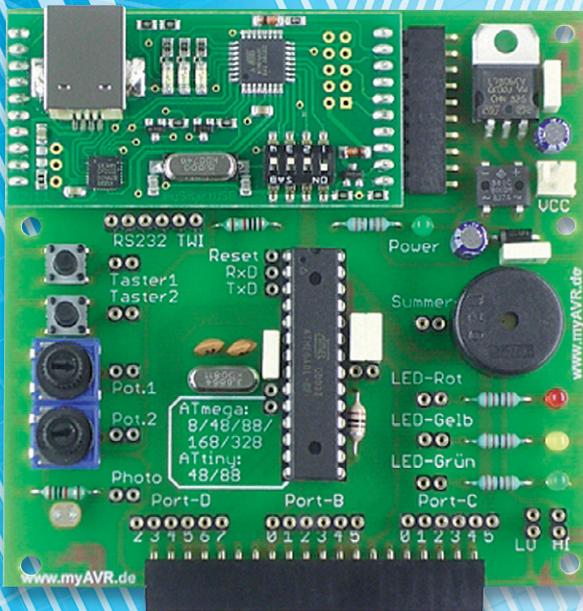




# Mikrocontroller-Einstieg

Teil 9: Serielle (UART-)Datenübertragung: Empfang



```
BASCOM-AVR IDE [2.0.7.5] - [C:\user\BASCOM-Programme\Blinker_attiny13.bas]
Datei Editieren Anzeigen Programmieren Werkzeuge Optionen Fenster Hilfe
Blinker_attiny13.bas
Sub
  BASCOM-Programm
  Einfacher Blinker
  In: -
  Out: LED mit Vorwiderstand an Portb.4

$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 4
$swstack = 4
$framesize = 10

Config PORTB.4 = Output

Do
  PORTB.4 = 1
  Waitms 500
  PORTB.4 = 0
  Waitms 500
Loop
End |

'Verwendeter Chip
'Verwendete Frequenz
'Rücksprungadressen (je 2), Registersicherungen (32)
'Parameterübergaben (je 2), LOCALs (je 2)
'Parameter (Daten-Laenge), Rechenbereich Funktionen
'B.4 als Ausgang definieren

'Schleifenbeginn
'B.4 auf 1
'Warteschleife 500 ms
'B.4 auf 0
'Warteschleife 500 ms
'Schleifenende
'Programmende
```



# mit **BASCOM-AVR**

Nachdem in den Teilen 7 und 8 der Artikelserie beschrieben wurde, wie Daten vom Mikrocontroller über die serielle Schnittstelle gesendet werden, geht es in diesem Artikel um das Empfangen von Daten.

## Empfangsmethoden

BASCOM stellt verschiedene Befehle für seriellen Datenempfang zur Verfügung, mit denen unterschiedliche Empfangsmethoden realisiert werden können. Die am leichtesten zu verstehende Methode ist der Empfang mit INPUT, wobei eine Zeichenfolge bzw. eine Zahl und ein Abschlusszeichen von einem Sender erwartet werden. Soll nur auf ein einziges Zeichen gewartet werden, stellt BASCOM den Befehl WAITKEY bereit. Der Befehl INKEY hält das Programm nicht an, sondern liest „im Vorbeiflug“ ein Zeichen von der seriellen Schnittstelle ein. Mit einem manuell programmierten Interrupt-Handler oder mit CONFIG SERIALIN kann man sehr elegant quasi nebenbei Daten seriell empfangen. In diesem und dem folgenden Teil der Artikelserie „Mikrocontroller-Einstieg mit BASCOM-AVR“ werden die verschiedenen Methoden für serielles Empfangen mit BASCOM anhand von Beispielen erläutert.

## Vom PC empfangen mit INPUT bzw. INPUTBIN

Sollen Daten von einem PC oder anderen Geräten empfangen werden, bietet der BASCOM-Befehl INPUT eine gute Möglichkeit, Daten seriell zu empfangen. Das Programm wartet beim INPUT-Befehl, bis Daten sowie Abschlusszeichen – je nach Konfiguration CR, LF, CRLF oder LFCR – empfangen wurden. Standardmäßig wird auf die Zeichenfolge bzw. Zahl gefolgt von CR (ASCII-Code 13) gewartet. CR (Carriage Return = Wagenrücklauf) bzw. LF (Line Feed = Zeilenschaltung) sind unsichtbare (Steuer-)Zeichen mit den ASCII-Codes 13 bzw. 10.

**Achtung:** Es muss sichergestellt sein, dass der Sender genau die erwartete Zeichenfolge sendet, weil das Programm sonst ewig wartet und nicht fortgesetzt wird. Bei einem Terminalprogramm oder einem selbst erstellten Programm auf dem PC kann man selbst bestimmen, was gesendet wird. Bei fertigen Modulen muss man im jeweiligen Handbuch nachlesen, in welcher Kombination CR und LF gesendet werden und das BASCOM-Programm darauf mit CONFIG INPUT einstellen. Beim reinen Zahlenempfang mit INPUTBIN wird nicht auf CR und/oder LF gewartet, sondern es wird gewartet, bis die der Eingabevariablen entsprechende Anzahl von Bytes empfangen wurde.

```

' BASCOM-Programm
'
' Serieller Empfang vom PC (über USB-UART-Umsetzer UM2102 o. Ä.)
' Mit INPUT
' Am PC senden mit BASCOM-Terminal, HTerm oder anderem Programm
'
' In: Serielles Signal an Pin  d.0  = Rxd
' Out: LEDs mit Vorwiderständen an B.1 (rot), B.2 (gelb)
$regfile = „M88def.dat“           'Verwendeter Chip
$crystal = 3686400                 'Verwendete Frequenz. Fuse-Bits: Externer Quarz.
$hwstack = 40                     'Rücksprungadressen (je 2), Registersicherungen (32)
$swstack = 40                     'Parameteruebergaben (je 2), LOCALs (je 2)
$framesize = 60                   'Parameter (Daten-Laenge), Rechenbereich Funktionen

$baud = 9600                       '9600 Bit pro Sekunde, 8 Datenbit, Keine Parität, 1 Stoppbit

```



```

Led_rot Alias Portb.1
Config Led_rot = Output
Led_gelb Alias Portb.2
Config Led_gelb = Output
Dim Eingabe As String * 5           'Eingabezeichenkette

Print "ELVjournal"
Print "Mikrocontroller-Einstieg ";   'Semikolon unterdrückt CRLF
Print "mit BASCOM-AVR"

Do
Print
Print "Eingabe Schaltbefehl (rot bzw. gelb) dann Enter"

Input Eingabe
Select Case Eingabe
    Case "rot" : Toggle Led_rot
    Case "gelb" : Toggle Led_gelb
    Case Else : Print "Eingabe unbekannt: " ; Eingabe
End Select
Loop
End

```

### Erläuterungen:

Mit \$baud wird zunächst die gleiche Datenübertragungsgeschwindigkeit wie beim Sender eingestellt. Beim Befehl INPUT – Eingabe – wartet (!) das Programm, bis die Eingabezeichenkette und das abschließende CR (ASCII 13) empfangen wird. Je nach Inhalt des Eingabestrings wird die rote bzw. die gelbe LED umgeschaltet. Wird nicht genau „rot“ oder „gelb“ empfangen, wird mit PRINT eine entsprechende Meldung ausgegeben, wie in [Bild 1](#) zu sehen.

### Vom PC empfangen mit WAITKEY

Bei INPUT wird auf Eingabezeichen und Abschlusszeichen (CR, LF, CRLF, LFCR) gewartet. Wenn nur ein einzelnes Zeichen empfangen und weiterverarbeitet werden soll, dann kann der BASCOM-Befehl WAITKEY eingesetzt werden.

```

' BASCOM-Programm
'
' Serieller Empfang vom PC (über USB-UART-Umsetzer UM2102 o. Ä.)
' Mit WAITKEY
' Am PC senden mit BASCOM-Terminal, HTerm oder anderem Programm
'
' In: Seriellles Signal an Pin  d.0  = Rxd
' Out: LEDs mit Vorwiderständen an B.1 (rot), B.2 (gelb)
$regfile = „M88def.dat“           'Verwendeter Chip
$crystal = 3686400                'Verwendete Frequenz. Fuse-Bits: Externer Quarz.
$hwstack = 40                    'Rücksprungadressen (je 2), Registersicherungen (32)
$swstack = 40                    'Parameteruebergaben (je 2), LOCALs (je 2)
$framesize = 60                  'Parameter (Daten-Laenge), Rechenbereich Funktionen

$baud = 9600                      '9600 Bit pro Sekunde, 8 Datenbit, Keine Parität, 1 Stoppbit

Led_rot Alias Portb.1
Config Led_rot = Output
Led_gelb Alias Portb.2
Config Led_gelb = Output
Dim Eingabe As String * 1         'Eingabezeichenkette

Print "ELVjournal"
Print "Mikrocontroller-Einstieg ";   'Semikolon unterdrückt CRLF
Print "mit BASCOM-AVR"

```

Bild 1: BASCOM-Terminal – Eingabe mit INPUT



```

Do
Print
Print "Eingabe Schaltbefehl (r bzw. g)"
Eingabe = Waitkey()           'Wartet auf Zeichen
Select Case Eingabe
  Case "r" : Toggle Led_rot
  Case "g" : Toggle Led_gelb
  Case Else : Print "Eingabe unbekannt: " ; Eingabe
End Select
Loop
End

```

#### Erläuterungen:

Dort wo im Programm „Eingabe = Waitkey()“ steht, hält das Programm an, bis ein einzelnes Zeichen empfangen wird, welches dann der Variablen „Eingabe“ zugewiesen wird. Der geänderte Ablauf ist in [Bild 2](#) zu sehen.

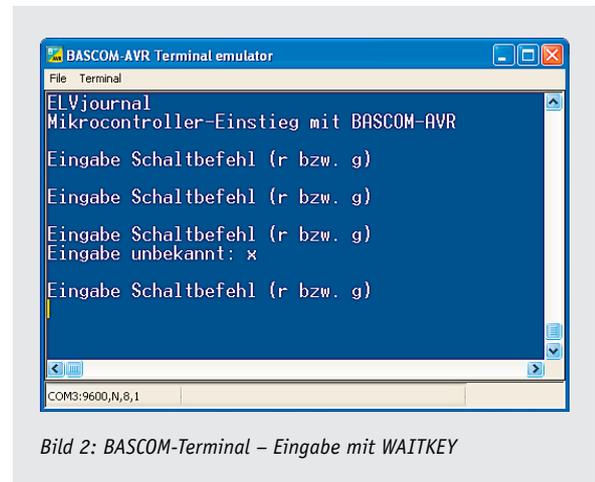


Bild 2: BASCOM-Terminal – Eingabe mit WAITKEY

### Vom PC empfangen mit INKEY

Bei INPUT/INPUTBIN und WAITKEY ist der Nachteil, dass das Programm anhält, bis ein Empfang stattgefunden hat. Oftmals ist aber nicht genau vorhersehbar, wann ein Empfang stattfindet, und man möchte daher das Programm nicht anhalten, sondern andere Aufgaben ausführen und „nebenbei“ prüfen lassen, ob Zeichen bzw. Zahlen empfangen wurden. Bei Verwendung des BASCOM-Befehls INKEY hält das Programm nicht an, sondern liest „im Vorbeiflug“ ein Zeichen aus dem Eingabepuffer.

```

' BASCOM-Programm
'
' Serieller Empfang vom PC (über USB-UART-Umsetzer UM2102 o. Ä.)
' Mit INKEY
' Am PC senden mit BASCOM-Terminal, HTerm oder anderem Programm
'
' In: Serielles Signal an Pin d.0 = Rxd
' Out: LEDs mit Vorwiderständen an B.1 (rot), B.2 (gelb)

$regfile = „M88def.dat“           'Verwendeter Chip
$crystal = 3686400                'Verwendete Frequenz. Fuse-Bits: Externer Quarz.
$hwstack = 40                    'Rücksprungadressen (je 2), Registersicherungen (32)
$swstack = 40                    'Parameteruebergaben (je 2), LOCALs (je 2)
$framesize = 60                  'Parameter (Daten-Laenge), Rechenbereich Funktionen
$baud = 9600                     '9600 Bit pro Sekunde, 8 Datenbit, Keine Parität, 1 Stoppbit

Led_rot Alias Portb.1
Config Led_rot = Output
Led_gelb Alias Portb.2
Config Led_gelb = Output
Dim Eingabe As String * 1        'Eingabezeichenkette

Print „Eingabe Schaltbefehl (r bzw. g)“

Do
'If Ischarwaiting() = 1 Then     'prüft, ob Eingabe vorliegt
  Eingabe = Inkey()             'Wartet nicht
  Select Case Eingabe
    Case „r“ : Toggle Led_rot
    Case „g“ : Toggle Led_gelb
    Case Else : Print „Eingabe unbekannt: „ ; Eingabe
  End Select
Print
Print „Eingabe Schaltbefehl (r bzw. g)“
'End If
Waitms 100
Loop
End

```



### Erläuterungen:

Sobald die Programmzeile „Eingabe = Inkey()“ erreicht wird, wird ein Zeichen eingelesen. Ist das Zeichen ein „r“, wird die rote LED umgeschaltet, und bei einem „g“ wird die gelbe LED umgeschaltet. Wenn kein neues Zeichen empfangen wurde, gibt INKEY eine 0 zurück. Wenn man auch eine 0 als Eingabezeichen erwartet, kann man das mit INKEY nicht vom Nicht-Empfang unterscheiden. Man kann durch eine vorangestellte Abfrage mit ISCHARWAITING abfragen, ob tatsächlich ein Zeichen empfangen wurde. Die Funktion ISCHARWAITING gibt eine 1 zurück, wenn etwas im seriellen Puffer steht.

### GPS-Empfänger

Jeder kennt und nutzt GPS-Empfänger in Outdoor-GPS-Empfängern, Navigationsgeräten oder Mobiltelefonen. Das Prinzip bei GPS (Global Positioning System) ist, dass Satelliten, die die Erde umkreisen, extrem präzise Signale senden, welche mit Empfängern empfangen und per PC oder Mikrocontroller ausgewertet werden. ELV bietet das GPS-Empfangsmodul NL-552ETTL an (Best.-Nr. J2-09 42 41), mit dessen Hilfe die GPS-Signale empfangen und in einem standardisierten Format (NMEA) [1] bereitgestellt werden und mit dem Mikrocontroller verarbeitet werden können.

Das Modul ist gemäß Bild 3 durch die Datenleitung und die gemeinsame GND-Leitung mit dem Mikrocontroller zu verbinden. Es wird mit 5 V Spannung versorgt. Sobald die Spannung am Modul anliegt, leuchtet eine grüne LED dauernd auf, was signalisiert, dass das Modul nach Satelliten sucht. Wenn die GPS-Daten komplett empfangen wurden, dann blinkt die grüne LED. Das Modul sollte nicht direkt neben PCs oder anderen Störquellen betrieben werden. Das Modul sollte möglichst „Sichtkontakt“ zu den Satelliten – also zum Himmel – haben. Ein Empfang in Gebäuden ist in Fensternähe möglich.

```
' BASCOM-Programm
'
' GPS-Empfang mit ELV-GPS-Modul Navilock NL-552ETTL

' In: Seriellles Signal von GPS an D.0
' Out: LCD Portd.2 bis Portd.7

'GPS-Signal (Auszug):
' „$GPRMC,hhmmss.sss,A,bbbb.bbbb,N,lllll.llll,E,x.xx,xxx.xx,ddmmjj,,*xx“
' Header Uhrzeit Status Breite Länge Datum
'Beispiel:
' „$GPRMC,092241.000,A,5433.0463,N,00959.5777,E,0.44,357.31,250314,,*01“
```

```
$regfile = „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
$baud = 38400                     '38500 Bit pro Sekunde, 8 Datenbit, Keine Parität, 1 Stoppbit
```

```
Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2
```

```
Config Lcd = 16 * 2
```

```
Cls
```

```
Cursor Off
```

```
Dim Gelesen As Byte               'eingelesenes Zeichen
```

```
Dim Teilstring(14) As String * 12
```

```
Dim Anzahl As Byte
```

```
Dim Gpsdaten As String * 98       'Rohdaten im NMEA-Format
```

```
Dim Header As String * 6          'Kopf=Header der NMEA-Daten. Zum Beispiel GPRMC
```

```
'-----Begrueessung-----
```

```
Cls
```

```
Cursor Off
```

```
Lcd "GPS mit NL-552ETTL "
```

```
Wait 4
```

```
Cls
```

```
Do
```

```
Gelesen = Inkey()
```

```
If Gelesen = 36 Then Input Gpsdaten '36 = $
```

```
Header = Mid(Gpsdaten , 1 , 5)      'Datensatz aus Gesamtstring lesen
```

```
If Header = "GPRMC" Then           'Wenn GPRMS-Datensatz
```

```
Cls
```

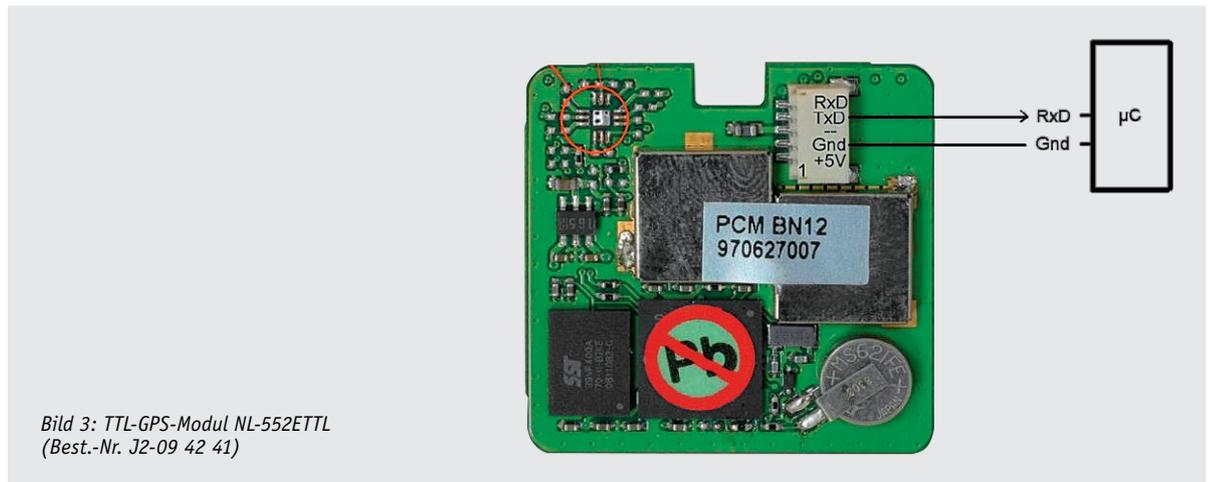


Bild 3: TTL-GPS-Modul NL-552ETTL  
(Best.-Nr. J2-09 42 41)

```

Anzahl = Split(gpsdaten , Teilstring(1) , ",")           'Gesamten Datensatz an Kommas aufsplitten
Gpsdaten = ""                                           'Variable für Rohdaten leeren

'Uhrzeit:
Locate 1 , 1
Lcd Left(teilstring(2) , 2) ; ":" ; Mid(teilstring(2) , 3 , 2) ; ":" ; Mid(teilstring(2) , 5 , 2)
Lcd " UTC"                                             'Ausgabe der UTC-Zeit

'Datum:
Locate 2 , 1
Lcd Left(teilstring(10) , 2) ; ":" ; Mid(teilstring(10) , 3 , 2) ; ":" ; "20" ; Mid(teilstring(10) , 5 , 2)

Wait 4                                                 '4 Sekunden die Daten anzeigen
Cls                                                    'dann LCD löschen für Anzeige weiterer Daten

'Breite:
Locate 1 , 2
Lcd Left(teilstring(4) , 2) : Lcd Chr(223)             'Grad u. Gradzeichen
Lcd Mid(teilstring(4) , 3 , 6)                        'Rest
Lcd " " ; Teilstring(5)                               'N/S

'Länge:
Locate 2 , 1
Lcd Left(teilstring(6) , 3) : Lcd Chr(223)             'Grad u. Gradzeichen
Lcd Mid(teilstring(6) , 4 , 6)                        'Rest
Lcd " " ; Left(teilstring(7) , 1)                     'E/W

'Status:
Locate 1 , 16
Lcd Teilstring(3)                                     'Status. A=active V=void=ungültig
Wait 5
End If
Loop
End

```

#### Erläuterungen:

Es werden mit INKEY „im Vorbeiflug“ (also ohne dass das Programm halten muss) Zeichen von der seriellen Schnittstelle eingelesen. Wenn das eingelesene Zeichen ein „\$“ ist, werden mit INPUT alle Zeichen eines GPS-Datensatzes eingelesen, welcher im Anschluss ausgewertet wird. Zunächst werden die ersten 5 Zeichen mit der MID-Funktion extrahiert und es wird geprüft, ob dieser Teil des gesamten Datensatzes „GPRMC“ ist und damit der Datensatzart entspricht, die hier ausgewertet werden soll. Die BASCOM-Funktion SPLIT ermöglicht die elegante Möglichkeit, den langen Datensatz an den Kommas zu trennen und die Teile dem Array Teilstring zuzuweisen. Im Weiteren kann dann direkt auf diese jeweiligen Arrayelemente zugegriffen werden und diese können wie hier angezeigt werden oder sie könnten zur Weiterverarbeitung verwendet werden. So könnte man zum Beispiel bei Status „Void“ die Anzeige der Koordinaten unterdrücken oder eine Meldung anzeigen.



## Ausblick

Nachdem in diesem Teil der Artikelserie „Mikrocontroller-Einstieg mit BASCOM-AVR“ die grundlegenden seriellen Empfangsmethoden dargestellt und in kleinen Versuchen sowie einem GPS-Beispiel zur Anwendung kamen, werden im folgenden Teil Me-

thoden für seriellen Empfang mit Interrupts beschrieben werden. Dabei wird ein Praxisbeispiel die Einbindung des ELV-Moduls FS20 WUE (Best.-Nr. J2-10 38 66) in eigene BASCOM-Anwendungen zeigen. Mit diesem Modul lassen sich die Daten von ELV-Wettersensoren und die Daten von FS20-Sendern empfangen und dann mit Hilfe von BASCOM beliebig für eigene Zwecke auswerten. **ELV**

[1] Ein NMEA-Datenstrom besteht aus seriell gesendeten Textzeichen. Die Daten werden in sogenannten Sätzen übertragen, wobei jeder Satz am Anfang ein Dollarzeichen (\$) und am Ende die ASCII-Steuerzeichen CR (ASCII 13) und LF (ASCII 10) hat. Die ersten Textzeichen nach dem Dollarzeichen geben Auskunft darüber, von wel-

cher Art von Gerät der Datensatz kommt. Beispiel: GP steht für einen Datensatz von einem GPS-Empfänger. Im Anschluss wird durch eine Buchstabenkennung angezeigt, welche Informationen von dem Gerät kommen. Die eigentlichen Daten werden getrennt durch Kommas in einer festgelegten Reihenfolge übertragen. Beispiel: „\$GPRMC,092241.000,A,5433.0463,N,00959.5777,E,0.44,357.31,250314,,\*01“ Nähere Informationen siehe: [www.kowoma.de/gps](http://www.kowoma.de/gps)



## 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](http://www.bascom-buch.de)
- [www.mcselec.com](http://www.mcselec.com)
- [www.atmel.com](http://www.atmel.com)
- Produktübersicht Bascom: [www.elv.de/bascom.html](http://www.elv.de/bascom.html)

### Empfohlene Produkte/Bauteile:

	Best.-Nr.	Preis
BASCOM-(Demo-)Lizenz von MCS Electronics <a href="http://www.mcselec.com">www.mcselec.com</a>	-	-
Atmel-AVRISP-mkII-Programmer	J2-10 03 55	€ 39,95
<b>oder</b> myAVR-Board MK2	J2-10 90 00	€ 49,-
Netzteil für myAVR-Board MK2	J2-10 90 01	€ 6,95
ATmega88	J2-10 07 62	€ 3,95
100-nF-Kondensator	J2-10 03 17	€ 0,08
Batteriehalter für 3x Mignon	J2-08 15 30	€ 0,75
Batterieclip für 9-V-Block-Batterie	J2-08 01 28	€ 0,30
BASCOM-Buch	J2-10 90 02	€ 54,-
Experimentier-Board 1202B	J2-07 72 89	€ 12,95
Schaltdraht-Sortiment	J2-05 47 68	€ 5,95
LED-Set	J2-10 63 56	€ 3,95
<b>oder</b> Leuchtdioden und Widerstände	J2-10 66 60 J2-10 66 57	€ 1,65 € 1,85
Piezo-Signalgeber	J2-00 73 87	€ 0,95
Mikroschalter und -taster	J2-10 66 67	€ 2,80
LC-Display, 2x 16 Zeichen	J2-05 41 84	€ 6,95
<b>oder</b> myAVR-LCD-Add-on		
Pin-Ausrichter	J2-00 84 63	€ 4,95
USB-UART-Umsetzer UM2102	J2-09 18 59	€ 5,95
USB-UART-Umsetzer mit Optokopplung U02102	J2-10 49 66	€ 12,95
Real-Time-Clock mit DCF77 RTC-DCF	J2-13 05 41	€ 11,95
FS20- und Wetterdaten-Empfänger FS20 WUE	J2-10 38 66	€ 14,95
Funk-Temperatursensor S300IA	J2-07 36 06	€ 39,95
Funk-Außensensor ASH2200-1	J2-07 36 05	€ 29,95
GPS-Empfangsmodul NL-552ETTL	J2-09 42 41	€ 34,95
USB-Stick-Interface STI 100	J2-07 59 50	€ 27,95
FS20-Diagnosetool FS20 DT	J2-06 68 13	€ 59,95

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

Preisstellung Februar 2014 – aktuelle Preise  
im Web-Shop