

## Arduino Uno R3



**Artikel-Nr.:** A000066

**Hersteller:** Arduino

**EAN:** 8058333490090

### Übersicht

Der Arduino Uno ist ein Mikrocontroller Board, basierend auf dem ATmega328 (datasheet). Er besitzt 14 digitale Input/Output Pins (von denen 6 als PWM Output nutzbar), 6 analoge Inputs, einen 16 MHz Keramik Resonator, eine USB Verbindung, einen Power Jack, einen ICSP Header und einen Reset Button. Er besitzt alles Nötige um den Mikrocontroller zu betreiben. Um loszulegen muss man den Arduino Uno lediglich per USB Kabel an einen Computer anschließen oder ihn mit einem AC-to-DC Netzteil oder einer Batterie verbinden. Der Uno unterscheidet sich von allen vorherigen Arduino Boards darin, dass er keinen extra FTDI USB-to-Serial Treiber Chip nutzt, sondern einen als USB-To-Serial Converter programmierten ATmega16U2 (in den Versionen bis R2 Atmega8U2). Revision 2 des Uno Boards besitzt einen Widerstand, welcher die 8U2 HWB line auf Ground zieht. Das macht es einfacher in den DFU mode? zu gelangen.

Revision 3 des Boards besitzt die folgenden Neuerungen:

- 1.0 pinout: SDA und SCL Pins wurden in der Nähe des AREF Pin hinzugefügt, sowie zwei weitere in der Nähe des Reset Pin. Einer davon ist der IOREF, der es Shields ermöglicht die Spannung des Arduino Boards zu nutzen. In Zukunft werden Shields sowohl mit Boards kompatibel sein, welche den AVR nutzen und mit 5V betrieben werden, sowie mit dem Arduino Due, welcher mit 3.3V betrieben wird. Der Zweite Pin wurde für zukünftige Neuerungen hinzugefügt und besitzt zurzeit noch keine Aufgabe.
- Verbesserte Reset Schaltung.
- Der Atmega 16U2 ersetzt den 8U2.

Uno bedeutet Eins auf Italienisch und steht für den baldigen Release der Arduino Version 1.0. Der Uno und die Version 1.0 werden die Referenz Version des Arduino für weitere Entwicklungen sein. Der Uno ist der Neueste in einer Reihe von USB Arduino Boards und das Referenz Modell für die Arduino Plattform; Ein Vergleich des Uno mit früheren Versionen finden Sie im Index of Arduino boards?.

### Zusammenfassung

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

## Schaltplan & Referenz Design

EAGLE Dateien: [arduino-uno-Rev3-reference-design.zip](#) (Funktioniert mit EAGLE 6.0 und neuer)

Schaltplan: [arduino-uno-Rev3-schematic.pdf](#)

Achtung: Das Arduino Referenz Design ist für den Betrieb mit einem Atmega8, Atmega168 oder Atmega328 ausgelegt. Das aktuelle Modell nutzt einen Atmega328. Im Schaltplan ist für Referenzzwecke ein Atmega8 eingezeichnet. Alle drei Prozessoren besitzen jedoch die gleiche Pin Konfiguration.

## Stromversorgung

Der Arduino Uno kann entweder über eine USB Verbindung oder über ein externes Netzteil mit Strom versorgt werden. Die Stromquelle wird automatisch ausgewählt.

Für eine externe Versorgung kann entweder ein AC-to-DC Netzteil oder eine Batterie genutzt werden. Für die Versorgung mit einem Netzteil muss dieses einen 2,1mm center-positiv Stecker besitzen, der mit der Strombuchse auf dem Board verbunden wird. Anschlüsse einer Batterie werden mit dem Gnd und dem Vin Pin des Power Connectors verbunden.

Das Board kann mit einer externen Spannung von 6 bis 20 Volt versorgt werden. Bei weniger als 7V Versorgungsspannung, kann es jedoch sein dass der 5V Pin weniger als fünf Volt bereitstellt und das Board instabil wird. Wenn mehr als 12V angelegt werden, kann der Spannungsregler überhitzen und das Board beschädigen. Der empfohlene Spannungsbereich liegt daher bei 7-12V.

Die Power Pins:

- VIN. An diesem Pin liegt die Input Spannung des Arduino an, wenn eine externe Stromquelle genutzt wird (anstatt der 5V einer USB Verbindung oder einer anderen regulierten Stromquelle). Sie können an diesen Pin Spannung anlegen oder, wenn eine externe Stromquelle mit der Strombuchse verbunden ist, Spannung abgreifen.
- 5V. An diesem Pin liegen die regulierten 5V vom Spannungsregler des Boards an. Das Board kann entweder über die DC Strombuchse mit Strom versorgt werden (7 - 12V), die USB Verbindung (5V) oder über den VIN Pin des Boards (7 - 12V). Eine Versorgung direkt über die 5V oder 3.3V Pins umgeht den Spannungsregler und kann das Board beschädigen. Es wird daher nicht empfohlen.
- 3V3. Eine Spannung von 3.3 Volt, die vom auf dem Board integrierten Spannungsregler bereitgestellt wird. Der maximale Output Strom liegt bei 50 mA.
- GND. Ground Pins (Erdung).
- IOREF. An diesem Pin liegt die Referenzspannung an, mit welcher der Mikrocontroller arbeitet. Ein korrekt konfiguriertes Shield kann die Spannung des IOREF nutzen um die richtige Stromquelle auszuwählen oder die Spannungsregler schalten um die Outputs mit 5V oder 3.3V zu versorgen.

## Speicher

Der ATmega328 besitzt 32 KB Speicher (von denen 0.5 KB vom Arduino Bootloader belegt sind). Er verfügt außerdem über 2 KB SRAM und 1 KB EEPROM, welcher mit der EEPROM library) ausgelesen und beschrieben werden kann.

## Input und Output

Jeder der 14 digitalen Pins des Arduino kann entweder als Input oder Output genutzt werden. Dafür stehen die Funktionen `pinMode()`, `digitalWrite()` und `digitalRead()` zur Verfügung. Sie arbeiten mit einer Spannung von 5 Volt. Jeder Pin kann einen maximalen Strom von 40mA bereitstellen oder aufnehmen und besitzt einen Pull-Up Widerstand von 20-50 kOhm, welcher by default nicht verbunden ist. Zusätzlich gibt es Pins für spezielle Funktionen:

- Serial: 0 (RX) und 1 (TX). Mit diesen Pins können TTL serielle Daten empfangen (RX) oder übertragen (TX) werden. Diese Pins sind mit den zugehörigen Pins des ATmega8U2 USB-to-TTL Serial Chip verbunden.
- External Interrupts: 2 and 3. Diese Pins können so konfiguriert werden, dass sie bei einem niedrigen Wert, einem Anstieg oder Fall, oder einer Änderung des Wertes einen Interrupt auslösen. Für mehr Informationen beachten Sie die Funktion `attachInterrupt()`.
- PWM: 3, 5, 6, 9, 10, and 11. Diese Pins verfügen über einen 8-Bit PWM Output, welcher über die Funktion `analogWrite()` gesteuert werden kann.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Diese Pins unterstützen SPI Kommunikation unter Verwendung der SPI library.
- LED: 13. Auf dem Board befindet sich eine LED, welche mit dem Pin 13 verbunden ist. Wird der Pin HIGH geschaltet, geht die LED an und wird er LOW geschaltet, geht sie aus.

Der Uno verfügt über 6 Analog Inputs. Sie tragen die Namen A0 bis A5 und besitzen jeweils eine Auflösung von 10 Bit (also 1024 Abstufungen). Standardmäßig messen sie von Erdung bis 5 Volt. Die Obergrenze lässt sich jedoch mit Hilfe des AREF Pins und der Funktion `analogReference()` ändern. Zusätzlich besitzen einige der Pins über spezielle Funktionen:

- TWI: A4 oder SDA Pin und A5 oder SCL Pin. Sie unterstützen TWI Kommunikation unter Verwendung der Wire Library.

Auf dem Board befinden sich darüber hinaus noch folgende Pins:

- AREF. Hier liegt die Referenz Spannung für die analogen Inputs an. Er wird unter Verwendung der Funktion `analogReference()` genutzt.
- Reset. Wird diese Leitung LOW gesetzt, wird der Mikrocontroller zurückgesetzt. Meistens wird dies für Reset Buttons auf Shields genutzt, wegen welchen man den Reset Button des Boards nicht mehr erreichen kann.

Für weitere Informationen beachten Sie auch [mapping between Arduino pins and ATmega328 ports?](#). Das Mapping des ATmega8, 168 sowie 328 ist identisch.

## Kommunikation

Der Arduino Uno besitzt eine Vielzahl von Möglichkeiten um mit einem Computer, einem anderen Arduino, oder einem anderen Mikrocontroller zu kommunizieren. Der ATmega328 verfügt an den Digital Pins 0 (RX) und 1 (TX) über UART TTL (5V) serielle Kommunikation. Der ATmega16U2 leitet diese serielle Kommunikation über eine USB Verbindung und stellt sie über einen virtuellen Com Port für die Software des Computers bereit. Die 16U2 Firmware nutzt den Standard USB COM Treiber, weswegen kein zusätzlicher Treiber benötigt wird. Für Windows wird jedoch eine .inf Datei benötigt. Die Arduino Software enthält einen Serial Monitor mit welchem man einfache Text Daten an das Arduino Board senden und von diesem empfangen kann. Die RX und TX LEDs des Boards blinken wenn Daten über den USB-to-Serial Chip und die USB Verbindung übertragen werden (jedoch nicht bei serieller Kommunikation über die Pins 0 und 1).

Die SoftwareSerial Library ermöglicht die serielle Kommunikation an jedem der Digitalen Pins des Arduino Uno.

Der ATmega328 unterstützt außerdem I2C (TWI) und SPI Kommunikation. Die Arduino Software enthält eine Wire Library, welche die Nutzung des I2C Bus vereinfacht; Für nähere Informationen beachten Sie die Wire Documentation. Für SPI Kommunikation kann die SPI Library verwendet werden.

## Programmierung

Der Arduino Uno kann mit der Arduino Software (download?) programmiert werden. Wählen sie Arduino Uno im Tools > Board Menü (je nach verwendetem Mikrocontroller). Für mehr Information beachten Sie die Referenz und dieTutorials?.

Der ATmega328 des Arduino Uno wird mit einem vorinstalliertem Bootloader? ausgeliefert, welcher den Upload von neuem Code ohne die Verwendung eines externen Hardware Programmers erlaubt. Er kommuniziert unter Verwendung des ursprünglichen STK500 Protokolls (Reference, C Header Dateien).

Der Bootloader kann umgangen werden und der Mikrocontroller über den ICSP (In-Circuit Serial Programming) Header programmiert werden. Lesen sie dafür diese Anleitung?.

Der ATmega16U2 (oder 8U2 in der Version Rev1 und Rev2 des Boards) Firmware Source Code ist verfügbar. Der ATmega16U2/8U2 besitzt einen DFU Bootloader wlechem man auf folgende Weisen aktivieren kann:

- Auf Rev1 Boards: Verbinden Sie die Löt Jumper auf der Rückseite des Boards (in der Nähe der Italien-Karte) und setzten Sie dann den 8U2 zurück.
- Auf Rev2 oder neueren Boards: Diese Boards besitzen einen Widerstand, welcher die 8U2/16U2 HWB Leitung herunter auf Ground zieht, was es vereinfacht, den DFU Modus zu aktivieren. Sie können dann Atmels FLIP Software (Windows) oder den DFU Programmer (Mac OS X and Linux) nutzen um neue Firmware zu laden. Außerdem können Sie die ISP Header mit einem externen Hardware Programmer benutzen (und den DFU Bootloader überschreiben). Mehr Informationen dazu gibt es in diesem von einem Anwender erstellten Tutorial.

## Automatischer (Software) Reset

Anstatt einen physikalischen Tastendruck des Reset Buttons vor einem Upload zu benötigen wurde der Arduino Uno so entworfen, dass er von einer Computer Software zurückgesetzt werden kann. Eine der Hardware Flow Control Leitungen (DTR) des ATmega8U2/16U2 ist über einen 100 nF Kondensator mit der Reset Leitung des ATmega328 verbunden. Wenn diese Leitung auf LOW geschaltet wird fällt die Reset Leitung lang genug um den Chip zurückzusetzen. Die Arduino Software nutzt dies um neuen Code mit einem einfachen Klick des Upload Buttons in der Arduino Entwicklungsumgebung hochladen zu können. Das bedeutet, dass der Bootloader einen kürzeren Aussetzer besitzt, da das Herabsetzen der DTR gut mit dem Start des Code Uploads koordiniert werden kann.

Dieses Setup hat noch andere Folgen. Wenn der Uno mit einem Computer mit dem Betriebssystem Mac OS X oder Linux verbunden ist, wird er jedes mal zurückgesetzt, wenn die Software eine Verbindung (via USB) herstellt. Für ungefähr eine halbe Sekunde läuft dann der Bootloader auf dem Arduino. Da dieser dafür ausgelegt ist fehlerhaften Code zu ignorieren (also alles außer einem Upload von neuem Code) ignoriert er die ersten paar Bytes an Daten welche über eine neue Verbindung an das Board gesendet werden. Achten sie deshalb darauf, dass Software, welche mit einem Sketch auf dem Arduino Board kommunizieren soll, eine Sekunde nach Herstellung der Verbindung abwartet, bevor Sie Daten an den Arduino sendet.

Der Uno besitzt eine Leiterbahn welche man durchtrennen kann um den Auto-Reset zu deaktivieren. Die Kontaktflächen auf beiden Seiten der Leiterbahn können zusammengelötet werden um den Auto-Reset zu reaktivieren. Die Leiterbahn ist Reset-EN benannt. Außerdem kann man den Auto-Reset deaktivieren, indem man einen 110 Ohm Widerstand zwischen 5V und Reset Leitung schaltet. Für mehr Informationen dazu beachten lesen sie diesen Forum Thread.

## USB-Überstrom Schutz

Der Arduino Uno besitzt eine zurücksetzbare Mehrfachsicherung, welche den USB Port ihres Computers vor Kurzschlüssen und Überstrom schützt. Auch wenn die meisten Computer einen internen Schutz für solche Fälle besitzen bietet diese Sicherung einen zusätzlichen Schutz. Wenn mehr als 500 mA über die USB Verbindung fließen durchtrennt die Sicherung die Verbindung bis der Kurzschluss bzw. die Überlastung entfernt wurde.

## Physikalische Eigenschaften

Die maximale Länge und Breite der Arduino Uno Platine sind 6,86cm (2.7 inch) und 5.3cm (2.1 inch). Der USB Port und die Strombuchse ragen etwas über diese Maße hinaus. Vier Löcher erlauben das Festschrauben des Boards auf Oberflächen und in Gehäuse. Bitte beachten sie, dass der Abstand zwischen den Digitalen Pins 7 und 8 0.16 Inch beträgt, also kein gerades Vielfaches des 0.1 Inch Abstands der anderen Pins.

Weitere Bilder:

