# **Mikrocontroller-Einstieg**

Teil 12: I<sup>2</sup>C-Lesen



## mit **BASCOM-AVR**

Nachdem in Teil 11 der I<sup>2</sup>C-Busaufbau, die Adressierung und das Schreiben vom Master zum Slave erläutert wurden, werden hier nun I<sup>2</sup>C-Anwendungen gezeigt, bei denen auch Daten vom Slave zum Master übertragen werden. Das Ansteuern einer 7-Segment-Anzeige per I<sup>2</sup>C, eine Temperatursensor-Auswertung und eine Echtzeituhr (Realtime-Clock = RTC) sind praktische Anwendungen des I<sup>2</sup>C-Busses in diesem Artikel.

### 7-Segment-Ansteuerung mit SAA1064

Unter der Bezeichnung I<sup>2</sup>C-4-Digit-LED-Display I2C-4DLED (Best.-Nr. J5-10 56 97) bietet ELV eine kompakte Einheit aus einem über I<sup>2</sup>C ansteuerbaren Displaycontroller für eine 4-stellige 7-Segment-Anzeige (SAA1064) mit 4 7-Segment-Anzeigen, einem präzisen I<sup>2</sup>C-Temperatursensor (MCP9801) und 4 Tastern an. Die 3 Teile (Anzeige, Temperatursensor und Taster) lassen sich wie in Bild 1 gemeinsam oder durch Abtrennen voneinander einzeln einsetzen.

In Bild 1 ist die einfache Beschaltung zwischen Mikrocontroller und dem I2C-4DLED-Modul zu sehen. Für die Verbindung zu der 4-stelligen 7-Segment-Anzeige und dem Temperatursensor sind neben einer gemeinsamen GND-Leitung und einer 5-V-Spannungsversorgung nur die 2 I<sup>2</sup>C-Leitungen SCL und SDA mit je einem Pull-up-Widerstand nötig. Die Tasten sind normale gegen GND tastende Drucktasten, von denen eine mit dem Eingangspin D.2 verbunden ist. Zunächst soll der schreibende Zugriff auf die 4 7-Segment-Anzeigen durch ein BASCOM-Programm dargestellt werden.

```
' BASCOM-Programm
 SAA1064 4fach-7-Segment-Treiber an I2C (C.4 und C.5)
  Linke Ziffer ist Stelle 1.
                                Stellen: 1234
   In: Taster an D.2 für Hochzählen
        4 7-Segment-Anzeigen über SAA1064 I2C-7-Segment-Anzeigentreiber
' Out:
' In/Out: I2C SDA = C.4 mit 10k Pull-up-Widerstand
' In/Out: I2C SCL = C.5 mit 10k Pull-up-Widerstand
$reafile = ...M88def.dat"
                                           'Verwendeter Chip
$crystal = 3686400
                                           'Verwendete Frequenz
$hwstack = 40
                                           'Rücksprungadressen (je 2), Registersicherungen (32)
\$swstack = 40
                                           'Parameteruebergaben (je 2), LOCALs (je 2)
$framesize = 60
                                           'Parameter (Daten-Laenge), Rechenbereich Funktionen
Config Debounce = 50
Declare Sub Ziffer anzeigen(byval Stelle As Byte , Byval Ziffer As Byte)
Declare Sub Zahl anzeigen(byval Zahl As Word )
Config Sda = Portc.4
                                           'I2C-Pins definieren
Config Scl = Portc.5
I2cinit
Const Saa1064 adresse = &B01110000
                                     '&H70 I2C-Adresse des SAA1064
'Adresse durch Anlegen einer Spannung an Pin 1
'Adr-Pin = GND -> &B01110000 &h70
'Adr-Pin = 1,9V -> &B01110010 &h72
'Adr-Pin = 3,1V -> &B01110100
                                &h74
Adr-Pin = 5V
                 -> &B01110110 &h76
                                                                  Mikro
                                                                 controlle
                                                                               Rpu
                                                                      SC
                                                                      SD
```

```
Ē
Config Pind.2 = Input
                                          'Taster
                                          'Pull-up-Widerstand
Portd_2 = 1
                                          'Segment, wo Dezimalpunkt leuchten soll (0 = kein Dezimalpunkt)
Dim Dezimalpunktstelle As Byte
Dim Stelle As Byte
Dim Zahl As Word
Dim I As Byte
Dim Ziffer(4) As Byte
'Anzeige sehr hell und 4-Stellen-Modus:
I2cstart
I2cwbyte Saa1064 adresse
I2cwbyte 0
                                          'Control-Byte folgt. Vgl. Tabelle 1 in Beschreibung des ELV I2C-4DLED
I2cwbyte &B011110111
'Bit 0 = 1 4
                                          'Vgl. Tabelle 2 in Beschreibung des ELV I2C-4DLED
                   4 Stellen mit Multiplex
  Bit 4 bis Bit 6 Helligkeit:
  000 = aus
   ...
' 111 = sehr hell
I2cstop
'ELV schreiben:
I2cstart
I2cwbyte Saa1064 adresse
I2cwbyte 1
                                          'Ab Register 01 (Digit 1) schreiben
12cwbyte &B01111001
                                          'E Segmente Digit 1
                                                                   .
Vgl. Tabelle 3
12cwbyte &B00111000
                                          'T.
                                               Segmente Digit 2
12cwbyte &B00111110
                                              Segmente Digit 3
                                          '11
12cwbvte &B00001000
                                               Segmente Digit 4
I2cstop
Wait 3
'Alle Ziffern auf 0:
For Stelle = 1 To 4
                                          'Für jede Stelle ..
                                          '.. eine 0 anzeigen
 Call Ziffer_anzeigen(stelle , 0)
Next Stelle
Wait 2
'Anzeige dunkler:
I2cstart
I2cwbyte Saa1064_adresse
                                          'Slaveadresse des SAA1064
                                          'In Control-Byte schreiben
I2cwbyte 0
                                                                                                           2345
12cwbyte &B00010111
                                          'Zu schreibende Zahl
 Bit 0 = 1
                   4 Stellen mit Multiplex
  Bit 4 bis Bit 6 Helligkeit:
' 000 = aus
' 111 = sehr hell
I2cstop
Wait 2
'Jede Stelle einzeln von 0 bis 9 zählen:
                                          'Für Stelle 1 bis 4
For Stelle = 1 To 4
                                          'Jeweils von 0 bis 9 zählen
  For I = 0 To 9
    Call Ziffer_anzeigen(stelle , I)
                                          'Ziffer an entsprechender Stelle anzeigen
    Waitms 500
  Next I
Next Stelle
Wait 2
                                (Zehntelsekunden und Hundertstelsekunden nach dem Dezimalpunkt)
'Von 00.00 bis 99.99 zählen:
Dezimalpunktstelle = 2
                                          'Dezimalpunkt hinter Digit 2
For Zahl = 0 To 9999
                                          'Von 0 bis 9999..
  Call Zahl_anzeigen(zahl)
                                          '..Zahl anzeigen
  Waitms 10
  If Pind.2 = 0 Then Exit For
                                          'Zum vorzeitigen Verlassen der Zählschleife
Next Zahl
Wait 2
'Mit Taste Zahl jeweils 1 erhöhen:
Dezimalpunktstelle = 0
                                          'Kein Dezimalpunkt
Zahl = \bar{0}
Do
 Debounce Pind.2 , 0 , Tastenroutine , Sub
Call Zahl_anzeigen(zahl)
Loop
End
Tastenroutine:
 Incr Zahl
Return
Sub Ziffer anzeigen(byval Stelle As Byte , Byval Ziffer As Byte)
'Ziffer an einer Stelle anzeigen
Local Segmente As Byte
Select Case Ziffer
  Case 0 : Segmente = & B00111111
                                          '&H3F
  Case 1 : Segmente = &B00000110
                                          '&H06
  Case 2 : Segmente = &B01011011
                                          '&H5B
  Case 3 : Segmente = &B01001111
                                          '&H4F
  Case 4 : Segmente = & B01100110
                                          '&H66
  Case 5 : Segmente = &B01101101
                                          '&H6D
  Case 6 : Segmente = &B01111101
                                          '&H7D
```

www.elvjournal.de

```
'&H07
  Case 7 : Segmente = & B00000111
 Case 8 : Segmente = &B01111111
                                                                                  '&H7F
  Case 9 : Segmente = & B01101111
                                                                                  '&H6F
  Case Else : Segmente = & B1000000
                                                                                  '&H80 Dezimalpunkt
End Select
If Dezimalpunktstelle = Stelle Then Segmente = Segmente Or &H80
                                                                                  'Gqf. Bit für Dezimalpunkt setzen
I2cstart
I2cwbyte Saa1064_adresse
                                                                                  'Slave-Adresse
                                                                                  'Register 1, 2, 3 oder 4
12cwbvte Stelle
                                                                                  'Bitmuster der Segmente
12cwbyte Segmente
I2cstop
End Sub
Sub Zahl_anzeigen(byval Zahl As Word )
'Komplette Zahl anzeigen
Local Zahl str As String
Local Zeichen As String
Local Position As Byte
Zahl_str = Str(zahl)
                                                                                 'Zahl in String '123'
While Len(zahl_str) < 4 : Zahl_str = " " + Zahl_str : Wend</pre>
                                                                                  'Gqf. links mit Leerz. auffüllen ' 123'
For Position = 1 To 4
                                                                                  'Im String von links
 Zeichen = Mid(zahl_str , Position , 1)
                                                                                  'Ein Zeichen
  Stelle = Position
                                                                                  'In Zahl von links
  Ziffer(stelle) = Val(zeichen)
                                                                                 'Als Ziffer
Next Position
For Stelle = 1 To 4
 Call Ziffer_anzeigen(stelle , Ziffer(stelle))
Next Stelle
End Sub
```

Erläuterungen:

Auch in diesem Demoprogramm sieht man die für I<sup>2</sup>C typischen Elemente, die grundsätzlich bereits in Teil 11 der Artikelserie erläutert wurden.

Zunächst erfolgt mit CONFIG Sda, CONFIG Scl und I2cinit die Beschreibung, welche Portpins für die I<sup>2</sup>C-Kommunikation verwendet werden sollen, sowie die Initialisierung des I<sup>2</sup>C-Busses.

Über die Jumper J7 bis J10 (welche je nach Einstellung 4 verschiedene Spannungswerte an Pin 1 des Displaycontrollers anlegen) lassen sich 4 4-stellige 7-Segment-Anzeigen adressieren. Man hat also insgesamt 16 Stellen, die über 2 I<sup>2</sup>C-Busleitungen angesteuert werden können! Im Programm wird die Konstante Saa1064\_adresse mit der Adresse des anzusteuernden Moduls definiert. Das Ansprechen der Anzeige erfolgt nach dem üblichen I<sup>2</sup>C-Schema:

- 1. Starten der I<sup>2</sup>C-Kommunikation mit I2cstart
- 2. Adressieren eines I<sup>2</sup>C-Slaves mit I2cwbyte Saa1064\_adresse
- 3. Senden des zu beschreibenden Registers des Displaytreibers mit I2cwbyte 0 für das Control-Byte oder I2cwbyte n für das Beschreiben der n-ten Stelle des Displays
- 4. Schreiben eines Werts mit I2cwbyte
- 5. I2cstop, um das Ende der I<sup>2</sup>C-Kommunikation anzugeben und den Bus freizugeben

Wenn bei Schritt 4 mehrere Schreibvorgänge hintereinander erfolgen, dann können die Stellen 1 bis 4 in einem Durchgang hintereinander beschrieben werden, wie man gut im Programm unter der Überschrift "ELV\_ schreiben" sehen kann. In einer FOR-NEXT-Schleife werden dann die Stellen 1 bis 4 nacheinander mit null (0) beschrieben. Es wird jeweils das Un-

terprogramm "Ziffer\_anzeigen" aufgerufen. In diesem Unterprogramm wird die anzuzeigende Ziffer in einen Segmentcode umgesetzt, der nach dem eben geschilderten I<sup>2</sup>C-Schema (Schritte 1 bis 5) an den Anzeigentreiber gesendet wird.

Unter der Überschrift "Jede Stelle einzeln von 0 bis 9 zählen" sieht man 2 FOR-NEXT-Schleifen – die äußere für die 4 Stellen und die innere für das Hochzählen der anzuzeigenden Ziffer.

Um eine 4-stellige Zahl anzuzeigen, wurde das Unterprogramm "Zahl\_anzeigen" erstellt, in dem die Zahl in die 4 Stellen zerlegt wird. Die Zahl wird dort zunächst in einen 4-stelligen String umgewandelt, dann werden die 4 Stellen hintereinander ausgegeben.

#### I<sup>2</sup>C-Lesen

Das Schema für das Lesen von Daten vom Slave (siehe Bild 2) ist sehr ähnlich zu dem oben geschilderten Schema.

- 1. Starten der I<sup>2</sup>C-Kommunikation mit I2cstart
- 2. Adressieren eines I<sup>2</sup>C-Slaves mit I2cwbyte Slave\_Schreibadresse
- 3. Schreiben des zu lesenden Registers des I<sup>2</sup>C-Slaves mit I2cwbyte
- 4. Neuer I<sup>2</sup>C-Start mit I2cstart
- 5. Adressieren der Lese-Adresse des I<sup>2</sup>C-Slaves mit I2cwbyte Slave\_Leseadresse
- 6. Ein Byte einlesen mit I2crbyte Variablenname
- 7. I2cstop, um das Ende der I<sup>2</sup>C-Kommunikation anzugeben

### Temperatur aus l<sup>2</sup>C-Temperatursensor lesen

Außer der 4-stelligen 7-Segment-Anzeige befindet sich auf dem Modul I2C-4DLED (Best.-Nr. J5-10 56 97) auch ein sehr präziser I<sup>2</sup>C-Temperatursensor vom Typ MCP9801. Das Auslesen des Temperaturwerts aus dem Temperatursensor erfolgt in der Funktion Mcp9801\_lesen nach dem Schema von Bild 2.

E



```
Sub Temperatur anzeigen(byval Temperatur As Single )
'Eingabe ist der Temperaturwert
'Ausgabe mit einer Nachkommastelle über SAA1064 auf 7-Segment-Anzeige
Local Vorkomma As Integer
Local Nachkomma_single As Single
'Local Nachkomma As Byte
Local Zahl_str As String *
Local Zeichen As String * 1
Local Ziffer As Byte
Local Zehner As Byte , Einer As Byte
If Temperatur < -9.5 Then Temperatur = -9.9
                                                                                 'Unter -9.5 wird immer -9.9° angezeigt
'Vorkomma
Vorkomma = Int(temperatur)
                                                                                  '-4.5 -> -4
Vorkomma = Abs(vorkomma)
                                                                                  '-4 -> 4
Zahl str = Str(vorkomma)
                                                                                  'Zahl in String '4'
While Len(zahl_str) < 2 : Zahl_str = " " + Zahl_str : Wend</pre>
                                                                                  'ggf. mit Leerz. links auffüllen ' 4'
Zeichen = Mid(zahl_str , 1 , 1)
Zehner = Val(zeichen)
                                                                                  'Linkes Zeichen
                                                                                  'Als Ziffer
Zeichen = Mid(zahl str , 2 , 1)
                                                                                  'Zweites Zeichen
Einer = Val(zeichen)
                                                                                  'Als Ziffer
If Temperatur < 0 Then</pre>
                                                                                  'Wenn negativ..
  I2cstart
  I2cwbyte Saa1064_adresse
  I2cwbyte 1
  12cwbyte &B0100000
                                                                                  '... dann Minuszeichen
  I2cstop
Elseif Temperatur < 10 Then
                                                                                  'Wenn einstellig vor dem Komma
  I2cstart
  12cwbyte Saa1064 adresse
  I2cwbyte 1
  12cwbyte &B01000000
                                                                                  '... dann Leerzeichen
  I2cstop
Else
                                                                                  'Wenn >= 10
 Call Ziffer_anzeigen(1 , Zehner , 0)
                                                                                  '... dann Zehner anzeigen ohne Dezimalpunkt
End If
Call Ziffer_anzeigen(2 , Einer , 1)
                                                                                  'Einer auf jeden Fall anzeigen u. Dezimalpunkt
'Nachkomma: .5 oder .0 oder .9 an Stelle 3 schreiben
Zahl_str = Fusing(temperatur , "#.#")
                                                                                  'Temperaturwert zu Zeichenkette
Zeichen = Right(zahl_str , 1)
                                                                                  'Zeichen ganz rechts ist die Nachkommastelle
Ziffer = Val(zeichen)
                                                                                  'Von Zeichen zu Zahl umwandeln
Call Ziffer anzeigen(3 , Ziffer , 0)
                                                                                  'Nachkommastelle ohne Dezimalpunkt anzeigen
'Gradzeichen an rechte Stelle = 4. Stelle schreiben
I2cstart
12cwbyte Saa1064 adresse
I2cwbyte 4
12cwbyte &B01100011
                                                                                  '° Gradzeichen
I2cstop
End Sub
Function Mcp9801_lesen() As Single
 Temperatur per I2C vom MCP9801 einlesen
'Ausgabe(=Wert der Funktion) ist der Temperaturwert mit einer Nachkommastelle
Local Vorkomma As Byte : Vorkomma = 0
Local Nachkomma As Byte : Nachkomma = 0
Local Sensorwert As Integer
I2cstart
I2cwbyte Mcp9801 adresse schreiben
                                                                                  'Schreibadresse des Sensors
I2cwbyte 0
                                                                                  '0 ist das Temperaturregister
I2cstart
I2cwbyte Mcp9801_adresse_lesen
                                                                                  'Leseadresse des Sensors
12crbyte Vorkomma , Ack
                                                                                  'Byteweise einlesen
I2crbyte Nachkomma , Nack
I2cstop
                                                                                  'Zwei gelesene Bytes zu einer 16-Bit-Variablen zusammenbauen
Sensorwert = Makeint(nachkomma ,
                                  Vorkomma)
Shift Sensorwert , Right , 7 , Signed
Mcp9801_lesen = Sensorwert / 2
                                                                                  'Vorzeichenberücksichtigend nach rechts schieben
```

#### Erläuterungen:

End Function

In der Funktion "MCP9801\_lesen" wird der Temperatursensor nach dem I2cstart-Befehl durch das Schreiben der Schreibadresse des Sensors aktiviert. Durch das Schreiben der Registernummer – hier 0 für das Temperaturregister – wird festgelegt, aus welchem Register nachfolgend ein Wert ausgelesen werden soll.

Nach einem erneuten I2cstart wird durch Schreiben der Leseadresse des Temperatursensors festgelegt, dass Daten vom I<sup>2</sup>C-Slave zum I<sup>2</sup>C-Master übertragen werden sollen. Mit den nachfolgenden Befehlen

I2crbyte Vorkomma , Ack 'Byteweise einlesen

I2crbyte Nachkomma , Nack

werden 2 Bytes beginnend ab Register 0 in die Variablen Vorkomma und Nachkomma eingelesen. Mit Ack bzw. Nack wird festgelegt, ob weitere Werte gelesen werden sollen (Ack) oder nicht (Nack). Die 2 eingelesenen Bytes werden mit Makeint zu einem Integerwert zusammengesetzt, der gemäß Datenblatt des Mcp9801-Sensors durch 7-maliges Rechtsschieben und Teilen durch 2 zum Temperaturwert umgewandelt wird. (Vgl. Seite 15/16 im Datenblatt des MCP9801.)

## RTC – Realtime-Clock

Ein weiteres sehr typisches Anwendungsbeispiel für I<sup>2</sup>C ist die Einbindung einer Realtime-Clock (RTC, Echtzeituhr). Mit einer Realtime-Clock lässt sich quasi die Uhrzeit "aufbewahren". Dazu wird die Uhrzeit (und das Datum) vom Mikrocontroller in die Realtime-Clock geschrieben und dort läuft die Uhrzeit (und das Datum) weiter – auch wenn die Spannungsversorgung zum Mikrocontroller unterbrochen wird. Die Uhrzeit, die inzwischen weitergelaufen ist, wird dann vom Mikrocontroller-Programm aus der Realtime-Clock ausgelesen, wenn der Mikrocontroller wieder mit Spannung versorgt wird. Das Auslesen der Uhrzeit ist auch in regelmäßigen zeitlichen Abständen möglich – wie hier im Beispiel 1x pro Sekunde. Die Spannungsversorgung der Realtime-Clock erfolgt mit einem Goldcap oder mit einer eigenen Batterie – typischerweise mit einer Knopfzellen-Batterie.

ELV bietet unter der Best.-Nr. J5-10 34 13 die Realtime-Clock I2C-RTC an, welche über den I<sup>2</sup>C-Bus angesteuert wird. Die Ansteuerung der I2C-RTC erfolgt wieder nach dem in Bild 2 beschriebenen Schema.

Mit diesem Modul kann man das Hauptprogramm entlasten, weil die Zeit und das Datum in der RTC weiterlaufen, und man kann auch nach Ausfall der Hauptspannungsversorgung wieder die korrekte Uhrzeit abrufen. Als Extrafunktionen bietet das Modul verschiedene zeitliche Interruptmöglichkeiten (0,5 s/1 s/1 min/1 h usw.) sowie 2 Alarmwecker.

Der Anschluss des RTC-Moduls erfolgt über die 2 I<sup>2</sup>C-Leitungen SCL und SDA mit je einem Pull-up-Widerstand sowie eine gemeinsame GND-Leitung (Bild 3). Das Modul benötigt eine 5-V-Spannungsversorgung. In der Beispielschaltung sind weiterhin der INTRA-Pin des RTC-Moduls sowie eine LED inklusive Vorwiderstand und ein Taster mit dem Mikrocontroller verbunden.

Im folgenden BASCOM-Programm wird jede Sekunde die Uhrzeit aus der RTC eingelesen und auf dem LC-Display (der Übersichtlichkeit halber in Bild 3 nicht dargestellt) angezeigt.

Wird beim Start des Programms der Taster gedrückt, wird eine Uhrzeit vom Mikrocontroller in das RTC-Modul geschrieben.

#### ' BASCOM-Programm





Bild 3: RTC-Anschluss (LCD-Anschluss nicht eingezeichnet)

Config Portb.0 = Input 'Taster zum Setzen der Uhrzeit 'Interen Pull-up-Widerstand aktivieren Portb.0 = 1Taster Alias Pinb.0 'Aliasnamen Taster für PINB.0 vergeben 'LED zum Anzeigen des Interruptimpulses Config Portb.2 = Output Portb.2 = 0'LED aus Led Alias Portb.2 'Aliasnamen für Portb.2 vergeben Config Sda = Portc.4
Config Scl = Portc.5 'I2C-Pins definieren I2cinit Dim Stunde As Byte Dim Minute As Byte Dim Sekunde As Byte Dim Stunde bcd As Byte Dim Minute bcd As Byte Dim Sekunde bcd As Byte Const Rtc schreibadresse = &B0110 0100 '&H64=100 I2C-Adresse der RTC Const Rtc\_leseadresse = &B0110\_0101 '&H65=101 I2C-Adresse der RTC Dim Neue\_sekunde As Bit 'RTC initialisieren I2cstart I2cwbyte Rtc schreibadresse 'Schreibadresse RTC 12cwbyte &HEO 'Ab Register &hE = Kontrollregister 1 schreiben 'I2cwbyte &B0000\_0000 'Alles normal **12cwbyte &**B0000\_0011 .. oder Sekundeninterrupt (vgl. Seite 17 im RS5C372A-Datenblatt) 12cwbyte &B0010\_0000 '24-Stunden-Format (vgl. Seite 18 im RS5C372A-Datenblatt) I2cstop If Taster = 0 Then 'Wenn Taster beim Start gedrückt ist ... Stunde = 20 '..dann (hier feste) Uhrzeit IN RTC schreiben Minute = 15 'Uhrzeit in RTC schreiben
'Uhrzeit in RTC schreiben
Stunde\_bcd = Makebcd(stunde)
Minute\_bcd = Makebcd(minute)
Sekunde\_bcd = 0 'In BCD-Format umwandeln I2cstart I2cwbyte Rtc\_schreibadresse
I2cwbyte &H00 'Schreibadresse RTC schreiben 'Ab Adresse 0=Sekunden 12cwbyte Sekunde\_bcd 'Sekunde im BCD-Format 'Minute im BCD-Format 12cwbyte Minute\_bcd 13 'Stunde im BCD-Format 12cwbyte Stunde bcd 12 N honoma-1 . . . I2cstop PWM 11 End If PWM 10 PWM 9 DIGITAL 8 PWM 6 C1s Lcd "RTC" Lowerline Lcd "ELV 5 'In Schleife die Uhrzeit einlesen und anzeigen DA Do 'Uhrzeit aus RTC sekündlich lesen If Neue\_sekunde = 1 Then 01 05 RX Neue\_sekunde = 0тх I2cstart 12cwbyte Rtc schreibadresse 'Schreibadresse RTC 12cwbyte &H00 'Ab Adresse 0 I2cstart I2cwbyte Rtc\_leseadresse 'Leseadresse RTC 'Sekunde im BCD-Format lesen 'Minute im BCD-Format lesen I2crbyte Sekunde bcd , Ack I2crbyte Minute\_bcd , Ack I2crbyte Stunde\_bcd , Nack 'Stunde im BCD-Format lesen I2cstop Stunde = Makedec(stunde\_bcd) 'Von BCD-Format in Dezimal umwandeln Minute = Makedec(minute\_bcd) Sekunde = Makedec(sekunde\_bcd) Locate 1 , 5 If Stunde < 10 Then Lcd "0" : Lcd Stunde ; ":" If Minute < 10 Then Lcd "0" : Lcd Minute ; ":" If Sekunde < 10 Then Lcd "0" : Lcd Sekunde ; ":" 'Uhrzeit auf LCD ausgeben End If Loop End Pcint0\_gruppe\_isr:
If Eingang\_pcint0 = 0 Then Neue\_sekunde = 1Set Led 'LED zur Kontrolle kurz leuchten lassen Waitms 50 Reset Led End If Return

Erläuterungen:

Mit Config Sda, Config Scl und I2cinit wird der I<sup>2</sup>C-Bus für die Verwendung definiert bzw. vorbereitet.

Unter der Überschrift "RTC initialisieren" wird der Sekundeninterrupt eingeschaltet und die RTC auf 24-h-Format eingestellt. Laut Datenblatt des im Modul verbauten RS5C372A (Seiten 15 bis 18) werden dafür Control-Register 1 und Control-Register 2 beschrieben.



Wenn die Bedingung Taster = 0 erfüllt ist, werden die Zeitregister (Register 0 bis Register 2) mit einer Uhrzeit beschrieben. Man könnte hier auch eine Routine zum Einstellen der Uhrzeit per Tasteneingabe programmieren. In der Hauptschleife (DO-LOOP) wird in jeder neuen Sekunde die Uhrzeit aus der RTC gelesen. Das erfolgt wieder nach dem Schema aus Bild 2: Nach Start der I<sup>2</sup>C-Übertragung mit I2cstart und Adressierung des RTC-Bausteins durch Schreiben seiner Schreibadresse wird definiert, dass nachfolgend ab Register 0 gelesen werden soll. Es werden dann die Sekunden-, Minuten und Stundenwerte eingelesen und angezeigt. Zu beachten ist, dass die Werte für die Uhrzeit im BCD-Format gespeichert sind, was beim Schreiben bzw. Lesen zu berücksichtigen ist. In der Routine Pcint0\_gruppe\_isr, die durch das Interruptsignal vom Modul getriggert 1x pro Sekunde aufgerufen wird, wird das Flag Neue\_sekunde gesetzt, welches in der Hauptschleife abgefragt wird.

## Ausblick

Zur Abrundung des Themas I<sup>2</sup>C werden im nächsten ELVjournal eine Echtzeituhr mit DCF-Uhr sowie 2 verschiedene Bewegungssensoren (3D und 6D) mit BASCOM angesprochen.



## Weitere Infos:

- Stefan Hoffmann: Einfacher Einstieg in die Elektronik mit AVR-Mikrocontroller und BASCOM. Systematische Einführung und Nachschlagewerk mit vielen Anregungen. ISBN 978-3-8391-8430-1
- www.bascom-buch.de

www.mcselec.com

- www.atmel.com
- Produktübersicht BASCOM: www.elv.de/bascom.html

Empfohlene Produkte/Bauteile:		BestNr.	Preis
BASCOM-(Demo-)Lizenz von MCS Electronics, www.mcselec.com		-	-
Atmel-AVRISP-mkII-Programmer		J5-10 03 55	€ 39,95
oder myAVR-Board MK2		J5-10 90 00	€ 49,-
Netzteil für myAVR-Board MK2		J5-10 90 01	€ 6,95
ATtiny13		J5-10 03 39	€ 1,95
ATmega8		J5-05 29 71	€ 3,20
ATmega88		J5-10 07 62	€ 3,95
100-nF-Kondensator		J5-10 03 17	€ 0,08
Batteriehalter für 3x Mignon		J5-08 15 30	€ 0,75
Batterieclip für 9-V-Block-Batterie		J5-08 01 28	€ 0,30
BASCOM-Buch		J5-10 90 02	€ 54,-
Experimentier-Board 1202B		J5-07 72 89	€ 12,95
Schaltdraht-Sortiment		J5-05 47 68	€ 5,95
LED-Set		J5-10 63 56	€ 3,95
oder Leuchtdioden		J5-10 66 60	€ 1,65
und Widerstände		J5-10 66 57	€ 1,85
Piezo-Signalgeber		J5-00 73 87	€ 0,95
Mikroschalter und -taster		J5-10 66 67	€ 2,80
LC-Display, 2 x 16 Zeichen		J5-05 41 84	€ 6,95
oder myAVR-LCD-Add-on-			
Pin-Ausrichter		J5-00 84 63	€ 4,95
I <sup>2</sup> C-Flip-Anzeige I2C-FA		J5-10 48 63	€ 8,95
LED-I <sup>2</sup> C-Steuertreiber, 16 Kanäle		J5-09 83 77	€ 12,95
I <sup>2</sup> C-4-Digit-LED-Display I2C-4DLED		J5-10 56 97	€ 16,95
I <sup>2</sup> C-Realtime-Clock I2C-RTC		J5-10 34 13	€ 6,50
Realtime-Clock mit DCF77 RTC-DCF		J5-13 05 41	€ 11,95
3-Achsen-Beschleunigungssensor 3D-BS	Komplettbausatz	J5-09 15 21	€ 6,95
	Fertiggerät	J5-10 48 93	€ 9,95
6-Achsen-Bewegungssensor 6D-BS		J5-13 05 98	€ 21,50
I <sup>2</sup> C-Bus-Displaymodul I2C-LCD		J5-09 92 53	€ 13,95
LED-Bussystem LED-B6		J5-08 53 20	€ 14,95
I <sup>2</sup> C-Kabel		J5-08 56 89	€ 2,95
2-pol. Anschlussleitung passend für Miniatur-Stiftbuchse		J5-07 60 55	€ 1,25
Verbindungskabel 2 Module		J5-08 56 90	€ 2,95
Adapterplatine AP-Si4735		J5-10 34 39	€ 18,95
Intelligentes Schrittmotor-Treibermodul iSMT		J5-09 27 20	€ 24,95
USB-I <sup>2</sup> C-Interface USB-I2C	Komplettbausatz	J5-09 22 55	€ 34,95
	Fertiggerät	J5-08 41 23	€ 24,95

Alle Infos zu den Produkten/Bauteilen finden Sie im Web-Shop