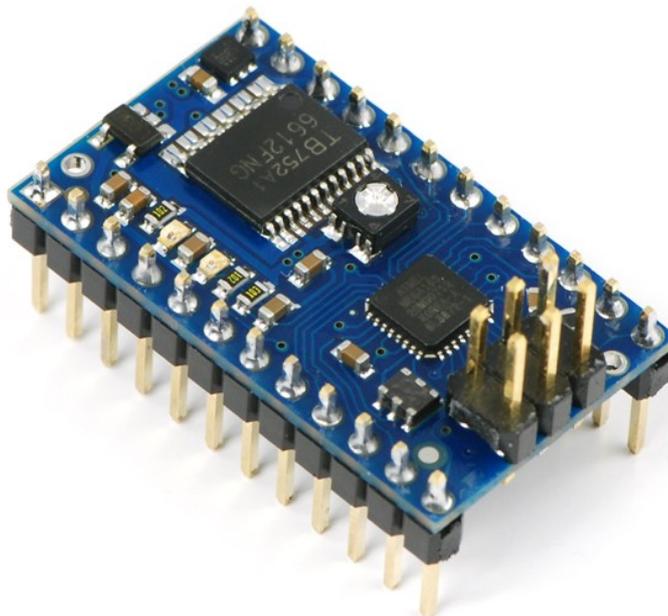


Anleitung - Stand vom 04.7.2011

# Baby Orangutan B-328

Miniatur Controllerboard Modul mit Motortreiber  
für universellen Einsatz!

Das winzige Modul für alle Aufgaben,  
ideal für Experten aber auch Einsteiger.



Das Controllerboard „Miniatur Baby Orangutan B-328“ ist ein komplettes Mikrocontrollerboard-Modul das alle wichtigen Bestandteile beinhaltet um vielfältige Aufgaben ohne weitere Komponenten zu erledigen. So ist neben dem Controller auch eine Spannungsstabilisierung und ein Motortreiber für 2 Motoren auf dem winzigen Board enthalten. Getaktet wird das Board mit bis zu 20 Mhz (quarzstabilisiert).

Das Board eignet sich beispielsweise ideal um einen kleineren bis mittleren Roboter komplett zu steuern. Motoren und Sensoren lassen sich dabei direkt anschließen.

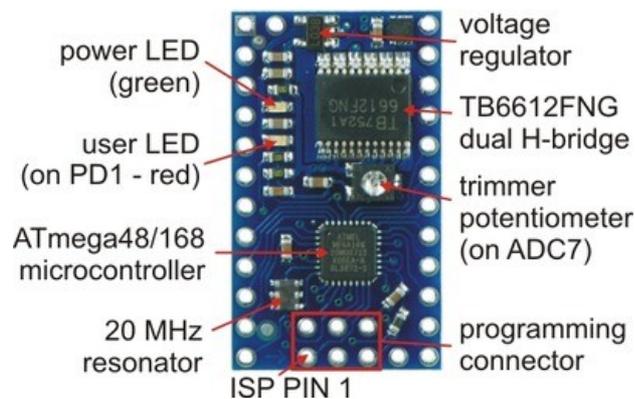
Aufgrund der geringen Größe und dem niedrigen Preis kann das Board auch in vielen anderen Anwenderbereichen eingesetzt werden. Analoge Schaltungen können durch dieses preiswerte Board sehr leicht mit etwas „Intelligenz“ ausgestattet werden. Beispiele wären: Modellbau, Modellflugzeugsteuerungen, Lichtschranken, Zählerschaltungen, blitzschnelle Foto-Auslöser mit IR-Lichtschranken oder Bewegungsmelder, GeoCache-Schaltungen uvm.

**Die Programmierung erfolgt per Bascom Basic-Compiler, C oder**

## Assembler. Beispiele auf mitgelieferter DVD enthalten.

### Features:

- Winzige Größe : ca. 3cm x 1,7cm
- Eingangsspannung zwischen 5 und 13.5 V (maximal 15 V)
- Zwei Motortreiber mit jeweils ca. 1 A Dauerleitung (max. 3 A peak)
- Programmierbarer 20MHz Atmel ATmega328P AVR Mikrocontroller (32 KB Flash, 2 KB RAM, 1 KB EEPROM)
- 18 user I/O lines, 16 davon digital und 8 auch analog verwendbar
- eine programmierbare LED auf dem Board
- Power LED
- Ein Potentiometer (Regler) am analogen Port (ADC7) frei verwendbar
- 20 MHz Quarz (wahlweise umschaltbar auch auf andere interne Taktfrequenzen)
- Pinkompatibel zu anderen Orangutan Boards



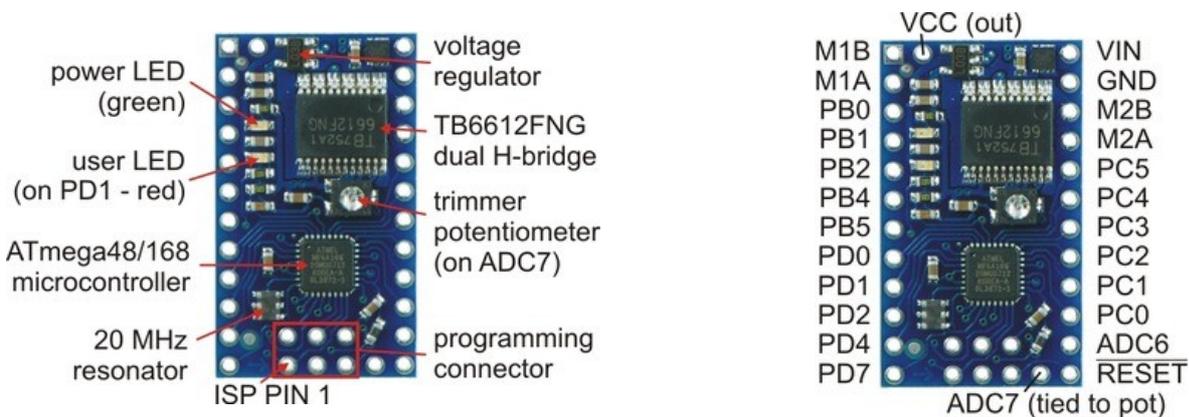
### Daten

- Prozessor: **ATmega328P** @ 20 MHz
- RAM im Controller: 2048 Bytes
- Programmspeicher (Flash): 32 Kbytes
- Motor Treiber: TB6612FNG
- Motor Anschlüsse: 2 Motoren mit programmierbarer Geschwindigkeit und Richtungswahl
- I/O Ports: 18 (16 digital / 8 analog)
- Maximale Belastbarkeit pro Port : 40 mA
- Minimale Spannung : 5 V
- Maximale Spannung: 13.5 V (absolutes Maximum 15 V)
- Motortreiber belastbar mit 1A pro Motor (Kurzzeitig/Peak 3A)
- Maximale PWM Frequenz: 80 kHz
- das Board besitzt einen Verpolungsschutz
- Standard ISP-Programmer anschließbar, z.B. der Atmel USB MK2 von Robotikhardware.de oder der Bascom USBISP mit Adapter (ebenfalls Robotikhardware.de)
- Einlötbare Stiftleisten – dadurch auch passend um Board auf Steckbrett oder Experimentierplatinen zu stecken

## Pinnbelegung

Da das Controllerboard bereits einen programmierbaren Motortreiber, Quarz, LED und Potentiometer (Einstellregler) beinhaltet, sind nicht alle programmierbaren I/O Ports für andere Aufgaben frei. Beachten Sie daher stets die Belegung im Schaltplan oder in der Tabelle etwas weiter hinten in dieser Dokumentation.

Das Board ist von Größe und Raster kompatibel zu dem DIP 24 Pin Raster (also 24 poliges IC Raster). Es kann also sehr leicht in integrierten Schaltungen auf Platinen eingesteckt oder reingelötet werden. Auch die Verwendung in sogenannten Steckbrettern, die gerne zum Experimentieren genutzt werden, paßt das Modul optimal.



### VIN

Eingangsspannung +5 Volt bis + 13.5 V, (absolutes maximum 15 V).

### GND

Eingangsspannung Minuspol (Masse)

### M1A und M1B

Motoranschlüsse für Motor 1 (belastbar mit 1A) Natürlich auch zum steuern von anderen Verbrauchern wie z.B. zwei Relais, zwei Lampen usw. verwendbar.

### M2A und M2B

Motoranschlüsse für Motor 2 (belastbar mit 1A) Natürlich auch zum steuern von anderen Verbrauchern wie z.B. zwei Relais, zwei Lampen usw. verwendbar.

### VCC (out)

Hier kann man die stabilisierten 5V abgreifen um eventuell weitere hardware damit zu versorgen. Der Pin sollte aber nicht über 50mA belastet werden, da sonst das Board beschädigt werden kann.

### RESET

Dies ist die Resetleitung des Controllers. Wird Sie kurzzeitig mit GND verbunden, so wird der Prozessor zurückgesetzt und das Programm startet von vorne!

### PC0 bis PC5

Diese Ports können, je nach Programmierung, als digitale Ein- und Ausgangsports aber auch als analoge Eingangsports genutzt werden. Analoge Ports eignen sich um Spannung zu messen.

### ADC6 und ADC7

Dies sind analoge Eingangsports, sie eignen sich zum Messen von Spannungen. ADC7 ist intern auf dem Board mit dem Potentiometer verbunden. Durch drehen des Potentiometers mit einem Schraubendreher ändert sich die Spannung. Die Spannung kann also im programm ermittelt werden, somit auch die Position des Potentiometers.

Wozu man diesen „Regler“ nutzt bleibt dem Anwender selbst überlassen.

PB0, PB3, PB4, PB5, PD0, PD1, PD2, PD3, PD4, & PD7

Dies sind digitale Ein- und Ausgangsports die zur freien Verwendung bereitstehen. Einige Ports haben zudem noch zusätzlich spezielle Fähigkeiten. Ausführlich wird dies im Datenblatt des Atmel Controllers beschrieben, welches ebenfalls auf der DVD abgelegt ist.

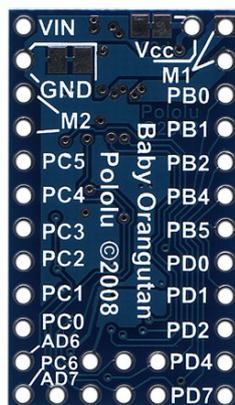
Zum Beispiel können PD0 und PD1 auch als serielle Schnittstelle genutzt werden. Über unser „USB zu Seriell Modul“ (siehe

[http://www.shop.robotikhardware.de/shop/catalog/product\\_info.php?cPath=73&products\\_id=225](http://www.shop.robotikhardware.de/shop/catalog/product_info.php?cPath=73&products_id=225) )

könnte man daraus zum Beispiel auch eine USB-Schnittstelle zum PC machen.

Port PD1 ist mit einer LED verbunden. Durch dessen Programmierung kann somit die LED ein und ausgeschaltet werden.

**Wichtig:** Die Ports PB4 und PB5 können zwar frei verwendet werden, sie werden aber auch während der Programmierung über ISP genutzt. Dadurch können diese während der Programmierung undefinierbar auf High oder Low gesetzt, achten Sie darauf das dies ihre Schaltung nicht stört Auch die Programmierung kann je nach Verwendung eventuell gestört werden wenn Sie diese Ports während der Programmierung an andere Komponenten anschließen.



Das Modul wird mit Stiftleisten zum einlöten geliefert. Die 6 polige Stiftleiste dient dabei zum Anschluss des Programmierkabels (Programmer). Wenn Sie mit [Bascom Basic](#) programmieren möchten, empfehlen wir ihnen den USB-Bascom Programmer

[http://www.shop.robotikhardware.de/shop/catalog/product\\_info.php?cPath=73&products\\_id=161](http://www.shop.robotikhardware.de/shop/catalog/product_info.php?cPath=73&products_id=161)

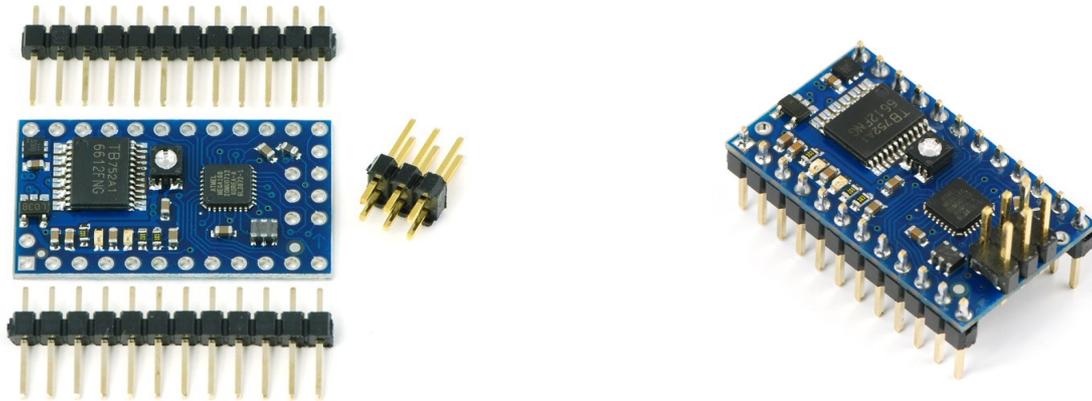
Da dieser aber einen 10 poligen Anschlussstecker besitzt, brauchen sie noch folgenden Adapter:

[http://www.shop.robotikhardware.de/shop/catalog/product\\_info.php?products\\_id=190](http://www.shop.robotikhardware.de/shop/catalog/product_info.php?products_id=190)

Möchten Sie dagegen in C mit dem Atmel Studio programmieren, so empfehlen wir ihnen den Programmer USB MK2:

[http://www.shop.robotikhardware.de/shop/catalog/product\\_info.php?products\\_id=201](http://www.shop.robotikhardware.de/shop/catalog/product_info.php?products_id=201)

Statt den Stiftleisten am rand können Sie bei bedarf natürlich auch Leitungen direkt anlöten.

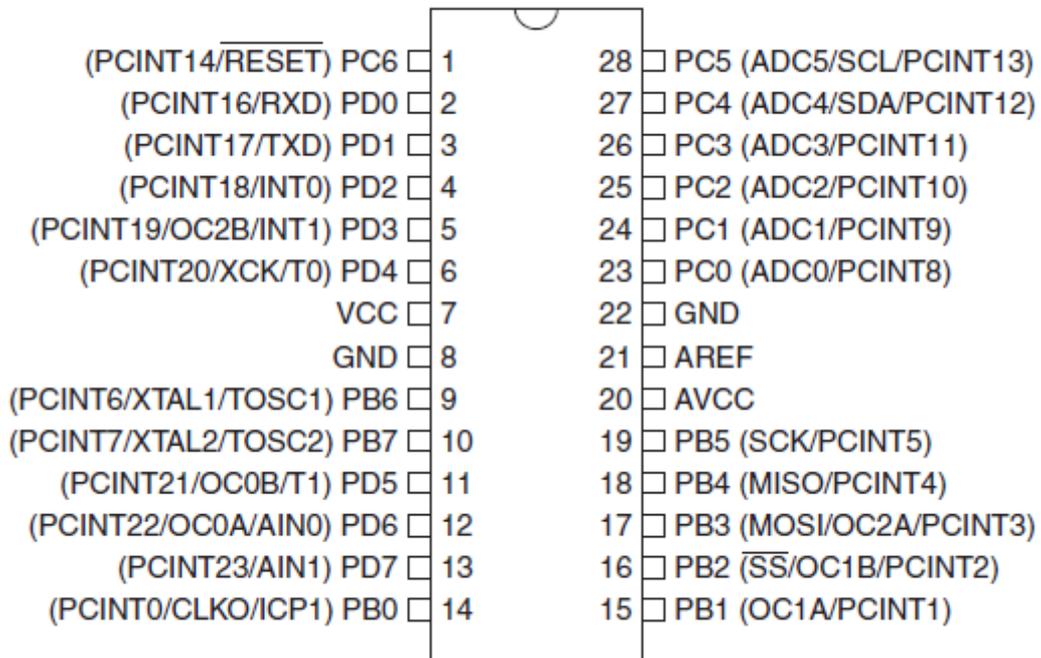


## Belegung der verschiedenen Ports

Pin	Orangutan Funktion	Hinweis /Alternative Funktion
PB0	Digitaler Ein- und Ausgang	Timer1 input capture (ICP1) divided system clock output (CLK0)
PB1	Digitaler Ein- und Ausgang	Timer1 PWM output A (OC1A)
PB2	Digitaler Ein- und Ausgang	Timer1 PWM output B (OC1B)
PB3	Motor 2 Steuerport	Timer2 PWM output A (OC2A) ISP programming line Damit läßt sich die Geschwindigkeit des Motors 2 einstellen
PB4	Digitaler Ein- und Ausgang	Frei verfügbar, wird aber auch zur Programmierung benötigt (MISO)
PB5	Digitaler Ein- und Ausgang	Frei verfügbar, wird aber auch zur Programmierung benötigt (SCK)
PB6	20 MHz Quarz	Hier ist Quarz angeschlossen, steht also nicht zur freien Verfügung
PB7	20 MHz Quarz	Hier ist Quarz angeschlossen, steht also nicht zur freien Verfügung
PC0	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 0 (ADC0)
PC1	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 1 (ADC1)
PC2	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 2 (ADC2)
PC3	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 3 (ADC3)
PC4	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 4 (ADC4) I2C/TWI input/output data line (SDA)
PC5	Analoger Eingangsport und digitaler Ein- und Ausgang	ADC input channel 5 (ADC5) I2C/TWI clock line (SCL)
PC6	RESET	Reset bin liegt intern auf High
PD0	Digitaler Ein- und Ausgang	Serielle Schnittstelle USART (RXD)
PD1	Digitaler Ein- und Ausgang	Serielle Schnittstelle USART (RXD) Rote LED auf dem Modul (high schaltet LED ein)
PD2	Digitaler Ein- und Ausgang	Externer Interrupt 0 (INT0)
PD3	Motor 2 Steuerport	Timer2 PWM Output B (OC2B)

PD4	Digitaler Ein- und Ausgang	USART external clock input/output (XCK) Timer0 external counter (T0)
PD5	Motor 1 Steuerport	Timer0 PWM output B (OC0B)
PD6	Motor 1 Steuerport	Timer0 PWM output A (OC0A)
PD7	Digitaler Ein- und Ausgang	
ADC6	Analoger Eingang	ADC input channel 6 (ADC6)
ADC7	Analoger Eingang	Verbunden mit Potentiometer auf dem Board ADC input channel 7 (ADC7)

**Die verschiedenen Funktionen der Controller Pin´s auf einen Blick:**



## Motor Verhalten bei Portzustand

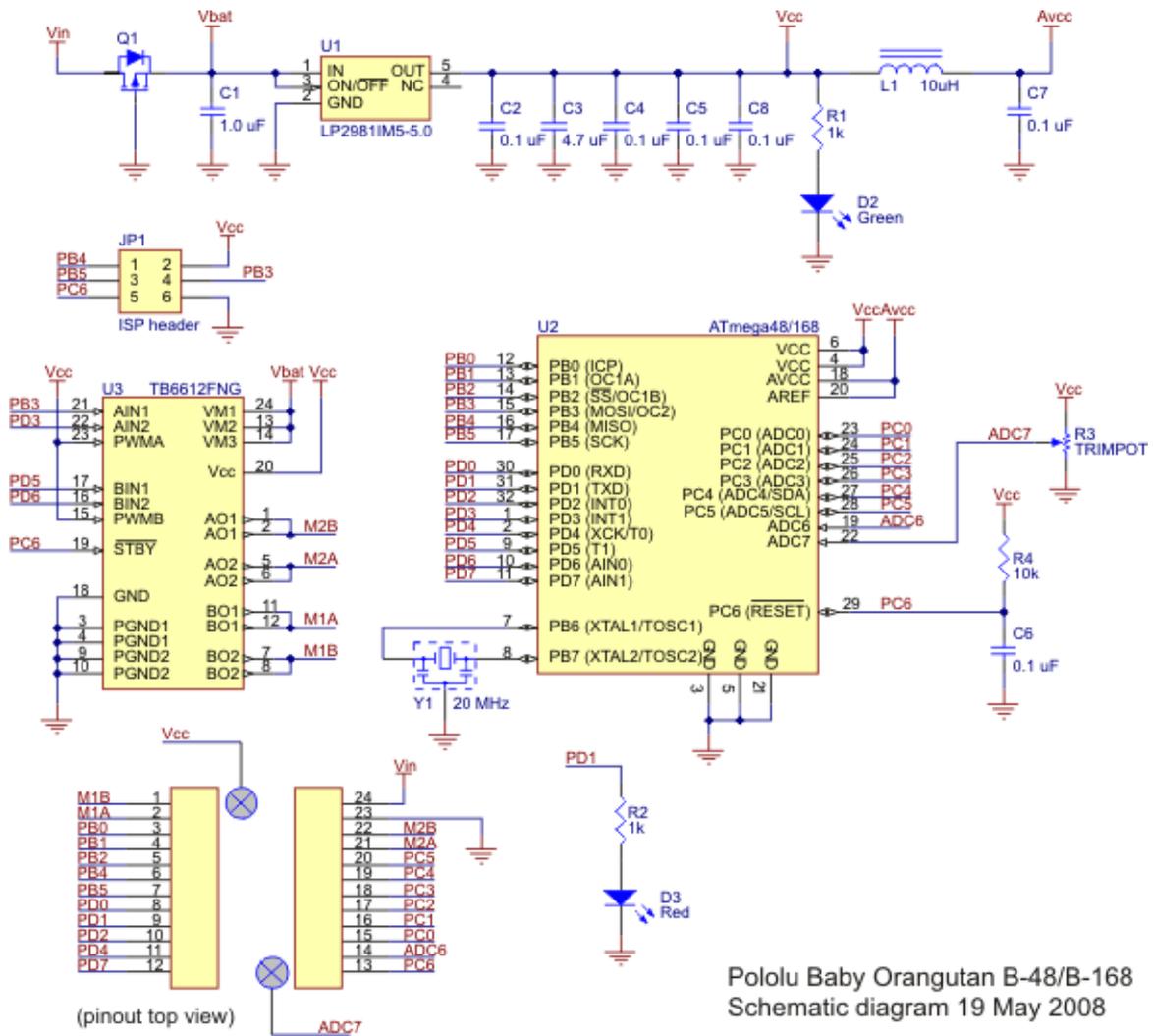
input		output		motor effect
PD5, PD3	PD6, PB3	M1A, M2A	M1B, M2B	
H	H	L	L	Bremse
L	H	L	H	Vorwärts
H	L	H	L	Rückwärts
L	L	OFF (high-impedance)		Leerlauf

Der Motor 1 wird kontrolliert über die PWM fähigen Timerausgänge PD5 (OC0B) und PD6 (OC0A). Durch entsprechende Timerprogrammierung (siehe auch Beispielprogramm) kann somit die Geschwindigkeit des Motors beliebig programmiert werden.

Der Motor 2 wird kontrolliert über die PWM fähigen Timerausgänge PD3 (OC2B) und PB3 (OC2A). Durch entsprechende Timerprogrammierung (siehe auch Beispielprogramm) kann somit die Geschwindigkeit des Motors beliebig programmiert werden.

## Schaltplan

Der Schaltplan gilt sowohl für das Orangutan B328 als auch dessen kleinere Varianten!



## Ein Bascom Beispielprogramm für Baby Orangutan B-328

Dieses einfache Beispiel bietet sich als erster Test an. Es macht nichts weiter als die LED auf dem Modul im Abstand von einer Sekunde blinken zu lassen.

```
'#####  
'LED-blinken_orangutan_B328  
'Dieses Programm demonstriert die Programmierung  
'des Mini Robot Boards Baby Orangutan B328  
'Bezug www.robotikhardware.de  
'Boardbeschreibung Roboternetz.de oder robotikhardware.de  
'Verwendeter Compiler Bascom  
'  
'Aufgabe:  
' Rote LED im Sekundentakt blinken lassen  
'  
'  
'Autor: Frank  
'Weitere Beispiele und Beschreibung der Hardware  
'unter http://www.Roboternetz.de bzw. robotikhardware.de  
'#####  
  
$prog &HFF , &HF6 , &HD9 , &HFC 'Standard Fusebits für Orangutan B328  
  
'Die üblichen Definitionen bei Standardprogrammen für Orangutan B-328  
$regfile = "m328pdef.dat"  
$crystal = 2000000 'Quarzfrequenz  
$hwstack = 32  
$framesize = 64  
$swstack = 32  
  
' ----- Anwendungsspezifische Configurationen -----  
  
Config Pind.1 = Output  
Red_led Alias Portd.1  
  
Do  
  Toggle Red_led  
  Wait 1  
Loop
```

## Motor ansteuern mit Baby Orangutan B-328

Dieses einfache Beispiel zeigt wie man die Geschwindigkeit der Motoren einstellt und diese beispielsweise langsam beschleunigt.

```
'#####  
'motorbeschleunigung_orangutan_b328.bas  
'Dieses Programm demonstriert die Programmierung  
'des Mini Robot Boards Baby Orangutan B-328  
'Bezug www.robotikhardware.de  
'Boardbeschreibung Roboternetz.de oder robotikhardware.de  
'Verwendeter Compiler Bascom  
'  
'Aufgabe:  
' Motor 1 wird langsam beschleunigt  
' Motor 1 wird langsam in andere Richtung beschleunigt  
' Motor 2 wird langsam beschleunigt  
' Motor 2 wird langsam in andere Richtung beschleunigt  
' Ende  
'  
'Autor: Frank  
'Weitere Beispiele und Beschreibung der Hardware  
'unter http://www.Roboternetz.de bzw. robotikhardware.de  
'#####  
  
$prog &HFF , &HF6 , &HD9 , &HFC           'Standard Fusebits für Orangutan B-328  
  
'Die üblichen Definitionen bei Standardprogrammen für Orangutan SV-328  
$regfile = "m328pdef.dat"  
$crystal = 2000000                        'Quarzfrequenz  
$hwstack = 32  
$framesize = 64  
$swstack = 32  
  
' ----- Anwendungsspezifische Configurationen -----  
  
'Ports benennen  
Red_led Alias Portd.1  
  
Motor1a Alias Portd.6  
Motor1b Alias Portd.5  
Motor2a Alias Portd.3  
Motor2b Alias Portb.3  
  
'Variablen  
Dim I As Integer  
  
Config Pind.1 = Output  
Config Pinb.3 = Output  
Config Pind.3 = Output  
Config Pind.5 = Output  
Config Pind.6 = Output
```

```
' PWM Register setzen
' see the ATmega48/168/328P datasheet for detailed register info
' configure for inverted PWM output on motor control pins
Tccr0a = &HF3
Tccr2a = &HF3
' use the system clock / 8 (2.5 MHz) as the timer clock
Tccr0b = &H02
Tccr2b = &H02

' Motor 2 beschleunigen
I = 0
Do
  Ocr2a = 0
  Ocr2b = I
  Waitms 500
  I = I + 5
Loop Until I > 255

Wait 2

' Motor 2 in andere Richtung beschleunigen
I = 0
Do
  Ocr2a = I
  Ocr2b = 0
  Waitms 500
  I = I + 5
Loop Until I > 255

Ocr2a = 0
Ocr2b = 0

' Motor 1 beschleunigen
I = 0
Do
  Ocr0a = 0
  Ocr0b = I
  Waitms 500
  I = I + 5
Loop Until I > 255

Wait 2

' Motor 1 in andere Richtung beschleunigen
I = 0
Do
  Ocr0a = I
  Ocr0b = 0
  Waitms 500
  I = I + 5
Loop Until I > 255

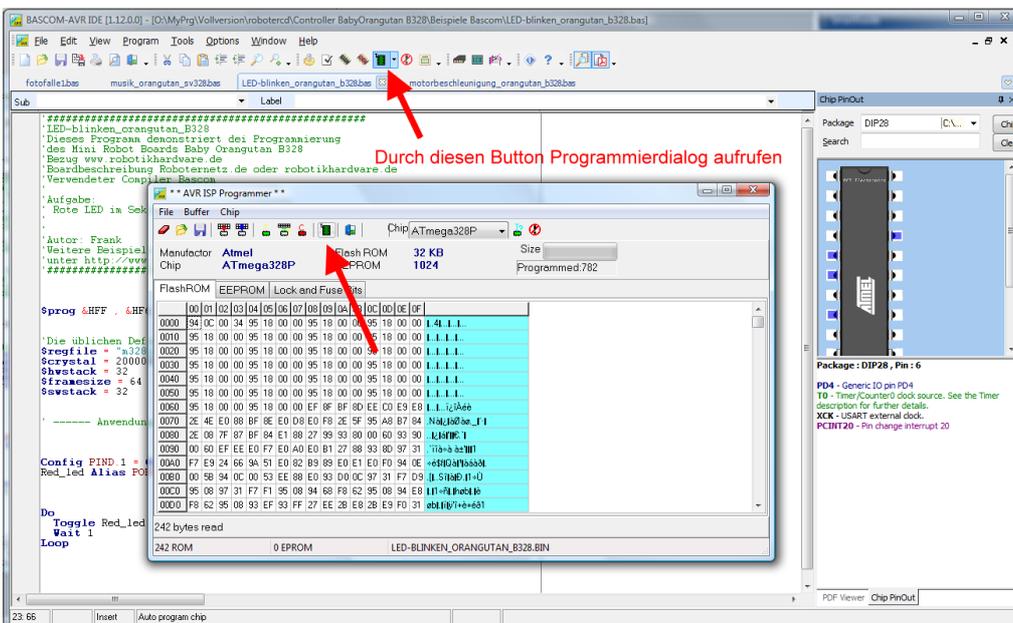
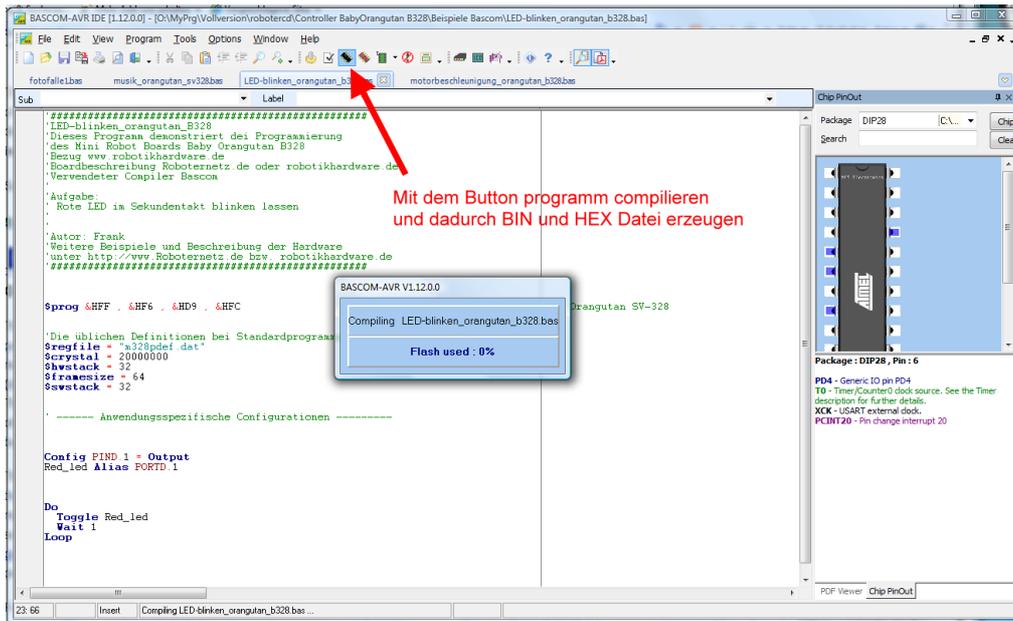
Ocr0a = 0
Ocr0b = 0

End
```

Habt ihr weitere interessante Beispielprogramme, so sendet diese einfach an [support@robotikhardware.de](mailto:support@robotikhardware.de). Wir packen diese dann auf unsere DVD! Bitte kurze Doku im Quelltext als Kommentar aufnehmen.

## Programme compilieren und übertragen

Wer schon mal mit einem Controller gearbeitet hat, zum Beispiel dem beliebten Einsteigerboard [RN-Control](#) von [robotikhardware.de](#), wird natürlich bereits wissen wie man Programme compiliert und in das Modul überträgt. Für Einsteiger möchte ich es hier noch mal kurz in Bildform darstellen.



## Empfehlenswerter Hardware

### Programmiersprache

Wenn Sie unter Bascom Basic programmieren wollen, dann empfehlen wir Ihnen folgenden Compiler: **Bascom Basic**. Bascom Basic ist eine sehr leicht zu erlernende Programmiersprache mit sehr starken Funktionen. Oft kann man hier in einigen wenigen Programmzeilen die selben Aufgaben lösen, wie bei einem 2-seitigen langem schwer lesbarem C-Source-Code-Programm. Mit jedem Modul, das Sie über Robotikhardware bestellen, erhalten Sie bereits eine verwendbare Demo-Version, die für kleinere Programme ausreicht. Für größere Programme sollten Sie die Vollversion erwerben.



### Programmer

Wenn Sie unter Bascom programmieren, dann empfehlen wir Ihnen den **Bascom-USB-Programmer**, den Sie ebenfalls über Robotikhardware.de beziehen können. Dieser Programmer ist besonders einfach zu bedienen und kann nahezu alle Atmel AVR Controller programmieren. Firmware Updates sind bei diesem Programmer, im Gegensatz zu Billig-Programmieren, nicht notwendig!



Sie benötigen dazu nur den **10 auf 6 Adapter**, den Sie ebenfalls im Shop erhalten.

Wenn Sie auch unter C oder unter Assembler programmieren wollen, dann empfehlen wir Ihnen den Programmer USB Atmel MK2. Dieser wird, im Gegensatz zum Bascom Programmer, auch von der kostenlosen Entwicklungsumgebung **AVR-Studio** unterstützt.



### Optional Usb-Schnittstelle

Wenn Sie auch Daten oder Debug-Informationen zwischen Controllermodul und PC austauschen möchten, dann empfehlen wir Ihnen als Zusatz das „USB zu Seriell Modul“. Dieses Modul lässt sich an die RX und TX Leitung anschließen und bietet dann eine USB-Schnittstelle, die einen virtuellen COM-Port am PC bereitstellt. Damit lassen sich dann Daten wie über eine normale serielle Schnittstelle zwischen Modul und Computer hin und her schicken.

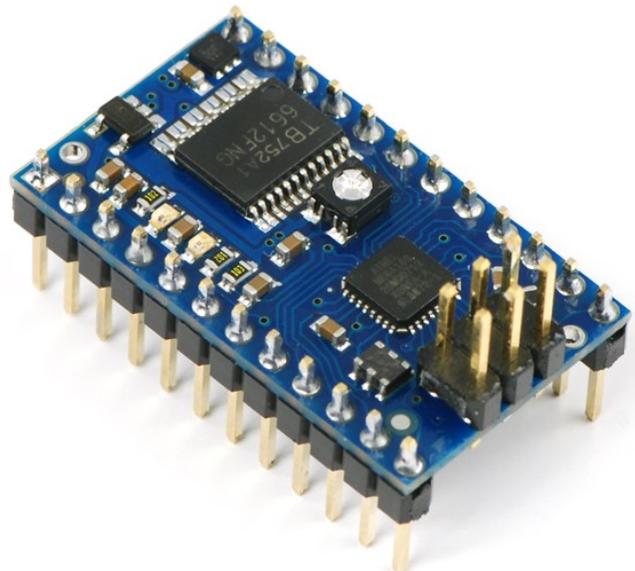


## Englische Originalanleitung vom Hersteller

**The Baby Orangutan B-328 is a very compact but complete robot controller, packing a high-performance AVR microcontroller and two motor drive channels in the same 24-pin form factor as competing units that include just a microcontroller. You can connect your battery, sensors, and motors directly to this small module to make a miniature robot, or you can use the Baby Orangutan as an auxiliary controller in larger robots. This version includes Atmel's new ATmega328P microcontroller with 32 KB of program memory and 2 KB of RAM.**

### Overview

The Baby Orangutan is a complete control solution for small robots, all packed into a tiny 1.2" x 0.7" 24-pin DIP package. Its compact design eliminates bulkier components such as the LCD and switches while retaining the most essential features of the Orangutan robot controller line: a programmable ATmega328P AVR microcontroller and a dual H-bridge for direct control of two DC motors. This integrated motor driver sets the Baby Orangutan B apart from similarly-sized microcontroller boards from other manufacturers. Two on-board indicator LEDs, a trimmer potentiometer, a 20 MHz resonator, and reverse battery protection round out the basic hardware features of the Baby Orangutan.



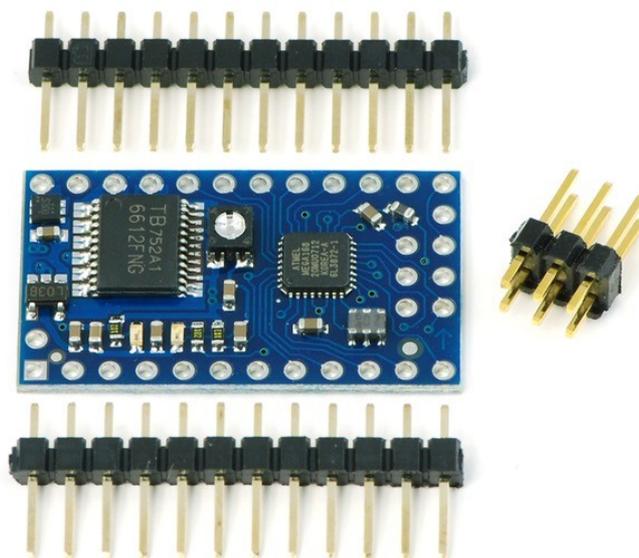
The removal of the larger Orangutan components also allows for a significantly improved manufacturing process that allows Pololu to offer the Baby Orangutan B at a very affordable price. Because the Orangutans are based on Atmel's powerful AVR microcontrollers, the Orangutans deliver significantly higher performance than other similar controller boards. The availability of free development software, such as Atmel's AVR Studio IDE and the WinAVR GCC C/C++ compiler and bascom-compiler.

## Specifications & On-Board Hardware

- overall unit dimensions: 1.2" x 0.7"
- input voltage: 5-13.5 V (15 V absolute maximum)
- two bidirectional motor ports can deliver  $\sim 1$  A continuous (3 A peak) per channel
- programmable 20 MHz Atmel ATmega328P AVR microcontroller (32 KB flash, 2 KB RAM, 1 KB EEPROM)
- 18 user I/O lines, 16 of which can be used for digital I/O and 8 of which can be used as analog input channels
- 1 user LED
- user potentiometer tied to ADC7
- 20 MHz external resonator
- pinout is compatible with Orangutan LV-168, so the same code will generally work on all of these devices

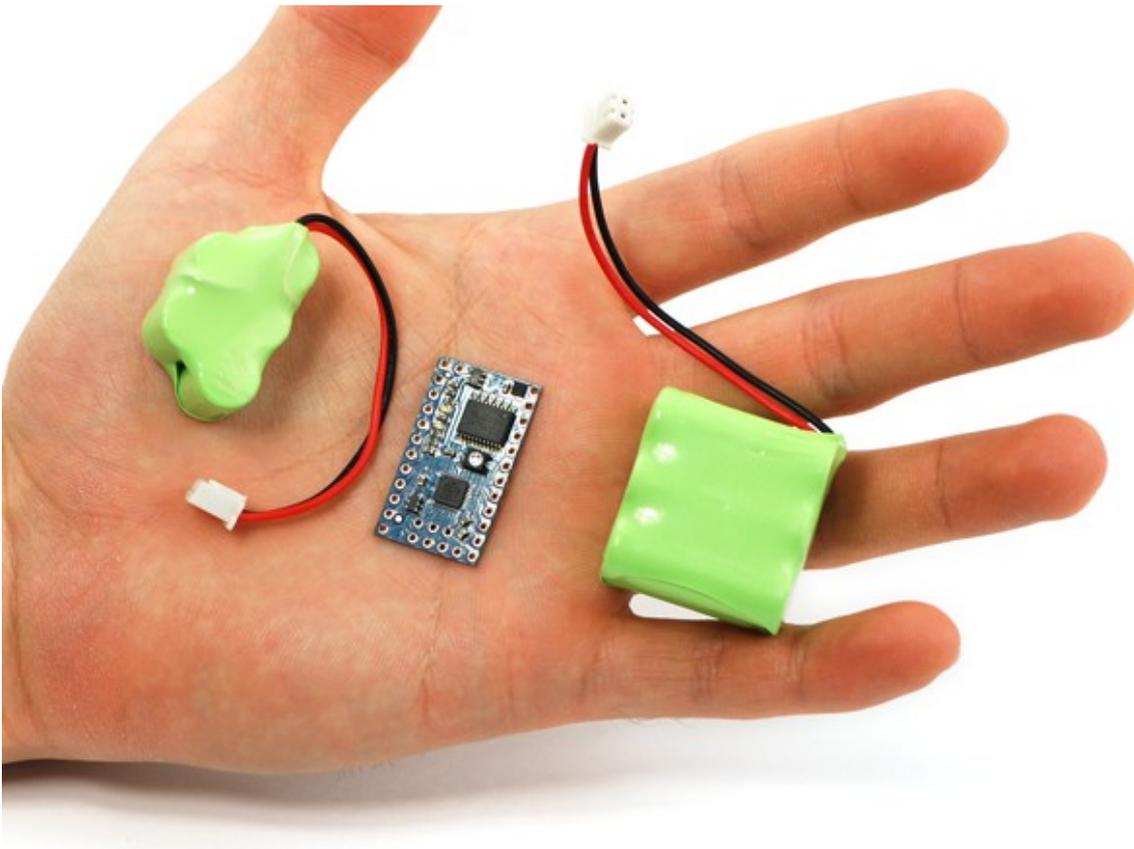
## Included components

The compact module can be used as a DIP component on breadboards or prototyping boards, or the pin-less versions can be used for space-constrained installations in miniature robots. The 0.1" header pins are included with the Baby Orangutan B but are not soldered in. Power pins, one of the motor outputs, and several I/O lines are all accessible from one side to enable use of the Baby Orangutan as a single in-line pin (SIP) package for applications that do not require all of the I/O lines. The small size and low cost of the Baby Orangutan makes it a perfect option for primary control of small robots or for auxiliary control on larger robots.

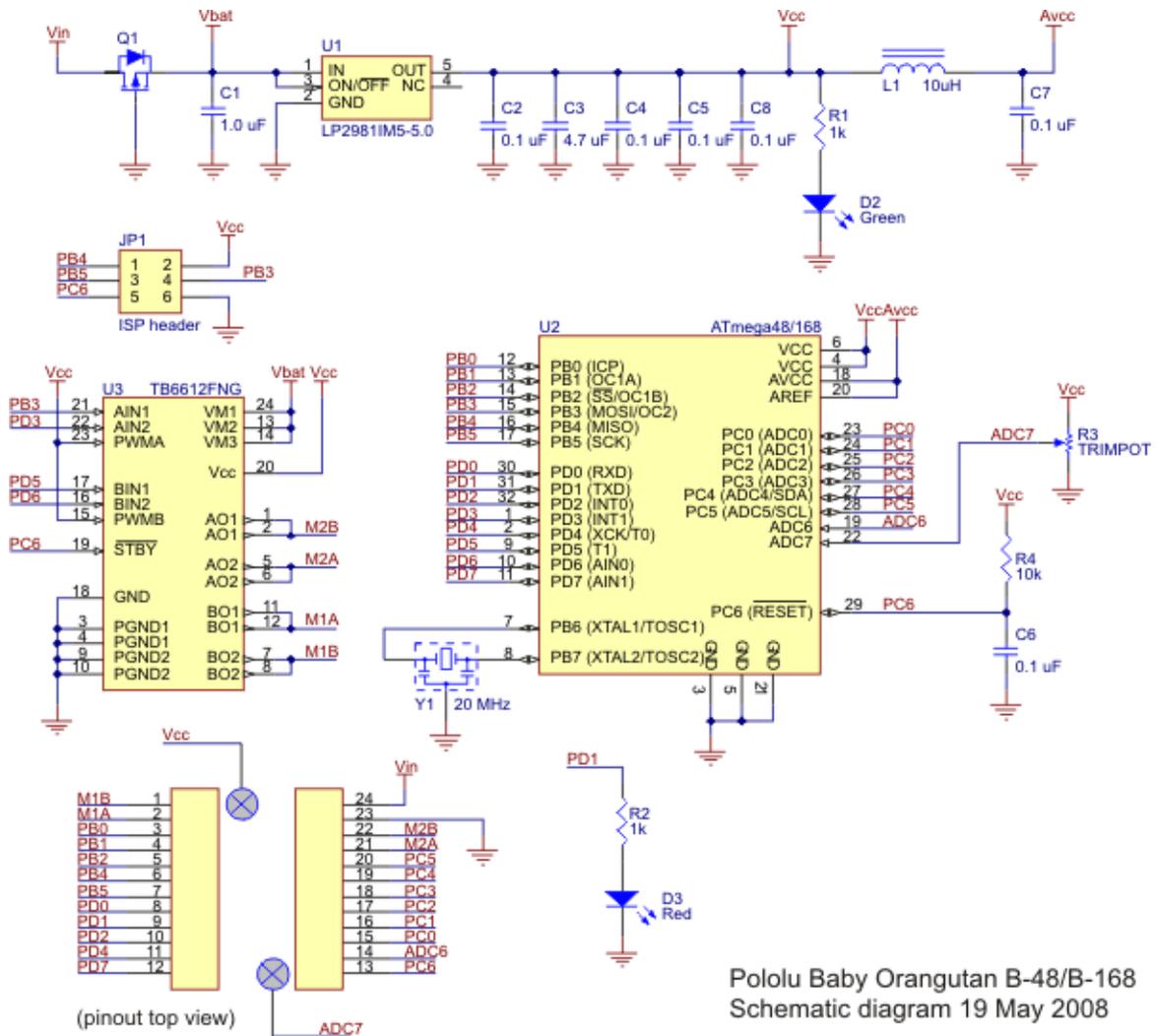


## General specifications

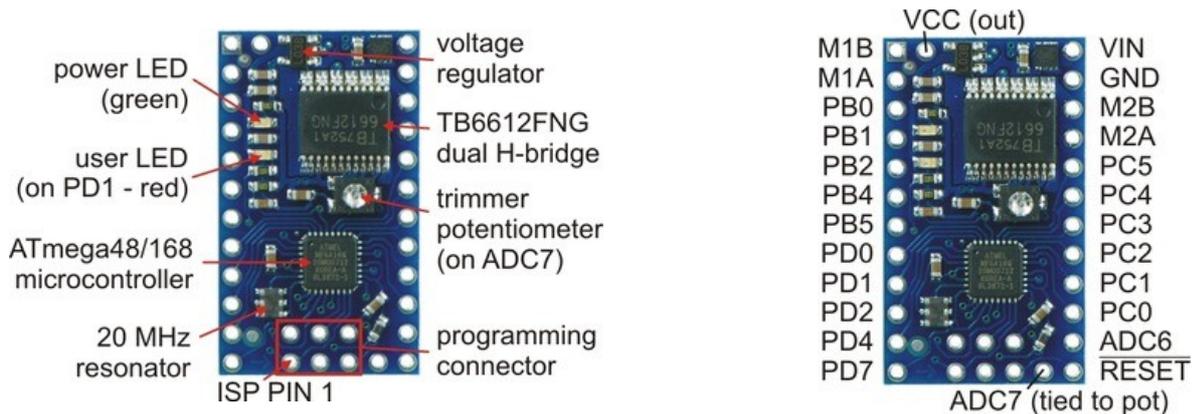
- Processor: ATmega328P @ 20 MHz
- RAM size: 2048 bytes
- Program memory size: 32 Kbytes
- Motor driver: TB6612FNG
- Motor channels: 2
- User I/O lines: 18 (16 digital / 8 analog)
- Max current on a single I/O: 40 mA
- Minimum operating voltage: 5 V
- Maximum operating voltage: 13.5 V
- Continuous output current per channel: 1 A
- Peak output current per channel: 3 A
- Maximum PWM frequency: 80 kHz
- Reverse voltage protection?: Y
- External programmer required?: Y
- Size: 1.20" x 0.70"
- Weight: 1.5 g



Pololu Baby Orangutan B-48/B-168/B-328 schematic diagram.



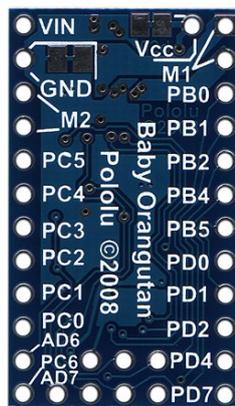
The Baby Orangutan contains a programmable **ATmega328P AVR** microcontroller, a TB6612FNG dual H-bridge for direct control of two DC motors, a 10k user trimmer potentiometer (connected to ADC7), a green power LED, a red user LED (connected to PD1), a 20 MHz resonator, and a reverse-battery-protection MOSFET, all contained in a tiny 1.2" x 0.7" 24-pin DIP package. Power pins, one of the motor outputs, and several I/O lines are all accessible from one side to enable use of the Baby Orangutan as a single in-line pin (SIP) package for applications that do not require all of the I/O lines.



- VIN should be from 5 to 13.5 V, with an absolute maximum of 15 V.
- RESET can be brought low to reset the controller, but it can otherwise be left disconnected (it is internally pulled high). This pin is labeled as PC6 in the ATmega48/168/328 datasheet (and on the Baby Orangutan silkscreen).
- Vcc can be used to tap into the Baby Orangutan's regulated 5V line. This line can supply a total of around 100 mA at 5 V, but thermal dissipation limits the total Vcc current to around 50 mA at 13.5 V. Note that attempting to pull too much current from Vcc could permanently damage the Baby Orangutan's voltage regulator.
- M1A & M1B are the outputs used to drive motor 1. These outputs can supply around 1 A continuous (3 peak).
- M2A & M2B are the outputs used to drive motor 2. These outputs can supply around 1 A continuous (3 peak).
- PC0 – PC5 can be used as both analog inputs and digital I/O lines
- ADC6 & ADC7 are dedicated analog inputs. Note that ADC7 is internally connected to the 10k user trimmer potentiometer.
- PB0, PB3, PB4, PB5, PD0, PD1, PD2, PD3, PD4, & PD7 are digital I/O lines with alternate functions determined by the AVR hardware peripherals to which they connect. For example, PD0 and PD1 connect to the ATmega48/168's UART and can be configured to function as RX and TX, respectively. Note that PD1 is internally connected to the red user LED, which may limit its ability to be used as an input (if the source cannot drive the PD1 hard enough, the voltage will be pulled below the AVR's high threshold by the LED-resistor circuit ).

**Warning:** Pins PB4 and PB5 are used as ISP programming pins in addition to digital user I/O lines. Be careful not to connect anything to these pins that might interfere with programming (e.g. large capacitance or an external device that could drive those lines during programming). Similarly, don't connect anything to those lines that might behave unexpectedly when they are driven during programming (e.g. if you use these lines as inputs to a motor driver IC, it could drive your motors in strange and potentially dangerous ways during programming).

You can tap into the Baby Orangutan's regulated 5V Vcc line using the pin labeled "Vcc" or either of the two pads on the bottom of the board directly to the left of this pin. You can tap into the Baby Orangutan's ground using the two pads on the bottom of the board directly to the right of the "GND" pin.



The Baby Orangutan ships with 0.1" header pins as shown in the left picture below: two 12×1 strips and one 3×2 ISP programming header. Both 12×1 strips can be soldered in to allow the module to be used as a DIP component on breadboards or prototyping boards, or a single 12×1 strip can be soldered in to allow the module to be used as a single in-line pin (SIP) component (since power pins, one of the motor outputs, and several I/O lines are all accessible from one side). The header pins can be left off and wires can be directly soldered to the Baby Orangutan for space-constrained installations.



If you solder in the 3×2 ISP header pins, the solder connections should be made along the bottom side of the board so the pin connections are available from the top side of the board, as shown in the right picture above. If you solder the ISP pins to the wrong side of the board, your programmer's ISP cable will not be able to connect correctly to the Baby Orangutan.

Pin	Orangutan Function	Notes/Alternate Functions
PB0	digital I/O	Timer1 input capture (ICP1) divided system clock output (CLK0)
PB1	digital I/O	Timer1 PWM output A (OC1A)
PB2	digital I/O	Timer1 PWM output B (OC1B)
PB3	M2 control line	Timer2 PWM output A (OC2A) ISP programming line
PB4	digital I/O	Caution: also an ISP programming line
PB5	digital I/O	Caution: also an ISP programming line
PB6	20 MHz resonator input	not accessible to the user
PB7	20 MHz resonator input	not accessible to the user
PC0	analog input and digital I/O	ADC input channel 0 (ADC0)
PC1	analog input and digital I/O	ADC input channel 1 (ADC1)
PC2	analog input and digital I/O	ADC input channel 2 (ADC2)
PC3	analog input and digital I/O	ADC input channel 3 (ADC3)
PC4	analog input and digital I/O	ADC input channel 4 (ADC4) I2C/TWI input/output data line (SDA)
PC5	analog input and digital I/O	ADC input channel 5 (ADC5) I2C/TWI clock line (SCL)
PC6	RESET pin	internally pulled high; active low digital I/O disabled by default
PD0	digital I/O	USART input pin (RXD)
PD1	digital I/O	connected to red user LED (high turns LED on) USART output pin (TXD)
PD2	digital I/O	external interrupt 0 (INT0)
PD3	M2 control line	Timer2 PWM output B (OC2B)
PD4	digital I/O	USART external clock input/output (XCK) Timer0 external counter (T0)
PD5	M1 control line	Timer0 PWM output B (OC0B)
PD6	M1 control line	Timer0 PWM output A (OC0A)
PD7	digital I/O	
ADC6	dedicated analog input	ADC input channel 6 (ADC6)
ADC7	dedicated analog input	connected to user trimmer potentiometer ADC input channel 7 (ADC7)

## Motor Driver Truth Table

input		output		motor effect
PD5, PD3	PD6, PB3	M1A, M2A	M1B, M2B	
H	H	L	L	brake
L	H	L	H	"forward"
H	L	H	L	"reverse"
L	L	OFF (high-impedance)		coast

Motor 1 is controlled by pins PD5 and PD6, and motor 2 is controlled by PD3 and PB3. These pins are connected to the ATmega48/168's four eight-bit hardware PWM outputs, which allows you to achieve variable motor speeds through hardware timers rather than software. This frees the CPU to perform other tasks while motor speed is automatically maintained.

The suggested procedure for using hardware PWM outputs to control the motors is as follows:

1. Make the four motor control pins outputs and drive them high; this drives all four motor outputs low.
2. Configure Timer0 and Timer2 to use a prescaler of 8, which results in a PWM frequency of  $20 \text{ MHz}/8/256 = 9.8 \text{ kHz}$ . Set these timers for inverted PWM mode output on both OCxA and OCxB, meaning that these PWM pins are set on timer compare match and cleared on timer overflow. This results in negative PWM pulses with duty cycles determined by registers OCR0A, OCR0B, OCR2A, and OCR2B.

3. You can command motor 1 to drive "forward" at a speed ranging from 0 – 255 by setting OCR0B = speed and holding fixed OCR0A = 0. You can command motor 1 to drive "reverse" at a speed ranging from 0 – 255 by setting OCR0A = speed and OCR0B = 0. During the period where the two input pins have opposite values, the motor drives at full speed. During the period where the two inputs have the same value (high), the motor brakes. Cycling between drive and brake and high frequency results in variable motor speed that changes as a function of PWM duty cycle. Analogous results can be obtained for motor 2 using OCR2A and OCR2B. (Note that the concept of "forward" is arbitrary as simply flipping the motor leads results in rotation in the opposite direction.)

Using these PWM settings, OCR0B = 255 is equivalent to holding PD5 low while OCR0A = 0 is equivalent to holding PD6 high. As you can see from the truth table above, in this state M1B connects to your battery's positive terminal and M1A connects to ground. Decreasing OCR0B to something less than 255 decreases the percentage of time PD5 is low, causing M1B to alternate between VIN and GND (and hence causing motor 1 to alternate between drive and brake). Similarly, OCR2B = 255 is equivalent to holding PD3 low while OCR2A = 0 is equivalent to holding PB3 high. In this state, M2B connects to your battery's positive terminal and M2A connects to ground.

**Bezugsquelle für dieses Modul:** <http://robotikhardware.de/>

**Modul-Hersteller:** Pololu/USA

**Aktuelle Infos/Ergänzungen:**

Pololu/USA - <http://www.pololu.com/catalog/product/1220>

**Aktuelle deutsche Doku:** <http://robotikhardware.de/>

**Übersetzung/ Deutsche Doku:** <http://robotikhardware.de/>

**Hilfreiches Technik-Forum:** <http://www.roboternetz.de/community/>

**Passende Bibliotheken für C-Programmierer:**

<http://www.pololu.com/catalog/product/1220/resources>

#### **Hinweise zur beschränkten Haftung**

Das Modul ist nicht für Geräte geeignet die direkt oder indirekt medizinischen, gesundheitlichen oder anderen Zwecken, bei denen ein Ausfall / Fehler zu Schäden an Personen oder Sachwerten führen würde. Soll das Modul für einen solchen Fall eingesetzt werden, so ist der Kunde für den Test und alle Zulassungen selbst verantwortlich. Robotikhardware.de übernimmt dafür keinerlei Haftung. Die Haftung beschränkt sich in allen Fällen auf den Austausch des fehlerhaften Moduls. Module die durch fehlerhaften Anschluß / Bedienung beschädigt wurden, können nicht ersetzt werden. Das Modul ist nicht für den kommerziellen Einsatz gedacht, sondern vornehmlich für Ausbildung, Entwicklung, Experimente und Modellbau. Eine andere Anwendung erfolgt auf eigenes Risiko. Die von uns angegebenen Daten sind weitestgehend Herstellerangaben für die wir keine Gewähr übernehmen können.

#### **Haftung, EMV-Konformität**

Alle Teile der Schaltung wurden sorgfältig geprüft und getestet. Trotzdem übernimmt der Entwickler/Vertrieber keine Garantie dafür daß alles einwandfrei funktioniert. Insbesondere wird keine Haftung für Schäden, die durch Nachbau, Inbetriebnahme etc. der hier vorgestellten Schaltungen entstehen. Derjenige, der den Bausatz zusammenbaut oder das Modul in eigene Schaltungen/Gehäuse integriert, gilt als Hersteller und ist damit selbst für die Einhaltung der geltenden Sicherheits- und EMV-Vorschriften verantwortlich.

Wenn nicht anders angegeben handelt es sich generell bei allen Bausätzen, Modulen und Boards um "nicht CE-geprüfte" Komponenten und sind konzipiert für den Einbau in Geräte oder Gehäuse. Bei der Anwendung müssen die CE-Normen eingehalten werden. Hierfür ist der Käufer verantwortlich.

Für Schäden die durch fehlerhaften Aufbau entstanden sind, direkt oder indirekt, ist die Haftung generell ausgeschlossen. Schadensersatzansprüche, gleich aus welchem Rechtsgrund, sind ausgeschlossen, soweit nicht vorsätzliches oder grob fahrlässiges Handeln vorliegt. Sofern wir haften, umfaßt unsere Haftung nicht solche Schäden, die nicht typischerweise erwartet werden konnten. Haftung und Schadensersatzansprüche sind auf den Auftragswert / Bauteilwert beschränkt. Bei der Lieferung von Fremdprodukten als auch Software gelten über diese Bedingungen hinaus die besonderen Lizenz- oder sonstigen Bedingungen des Herstellers.