

\$regfile = "2313def.dat"

\$crystal = 4000000

\$baud = 1200

' Variablen

Dim Temp\_ziffer As Byte

'welche ziffer soll bearbeitet werden

Dim Bcd\_aus As Byte  
soll

'enthält den Wert der dez zahl die ausgegeben werden

Dim Ausgang As Byte  
Tabelle

'Segmentfolge die Ausgegeben wird aus Segmentcode -

Dim Sek\_blink As Bit

'Blinkt im Sekunden Takt

Dim Zl\_sekunde As Byte

'Zeit Variablen

Dim Zl\_minute As Byte

'enthalten die Uhrzeit

Dim Zl\_stunde As Byte

'Zl-... => Low Wert der Zeit

Dim Zh\_sekunde As Byte

'Zh-... => High Wert der Zeit

Dim Zh\_minute As Byte

Dim Zh\_stunde As Byte

' Ein-/Ausgänge

Config Portb = Output

'Port B als Ausgang

Ddrd = &B0111100

'1 Ausgang, 0 Eingang

Portd = &B1000011

'1 = PullUp aktiviert Eingang (Taster gegen Masse)

'Ausgänge:

Segment\_a Alias Portb.0

Segment\_b Alias Portb.1

Segment\_c Alias Portb.2

Segment\_d Alias Portb.3

Segment\_e Alias Portb.4

Segment\_f Alias Portb.5

Segment\_g Alias Portb.6

Dezimalpunkt Alias Portb.7

Ziffer\_1 Alias Portd.2

Ziffer\_2 Alias Portd.3

Ziffer\_3 Alias Portd.4

Ziffer\_4 Alias Portd.5

'Eingänge:

T\_min\_stellen Alias Pind.0

T\_h\_stellen Alias Pind.1

' Sonstiges

Const Load\_timer1 = 62755 'Konstante auf die der Timer1 geladen wird

'4MHz = 62755 | 62760 etwas zu langsam <> 62750 etwas zu schnell

'+++++

' Timer0 für Multiplexausgabe

'+++++

On Timer0 7segment

'Interrupt-Routine für Timer0-Overflow



```
Incr Zh_sekunde                                'und der High Wert um eins erhöht
  If Zh_sekunde = 6 Then                        'Ist der High Wert 6 (entspricht 60 Sekunden)
    Zh_sekunde = 0                             'wird er auf 0 gesetzt
    Incr Zl_minute                             'und der Minuten Low Wert um eins erhöht
  End If
End If
```

```
If Zl_minute = 10 Then
  Zl_minute = 0
  Incr Zh_minute
End If
```

```
If Zh_minute = 6 Then
  Zh_minute = 0
  Incr Zl_stunde
End If
```

```
If Zl_stunde = 10 Then
  Zl_stunde = 0
  Incr Zh_stunde
End If
```

```
If Zh_stunde = 2 And Zl_stunde = 4 Then
  Zl_stunde = 0
  Zh_stunde = 0
End If
```



```
If T_min_stellen = 0 And T_h_stellen = 0 Then      'Wenn "T_Min_stellen" und "T_H_stellen"  
gleichzeitig betätigt
```

```
    Zl_minute = 0                                'dann werden Minuten und Stunden auf 0 gestzt
```

```
    Zh_minute = 0
```

```
    Zl_stunde = 0
```

```
    Zh_stunde = 0
```

```
End If
```

```
Loop                                             'Ende Hauptschleife
```

```
End                                             'Programm Ende
```

```
'+++++
```

```
' Interrupt-Routine für Ziffer ausgabe im 7 Segmentcode
```

```
'+++++
```

```
' Bei jedem Aufruf wir nur eine Ziffer ausgegeben, d.h nur eine Anzeige wird
```

```
' angesteuert. Die anderen bleiben dunkel. Bei jedem Aufruf wird eine Anzeige
```

```
' weitergeschaltet.
```

```
,
```

```
' Anzeigen anordnung
```

```
,
```

```
' 1111 2222 3333 4444
```

```
' 1 1 2 2 3 3 4 4
```

```
' 1111 2222 3333 4444
```

```
' 1 1 2 2 3 3 4 4
```

```
' 1111o 2222o 3333o 4444o
'
'          ^-- Dezimalpunkt
'          ^---- 4. Ziffer | Minute Low Wert | Variable Zl_minute
'          ^----- 3. Ziffer | Minute High Wert | Variable Zh_minute
'          ^----- 2. Ziffer | Stunde Low Wert | Variable Zl_stunde
'          ^----- 1. Ziffer | Stunde High Wert | Variable Zh_stunde
```

7segment:

```
If Temp_ziffer = 4 Then Temp_ziffer = 0          'bestimmt die auszugebende Ziffer (1 bis 4)
```

```
Incr Temp_ziffer                                'bei jedem aufruf des Lable wir eine andere Ziffer
ausgegeben
```

```
Portb = &B11111111                             'Alle Ziffern aus, ansonsten gibt es "Geister" Anzeigen
```

```
Select Case Temp_ziffer                        'welche Ziffer soll angesteuert werden
```

```
Case 1 : Ziffer_1 = 1                          '1. Ziffer
```

```
    Ziffer_2 = 0
```

```
    Ziffer_3 = 0
```

```
    Ziffer_4 = 0
```

```
    Bcd_aus = Zh_stunde
```

```
Case 2 : Ziffer_1 = 0                          '2. Ziffer
```

```
    Ziffer_2 = 1
```

```
    Ziffer_3 = 0
```

Ziffer\_4 = 0

Dezimalpunkt = Sek\_blink                   'Dezimalpunkt

Bcd\_aus = Zl\_stunde

Case 3 : Ziffer\_1 = 0                   '3. Ziffer

Ziffer\_2 = 0

Ziffer\_3 = 1

Ziffer\_4 = 0

Bcd\_aus = Zh\_minute

Case 4 : Ziffer\_1 = 0                   '4. Ziffer

Ziffer\_2 = 0

Ziffer\_3 = 0

Ziffer\_4 = 1

Bcd\_aus = Zl\_minute

End Select

'Ermittelt aus der Segmentfolge-Tabelle den 7 Segmentcode für die auszugebende Zahl

Select Case Bcd\_aus

Case 1 : Ausgang = Lookup(1 , Segmentfolge)

Case 2 : Ausgang = Lookup(2 , Segmentfolge)

Case 3 : Ausgang = Lookup(3 , Segmentfolge)

Case 4 : Ausgang = Lookup(4 , Segmentfolge)

Case 5 : Ausgang = Lookup(5 , Segmentfolge)

Case 6 : Ausgang = Lookup(6 , Segmentfolge)

Case 7 : Ausgang = Lookup(7 , Segmentfolge)

Case 8 : Ausgang = Lookup(8 , Segmentfolge)

Case 9 : Ausgang = Lookup(9 , Segmentfolge)



Case 0 And Temp\_ziffer = 1 : Ausgang = Lookup(10 , Segmentfolge) 'Nullstellenunterdrückung bei der ersten Ziffer

Case 0 And Temp\_ziffer > 1 : Ausgang = Lookup(0 , Segmentfolge) 'wenn nicht erste Ziffer Null anzeigen

End Select

'Ausgabe des 7 Segmentcodes (Invertierend [Not] bei Anz. mit gem. Anode)

Segment\_a = Not Ausgang.0 'Segment A

Segment\_b = Not Ausgang.1 'Segment B

Segment\_c = Not Ausgang.2 'Segment C

Segment\_d = Not Ausgang.3 'Segment D

Segment\_e = Not Ausgang.4 'Segment E

Segment\_f = Not Ausgang.5 'Segment F

Segment\_g = Not Ausgang.6 'Segment G

Return

'+++++

' Interrupt-Routine für Timer1 / Zeitbasis

'+++++

Sekunde\_plus:

Load Timer1 , Load\_timer1

'Lädt den Timer auf einen wert vor

Incr Zl\_sekunde

'Erhöht den Low Wert der Sekunde um eins

Toggle Sek\_blink

'Ändert sich jede Sek., 1sec an 1sec aus

Return

'+++++

' 7 Segmentcode - Tabelle

'+++++

'

'--A--

' | |

' F B

' | |

'--G--

' | |

' E C

' | |

'--D--

Segmentfolge:

'GFEDCBA / "1" Segment AN

Data &B0111111

'0

Data &B0000110

'1

Data &B1011011

'2

Data &B1001111	'3
Data &B1100110	'4
Data &B1101101	'5
Data &B1111101	'6
Data &B0000111	'7
Data &B1111111	'8
Data &B1101111	'9
Data &B0000000	'leeres Feld (zur Nullstellenunterdrückung)