OVERVIEW

I call my idea an 'MM5369E' as it represents the equivalent of a 5369 IC plus all the 'glue' necessary to sync-lock this 60 Hz generator to the power line. Previously, I had used either the powerline or an MM5369 as a time base for my projects, but never both, as the circuits to detect missing pulses phase-lock the crystal to the line, and transfer between the two sources was just too complicated. Now it's all contained in an inexpensive 8-pin package.

As a bonus, I was able to include a divide-by-sixty circuit to provide a 1Hz output to further simplify time-keeping functions.

APPLICATION OPERATION

My circuit uses a standard 3.579545 MHz NTSC 'color burst' crystal for the main oscillator, just as the MM5369 does. This frequency is divided by 4 by the PIC12C508, then by 256 in the TMR0 prescaler, and counted in TMR0.

It works out that 233 counts of TMR0 and 11 more instruction cycles gives exactly four, 60 Hz periods. This fact is used to provide the 60 Hz output. (It's a little 'jittery' though, if viewed on a scope, as the four cycles are not exactly the same period. Phase jitter, however, is not important for this type of time-keeping function.)

Every time that the 60 Hz output goes high, a counter is incremented as well, and this counter value is then used to drive the 1Hz output.

Sync-lock is achieved by allowing a high-to-low signal at GP2 to reset the main counter, TMR0. As long as these transitions occur, the output will continue to follow the input, and the output will have exactly the same frequency as the line frequency. As this is traceable back to national standards, it is better as a long-term reference than the crystal. As to deciding which source to use for the output, a very narrow window is set up to detect whether a 60 Hz sync pulse occurred before the timer timed out, or not. This allows an AC failure to be detected during the cycle that it failed, and a 3.58 MHz derived one to be inserted with no loss of output cycles. The narrow window does slow sync-lock down considerably, however, on return of AC, typically taking about 30 seconds to regain lock. (If the two frequencies were identical, but only differed in phase, the circuit could theoretically never return to sync-lock).

The synchronizing input is provided from either a half-wave or full-wave rectified 60 Hz source and is connected to the GP2 Schmitt trigger input through a current limiting resistor, so that peak inputs greater than the 5V supply can be used without damaging the input. The input must have a peak value of at least VDD (5V). This pin also has a large value resistor connected from it to VSS to provide a reference for floating inputs. No other special components are required for circuit operation.
BILL OF MATERIALS (BOM)

- PIC12C508
- 3.58 MHz Crystal
- 2 x 33 pF caps
- 1 x 47 KΩ, 1 x 470 KΩ resistors
APPENDIX A: SOURCE CODE

; MM5369E
; ---------
; by Jim Nagy, August 1997
;
; A replacement circuit for the MM5369 60Hz generator, using the PIC12C508
; In addition to providing a 60 Hz output from a 3.58MHz crystal,
; this circuit also provides a 1Hz output, and a sync input.
; The sync input provides better long term stability than the crystal,
; as it uses the power line frequency (which is regularly corrected).
; It can be either half-wave, or full-wave 60Hz, positive going.
;
; Circuit connections are as follows:
; - 60Hz output is from GP0 (pin 7)
; - 1Hz output is from GP1 (pin 6)
; - Sync input is on GP2 (pin 5), through a 47KΩ limiting resistor (a 470 KΩ
;   resistor is connected from this pin to Vss, to ensure a ground reference)
; - GP3 (pin 4) is configured as an active low MCLR, with internal pullup
; - A 3.579545 MHz (color burst) crystal is connected to pins 2 and 3,with
;   33 pF capacitors from each pin to Vss as well.
; - +5V is connected to pin 1, gnd to pin 8
;
;***************************************************************************
;
; Standard Equates
W EQU 0
F EQU 1

GPWUF EQU 7
PA0 EQU 5
TO EQU 4
PD EQU 3
Z EQU 2
Zero EQU 2
DC EQU 1
C EQU 0
Carry EQU 0

MCLRDdisabled EQU 0
MCLREnabled EQU H'10'
CodeProtect EQU 0
NoCodeProtect EQU H'08'
WDDTDisabled EQU 0
WDEnabled EQU H'04'
IntRCOsc EQU H'02'
ExtRCOsc EQU H'03'
XTOsc EQU H'01'
LPOsc EQU 0

; '508 Registers
INDF EQU H'00'
TMRO EQU H'01'
PCL EQU H'02'
STATUS EQU H'03'
FSR EQU H'04'
OSCCAL EQU H'05'
GPIO EQU H'06'

; program variables
Cycles EQU H'07' ; Cycle counter for 1 Hz output

; Setting the ID words...
Discrete Logic Replacement

```assembly
ORG H'0200'
ID0 Data.W H'0000'
ID1 Data.W H'0000'
ID2 Data.W H'0000'
ID3 Data.W H'0000'

; and the Fuses...

ORG H'0FFF'
CONFIG Data.W MCLREnabled + NoCodeProtect + WDTEnabled + XTosc

; PIC starts here on power up...

ORG H'00'
Init
CLRWDT ; setting up options...
MOVLW B'11000111' ; TMR0 uses int clock input, /256 prescaler
OPTION ; no pullups, and no wakeup on pin change
CLRF GPIO
MOVLW B'00111100' ; Want GP0 and GP1 as outputs,
TRIS GPIO ; others are inputs
CLRF Cycles ; prime the 1 Hz counter
DECF Cycles,F
Main
CLRF TMR0 ; start timing

; produce 1 cycle of 59 counts (16.88 msec)
Cycle1 BSF GPIO,0 ; set 60Hz output high
CALL OneHz ; and service the 1 Hz output
c11
CLRWDT ; reset the watchdog
MOVLW D'30' ; wait for 30 cycles (8.6 msec) to pass
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c11
BCF GPIO,0 ; then set 60Hz output low
c12
MOV LW D'36' ; wait to open the sync window, as
SUBWF TMR0,W ; sync may still be low due to jitter
BTFSS STATUS,Carry
GOTO c12

; check for a high sync input
GOTO c14 ; and 'arm' the circuits if it is
c13
BTFSC GPIO,2
CLRWDT ; else, reset the watchdog
MOVLW D'59'
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c13
GOTO Cycle2

; sync input was high, wait for low input
GOTO Main ; and terminate this counter loop if it is
CLRWDT ; else, reset the watchdog
MOVLW D'59'
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c14

GOTO GPIO,0 ; set 60Hz output high
```

DS40160A/4_008-page 4 © 1997 Microchip Technology Inc.
CALL OneHz ; and service the 1 Hz output
CLRWDT ; reset the watchdog
MOVLW D'88' ; wait for 29 cycles (8.3 msec) to pass
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c21
BCF GPIO,0 ; then set 60Hz output low

MOVLW D'94' ; wait to open the sync window
SUBWF TMR0,W ; (sync may still be low due to jitter)
BTFSS STATUS,Carry
GOTO c22

MOVLW D'117' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c23 ; and repeat until one of these occur
GOTO Cycle3

BTFSC GPIO,2 ; check for a high sync input
GOTO c24 ; and 'arm' the circuits if it is
CLRWDT ; else, reset the watchdog
MOVLW D'117' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c23

BTFSS GPIO,2 ; sync input was high, wait for low input
GOTO Main ; and terminate this counter loop if it is
CLRWDT ; else, reset the watchdog
MOVLW D'117' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c24

CLRWDT ; reset the watchdog
MOVLW D'146' ; wait for 29 cycles (8.3 msec) to pass
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c31
BCF GPIO,0 ; then set 60Hz output low

MOVLW D'152' ; wait to open the sync window
SUBWF TMR0,W ; (sync may still be low due to jitter)
BTFSS STATUS,Carry
GOTO c32

BTFSC GPIO,2 ; check for a high sync input
GOTO c34 ; and 'arm' the circuits if it is
CLRWDT ; else, reset the watchdog
MOVLW D'175' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c33 ; and repeat until one of these occur
GOTO Cycle4

BTFSS GPIO,2 ; sync input was high, wait for low input
GOTO Main ; and terminate this counter loop if it is
CLRWDT ; else, reset the watchdog
MOVLW D'175' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c34

; produce 1 cycle of 58 counts (16.59 msec)
; Cycle3 BSF GPIO,0 ; set 60Hz output high
CALL OneHz ; and service the 1 Hz output
CLRWDT ; reset the watchdog
MOVLW D'146' ; wait for 29 cycles (8.3 msec) to pass
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c31
BCF GPIO,0 ; then set 60Hz output low

MOVLW D'152' ; wait to open the sync window
SUBWF TMR0,W ; (sync may still be low due to jitter)
BTFSS STATUS,Carry
GOTO c32

BTFSC GPIO,2 ; check for a high sync input
GOTO c34 ; and 'arm' the circuits if it is
CLRWDT ; else, reset the watchdog
MOVLW D'175' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c33 ; and repeat until one of these occur
GOTO Cycle4

BTFSS GPIO,2 ; sync input was high, wait for low input
GOTO Main ; and terminate this counter loop if it is
CLRWDT ; else, reset the watchdog
MOVLW D'175' ; and check for end of cycle
SUBWF TMR0,W
BTFSS STATUS,Carry
GOTO c34

; produce 1 cycle of 58 counts (16.59 msec) and ~11 cycles
; At this point, we've counted 59+58+58+58=233 cycles, or 66.654 msec.
; We only have to delay a few machine cycles before repeating all...
Cycle4 BSF GPIO,0 ; set 60Hz output high
CALL OneHz ; and service the 1 Hz output
CLRWDT ; reset the watchdog
MOVLW D'204'; wait for 29 cycles (8.3 msec) to pass
SUBWF TMRO,W
BTFSS STATUS,Carry
GOTO c41
BCF GPIO,0 ; then set 60Hz output low
MOVLW D'210' ; wait to open the sync window
SUBWF TMRO,W ; (sync may still be low due to jitter)
BTFSS STATUS,Carry
GOTO c42
BTFSC GPIO,2 ; check for a high sync input
GOTO c44 ; and 'arm' the circuits if it is
CLRWDT ; else, reset the watchdog
MOVLW D'233' ; and check for end of cycle
SUBWF TMRO,W
BTFSS STATUS,Carry
GOTO c43
GOTO Main ; and repeat until one of these occur
GOTO Main ;

; ******************************************************
; OneHz - routine to service the 1Hz system

OneHz INCF Cycles,F ; the output has just gone high -count it
MOVLW D'30'
SUBWF Cycles,W ; compare count to 30
BTFSC STATUS,Carry
GOTO GT29
BSF GPIO,1 ; 0<count<30, set output high
RETLW 0

GT29 MOVLW D'60' ; we're >29, but may be 60...
SUBWF Cycles,W ; compare count to 60
BTFSC STATUS,Carry
GOTO GT59
BCF GPIO,1 ; 29<count<60, set output low
RETLW 0

GT59 CLRF Cycles ; reset cycle counter, then
BSF GPIO,1 ; set output high
RETLW 0

END