```
' Source file for CO nixie clock v2.2
' Bascom version 2.0.7.7 or higher
' www.circuitsonline.net

$regfile = "m8def.dat"
$crystal = 4000000       ' 4MHz
$hwstack = 128
$swstack = 64
$framesize = 64
$lib "I2C_TWI.LBX"
$version 2 , 2 , 28

Config Base = 0       ' arrays start at 0

' todo ----------------------------------------

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

'Const Nixie_display_type = "ZM1180" ' enable correct display type constant
Const Nixie_display_type = "Z5700M"

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

Const Display_mode_hours = 0
Const Display_mode_minutes = 1

Dim N As Byte

Dim Display_time(6) As Byte , Old_display_time(6) As Byte
Dim I2c_time(6) As Byte , Check_i2c_time_flag As Bit

Dim Set_display_time As Bit , Set_display_time_index As Byte

Dim Display_mode As Bit

Dim Global_ontime As Word
Dim Crossfade(6) As Integer , New_digit_on As Word
Dim Crossfade_speed As Word

Dim Timer_flag As Byte

Dim Blink_counter As Byte
Dim Blink_active As Bit

Dim Local_ontime As Word , Local_offtime As Word
Dim Local_crossfade As Integer

Dim Bcd_seconds As Byte , Bcd_minutes As Byte , Bcd_hours As Byte       'used by I2C time routines

Dim Show_debug As Bit
Dim Debug_value As Word , Debug_display_value(6) As Byte

Dim Active_nixie As Byte , Active_nixie_index As Byte

Dim Register_data As Word

Dim Button_set_flag As Bit
Dim Button_up_flag As Bit
Dim Button_down_flag As Bit

Declare Sub I2c_write_time()       ' writes the I2C_time to the I2C RTC
Declare Sub I2c_read_time()       ' reads the I2C_time from the I2C RTC

#if Nixie_display_type = "Z5700M"
   Shiftreg_data Alias Portc.1
   Shiftreg_clock Alias Portd.3
   Shiftreg_latch Alias Portc.0
   Shiftreg_enable Alias Portd.4

   Nixie0 Alias Register_data.15
   Nixie1 Alias Register_data.14
   Nixie2 Alias Register_data.1
   Nixie3 Alias Register_data.0
   Colon Alias Register_data.7
#endif

#if Nixie_display_type = "ZM1180"
   Shiftreg_data Alias Portd.4
   Shiftreg_clock Alias Portc.0
   Shiftreg_latch Alias Portc.1
   Shiftreg_enable Alias Portd.3

   Nixie0 Alias Register_data.8
   Nixie1 Alias Register_data.15
   Nixie2 Alias Register_data.9
   Nixie3 Alias Register_data.14
   Colon Alias Register_data.4
#endif

Button_set Alias Pinb.0
Button_up Alias Pind.7
Button_down Alias Pind.6
```

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

Config Portb = &B11111110
Config Portc = &B11111111
Config Portd = &B00111011

Portb = &B00000001       ' enable pull-ups
Portd = &B11000100

Config Sda = Portc.4
Config Scl = Portc.5

Config Int0 = Falling       ' RTC 1Hz, inverted
Enable Int0
On Int0 Int0_isr

Enable Interrupts

Waitms 200       ' allow charging of the debounce capacitors to prevent false button triggers

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

Active_nixie = 1
Active_nixie_index = 0
Global_ontime = 1700
Crossfade_speed = 30
Set_display_time = 0
Display_mode = Display_mode_hours

Call I2c_read_time()


'#########################################################################################################

Do       ' mainloop
   Local_ontime = Global_ontime       ' Ontime(active_nixie_index)
   Local_offtime = 2000 - Local_ontime
   Local_crossfade = Crossfade(active_nixie_index)
   If Local_crossfade > Local_ontime Then Local_crossfade = Local_ontime

   If Set_display_time = 1 Then
      Incr Blink_counter
      If Blink_counter = 0 Then Toggle Blink_active
      If Set_display_time_index = Active_nixie_index Then
         If Blink_active = 1 Then       ' divide ontime & crossfade by 2
            Local_ontime = Local_ontime \ 2
            Local_offtime = 2000 - Local_ontime       ' calculate new offtime
            Local_crossfade = Local_crossfade \ 2
         End If
      End If
   End If

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

   Register_data = &B00000000_00000000       ' turn all anodes and cathodes off
   Colon = 1       ' inverted

   Shiftout Shiftreg_data , Shiftreg_clock , Register_data , 1
   Set Shiftreg_latch       ' latch
   Reset Shiftreg_latch

   Waitus Local_offtime       ' offtime

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

   Register_data = Lookup(display_time(active_nixie_index) , Nixie_cathodes)

   Nixie0 = Active_nixie.0       ' turn correct nixie on
   Nixie1 = Active_nixie.1
   Nixie2 = Active_nixie.2
   Nixie3 = Active_nixie.3
   Colon = 0       ' inverted

   Shiftout Shiftreg_data , Shiftreg_clock , Register_data , 1
   Set Shiftreg_latch       ' latch
   Reset Shiftreg_latch

   If Crossfade(active_nixie_index) <= 0 Then       ' no fade, normal wait time
      Waitus Local_ontime
   Else       ' crossfade, wait & switch digit
      New_digit_on = Local_ontime - Local_crossfade
      Waitus New_digit_on
      Register_data = Lookup(old_display_time(active_nixie_index) , Nixie_cathodes)

      Nixie0 = Active_nixie.0       ' turn correct nixie on
      Nixie1 = Active_nixie.1
      Nixie2 = Active_nixie.2
      Nixie3 = Active_nixie.3
      Colon = 0       ' inverted

      Shiftout Shiftreg_data , Shiftreg_clock , Register_data , 1
      Set Shiftreg_latch       ' latch
```

```
      Reset Shiftreg_latch
      Waitus Local_crossfade
      Crossfade(active_nixie_index) = Crossfade(active_nixie_index) - Crossfade_speed       ' decrease actual crossfade
   End If

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

   If Display_mode = Display_mode_hours Then
      If Active_nixie_index = 5 Then       ' 4 nixie tubes
         Active_nixie_index = 2
         Active_nixie = 1
      Else
         Incr Active_nixie_index
         Shift Active_nixie , Left
      End If
   End If

   If Display_mode = Display_mode_minutes Then
      If Active_nixie_index >= 3 Then       ' 4 nixie tubes
         Active_nixie_index = 0
         Active_nixie = 1
      Else
         Incr Active_nixie_index
         Shift Active_nixie , Left
      End If
   End If

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

   If Timer_flag = 1 Then       ' timekeeping
      Timer_flag = 0

      Old_display_time(0) = Display_time(0)       ' save old digit
      Crossfade(0) = Global_ontime       ' set fade value
      Incr Display_time(0)       ' increase single seconds

      If Display_time(0) = 10 Then
         Display_time(0) = 0
         Old_display_time(1) = Display_time(1)       ' save old digit
         Crossfade(1) = Global_ontime       ' set fade value
         Incr Display_time(1)       ' increase ten seconds

         If Display_time(1) = 6 Then
            Display_time(1) = 0
            Old_display_time(2) = Display_time(2)       ' save old digit
            Crossfade(2) = Global_ontime       ' set fade value
            Incr Display_time(2)       ' increase single minutes

            If Display_time(2) = 10 Then
               Display_time(2) = 0
               Old_display_time(3) = Display_time(3)       ' save old digit
               Crossfade(3) = Global_ontime       ' set fade value
               Incr Display_time(3)       ' increase ten minutes
               If Display_time(3) = 3 Then
                  Set Check_i2c_time_flag       ' 1800 seconds after hour change, set check I2C time flag to start time correction
               End If

               If Display_time(3) = 6 Then
                  Display_time(3) = 0
                  Old_display_time(4) = Display_time(4)       ' save old digit
                  Crossfade(4) = Global_ontime       ' set fade value
                  Incr Display_time(4)       ' increase single hours

                  If Display_time(4) = 10 Then
                     Display_time(4) = 0
                     Old_display_time(5) = Display_time(5)       ' save old digit
                     Crossfade(5) = Global_ontime       ' set fade value
                     Incr Display_time(5)       ' increase ten hours
                  End If

                  If Display_time(5) = 2 Then
                     If Display_time(4) = 4 Then
                        Display_time(4) = 0
                        Display_time(5) = 0
                     End If
                  End If

               End If       ' Display_time(3) = 6
            End If       ' Display_time(2) = 10
         End If       ' Display_time(1) = 6
      End If       ' Display_time(0) = 10

   End If       ' Timer1_fired = 1

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

   If Set_display_time = 0 Then       ' function only works when not setting the time
      If Button_down = 0 Then
         If Button_down_flag = 0 Then
            Button_down_flag = 1

         Toggle Display_mode
```

```
            For N = 0 To 5
               Crossfade(n) = 0       ' set all crossfade values to 0 because registers weren't updated by the display routine
            Next

         End If
      Else
         Button_down_flag = 0
      End If       ' Button_up = 0
   End If       ' Set_display_time = 0

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

   If Button_set = 0 Then
      If Button_set_flag = 0 Then
         Button_set_flag = 1

         If Set_display_time = 1 Then
            Decr Set_display_time_index
            If Set_display_time_index = 1 Then
               Display_mode = Display_mode_minutes
            End If
            If Set_display_time_index = 255 Then
               Reset Set_display_time       ' turn time setting off
               Call I2c_write_time()       ' save I2C time to I2C RTC
               Display_mode = Display_mode_hours       ' turn back to hour mode
            End If
         Else       ' turn time setting on
            Set Set_display_time
            Set_display_time_index = 5       ' first digit
            Display_mode = Display_mode_hours       ' make first digit visible
         End If

      End If       ' Button_set_flag = 0
   Else
      Button_set_flag = 0
   End If       ' Button_set = 0

   If Set_display_time = 1 Then
      If Button_up = 0 Then
         If Button_up_flag = 0 Then
            Button_up_flag = 1
            Select Case Set_display_time_index
               Case 0
                  Incr Display_time(0)       ' increase single seconds
                  If Display_time(0) = 10 Then Display_time(0) = 0
               Case 1
                  Incr Display_time(1)       ' increase ten seconds
                  If Display_time(1) = 6 Then Display_time(1) = 0
               Case 2
                  Incr Display_time(2)       ' increase single minutes
                  If Display_time(2) = 10 Then Display_time(2) = 0
               Case 3
                  Incr Display_time(3)       ' increase ten minutes
                  If Display_time(3) = 6 Then Display_time(3) = 0
               Case 4
                  Incr Display_time(4)       ' increase single hours
                  If Display_time(4) = 10 Then Display_time(4) = 0
                  If Display_time(5) = 2 Then
                     If Display_time(4) = 4 Then Display_time(4) = 0       ' also reset single hours to prevent 24h
                  End If
               Case 5
                  Incr Display_time(5)       ' increase ten hours
                  If Display_time(5) = 3 Then Display_time(5) = 0
                  If Display_time(4) > 3 Then
                     If Display_time(5) = 2 Then Display_time(5) = 0       ' also reset ten hours to prevent 33h
                  End If
            End Select
         End If
      Else
         Button_up_flag = 0
      End If       ' Button_up = 0

      If Button_down = 0 Then
         If Button_down_flag = 0 Then
            Button_down_flag = 1
            Select Case Set_display_time_index
               Case 0
                  Decr Display_time(0)       ' decrease single seconds
                  If Display_time(0) = 255 Then Display_time(0) = 9
               Case 1
                  Decr Display_time(1)       ' decrease ten seconds
                  If Display_time(1) = 255 Then Display_time(1) = 5
               Case 2
                  Decr Display_time(2)       ' decrease single minutes
                  If Display_time(2) = 255 Then Display_time(2) = 9
               Case 3
                  Decr Display_time(3)       ' decrease ten minutes
                  If Display_time(3) = 255 Then Display_time(3) = 5
               Case 4
                  Decr Display_time(4)       ' decrease single hours
                  If Display_time(4) = 255 Then
                     If Display_time(5) = 2 Then
                        Display_time(4) = 3       ' set to 3 to prevent 29h
```

```
                  Else
                     Display_time(4) = 9
                  End If
               End If
            Case 5
               Decr Display_time(5)       ' decrease ten hours
               If Display_time(5) = 255 Then
                  If Display_time(4) > 3 Then
                     Display_time(5) = 1
                  Else
                     Display_time(5) = 2
                  End If
               End If
         End Select
      End If
   Else
      Button_down_flag = 0
   End If       ' Button_down = 0

End If       ' Set_display_time = 1

Loop       ' mainloop

'#############################################################################################

Sub I2c_write_time()
   For N = 0 To 5
      I2c_time(n) = Display_time(n)       ' copy display time to I2C time
   Next

   Bcd_seconds = I2c_time(1)
   Shift Bcd_seconds , Left , 4
   Bcd_seconds = Bcd_seconds Or I2c_time(0)

   Bcd_minutes = I2c_time(3)
   Shift Bcd_minutes , Left , 4
   Bcd_minutes = Bcd_minutes Or I2c_time(2)

   Bcd_hours = I2c_time(5)
   Shift Bcd_hours , Left , 4
   Bcd_hours = Bcd_hours Or I2c_time(4)

   I2cstart
   I2cwbyte &HA0       ' address
   I2cwbyte &H00       ' control register address
   I2cwbyte &B10000000       ' stop counting
   I2cstop

   I2cstart       'generate start
   I2cwbyte &HA0       ' address
   I2cwbyte &H02       ' seconds register address
   I2cwbyte Bcd_seconds
   I2cwbyte Bcd_minutes
   I2cwbyte Bcd_hours
   I2cstop

   I2cstart
   I2cwbyte &HA0       ' write address
   I2cwbyte &H00       ' control register address
   I2cwbyte &B00000000       ' start counting
   I2cstop
End Sub

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

Sub I2c_read_time()
   I2cstart
   I2cwbyte &HA0       ' address
   I2cwbyte &H02       ' seconds register address
   I2cstart       ' repeat start
   I2cwbyte &HA1       ' address for reading
   I2crbyte Bcd_seconds , Ack
   I2crbyte Bcd_minutes , Ack
   I2crbyte Bcd_hours , Nack
   I2cstop

   I2c_time(0) = Bcd_seconds And &B0000_1111
   Shift Bcd_seconds , Right , 4
   I2c_time(1) = Bcd_seconds

   I2c_time(2) = Bcd_minutes And &B0000_1111
   Shift Bcd_minutes , Right , 4
   I2c_time(3) = Bcd_minutes

   I2c_time(4) = Bcd_hours And &B0000_1111
   Shift Bcd_hours , Right , 4
   I2c_time(5) = Bcd_hours

   For N = 0 To 5
      Display_time(n) = I2c_time(n)       ' copy I2C time to display time
   Next
End Sub
```

```
'#################################################################################################

Int0_isr:
   Timer_flag = 1
Return

End

#if Nixie_display_type = "Z5700M"
   Nixie_cathodes:
      Data &B00000000_00010000%       ' 0
      Data &B00001000_00000000%       ' 1
      Data &B00010000_00000000%       ' 2
      Data &B00000100_00000000%       ' 3
      Data &B00000000_01000000%       ' 4
      Data &B00000000_00000100%       ' 5
      Data &B00000000_00100000%       ' 6
      Data &B00000000_00001000%       ' 7
      Data &B00100000_00000000%       ' 8
      Data &B00000010_00000000%       ' 9
#endif

#if Nixie_display_type = "ZM1180"
   Nixie_cathodes:
      Data &B00000000_00000010%       ' 0
      Data &B00000000_10000000%       ' 1
      Data &B00000000_00000001%       ' 2
      Data &B00000000_00001000%       ' 3
      Data &B00000100_00000000%       ' 4
      Data &B00100000_00000000%       ' 5
      Data &B00001000_00000000%       ' 6
      Data &B00010000_00000000%       ' 7
      Data &B00000000_00100000%       ' 8
      Data &B00000000_00000100%       ' 9
#endif
```