DCF77 DigiClock

digits.zip

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

```
' ---------------------------------------------------------------------------
'
'      DigiClock for DynaMite / Maxmite
'
'      One Maximite drives one screen, three screens make one DigiClock
'      Each screen is displaying two digits, either HH, MM or SS.
'
'     This program uses V4.4B of the MMBasic language, so the DuinomMites
have
'      to be flashed with the most recent firmware.
'
'      The clock is automatically set via DCF77 (Conrad Receiver)
'
http://www.conrad.de/ce/de/product/641138/C-Control-DCF-Empfaengerplatine
'
'     Wiring: The first device (SECOND) needs to receive the DCF signal on
Pin 1
'     The other devices get their time via a UART connection on COM1. So
connect Pin 16
'     of the SECOND board to pin 15 on the MINUTE and HOUR board.
'
'             (External View)
'
'             + GND for DCF Board
'             |
'             |
'             |
'             |   1  DCF In for SECOND module
'             +   +
'             o o o o o o o o o o o o o
'
'             o o o o o o o o o o o o o
'               ^         ^ +
'               |         | 15 Rx Connect to Tx of SECOND Module
'               |         +
'               |         16 Tx Connect to Rx of MINUTE and HOUR Module
'               |
'               + 3.3V for DCF Board
'
'     You will need a 10 kOhm Pullup between Vcc and the non-inverting
output of
'     the DCF module (pulling high to 3.3V) as the module has open collector
```

```
'     outputs.
'
'      Take a look at "DuinoMite MMBasic ReadMe.pdf" for more info.
'
'     The bitmaps and all basic files have to be copied to the SD card,
change
'      the DisplayMode variable for each board accordingly.
'
'     You can of course replace the clock images by your own ones, but stick
to
'      the size and the BMP format - some BMP sizes make ugly bit noise at
the end.
'
'     v2.0 written in 2014 by Tim Hagemann (tim@way2.net)
'
'     Use this code free of charge for private or educational uses. Ask for
'      permission in case of commercial usage.
'
' ---------------------------------------------------------------------------

DisplayMode = 2                      ' 0: HH, 1: MM, 2:SS
DigitY = 50                          ' Y pos of both digits
DigitLX = 53                         ' X pos of left digit
DigitRX = 250                        ' X pos of right digit
DCFPin = 1                           ' Pin for the DCF77 signal
SignalY = 300                        ' Position of the search signal line
SignalHeight = 20                    ' Pulse height on screen

'
' ----------- initialization
'

WaitingForTime = 1
FirstSyncFound = 0
OldSignalY = SignalY

oldsignal = 1

Dim dcfbit(60)

' --- configure pin as input

SetPin DCFPin, 2

cls

'
' ----------- welcome screen
'

print "Welcome to DClock v2.0"
```

```
print "----------------------"
print
print "Starting time sync."
print

if DisplayMode = 2 then
    print "This is the 'SECOND' and the DCF77 Unit"
endif

if DisplayMode = 1 then
    print "This is the 'MINUTE' Unit"
endif

if DisplayMode = 0 then
    print "This is the 'HOUR' Unit"
endif

'
' ----------- main program
'

if DisplayMode = 2 then

    DisplaySecondAndSendDCF
else

    DisplayDigitAndReceiveDCF

endif

' ---------------------------------------------------------------------------

sub DisplaySecondAndSendDCF

    ' --- setup the ticker for the DCF signal

    SetTick 10, CheckDCF77

    ' --- com1: Rx is pin 15, Tx is pin 16
    OPEN "COM1:4800" AS #1

    Do
           ' --- deferred actions

           if DoCls = 1 then
                cls
                DoCls = 0
           endif
           if UpdateLastSync = 1 then
                print @(0,415) "Last synced: ";LastSync$;"
";
```

```
                UpdateLastSync = 0
          endif


          currt$ = Time$

          ' --- when time changed

          If currt$ <> oldt$ Then
                 s1 = Val(Mid$(currt$,7,1))
                 s2 = Val(Mid$(currt$,8,1))

                 ' --- main program

                 If WaitingForTime = 0 Then

                       print #1,"TT";currt$

                       ' --- main display routine
                       PrintDigit DigitLX, DigitY, s1
                       PrintDigit DigitRX, DigitY, s2

                 EndIf

                 ' --- save time for compare

                 oldt$ = currt$

          EndIf

          ' --- wait a bit

          Pause 50
    Loop

end sub

' ---------------------------------------------------------------------

Sub DisplayDigitAndReceiveDCF

    ' --- com1: Rx is pin 15, Tx is pin 16
    OPEN "COM1:4800" AS #1

    do
        if LOC(#1) <> 0 then
            ' ---- read serial input
            a$ = input$(50,#1)
            ' ---- add to the receive uffer
            r$ = r$ + a$
            'print "'";r$;"'"
```

```
            ' ---- now process
            chr13 = instr(r$,chr$(13))
            if chr13 <> 0 then
                ' ---- we found a CR
                command$ = left$(r$,chr13)
                cmd$     = left$(command$,2)
                param$    = mid$(command$,3)
                commandfound = 0
                if cmd$="TT" then
                       commandfound = 1
                    h1 = Val(Mid$(param$,1,1))
                    h2 = Val(Mid$(param$,2,1))
                    m1 = Val(Mid$(param$,4,1))
                    m2 = Val(Mid$(param$,5,1))

                    ' --- main display routine
                    if FirstCLS = 0 then
                        cls
                        FirstCLS = 1
                    endif

                    If DisplayMode = 0 Then
                        PrintDigit DigitLX, DigitY, h1
                        PrintDigit DigitRX, DigitY, h2
                    Else
                        PrintDigit DigitLX, DigitY, m1
                        PrintDigit DigitRX, DigitY, m2
                    EndIf
                endif
                if commandfound = 0 then
                    'print "Error: command ";command$;" is invalid"
                endif
                r$ = ""
            endif

        else
            Pause 50
        endif
    loop

end sub

' ---------------------------------------------------------------------------

Sub PrintDigit (x,y,v)

       bitmapname$ = Str$(v) + ".bmp"
       LoadBMP bitmapname$, x, y

End Sub
```

```
' ---------------------------------------------------------------------------

Sub CheckDCF77

       ' --- buffer the signal to avoid glitches

       signal = Pin(DCFPin)
       ' --- update the counters
       if signal = 1 or ignorepulse = 1 then
            highcount = highcount + 1
       else
            lowcount = lowcount + 1
       endif

       ' --- in search mode, display a fancy osci like graphic
        if WaitingForTime = 1 then

            ' --- clear old line
            Line (SignalX,SignalY) - (SignalX,SignalY-SignalHeight),0
            ' --- draw new line
            NewSignalY = SignalY-signal*SignalHeight
            Line (SignalX,OldSignalY) - (SignalX,NewSignalY),1
            OldSignalY = NewSignalY
            ' --- do the X warp around
            SignalX = SignalX + 1
            if (SignalX > MM.HRES) then
                SignalX = 0
            endif
        endif

       ' --- falling edge
       if oldsignal = 1 and signal = 0 then

            ' --- this is the falling edge

            'print "Lowcnt";lowcount;"  Highcnt";highcount

            ' --- filter out very small pulses

            if highcount < 7 then
                ignorepulse = 1
                'print "Ignore pulse - too small"
            else
                if lowcount > 120 then

                    'print "Lowcount detected"
                    ' --- two cases: we found the sync and completed the
full datagram
                    '                 or we found the first sync having read
only rubbish
```

```
                    If FirstSyncFound = 1 Then

                           ' --- the is then the 2nd, 3rd,... sync.

                           ProcessDCFTime

                    Else
                           ' --- ok, now we got the first one!

                           FirstSyncFound = 1
                           if WaitingForTime = 1 then
                                print @(0,150) "First minute break
found...now receiving time...";spc(40)
                           endif
                    EndIf

                    ' --- the next bit will be bit 1, we ignore bit 60 -->
anyway 0

                    bitindex = 1
                else

                    'print "Pulse detected"
                    ' --- only process, if we completed the first round

                    If FirstSyncFound = 1 Then

                        if WaitingForTime = 1 then
                            print @(0,150) "Receiving bit number";bitindex;"
of 58";spc(40)
                        endif

                        If bitindex <= 60 Then
                           ' --- more than 100 ms? The this is a "1"

                           If highcount > 10 Then
                                  dcfbit(bitindex) = 1
                           Else
                                  dcfbit(bitindex) = 0
                           EndIf
                           'Print "BIT :", dcfbit(bitindex)
                           bitindex = bitindex + 1
                        Else

                          if WaitingForTime = 1 then
                                print @(0,150) "Lost sync. Try
again.";spc(40)
                          endif
                          ' --- something was wrong. Restart sync
                          FirstSyncFound = 0
                        EndIf
```

```
                    EndIf
                endif
                   lowcount = 0
                highcount = 0
                ignorepulse = 0

            endif
        endif
        oldsignal = signal
End Sub

' ---------------------------------------------------------------------

Sub ProcessDCFTime

        idx = 21
        mm = dcfbit(idx) + dcfbit(idx+1)*2 + dcfbit(idx+2)*4 +
dcfbit(idx+3)*8 + dcfbit(idx+4)*10 + dcfbit(idx+5)*20 + dcfbit(idx+6)*40

        p1 = Party(21,28)

        idx = 29
        hh = dcfbit(idx) + dcfbit(idx+1)*2 + dcfbit(idx+2)*4 +
dcfbit(idx+3)*8 + dcfbit(idx+4)*10 + dcfbit(idx+5)*20

        p2 = Party(29,35)

        idx = 36
        dd = dcfbit(idx) + dcfbit(idx+1)*2 + dcfbit(idx+2)*4 +
dcfbit(idx+3)*8 + dcfbit(idx+4)*10 + dcfbit(idx+5)*20

        idx = 45
        mo = dcfbit(idx) + dcfbit(idx+1)*2 + dcfbit(idx+2)*4 +
dcfbit(idx+3)*8 + dcfbit(idx+4)*10

        idx = 50
        yy = dcfbit(idx) + dcfbit(idx+1)*2 + dcfbit(idx+2)*4 +
dcfbit(idx+3)*8 + dcfbit(idx+4)*10 + dcfbit(idx+5)*20 + dcfbit(idx+6)*40 +
dcfbit(idx+7)*80

        p3 = Party(36,58)

        'Print "//////////// time", hh , mm, dd,mo,yy,p1,p2,p3

        ' --- now do some checks

        dcferr = 0

        If Fix(p1/2) <> p1/2 Then
               dcferr = 1
        EndIf
```

```
        If Fix(p2/2) <> p2/2 Then
                dcferr = 1
        EndIf

        If Fix(p3/2) <> p3/2 Then
                dcferr = 1
        EndIf

        If mo<1 Or mo>12 Then
                dcferr = 1
        EndIf

        If dd<1 Or dd>31 Then
                dcferr = 1
        EndIf

        If mm>59 Then
                dcferr = 1
        EndIf

        If hh>23 Then
                dcferr = 1
        EndIf

        If yy>25 Then
                dcferr = 1
        EndIf

        ' --- now process

        If dcferr = 0 Then

                Time$ = Str$(hh)+":"+Str$(mm)+":00"
                if WaitingForTime = 1 then
                  WaitingForTime = 0
                  ' --- deferred CLS - will otherwise get us out of sync
                  DoCls = 1
                endif
                ' --- deferred PRINT - will otherwise get us out of sync
                UpdateLastSync = 1
                LastSync$ = time$
        EndIf

End Sub

' ---------------------------------------------------------------------------

Function Party(idxa,idxe)

        p = 0
        For i = idxa To idxe
```

```
            p = p + dcfbit(i)
        Next i

        Party = p
End Function
```