Electromechanical Switch Replacement

Replacing Electromechanical Switches

Author: Lon Glazner
Solutions Cube
Chico, CA
email: solcubed@solutions-cubed.com

DESIGN IDEA

The Smart Switch is the ultimate in electronic switch design. No switch has ever had such an immense impact on mankind's technical progress. It is assumed, and correctly so, that any device with such far reaching effects must be based on a product by Microchip Technology Inc. The Smart Switch provides a PWM output at 61Hz. This PWM output can be used to control pass elements or low-side switches(switches to ground).

In the case of pass elements this switch design could be used to produce variable DC voltages and/or high current sources through PNP pass elements. With LC filters and spike protection diodes the device could work well as a battery charger or even a step-down voltage regulator. In low cost applications the LC filters could be left off. It could also be useful as a volume control, or as an element in an automatic gain control circuit.

For extremely low cost designs, a switch to ground could be controlled with the PWM output. For easy interface, a logic level N-channel FET could easily be used to switch 80V at 8A to ground. This could allow the switch to control very large motors with varying speeds.

In this design, a low cost transistor supply is used to provide the current required to run the PIC12CXXX at 5V. Various transistors could be specified in this supply circuit based on the supply voltage accepted.

HARDWARE METHODOLOGY

As discussed earlier, the basis for this design is a switching element that allows for adjustable control of various loads. Because the PIC12CXXX allows the designer to use the internal MCLR and the internal 4Mhz oscillator, this circuit can be built with a minimum of expense. Even the pull up resistors on the manual control switches could be replaced by the PIC12CXXX's internal pull ups. The generic aspects of the PWM circuit allow it to fit into many applications. In this particular example the control of the PWM duty cycle can occur in two ways. The first, and primary method, is through manual control. The Smart Switch polls the input switches. If the increase switch is pressed, it causes the PWM duty cycle to increment. If the decrease switch is pressed, it causes the PWM signal to decrement. The software debounce implemented is 15ms in duration. This debounce routine also serves as a typematic type function. In other words, if you continue to press one of the switches it will continuously increment or decrement the duty cycle. This can give the feel of smooth manual control of a load supply in designs where a potentiometer can't be used. The manual switches could easily be replaced with comparator outputs. In such a design, voltage thresholds could trigger a slewing of the duty cycle. This could be used to implement dual level float chargers for high capacity batteries or automated transition to a battery backup system.

The PWM output of 61Hz was chosen for software reasons, and because this frequency is about as slow as you can get and still generate smooth motor control with direct PWM.

The PIC12CXXX is powered by a zener diode controlled transistor supply. The emitter of the transistor will maintain a voltage of roughly 0.6V below the value of the zener diode. In this case it is roughly 5V.

Using the bill of materials below the Smart Switch could act as a 20A switch with a 35V supply. Package power dissipation for the logic level FET would have to be addressed.

TABLE 1: BILL OF MATERIALS

Designation	Part	Specifications
R1	51 ohm	5% 1/8 watt
R2	10K ohm	5% 1/8 watt
R3	10K ohm	5% 1/8 watt
R4	4.7K ohm	5% 1/8 watt
C1	10uF	10V
D1	5.6V	1/2 watt
Q1	2N3904	max 40V Vceo
Q4	F25N05L	50V @ 25A
U1	12C508-4/P	neato

SOFTWARE METHODOLOGY

The heart of the Smart Switch is the standard 15us delay. The usefulness of varying PWM outputs can be noted by the inclusion of PWM hardware in more complex PICmicro chips. If people didn't want it, it wouldn't be there. But many of Microchips customers still make use of the PIC16C5X cores for their designs. Multiplexing different time based functions without interrupts can be daunting, but with a little work, all PICmicros are capable of multitasking time based functions.

In this design, we started with an 15us standardized delay. Every routine in the program uses this delay. There is nothing sacred about the delay period. We set the TMR0 prescaler to 64us. The 15us delay simply checks TMR0 for a rollover. If a TMR0 rollover occurs the delay routine tests the output pin for the current logic level. It then loads the TMR0 register with either the positive duty cycle or the negative duty cycle duration and toggles the output pin. As long as the delay routine is called every 64us you can't miss a TMR0 rollover. And therefore the PWM output will remain constant.

With a standardized delay as a building block its easy to add other time based functions to the design. In the case of this design we added a 2400 baud serial interface that can be used to modify the duty cycle storage register. With this routine the PWM duty cycle could be continually updated based on the needs of your system. Additional I/Os are available to interface to EEPROM or to drive status LEDs. Microchip's new 16 byte EEPROM would be ideal for storage of the current duty cycle value, which would add the aspect of non-volatility to the Smart Switch.

RAM Used:	6 bytes
Subroutine Bytes:	110
Program Bytes (as presented):	139
Program Cycles (min, no PWM adjustments):	35
Program Cycles (max, switch press/15ms debounce loop):	14,191

FIGURE 1: SMART SWITCH – ADJUSTABLE HIGH CURRENT PASS ELEMENT

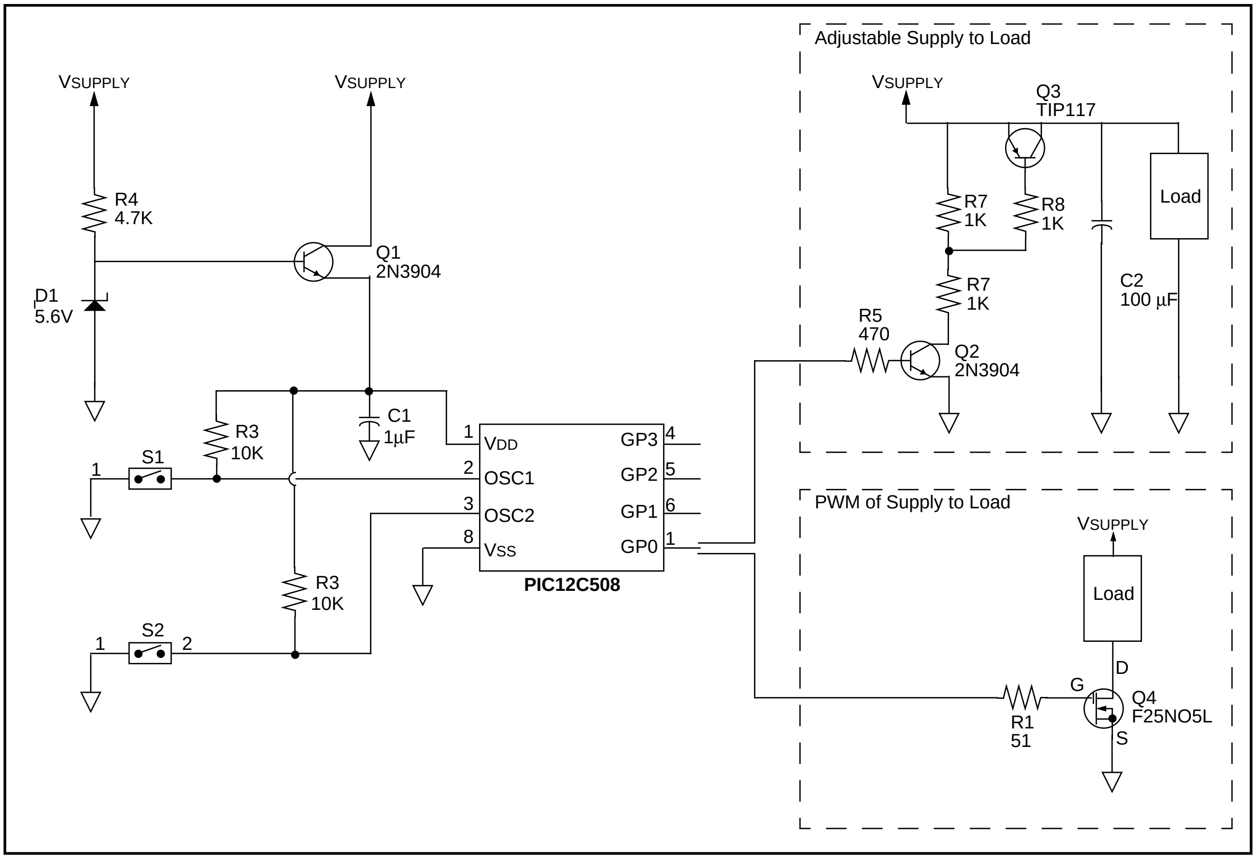


FIGURE 2: MAIN ROUTINE FLOWCHART

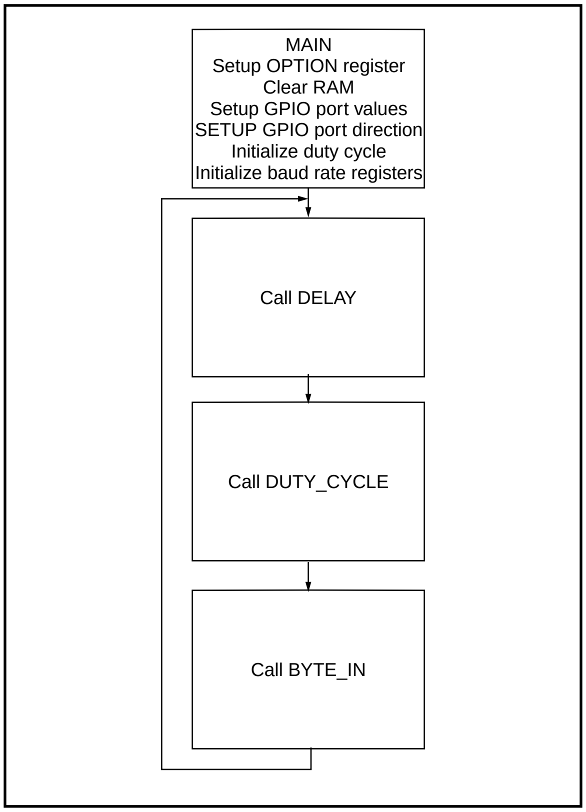


FIGURE 3: DELAY ROUTINE FLOWCHART

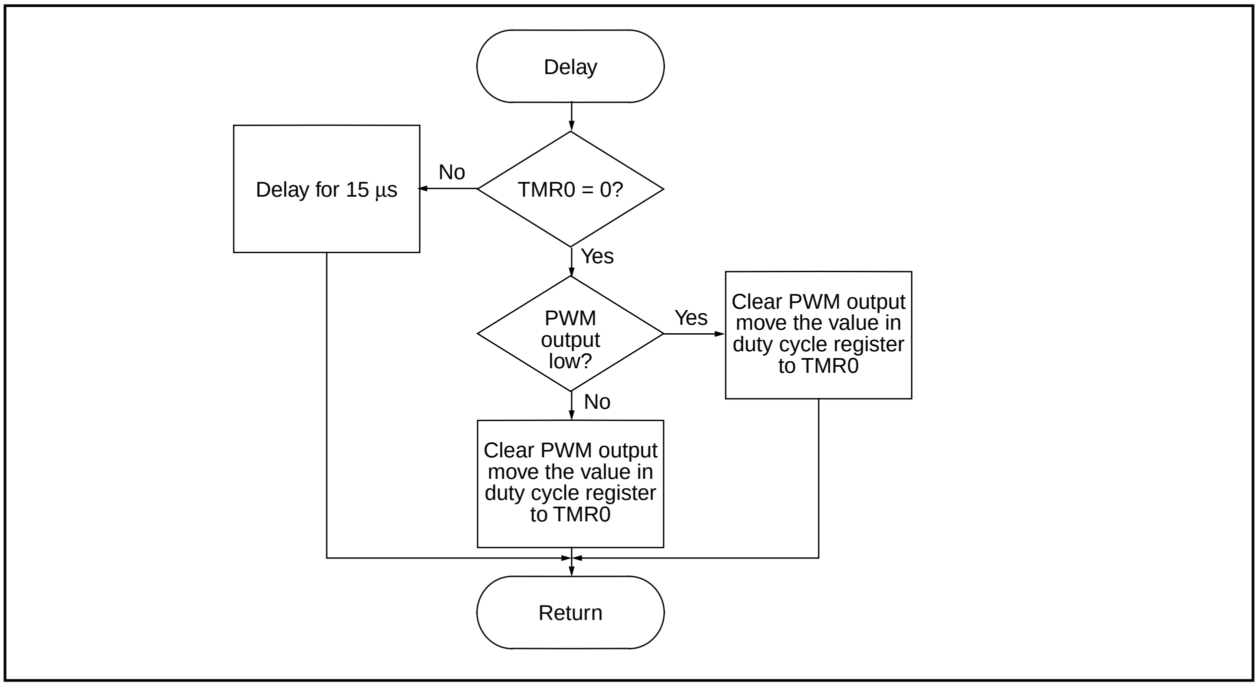

Delay
TMR0 = 0?
No
Delay for 15 µs
Yes
PWM output low?
Yes
Clear PWM output move the value in duty cycle register to TMR0
No
Clear PWM output move the value in duty cycle register to TMR0
Return

FIGURE 4: DUTY_CYCLE ROUTINE FLOWCHART

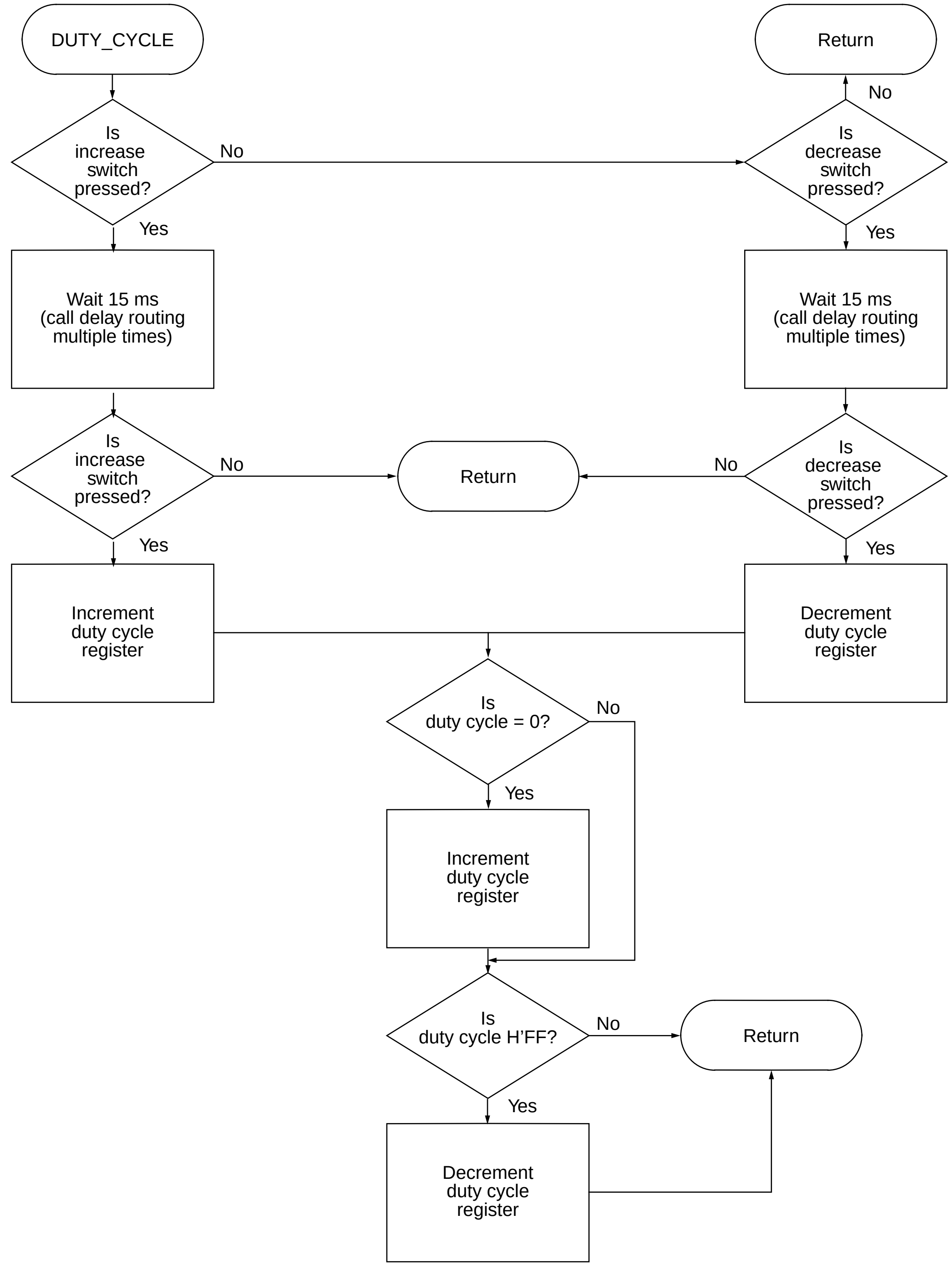
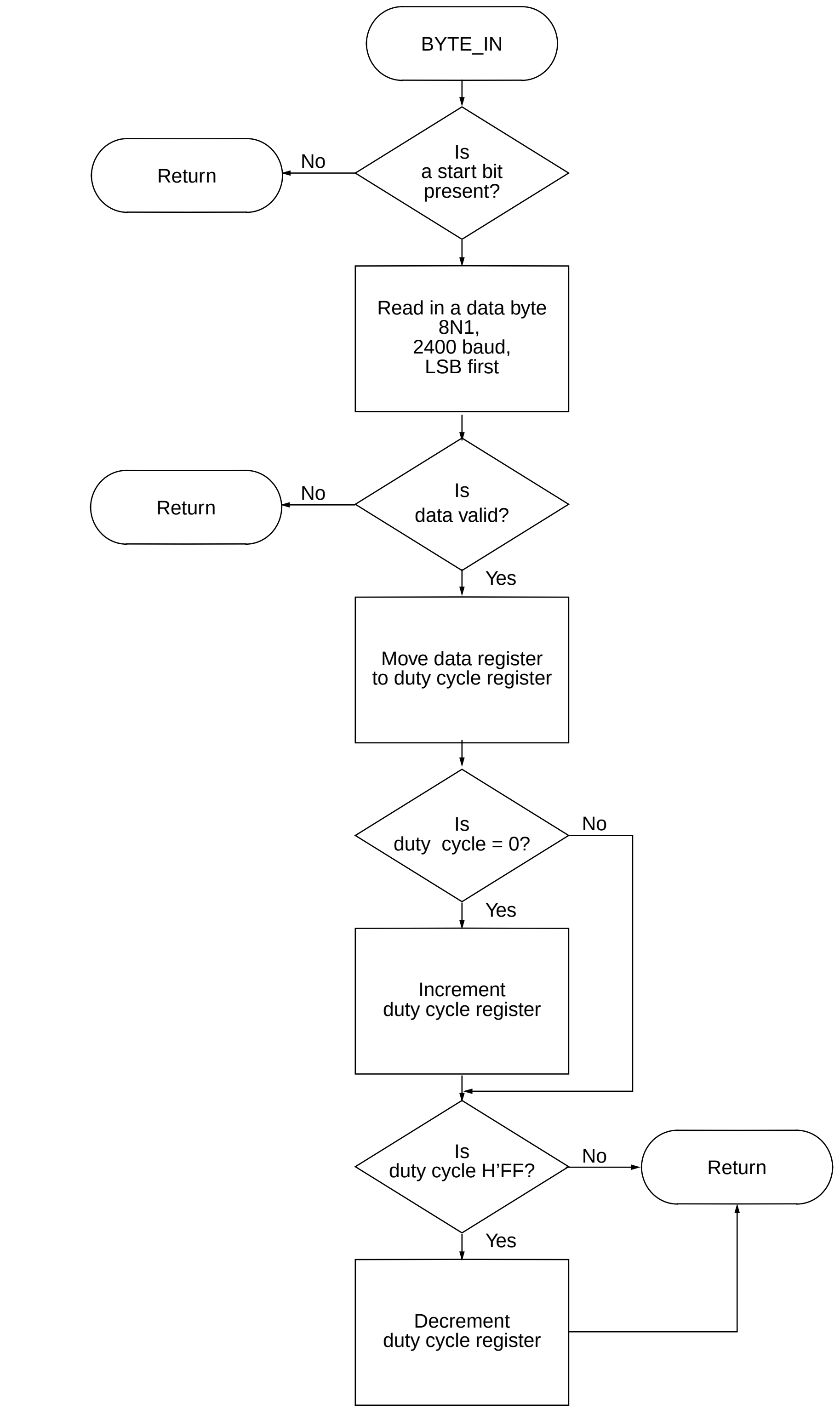

DUTY_CYCLE
Is increase switch pressed?
No
Yes
Wait 15 ms (call delay routing multiple times)
Is increase switch pressed?
No
Yes
Return
Increment duty cycle register
Return
No
Is decrease switch pressed?
Yes
Wait 15 ms (call delay routing multiple times)
Is decrease switch pressed?
No
Yes
Decrement duty cycle register
Is duty cycle = 0?
No
Yes
Increment duty cycle register
Is duty cycle H’FF?
No
Return
Yes
Decrement duty cycle register

FIGURE 5: BYTE_IN ROUTINE FLOWCHART



APPENDIX A: SOURCE CODE

```
MPASM 01.40 Released           SMART3.ASM   6-13-1997  17:04:51         PAGE  1

             LOC  OBJECT CODE     LINE SOURCE TEXT
             VALUE

             00001 ;*****************************************************************************
             00002 ;*****************************************************************************
             00003 ;****     SOLUTIONS CUBED                                                 ****
             00004 ;****     Frank Rossini, Lon Glazner, David Brobst                        ****
             00005 ;*****************************************************************************
             00006 ;*****************************************************************************
             00007 ;
             00008 ;
             00009 ;*****************************************************************************
             00010 ;**** Smart Switch Assembly Code Listing ****
             00011 ;*****************************************************************************
             00012 ;
             00013 ;       The purpose of this code is to develop a pass element controller that
             00014 ;       can be used to supply current to a load.  Two hardware configurations
             00015 ;       allow for shunting the load's low side to ground with a PWM signal,
             00016 ;       or providing a variable DC voltage to a load with a pass element.
             00017 ;       The current handling capabilities of the switch can be determined
             00018 ;       by the chosen pass elements.  The duty cycle adjustment switches may
             00019 ;       be replaced by comparator input signals.  The I/O's left over
             00020 ;       could be used implement serial communication or data storage
             00021 ;       in EEPROM. In this code example the constant 15us delay that provided
             00022 ;       the backbone of the program is also used to implement serial
             00023 ;       communication at 2400 baud.
             00024 ;
             00025 ;*****************************************************************************
             00026 ;
             00027 ;
             00028 ;*****************************************************************************
             00029 ;*****************************************************************************
             00030 ;****   Define registers, constants, processor, and assembler directives ****
             00031 ;*****************************************************************************
             00032 ;*****************************************************************************
             00033 ;
             00034 ;Processor
             00035 ;
             00036         LIST    P=12C508                ;Processor used
             00037 ;
             00038 ;       fuses:
             00039 ;               WDT     - on
             00040 ;               OSC     - internal RC
             00041 ;               MCLR    - internal MCLR
             00042 ;               CP      - code protect on
             00043 ;
             00044 ;Processor defined registers and bits
             00045 ;
             00046         INCLUDE "C:\PIC\HEADERS\P12C508.INC"     ;Microchip include file
             00001         LIST
             00002 ; P12C508.INC  Standard Header File, Version 1.01    Microchip Technology, Inc.
             00103         LIST
             00047 ;
             00048 ;Program defined registers
             00049 ;
  00000006   00050 GPIO            EQU     H'06'           ;Output port register
  00000007   00051 TEMP0           EQU     H'07'           ;Temporary storage register
  00000008   00052 TEMP1           EQU     H'08'           ;Temporary storage register
  00000009   00053 DUTY_CYC        EQU     H'09'           ;Storage register for duty cycle
  0000000A   00054 DATA_REG        EQU     H'0A'           ;Storage register for serial data
  0000000B   00055 FULL_BIT        EQU     H'0B'           ;Holds full bit period delay
  0000000C   00056 QURT_BIT        EQU     H'0C'           ;Holds a quarter bit period delay
```

```

             ;
             ;Program defined bits
             ;
             ;GPIO port bits
  00000000   OUTPUT          EQU     H'00'           ;GPIO, output enable pin
  00000003   INPUT           EQU     H'03'           ;GPIO, input for serial data
  00000004   SW1             EQU     H'04'           ;GPIO, input switch #1
  00000005   SW2             EQU     H'05'           ;GPIO, input switch #2
             ;****************************************************************************
             ;
             ;
             ;****************************************************************************
             ;****************************************************************************
             ;****                           Reset Vector                            ****
             ;****************************************************************************
             ;****************************************************************************
0000                 ORG     H'000'
0000 0025            MOVWF   OSCCAL                  ;Move internal trim value to osccal
0001 0A71            GOTO    MAIN
             ;****************************************************************************
             ;****************************************************************************
             ;BYTE_IN:  Byte_in receives data from a master at 2400 baud, 8N1, LSB first.
             ;       If a valid byte of data is read in then it replaces the current
             ;       duty cycle value in the SMART SWITCH.  In this way serial
             ;       communication can control the duty cycle of the switch.
             ;
             ;       Called From:            MAIN
             ;       Modified Registers:     STATUS, TEMP0, TEMP1, GPIO, DATA_REG
             ;                               DUTY_CYC
             ;       Subroutines Called:     DELAY
             ;       Enabled Interrupts:     NONE
             ;
0002         BYTE_IN
0002 0004            CLRWDT
0003 0666            BTFSC   GPIO,INPUT              ;Test for a start bit
0004 0800            RETLW   H'00'
0005 020C            MOVF    QURT_BIT,W              ;Set up timer for start bit check
0006 0027            MOVWF   TEMP0
0007         delay_loop
0007 095B            CALL    DELAY
0008 0666            BTFSC   GPIO,INPUT              ;Make sure input remains low
0009 0800            RETLW   H'00'
000A 02E7            DECFSZ  TEMP0
000B 0A07            goto    delay_loop
000C 0C08            MOVLW   H'08'                   ;Set up temp to count 8 bits
000D 0028            MOVWF   TEMP1
000E         more_bits
000E 020B            MOVF    FULL_BIT,W              ;Load temp with baud rate value
000F 0027            MOVWF   TEMP0
0010         first_bit
0010 095B            CALL    DELAY                   ;22*18us + 20us = 0.415ms ->2403 baud
0011 02E7            DECFSZ  TEMP0
0012 0A10            goto    first_bit
0013 0766            BTFSS   GPIO,INPUT              ;Test input value
0014 0403            BCF     STATUS,C                ;Input was low so clear carry bit
0015 0666            BTFSC   GPIO,INPUT
0016 0503            BSF     STATUS,C                ;Input was high so set carry bit
0017 032A            RRF     DATA_REG                ;Rotate carry bit into data reg.
0018 0000            NOP
0019 0000            NOP
001A 0000            NOP
001B 0000            NOP                             ;A little delay to get 2400 baud
001C 0000            NOP
001D 0000            NOP
001E 0000            NOP
```

```
001F 0000    00123        NOP
0020 0000    00124        NOP
0021 0000    00125        NOP
0022 02E8    00126        DECFSZ  TEMP1
0023 0A0E    00127        goto    more_bits               ;1/(0.396ms + 20us) = .415 -> 2403baud
0024 020B    00128        MOVF    FULL_BIT,W
0025 0027    00129        MOVWF   TEMP0
0026 02A7    00130        INCF    TEMP0                   ;Add a little more to stop bit
0027         00131 stop_bit
0027 095B    00132        CALL    DELAY                   ;23 * 18us = .414ms -> 2415 baud
0028 02E7    00133        DECFSZ  TEMP0
0029 0A27    00134        goto    stop_bit
002A 0766    00135        BTFSS   GPIO,INPUT              ;Test for valid stop bit
002B 0800    00136        RETLW   H'00'
002C 020A    00137        MOVF    DATA_REG,W              ;Move received data to W
002D 0029    00138        MOVWF   DUTY_CYC
002E 0643    00139        BTFSC   STATUS,Z                ;See if duty cycle is H'00'
002F 02A9    00140        INCF    DUTY_CYC                ;If so increment duty cycle
0030 0289    00141        INCF    DUTY_CYC,W              ;See if duty cycle is H'FF'
0031 0643    00142        BTFSC   STATUS,Z
0032 00E9    00143        DECF    DUTY_CYC                ;If so decrement duty cycle
0033 0800    00144        RETLW   H'00'
0034 0800    00145        RETLW   H'00'
             00146 ;*****************************************************************************
             00147 ;DUTY_CYCLE: This routine monitors the input pins for logic low levels and
             00148 ;       increments or decrements the current duty cycle depending on which
             00149 ;       switch is pressed.  Afterwards the 15us delay routine will
             00150 ;       automatically adjust the duty cycle.  The duty cycle is required to
             00151 ;       be from H'01' to H'FE'.
             00152 ;
             00153 ;       Called From:            MAIN
             00154 ;       Modified Registers:     DUTY_CYC, STATUS
             00155 ;       Subroutines Called:     DELAY
             00156 ;       Enabled Interrupts:     NONE
             00157 ;
0035         00158 DUTY_CYCLE
0035 0686    00159        BTFSC   GPIO,SW1                ;Test for change of duty cycle
0036 0A45    00160        goto    check_decrement         ;Check next switch
0037 0C16    00161        MOVLW   H'16'                   ;Start switch debounce
0038 0028    00162        MOVWF   TEMP1
0039         00163 debounce_loop_0
0039 0C20    00164        MOVLW   H'20'
003A 0027    00165        MOVWF   TEMP0
003B         00166 debounce_loop_1
003B 0004    00167        CLRWDT
003C 095B    00168        CALL    DELAY                   ;Maintain current PWM output
003D 02E7    00169        DECFSZ  TEMP0                   ;Apply 15ms debounce to switch
003E 0A3B    00170        goto    debounce_loop_1
003F 02E8    00171        DECFSZ  TEMP1
0040 0A39    00172        goto    debounce_loop_0
0041 0686    00173        BTFSC   GPIO,SW1                ;Check switch after debounce
0042 0800    00174        RETLW   H'00'
0043 02A9    00175        INCF    DUTY_CYC                ;If good then increment duty cycle
0044 0A54    00176        goto    check_duty_threshholds
0045         00177 check_decrement
0045 06A6    00178        BTFSC   GPIO,SW2                ;Test for change of duty cycle
0046 0800    00179        RETLW   H'00'
0047 0C16    00180        MOVLW   H'16'                   ;Start switch debounce
0048 0028    00181        MOVWF   TEMP1
0049         00182 debounce_loop_2
0049 0C20    00183        MOVLW   H'20'
004A 0027    00184        MOVWF   TEMP0
004B         00185 debounce_loop_3
004B 0004    00186        CLRWDT
004C 095B    00187        CALL    DELAY                   ;Maintain current PWM output
004D 02E7    00188        DECFSZ  TEMP0                   ;Apply 15ms debounce to switch
```

```
004E 0A4B    00189         goto    debounce_loop_3
004F 02E8    00190         DECFSZ  TEMP1
0050 0A49    00191         goto    debounce_loop_2
0051 06A6    00192         BTFSC   GPIO,SW2                ;Check switch after debounce
0052 0800    00193         RETLW   H'00'
0053 00E9    00194         DECF    DUTY_CYC                ;If good then decrement switch
0054         00195 check_duty_threshholds
0054 0209    00196         MOVF    DUTY_CYC,W
0055 0643    00197         BTFSC   STATUS,Z                ;See if duty cycle is H'00'
0056 02A9    00198         INCF    DUTY_CYC                ;If so increment duty cycle
0057 0289    00199         INCF    DUTY_CYC,W              ;See if duty cycle is H'FF'
0058 0643    00200         BTFSC   STATUS,Z
0059 00E9    00201         DECF    DUTY_CYC                ;If so decrement duty cycle
005A 0800    00202         RETLW   H'00'
             00203 ;****************************************************************************
             00204 ;DELAY:  A standard 15us delay.  Instructions + loop + call and return
             00205 ;       equal 15us.  If timer zero rolls over then the output
             00206 ;       (GPIO,OUTPUT) is toggled.  The value in DUTY_CYC is the number
             00207 ;       of 64us periods that GPIO,OUTPUT stays low.  The complement
             00208 ;       of DUTY_CYC is the number of 64us periods that GPIO,OUTPUT stays
             00209 ;       high.  This routine generally maintains the duty cycle of the
             00210 ;       pass elements PWM.
             00211 ;
             00212 ;       Called From:            MAIN
             00213 ;       Modified Registers:     TMR0, DUTY_CYC, GPIO, STATUS
             00214 ;       Subroutines Called:     NONE
             00215 ;       Enabled Interrupts:     NONE
             00216 ;
005B 0201    00217 DELAY   MOVF    TMR0,W                  ;Test for TMR0 rollover
005C 0743    00218         BTFSS   STATUS,Z
005D 0A69    00219         goto    hold                    ;If no rollover don't change output
005E 0606    00220         BTFSC   GPIO,OUTPUT
005F 0A65    00221         goto    t_clr
0060 0506    00222         BSF     GPIO,OUTPUT             ;Set output
0061 0249    00223         COMF    DUTY_CYC,W              ;Complement duty cycle to W
0062 0021    00224         MOVWF   TMR0                    ;Move value to TMR0
0063 0000    00225         NOP                             ;Timing no-op
0064 0A70    00226         goto    done                    ;Get out of routine
0065 0406    00227 t_clr   BCF     GPIO,OUTPUT             ;Clear output
0066 0209    00228         MOVF    DUTY_CYC,W              ;Move duty cycle reg to W
0067 0021    00229         MOVWF   TMR0                    ;Move value to TMR0
0068 0A70    00230         goto    done                    ;Get out of routine
0069 0000    00231 hold    NOP
006A 0000    00232         NOP                             ;If TMR0 doesn't roll over
006B 0000    00233         NOP                             ;then count out a standard delay
006C 0000    00234         NOP
006D 0000    00235         NOP
006E 0000    00236         NOP
006F 0000    00237         NOP
0070 0800    00238 done    RETLW   H'00'
             00239 ;****************************************************************************
             00240 ;****                           Main Program                             ****
             00241 ;****************************************************************************
0071         00242 MAIN
             00243 ;
0071         00244 OPTION_SETUP
0071 0CC5    00245         MOVLW   H'C5'                   ;1100 0101
0072 0002    00246         OPTION                          ;Pull-up disabled, TMR0 1:64
0073         00247 CLEAR_REGISTERS
0073 0067    00248         CLRF    TEMP0                   ;Clear first RAM location for use
0074 0C18    00249         MOVLW   H'18'                   ;Number of registers to clear
0075 0027    00250         MOVWF   TEMP0
0076 0C08    00251         MOVLW   H'08'                   ;Start of RAM clearing
0077 0024    00252         MOVWF   FSR
0078         00253 clear_loop
0078 0060    00254         CLRF    INDF                    ;Clear register pointed to
```

```
0079 02A4    00255         INCF    FSR,F                   ;Go to next RAM location to clear
007A 02E7    00256         DECFSZ  TEMP0,F                 ;Check to see if all clearing done
007B 0A78    00257         goto    clear_loop
007C         00258 PORT_SETUP
007C 0C3E    00259         MOVLW   H'3E'                   ;0011 1110
007D 0026    00260         MOVWF   GPIO                    ;Set output low
007E 0000    00261         NOP
007F 0C3E    00262         MOVLW   H'3E'                   ;0011 1110
0080 0006    00263         TRIS    GPIO                    ;Set GP0 direction as an output
0081         00264 REGISTER_SETUP
0081 0C01    00265         MOVLW   H'01'                   ;Start duty cycle at zero
0082 0029    00266         MOVWF   DUTY_CYC                ;(1 is as close to zero as possible)
0083 0C16    00267         MOVLW   H'16'                   ;Start duty cycle at zero
0084 002B    00268         MOVWF   FULL_BIT                ;Initialize serial communication
0085 0C06    00269         MOVLW   H'06'                   ;for 2400 baud
0086 002C    00270         MOVWF   QURT_BIT
0087 0061    00271         CLRF    TMR0
             00272 ;****************************************************************************
             00273 ;****************************************************************************
0088         00274 MAIN_LOOP
0088 0004    00275         CLRWDT
0089 095B    00276         CALL    DELAY                   ;Maintain PWM
008A 0935    00277         CALL    DUTY_CYCLE              ;Test for manual adjustment
008B 0902    00278         CALL    BYTE_IN                 ;Test for serial data adjustment
008C 0A88    00279         GOTO    MAIN_LOOP               ;Do it all again
             00280 ;****************************************************************************
             00281 ;
             00282 ;End of code indicator
             00283 ;
             00284         END

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

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXX--- ---------------- ---------------- ----------------

All other memory blocks unused.

Program Memory Words Used:   141
Program Memory Words Free:   370


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :    16 reported,     0 suppressed
```

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office

Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602-786-7200 Fax: 602-786-7277
Technical Support: 602 786-7627
Web: http://www.microchip.com

Atlanta

Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago

Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton

Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 714-263-1888 Fax: 714-263-1338

New York

Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516-273-5305 Fax: 516-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong

Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India

Microchip Technology Inc.
India Liaison Office
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-4036 Fax: 91-80-559-9840

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai

Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hong Qiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore

Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#07-02 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan, R.O.C

Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2-717-7175 Fax: 886-2-545-0139

EUROPE

United Kingdom

Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44-1628-851077 Fax: 44-1628-850259

France

Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
2 Rue du Buisson aux Fraises
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Müchen, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa 222 Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

8/29/97

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.