================================================;
                      00417 ; This subroutine copies 33 parameters from EEPROM to RAM followed by the    ;
                      00418 ; checksum                                                                    ;
                      00419 ;                                                                             ;
                      00420 ; On exit  SCNT   zero                                                        ;
                      00421 ;          FSR    undefined                                                   ;
                      00422 ;_____________________________________________________________________________;
                      00423
                      00424         #DEFINE HASH    0X1D
06AB                  00425 EEPROM_READ
                      00426
                      00427         MOVLF   34,SCNT
06AB   3022               M     MOVLW   34
06AC   00F1               M     MOVWF   0X71
                      00428         MOVLF   EEPROM_START,FSR; Start address to FSR
06AD   3042               M     MOVLW   0X42
06AE   0084               M     MOVWF   FSR
                      00429         BANK02
06AF   1283               M     BCF STATUS,RP0
06B0   1703               M     BSF STATUS,RP1
06B1   018D           00430         CLRF    EEADR
06B2   01F5           00431         CLRF    TEMP0           ; accumulator for checksum
                      00432
                      00433 EER_L1  BANK03                  ; Bank 3
06B3   1683               M     BSF STATUS,RP0
06B4   1703               M     BSF STATUS,RP1
06B5   138C           00434         BCF     EECON1,EEPGD    ; point to EEPROM not PROG mem
06B6   140C           00435         BSF     EECON1,RD       ; fire a read
                      00436
06B7   1283           00437         BCF     STATUS,RP0      ; Bank 3->2
06B8   080C           00438         MOVFW   EEDATA
06B9   0080           00439         MOVWF   INDF            ; Data to RAM at indirect pointer
06BA   06F5           00440         XORWF   TEMP0           ; accumulate checksum
                      00441
                      00442 ;       MOVLW   HASH
                      00443 ;       RLF     EEDATA
                      00444 ;       RLF     TEMP0
                      00445 ;       BTFSC   STATUS,C
                      00446 ;       XORWF   TEMP0
                      00447
06BB   0A8D           00448         INCF    EEADR
06BC   0A84           00449         INCF    FSR,F           ; increment address
06BD   0BF1           00450         DECFSZ  SCNT,F
06BE   2EB3           00451         GOTO    EER_L1
                      00452
06BF   0875           00453         MOVFW   TEMP0           ; Also sets or clears Z |
```

```
06C0   1303           00454         BCF     STATUS,RP1      ; Bank 2->0             |
06C1   00E3           00455         MOVWF   _CHECKSUM       ;                       |
                      00456                                 ;                       |
06C2   1220           00457         BCF     ST1_EECSF       ;                       |
06C3   1D03           00458         SKPZ                    ; for use here        <--
06C4   1620           00459         BSF     ST1_EECSF
                      00460
06C5   0008           00461         RETURN
                      00462
                      00463 ;=========================================================================;
                      00464 ; This subroutine copies 33 parameters from RAM to EEPROM accumulating a     ;
                      00465 ; checksum in _CHECKSUM and writing this to the 34th line of EEPROM          ;
                      00466 ;                                                                           ;
                      00467 ; On exit  SCNT   zero                                                      ;
                      00468 ;          FSR    undefined                                                 ;
                      00469 ;          _CHECKSUM  zero                                                  ;
                      00470 ;___________________________________________________________________________;
                      00471
06C6                  00472 EEPROM_WRITE
                      00473
                      00474         MOVLF   34,SCNT
06C6   3022               M     MOVLW   34
06C7   00F1               M     MOVWF   0X71
06C8   01E3           00475         CLRF    _CHECKSUM
                      00476         MOVLF   EEPROM_START,FSR                ; Start address to FSR
06C9   3042               M     MOVLW   0X42
06CA   0084               M     MOVWF   FSR
                      00477         BANK02
06CB   1283               M     BCF STATUS,RP0
06CC   1703               M     BSF STATUS,RP1
06CD   018D           00478         CLRF    EEADR
                      00479
06CE   1703           00480 EEW_L1  BSF     STATUS,RP1      ; Bank 0->2
                      00481         MOVFF   INDF,EEDATA     ; Data from RAM at indirect pointer
06CF   0800               M     MOVF    INDF,W
06D0   008C               M     MOVWF   EEDATA
                      00482
06D1   1683           00483         BSF     STATUS,RP0      ; Bank 2->3
06D2   138C           00484         BCF     EECON1,EEPGD    ; point to EEPROM not PROG mem
06D3   150C           00485         BSF     EECON1,WREN     ; enable writes
                      00486
06D4   138B           00487         BCF     INTCON,GIE      ; disable interrupts for safety
                      00488         MOVLF   0X55,EECON2     ; write $55
06D5   3055               M     MOVLW   0X55
06D6   008D               M     MOVWF   EECON2
                      00489         MOVLF   0XAA,EECON2     ; then $AA to prime write
06D7   30AA               M     MOVLW   0XAA
06D8   008D               M     MOVWF   EECON2
06D9   148C           00490         BSF     EECON1,WR       ; fire the write
06DA   178B           00491         BSF     INTCON,GIE      ; reenable interrupts
                      00492
06DB   188C           00493 EEW_WT  BTFSC   EECON1,WR       ; wait till write complete
06DC   2EDB           00494         GOTO    EEW_WT
```

```
06DD   110C          00495         BCF     EECON1,WREN     ; then disable writes
                     00496
06DE   1283          00497         BCF     STATUS,RP0      ; Bank 3->2
06DF   0A8D          00498         INCF    EEADR,F
06E0   080C          00499         MOVFW   EEDATA
                     00500
06E1   1303          00501         BCF     STATUS,RP1      ; Bank 2->0
06E2   06E3          00502         XORWF   _CHECKSUM,F
                     00503
06E3   0A84          00504         INCF    FSR,F           ; increment address
06E4   0BF1          00505         DECFSZ  SCNT,F
06E5   2ECE          00506         GOTO    EEW_L1
                     00507
                     00508         BANK00
06E6   1283              M     BCF STATUS,RP0
06E7   1303              M     BCF STATUS,RP1
06E8   0008          00509         RETURN
                     00510
                     00511 ;       BCF     STATUS,RP0      ; Bank 3->2;
                     00512 ;
                     00513 ;       INCF    EEADR,F
                     00514 ;       MOVLW   HASH            ; Prepare for a possible hash
                     00515 ;       RLF     EEDATA          ; Shift the data into carry
                     00516 ;
                     00517 ;       BCF     STATUS,RP1      ; Bank 2->0
                     00518 ;       RLF     _CHECKSUM,F     ; Shift data on into acc
                     00519 ;       BTFSC   STATUS,C        ; if bit drops out
                     00520 ;       XORWF   _CHECKSUM,F     ; then hash
                     00521 ;
                     00522 ;       INCF    FSR,F           ; increment address
                     00523 ;       DECFSZ  SCNT,F
                     00524 ;       GOTO    EEW_L1
                     00525 ;
                     00526 ;       BANK00
                     00527 ;       RETURN
                     00528
                     00529 ;=================================================================================;
                     00530 ; This subroutine releases the EDFA                                               ;
                     00531 ;_________________________________________________________________________________;
                     00532
06E9   1206          00533 EDFA_ON BCF     EDFASD          ; Release the shutdown line
                     00534
06EA   1408          00535         BSF     EDFALEDN        ; Turn off the LED
06EB   13A0          00536         BCF     ST1_EDFA        ; Clear the status bit
                     00537         ANDLF   0X0F,_STATUS2,F
06EC   300F              M     MOVLW   0X0F
06ED   05A1              M     ANDWF   0X21,F
                     00538
06EE   0008          00539         RETURN
                     00540
                     00541 ;=================================================================================;
                     00542 ; This turns the PSUs on and waits ~100mS for them to stabilize                   ;
                     00543 ;_________________________________________________________________________________;
```

```
                    00544
06EF   1106         00545 PSU_ON  BCF     VNENN           ; Turn the PSUs on
06F0   1086         00546         BCF     VPENN           ;
                    00547
                    00548
                    00549 ;       CLRF    TMR1L;
                    00550 ;       MOVLF   0x01,TMR1H      ; Set TIMER1 to 0x0100 (prescaler is 8)
                    00551 ;
                    00552 ;PSU_ON1        TSTF    TMR1H           ; Wait until it overflows to zero
                    00553 ;       SKPZ                    ; This takes 255*256*8 clocks
                    00554 ;       GOTO    PSU_ON1         ; At 5MHz that is ~104mS
                    00555
06F1   01F2         00556         CLRF    LCNT
06F2   01F1         00557 PSU_ON1 CLRF    SCNT
06F3   0000         00558 PSU_ON2 NOP
06F4   0000         00559         NOP
06F5   0000         00560         NOP
06F6   0000         00561         NOP
06F7   0000         00562         NOP
06F8   0000         00563         NOP
06F9   0000         00564         NOP
06FA   0000         00565         NOP
06FB   0000         00566         NOP
06FC   0000         00567         NOP
06FD   0000         00568         NOP
06FE   0000         00569         NOP
06FF   0000         00570         NOP
0700   0000         00571         NOP
0701   0000         00572         NOP
0702   0000         00573         NOP
0703   0000         00574         NOP
0704   0000         00575         NOP
0705   0000         00576         NOP
0706   0000         00577         NOP
0707   0BF1         00578         DECFSZ  SCNT
0708   2EF3         00579         GOTO    PSU_ON2
0709   0BF2         00580         DECFSZ  LCNT
070A   2EF2         00581         GOTO    PSU_ON1
                    00582
070B   1488         00583         BSF     PSULEDN         ; Turn off the LED
070C   1320         00584         BCF     ST1_PSU         ; Clear the status bits
                    00585         ANDLF   0XF0,_STATUS2,F
070D   30F0             M     MOVLW   0XF0
070E   05A1             M     ANDWF   0X21,F
                    00586
070F   0008         00587         RETURN
                    00588
```

```
                     00589 ;=========================================================================;
                     00590 ; This subroutine inhibits select LEDS                                    ;
                     00591 ; The W register contains a mask.  A one inhibits the LED at that position   ;
                     00592 ;_________________________________________________________________________;
                     00593
                     00594 LEDSINH BANK01
0710   1683              M     BSF STATUS,RP0
0711   1303              M     BCF STATUS,RP1
0712   0088           00595         MOVWF   TRISD
                      00596         BANK00
0713   1283              M     BCF STATUS,RP0
0714   1303              M     BCF STATUS,RP1
0715   0008           00597         RETURN
                      00598
                      00599 ;*************************************************************************
                      00600 ; LOOK UP TABLES.  These do not use computed GOTOs because of the
                      00601 ; difficulties with relocation and with interrupts
                      00602 ;
                      00603 ;
                      00604 ;== Look up table for control codes for external DAC ======================
                      00605
                      00606 XDLUT   TSTW
0716   3800              M     IORLW   0       ;
0717   1903           00607         SKPNZ
0718   3410           00608         RETLW   0X10            ; W=0 control code to load input reg 0
                      00609         DECW
0719   3EFF              M     ADDLW   0XFF
071A   1903           00610         SKPNZ
071B   3450           00611         RETLW   0X50            ; W=1 control code to load input reg 1
                      00612         DECW
071C   3EFF              M     ADDLW   0XFF
071D   1903           00613         SKPNZ
071E   3490           00614         RETLW   0X90            ; W=2 control code to load input reg 2
071F   3440           00615         RETLW   0X40            ; W>2 control code to transfer input reg
                      00616
                      00617 ;== Look up table for control codes for external ADC ======================
                      00618
                      00619 XALUT   TSTW
0720   3800              M     IORLW   0       ;
0721   1903           00620         SKPNZ
0722   3497           00621         RETLW   0X97            ; W=0 control code to read ADC 0
                      00622         DECW
0723   3EFF              M     ADDLW   0XFF
0724   1903           00623         SKPNZ
0725   34D7           00624         RETLW   0XD7            ; W=1 control code to read ADC 1
                      00625         DECW
0726   3EFF              M     ADDLW   0XFF
0727   1903           00626         SKPNZ
0728   34A7           00627         RETLW   0XA7            ; W=2 control code to read ADC 2
0729   34E7           00628         RETLW   0XE7            ; W=3 control code to read ADC 3
                      00629
                      00630
```

```
                    00631
                    00299
                    00300
                    00301 ;EEPROM
                    00302
2100                00303     ORG 0X2100
                    00304 ; Target 0x42
2100   0008 0008    00305     DE  0X08,0X08   ; uplimit/256 , low limit for Rx0
2102   0008 0008    00306     DE  0X08,0X08   ; uplimit/256 , low limit for Rx1
2104   0008 0008    00307     DE  0X08,0X08   ; uplimit/256 , low limit for Rx2
                    00308
                    00309 ;Target '0x48
                    00310  IFDEF TEST
                    00311     DE  0X99,0X66   ; Uplimit,lowlimit negative supply (2-3V)
                    00312     DE  0X99,0X66   ; Uplimit,lowlimit positive supply (2-3V)
                    00313     DE  0X99,0X66   ; Uplimit,lowlimit control supply (2-3V)
                    00314     DE  0XD0,0X00   ; Uplinit,lowlimit temperature (-60/70C)
                    00315  ELSE
2106   0028 0012    00316     DE  0X28,0X12   ; Uplimit,lowlimit negative supply (+_10%)
2108   00B4 0096    00317     DE  0XB4,0X96   ; Uplimit,lowlimit positive supply (+-10%)
210A   00E1 00AF    00318     DE  0XE1,0XAF   ; Uplimit,lowlimit control supply (7-9V)
210C   0082 0000    00319     DE  0X82,0X00   ; Uplinit,lowlimit temperature (-60/70C)
                    00320  ENDIF
                    00321
                    00322 ; Target 0x50
210E   0002 0000    00323     DE  0X02,0X00   ; Fine tune Rx0 (Mid range)
2110   0002 0000    00324     DE  0X02,0X00   ; Fine tune Rx1 (Mid range)
2112   0002 0000    00325     DE  0X02,0X00   ; Fine tune Rx2 (Mid range)
                    00326
                    00327 ; Target 0x56
2114   0008 0000 0000 00328     DE  0X08,0X00,0X00  ; 1mw, zero cals Rx0
2117   0008 0000 0000 00329     DE  0X08,0X00,0X00  ; 1mw, zero cals Rx1
211A   0008 0000 0000 00330     DE  0X08,0X00,0X00  ; 1mw, zero cals Rx2
211D   0000          00331     DE  0X00        ; spare
                    00332
                    00333 ; Target 0x60
211E   0056 0066 0074 00334     DE  0X56,0X66,0X74  ; ID bytes for ID=666, fixed 8 byte length
                    00335                         ; address space 256  mode 1
2121   009A          00336     DE  0X9A        ; checksum
                    00337
                    00338
                    00339 ;================================================================================
                    00340
                    00341     END
```

```
MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : X---XXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX ----------------
0100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0140 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0180 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
01C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXX----- ----------------
0400 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0440 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0480 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
04C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0500 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0540 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0580 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
05C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0600 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0640 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0680 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
06C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0700 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXX------ ----------------
2000 : XXXX---X-------- ---------------- ---------------- ----------------
2100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XX-------------- ----------------

All other memory blocks unused.

Program Memory Words Used:  1154
Program Memory Words Free:  7038


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,    95 suppressed
```