

# Entfernungsmessung mit dem srf02

Das srf12-Modul ist ein Ultraschallsensor, der Entfernungen zwischen 15 cm und 6 m messen kann. Die Kommunikation erfolgt über einen i2c-Bus oder über RS232 (TTL). Zum Betrieb ist eine Quelle mit 5 V erforderlich; die Stromaufnahme beträgt etwa 4 mA.

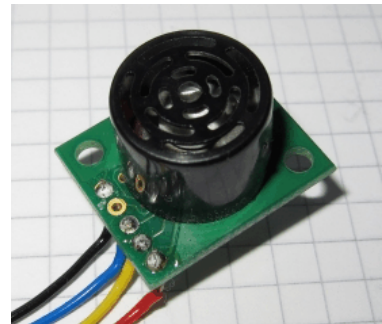


Abbildung 1

An dieser Stelle soll dargestellt werden, wie das srf02-Modul zusammen mit unserer Attiny-Platine Entfernungen misst. Diese sollen in der Einheit cm über die serielle Schnittstelle und ein Terminalprogramm ausgegeben werden.

Um die vielfältigen Aufgaben zu erfüllen, besitzt der Baustein 6 Register, ein Befehlsregister und mehrere Datenregister (Abb. 2). Damit der srf02 einen Befehl ausführt, muss dieser in das Befehlsregister geschrieben werden. Dazu muss man zunächst den Registerzeiger auf die Nummer 0 setzen. Dann überträgt man den Befehlswert; er wird automatisch in dem vom Registerzeiger adressierten Register abgelegt. Sobald der Befehl im Befehlsregister steht, wird er ausgeführt. Mit einem solchen Befehl kann man z. B. eine Entfernungsmessung veranlassen. Will man z. B. einen Entfernungswert in der Einheit cm erhalten, lautet der zugehörige Befehlscode \$50 = 80. Ungefähr 65 ms nach der Übertragung dieses Codes steht der Entfernungswert in den Registern 2 und 3. Wenn die Entfernung kleiner als 255 cm beträgt, reicht es aus, das Lowbyte in Register 3 zu betrachten; ansonsten ergibt sich die Entfernung aus

$$e = 255 * \text{Highbyte} + \text{Lowbyte}.$$

Um z. B. den Wert aus dem Register 3 zu lesen, muss zunächst der Registerzeiger auf 3 gesetzt werden. Anschließend muss ein Lesevorgang durchgeführt werden.

| Registernummer | Rückgabe wenn Register gelesen               | Funktion wenn Register beschrieben wird |
|----------------|--|---|
| 0              | Software Firmware Revision                   | Befehlsregister                         |
| 1              | Ungenutzt<br>(liefert immer den Wert Hex 80) | nicht möglich                           |
| 2              | Entfernungswert<br>(High Byte)               | nicht möglich                           |
| 3              | Entfernungswert<br>(Low Byte)                | nicht möglich                           |
| 4              | Autotune Minimum<br>(High Byte)              | nicht möglich                           |
| 5              | Autotune Minimum<br>(Low Byte)               | nicht möglich                           |

Abbildung 2: Die Register des srf02 (nach: srf02-Manual)

Viele Sensoren arbeiten nach einem ähnlichen Prinzip; insofern ist es ganz lehrreich, die Programmierung des srf02-Moduls genauer zu studieren. In unserem Beispiel (Durchführung einer Entfernungsmessung in cm) soll die Kommunikation zwischen dem Attiny und dem srf02-Modul über den I2C-Bus erfolgen. Die benötigten Schritte sind dann:

- I2C-Start-Befehl und Auswahl/Aktivieren des Bausteins durch Schreiben der Write-Adresse auf I2C-Bus (Diese Adresse ist standardmäßig \$E0.)
- Nach dem Schreiben einer Write-Adresse immer erst Auswahl des Registers durch Schreiben der Register-Nummer; in diesem Fall ist die Registernummer 0.
- Jetzt Befehlscode (\$51 = 81) schreiben; er wandert automatisch in das Befehlsregister.
- I2C-Stopp-Befehl ausführen
- 65 ms warten
- I2C-Start-Befehl und Auswahl/Aktivieren des Bausteins durch Schreiben der Write-Adresse auf I2C-Bus
- Nun Register 2 auswählen
- I2C-Stopp-Befehl ausführen
- I2C-Start-Befehl und Lesevorgang einleiten durch Schreiben der Read-Adresse \$E1 (Die Read-Adresse ist immer um 1 größer als die Write-Adresse.)
- Lesen des Bytes (aus Register 2) mit ACK; durch das ACK-Bit erhöht das srf02-Modul nach dem Lesevorgang den Registerzeiger automatisch um 1.
- Lesen des Bytes (aus Register 3) mit NACK
- I2C-Stopp-Befehl ausführen

Man beachte: Für das Lesen ist erst ein I2C-Stopp-Befehl zu geben und dann die Read-Adresse \$E1 auf den Bus zu schreiben; nach dem Schreiben der Read-Adresse erwartet das srf02-Modul keine Registernummer; vielmehr gibt es bei den nächsten Clock-Signalen den Wert des vom Registerzeiger adressierten Registers aus. Das zugehörige BASCOM-Programm befindet sich am Ende dieses Dokuments.

| Befehlscode |      | Funktion   |
|-------------|------|--|
| Decimal     | Hex  |  |
| 80          | 0x50 | Startet Messvorgang (gemessen wird in der Einheit inches)  |
| 81          | 0x51 | Startet Messvorgang (gemessen wird in der Einheit Zentimeter)  |
| 82          | 0x52 | Startet Messvorgang (gemessen wird in der Einheit Microsekunden)   |
| 86          | 0x56 | Synchron Messvorgang (gemessen wird in der Einheit inches)   |
| 87          | 0x57 | Synchron Messvorgang (gemessen wird in der Einheit Zentimeter)   |
| 88          | 0x58 | Synchron Messvorgang (gemessen wird in der Einheit Microsekunden)  |
| 92          | 0x5C | Erzeugt einen 8 zyklischen 40khz Impuls/Ton  |
| 96          | 0x60 | Startet automatische Kalibrierung – Befehl kann ignoriert werden da dies bereits beim einschalten (Power on) automatisch erfolgt |
| 160         | 0xA0 | Erste Sequence zum ändern der I2C-Bus Slave ID   |
| 165         | 0xA5 | Dritte Sequence zum ändern der I2C-Bus Slave ID  |
| 170         | 0xAA | Zweite Sequence zum ändern der I2C-Bus Slave ID  |

**Abbildung 3**

Neben dem schon angegebenen Befehl zur Entfernungsmessung gibt es Reihe weiterer Befehle. Damit kann z. B. eine Entfernungsmessung in inch oder eine Veränderung der srf02-Adresse vorgenommen werden (Abb. 3).

```

$regfile = "attiny2313.dat"                'Attiny2313
$crystal = 4000000                        '4 MHz
$baud = 9600
$framesize = 10

'*****
'***** Deklarationen *****
Declare Function Entfernungsmessen(byval Awrite As Byte) As Byte
Dim Aadr0 As Byte
Dim Entfernung As Byte
Dim Awrite As Byte

'*****
'***** Initialisierung *****
Ddrb = &B11111111                          'Port B als Ausgangsport
Ddrd = &B01110000                          'D4, D5, D6 als Ausgang; Rest als Eingang
Portd = &B10001111                          'Eingänge auf high legen
Adr0 = &HE0                                  'Standard-Adresse im write-Modus
Config Scl = Portb.7
Config Sda = Portb.5

'*****
'***** Hauptprogramm *****
Do
  Entfernung = Entfernungsmessen(adr0)
  Printbin Entfernung
  Waitms 900
Loop

'*****
'***** Unterprogramme *****
Sub Entfernungsmessen(byval Awrite As Byte , E As Byte)
Local Lob As Byte
Local Hib As Byte
Local Aread As Byte
Local Gefunden As Byte
Printbin Awrite
Aread = Awrite + 1                          'Adresse im read-Modus
'Adressieren (write)
  I2cstart
  I2cwbyte Awrite
  Gefunden = Not Err                        'Err = 1 wenn nicht gefunden (s. u.)
'Einstellen
  I2cwbyte 0                                'Befehlsregister (0) auswählen
  I2cwbyte 81                               'startet damit Messvorgang in Zentimetern
  I2cstop
  Portb.0 = Gefunden                       'LED an PortB blinkt auf während Messung
  Waitms 100                               'minimale Wartezeit für Messung ist 65 ms
  Portb.0 = 0
'Adressieren (write)
  I2cstart
  I2cwbyte Awrite
'Einstellen
  I2cwbyte 2                                'Leseregister (2) auswählen (High-Byte)
  I2cstop
'Adressieren (read)
  I2cstart
  I2cwbyte Aread
'Lesen (erst Reg. 2 und dann Reg. 3)
  I2crbyte Hib , Ack                        'Nach Ack automatisch nächstes Register ansteuern
  I2crbyte Lob , Nack                       'Enden des Lesevorgangs mit NACK
  I2cstop
  Entfernungsmessen = Lob                   'Bei Messungen unter 2,55 m nur Lob relevant
End Sub

```