Electromechanical Switch Replacement

Smart Switch for Car Windscreen Wiper Control

Author: *Marc Hoffknecht*
Aachen, Germany
email: hofknecht@online.de

PIC12C508 replaces potentiometer and multi-stage switch and increases user-friendliness.

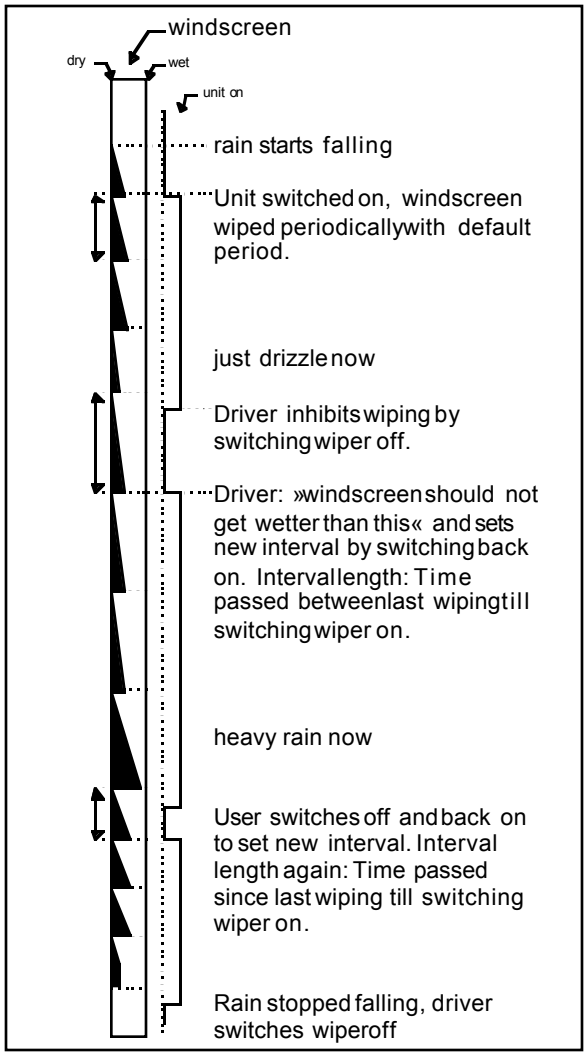
APPLICATION OPERATION

The usual wiper control in a car has two problems, both of which this application can solve. First, it uses too many parts, usually an on/off switch and either a potentiometer to adjust the wiping interval or a multi-stage switch. Second, it is not very user-friendly: You either have a limited number of interval periods or, if the wiper is controlled via a potentiometer, you have to adjust the interval period, watch the windscreen if the interval is sufficient (takes at least one or two times wiping), re-adjust the period and so on.

This application uses a single switch and a PIC12C508 to adjust the wiper interval settings. The main point is that the driver decides when the windscreen is too 'wet'. It is easiest to understand the operation using the attached graphics.

Upon switching the unit on, the windscreen is wiped periodically with a default interval. By switching the unit off, the driver inhibits wiping causing the windscreen to get wetter and wetter. As soon as the driver decides – the windscreen should not get wetter than this– he/she switches it back on. Doing so, the driver sets the new interval according to the time passed between the last clearing of the windscreen and switching the unit back on. This way the driver can either lengthen or shorten the interval to exactly what he wishes - the wipers will not go too fast (by the way, often it is the case in traffic jams, there's just no suitable setting!) and it won't go too slow.

OPERATION FLOWCHART

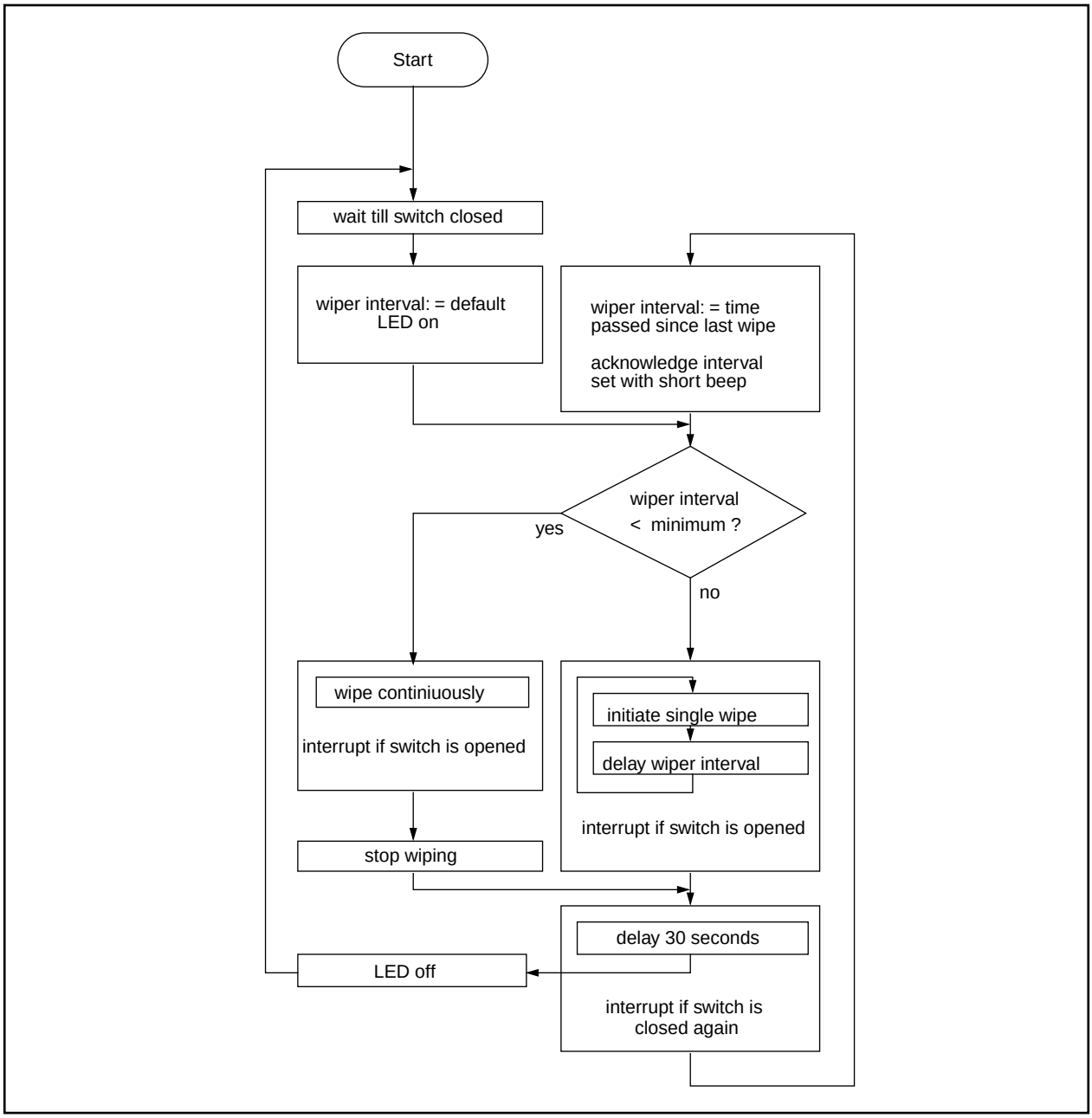


The software shows how to:

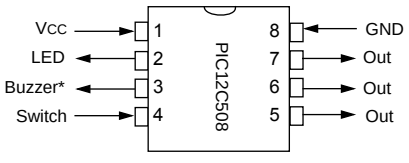
- Implement an accurate timer with a period longer than possible with the internal timer (very efficient code! a subroutine which just has to be called every now and then).
- Generate software interrupts.
- Debounce switches in an interrupt routine and protect the software against noise on the switch input.
- Return boolean values (again very efficiently!).

Electromechanical Switch Replacement

FLOW CHART



GRAPHICAL HARDWARE REPRESENTATION



*Open drain in order to make it shareable with other units.

MICROCHIP TOOLS USED

Assembler/Compiler Version:

MPASM V1.4

APPENDIX A: SOURCE CODE

```
;*************************************************************************
;* Project: Smart Wiper Switch                                           *
;*************************************************************************

                            processor 12c508
                            radix dec
                            include "p12c508.inc"

#define                     __12C508
                            __config _WDT_OFF & _IntRC_OSC & _MCLRE_OFF & _CP_ON

#define zero     STATUS, 2
#define carry    STATUS, 0

#define TRUE      0
#define FALSE   -1

                            CBLOCK 0x07        ; start of RAM
                            ENDC

                            MOVWF OSCCAL
                            GOTO Main

                            ;

TMR0overrun      EQU 16384                     ; timer0 overrun every 16.4ms
                                               ; remember to change the option value
#define   ms    1000/TMR0overrun               ; in the main program when changing
#define secs 1000000/TMR0overrun               ; this

;* Hardware *************************************************************

#define Switch              GPIO, 3
#define LED                 GPIO, 5
#define Buzzer              GPIO, 4

#define          LedOn      BCF LED            ; LED output is activ low
#define LedOff              BSF LED

BuzzerOn                    MACRO
                            BCF Buzzer         ; Buzzer is open drain in order
                            MOVLW b'001000'    ; to share it with other units
                            TRIS GPIO
                            ENDM
BuzzerOff        MACRO
                            MOVLW b'011000'
                            TRIS GPIO
                            ENDM
OutputOn         MACRO
                            MOVLW b'111000'    ; output is activ low
                            ANDWF GPIO
                            ENDM
OutputOff                   MACRO
                            MOVLW b'000111'
                            IORWF GPIO
                            ENDM

                                               ;

WiperThreshold              EQU 500*ms
MinimumInterval             EQU   1*secs
DefaultInterval             EQU   2*secs

BeepLength                  EQU 200*ms
```

```
DenoiseTime                 EQU  50*ms
DebounceTime                EQU  50*ms

DisableAfter                EQU  20*secs

;* Macros ********************************************************************

TWSTrue                     MACRO             ; (T)est (W) and (S)kip if (True)
                                              IORLW 0
                                              BTFSS zero
                                              ENDM
TWSFalse                    MACRO
                                              IORLW 0
                                              BTFSC zero
                                              ENDM

#define          SkipIfZero BTFSS zero
#define          DoIfZero   BTFSC zero
#define          RET                          RETLW 0

;* Switch ********************************************************************

                            CBLOCK
                            Denoise
                            Debounce
                            Flags
                            ENDC
#define                     SwitchClosed       Flags, 0

                            ;

HandleSwitch     MACRO
                            BTFSS Switch
                            GOTO HS.closed

HS.opened        MOVLW DebounceTime; switch open now, so
                            MOVWF Debounce     ; reset timer for 'switch closed'
                            BTFSS SwitchClosed
                            GOTO HS.done       ; switch is already denoised
                            DECFSZ Denoise     ; otherwise, wait till switch
                            GOTO HS.done       ; is stable a certain time
                            BCF SwitchClosed
                            GOTO HS.done

HS.closed        MOVLW DenoiseTime; as above
                            MOVWF Denoise
                            BTFSC SwitchClosed
                            GOTO HS.done
                            DECFSZ Debounce
                            GOTO HS.done
                            BSF SwitchClosed
HS.done
                            ENDM

;* Timer *********************************************************************

                            CBLOCK
                            Timer0L
                            Timer0H
                            Timer1L
                            Timer1H
                            OldTMR0
                            ENDC


IncreaseTimer1  MACRO
                            INCFSZ Timer1L
                            GOTO Timer1.done
```

```
                            INCF Timer1H
Timer1.done
                            ENDM


Interrupt           MOVF OldTMR0, W               ; increase Timer on TMR0-overflow
                            SUBWF TMR0, W         ; overflow, if OldTMR0 > TMR0
                            BTFSC carry
                            GOTO Interrupt.done
                            ADDWF OldTMR0
                                                  ; program enters here every 16.4ms
                            HandleSwitch
                            IncreaseTimer1

                            INCFSZ Timer0L
                            RETLW FALSE
                            INCFSZ Timer0H
                            RETLW FALSE
                            RETLW TRUE            ; return TRUE upon hitting zero
                                                  ; in timer0 !
Interrupt.done      ADDWF OldTMR0
                            RETLW FALSE

                            ;

LoadTimer0          MACRO Value
                            MOVLW low(-Value)
                            MOVWF Timer0L
                            MOVLW high(-Value)
                            MOVWF Timer0H
                            ENDM

;* Subroutines *************************************************************

Beep                        LoadTimer0 BeepLength
Beep.loop           BTFSC TMR0, 2                 ; this will generate about 2 kHz
                            BuzzerOn
                            BTFSS TMR0, 2
                            BuzzerOff
                            CALL Interrupt
                            TWSTrue
                            GOTO Beep.loop
                            BuzzerOff
                            RET

                            ;

Delay                       MACRO Value
                            LOCAL Loop
                            LoadTimer0 Value
Loop                        CALL Interrupt
                            TWSTrue
                            GOTO Loop
                            ENDM

;*****************************************************************************

                            CBLOCK
                            IntervalL
                            IntervalH
                            ENDC


Main                     MOVLW b'10010101'; pullups on
                            OPTION            ; -> TMR0overrun every 16.384us

                            BuzzerOff         ; this will also set TRIS correctly
```

```
                            OutputOff

                            MOVLW DebounceTime
                            MOVWF Debounce
                            BCF SwitchClosed

                            ;

Main.loop           LedOff
                            CALL Interrupt
                            BTFSS SwitchClosed
                            GOTO Main.loop

                            MOVLW low(DefaultInterval); Interval:= DefaultInterval
                            MOVWF IntervalL    ;
                            MOVLW high(DefaultInterval);
                            MOVWF IntervalH    :
                            LedOn

CheckInterval       MOVLW high(MinimumInterval); if Interval<MinimumInterval
                            SUBWF IntervalH, W ; then GOTO Wipe.continuous
                            BTFSS carry        ;
                            GOTO Wipe.continuous;
                            BTFSS zero         ;
                            GOTO Wipe.interval ;
                            MOVLW low(MinimumInterval);
                            SUBWF IntervalL, W ;
                            BTFSS carry        ;
                            GOTO Wipe.continuous;

                            ;-------------------------------;

Wipe.interval       OutputOn                    ; initiate single wipe
                            CLRF Timer1L       ; always clear timer1 upon
                            CLRF Timer1H       ; wiping
                            Delay WiperThreshold;
                            OutputOff          ;

Wipe.int.loop       CALL Interrupt

                            BTFSS SwitchClosed
                            GOTO Off?

                            MOVF IntervalH, W  ; if timer1<Interval
                            SUBWF Timer1H, W   ; then GOTO Wipe.int.loop
                            BTFSS carry        ; else GOTO Wipe.interval
                            GOTO Wipe.int.loop ;
                            BTFSS zero         ;
                            GOTO Wipe.interval ;
                            MOVF IntervalL, W  ;
                            SUBWF Timer1L, W   ;
                            BTFSS carry        ;
                            GOTO Wipe.int.loop ;
                            GOTO Wipe.interval ;

                            ;-------------------------------;

Wipe.continuous OutputOn
                            CLRF Timer1L       ; always clear timer1 upon
                            CLRF Timer1H       ; wiping
                            CALL Interrupt
                            BTFSC SwitchClosed
                            GOTO Wipe.continuous
                            OutputOff

                            ;-------------------------------;
```

```
Off?                        LoadTimer0 DisableAfter
Off?.loop        BTFSC SwitchClosed
                            GOTO NewIntervalSet
                            CALL Interrupt
                            TWSTrue
                            GOTO Off?.loop
                            GOTO Main.loop     ; 30 seconds expired

                            ;------------------------------;

NewIntervalSet  MOVF Timer1L, W
                            MOVWF IntervalL
                            MOVF Timer1H, W
                            MOVWF IntervalH

                            CALL Beep

                            GOTO CheckInterval

;*****************************************************************************

                            END
```

NOTES:

NOTES:

NOTES:

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