```
'*****************************************************************
'*  Name    : Inverter.BAS                                       *
'*  Author  : Sougata Das                                        *
'*  Notice  : Copyright (c) 2005 The Andig Technologies          *
'*          : All Rights Reserved                                *
'*  Date    : 30-07-2005                                         *
'*  Version : 1.0                                                *
'*  Notes   : Uses a PIC16F72. Outputs MOSFET DRIVE THROUGH    *
'*          ' HWPM + CD4081 FOR CHANNEL CONTROL.                 *
'*          ' LOOKUP TABLE FOR SINE                              *
'*          ' INTERRUPT 1)MOS-DRIVE,
'*  modified by :isaac  (06/05/2008)                             *
'*****************************************************************
                                '
    DEFINE OSC 20               'USES A 20MHz CRYSTAL
    DEFINE USE_LFSR 1           'Makes some code shorter
'*******************************************************************************
' Define interrupt handler      '
    DEFINE INTHAND myint        '
'Variables for saving state in interrupt handler ******************************
wsave           VAR     BYTE bankA system       ' Saves W
ssave           VAR     BYTE bankA system       ' Saves STATUS
fsave           VAR     WORD bankA system       ' Saves FSR0
'///////////////////////////////////////////////////////////////////////////////

TIM0INT         VAR BIT  bankA system   ' TIMER0 INTERRUPT CLEARED WHEN IT OCCURS

PHASE           VAR BIT  bankA system   ' DRIVE PHASE (DETERMINE WHICH CYCLE)

INTVAL          VAR WORD bankA system   ' PWM UPDATE INTERRUPT (ADJUSTED TO LOCK TO
AC)
INDEX           VAR BYTE bankA system   ' SINE-TABLE INDEX POINTER
PWMVAL          VAR BYTE bankA system   ' PWMVALUE FOR FIRING
TEMP            VAR WORD                ' TEMPORARY VARIABLE

    MOS1        VAR PORTC.0             ' MOSFET CHANNEL-1 GATE DRIVE
    MOS2        VAR PORTC.1             ' MOSFET CHANNEL-2 GATE DRIVE
    PULSE       VAR PORTC.2             ' MODULATING PWM OUTPUT (32KHz,
SINE-TABLED)

'******PORT DATA DIRECTIONS ****************************************************
    LATC     = 0                    ' CLEAR THE PORTC DATA LATCHES
    TRISC    = %10000000            ' SET PORTC DIRECTIONS
    PORTC    = 0                    ' MAKE-SURE THAT THE MOSFETS ARE TURNED-OFF

'*******************************************************************************

GOTO STARTUP ' SKIP AROUND THE INTERRUPT HANDLER

' ***Assembly language INTERRUPT handler*************************************
Asm
myint

; Save the state of critical registers
                movwf   wsave                   ; Save W
                swapf   STATUS, W               ; Swap STATUS to W (swap avoids
changing STATUS)
                clrf    STATUS                  ; Clear STATUS
                movwf   ssave                   ; Save swapped STATUS

; Save the FSR value because it gets changed below
                movf    Low FSR0, W             ; Move FSR0 lowbyte to W
```

```
                movwf   fsave                   ; Save FSR0 lowbyte
                movf    High FSR0, W       ; Move FSR0 highbyte to W
                movwf   fsave+1                 ; Save FSR0 highbyte
; CHECK FOR TIMER0 INTERRUPT EVERY 312.5uS
TIMEKEEP

        btfss   INTCON, TMR0IF      ; TEST IF T0 OVERFLOWED
        GoTo    FINISHED            ; SKIP IF SET
        MOVLW   0xFC                ; MOVE '11111100
        ANDWF   PORTC,1             ; ADD W WITH PORTC AND PLACE IN PORTC
        movff   INTVAL+1,TMR0H      ; MOVE THE HIGH VALUE TO W REG
        movff   INTVAL,  TMR0L      ; MOVE THE LOW VALUE TO W REG
        bcf     TIM0INT             ; CLEAR THE TIM0INT SOFTWARE FLAG
        bcf     INTCON, TMR0IF      ; CLEAR THE TIMER0 INT HARDWARE FLAG
        incf    INDEX, 1            ; INCREMENT INDEX
        movf    INDEX, W            ;
        sublw   .32                 ; Substract INDEX FROM 32
        btfss   STATUS,Z
        GoTo    FINISHED            ; SKIP PHASE CHANGE (INDEX RESET)
        clrf    INDEX               ; CLEAR THE INDEX
        btg     PHASE               ; TOGGLE THE PHASE
; Restore FSR, PCLATH, STATUS and W registers
FINISHED
                movf    fsave, W                    ; retrieve FSR0 lowbyte value
                movwf   Low FSR0                    ; Restore it to FSR0 lowbyte
                movf    fsave+1, W                  ; retrieve FSR0 highbyte value
                movwf   High FSR0                   ; Restore it to FSR0 highbyte
                swapf   ssave, W                    ; Retrieve the swapped STATUS
value (swap to avoid changing STATUS)
                movwf   STATUS                      ; Restore it to STATUS
                swapf   wsave,  F                   ; Swap the stored W value
                swapf   wsave,  W                   ; Restore it to W (swap to avoid
changing STATUS)
        retfie                                      ; Return from the interrupt
EndAsm
STARTUP:
'***PWM SETUP ////////////////////////////////////////////////////////////

    PR2     = $F9           ' SET THE PWM PERIOD TO 255}
    T2CON   = %00000100     ' PRESCALE=1, TIMER2 ON    } YIELDS A 20KHz PWM
    CCPR1L  = 0             ' SET THE DUTY CYCLE INITIALLY TO 0
    CCP1CON = %00001100     ' CONFIGURE AND START AS PWM
'***END OF PWM SETUP //////////////////////////////////////////////////////

'* TIMER ZERO SETUP//////////////////////////////////////////////////////////
T0CON  = %00001000          ' NO-PRESCALE , 16BIT, OFF AT THIS MOMENT
TMR0H = $F9                 ' LOAD THE HIGH BYTE
TMR0L = $E6                 ' LOAD THE LOW BYTE
'* END OF TIMER ZERO SETUP//////////////////////////////////////////////////

'** SET UP THE INTERRUPTS//////////////////////////////////////////////////
INTCON  = %00110000 ' TMR0 = 1, INT0 = 1 , PIE = 0, GIE=0 AT THIS MOMENT,

'** END OF SET UP OF THE INTERRUPTS//////////////////////////////////////////

'** SET UP TIMER1 FOR AC-SYCN COUNTING (FREE-RUNNING)
                  ' THIS VALUE IS ADJUSTED ONLY DURING SYNCHRONIZE PHASE
INTCON.7 = 1         ' ENABLE GLOBAL INTERRUPT
 T0CON.7  = 1         ' START TIMER 0

FIRE:
TEMP = TMR0L                    ' NEEDED FOR THE SIMULATOR TO UPDATE TMR0H
IF TIM0INT = 0 THEN             ' IF A FRESH FIRING IS NEEDED THEN
```

```
   TIM0INT = 1                    ' SET THE SOFTWARE INT FIRING FLAG

    LOOKUP INDEX, [0,25,50,74,98,120,142,162,180,197,212,_ ' GET THE PWMVALUE
    225,235,244,250,254,255,254,250,244,235,_
    225,212,197,180,162,142,120,98,74,50,25],PWMVAL

    CCPR1L = PWMVAL               ' LOAD THE PWMVAL TO THE HWPWM
         IF PHASE = 0 THEN        ' FIRE THE MOSFETS ACCORDING TO THE PHASE VALUE
             MOS1 = TIM0INT       ' FIRE USING SAFETY LOGIC
             ELSE                 ' EITHER CHANNEL 1 OR CHANNEL 2 (DEPENDS ON PHASE)
             MOS2 = TIM0INT       ' FIRING USING SAFETY LOGIC
         ENDIF

ENDIF
GOTO FIRE
```