```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0000                 00001 .;*****************************************************************************
                     00002 ; BradyBot the Robot
                     00003 ;*****************************************************************************
                     00004
                     00005 #include <p16F690.inc>
                     00001         LIST
                     00002 ; P16F690.INC  Standard Header File, Version 1.00    Microchip Technology, Inc.
                     00607         LIST
2007   30D4          00006          __config (_INTRC_OSC_NOCLKOUT & _WDT_OFF & _PWRTE_OFF & _MCLRE_OFF & _CP_OFF & _BOR_OFF & _IESO_
                           OFF & _FCMEN_OFF)
                     00007
                     00008         cblock 0x20
                     00009 ; variables for timer and flashing
  00000020           00010 Tick
  00000021           00011 hiB     ; MSB of 16-bit random number.
  00000022           00012 lowB    ; LSB of 16-bit random number.
  00000023           00013 Flash
  00000024           00014 Whisks
                     00015
  00000025           00016 PRTA    ; Staging area for PORTA
  00000026           00017 PRTB    ; Staging Area for PortB
  00000027           00018 PRTC    ; Staging area for PORTC
                     00019 ; variables for Pulsout
  00000028           00020 phiB    ; MSB of time.
  00000029           00021 plowB   ; LSB of time.
  0000002A           00022 pin     ; Pin number to pulse (0-7).
                     00023
  0000002B           00024 mhiB    ; MSB of time.
  0000002C           00025 mlowB   ; LSB of time.
                     00026
  0000002D           00027 temp    ; Temporary variables for time delay
  0000002E           00028 temp2   ; between pulses
                     00029
                     00030 ; values for wheel movement
  0000002F           00031 lwheel  ; value for left wheel
  00000030           00032 rwheel  ; value for right wheel
                     00033
  00000031           00034 seconds ; how many seconds to stay on current path
                     00035         endc
                     00036
                     00037 ;*******************************
                     00038 ; Stack
  0000004E           00039 _work   equ     h'4e'
  0000004F           00040 _status equ     h'4f'
                     00041
                     00042 ;*******************************
                     00043 ; seconds counters
  000000E1           00044 SEC5 equ d'225'
  00000087           00045 SEC3 equ d'135'
  0000002D           00046 SEC1 equ d'45'
  00000016           00047 SECHALF equ d'22'
                     00048
                     00049
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00050 ;*********************************
                     00051 ; PORTA bits
  00000000           00052 WHISKB equ 0
  00000001           00053 WHISKL equ 1
  00000002           00054 WHISKC equ 2
  00000003           00055 WHISKR equ 3
                     00056
                     00057 ;*********************************
                     00058 ; PORTB bits
  00000004           00059 GREEN  equ 4
  00000005           00060 YELLOW equ 5
  00000006           00061 WHEELR equ 6
  00000007           00062 WHEELL equ 7
                     00063
                     00064 ;*********************************
                     00065 ; PORTC bits
  00000000           00066 FLASHR equ 0
  00000001           00067 FLASHL equ 1
  00000004           00068 WHLEDB equ 4
  00000005           00069 WHLEDL equ 5
  00000006           00070 WHLEDC equ 6
  00000007           00071 WHLEDR equ 7
                     00072
                     00073 ;*********************************
                     00074 ; Left and Right values for SERVO motors
                     00075 ; Center point is 136
  00000064           00076 RFW equ d'100'
  000000A0           00077 LFW equ d'160'
  000000A0           00078 RBW equ d'160'
  00000064           00079 LBW equ d'100'
                     00080 ;*********************************
                     00081
0000                 00082         org 0
0000   2830          00083         goto    main
                     00084
0004                 00085         org 4
0004   2945          00086         goto    IntISR
                     00087
                     00088 ;***********************************************
                     00089 ;***********************************************
                     00090 ;* Put the Case functions here to allow for PCL
                     00091 ;***********************************************
                     00092 ;***********************************************
                     00093 ; determine which Tick
0005                 00094 CaseTick:
0005   0782          00095         addwf   PCL,f                ; jump to the right one
0006   2950          00096         goto    Tick0
0007   2950          00097         goto    Tick1
0008   2950          00098         goto    Tick2
0009   2951          00099         goto    Tick3
000A   2954          00100         goto    Tick4
000B   2954          00101         goto    Tick5
000C   2954          00102         goto    Tick6
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

000D   2955           00103         goto    Tick7
000E   2958           00104         goto    Tick8
000F   2958           00105         goto    Tick9
0010   2958           00106         goto    Tick10
0011   2959           00107         goto    Tick11
0012   295C           00108         goto    Tick12
0013   295C           00109         goto    Tick13
0014   295C           00110         goto    Tick14
0015   295D           00111         goto    Tick15
                      00112
0016                  00113 CaseWhisk:
0016   0782           00114         addwf   PCL,f                 ; jump to the right one
0017   2877           00115         goto    MoveCase ; None are pressed, Just go move normally
0018   2877           00116         goto    MoveCase ; 0001
0019   2867           00117         goto    Tright   ; 0010
001A   2867           00118         goto    Tright   ; 0011
001B   286F           00119         goto    TbackR   ; 0100
001C   286F           00120         goto    TbackR   ; 0101
001D   286F           00121         goto    TbackR   ; 0110
001E   286F           00122         goto    TbackR   ; 0111
001F   286B           00123         goto    Tleft    ; 1000
0020   286B           00124         goto    Tleft    ; 1001
0021   286F           00125         goto    TbackR   ; 1010
0022   2873           00126         goto    TbackL   ; 1011
0023   2873           00127         goto    TbackL   ; 1100
0024   2873           00128         goto    TbackL   ; 1101
0025   2873           00129         goto    TbackL   ; 1110
0026   2873           00130         goto    TbackL   ; 1111
                      00131
0027                  00132 CaseMove:
0027   0782           00133         addwf   PCL,f                 ; jump to the right one
0028   287A           00134         goto    Mfwd5  ; 0
0029   287E           00135         goto    Mfwd3  ; 1
002A   2882           00136         goto    Mfwd1  ; 2
002B   2884           00137         goto    Mbwd1  ; 3
002C   2886           00138         goto    Mright ; 4
002D   2888           00139         goto    Mleft  ; 5
002E   288A           00140         goto    Msit3  ; 6
002F   287E           00141         goto    Mfwd3  ; 7
                      00142
                      00143 ;************************************
                      00144 ;************************************
                      00145 ;* Now do the Main Loop
                      00146 ;************************************
                      00147 ;************************************
                      00148
0030                  00149 main:
0030   1303           00150         bcf             STATUS,RP1
0031   1683           00151         bsf             STATUS,RP0
0032   3007           00152         movlw   b'00000111'     ; configure timer0 from the processor clock
                      00153
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0033   0081           00154         movwf   OPTION_REG
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0034   0187           00155         clrf    TRISC
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0035   0186           00156         clrf    TRISB
0036   300A           00157         movlw   b'00001010' ; input bits for the whiskers
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0037   0085           00158         movwf   TRISA
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0038   0096           00159         movwf   IOCA
                      00160 ;       movlw   b'00000000'
                      00161 ;       movwf   WPUA
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0039   1381           00162         bcf     OPTION_REG,NOT_RABPU ; activate pullup resistors
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
003A   0195           00163         clrf    WPUA
                      00164
003B   1703           00165         bsf             STATUS,RP1
003C   1283           00166         bcf             STATUS,RP0
                      00167
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
003D   0196           00168         clrf    IOCB        ; make sure B doesn't case an interrupt
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
003E   019E           00169         clrf    ANSEL
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
003F   019F           00170         clrf    ANSELH
                      00171
0040   1303           00172         bcf             STATUS,RP1
0041   1283           00173         bcf             STATUS,RP0
                      00174
0042   0181           00175         clrf    TMR0
0043   0187           00176         clrf    PORTC
0044   01A7           00177         clrf    PRTC
0045   0186           00178         clrf    PORTB
0046   01A6           00179         clrf    PRTB
0047   0185           00180         clrf    PORTA
0048   01A5           00181         clrf    PRTA
0049   01A0           00182         clrf    Tick
004A   01A4           00183         clrf    Whisks
004B   01A3           00184         clrf    Flash
                      00185
                      00186 ; initialize random number
004C   300D           00187         movlw   d'13'               ; Arbitrary starting values
004D   00A2           00188         movwf   lowB
                      00189 ;       movlw   d'99'               ; for the shift register.
Warning[202]: Argument out of range.  Least significant bits used.
004E   308C           00190         movlw   RNDSEED             ; Choose a label as a random seed
004F   00A1           00191         movwf   hiB
                      00192
                      00193 ; turn on the interrupts
0050   168B           00194         bsf             INTCON,T0IE
0051   178B           00195         bsf             INTCON,GIE
0052   158B           00196         bsf             INTCON,RABIE
                      00197
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0053   20FB           00198         call    SitStill3
0054   208D           00199         call    MoveFwd5
0055   20FB           00200         call    SitStill3
                      00201
0056                  00202 Forever:
                      00203 ;*******************************
                      00204 ; Check the Whiskers
                      00205 ;*******************************
0056   12A7           00206         bcf     PRTC,WHLEDL
0057   13A7           00207         bcf     PRTC,WHLEDR
0058   1727           00208         bsf     PRTC,WHLEDC
                      00209
0059   18A4           00210         btfsc   Whisks,WHISKL         ; set the bits
005A   16A7           00211         bsf     PRTC,WHLEDL
                      00212
005B   19A4           00213         btfsc   Whisks,WHISKR         ; set the bits
005C   17A7           00214         bsf     PRTC,WHLEDR
                      00215
005D   1924           00216         btfsc   Whisks,WHISKC         ; set the bits
005E   1727           00217         bsf     PRTC,WHLEDC
                      00218
005F   1824           00219         btfsc   Whisks,WHISKB         ; set the bits
0060   1627           00220         bsf     PRTC,WHLEDB
                      00221
0061   0827           00222         movf    PRTC,w
0062   0087           00223         movwf   PORTC                 ; move it from staging to port
                      00224 ;
0063   0824           00225         movf    Whisks,w              ; Get the original value again
0064   01A4           00226         clrf    Whisks
0065   390F           00227         andlw   h'0f'                 ; Just look at the lower 4 bits
                      00228
0066   2816           00229         goto    CaseWhisk
                      00230
                      00231 ;*******************************
                      00232 ;* These are the reactions to the Whiskers
                      00233 ;*******************************
0067                  00234 Tright:
0067   20C9           00235         call    MoveBwdH
0068   20E7           00236         call    TurnRightH
0069   158B           00237         bsf             INTCON,RABIE
006A   288C           00238         goto    EndMove
                      00239
006B                  00240 Tleft:
006B   20C9           00241         call    MoveBwdH
006C   20F1           00242         call    TurnLeftH
006D   158B           00243         bsf             INTCON,RABIE
006E   288C           00244         goto    EndMove
                      00245
006F                  00246 TbackR:
006F   20BF           00247         call    MoveBwd1
0070   20D3           00248         call    TurnRight
0071   158B           00249         bsf             INTCON,RABIE
0072   288C           00250         goto    EndMove
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00251
0073                 00252 TbackL:
0073   20BF          00253         call    MoveBwd1
0074   20DD          00254         call    TurnLeft
0075   158B          00255         bsf             INTCON,RABIE
0076   288C          00256         goto    EndMove
                     00257
0077                 00258 MoveCase:
                     00259 ; Case the move
0077   0822          00260         movf    lowB,w                ; take advantage of the Random Number
0078   3907          00261         andlw   h'07'
0079   2827          00262         goto    CaseMove
                     00263
007A                 00264 Mfwd5:
007A   208D          00265         call    MoveFwd5
007B   288C          00266         goto    EndMove
007C                 00267 Mbwd5:
007C   2097          00268         call    MoveBwd5
007D   288C          00269         goto    EndMove
007E                 00270 Mfwd3:
007E   20A1          00271         call    MoveFwd3
007F   288C          00272         goto    EndMove
0080                 00273 Mbwd3:
0080   20AB          00274         call    MoveBwd3
0081   288C          00275         goto    EndMove
0082                 00276 Mfwd1:
0082   20B5          00277         call    MoveFwd1
0083   288C          00278         goto    EndMove
0084                 00279 Mbwd1:
0084   20BF          00280         call    MoveBwd1
0085   288C          00281         goto    EndMove
0086                 00282 Mright:
0086   20D3          00283         call    TurnRight
0087   288C          00284         goto    EndMove
0088                 00285 Mleft:
0088   20DD          00286         call    TurnLeft
0089   288C          00287         goto    EndMove
008A                 00288 Msit3:
008A   20FB          00289         call    SitStill3
008B   288C          00290         goto    EndMove
                     00291
008C                 00292 EndMove:
008C   2856          00293         goto    Forever
                     00294
008D                 00295 MoveFwd5:
008D   1626          00296         bsf     PRTB,GREEN
008E   12A6          00297         bcf     PRTB,YELLOW
008F   30E1          00298         movlw   SEC5                  ; move forward 5 seconds
0090   00B1          00299         movwf   seconds
0091   3064          00300         movlw   RFW
0092   00B0          00301         movwf   rwheel
0093   30A0          00302         movlw   LFW
0094   00AF          00303         movwf   lwheel
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0095   2101           00304         call    MoveWheel
0096   3400           00305         retlw   0h
                      00306
0097                  00307 MoveBwd5:
0097   1226           00308         bcf     PRTB,GREEN
0098   16A6           00309         bsf     PRTB,YELLOW
0099   30E1           00310         movlw   SEC5                    ; move backward 5 seconds
009A   00B1           00311         movwf   seconds
009B   30A0           00312         movlw   RBW
009C   00B0           00313         movwf   rwheel
009D   3064           00314         movlw   LBW
009E   00AF           00315         movwf   lwheel
009F   2101           00316         call    MoveWheel
00A0   3400           00317         retlw   0h
                      00318
00A1                  00319 MoveFwd3:
00A1   1626           00320         bsf     PRTB,GREEN
00A2   12A6           00321         bcf     PRTB,YELLOW
00A3   3087           00322         movlw   SEC3                    ; move forward 3 seconds
00A4   00B1           00323         movwf   seconds
00A5   3064           00324         movlw   RFW
00A6   00B0           00325         movwf   rwheel
00A7   30A0           00326         movlw   LFW
00A8   00AF           00327         movwf   lwheel
00A9   2101           00328         call    MoveWheel
00AA   3400           00329         retlw   0h
                      00330
00AB                  00331 MoveBwd3:
00AB   1226           00332         bcf     PRTB,GREEN
00AC   16A6           00333         bsf     PRTB,YELLOW
00AD   3087           00334         movlw   SEC3                    ; move backward 3 seconds
00AE   00B1           00335         movwf   seconds
00AF   30A0           00336         movlw   RBW
00B0   00B0           00337         movwf   rwheel
00B1   3064           00338         movlw   LBW
00B2   00AF           00339         movwf   lwheel
00B3   2101           00340         call    MoveWheel
00B4   3400           00341         retlw   0h
                      00342
00B5                  00343 MoveFwd1:
00B5   1626           00344         bsf     PRTB,GREEN
00B6   12A6           00345         bcf     PRTB,YELLOW
00B7   302D           00346         movlw   SEC1                    ; move forward 1 seconds
00B8   00B1           00347         movwf   seconds
00B9   3064           00348         movlw   RFW
00BA   00B0           00349         movwf   rwheel
00BB   30A0           00350         movlw   LFW
00BC   00AF           00351         movwf   lwheel
00BD   2101           00352         call    MoveWheel
00BE   3400           00353         retlw   0h
                      00354
00BF                  00355 MoveBwd1:
00BF   1226           00356         bcf     PRTB,GREEN
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

00C0   16A6           00357         bsf     PRTB,YELLOW
00C1   302D           00358         movlw   SEC1                    ; move backward 1 seconds
00C2   00B1           00359         movwf   seconds
00C3   30A0           00360         movlw   RBW
00C4   00B0           00361         movwf   rwheel
00C5   3064           00362         movlw   LBW
00C6   00AF           00363         movwf   lwheel
00C7   2101           00364         call    MoveWheel
00C8   3400           00365         retlw   0h
                      00366
00C9                  00367 MoveBwdH:
00C9   1226           00368         bcf     PRTB,GREEN
00CA   16A6           00369         bsf     PRTB,YELLOW
00CB   3016           00370         movlw   SECHALF                    ; move backward 1/2 seconds
00CC   00B1           00371         movwf   seconds
00CD   30A0           00372         movlw   RBW
00CE   00B0           00373         movwf   rwheel
00CF   3064           00374         movlw   LBW
00D0   00AF           00375         movwf   lwheel
00D1   2101           00376         call    MoveWheel
00D2   3400           00377         retlw   0h
                      00378
00D3                  00379 TurnRight:
00D3   1626           00380         bsf     PRTB,GREEN
00D4   16A6           00381         bsf     PRTB,YELLOW
00D5   302D           00382         movlw   SEC1                    ; Turn right for 1 second
00D6   00B1           00383         movwf   seconds
00D7   30A0           00384         movlw   RBW
00D8   00B0           00385         movwf   rwheel
00D9   30A0           00386         movlw   LFW
00DA   00AF           00387         movwf   lwheel
00DB   2101           00388         call    MoveWheel
00DC   3400           00389         retlw   0h
                      00390
00DD                  00391 TurnLeft:
00DD   1626           00392         bsf     PRTB,GREEN
00DE   16A6           00393         bsf     PRTB,YELLOW
00DF   302D           00394         movlw   SEC1                    ; Turn left for 1 second
00E0   00B1           00395         movwf   seconds
00E1   3064           00396         movlw   RFW
00E2   00B0           00397         movwf   rwheel
00E3   3064           00398         movlw   LBW
00E4   00AF           00399         movwf   lwheel
00E5   2101           00400         call    MoveWheel
00E6   3400           00401         retlw   0h
                      00402
00E7                  00403 TurnRightH:
00E7   1626           00404         bsf     PRTB,GREEN
00E8   16A6           00405         bsf     PRTB,YELLOW
00E9   3016           00406         movlw   SECHALF                    ; Turn right for 1/2 second
00EA   00B1           00407         movwf   seconds
00EB   30A0           00408         movlw   RBW
00EC   00B0           00409         movwf   rwheel
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

00ED   30A0           00410         movlw   LFW
00EE   00AF           00411         movwf   lwheel
00EF   2101           00412         call    MoveWheel
00F0   3400           00413         retlw   0h
                      00414
00F1                  00415 TurnLeftH:
00F1   1626           00416         bsf     PRTB,GREEN
00F2   16A6           00417         bsf     PRTB,YELLOW
00F3   3016           00418         movlw   SECHALF                 ; Turn left for 1/2 second
00F4   00B1           00419         movwf   seconds
00F5   3064           00420         movlw   RFW
00F6   00B0           00421         movwf   rwheel
00F7   3064           00422         movlw   LBW
00F8   00AF           00423         movwf   lwheel
00F9   2101           00424         call    MoveWheel
00FA   3400           00425         retlw   0h
                      00426
00FB                  00427 SitStill3:
00FB   1226           00428         bcf     PRTB,GREEN
00FC   12A6           00429         bcf     PRTB,YELLOW
00FD   3087           00430         movlw   SEC3                    ; Sit Still for 3 seconds
00FE   00B1           00431         movwf   seconds
00FF   2122           00432         call    SitStill
0100   3400           00433         retlw   0h
                      00434
                      00435 ;*******************************
                      00436 ; Move the Wheel
                      00437 ;*******************************
0101                  00438 MoveWheel:
0101   0831           00439         movf    seconds,w
0102   00AC           00440         movwf   mlowB
                      00441
0103   09AC           00442         comf    mlowB,f                 ; work with 2's comp
0104   0AAC           00443         incf    mlowB,f
                      00444
0105                  00445 MoveLoop:
0105   0830           00446         movf    rwheel,w                ; start with the right wheel
0106   00A9           00447         movwf   plowB
0107   01A8           00448         clrf    phiB
                      00449
0108   1726           00450         bsf     PRTB,WHEELR             ; Set the staging PORTB
0109   0826           00451         movf    PRTB,w
010A   0086           00452         movwf   PORTB
010B   2133           00453         call    Pulsout                 ; Pulse pin high for 10 x plow/hi cycles.
010C   1326           00454         bcf     PRTB,WHEELR
010D   0826           00455         movf    PRTB,w
010E   0086           00456         movwf   PORTB
                      00457
010F   082F           00458         movf    lwheel,w                ; now do the left wheel
0110   00A9           00459         movwf   plowB
0111   01A8           00460         clrf    phiB
                      00461
0112   17A6           00462         bsf     PRTB,WHEELL
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0113   0826           00463         movf    PRTB,w
0114   0086           00464         movwf   PORTB
0115   2133           00465         call    Pulsout               ; Pulse pin high for 10 x plow/hi cycles.
0116   13A6           00466         bcf     PRTB,WHEELL
0117   0826           00467         movf    PRTB,w
0118   0086           00468         movwf   PORTB
                      00469
0119   3000           00470         movlw   d'0'                  ; 20,000 u-cycle pulse
011A   00A9           00471         movwf   plowB
011B   3008           00472         movlw   d'8'
011C   00A8           00473         movwf   phiB
011D   2133           00474         call    Pulsout               ; Off cycle for the Pulse
                      00475
011E                  00476 MoveCnt:
011E   0AAC           00477         incf    mlowB,f               ; lowB * lowB+1.
011F   1D03           00478         btfss   STATUS,Z              ; Overflow in lowB?
0120   2905           00479         goto    MoveLoop              ; If not overflow, do it again.
                      00480
0121                  00481 MoveExit:
0121   3400           00482         retlw   0h
                      00483
0122                  00484 SitStill:
0122   0831           00485         movf    seconds,w
0123   00AC           00486         movwf   mlowB
                      00487
0124   09AC           00488         comf    mlowB,f               ; work with 2's comp
0125   0AAC           00489         incf    mlowB,f
                      00490
0126   1326           00491         bcf     PRTB,WHEELR           ; Clear the bits for the entire Loop
0127   13A6           00492         bcf     PRTB,WHEELL
0128   0826           00493         movf    PRTB,w
0129   0086           00494         movwf   PORTB
                      00495
012A                  00496 SitLoop:
012A   30C8           00497         movlw   d'200'                ; 22,000 u-cycle pulse
012B   00A9           00498         movwf   plowB
012C   3008           00499         movlw   d'8'
012D   00A8           00500         movwf   phiB
012E   2133           00501         call    Pulsout               ; Off cycle for the Pulse
                      00502
012F   0AAC           00503         incf    mlowB,f               ; lowB * lowB+1.
0130   1D03           00504         btfss   STATUS,Z              ; Overflow in lowB?
0131   292A           00505         goto    SitLoop               ; If not overflow, do it again.
                      00506
0132   3400           00507         retlw   0h
                      00508
                      00509 ;
                      00510 ; Pause for the Pulsout Value
                      00511 ; This assumes that the bits have already been set
                      00512 ;
0133                  00513 Pulsout:
0133   09A8           00514         comf    phiB,f                ; Take twos complement
0134   09A9           00515         comf    plowB,f               ; of the 16-bit counter
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0135   0AA9           00516         incf    plowB,f
0136   1903           00517         btfsc   STATUS,Z              ; If zero, lowB overflowed,
0137   0AA8           00518         incf    phiB,f                ; so carry into hiB.
                      00519
                      00520 ; The main timing loop. Remove the nops for 5-cycle resolution.
0138                  00521 Pulsout_loop:
0138   2939           00522         goto    $+1                   ; Two-cycle "nop."
0139   293A           00523         goto    $+1                   ; Two-cycle "nop."
013A   0000           00524         nop
013B   0AA9           00525         incf    plowB,f               ; lowB = lowB+1.
013C   1903           00526         btfsc   STATUS,Z              ; Overflow in lowB?
013D   0FA8           00527         incfsz  phiB,f                ; Yes: hiB=hiB+1, watch for overflow.
013E   2938           00528         goto    Pulsout_loop          ; If not overflow, do it again.
013F   3400           00529         retlw   0h
                      00530
                      00531 ; Delay routine for demonstration. Not required by Pulsout.
                      00532
0140                  00533 delay:
0140   0BAD           00534         decfsz  temp,f                ; Time delay to provide spacing
0141   2940           00535         goto    delay
0142   0BAE           00536         decfsz  temp2,f               ; between pulses.
0143   2940           00537         goto    delay
0144   3400           00538         retlw   0h
                      00539
                      00540
                      00541
                      00542 ;***********************************
                      00543 ; Inturupt Handler
                      00544 ;***********************************
0145                  00545 IntISR:
0145   00CE           00546         movwf   _work                          ; save W
0146   0E03           00547         swapf   STATUS,w
0147   00CF           00548         movwf   _status
                      00549 ;***********************************
0148   1A8B           00550         btfsc   INTCON,T0IE                   ; make sure TMR0 is set
0149   1D0B           00551         btfss   INTCON,T0IF                   ;  and that this was a TMR0 interrupt
014A   296D           00552         goto    IntRegA               ; Go check to see if it was a RegA Interupt
014B   110B           00553         bcf             INTCON,T0IF                   ; clear the interrupt
                      00554 ;***********************************
                      00555 ; This is set to Tick every 1/16 second
014C   0AA0           00556         incf    Tick,f                ; keep track of the Tick count
                      00557
014D   0820           00558         movf    Tick,w
014E   390F           00559         andlw   h'0f'                 ; which 1/16th of a second is it?
014F   2805           00560         goto    CaseTick
                      00561 ;***********************************
                      00562 ;***********************************
                      00563 ;***********************************
                      00564 ;***********************************
                      00565 ; Timer Functions
                      00566 ;***********************************
                      00567 ;***********************************
                      00568 ;***********************************
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00569 ;**********************************
0150                 00570 Tick0:
0150                 00571 Tick1:
0150                 00572 Tick2:
0150   2968          00573         goto    IntEnd
0151                 00574 Tick3:
0151   217D          00575         call    Random                ; Generate a random number
0152   0822          00576         movf    lowB,w                ; Move it into W
                     00577 ;       movlw   b'1001'               ; Use this for alternate flash
0153   295F          00578         goto    TickSet               ; Go light the lights
0154                 00579 Tick4:
0154                 00580 Tick5:
0154                 00581 Tick6:
0154   2968          00582         goto    IntEnd
0155                 00583 Tick7:
0155   217D          00584         call    Random                ; Generate a random number
0156   0822          00585         movf    lowB,w                ; Move it into W
                     00586 ;       movlw   b'0110'               ; Use this for alternate flash
0157   295F          00587         goto    TickSet               ; Go light the lights
0158                 00588 Tick8:
0158                 00589 Tick9:
0158                 00590 Tick10:
0158   2968          00591         goto    IntEnd
0159                 00592 Tick11:
0159   217D          00593         call    Random                ; Generate a random number
015A   0822          00594         movf    lowB,w                ; Move it into W
                     00595 ;       movlw   b'1001'               ; Use this for alternate flash
015B   295F          00596         goto    TickSet               ; Go light the lights
015C                 00597 Tick12:
015C                 00598 Tick13:
015C                 00599 Tick14:
015C   2968          00600         goto    IntEnd
015D                 00601 Tick15:
015D   217D          00602         call    Random                ; Generate a random number
015E   0822          00603         movf    lowB,w                ; Move it into W
                     00604 ;       movlw   b'0110'               ; Use this for alternate flash
015F                 00605 TickSet:                           ; light the lights
015F   00A3          00606         movwf   Flash
0160   1027          00607         bcf     PRTC,FLASHR           ; clear the bits
0161   10A7          00608         bcf     PRTC,FLASHL           ; Use PRTC to avoid destructive read
                     00609
0162   18A3          00610         btfsc   Flash,1               ; set the bits
0163   14A7          00611         bsf     PRTC,FLASHL
                     00612
0164   1823          00613         btfsc   Flash,0               ; set the bits
0165   1427          00614         bsf     PRTC,FLASHR
                     00615
0166   0827          00616         movf    PRTC,w
0167   0087          00617         movwf   PORTC                 ; move it from staging to port
                     00618
0168                 00619 IntEnd:
                     00620 ; **************************************
0168   0E4F          00621         swapf   _status,W
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0169   0083           00622         movwf   STATUS
016A   0ECE           00623         swapf   _work,f
016B   0E4E           00624         swapf   _work,w
016C   0009           00625         retfie
                      00626
                      00627 ; *************************************
                      00628 ; *************************************
                      00629 ; *************************************
                      00630 ; *************************************
                      00631 ; * Check for RegA Interupt
                      00632 ; *************************************
                      00633 ; *************************************
                      00634 ; *************************************
                      00635 ; *************************************
016D                  00636 IntRegA:
                      00637 ;***********************************
016D   198B           00638         btfsc   INTCON,RABIE                  ; make sure RB is set
016E   1C0B           00639         btfss   INTCON,RABIF                  ;  and that this was a TMR0 interrupt
016F   2968           00640         goto    IntEnd                    ; Go check to see if it was a RegB Interupt
0170   0805           00641         movf    PORTA,w
0171   100B           00642         bcf             INTCON,RABIF                  ; clear the interrupt
0172   118B           00643         bcf     INTCON,RABIE
                      00644 ;***********************************
                      00645 ;******************              ;  check for Whiskers
0173   0805           00646         movf    PORTA,w
0174   00A5           00647         movwf   PRTA
0175   390A           00648         andlw   b'00001010'           ; only use two bits so far
                      00649 ;       swapf   PRTA,w                ; Swap if around
                      00650
0176   00A4           00651         movwf   Whisks                ; Update the Whisker settings
                      00652
0177   30FF           00653         movlw   h'ff'                 ; stop the move counter
0178   00AC           00654         movwf   mlowB
0179   00A9           00655         movwf   plowB
017A   00A8           00656         movwf   phiB
                      00657
                      00658 ;       bsf     PRTC,2
                      00659 ;       bsf     PRTC,3
                      00660
017B   158B           00661         bsf             INTCON,RABIE
                      00662
017C   2968           00663         goto    IntEnd
                      00664 ;******************
                      00665
                      00666
                      00667 ;
                      00668 ; Generate a Random Number
                      00669 ;
017D                  00670 Random:
017D   0821           00671         movf    hiB,w                 ; First, ensure that hiB and lowB aren't
017E   0422           00672         iorwf   lowB,w                ; all zeros. If they are, NOT hiB to FFh.
017F   1903           00673         btfsc   STATUS,Z              ; Otherwise, leave hiB and lowB as is.
0180   09A1           00674         comf    hiB,f
```

```
LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

0181   3080           00675         movlw   0x80                ; We want to XOR hiB.7, hiB.6, hiB.4
0182   1B21           00676         btfsc   hiB,d'6'            ; and lowB.3 together in W. Rather than
0183   06A1           00677         xorwf   hiB,f               ; try to line up these bits, we just
0184   1A21           00678         btfsc   hiB,d'4'            ; check to see whether a bit is a 1. If it
0185   06A1           00679         xorwf   hiB,f               ; is, XOR 80h into hiB. If it isn't,
0186   19A2           00680         btfsc   lowB,d'3'           ; do nothing. When we're done, the
0187   06A1           00681         xorwf   hiB,f               ; XOR of the 4 bits will be in hiB.7.
0188   0D21           00682         rlf     hiB,w               ; Move hiB.7 into carry.
0189   0DA2           00683         rlf     lowB,f              ; Rotate c into lowB.0, lowB.7 into c.
018A   0DA1           00684         rlf     hiB,f               ; Rotate c into hiB.0.
018B   3400           00685         retlw   0h
                      00686
018C                  00687 RNDSEED:
018C   0000           00688         nop
                      00689
                      00690     end
```

SYMBOL TABLE

LABEL	VALUE
.	00000000
ABDEN	00000000
ABDOVF	00000007
ADCON0	0000001F
ADCON1	0000009F
ADCS0	00000004
ADCS1	00000005
ADCS2	00000006
ADDEN	00000003
ADFM	00000007
ADIE	00000006
ADIF	00000006
ADON	00000000
ADRESH	0000001E
ADRESL	0000009E
ANS0	00000000
ANS1	00000001
ANS10	00000002
ANS11	00000003
ANS2	00000002
ANS3	00000003
ANS4	00000004
ANS5	00000005
ANS6	00000006
ANS7	00000007
ANS8	00000000
ANS9	00000001
ANSEL	0000011E
ANSELH	0000011F
BAUDCTL	0000009B
BF	00000000
BRG0	00000000
BRG1	00000001
BRG10	00000002
BRG11	00000003
BRG12	00000004
BRG13	00000005
BRG14	00000006
BRG15	00000007
BRG16	00000003
BRG2	00000002
BRG3	00000003
BRG4	00000004
BRG5	00000005
BRG6	00000006
BRG7	00000007
BRG8	00000000
BRG9	00000001
BRGH	00000002
C	00000000
C1CH0	00000000
C1CH1	00000001
C1IE	00000005

SYMBOL TABLE

LABEL	VALUE
C1IF	00000005
C1OE	00000005
C1ON	00000007
C1OUT	00000006
C1POL	00000004
C1R	00000002
C1SEN	00000005
C1VREN	00000007
C2CH0	00000000
C2CH1	00000001
C2IE	00000006
C2IF	00000006
C2OE	00000005
C2ON	00000007
C2OUT	00000006
C2POL	00000004
C2R	00000002
C2REN	00000004
C2SYNC	00000000
C2VREN	00000006
CCP1CON	00000017
CCP1IE	00000002
CCP1IF	00000002
CCP1M0	00000000
CCP1M1	00000001
CCP1M2	00000002
CCP1M3	00000003
CCPR1H	00000016
CCPR1L	00000015
CHS0	00000002
CHS1	00000003
CHS2	00000004
CHS3	00000005
CKE	00000006
CKP	00000004
CM1CON0	00000119
CM2CON0	0000011A
CM2CON1	0000011B
CREN	00000004
CSRC	00000007
CaseMove	00000027
CaseTick	00000005
CaseWhisk	00000016
D	00000005
DATA_ADDRESS	00000005
DC	00000001
DC1B0	00000004
DC1B1	00000005
D_A	00000005
ECCPAS	0000001D
ECCPAS0	00000004
ECCPAS1	00000005
ECCPAS2	00000006

SYMBOL TABLE

LABEL	VALUE
ECCPASE	00000007
EEADR	0000010D
EEADRH	0000010F
EECON1	0000018C
EECON2	0000018D
EEDAT	0000010C
EEDATA	0000010C
EEDATH	0000010E
EEIE	00000004
EEIF	00000004
EEPGD	00000007
EndMove	0000008C
F	00000001
FERR	00000002
FLASHL	00000001
FLASHR	00000000
FSR	00000004
Flash	00000023
Forever	00000056
GIE	00000007
GO	00000001
GO_DONE	00000001
GREEN	00000004
HTS	00000002
I2C_DATA	00000005
I2C_READ	00000002
I2C_START	00000003
I2C_STOP	00000004
INDF	00000000
INTCON	0000000B
INTE	00000004
INTEDG	00000006
INTF	00000001
IOC	00000096
IOC0	00000000
IOC1	00000001
IOC2	00000002
IOC3	00000003
IOC4	00000004
IOC5	00000005
IOCA	00000096
IOCA0	00000000
IOCA1	00000001
IOCA2	00000002
IOCA3	00000003
IOCA4	00000004
IOCA5	00000005
IOCB	00000116
IOCB4	00000004
IOCB5	00000005
IOCB6	00000006
IOCB7	00000007
IRCF0	00000004

SYMBOL TABLE

LABEL	VALUE
IRCF1	00000005
IRCF2	00000006
IRP	00000007
IntEnd	00000168
IntISR	00000145
IntRegA	0000016D
LBW	00000064
LFW	000000A0
LTS	00000001
MC1OUT	00000007
MC2OUT	00000006
MSK	00000093
Mbwd1	00000084
Mbwd3	00000080
Mbwd5	0000007C
Mfwd1	00000082
Mfwd3	0000007E
Mfwd5	0000007A
Mleft	00000088
MoveBwd1	000000BF
MoveBwd3	000000AB
MoveBwd5	00000097
MoveBwdH	000000C9
MoveCase	00000077
MoveCnt	0000011E
MoveExit	00000121
MoveFwd1	000000B5
MoveFwd3	000000A1
MoveFwd5	0000008D
MoveLoop	00000105
MoveWheel	00000101
Mright	00000086
Msit3	0000008A
NOT_A	00000005
NOT_ADDRESS	00000005
NOT_BOR	00000000
NOT_DONE	00000001
NOT_PD	00000003
NOT_POR	00000001
NOT_RABPU	00000007
NOT_T1SYNC	00000002
NOT_TO	00000004
NOT_W	00000002
NOT_WRITE	00000002
OERR	00000001
OPTION_REG	00000081
OSCCON	0000008F
OSCTUNE	00000090
OSFIE	00000007
OSFIF	00000007
OSTS	00000003
P	00000004
P1M0	00000006

SYMBOL TABLE

LABEL	VALUE
P1M1	00000007
PCL	00000002
PCLATH	0000000A
PCON	0000008E
PDC0	00000000
PDC1	00000001
PDC2	00000002
PDC3	00000003
PDC4	00000004
PDC5	00000005
PDC6	00000006
PEIE	00000006
PIE1	0000008C
PIE2	0000008D
PIR1	0000000C
PIR2	0000000D
PORTA	00000005
PORTB	00000006
PORTC	00000007
PR2	00000092
PRSEN	00000007
PRTA	00000025
PRTB	00000026
PRTC	00000027
PS0	00000000
PS1	00000001
PS2	00000002
PSA	00000003
PSSAC0	00000002
PSSAC1	00000003
PSSBD0	00000000
PSSBD1	00000001
PSTRCON	0000019D
PULSR	00000002
PULSS	00000003
PWM1CON	0000001C
Pulsout	00000133
Pulsout_loop	00000138
R	00000002
RABIE	00000003
RABIF	00000000
RBW	000000A0
RCIDL	00000006
RCIE	00000005
RCIF	00000005
RCREG	0000001A
RCSTA	00000018
RD	00000000
READ_WRITE	00000002
RFW	00000064
RNDSEED	0000018C
RP0	00000005
RP1	00000006

SYMBOL TABLE

LABEL	VALUE
RX9	00000006
RX9D	00000000
R_W	00000002
Random	0000017D
S	00000003
SBOREN	00000004
SCKP	00000004
SCS	00000000
SEC1	0000002D
SEC3	00000087
SEC5	000000E1
SECHALF	00000016
SENB	00000003
SMP	00000007
SPBRG	00000099
SPBRGH	0000009A
SPEN	00000007
SR0	00000006
SR1	00000007
SRCON	0000019E
SREN	00000005
SSPADD	00000093
SSPBUF	00000013
SSPCON	00000014
SSPEN	00000005
SSPIE	00000003
SSPIF	00000003
SSPM0	00000000
SSPM1	00000001
SSPM2	00000002
SSPM3	00000003
SSPMSK	00000093
SSPOV	00000006
SSPSTAT	00000094
STATUS	00000003
STRA	00000000
STRB	00000001
STRC	00000002
STRD	00000003
STRSYNC	00000004
SWDTEN	00000000
SYNC	00000004
SitLoop	0000012A
SitStill	00000122
SitStill3	000000FB
T0CS	00000005
T0IE	00000005
T0IF	00000002
T0SE	00000004
T1CKPS0	00000004
T1CKPS1	00000005
T1CON	00000010
T1GINV	00000007

SYMBOL TABLE

LABEL	VALUE
T1GSS	00000001
T1IE	00000000
T1IF	00000000
T1OSCEN	00000003
T2CKPS0	00000000
T2CKPS1	00000001
T2CON	00000012
T2IE	00000001
T2IF	00000001
TMR0	00000001
TMR1CS	00000001
TMR1GE	00000006
TMR1H	0000000F
TMR1IE	00000000
TMR1IF	00000000
TMR1L	0000000E
TMR1ON	00000000
TMR2	00000011
TMR2IE	00000001
TMR2IF	00000001
TMR2ON	00000002
TOUTPS0	00000003
TOUTPS1	00000004
TOUTPS2	00000005
TOUTPS3	00000006
TRISA	00000085
TRISA0	00000000
TRISA1	00000001
TRISA2	00000002
TRISA3	00000003
TRISA4	00000004
TRISA5	00000005
TRISB	00000086
TRISB4	00000004
TRISB5	00000005
TRISB6	00000006
TRISB7	00000007
TRISC	00000087
TRISC0	00000000
TRISC1	00000001
TRISC2	00000002
TRISC3	00000003
TRISC4	00000004
TRISC5	00000005
TRISC6	00000006
TRISC7	00000007
TRMT	00000001
TUN0	00000000
TUN1	00000001
TUN2	00000002
TUN3	00000003
TUN4	00000004
TX9	00000006

SYMBOL TABLE

LABEL	VALUE
TX9D	00000000
TXEN	00000005
TXIE	00000004
TXIF	00000004
TXREG	00000019
TXSTA	00000098
TbackL	00000073
TbackR	0000006F
Tick	00000020
Tick0	00000150
Tick1	00000150
Tick10	00000158
Tick11	00000159
Tick12	0000015C
Tick13	0000015C
Tick14	0000015C
Tick15	0000015D
Tick2	00000150
Tick3	00000151
Tick4	00000154
Tick5	00000154
Tick6	00000154
Tick7	00000155
Tick8	00000158
Tick9	00000158
TickSet	0000015F
Tleft	0000006B
Tright	00000067
TurnLeft	000000DD
TurnLeftH	000000F1
TurnRight	000000D3
TurnRightH	000000E7
UA	00000001
ULPWUE	00000005
VCFG	00000006
VP6EN	00000004
VR0	00000000
VR1	00000001
VR2	00000002
VR3	00000003
VRCON	00000118
VRR	00000005
W	00000000
WCOL	00000007
WDTCON	00000097
WDTPS0	00000001
WDTPS1	00000002
WDTPS2	00000003
WDTPS3	00000004
WHEELL	00000007
WHEELR	00000006
WHISKB	00000000
WHISKC	00000002

SYMBOL TABLE

LABEL	VALUE
WHISKL	00000001
WHISKR	00000003
WHLEDB	00000004
WHLEDC	00000006
WHLEDL	00000005
WHLEDR	00000007
WPU	00000095
WPUA	00000095
WPUA0	00000000
WPUA1	00000001
WPUA2	00000002
WPUA4	00000004
WPUA5	00000005
WPUB	00000115
WPUB4	00000004
WPUB5	00000005
WPUB6	00000006
WPUB7	00000007
WR	00000001
WREN	00000002
WRERR	00000003
WUE	00000001
Whisks	00000024
YELLOW	00000005
Z	00000002
_BOR_NSLEEP	00003EFF
_BOR_OFF	00003CFF
_BOR_ON	00003FFF
_BOR_SBODEN	00003DFF
_CPD_OFF	00003FFF
_CPD_ON	00003F7F
_CP_OFF	00003FFF
_CP_ON	00003FBF
_EC_OSC	00003FFB
_EXTRC	00003FFF
_EXTRCIO	00003FFE
_EXTRC_OSC_CLKOUT	00003FFF
_EXTRC_OSC_NOCLKOUT	00003FFE
_FCMEN_OFF	000037FF
_FCMEN_ON	00003FFF
_HS_OSC	00003FFA
_IESO_OFF	00003BFF
_IESO_ON	00003FFF
_INTOSC	00003FFD
_INTOSCIO	00003FFC
_INTRC_OSC_CLKOUT	00003FFD
_INTRC_OSC_NOCLKOUT	00003FFC
_LP_OSC	00003FF8
_MCLRE_OFF	00003FDF
_MCLRE_ON	00003FFF
_PWRTE_OFF	00003FFF
_PWRTE_ON	00003FEF
_WDT_OFF	00003FF7

SYMBOL TABLE

LABEL	VALUE
_WDT_ON	00003FFF
_XT_OSC	00003FF9
__16F690	00000001
_status	0000004F
_work	0000004E
delay	00000140
hiB	00000021
lowB	00000022
lwheel	0000002F
main	00000030
mhiB	0000002B
mlowB	0000002C
phiB	00000028
pin	0000002A
plowB	00000029
rwheel	00000030
seconds	00000031
temp	0000002D
temp2	0000002E

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

```
0000 : X---XXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
00C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0100 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0140 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0180 : XXXXXXXXXXXXX--- ---------------- ---------------- ----------------
2000 : -------X-------- ---------------- ---------------- ----------------
```

All other memory blocks unused.

Program Memory Words Used: 394
Program Memory Words Free: 3702

Errors   :   0
Warnings :   1 reported,   0 suppressed
Messages :   10 reported,   0 suppressed