Electromechanical Switch Replacement

Programmable Lights

Author: *Kirill Yelizarov V.*
Moscow Power Engineering Institute
Moscow, Russia
email: tihonov@srv-vmss.mpei.ac.ru

APPLICATION OPERATION

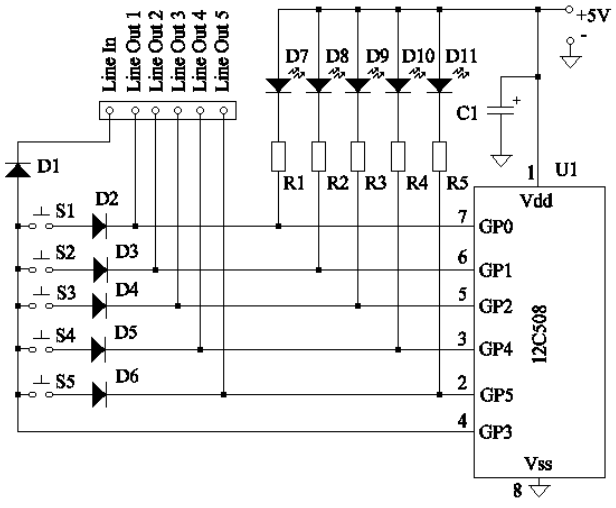
The electronic switch discussed in this application note may be used to operate five different appliances. It is based on the PIC12C508 and requires few external components. The program has a built in interpreter with loops which can output data to PICmicro pins. The language used in the interpreter has seven commands. The switch has five buttons to operate five outputs, one line-in terminal, and five line-out terminals.

With a few external components added, you can make Christmas lights for a Christmas tree. Attach five horns to your car, and they will play your favorite melodies. It is popular to animate car brake lights now. If connected to a timer via 'Line In', this switch can manipulate five appliances. Switches can be connected together through their "Line Out' and 'Line In' terminals. This will increase the number of connected appliances.

The switch itself has five buttons. Test key will start a test program called Test or program #0. Decrease Key or Increase Key will change an internal delay for lights change. Program Key will skip the program flow to the next program. If the last program line reached the first is fetched. Reset key will restore default settings (an internal delay for lights change) and start program #1.

I used a prototype board and handwired the circuit together. There are five buttons (S1-S5), a capacitor (C1), five resistors (R1 - R5), six diodes (D1-D6) and five red LEDs (D7-D11) on the board.

FIGURE 1: SCHEMATIC DIAGRAM



Note: Each Line In and Line Out terminal must have its own grounding terminal.

THE LIGHTS PARTS LIST

Capacitors:

C1 - 47 µf electrolytic

Diodes:

D1-D6 - Any type diodes

D7-D11 - Any type red light emitting diodes (LEDs may be replaced with Opto-relays or opto-triacs See Programmable Timer with Time Correction on how to connect to outlet. Special care should be taken when a transformerless power supply is used (see TB008) because any damage to grounding circuit will destroy ICs if two or more Programmable Lights are connected together.)

Resistors:

R1-R5 – depends on the type of LED

V_O = PIC12C508 output low voltage (0.6V Max)

V_{LED} = Input LED voltage (may vary from 0.8V IR to 2.0V green LED) (1.5 V red LED)

I_{LED} = LED current (10 mA Typical)

$$R1 = \frac{5 - V_{LED} - V_O}{I_{LED}} = 290\Omega$$

Miscellaneous

S1-S5 - normally open pushbutton switches

U1 - PIC12C508 programmed with Lights code

SOFTWARE

The program outputs the value of the Light command every 16 ms.

How the Keyboard Works.

First, all outputs (GP5, GP4, GP2, GP1, GP0) are raised high. Then, the first output GP0 (connected to button S1) is driven low (for 5 µs). On the third microsecond S1 is sampled. If the button is pressed, then GP3 is low. Each button has two key flags. Buttons are tested every 16 ms. If button is not pressed, then the flag is cleared. When the button is pressed and the flag is cleared, it is raised and the program continues to output lights. If the flag is raised already, then a required function is activated, and the second flag is raised else it is assumed to be a key debounce and flags are cleared. All buttons are tested sequentially. A low signal on the 'Line In' terminal will animate button S1 depression (this may be changed in code). To prevent reading buttons on polling of remote Programmable Lights connected to 'Line In', an additional check is made when all pins are high (this is needed if two appliances are absolutely synchronized).

PROGRAMMABLE LIGHTS MACRO LANGUAGE

Macro language used in this appliance has 7 commands. The are summarized in Table 1.

TABLE 1: TABLE OF COMMANDS

Command	Description
Light	Set new light
MarkLabel	Mark a label
ReturnToLabel	Return to label
SetRepeatValue	Set repeat value
JumpToProgram	Jump to program
StartOfProgram	Start of program
RestartProgram	Return to start of program

Light

This command is used to set microcontroller pins high or low. It is advisable to write this command like this:

```
Light b'00001'
```

Binary data send to PICmicro pins is (from right to left): GP5, GP4, GP2, GP1, GP0, where 0 sets the corresponding pin high, and 1 sets low. The command shown in the example will turn GP5, GP4,GP2 and GP1 high, and GP0 low.

SetRepeatValue MarkLabel RetunToLabel

These commands are used to make a loop with counter in the program. SetRepeateValue should precede MarkLabel command and is used to set the number of times the program will when ReturnToLabel command found roll over to MarkLabel. There is no default value, so each MarkLabel command must have a SetRepeatValue command. There may be 31 labels in the program. When a ReturnToLabel command is found the interpreter decreases repeat value and if it is not zero than it walks backward (this is significant, because in this case there may be more than 31 loops in the program) until a MarkLabel command is found. The repeat value should be in the range from 1 to 31 (If SetRepeatValue is set to 0 then commands inside loop will be output 256 times). An example shows how to write a loop correctly:

```
SetRepeatValue 10
MarkLabel 2
Light b'00001'
ReturnToLabel 2
```

This code example sets repeat value to 10 times and uses label #2. This code will set pin GP0 low ten times longer than the basic delay. Label numbers make no sense now but with some improvement to the code some label numbers may have unique counters and loops may be nested.

Start of Program RestartProgram

Part of program may be placed into an endless loop. To switch between endless loops this appliance has three buttons: Reset, Test and Program. There may be 31 loops. An example shows how it works:

```
StartOfProgram 5
.
.
.
StartOfProgram 1
Light b’00001’
RestartProgram 5
RestartProgram 1
```

If the current program number is #1 then command RestartProgram 5 is ignored. When RestartProgram 1 is fetched the interpreter walks forward until StartOf-Program 1 is found. Programs may be nested or crossed.

This commands may be useful to skip some code when several programs are nested or crossed. This is shown in the example:

```
StartOfProgram 1
Light b’00001’
RestartProgram 1
Light b’00100’
StartOfProgram 1
Light b’00010’
RestartProgram 1
```

In this case if the current program number is 1 then the interpreter will output 00001 and 00010 skipping 00100. This is because when RestartProgram is found the interpreter walks forward until the desired StartOf-Program is found.

Jump to Program

This command will jump to the desired StartOfProgram command. An example shows how it works:

```
.
.
.
StartOfProgram 1
Light b’00001’
RestartProgram 1
.
.
.
JumpToProgram 1
.
.
.
```

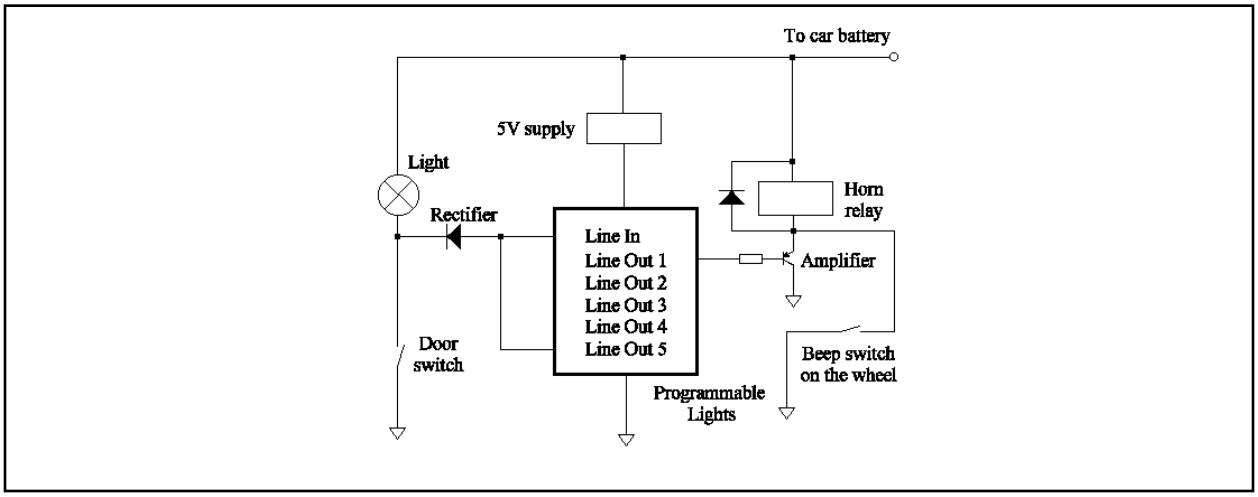
When JumpToProgram 1 is fetched the interpreter walks forward until StartOfProgram 1 is found. It is not the same as with RestartProgram since current program number may be not #1 when JumpToProgram 1 is fetched.

With the help of this command a simple alarm system bay be done. System block diagram is shown in figure 2. Look at the code:

```
define      MAX_PROGRAM=1
      StartOfProgram Test
      Lights b’00000’
      RestartProgram Test

      StartOfProgram 1
      SetRepeatValue 10
      MarkLabel 1
      Lights b’10001’
      Lights b’10000’
      ReturnToLabel 1
      JumpToProgram Test
```

FIGURE 2: CAR ALARM SYSTEM BLOCK DIAGRAM



Line in is connected to a car door open switch (normally closed contacts with one terminal connected to the ground) through a rectifier. GP0 is connected through an amplifier and a relay to a horn. GP5 is connected to line in. Other outputs maybe connected to parking lights and ignition immobilizer. Once the door is open this will change the current program number to #1 and the horn will beep 10 times. Line-in will be blocked with GP5 low signal to prevent several restarting of program number 1. Then the program will jump to test program and line in will be connected again. If the door is still opened there will be another beep. To switch the alarm off you need to push a Test button. If the thief tries to switch the car battery off and then on again the program will automatically start from program #1 and an alarm will occur.

QUICK CODE IDEAS

The code has a macro to test pressed buttons and a small function to read tables.

A macro to test pressed buttons is useful when the microcontroller has no interrupt flag on pin change and the only way to add a keyboard is to poll microcontrollers pins. This macro needs two bytes to store key flags (for 8 buttons or less).

I know several table read functions. Some are simple the rest are too long. This function is ideal for PIC12C508/509 since a call may be done only for the first 256 bytes of memory page code.

MICROCHIP TOOLS USED

Development Tools:

This code was written and debugged with MPLAB™ for Windows®/16 Version 3.22.02.

Assembler/Compiler version:

MPASM v01.50

APPENDIX A: SOURCE CODE

```
;Programmable Lights
;Author: Kirill Yelizarov

         LIST                    P=PIC12C508, R=DEC
         INCLUDE                 <p12c508.inc>

         __CONFIG _IntRC_OSC & _WDT_OFF & _CP_OFF & _MCLRE_OFF

;                      ------------ D A T A ------------

MPC                         equ      0x07           ;Macro program counter
KeyFlags                    equ      0x08           ;Keyboard flags
Keys                        equ      0x09           ;Keyboard keys
Command                     equ      0x0a           ;Current command
Value                       equ      0x0b           ;Local data
Program                     equ      0x0c           ;Current program number
TimeDelay                   equ      0x0d           ;Time delay changed with Inc or Dec buttons
TimeCount                   equ      0x0e           ;Current time count
RepeatCount                 equ      0x0f           ;Counter for a loop

;----- KeyBoard bits -----

ResetKey                    equ      5              ;Switch to program #1
TestKey                     equ      4              ;Switch to program #0 (called Test)
LineIn                      equ      3              ;Line in for all buttons
DecKey                      equ      2              ;Decrease delay between lights flow
IncKey                      equ      1              ;Increase delay between lights flow
ProgKey                     equ      0              ;Switch to next program

;                ---------- Lights macro language -------------
#define  Light              retlw    0x00+          ;Set new light
#define  MarkLabel          retlw    0x20+          ;Mark a label #(1...31)
#define  ReturnToLabel      retlw    0x40+          ;Return to label #(1...31)
#define  SetRepeatValue     retlw    0x80+          ;Set repeat value (1...31)
#define  JumpToProgram      retlw    0xa0+          ;Jump to program #(1...31)
#define  StartOfProgram     retlw    0xc0+          ;Start of program #(1...31)
#define  RestartProgram     retlw    0xe0+          ;Return to start of program #(1...31)

Mark                        equ      0x20
ReturnTo equ     0x40
RepeatValueequ              0x80
JumpTo                      equ      0xa0
StartOf                     equ      0xc0
Restart                     equ      0xe0

Button                      equ      0x00
Line                        equ      0x01

MAX_PROGRAMequ              12
Test                        equ      0              ;Test program

;                      ------------ M A C R O ----------
;This macro is used to search pressed button
;total button de bounce test is 2*16 ms = 32 ms with one extra test on the 16th ms

TestButtonmacro             source,line,butt,key,flag,out
                            local    reset
                            local    test

;source - type of signal (Button - synchronized by the lights, Line - not syncronized)
;line - input test GPIO pin
;butt - button output GPIO pin
;key - depression memory location
;flag - key flag memory location
;out - label to return from macro
```



```
            bcf         GPIO,butt           ;drive butt pin low
            nop                             ;wait
            btfsc       GPIO,line           ;look what happens on line pin
            goto        reset               ;if high go and reset all for this button
            bsf         GPIO,butt           ;drive butt pin high

            IF          source==Button      ;extra test for syncronized buttons
            nop                             ;wait
            btfss       GPIO,line           ;test pin for high value
            goto        reset               ;if pin is still low then it's a line in signal
            ENDIF

            IF          source==Line
            nop                             ;wait
            btfsc       GPIO,line           ;test line in if signal is still low
            goto        reset               ;else a polling signal was read
            ENDIF

            btfsc       key,butt            ;test key depression for butt pin
            goto        test                ;yes there was a depression on last test, go to test
            bsf         key,butt            ;no then set key depression flag
            goto        out                 ;and wait for the next test
reset:
            bsf         GPIO,butt           ;drive butt pin high
            bcf         key,butt            ;clear depression key
            bcf         flag,butt           ;clear key flag
            goto        out                 ;test next
test:
            btfsc       flag,butt           ;test key flag
            goto        out                 ;if set then test next (to prevent multiple run of the
                                            ; same function while the button is pressed)
            bsf         flag,butt           ;set key flag
                                            ;function code will be merged here
            endm


;                   ------------ C O D E ------------

      org       0

      goto      Start                             ;Skip subroutine and table

;This multifunction subroutine can fetch next, or previous, or current command from the table

DecPC                                             ;Get previous command
      decf      MPC,F
      movlw     low Table
      xorwf     MPC,W
      btfss     STATUS,Z
      goto      ReadPC
      movlw     low (Start-1)
      movwf     MPC
      goto      ReadPC
IncPC                 ;Get next command
      incf      MPC,F
      movlw     low Start
      xorwf     MPC,W
      btfss     STATUS,Z
      goto      ReadPC
      movlw     low (Table+1)
      movwf     MPC
ReadPC          ;Read current command
      movf      MPC,W
      call      Table
      movwf     Value
```

```
      retlw      0

;                -------------------- P R O G R A M ---------------------
Table                                     ;Start of program table
      movwf      PCL

;*********************************************
;*            Christmas Lights                *
;*                  1997                      *
;*        Author: Kirill Yelizarov            *
;*********************************************

          StartOfProgram Test
          Light      b'11111'
          RestartProgram Test

          StartOfProgram 1

          Light      b'11111'
          Light      b'11111'
          Light      b'11111'
          Light      b'01111'
          Light      b'01111'
          Light      b'01111'
          Light      b'00111'
          Light      b'00111'
          Light      b'00111'
          Light      b'00011'
          Light      b'00011'
          Light      b'00011'
          Light      b'00001'
          Light      b'00001'
          Light      b'00001'

          SetRepeatValue 10
          MarkLabel 1
          StartOfProgram 2
          Light      b'10000'
          Light      b'01000'
          Light      b'00100'
          Light      b'00010'
          Light      b'00001'
          RestartProgram 2
          ReturnToLabel 1

          SetRepeatValue 10
          MarkLabel 2
          StartOfProgram 3
          Light      b'10001'
          Light      b'11000'
          Light      b'01100'
          Light      b'00110'
          Light      b'00011'
          RestartProgram 3
          ReturnToLabel 2

          SetRepeatValue 10
          MarkLabel 3
          StartOfProgram 4
          Light      b'10011'
          Light      b'11001'
          Light      b'11100'
          Light      b'01110'
          Light      b'00111'
          RestartProgram 4
          ReturnToLabel 3
```

```
SetRepeatValue 10
MarkLabel 4
StartOfProgram 5
Light       b'10111'
Light       b'11011'
Light       b'11101'
Light       b'11110'
Light       b'01111'
RestartProgram 5
ReturnToLabel 4

SetRepeatValue 10
MarkLabel 5
Light       b'00111'
Light       b'10011'
Light       b'11001'
Light       b'11100'
Light       b'01110'
ReturnToLabel 5

SetRepeatValue 10
MarkLabel 6
Light       b'00110'
Light       b'00011'
Light       b'10001'
Light       b'11000'
Light       b'01100'
ReturnToLabel 6

SetRepeatValue 10
MarkLabel 7
Light       b'00100'
Light       b'00010'
Light       b'00001'
Light       b'10000'
Light       b'01000'
ReturnToLabel 7

Light       b'00100'
Light       b'00010'
Light       b'00001'

StartOfProgram 6
Light       b'10000'
Light       b'10000'
Light       b'10000'
Light       b'11000'
Light       b'11000'
Light       b'11000'
Light       b'11100'
Light       b'11100'
Light       b'11100'
Light       b'11110'
Light       b'11110'
Light       b'11110'
Light       b'11111'
Light       b'11111'
Light       b'11111'
Light       b'11110'
Light       b'11110'
Light       b'11110'
Light       b'11100'
Light       b'11100'
Light       b'11100'
Light       b'11000'
```

```
Light       b'11000'
Light       b'11000'
Light       b'10000'
Light       b'10000'
Light       b'10000'
RestartProgram 6

SetRepeatValue 10
MarkLabel 8
StartOfProgram 7
Light       b'00001'
Light       b'00010'
Light       b'00100'
Light       b'01000'
Light       b'10000'
RestartProgram 7
ReturnToLabel 8

SetRepeatValue 10
MarkLabel 9
StartOfProgram 8
Light       b'10001'
Light       b'00011'
Light       b'00110'
Light       b'01100'
Light       b'11000'
RestartProgram 8
ReturnToLabel 9

SetRepeatValue 10
MarkLabel 10
StartOfProgram 9
Light       b'11001'
Light       b'10011'
Light       b'00111'
Light       b'01110'
Light       b'11100'
RestartProgram 9
ReturnToLabel 10

SetRepeatValue 10
MarkLabel 11
StartOfProgram 10
Light       b'11101'
Light       b'11011'
Light       b'10111'
Light       b'01111'
Light       b'11110'
RestartProgram 10
ReturnToLabel 11

SetRepeatValue 10
MarkLabel 12
Light       b'11100'
Light       b'11001'
Light       b'10011'
Light       b'00111'
Light       b'01110'
ReturnToLabel 12

SetRepeatValue 10
MarkLabel 13
Light       b'01100'
Light       b'11000'
Light       b'10001'
Light       b'00011'
```

```
        Light      b'00110'
        ReturnToLabel 13

        SetRepeatValue 10
        MarkLabel 14
        Light      b'00100'
        Light      b'01000'
        Light      b'10000'
        Light      b'00001'
        Light      b'00010'
        ReturnToLabel 14

        Light      b'00100'
        Light      b'01000'
        Light      b'10000'

        StartOfProgram 11
        Light      b'00001'
        Light      b'00001'
        Light      b'00001'
        Light      b'00011'
        Light      b'00011'
        Light      b'00011'
        Light      b'00111'
        Light      b'00111'
        Light      b'00111'
        Light      b'01111'
        Light      b'01111'
        Light      b'01111'
        RestartProgram 1
        Light      b'11111'
        Light      b'11111'
        Light      b'11111'
        Light      b'01111'
        Light      b'01111'
        Light      b'01111'
        Light      b'00111'
        Light      b'00111'
        Light      b'00111'
        Light      b'00011'
        Light      b'00011'
        Light      b'00011'
        Light      b'00001'
        Light      b'00001'
        Light      b'00001'
        RestartProgram 11

        StartOfProgram 12
        Light      b'10001'
        Light      b'00000'
        Light      b'01010'
        Light      b'00000'
        Light      b'00100'
        Light      b'00000'
        Light      b'01010'
        Light      b'00000'
        RestartProgram 12

;               ------------ M A I N -------------
Start:

        IF         Start>0x100
        ERROR      "Lights Message: Program Table too large."
        ENDIF

        clrf       Keys           ;Reset key depression
```

```
         clrf       KeyFlags       ;Reset keyboard flags
         clrf       TMR0           ;clear TMR0
         movlw      b'10000101'    ;Enable weak pull-up on GP3 and set prescaler to 1:64
;        movlw      b'10000000'    ;*****For MPLAB debug
         option

         clrf       GPIO
         comf       GPIO,F         ;Turn OFF the lights
         movlw      b'00001000'    ;Set GP3 (LineIn) as input and the rest are outputs
         tris       GPIO

         movlw      low (Start-1)  ;Set program counter to the end of Table
         movwf      MPC

         movlw      0x10
;        movlw      0x01           ;*****For MPLAB debug
         movwf      TimeCount      ;Each time TMR0 overflows this value will decrement
         movwf      TimeDelay      ;Set time dalay between two Light commands to 16*16 ms
                                   ;= 256 ms may be changed with Inc and Dec keys
         movlw      0x01
         movwf      Program        ;Set program #1

SetProgram:
         call       IncPC          ;Get next command
         movf       Value,W
         andlw      b'11100000'
         xorlw      StartOf        ;If it's a StartOfProgram then continue
         btfss      STATUS,Z       ;else go and get another command
         goto       SetProgram
         movf       Value,W
         andlw      b'00011111'
         xorwf      Program,W      ;Be shure the right program found
         btfss      STATUS,Z       ;else go and get another command
         goto       SetProgram

MainLoop:
         call       IncPC          ;Get next command
         movf       Value,W
         andlw      b'11100000'    ;Test Light Command
         btfsc      STATUS,Z
         goto       SetLights
         movwf      Command        ;Save Command

         xorlw      ReturnTo       ;Test ReturnToLabel command
         btfss   STATUS,Z
         goto       TestRestartProgram    ;else analyze next command
         movf       Value,W
         andlw      b'00011111'
         movwf      Command        ;Save label in Command because it's contents are no longer
needed
         decfsz     RepeatCount,F  ;decrease loop counter set by
         goto       FindLabel
         goto       MainLoop
FindLabel:
         call       DecPC          ;Search backward for a desired label
         movf       Value,W
         andlw      b'1110000'
         xorlw      Mark
         btfss      STATUS,Z
         goto       FindLabel
         movlw      b'00011111'
         andwf      Value,W
         xorwf      Command,W
         btfss      STATUS,Z
         goto       FindLabel
         goto       MainLoop
```

```
TestRestartProgram:  ;Look if it's a RestartProgram command
         movf       Command,W
         xorlw      Restart
         btfss   STATUS,Z
         goto       TestRepeatValue
         movlw      b'00011111'
         andwf      Value,W
         xorwf      Program,W
         btfss      STATUS,Z
         goto       MainLoop
         goto       SetProgram

TestRepeatValue:
         movf       Command,W
         xorlw      RepeatValue
         btfss   STATUS,Z
         goto       TestJumpTo
         movlw      b'00011111'
         andwf      Value,W
         movwf      RepeatCount
         goto       MainLoop

TestJumpTo:
         movf       Command,W
         xorlw      JumpTo
         btfss   STATUS,Z
         goto       MainLoop
         movlw      b'00011111'
         andwf      Value,W
         movwf      Program
         goto       SetProgram

SetLights:
         movlw      b'00110111'
         btfsc      Value,4
         bsf        Value,5
         bcf        Value,4
         btfsc      Value,3
         bsf        Value,4
         andwf      Value,F        ;cut everything but the lights

Wait1:
         btfss      TMR0,7         ;wait till seventh bit rise,  comment it when in MPLAB debug
         goto       Wait1          ;***** Comment it when in MPLAB debug

         movlw      b'11111111'
         movwf      GPIO

         TestButton Button,LineIn,ProgKey,Keys,KeyFlags,TestInc
NextProgram:
         incf       Program,F
         movlw      MAX_PROGRAM+1
         xorwf      Program,W
         btfss      STATUS,Z
         goto       SetProgram
         movlw      0x01
         movwf      Program
         goto       SetProgram

TestInc:
         TestButton Button,LineIn,IncKey,Keys,KeyFlags,TestDec
         movlw      0x01
         movwf      TimeCount      ;Set TimeCount to one to make delay change fast
         incfsz     TimeDelay,F
         goto       TestDec
```

```
         movlw      0xff
         movwf      TimeDelay

TestDec:
         TestButton Button,LineIn,DecKey,Keys,KeyFlags,TestLine
         movlw      0x01
         movwf      TimeCount      ;Set TimeCount to one to make delay change fast
         decfsz     TimeDelay,F
         goto       TestLine
         movlw      0x01
         movwf      TimeDelay

TestLine:
         TestButton Line,LineIn,LineIn,Keys,KeyFlags,TestTest
         goto       NextProgram

TestTest:
         TestButton Button,LineIn,TestKey,Keys,KeyFlags,TestReset
         clrf       Program
         goto       SetProgram

TestReset:
         TestButton Button,LineIn,ResetKey,Keys,KeyFlags,RestoreLights
         goto       Start

RestoreLights:
         comf       Value,W
         movwf      GPIO           ;output Lights to GPIO
Wait0:
         btfsc      TMR0,7         ;wait till the seventh bit reset, *****Comment it when in
MPLAB debug
         goto       Wait0          ;*****Comment it when in MPLAB debug
         decfsz     TimeCount,F
         goto       Wait1          ;if zero not reached then make another key check
         movf       TimeDelay,W    ;reset TimeCount with TimeDelay value
         movwf      TimeCount
         goto       MainLoop

         org        0x1ff
         movlw      b'01110000'    ;set OSCCAL
         end
```

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

HongKong
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