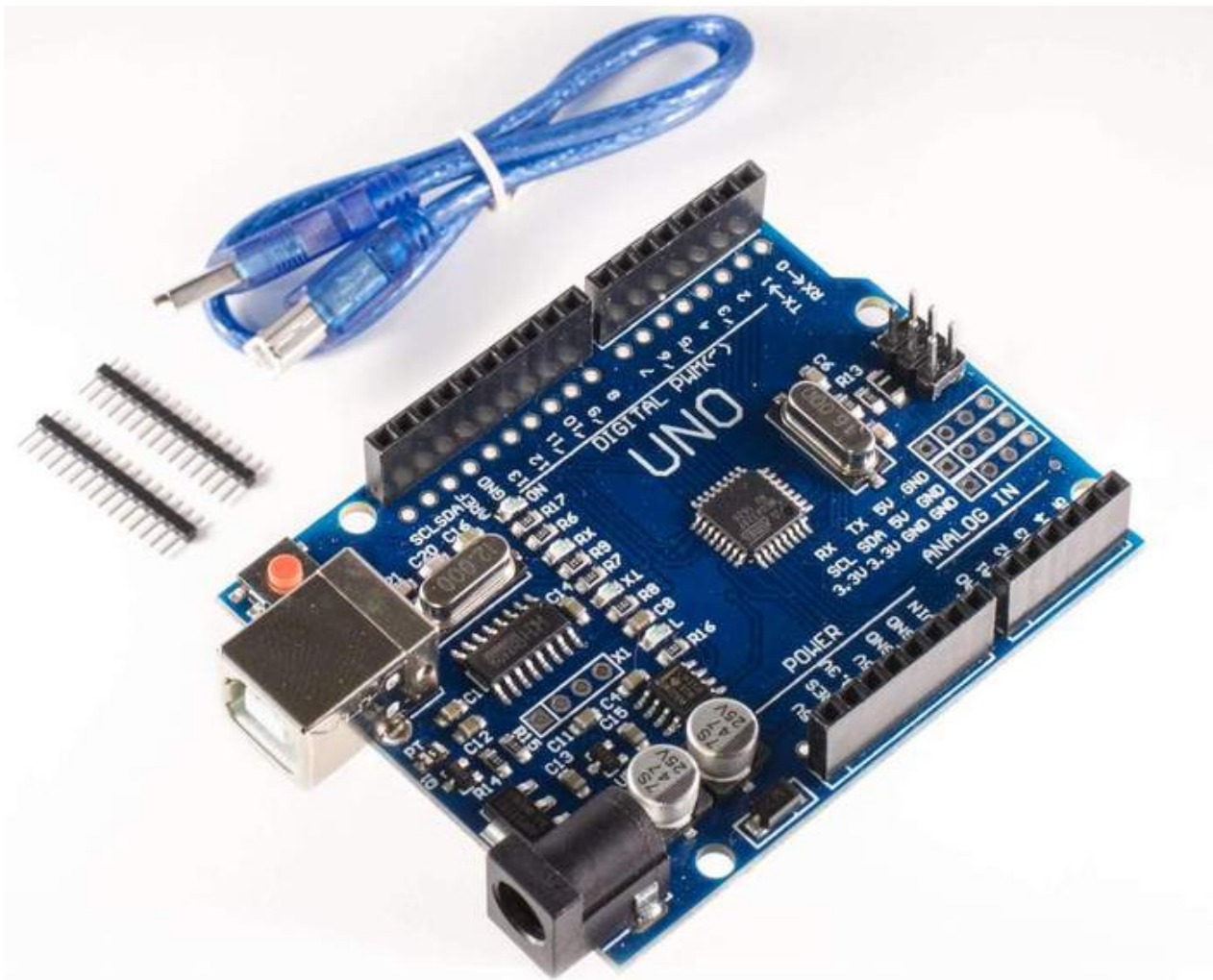


# AZ-Delivery

## Willkommen!

Und herzlichen Dank für den Erwerb unseres AZ-Delivery Uno R3. Auf den folgenden Seiten gehen wir gemeinsam mit Ihnen die ersten Schritte, von der Einrichtung bis zur Nutzung des Geräts, durch.

**Viel Spaß!**



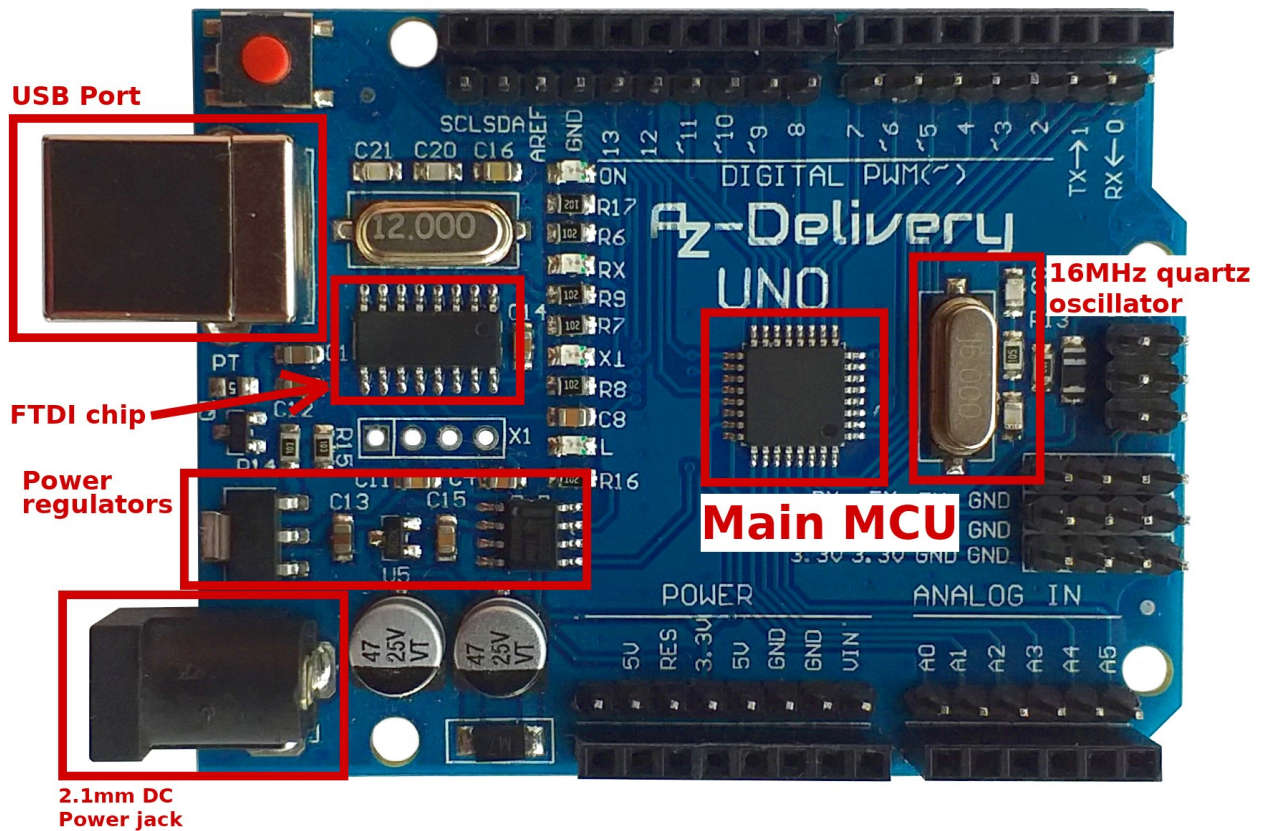
Der UNO ist das beste Board, um mit Elektronik und der Programmierung zu beginnen. Der UNO ist der am häufigsten benutzte und am besten dokumentierte Controller der Arduino-Familie.

I. d. R. Benötigt man zum arbeiten mit Mikrocontrollern eine stabilisierte Stromversorgung, eine Reset-Schaltung, ein Programmiergerät und vieles mehr. Alle erforderlichen Komponenten sind am UNO bereits vorhanden.

Programmiert wird das Modul über die Arduino-IDE, diese liefert viele kommentierte Code-Beispiele, um sehr einfach in das Thema einsteigen zu können. Es ist nicht notwendig die interne Funktionsweise des Mikrocontrollers zu verstehen, um diesen programmieren zu können. Schließen Sie einfach Ihr Arduino-Board über ein USB-Kabel an Ihren PC an, installieren und starten Sie die Arduino IDE, suchen und laden Sie das Programm, das Sie benötigen, auf Ihr Board und Los geht's. Es gibt bereits endlose Code- und Bibliotheksbeispiele, die online, meist auf GitHub, zur Verfügung stehen. Es gibt auch zahlreiche andere Arduino kompatible Boards, wie z.B. Shields, oder viele Sensoren, die so gebaut sind, dass Sie sie einfach, ohne Beschaltung an Ihr Arduino-Board anschließen können. Im Onlineshop von AZ-Delivery finden Sie eine Vielzahl von Shields, Sensoren und Modulen.

## Spezifikationen

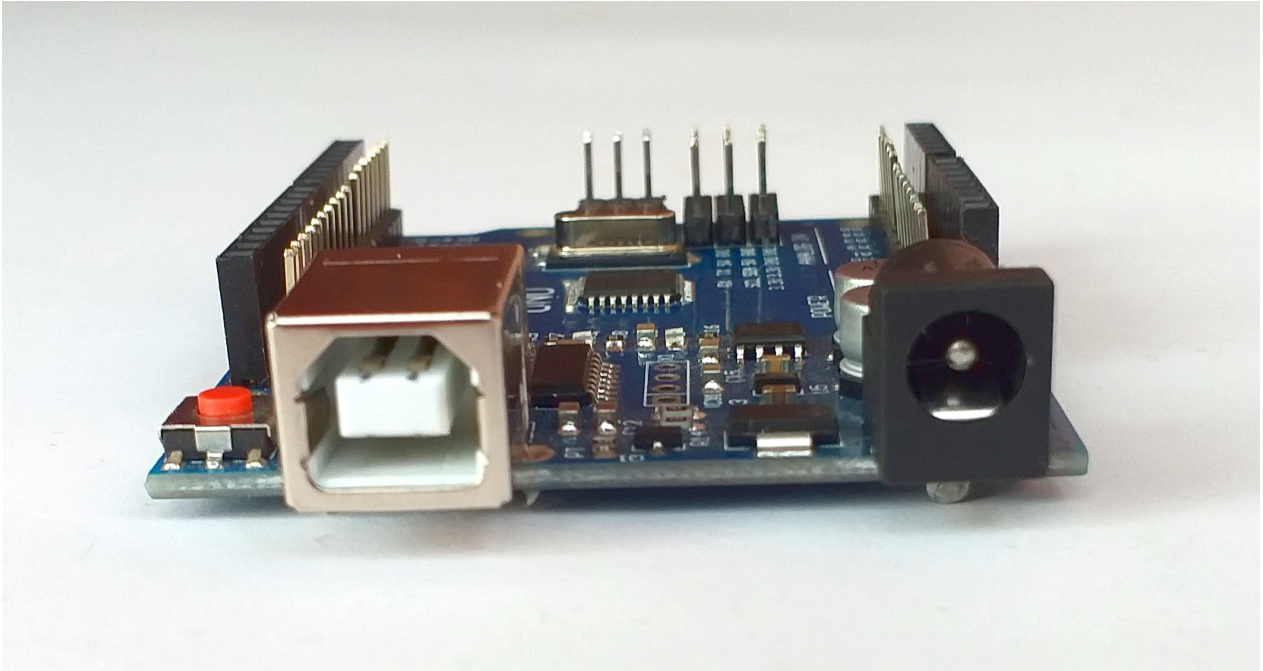
Mikrocontroller:	ATmega328P
Betriebsspannung:	5V
Versorgungsspannung:	7-12V
Input Voltage (limit):	6-20V
Digital I/O Pins:	14
PWM Digital I/O Pins:	6
Analog Input Pins:	6
Maximale Belastbarkeit eines I/O Pins:	20mA
Maximale Belastbarkeit des 3.3V Pin:	50mA
Flash Speicher:	32KB von denen 0.5KB vom Bootloader benutzt werden
SRAM:	2KB (ATmega328P)
EEPROM:	1KB (ATmega328P)
Takt:	16MHz
LED_BUILTIN:	Verbunden mit Digital I/O pin 13
Länge:	69mm
Breite:	54mm
Gewicht:	25g



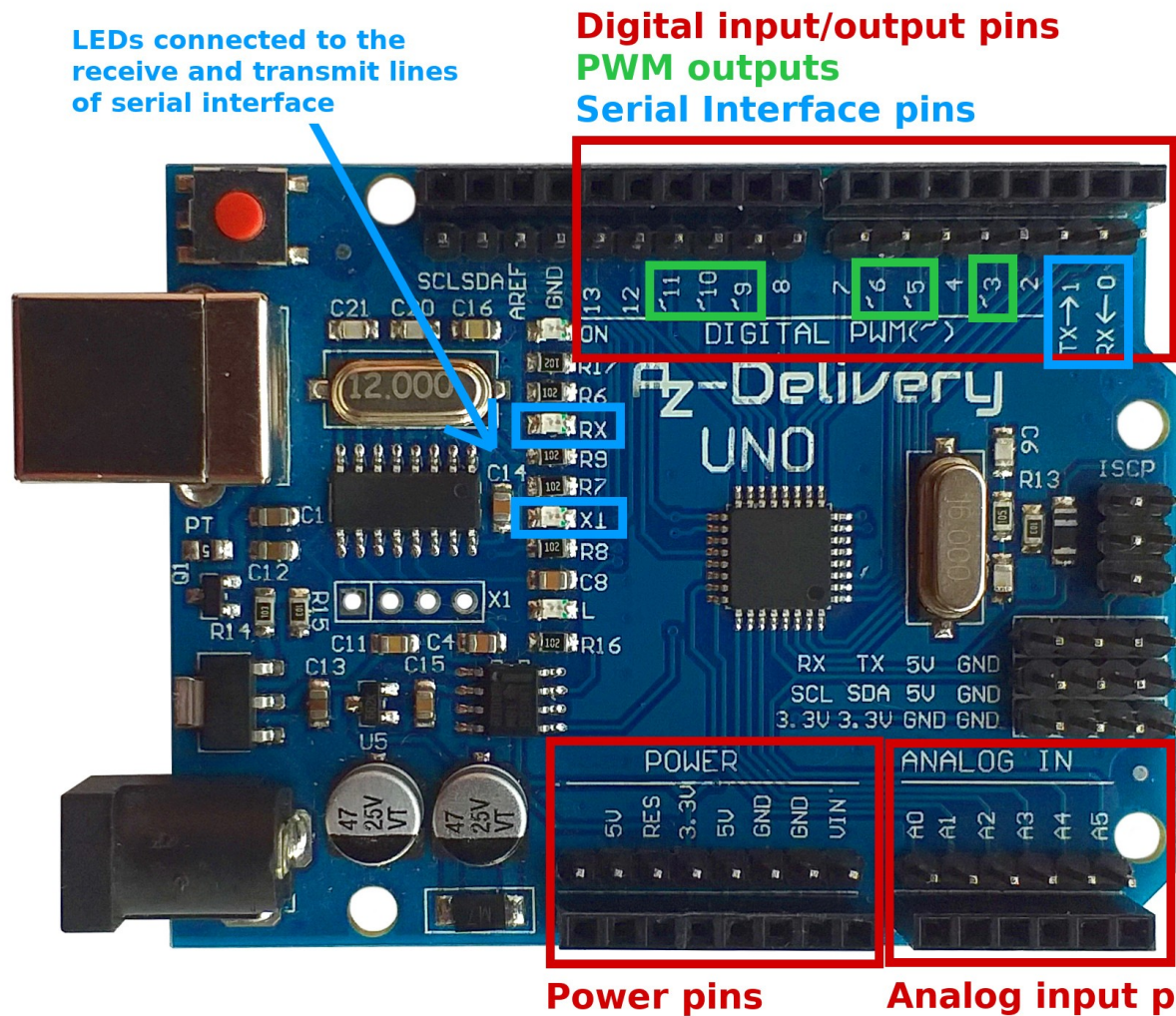
Der Uno R3 verfügt über eine ATmega328P CPU mit 16MHz. Der CH340 USB-Serial Chip stellt die Schnittstelle zwischen dem Mikrocontroller und Ihrem PC dar.

Im Übrigen können Sie auch die USB-Schnittstelle zur Stromversorgung Ihres Boards nutzen.





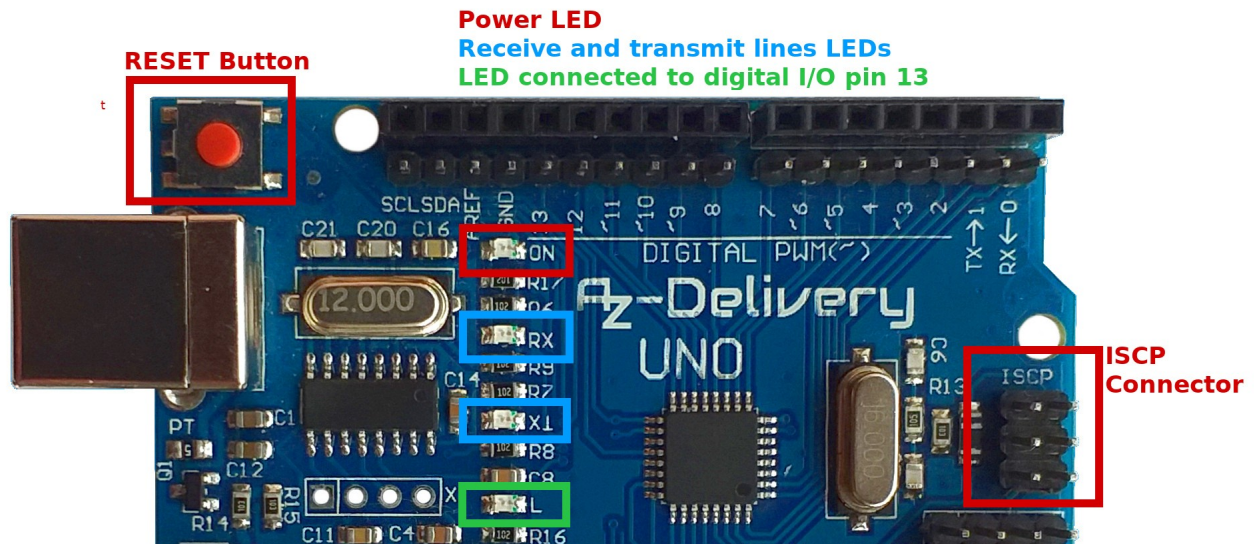
Der Uno R3 verfügt über zwei, auf dem Board befindliche Festspannungsregler mit +5V und 3.3V Spannung. Durch den 2.1mm Stecker können Sie diese mit Strom versorgen. (Wir empfehlen eine Eingangsspannung von 7 – 12 V). Durch die am Uno vorhandene Beschaltung, stehen auf dem Board sowohl stabile 5V als auch stabilisierte 3.3V zur Verfügung.



Der Uno ist so aufgebaut, dass digitale Ein-/Ausgangspins von analogen Eingangspins getrennt gruppiert werden. Es gibt also 6 analoge Eingangspins und getrennte 14 digitale Ein-/Ausgangspins. 6 von 14 digitalen Ein-/Ausgangspins können als PWM-Ausgänge (Pulsweitenmodulation) verwendet werden. Diese Pins sind mit dem Tildezeichen "~" gekennzeichnet (D3, D5, D6, D9, D10 und D11).

Die digitalen Ein-/Ausgangspins 0 und 1 sind mit den Empfangs- und Sendeleitungen der seriellen USART-Schnittstelle verbunden. Wir empfehlen daher, diese digitalen I/O-Pins niemals als digitale Ein- oder Ausgänge zu verwenden, da die serielle Schnittstelle jedes Mal verwendet

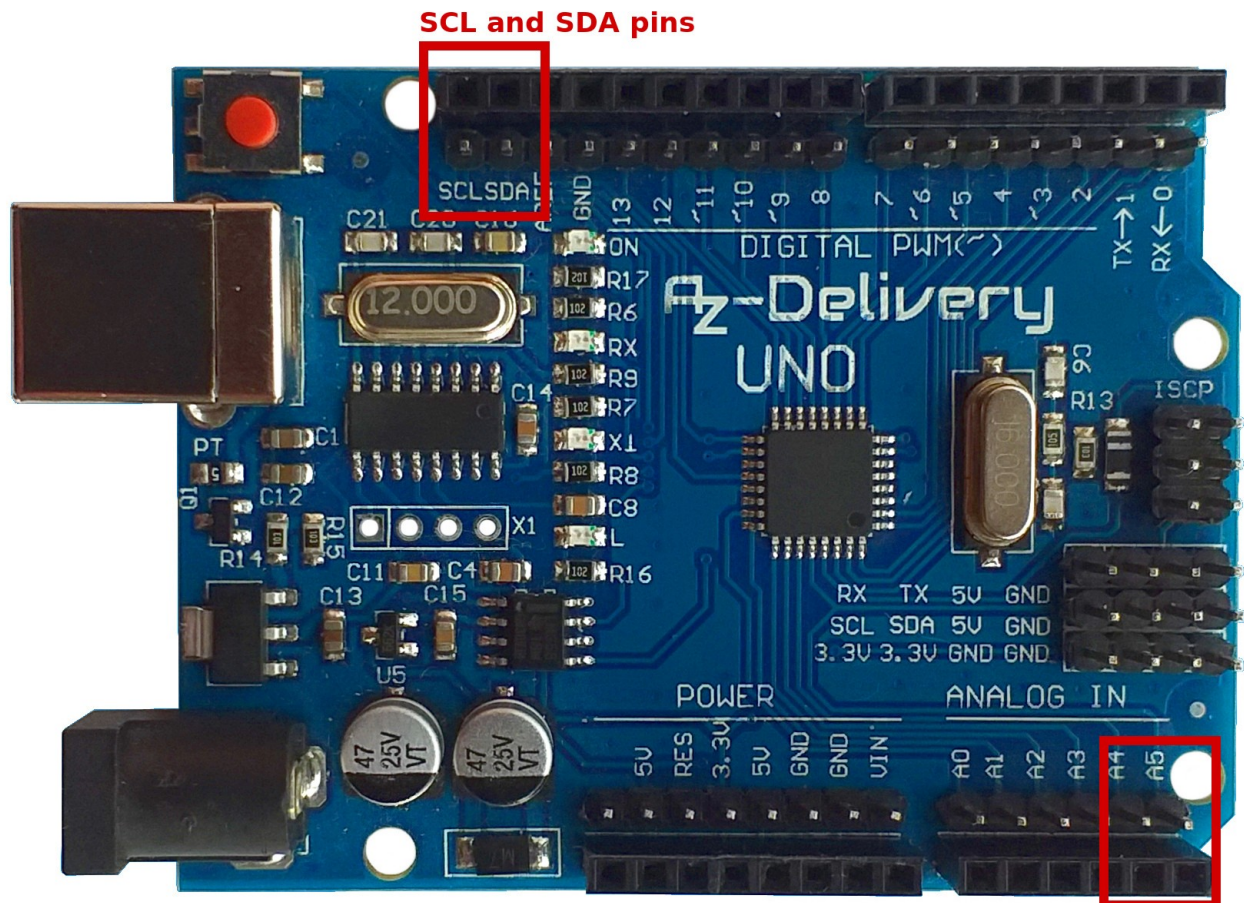
wird, wenn Sie ein neues Programm in Ihren Mikrocontroller auf den Uno laden. Bei einer Verwendung dieser Pins können während des Programmiervorgangs Fehler auftauchen und u.U. funktioniert die angeschlossene Peripherie nicht richtig.



"OnBoard" des UNO gibt es eine Reset-Taste für den Hauptprozessor, vier LEDs sowie einen ICSP-Anschluss der zur externen Programmierung des ATmega328 dient.

Eine LED leuchtet auf, wenn wir das UNO-Board mit Strom versorgen, die zweite LED ist mit Pin 13 verbunden und kann frei programmiert werden. Zwei weitere LEDs sind mit der RX und TX Leitung verbunden und blinken wenn darüber Daten übertragen werden.



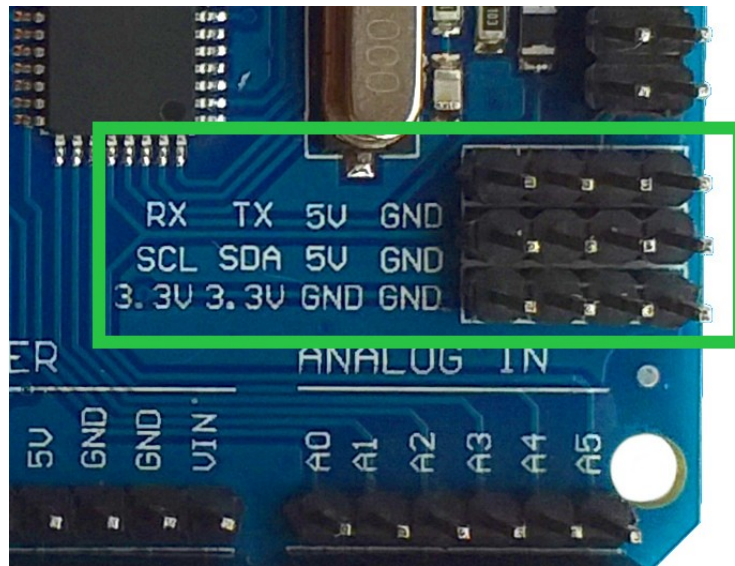


**Alternative functions of A4 and A5  
are SDA and SCL connections**

Der UNO R3 verfügt außerdem über zwei zusätzliche Pins (SDA und SCL), welche direkt mit den Pins A4 und A5 verbunden sind, diese dienen als Schnittstelle für den Anschluss von I2C-Geräten (TWI – Two Wire Interface).



# AZ-Delivery



Unser AZ-Delivery UNO R3 verfügt ausserdem über die oben abgebildeten Pins. Diese können wie beschriftet verwendet werden.

## Schnittstellen

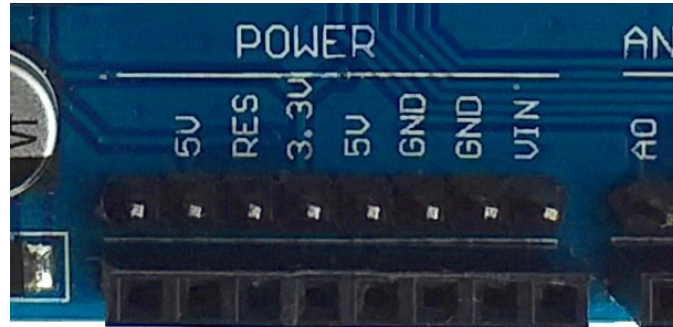
Wie bereits geschrieben, haben die digitalen I/O-Pins D0 und D1 alternative Funktionen. Sie sind mit den Empfangs- und Sendeleitungen der seriellen Schnittstelle verbunden.

Es gibt zwei weitere Kommunikationsschnittstellen, die vom ATmega328P-Mikrocontroller nativ unterstützt werden, Serial Peripheral Interface - SPI und Inter-Integrated Circuit Interface - I2C (oder TWI - Two Wire Interface).

Für die SPI-Schnittstelle werden die digitalen I/O-Pins D10, D11, D12 und D13 verwendet. Ihre Funktionen sind SS, MOSI, MISO und SCK.

Für die I2C-Schnittstelle werden die analogen Eingänge A4 und A5 oder zwei zusätzliche Pins SDA und SCL verwendet. Ihre Funktionen sind SDA und SCL.

## Power Pins



Power-Pin-Leiste (von rechts nach links):

**VIN** - Dies ist ein Spannungseingang, der als weitere externe Stromversorgung für den Hauptmikrocontroller dient, wenn er nicht über den USB-Anschluss mit Strom versorgt wird.

**GND** - GrouND, Masse, 0V

**GND** - GrouND, Masse, 0V

**5V** - +5V-Ausgang

**3.3V** - +3.3V-Ausgang

**RES** - RESET, wenn Sie diesen Pin mit GND Verbinden (auf Masse ziehen), führt der Hauptprozessor einen Neustart aus.

**5V** - Dieser Pin kann als Spannungsausgang benutzt werden, eigentlich ist dieser als Referenzspannung für externe Geräte oder Shields vorgesehen und gibt das Logiclevel für die Peripherie vor. (+5V Referenzspannung)

- Der letzte Pin ist unbeschaltet.



## Arduino Leistungs-, Strom- und Spannungsbegrenzungen

### Voltage Input Limits:

Eingangsspannung: Um den UNO mit Strom zu versorgen, können Sie entweder die USB- oder die 2.1mm- Buchse verwenden, die 2.1mm Buchse ist mit den Pins GND und VIN verbunden. Folgende Einschränkungen sind zu beachten:

» **Empfohlene Eingangsspannung über 2.1mm-Buchse: 7~12V.**

Diese Eingangsspannungen können auf unbestimmte Zeit verwendet werden.

» **Absolute Spannungsgrenzen für die Versorgung des Arduino: 6~20V**

- Eine Versorgungsspannung von unter 7V kann zur Instabilität des 5V-Pegels auf dem Board führen, sowie falsche Werte für Funktionen mit analogRead() zur Folge haben.

- Ein dauerhafter Betrieb mit 12V oder mehr, kann zur Erwärmung des Festspannungsreglers des Arduino und somit zur Überhitzung führen. Kurze Zeiträume sollten kein Problem darstellen. Wird der Festspannungsregler im Betrieb zu heiß, ist ein reduzieren der Spannung erforderlich, um Ihr Board nicht zu beschädigen und die Wärmeentwicklung zu reduzieren.

### Achtung!!!

Bevor Sie empfindliche Teile des UNO (so ziemlich das gesamte Board) berühren, sollten Sie das Metallteil des USB-Steckers berühren, um sich auf dem Board zu erden und die von Ihnen aufgebaute statische Spannung sicher zu entladen.

» **Spannungsbegrenzung an I/O-Pins: -0.5 to +5.5V max.**

Wenn Sie eine Spannung an einem digitalen oder analogen Arduino-Eingangspin einlesen, stellen Sie sicher, dass sie zwischen 0 und 5V liegt. Liegen die Werte außerhalb dieser Begrenzungen, können Sie die Spannung mit einem Spannungsteiler absenken. Dadurch wird die Eingangsspannung skaliert, um analoge oder digitale Messungen von Spannungen außerhalb des zulässigen Bereichs zu ermöglichen. Wenn Ihr Eingangssignal digital ist, und Sie keine skalierten analogen Messwerte benötigen, können Sie durch hinzufügen eines einfachen Widerstandes die Spannung begrenzen (Der Prozessor verfügt über interne Dioden), anstatt diese zu skalieren. Durch Hinzufügen eines 10k $\Omega$ -Widerstandes in Reihe mit dem Eingangspin (beliebiger Eingangspin), können Eingangsspannungen bis zu -10,5V oder bis zu +15,5V erreicht werden.

## Beschränkungen des ausgehenden Stroms:

- » **Gesamte maximale Stromaufnahme des UNO bei Stromversorgung über einen USB-Anschluss: 500mA**
- » **Der Uno hat eine "rückstellbare Polyfuse, die die USB-Anschlüsse Ihres Computers vor Kurzschlüssen und Überstrom schützt".**
- » **Gesamte maximale Stromaufnahme bei Versorgung über ein externes Netzteil: 1A**

**Hinweis:** Wenn der UNO nicht über USB mit Strom versorgt wird, wird die gesamte 5-V-Strombegrenzung durch den Spannungsregler auf Ihrer Platine und / oder Ihr Eingangsnetzteil begrenzt, je nachdem, welcher Wert weniger Strom liefert. Nehmen wir an, Ihre Stromversorgung für den Arduino kann  $7 \div 12V$  und  $\geq 1A$  liefern. In diesem Fall wird die 5-V-Leistung streng durch den Spannungsregler Ihres Boards begrenzt.

- » **Gesamte maximale Stromaufnahme über den Arduino "5V" -Pin und "GND": wie oben angegeben.**
- » **Maximaler Gesamtstrom pro Eingangs- / Ausgangspin: 40 mA**
- » **Die Summe ALLER Ströme an den I/O-Pins dürfen **200mA!!!** nicht übersteigen!**



**Hinweis:** Dieser Teil führt zu den meisten Verständnisproblemen. Trotz der Tatsache, dass der am Board befindliche Festspannungsregler bei 5V 1A leisten kann, darf die Summe aller über den Prozessor laufenden Ströme nie mehr als 200mA erreichen. Wenn Sie also planen, 10 LEDs am Mikrocontroller zu betreiben, gelangen Sie damit bereits an das zulässige Maximum! Mehr als 200mA (an den GPIOs) beschädigen möglicherweise Ihren Controller. Um das Problem zu umgehen, empfiehlt es sich Transistoren zu benutzen, da diese zum Schalten wenig Strom benötigen und die Last direkt vom 5V-Pin beziehen können, um angeschlossene Peripherie und Sensoren über den auf dem Board befindlichen Festspannungsregler zu betreiben, anstatt die GPIOs unnötig zu belasten.



## **Installation des CH340-Treibers**

Während das Original von Arduino einen Atmel-Chip verwendet, der unter Umständen eine manuelle Treiberinstallation benötigt, ist bei den meisten Boards von AZ-Delivery ein CH340-Chip im Einsatz, der von Windows automatisch erkannt wird.

Sollte das einmal nicht der Fall sein, laden sie hier den aktuellen Treiber herunter und entpacken Sie diesen:

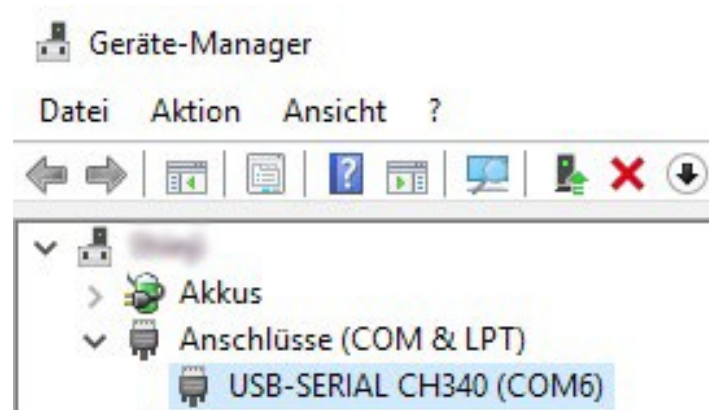
» Windows: [http://www.wch.cn/download/CH341SER\\_ZIP.html](http://www.wch.cn/download/CH341SER_ZIP.html)

» Mac: [http://www.wch.cn/download/CH341SER\\_MAC\\_ZIP.html](http://www.wch.cn/download/CH341SER_MAC_ZIP.html)

Unter Windows installieren Sie ihn einfach durch das Ausführen der "SETUP.EXE" im Ordner "CH341SER". Mac-Nutzer folgen am besten den Installationsanweisungen, die dem Treiberpaket beiliegen.

Nach dem erneuten Anschließen des Nanos sollte dieser als "USB-SERIAL CH340"-Gerät (Windows) erkannt werden.

# Az-Delivery





## Arduino IDE

Um mit den Arduino-Boards arbeiten zu können, benötigen Sie die Arduino-IDE. Diese können Sie über folgenden Link beziehen:  
<https://www.arduino.cc/en/Main/Software>

Bitte achten Sie darauf, die Installations-Datei zu verwenden, anstelle der APP:

Download the Arduino IDE





**ARDUINO 1.8.9**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  

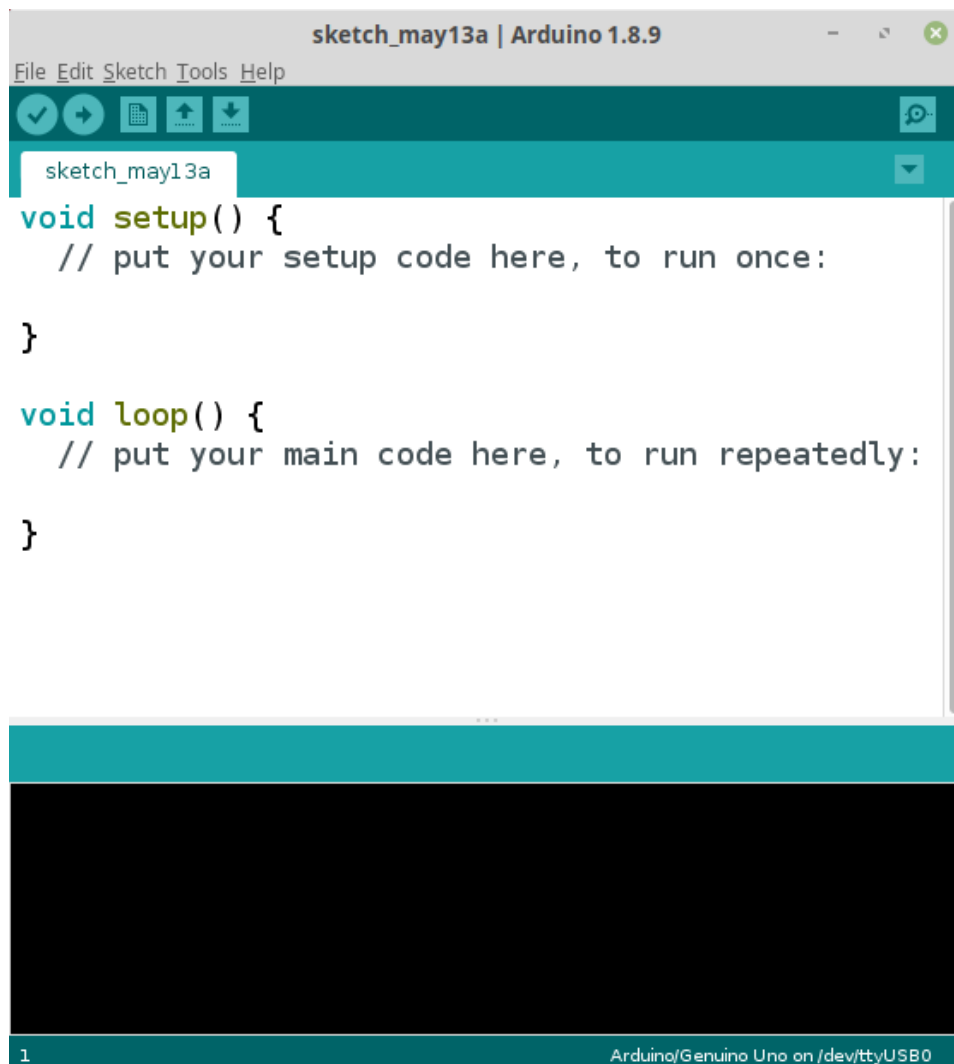

**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

# Az-Delivery

Wenn Sie die Software installieren und das Programm öffnen, wird dieses Startfenster angezeigt:



Das geöffnete Programmbeispiel wird als leere Skizze bezeichnet. Eine Skizze ist der Rahmen, in dem wir unseren Code schreiben. Sie besteht aus zwei wesentlichen Teilen, `setup()` Funktion und `loop()` Funktion, Sie können aber auch eine beliebige Anzahl anderer Funktionen hinzufügen.

# Az-Delivery

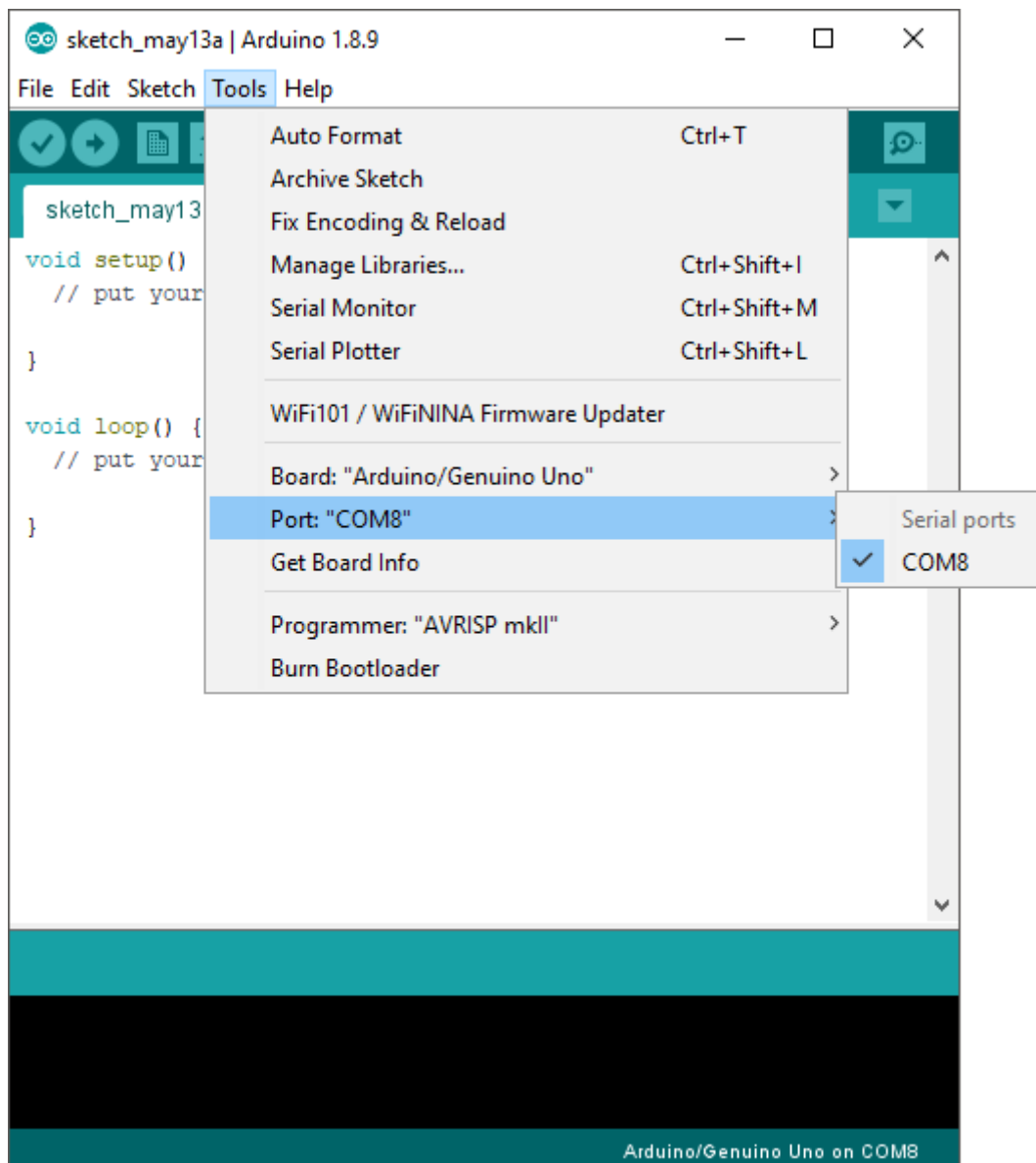
Die `setup()` Funktion wird aufgerufen, wenn eine Skizze beginnt. Benutzen Sie diese, um Variablen zu deklarieren, Pin-Modi zu initialisieren und Bibliotheken einzubinden, etc. Die `setup()` Funktion wird nur einmal, nach jedem Einschalten oder Zurücksetzen des Arduino-Boards, ausgeführt.

Die `loop()` Funktion läuft nach der `setup()`-Schleife und wiederholt sich auf unbestimmte Zeit, immer und immer wieder, die so genannte "endless" loop Funktion. Diese läuft so lange durch, wie der Controller in Betrieb ist. Ein Ende der Schleife sollte niemals erreicht werden, da dann der Controller seinen Dienst einstellt und keine Funktion mehr hat. Hier schreiben wir auch die Logik und Algorithmen, mit denen unsere Anwendung für Arduino-Boards arbeitet.



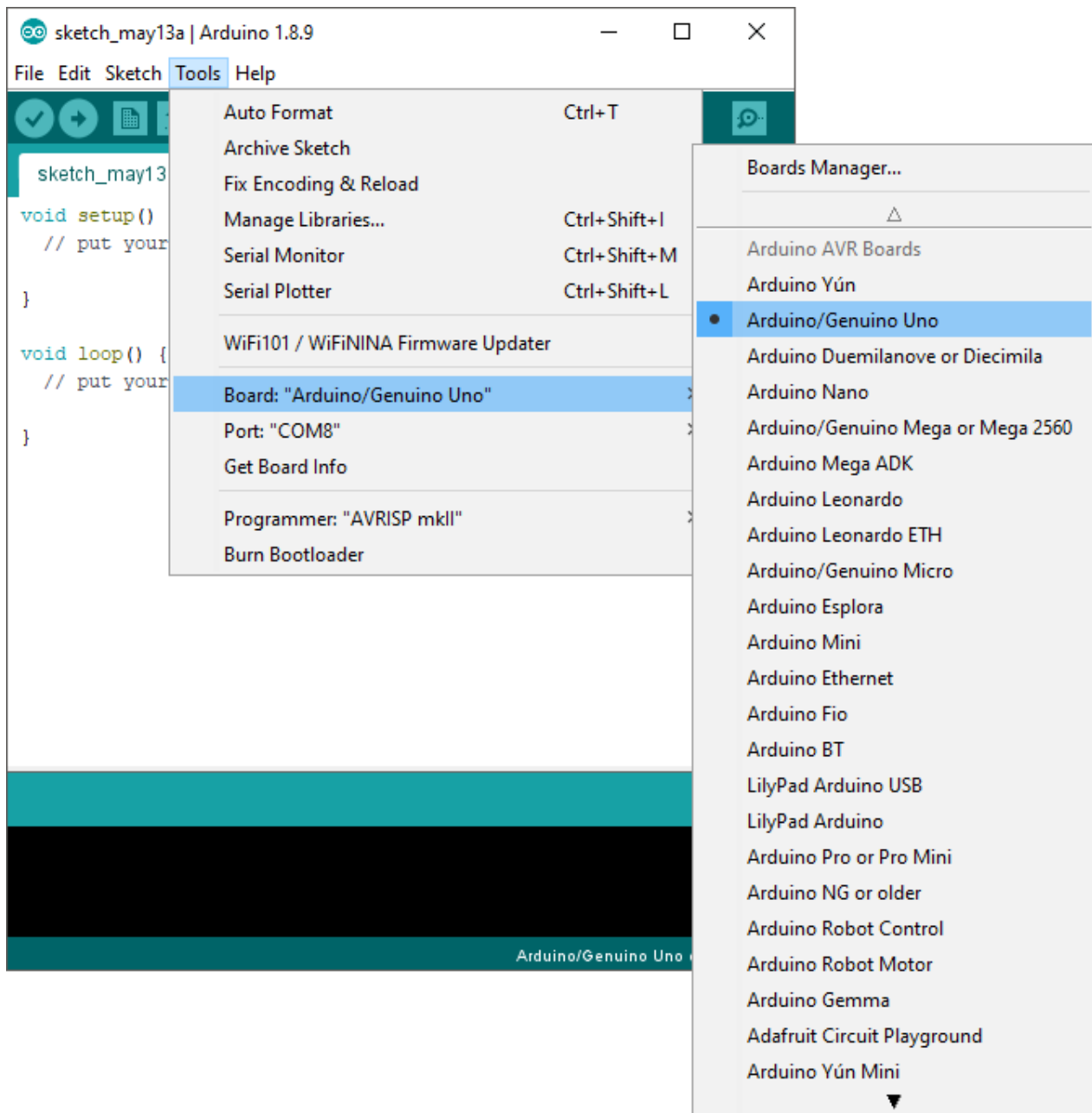
# Az-Delivery

Wenn wir unseren Arduino Uno über ein USB A zu B Kabel an den PC anschließen, müssen wir in der Arduino IDE zunächst den USB-Anschluss auswählen, an dem unser Arduino Uno Board angeschlossen ist, diese werden als COM-Ports bezeichnet.



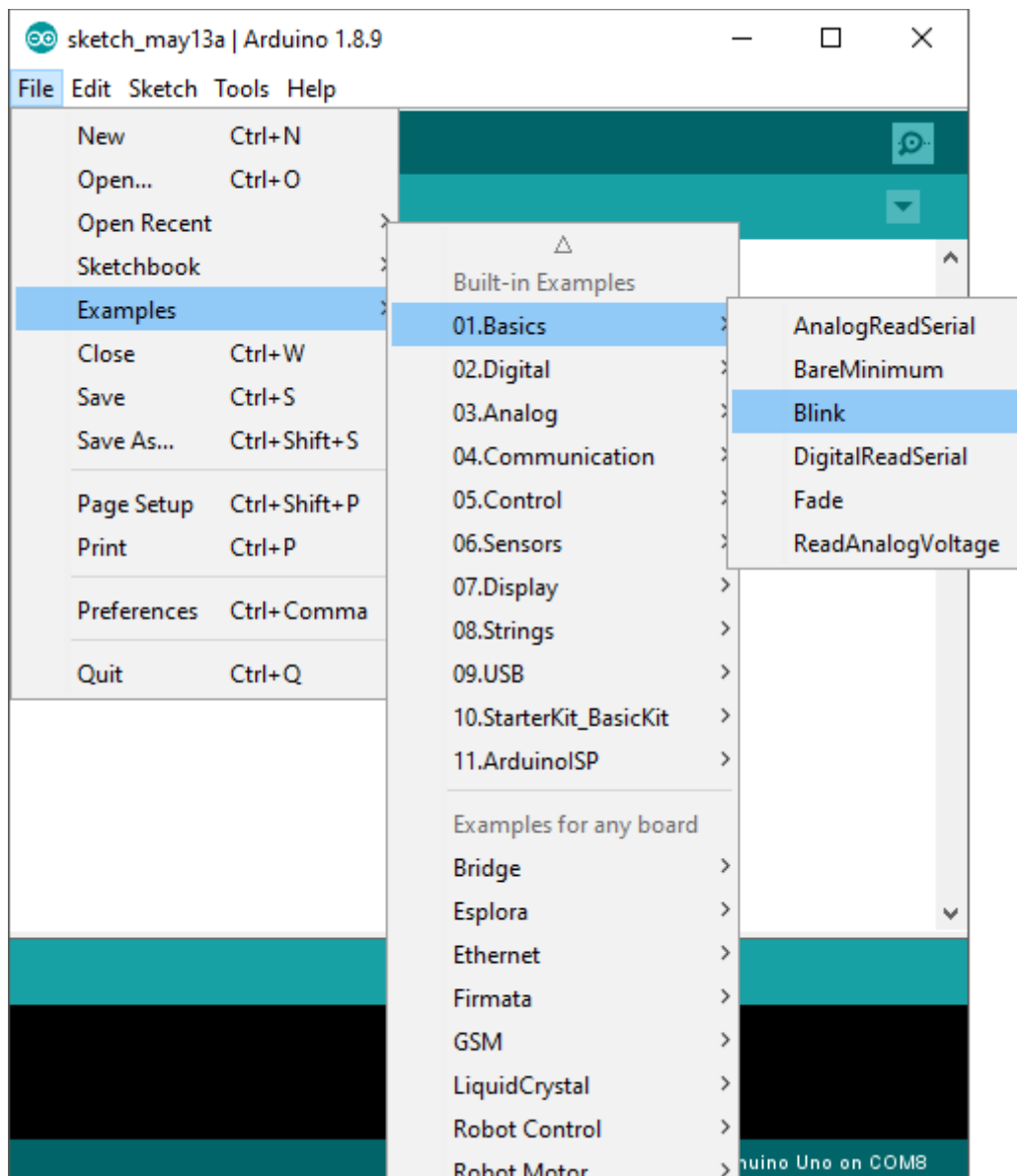
Im Anschluss müssen wir das Board definieren, welches mit dem PC verbunden wurde und das wir programmieren möchten, wie in der nächsten Abbildung zu erkennen:

# Az-Delivery



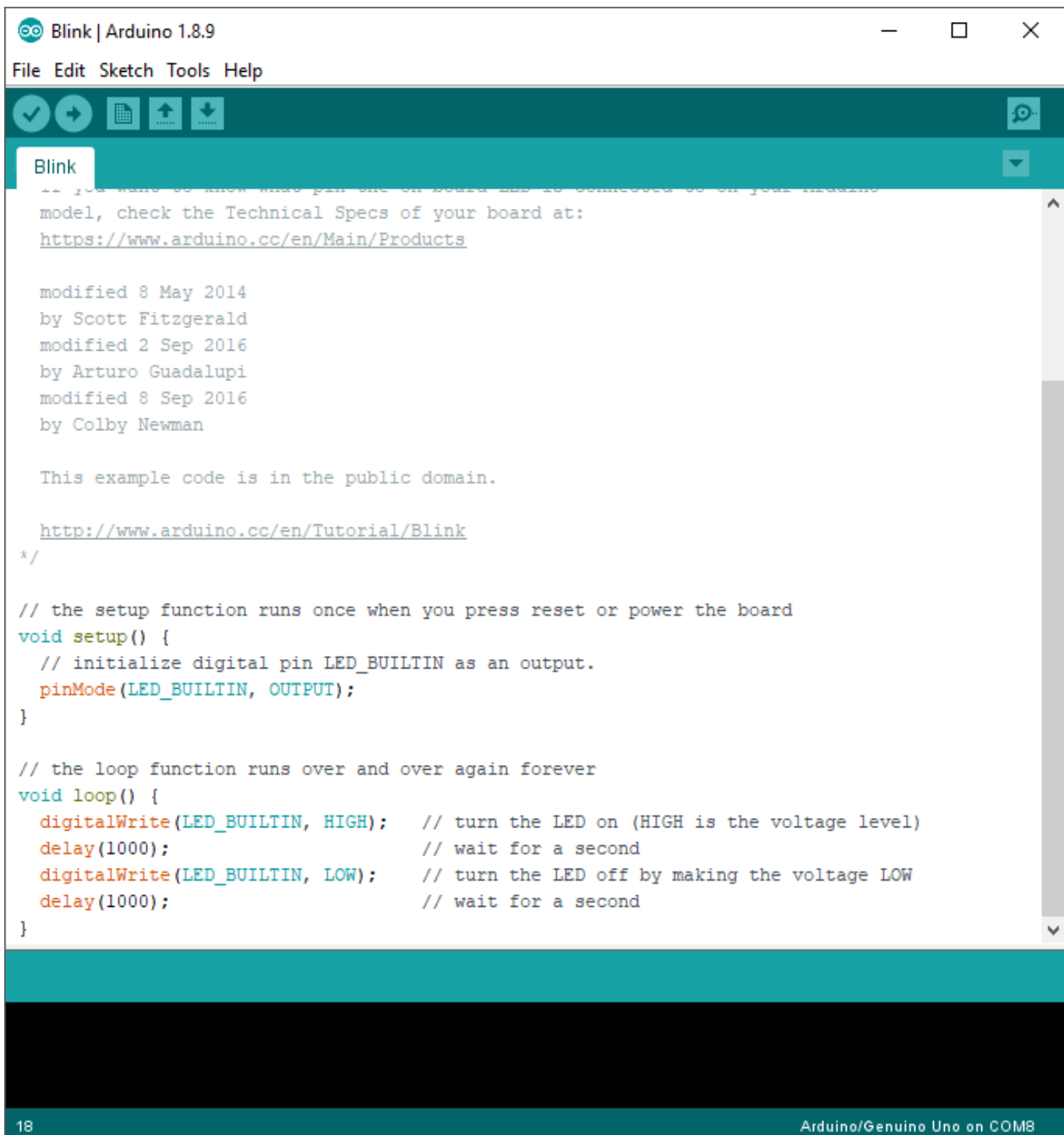
## Anwendungsbeispiel

Jetzt können wir mit der Programmierung beginnen. Die Arduino IDE wird mit vielen vorgefertigten Skizzenbeispielen geliefert, die Sie verwenden und anpassen können. Heute werden wir das Beispiel einer BLINK-Skizze verwenden. Diese finden Sie unter Datei -> Beispiele -> Basics -> Blink.



Im Anschluss öffnet sich ein Fenster mit dem Code:

# Az-Delivery

The image is a screenshot of the Arduino IDE interface. At the top, the window title is "Blink | Arduino 1.8.9". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The main text area shows the "Blink" sketch. It starts with a comment: "If you want to know what pin on board LED is connected to on your Arduino model, check the Technical Specs of your board at: https://www.arduino.cc/en/Main/Products". This is followed by a list of modifications: "modified 8 May 2014 by Scott Fitzgerald", "modified 2 Sep 2016 by Arturo Guadalupi", "modified 8 Sep 2016 by Colby Newman", and a statement "This example code is in the public domain." with a link "http://www.arduino.cc/en/Tutorial/Blink". The code then defines a setup function to initialize the LED\_BUILTIN pin as an output and a loop function that turns the LED on (HIGH) for 1000ms and off (LOW) for 1000ms. The status bar at the bottom shows "18" on the left and "Arduino/Genuino Uno on COM8" on the right.

```
Blink | Arduino 1.8.9
File Edit Sketch Tools Help

[Icons: Check, Run, Upload, Download]

Blink
If you want to know what pin on board LED is connected to on your Arduino
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

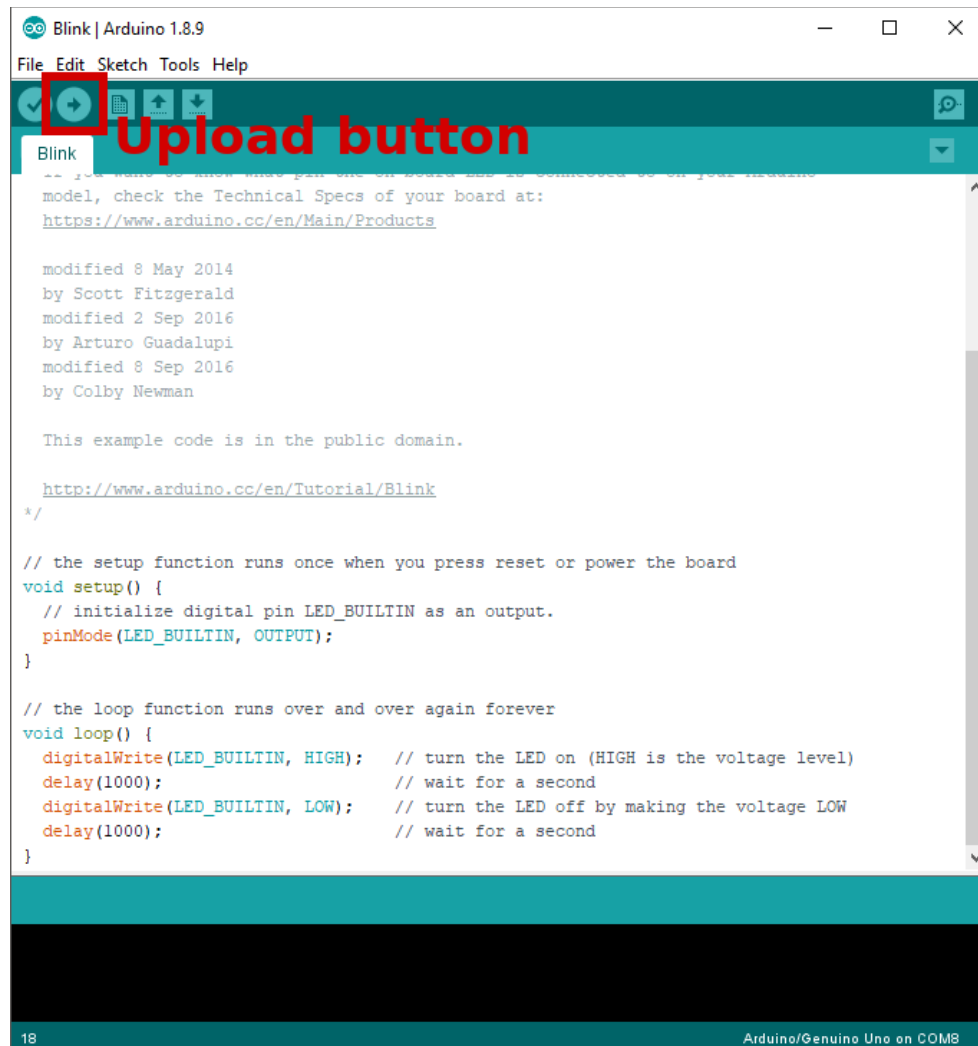
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

18 Arduino/Genuino Uno on COM8
```

Der Sketch beginnt, indem die "LED\_BUILTIN" (auf dem Board befindliche LED mit Pin 13 verbunden) als Output definiert wird. Im Anschluss wird der Pin auf HIGH gesetzt (LED leuchtet auf). Nach einer Wartezeit von 1000ms (delay()) wird dieser wieder in den LOW Zustand versetzt. Nach einer erneuten Wartezeit von 1000ms wird die loop-Schleife wiederholt.

# Az-Delivery

Haben wir den Programmcode nach unseren Bedürfnissen angepasst, können wir den Sketch wie in der Abbildung unten zu sehen auf das Board laden:



Im Anschluss blinkt die LED im Sekundentakt.





**Geschafft, jetzt kann der Controller mit Ihren selbsterstellten Programmen oder Vorlagen aus dem Internet programmiert werden.**



**Wenn Sie auf der Suche nach hochwertigen Produkten für Arduino und Raspberry Pi sind, ist die AZ-Delivery Vertriebs GmbH das richtige Unternehmen, um diese zu beziehen. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, E-Books, Bibliotheken und Unterstützung durch unsere technischen Experten.**

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>