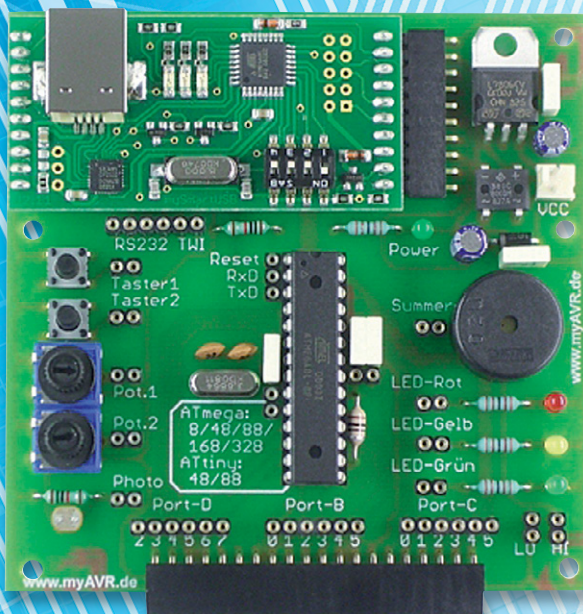




Mikrocontroller-Einstieg

Teil 8: Serielle (UART-)Datenübertragung: weitere Anwendungen



```
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

  Sregfile = "attiny13.dat"
  Socrystal = 1200000
  Ssystack = 4
  Ssvstack = 4
  Sframesize = 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)
'Parameteruebergaben (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
```



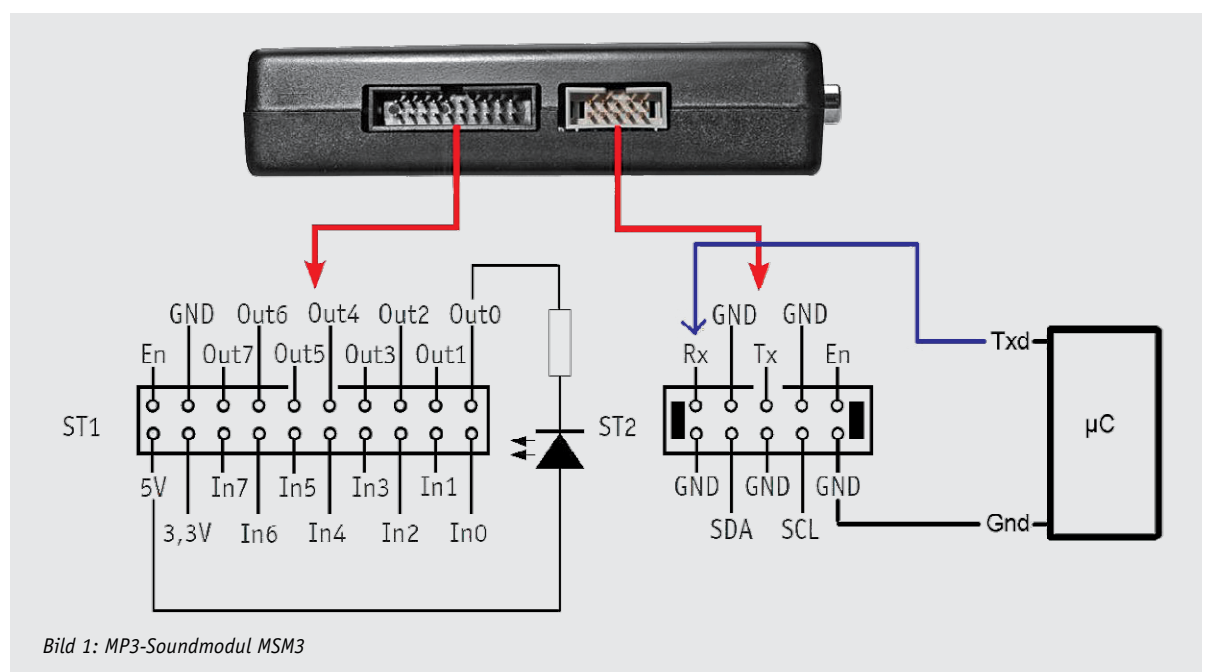
mit **BASCOM-AVR**

Die serielle UART-Schnittstelle eignet sich hervorragend, um Sensoren oder andere Produkte in eigene Anwendungen einzubinden. Nachdem immer wieder ELV-Leser den Wunsch nach BASCOM-Beispielen für die Ansteuerung von ELV-Produkten geäußert haben und im Teil 7 unserer Artikelserie „Mikrocontroller-Einstieg mit BASCOM-AVR“ die Grundlagen der seriellen Datenübertragung und die Ansteuerung des ELV FS20-Senders FS20 US beschrieben wurden, geht es nun um die Einbindung des ELV Soundmoduls MSM3 sowie um eine einfache, aber sehr wirkungsvolle Möglichkeit, eine eigene Lichtshow mit DMX-Spots zu realisieren.

MP3-Soundmodul MSM3 ansteuern

Unter dem Produktnamen MSM3 bietet ELV ein sehr mächtiges MP3-Soundmodul an, bei dem sich Tausende von Sounddateien auf einer handelsüblichen SD-Speicherkarte ablegen und im Rahmen eigener Projekte mittels einfacher serieller Befehle abspielen lassen. Anwendungsbeispiele finden sich im Modell-

bau, bei diversen Ansage-Situationen wie in Museen oder Ausstellungen oder bei der akustischen Ausgabe von Werten oder Stati in Laborumgebungen. Der Hardwareaufwand ist sehr gering: Der TxD-Pin des Mikrocontrollers wird mit dem RxD-Pin des Soundmoduls verbunden ([Bild 1](#)). Desweiteren benötigt man die Gnd-Verbindung zwischen Mikrocontroller und





Soundmodul, eine Versorgungsspannung (8 bis 16 V) sowie einen Lautsprecher der direkt an den im Soundmodul integrierten Verstärker angeschlossen wird. Außer der Ansteuerung der Sounddateien ist auch das Schalten von 8 Open-Drain-Kanälen möglich. Die Ansteuerung erfolgt seriell mit einer Baudrate von 9600 Baud, weshalb auch der ansteuernde Mikrocontroller auf genau diese Baudrate eingestellt werden muss.

```
' BASCOM-Programm
,
' Serielle Ausgabe an MSM3 (MP3-Soundmodul)
' Kommandos von AVR zum MSM3 senden
,
' In: Taster an B.0 und B.1
' Out: Serielles Signal an Pin d.1 = Txd
' Out: LED an Portb.2

$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 Bits pro Sekunde, 8 Datenbits, Keine Parität, 1 Stoppbit

Config Portb.0 = Input             'Taster1
Portb.0 = 1                       'Pullup-Widerstand
Taster1 Alias Pinb.0

Config Portb.1 = Input             'Taster2
Portb.1 = 1                       'Pullup-Widerstand
Taster2 Alias Pinb.1

Config Portb.2 = Output            'LED
Led Alias Portb.2

Const Startzeichen = &H02         'Das Startzeichen ist immer &h02 (hexadezimal 2)
Const Abschlusszeichen = &H03     'Das Abschlusszeichen ist immer &h03 (hexadezimal 3)

Dim Msm3befehl As String * 1      'Variable für MSM3-Befehl („F“, „P“ usw.)
Dim Msm3daten As Word             'Variable für MSM-Daten (Dateinamen bzw. Parameter)
Dim Zahl As Word

' Von 1 bis 5 zählen:
Msm3befehl = „F“
For Zahl = 1 To 5
    Printbin Startzeichen ;
    Print Msm3befehl ; Zahl;      ' „F1“ bis „F5“ PRINT wandelt Zahl zu Zeichen.
    Printbin Abschlusszeichen;
    Wait 2
Next

'Verschiedene Möglichkeiten:
Printbin &H02 ; : Print „F“ ; 20 ; : Printbin &H03 ; : Wait 3           'Sound 20

Printbin &H02 ; : Print „F“ ; „911“ ; : Printbin &H03 ; : Wait 3       'Sound 911

Printbin &H02 ; : Print „P1“ ; : Printbin &H03;                        'Playlist anspielen

Do

If Taster1 = 0 Then               'Wenn Taster1 gedrueckt..
    Led = 1                       'Kontroll-LED an
    Msm3befehl = „F“
```





```

Msm3daten = 1001
Printbin Startzeichen ;
Print Msm3befehl ; Msm3daten;
' Print „F1001“; ,Alternativschreibweise
Printbin Abschlusszeichen;
Waitms 500 : Led = 0
End If

If Taster2 = 0 Then
    Led = 1
    Printbin Startzeichen;
    Print „S1“;
    Printbin Abschlusszeichen;

    Printbin Startzeichen;
    Print „0255255“;
    Printbin Abschlusszeichen;

    Wait 1 : Led = 0

    Printbin Startzeichen;
    Print „0000255“;
    Printbin Abschlusszeichen;
    Waitms 500
End If

Loop
End

```

'Sounddatei abspielen
'Msm3befehl ; Dateinummer
'Kontroll-LED aus
'Wenn Taster2 gedrueckt ..
'Kontroll-LED an
'Outputmode manuell
'Alle Ausgänge an
'Kontroll-LED aus
'Alle Ausgänge aus

Erläuterungen:

Mit \$BAUD wird im BASCOM-Programm die Baudrate eingestellt. Laut Protokoll des Soundmoduls muss zunächst ein Startzeichen (&h02) gesendet werden. Danach der Befehl an das Soundmodul gemäß Tabelle 2 im Datenblatt. „F“ steht für das Abspielen einer Sounddatei. Mit „P“ wird eine Playlist abgespielt. Mit „O“ werden die Ausgänge angesteuert. Im vorstehenden BASCOM-Programm werden Start- und Abschlusszeichen als Konstanten definiert, um die Lesbarkeit des Programms im Weiteren zu verbessern. In der FOR-NEXT-Schleife werden die Ziffern 1 bis 5 „vorgelesen“, indem die zugeordneten Sounddateien angesteuert werden. Es wird jeweils erst das Startzeichen &h02 als Zahl mit PRINTBIN gesendet. Dann der Befehl „F“ als ASCII-Zeichen mit PRINT sowie die Zahl der Sounddatei gefolgt vom Abschlusszeichen.

Nach der FOR-NEXT-Schleife werden die Sounddateien 20 bzw. 911 sowie eine Playlist abgerufen.

In der Hauptschleife (DO-LOOP) werden zwei Taster abgefragt. Wird Taster 1 gedrückt, dann wird die Sounddatei mit der Nummer 1001 abgespielt. Wird Taster 2 gedrückt, werden die Ausgänge des MSM3-Moduls eingeschaltet und nach einer Wartezeit von einer Sekunde wieder ausgeschaltet.

DMX-Ansteuerung von LED-Spots

DMX ist ein Übertragungsprotokoll, das vor allem in der Bühnen- und Veranstaltungstechnik benutzt wird, um Lichtanlagen und andere Geräte über einen Bus zu steuern. Es handelt sich um ein asynchrones serielles Protokoll, welches physikalisch auf RS485 basiert. Bei RS485 werden nicht wie bei UART (CMOS/TTL-Pegel) und RS232 positive bzw. negative Spannungspegel gegenüber einem Bezugspotential genutzt, sondern es wird ein symmetrisches Übertragungsverfahren angewendet, bei dem die Spannungsdifferenz zwischen den beiden Datenleitungen A und B für die Darstellung der Logikpegel eins und null verwendet wird. Dadurch ist die Datenübertragung robuster gegen Störimpulse. Um einen Mikrocontroller als Sender mit DMX-Empfängern zu verbinden, muss daher ein Pegelwandler zwischengeschaltet werden, der die UART-Spannungspegel auf die RS485-Spannungspegel umsetzt (Bild 2). Als Pegelwandler eignet sich ein SN75176 oder ein MAX485. Es lässt sich dann mit BASCOM mit wenig Aufwand ein DMX-Sender erstellen, mit dem DMX-Geräte angesteuert werden können.

Eine Lichtsteuerung für Veranstaltungen lässt sich gut durch einen oder mehrere ELV DMX-Spots (Best.-Nr. JQ-10 25 14) realisieren. Mehrere DMX-Empfänger können hintereinander geschaltet werden. Am Ende sollte der Bus durch einen 120-Ω-Widerstand terminiert werden.

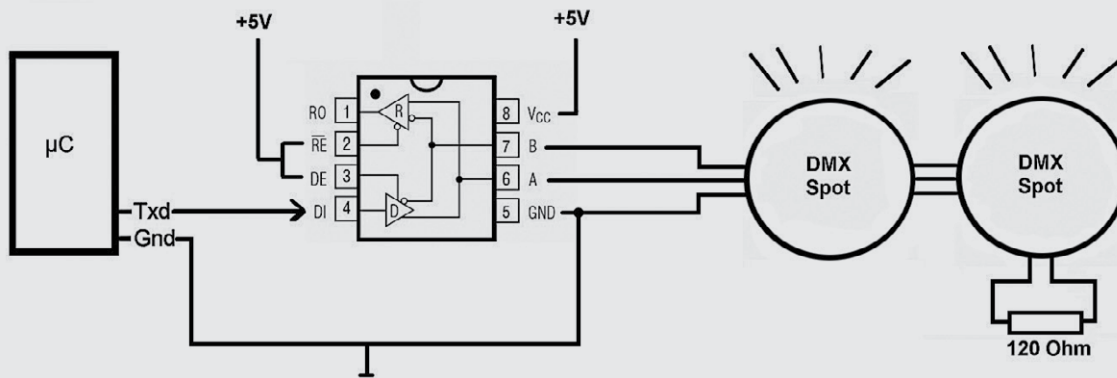


Bild 2: DMX-Spot(s) werden über Pegelwandler (SN75176 oder Max485) an Mikrocontroller angeschlossen.

```
' BASCOM-Programm
,
' DMX-Steuerung mit ATmega88
' Beispiel ELV eurolite LED PST-9W RGB DMX Spot #10 25 14
' ** Version fuer ELV DMX Spot. Andere DMX-Empfänger haben ggf. andere Kodierungen **
' 16 MHz Quarz 2x 22pF nach Gnd
' Ausgabe an TxD von ATmega88. Dahinter z.B. SN75176 Levelshifter. Erzeugt RS-485-Level
,
' In: -
' Out: PortD.1: DMX-Signal an Levelshifter ausgeben.
,
$regfile = „M88def.dat“           'Benutzter Mikrocontroller
$crystal = 16000000               ' Taktfrequenz

$hwstack = 40                     'Rücksprungadressen (je 2), Registersicherungen (32)
$swstack = 40                     'Parameteruebergaben (je 2), LOCALs (je 2)
$framesize = 60                   'Parameter (Daten-Laenge), Rechenbereich Funktionen

Config Portd.1 = Output            'TxD-Ausgang

' Serielle Schnittstelle für DMX konfigurieren:.
Config Com1 = 250000 , Synchron = 0 , Parity = None , Stopbits = 2 , Databits = 8 , Clockpol = 0

Const Timer1startwert = 3036
Config Timer1 = Timer , Prescale = 256           '1-Sekundentimer
Timer1 = Timer1startwert
On Timer1 Timer1_isr
Enable Timer1
Start Timer1
Enable Interrupts

Const Max_dmx_adresse = 10           '2 Spots mit je 5 Kanälen

Dim Dmx_daten(max_dmx_adresse) As Byte      'Array mit DMX-Daten
Dim Dauer As Word                        'Dauer einer Szene

Dim I As Word                            'Schleifenvariablen

Do
Restore Rot
Gosub Anzeigen

Restore Gruen
Gosub Anzeigen
```



```
Restore Blau
Gosub Anzeigen
```

```
Restore Weiss
Gosub Anzeigen
```

```
Restore Rot_blinkend
Gosub Anzeigen
```

```
Restore Gruen_blinkend
Gosub Anzeigen
```

```
Restore Rot
Gosub Anzeigen
```

```
Restore Rot_halbhell
Gosub Anzeigen
```

```
Restore Rot_viertelhell
Gosub Anzeigen
```

```
Restore Dunkel
Gosub Anzeigen
```

```
Loop
End
```

Anzeigen:

```
Gosub Szenendaten_zuweisen
Do
  Gosub Dmx_daten_senden
Loop Until Dauer = 0
Return
```

Szenendaten_zuweisen:

```
Read Dauer
For I = 1 To Max_dmx_adresse
  Read Dmx_daten(i)
Next I
Return
```

Dmx_daten_senden:

```
Baud = 500
Printbin 0;
Waitms 1
Baud = 250000
Printbin 0;
For I = 1 To Max_dmx_adresse
  Printbin Dmx_daten(i);
Next
Waitms 1
Return
```

Timer1_isr:

```
Timer1 = Timer1startwert
If Dauer > 0 Then Decr Dauer
Return
```

```
' Szenen im Format:
' Dauer(in Sekunden),
' rot, grün, blau, gesamt, Funktion für Gerät 1 Ab Adresse 001
' rot, grün, blau, gesamt, Funktion für Gerät 2 Ab Adresse 006
```



'Erst Szene inkl. Dauer zuweisen ..

'.. dann senden

'bis Sekunden heruntergezählt

'DMX-Werte einlesen

'DMX-Daten gemäß DMX-Protokoll senden

'niedrige Geschwindigkeit

' 0 senden

'normale DMX-Geschwindigkeit

'Startbyte

' Daten senden

'Im Sekundentakt herunterzählen



Rot:

```
Data 3% 'Dauer
Data 255 , 0 , 0 , 255 , 0 '1. Gerät rot
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Rot_halbhell:

```
Data 3% 'Dauer
Data 255 , 0 , 0 , 65 , 0 '1. Gerät rot halbhell
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Rot_viertelhell:

```
Data 3% 'Dauer
Data 255 , 0 , 0 , 32 , 0 '1. Gerät rot halbhell
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Grün:

```
Data 3% 'Dauer
Data 0 , 255 , 0 , 255 , 0 '1. Gerät grün
Data 0 , 255 , 0 , 255 , 0 '2. Gerät grün
```

Blau:

```
Data 3% 'Dauer
Data 0 , 255 , 0 , 255 , 0 '1. Gerät blau
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Rot_blinkend:

```
Data 5% 'Dauer
Data 255 , 0 , 0 , 255 , 250 '1. Gerät rot blinkend
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Grün_blinkend:

```
Data 3% 'Dauer
Data 0 , 255 , 0 , 255 , 0 '1. Gerät grün blinkend
Data 0 , 255 , 0 , 255 , 0 '2. Gerät grün
```

Weiss:

```
Data 3% 'Dauer
Data 255 , 255 , 255 , 255 , 0 '1. Gerät weiß
Data 255 , 0 , 0 , 255 , 0 '2. Gerät rot
```

Dunkel:

```
Data 3% 'Dauer
Data 0 , 0 , 0 , 0 , 0 '1. Gerät aus
Data 0 , 0 , 0 , 0 , 0 '2. Gerät aus
```

Erläuterungen:

Wegen der bei DMX hohen Datenübertragungsrate von 250 kBaud wird hier ein externer 16-MHz-Quarz verwendet und dieses dem Compiler mit der \$CRYSTAL-Direktive mitgeteilt. Mit CONFIG Com1 wird eine serielle Schnittstelle mit 250 kBaud (!) definiert, wie es das DMX-Protokoll erfordert. Wie dem Datenblatt des DMX-Spots zu entnehmen ist, müssen pro DMX-Spot 5 Byte gesendet werden (rot, grün, blau, Master, Funktion). Da hier zwei Spots angesteuert werden sollen, müssen laut DMX-Protokoll immer wieder erst die 5 Byte für den ersten DMX-Spot und dann die 5 Byte für den zweiten Spot gesendet werden. Die hintereinandergeschalteten DMX-Geräte werden durch DIP-Schalter so codiert, dass sie die ersten 5 Byte bzw. die zweiten 5 Byte (usw.) empfangen und die anderen Byte für sich ignorieren.

DaseigentlicheSendenderDatenbyteswirdimUnterprogramm dmx_daten_senden mit PRINTBIN umgesetzt. Um dem DMX-Protokoll zu entsprechen, wird vor den Datenbytes ein Startbyte mit dem Wert null gesendet und davor ein Nullpegel, der hier durch temporäres Herunterschalten der Baudrate erfolgt. Durch einen Ein-Sekunden-Timer und die in einer Tabelle hinterlegten Daten können Szenen gesendet werden, deren Dauer und Lichtverhältnisse übersichtlich beschreibbar sind. Eine Erweiterung auf weit mehr als zwei Spots/Geräte ist leicht möglich.



Ausblick

Für das serielle Senden von Daten eignen sich die bisher benutzten BASCOM-Befehle PRINT und PRINTBIN. Je nach Empfänger werden die Übertragungsparameter (Baudrate, Anzahl Datenbits, Paritätsbit und Stoppbits) sowie der jeweilige Befehl verwendet.

Mit BASCOM-Programmen lassen sich Daten auch seriell vom PC, von anderen Mikrocontrollern bzw. von Sensoren oder Modulen diverser Art empfangen. Die Einbindung von ELV-Modulen in eigene Anwendungen wird dadurch leicht möglich.

Im nächsten Teil der Artikelserie „Mikrocontroller-Einstieg mit BASCOM-AVR“ werden verschiedene Methoden vorgestellt, mit UART Daten zu empfangen. **ELV**



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

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

Empfohlene Produkte/Bauteile:	Best.-Nr.	Preis
BASCOM-(Demo-)Lizenz von MCS Electronics	–	–
www.mcselec.com	–	–
Atmel-AVRISP-mkII-Programmer	JQ-10 03 55	€ 39,95
oder myAVR-Board MK2	JQ-10 90 00	€ 49,–
Netzteil für myAVR-Board MK2	JQ-10 90 01	€ 6,95
ATmega88	JQ-10 07 62	€ 3,95
100-nF-Kondensator	JQ-10 03 17	€ 0,08
Batteriehalter für 3x Mignon	JQ-08 15 30	€ 0,75
Batterieclip für 9-V-Block-Batterie	JQ-08 01 28	€ 0,30
BASCOM-Buch	JQ-10 90 02	€ 54,–
Experimentier-Board 1202B	JQ-07 72 89	€ 12,95
Schalt draht-Sortiment	JQ-05 47 68	€ 5,95
LED-Set	JQ-10 63 56	€ 3,95
oder Leuchtdioden	JQ-10 66 60	€ 1,65
und Widerstände	JQ-10 66 57	€ 1,85
Piezo-Signalgeber	JQ-00 73 87	€ 0,95
Mikroschalter und -taster	JQ-10 66 67	€ 2,80
LC-Display, 2x 16 Zeichen	JQ-05 41 84	€ 6,95
oder myAVR-LCD-Add-on		
Pin-Ausrichter	JQ-00 84 63	€ 4,95
USB-UART-Umsetzer UM2102	JQ-09 18 59	€ 5,95
USB-UART-Umsetzer mit Optokopplung U02102	JQ-10 49 66	€ 12,95
MP3-Soundmodul MSM3	JQ-10 57 29	€ 37,95
UART-FS20-Sender FS20 US	JQ-09 87 89	€ 19,95
Funk-Schaltsteckdose FS20 ST	JQ-10 45 37	€ 29,95
Funk-Dimmer FS20 DI-5	JQ-10 46 46	€ 39,95
DMX-Spot	JQ-10 25 14	€ 39,95
FS20-Diagnosetool FS20 DT	JQ-06 68 13	€ 59,95

Preisstellung
November 2013 –
aktuelle Preise
im Web-Shop