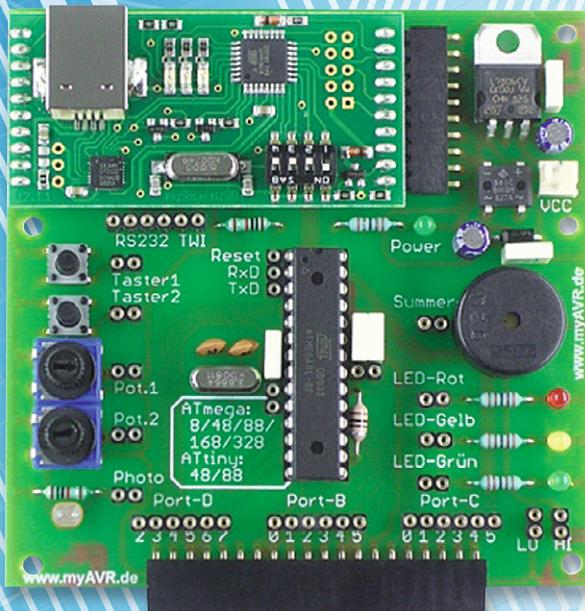




Mikrocontroller-Einstieg

Teil 16: Arduino, ASURO, NIBObee, Raspberry Pi



```
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
$sfrsize = 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
' Wartschleife 500 ms
' B.4 auf 0
' Wartschleife 500 ms
' Schleifenende
' Programmende
```



mit BASCOM-AVR

BASCOM bietet eine sehr mächtige Programmierumgebung für Mikrocontroller, die sehr einfach zu erlernen ist, schnelle Erfolge bringt und Hochsprachensupport für sehr viele Themen bietet. Im einfachsten Fall benutzt man für die Programmentwicklung einen Windows-PC mit der BASCOM-Entwicklungsumgebung, einen Pegelwandler für die serielle Schnittstelle und einen Mikrocontroller mit Bootloader (vgl. [Elektronikwissen](#)). Im Normalfall benutzt man einen PC mit der BASCOM-Entwicklungsumgebung, einen sogenannten Programmer, und den Ziel-Mikrocontroller (Bild 1). Oft verwendet man für die Entwicklung ein sogenanntes Entwicklungsboard, auf dem ein Stecksockel für einen AVR-Mikrocontroller, ein paar Leuchtdioden, Taster, Potentiometer, ein UART-Wandler und ein Programmer verbaut sind. Ein sehr verbreitetes Entwicklungsboard ist zum Beispiel das Entwicklungsboard myAVR MK2 (myAVR MK2, Best.-Nr. J9-10 90 00).

BASCOM ist keine geschlossene Umgebung und die Auswahl an Hardware, auf der BASCOM-Programme

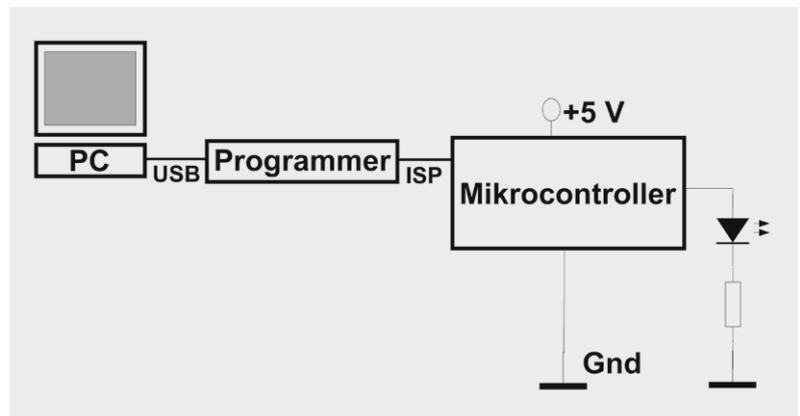


Bild 1: Entwicklungsszenario: PC -> ISP-Programmer -> AVR

laufen, sowie der Hardware, die mit BASCOM-Programmen angesprochen werden kann, ist sehr vielfältig.

In diesem letzten Teil der Artikelserie wird gezeigt, wie man BASCOM in Verbindung mit Arduino, ASURO, NIBObee oder Raspberry Pi einsetzen kann.

ISP

Die Bezeichnung ISP steht für „In-System Programming“ und deutet an, dass der Mikrocontroller während der Programmierung in seiner Schaltung verbleiben

Bootloader

Ein Bootloader ist ein kleines Programm, welches auf einem Mikrocontroller gespeichert ist und ermöglicht, dass Programme ohne Programmer durch Nutzung der seriellen Schnittstelle auf den Mikrocontroller übertragen werden.

Vorteile: keine Programmer-Hardware benötigt und dadurch auch zum Beispiel Firmware-Update durch den Benutzer eines Gerätes möglich. Nur die seriellen Anschlusspins sowie Reset und GND erforderlich.

Nachteile: meist immer noch ein (Pegel-)Wandler (von USB oder RS232 zu TTL) für die serielle Verbindung benötigt, Platzverbrauch im Speicher, Verlangsamung beim Start, nur speziell vorbereitete Mikrocontroller verwendbar – keine „nackten“ Mikrocontroller aus dem Versandhaus. Beim Arduino und ASURO sind Bootloader auf den mitgelieferten Mikrocontrollern vorhanden. Wenn einmal nicht per Bootloader, sondern per ISP übertragen („gebrannt“) wurde, dann ist der Bootloader überschrieben. Ein neuer Bootloader kann per ISP aufgespielt werden.



kann. Während der Entwicklung eines BASCOM-Programms arbeitet man in der BASCOM-Entwicklungs-umgebung auf einem Windows-PC. Das kompilierte BASCOM-Programm wird mit einem Programmer auf einen AVR-Mikrocontroller übertragen (Bild 1).

Ein AVR-Mikrocontroller hat für die In-System-Programmierung (ISP) folgende Anschlüsse: Reset, MISO, MOSI, SCK, +5 V und GND (Bild 3). Sie sind im Pin-out eines AVR-Mikrocontrollers sehr gut zu erkennen (exemplarisch für ATmega88 in Bild 2).

Mit diesem Wissen (AVR-Mikrocontroller mit ISP-Programmierspins vorhanden) kann man sich käuflich erwerbbar Plattformen ansehen und schnell erkennen, ob BASCOM auf diesen Plattformen verwendet werden kann.

Bei den Plattformen Arduino, ASURO und NIBObee sieht man am jeweiligen Schaltplan bzw. in der jeweiligen Produktbeschreibung, dass als Hauptprozessoren AVR-Mikrocontroller verwendet werden und die ISP-Pins (mehr oder weniger einfach) zugreifbar sind. Es ist möglich und bietet sich an, die genannten Plattformen mit BASCOM zu programmieren – auch wenn das ursprünglich nicht vorgesehen ist. Der Vorteil ist, dass man auf diese Weise sehr viele verschiedene Hardware-Plattformen (auch Roboter oder Roboterbausätze) als Basis für BASCOM-Anwendungen verwenden kann.

Arduino

Arduino ist eine weit verbreitete Plattform, die unterschiedliche Hardware-Platinen und eine eigene Arduino-Entwicklungsumgebung bietet. Die Entwicklung geschieht normalerweise mit einer Programmiersprache, die der Sprache C bzw. C++ sehr ähnlich ist und sehr viele fertige Bibliotheken bietet. Das Übertragen des kompilierten Programms

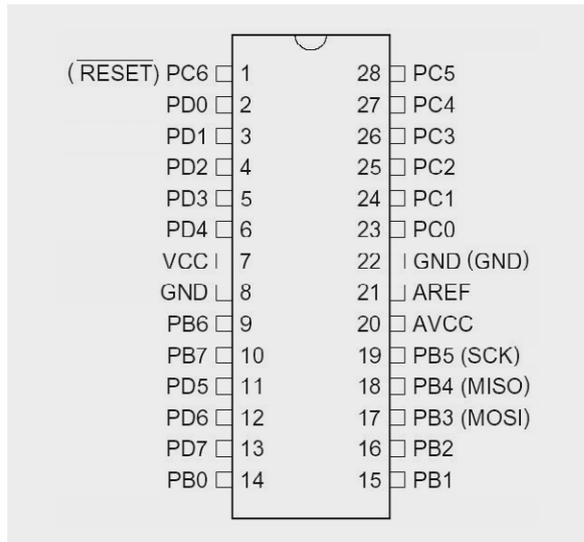


Bild 2: Pin-out eines AVR-Mikrocontrollers mit ISP-Pins

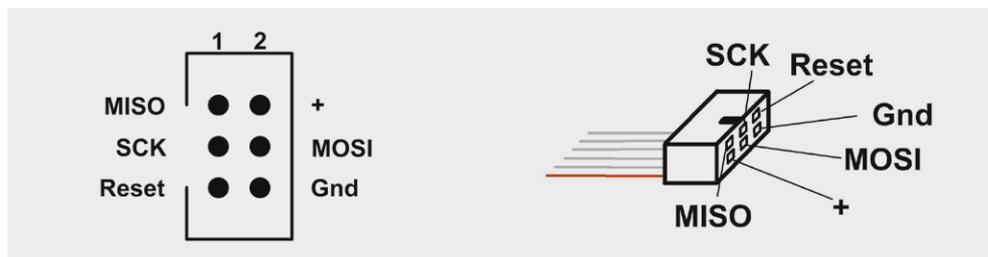


Bild 3: ISP-Schnittstelle

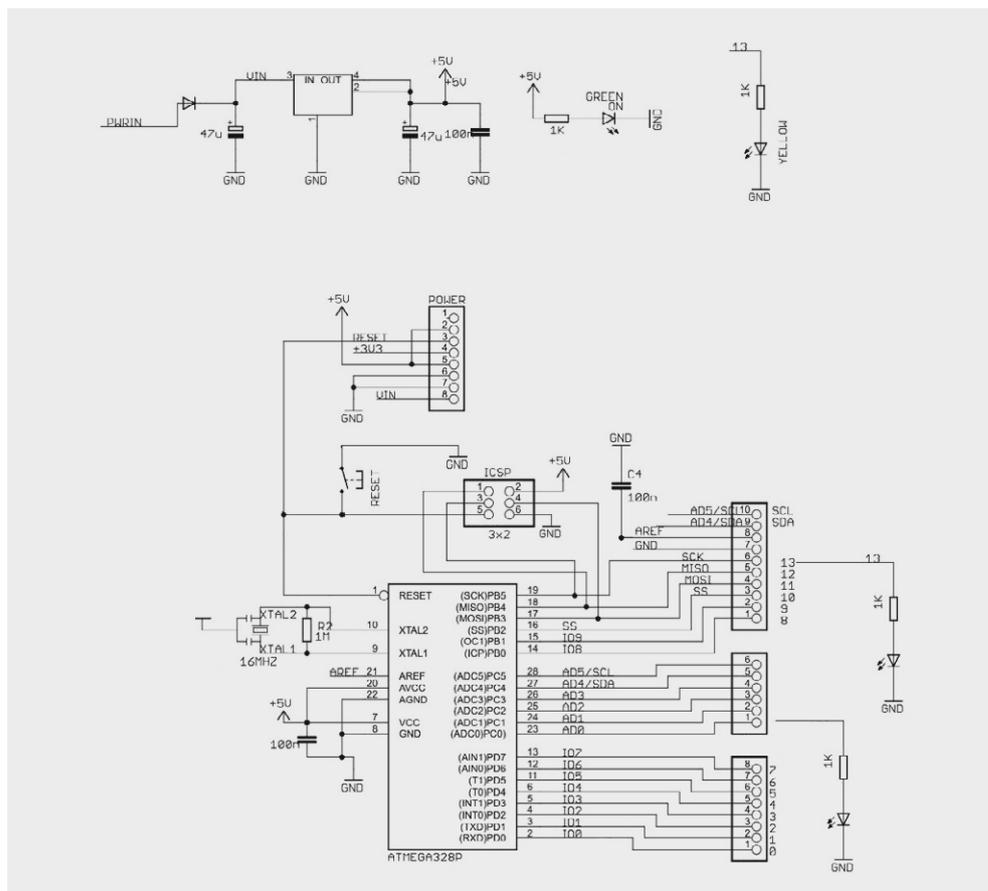


Bild 4: Schaltplan des Arduino UNO



erfolgt im Normalfall durch eine serielle Verbindung auf einen AVR-Mikrocontroller, auf dem sich ein Bootloader befindet.

Wenn man sich stellvertretend für Arduinos einen Schaltplanausschnitt des Arduino UNO (Best.-Nr. J9-11 83 69) ansieht (Bild 4), kann man sehen, dass außer einem leistungsfähigen Mikrocontroller vom Typ ATmega328 auch ein komplett verschalteter 16-MHz-Quarz, eine bereits integrierte Leuchtdiode, Buchsenleisten, Stromversorgung usw. vorhanden sind. Man kann die Arduino-Hardware als Hardware-Entwicklungsplattform oder als Runtime-Hardware benutzen.

In Bild 5 ist der Ausschnitt mit den ISP-Pins herausgezoomt. Deutlich zu sehen ist, dass der Anschluss der Darstellung in Bild 3 entspricht und einer Programmierung des Arduino per ISP nichts im Wege steht.

Wir haben also bei den Arduino-Hardware-Bausteinen jeweils einen AVR-Mikrocontroller, der sich über den ISP-Anschluss von BASCOM aus programmieren lässt.

In Bild 6 sieht man ein Arduino-UNO-Board, bei dem (rechts) das Kabel vom Programmer sowie zwei extern gesteckte 5-V-LEDs (LEDs mit jeweils integriertem Vorwiderstand) zu sehen sind.

Das Übertragen des kompilierten BASCOM-Programms auf den Arduino UNO erfolgt von BASCOM aus (wie im bisherigen Verlauf dieser Artikelserie) per ISP-Schnittstelle. Wenn gewünscht, dann könnte man auch ohne einen extra ISP-Programmer das Programm mit Hilfe des Bootloaders des Arduino übertragen. Man kann jede Arduino-Hardware mit BASCOM nutzen. Die Arduino-Entwicklungsumgebung benötigt man als BASCOM-Entwickler nicht!

Ein einfaches BASCOM-Programm mit den üblichen blinkenden Hallo-Welt-LEDs zeigt, dass auf Arduino ganz normale BASCOM-Programme verwendet werden können.

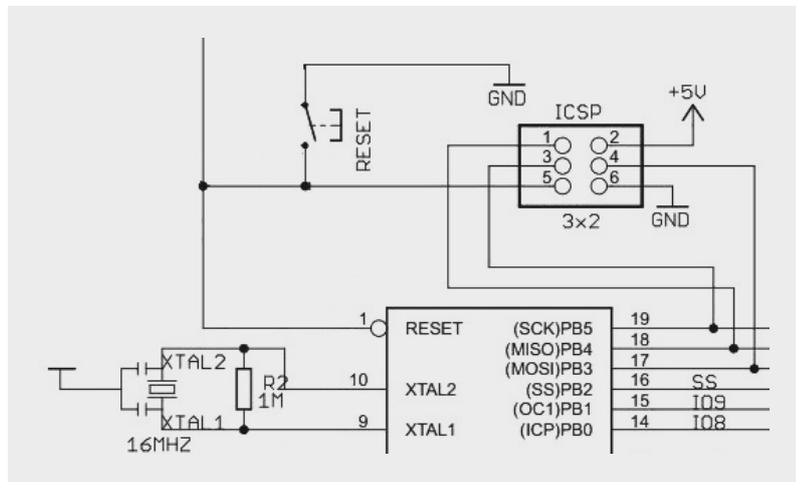


Bild 5: Schaltplan Arduino UNO – ISP-Ausschnitt

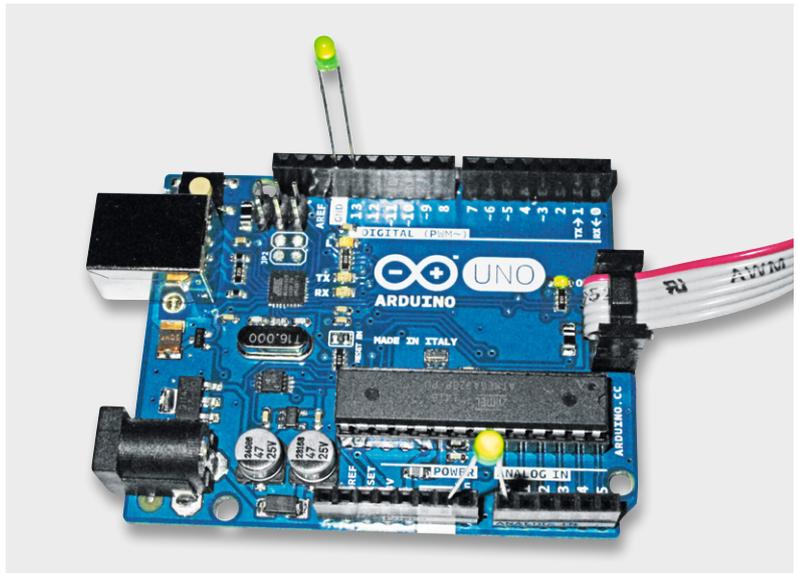


Bild 6: Arduino UNO (LEDs mit integrierten Vorwiderständen!)

```
' BASCOM-Programm
' Arduino Uno R3 DIL
'
' In: -
' Out: LEDs

$regfile = „M328pdef.dat“
$crystal = 16000000
$hwstack = 40
$swstack = 40
$framesize = 60

Config Portb.5 = Output
Led_gelb Alias Portb.5

Config Portc.0 = Output
Led_gruen Alias Portc.0

Do
  Toggle Led_gelb
  Toggle Led_gruen
  Wait 1
Loop
End
```

```
'Verwendeter Chip
'Verwendete Frequenz
'Rücksprungadressen (je 2), Registersicherungen (32)
'Parameteruebergaben (je 2), LOCALs (je 2)
'Parameter (Daten-Laenge), Rechenbereich Funktionen

'Gelbe Onboard-LED und Arduino-Pin 13

'Arduino-Pin A0
```

Erläuterungen zum Programm:

Mit \$regfile muss der Mikrocontroller, der auf dem Arduino-Board verbaut ist, eingestellt werden. Nach der üblichen Einstellung der Stackwerte werden die Ausgabepins konfiguriert und Aliasnamen vergeben. Das Programm zeigt die Verwendung der auf dem Board verbauten LED (an Portb.5). Die Arduino-Hardware wird als ganz normale, fertig aufgebaute Hardware verwendet. Spannungsversorgung und Quarz sind bereits sofort nutzbar verlötet.

Alle Themen der BASCOM-Artikelserie wie I²C, serielle Übertragung, 1-Wire usw. können mit einem BASCOM-Programm auf Arduino-Hardware realisiert werden.

ASURO

Der ASURO (Bild 7) (Best.-Nr. J9-09 73 14, J9-10 71 95) wurde als kleine Roboter-Plattform am Deutschen Zentrum für Luft- und Raumfahrt (DLR) für die Lehre entwickelt. ASURO ist eine Abkürzung und bedeutet „Another Small and Unique Robot from Oberpfaffenhofen“. Bild 8 zeigt die Komponenten eines ASURO-Roboters. Im ASURO ist ein ATmega8 verbaut, den man mit BASCOM programmieren kann. Wie im Arduino gibt es auch beim ASURO einen vorinstallierten Bootloader, mit dessen Hilfe das kompilierte BASCOM-Programm vom PC auf den Mikrocontroller übertragen werden kann. Der Bootloader belegt 1 kB von insgesamt 8 kB Flash-Speicher des ATmega8. Um den Mikrocontroller des ASURO per ISP statt mit dem Bootloader zu beschreiben, kann man sich eine kleine Huckepack-Platine bauen, bei der alle Pins bis auf den Reset-Pin 1:1 verbunden und die ISP-Pins auf eine ISP-Buchse gemäß Bild 3 geführt werden. Der im ASURO verbaute Mikrocontroller muss dann gegen einen frischen ATmega8 (ohne Bootloader) getauscht werden. Im Schaltplan des ASURO ([1] Seite 74) erkennt man gut den AVR-Mikrocontroller mit seinen ISP-Anschlusspins. Im Internet findet man Beispiele für BASCOM-Programme für den ASURO [2] und Huckepack-Platinen [3].



Bild 7: ASURO-Roboter

NIBObee

Der NIBObee (Bild 9, Best.-Nr. J9-10 21 12) ist ein Roboterbausatz von nicali Systems, der sich hervorragend als Einstiegs- und Lernobjekt eignet. Beim NIBObee ist als Haupt-Mikrocontroller ein ATmega16 eingebaut. Des Weiteren sind Taster, LEDs, Motoren usw. verbaut ([4] und [5]).

Für die Programmierung wird beim NIBObee ein ATtiny44 als Programmierer verwendet. Man kann diesen ATtiny44 aus seiner Fassung entfernen und stattdessen einen vorhandenen Programmierer verwenden. Somit kann man den NIBObee mit BASCOM und ISP verwenden.

Raspberry Pi

Für gut 30 Euro erhält man mit dem Raspberry Pi (Bild 10) ein sehr universell einsetzbares Gerät, welches sich durch Schnittstellen zu einem Bildschirm (HDMI), LAN, Tastatur/Maus (USB), WLAN-Stick (USB), USB-Stick/Festplatte (USB) und SD-Karte für extrem viele Anwendungen individuell einsetzen lässt. Der Raspberry Pi lässt sich nicht mit BASCOM programmieren, weil er keinen AVR-Mikrocontroller als Basis verwendet,

aber er ergänzt ganz hervorragend BASCOM-Projekte. EIN Vorteil des Raspberry ist, dass sich sehr leicht grafische Anwendungen für große Bildschirme erstellen lassen, die sich mit BASCOM ansprechen lassen. Auch eine Internetverbindung gehört zu den Standardthemen des Raspberry. Man kann dadurch sehr einfach Daten aus BASCOM-Anwendungen über einen Webbrowser (im LAN/WLAN oder über das Internet) zur Verfügung stellen bzw. über einen Browser Daten an eine BASCOM-Anwendung übergeben.

Komplexe Home-Automatisierungs-Projekte lassen sich durch die Kombination aus BASCOM-Anwendung plus Raspberry sehr schön realisieren.

Bild 11 zeigt eine mit Python und Tkinter in kurzer Zeit erstellte Oberfläche, die die seriell vom BASCOM-Programm empfangenen Daten auf einem großen (HDMI-)Bildschirm anzeigt.

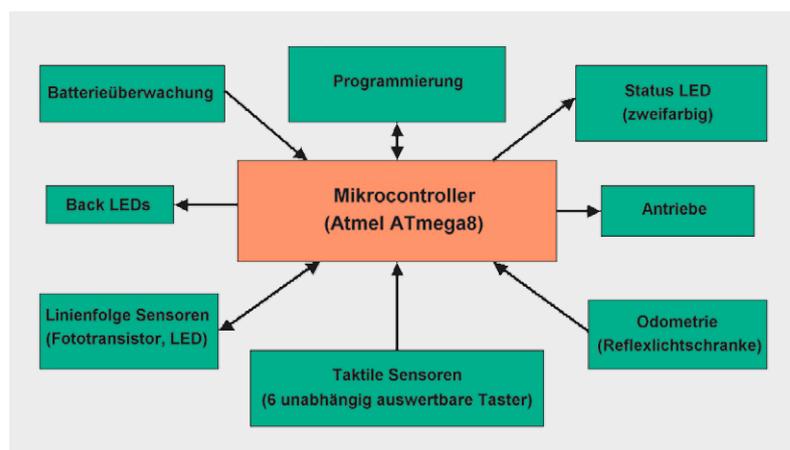


Bild 8: ASURO-Blockbild



Bild 9: NIBObee



Auf dem AVR-Mikrocontroller werden die Daten gewonnen und, wie im folgenden Listing angedeutet, zum Raspberry gesendet.

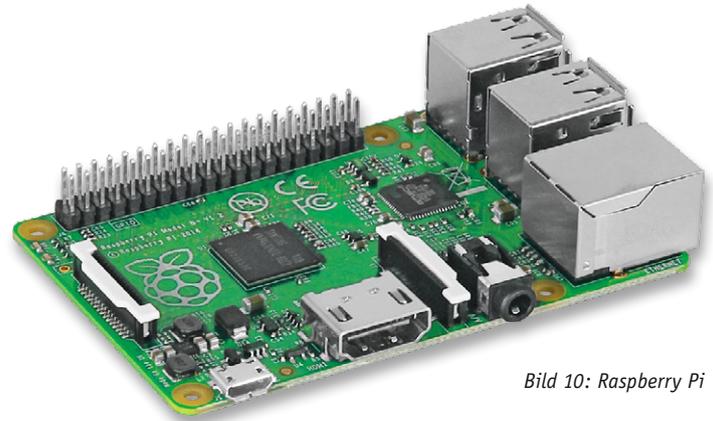


Bild 10: Raspberry Pi

```
' BASCOM-Programm
'
' Serielle Ausgabe an Raspberry Pi
' Texte/Zahlen von AVR zum Raspberry Pi senden mit PRINT
' Am Raspberry Pi empfangen mit Python/Tkinter-Programm
'
' In: Analogwert an C.0 und C.1
' Out: Serielles Signal an Pin D.1 = Txd

$regfile = „M328pdef.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 = 9600                          '9600 Bits pro Sekunde, 8 Datenbits, Keine Parität, 1 Stoppbits

Config Adc = Single , Prescaler = Auto , Reference = Avcc

Dim Adcwert0 As Word
Dim Adcwert1 As Word
Dim Adcsingle As Single
'
Do
Adcwert0 = Getadc(0)                  'Analogwerte einlesen
Adcwert1 = Getadc(1)

Print "$A";                          'Kennzeichen ausgeben
Print Adcwert0;                       'Ganzzahl ausgeben
Print Chr(010);

Adcsingle = Adcwert1 / 10              'Kommazahl erzeugen
Print "$B";                           'Kennzeichen ausgeben
Print Fusing(adcsingle , "#.#");      'Kommazahl ausgeben
Print Chr(010) ;

Print "$C";                           'Kennzeichen ausgeben
If Adcsingle < 60.0 Then
  Print "o.k.";                       'String ausgeben
Else
  Print "Alarm";
End If
Print Chr(010);
Waitms 10
Loop

End
```

Erläuterungen zum Programm:

Das BASCOM-Programm bildet das Skelett, an welchem die serielle Datenübertragung zum Raspberry dargestellt wird. Mit \$baud wird die Übertragungsrate festgelegt, die auf dem Raspberry übereinstimmend definiert sein muss. Nachdem im Programm stellvertretend für Sensorauswertungen zwei Analogeingänge ausgewertet wurden, sieht man die Ausgabe über die serielle Schnittstelle. Zunächst wird jeweils ein Kennzeichen – \$A, \$B oder \$C – an den Raspberry gesendet. Auf diese Weise ist es möglich, auf dem Raspberry aus dem seriellen Datenstrom die gesendeten Werte entsprechend zuzuordnen. Danach wird jeweils eine Ganzzahl, eine Kommazahl bzw. ein String an den Raspberry gesendet.



Bild 11: Raspberry-Pi-Anwendung



Fazit

Die im Verlauf unserer Artikelserie „Mikrocontroller-Einstieg mit BASCOM-AVR“ vorgestellten Möglichkeiten und Techniken lassen sich sehr gut in Kombination mit verbreiteten Plattformen verwenden. Bei Arduino, ASURO und NIBObee können BASCOM-Programme per ISP auf den jeweiligen Mikrocontroller programmiert („gebrannt“) werden. Die Kombination aus BASCOM-Programm auf einem AVR-Mikrocontroller und Raspberry Pi ermöglicht auf einfache Weise grafische Anwendungen auf einem großen Monitor, LAN/WLAN- oder Internetverbindungen und vieles Weitere. Bei beengten Platzverhältnissen (Modellbau), einfachen Anwendungen (Blinker, Ampel, Spiele ...) oder batteriebetriebenen Geräten bietet sich die Benutzung eines einfachen Mikrocontrollers an. Die angesprochenen Möglichkeiten sind als Ergänzung und keineswegs als Ablösung von BASCOM auf einem einzelnen Mikrocontroller zu verstehen.

Ausblick

In den bisherigen Teilen der BASCOM-Artikelserie wurden verschiedene Themen wie UART, I²C, SPI, 1-Wire usw. behandelt. In diesem abschließenden Teil wurde betrachtet, dass man BASCOM auf Arduino, ASURO oder NIBObee einsetzen bzw. die Möglichkeiten eines Raspberry Pi nutzen kann.

Insgesamt hat man damit als Entwickler eigener Anwendungen mit BASCOM eine sehr mächtige Programmierumgebung zur Verfügung, mit der sich alle Themen lösen lassen. 



Weitere Infos:

- [1] ASURO-Beschreibung mit Schaltplan:
http://files.elv.de/service/manuals_hw/59760_Roboter_Asuro_UM.pdf
- [2] ASURO-BASCOM-Programme: <http://wiesolator.de/index.php?area=Robotik&topic=Asuro>
- [3] ISP-Erweiterung für ASURO:
<https://asuroprojekt.wordpress.com/2010/03/14/spi-erweiterung-fur-isp/>
- [4] NIBObee-Handbuch:
http://files.elv.de/Assets/Produkte/10/1021/102112/Downloads/102112_nibobee_um.pdf
- [5] NIBObee-Schaltplan:
http://files.elv.de/Assets/Produkte/10/1021/102112/Downloads/102112_nibobee_data.pdf
- 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
- Produktübersicht BASCOM: www.elv.de/bascom.html

Empfohlene Produkte/Bauteile:

| Empfohlene Produkte/Bauteile: | Best.-Nr. | Preis |
|--|-------------|---------|
| BASCOM-(Demo-)Lizenz von MCS Electronics, www.mcselec.com | - | - |
| ATmega88 | J9-10 07 62 | € 3,95 |
| BASCOM-Buch | J9-10 90 02 | € 54,- |
| Arduino Micro | J9-10 97 74 | € 24,95 |
| Arduino UNO | J9-11 83 69 | € 29,95 |
| ASURO Miniroboter | J9-09 73 14 | € 49,95 |
| ASURO Miniroboter | J9-10 71 95 | € 49,95 |
| NIBObee Roboterbausatz | J9-10 21 12 | € 49,90 |
| Raspberry Pi Typ B+ Starter-Set | J9-11 82 00 | € 79,95 |
| Raspberry Pi Typ B+ 512 MB | J9-11 78 48 | € 29,95 |
| Raspberry Pi 2 Typ B 1 GB | J9-11 93 85 | € 37,95 |
| Noobs Betriebssystem für Raspberry Pi | J9-11 81 86 | € 14,95 |
| EDIMAX USB-WLAN-Adapter | J9-11 18 84 | € 7,95 |
| HDMI-zu-DVI-Adapterkabel | J9-11 70 78 | € 2,95 |
| Netzteil für Raspberry Pi | J9-11 18 30 | € 4,95 |
| hama-Kartenleser „All in one“ | J9-08 79 58 | € 8,95 |