

# Vielfältiger Schmetterling

Atmels AVR-Mikrocontroller sind durch ihre einfache und komfortable Handhabung in Entwicklerkreisen sehr beliebt. Mit dem Starterkit AVR Butterfly steigen Sie preisgünstig in die Mikrocontroller-Entwicklung unter Linux ein, beispielsweise wenn Sie eine Robotersteuerung entwickeln wollen. Nils Faerber



Die **AVR-Mikrocontroller** von Atmel sind Minicomputer in besonders kompakter Bauform, die sich mit ihren zahlreichen Schnittstellen selbst für komplexere Steuer- und Regelaufgaben eignen. Zwar läuft kein Linux-Kernel auf dem Mikrocontroller, für die Software-Entwicklung unter Linux finden Sie jedoch alle nötigen Crosscompiler und Werkzeuge kostenlos im Internet.

## AVR-Hardware

Die AVR-Mikrocontroller-Familie [1] von Atmel basiert auf einem 8-Bit-Risc-CPU-Kern mit 32 8-Bit-Registern. Der Befehlssatz der CPU-Familie umfasst etwa 130 Instruktionen und ist sehr gut dokumentiert. Da es sich um eine Risc-CPU handelt, arbeitet sie fast alle Instruktionen in einem Taktzyklus ab. Bei einer Takt rate von 16 MHz erreichen AVR-Controller also fast 16 Mips (Million Instructions Per Second). Im Controller sind SRAM, Flash und EEPROM integriert. Der Mikrocontroller

ist für kleine Applikationen gedacht, der RAM also entsprechend. Der Flash-Speicher dient als Programmspeicher und das EEPROM nimmt kleinere Datenmengen auf, etwa Einstellungen oder Programmparameter.

Die AVR-Controller haben für den Hobbyentwickler einige Vorteile, so ist der Assembler-Befehlssatz sehr komfortabel – Sie brauchen also keine Angst davor zu haben, Programme in Assembler schreiben zu müssen. Für den Programmtransfer zum Mikrocontroller ist außerdem keine aufwändige Programmierhardware erforderlich, eine synchrone serielle Vierdraht-Schnittstelle (SPI) reicht aus. Diese kann zudem unproblematisch mit der seriellen oder parallelen Schnittstelle eines Standard-PC emuliert werden. Das tatsächliche Programmier-Interface reduziert sich damit auf ein paar Drähte, Widerstände und Kondensatoren.

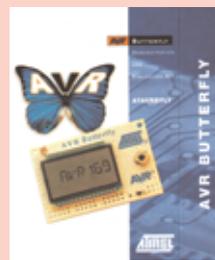
Das Starterkit AVR Butterfly ist ein gut ausgestattetes, fertig aufgebautes und mit 30 bis 40 Euro ungewöhnlich preis-

wertes System. Ein paar Bezugsquellen nennt **Tabelle 1**, Ausstattungsdetails beschreibt der Kasten „**Hardware-Ausstattung von Butterfly**“. Highlights sind das LC-Display mit 100 Segmenten, die eingebauten Sensoren und die serielle Schnittstelle mit RS-232-Pegeln. Das Ganze hat Atmel zusammen mit einer 3-Volt-Knopfzelle für die Stromversorgung auf einer Platine von der Größe einer Visitenkarte untergebracht, Butterfly ist also sogar portabel zu betreiben.

## Benötigte Teile

Für die Programmierung sind noch einige Bauteile für den Adapter und vor allem ein geregeltes 5-V-Netzteil erforderlich, weil Butterfly normalerweise mit +3-V-Versorgungsspannung betrieben

### Atmel AVR Butterfly



Kit zum Einstieg in die Atmel-Mikroprozessor-Programmierung, fertig aufgebaute und mit unterschiedlichen Sensoren ausgestattete Platine

**Hersteller:** Atmel [<http://www.atmel.com>]

**Lieferumfang:** AVR Butterfly mit Knopfzelle und vorinstallierter Beispiel-Firmware

**Funktionsumfang:** Uhrzeit, Datum, rudimentäre Soundausgabe, elektronisches Namensschild, Temperatur-, Helligkeits- und Spannungsmessung

**Preis:** ca. 30 bis 40 Euro

### Hardware-Ausstattung von Butterfly

- Atmel Atmega 169 AVR Controller
- LC-Display mit 100 Segmenten
- 4 MBit separater Daten-Flash
- 32-kHz-Oszillator zum Betrieb der Real-time-Clock (RTC)
- Vier-Wege-Joystick zur Bedienung
- Lichtsensor (LDR)
- Temperatursensor (NTC)
- Piezolausprecher
- Serielle Schnittstelle mit RS-232-Pegeln
- A/D-Wandler-Eingang für Spannungen von 0 bis 5 Volt
- Ein-/Ausgänge für: A/D-Wandler, Spannung, UART, USI, Port-B JTAG
- Port-D und In-System-Programmer (ISP)

wird, die parallele PC-Schnittstelle aber +5-V liefert. Für den Programmieradapter brauchen Sie noch einen 25-poligen Sub-D-Stecker zum Anschluss an die parallele Schnittstelle, eine mindestens 6-polige und etwa 50 Zentimeter lange Leitung, am besten Flachbandleitung, und eine 2x3-polige Stiftleiste, also zweireihig mit je drei Pins und 2,54 Millimeter Rastermaß.

Der Anschluss erfolgt entsprechend **Tabelle 2**, die Belegung der ISP-Kontakte am Butterfly zeigt **Abbildung 1**. Die Stiftleiste löten Sie entweder fest auf die Butterfly-Platine oder stecken sie nur zum Programmieren in die Lötungen. Die zweite Methode belässt Butterfly in seinem Originalzustand, erfordert aber eine ruhige Hand.

Bevor Sie nun direkt den LötKolben zücken, entfernen Sie noch die Batterie von der Platine oder isolieren sie mit einem Papierstreifen. Die In-System-Programmer-Schnittstelle dürfen Sie wegen der genannten Spannungsunterschiede ausschließlich mit einer externen Stromversorgung benutzen.

Bei langen Flachbandleitungen sollten Sie – wie in **Abbildung 2** gezeigt – neben jeder Signalleitung eine Masseleitung führen, um Einstrahlungen und

Übersprechen zu vermeiden – dafür brauchen Sie dann doppelt so viele Adern wie normal. Bei kurzen Kabelstrecken sind die zusätzlichen Masseleitungen nicht erforderlich.

## Software

Am einfachsten geht die Software-Installation unter Debian GNU/Linux vonstatten. Der Unstable-Zweig der Distribution enthält im APT bereits alle nötigen Pakete, die älteren Versionen unterstützen den neuen AVR-Controller von Butterfly noch nicht. Sie brauchen also lediglich die betreffenden Pakete nachzuinstallieren. Das sind der Micro-In-System-Programmer »uisp« **[3]**, der fast alle bekannten Interface- und AVR-Controller-Varianten beherrscht; »binutils-avr« mit den Binary Utilities, der AVR-GCC »gcc-avr« sowie die »avr-libc« **[4]**, eine speziell auf den AVR angepasste Version der Libc mit einem kleinen Subset der gewohnten Funktionen.

Für Nicht-Debian-Systeme gibt es im Internet verschiedene vorkompilierte Versionen, etwa bei AVR-Freaks **[5]**, RPM-Pakete sind über RPM Find **[6]** zu finden. Nach der Installation der genannten Pakete sind Sie eigentlich schon für die Entwicklung gerüstet. Sie sollten allerdings die entsprechende Dokumentation von Atmel **[1]** herunterladen und durchsehen, sowohl die für das Butterfly-Kit selbst wie auch für den Controller Atmega 169.

## Beispiel

Das Kit wird ab Werk mit einem Beispielprogramm ausgeliefert, das die Funktion der Sensoren, des Joysticks und des Displays demonstriert. So lässt sich Butterfly als Uhr oder Kalender einsetzen, es zeigt die Temperatur oder Helligkeit auf dem Display an, misst Span-

Tabelle 1: Bezugsquellen

Anbieter	Preis
<a href="http://www.reichelt.de">[http://www.reichelt.de]</a>	ca. 39 Euro
<a href="http://www.sander-electronic.de">[http://www.sander-electronic.de]</a>	ca. 33 Euro
<a href="http://shop.mikrocontroller.net">[http://shop.mikrocontroller.net]</a>	ca. 30 Euro
<a href="http://www.kernelconcepts.de">[http://www.kernelconcepts.de]</a>	ca. 28 Euro

Tabelle 2: Anschlussbelegung

Parallelport	Butterfly	Netzteil
19-25	GND	GND
1	PB1/SCK	
2	PB2/MOSI	
11	PB3/MISO	
16	RESET	
	VCC	+5V

RESET ■ ● VCC  
 PB3 / MISO ● ● PB2 / MOSI  
 PB 1 / SCK ● ● GND

**Abbildung 1: Butterfly ist für den Anschluss eines Programmierkabels vorbereitet, das Bild zeigt die Signalbelegung des Löttaugen-Felds auf der Platine.**

nungen bis 10 Volt oder spielt einige Soundsamples über den Piezolausprecher ab. Ein besonderer Clou ist die Funktion als digitales Namensschild: Sie geben einfach Ihren Namen ein – er wird auf dem Display angezeigt.

Die Quellen des Beispielprogramms gibt es bei Atmel zum Download. Der Quellcode und die Projektdatei sind aber für die Windows-Entwicklungsumgebung AVR-Studio gedacht und lassen sich mit den GNU-Tools nicht übersetzen. Eine Gruppe von Enthusiasten hat den Atmel-Quellcode für die GNU-Toolchain überarbeitet und stellt ihn auf der Seite von Martin Thomas [7] zur Verfügung. Dort gibt es viele weitere Informationen rund um Butterfly sowie viele Links zu anderen Projekten. Da die Beispielapplikation die gesamte Peripherie des Kits unterstützt, lässt sie sich gut als Einstiegs- und Lernbeispiel verwenden.

## Selbst gemacht

So übersetzen Sie ein AVR-Programm unter Linux und übertragen es zum Butterfly: Von der Seite von Martin Thomas [7] laden Sie das für die GNU-Tools angepasste Quellpaket »bf\_gcc\_20031205.zip« herunter und entpacken es. Im Verzeichnis »bf\_gcc« sollte ein »make« genügen, um alle Dateien zu übersetzen. Wenn nicht, ist das ein Indiz dafür, dass die Entwicklungspakete nicht korrekt installiert sind. Am Ende des erfolgreichen Compilerlaufs entsteht die Datei »main.hex«, sie ist das AVR-Programm im Intel-Hexformat, ein Format, in dem Binärdateien für Mikrocontroller üblicherweise gespeichert werden.

### Der Autor

Nils Faerber arbeitet als Systementwickler für Embedded Linux. Zu erreichen ist er per E-Mail unter: [nils.ferber@kernelconcepts.de]

Butterfly muss jetzt mit der Datei »main.hex« programmiert werden, dazu dient das Programm »uisp«. Der Programmieradapter-Typ, der hier zum Einsatz kommt, heißt Direct AVR Parallel Access oder kurz »dapa«. Schließen Sie die Platine an den Rechner an, zuerst das Parallelportkabel und dann die Stromversorgung. Zum Testen der Verbindung rufen Sie »uisp« einfach mit der Angabe des Adapter-Typs auf, das Programm versucht dann, die Identifikation des Mikrocontrollers auszulesen. Klappt das und zeigt »uisp« den richtigen Baustein an, funktioniert das Kabel:

```
# uisp -dprog=dapa
Atmel AVR ATmega169 is found.
```

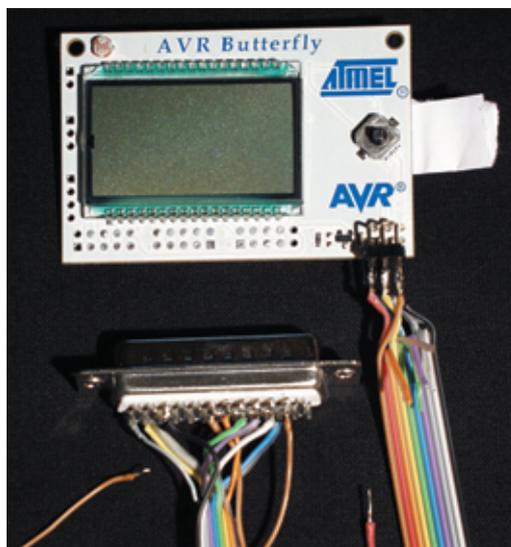
Erscheint diese Meldung nicht, trennen Sie sofort alle Verbindungen und prüfen alle Anschlüsse.

Als Nächstes folgt die Programmierung. Der Programmspeicher des Atmega 169 ist ein Flash, den Sie vor der Programmierung löschen müssen:

```
# uisp -dprog=dapa --erase --segment=flash
Atmel AVR ATmega169 is found.
Erasing device ...
Reinitializing device
Atmel AVR ATmega169 is found.
```

Nachdem Sie den Mikrocontroller gelöscht haben, übertragen Sie das neue Programm:

```
# uisp -dprog=dapa --upload if=main.hex
Atmel AVR ATmega169 is found.
Uploading: flash
```



**Abbildung 2: AVR Butterfly hat rechts neben dem 100-Segment-Display einen Joystick. Unten ist der Programmier-Adapter mit zusätzlichen Masseleitungen zu sehen.**

Um sicherzustellen, dass die Übertragung tatsächlich erfolgreich war, sollten Sie noch einen Vergleich (Verify) durchführen:

```
# uisp -dprog=dapa --verify if=main.hex
Atmel AVR ATmega169 is found.
Verifying: flash
```

Gibt es keine Fehlermeldung, war der Vergleich erfolgreich, das Programm wurde also korrekt übertragen. Im LC-Butterfly-Display läuft nun der Text »AVR BUTTERFLY GCC« durch.

## Große Vielfalt

Martin Thomas stellt auf seiner Seite [7] einige Anwendungsbeispiele vor: Butterfly als Datenlogger oder automatischer SMS-Versand über ein seriell angeschlossenes Mobiltelefon. Mit dem Joystick und dem Display taugt Butterfly theoretisch sogar zu einer rudimentären Spielekonsole für unterwegs.

Die beschriebene Entwicklungssoftware unterstützt nicht nur Butterfly mit dem Atmega 169, sondern fast alle Mikrocontroller der Atmel-AVR-Familie. Wer will, kann also mit den gleichen Methoden und den gleichen Programmen auch Software für eine große Bandbreite an Mikrocontrollern entwickeln, von Mini-Controllern mit gerade mal acht Beinchen bis hin zu den großen AVR-Controllern mit viel Speicher im SMD-PLCC-Gehäuse. (mdö)

### Infos

- [1] Atmel AVR: [<http://www.atmel.com/products/AVR>]
- [2] AVR Butterfly: [<http://www.atmel.com/products/AVR/butterfly>]
- [3] UISP: [<http://savannah.nongnu.org/projects/uisp>] und [<http://savannah.nongnu.org/download/uisp>]
- [4] AVR LIBC: [<http://savannah.nongnu.org/projects/avr-libc>]
- [5] AVR Freaks: [<http://www.avrfreaks.net/>]
- [6] RPM Find: [<http://rpmfind.net/>]
- [7] AVR Projects, Butterfly Application Port für GCC: [[http://www.siawari.arubi.uni-kl.de/avr\\_projects/](http://www.siawari.arubi.uni-kl.de/avr_projects/)]