

# AZ-Delivery

## Willkommen!

Vielen Dank, dass sie sich für unser Nano V3.0 Pro Board mit USB-Kabel von AZ- Delivery entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären, wie Sie das Gerät einrichten und nutzen können.

**Viel Spaß!**



# Az-Delivery

Das Nano V3.0 ist ein großartiges Gerät, das für das Erlernen von Elektronik oder für die Erstellung von Prototypen und das Erlernen der Programmierung bestens geeignet ist. Der Nano V3.0 ist eigentlich ein Mikrocontroller, so konstruiert, dass Sie keine zusätzlichen Komponenten benötigen. Wenn Sie einen einzelnen Mikrocontroller verwenden, brauchen Sie eine stabile DC-Stromversorgung, einen externen Programmierer, eine Reset-Schaltung und viele andere Dinge. Mit einem Nano V3.0 haben Sie Alles auf einer Platine. Und das Leistungsstärkste am Nano V3.0 ist das Arduino IDE (Integrated Development Environment) mit einer endlosen Anzahl von leicht verständlichen Codebeispielen, die bereits dafür geschrieben wurden. Sie müssen nicht die interne Arbeitsweise des Onboard-Mikrocontrollers erlernen, um ihn zu programmieren. Schließen Sie einfach Ihr Nano V3.0-Board über ein USB-Kabel an Ihren PC an, installieren und starten Sie die Arduino-IDE, suchen Sie das benötigte Programm und laden Sie es auf Ihr Board hoch. Das war's. Es gibt endlose Code- und Librarybeispiele, die bereits online verfügbar sind. Sie müssen nur noch danach suchen. Außerdem gibt es zahlreiche andere Arduino-kompatible Boards, wie z.B. sogenannte "Shields", das sind aufsteckbare Erweiterungen, oder Sensoren, die so gebaut sind, dass Sie sie einfach an Ihr Nano V3.0-Board anschließen können. Durchsuchen Sie einfach unseren Online-Shop az-delivery.de und Sie werden Alles finden, was Sie benötigen.

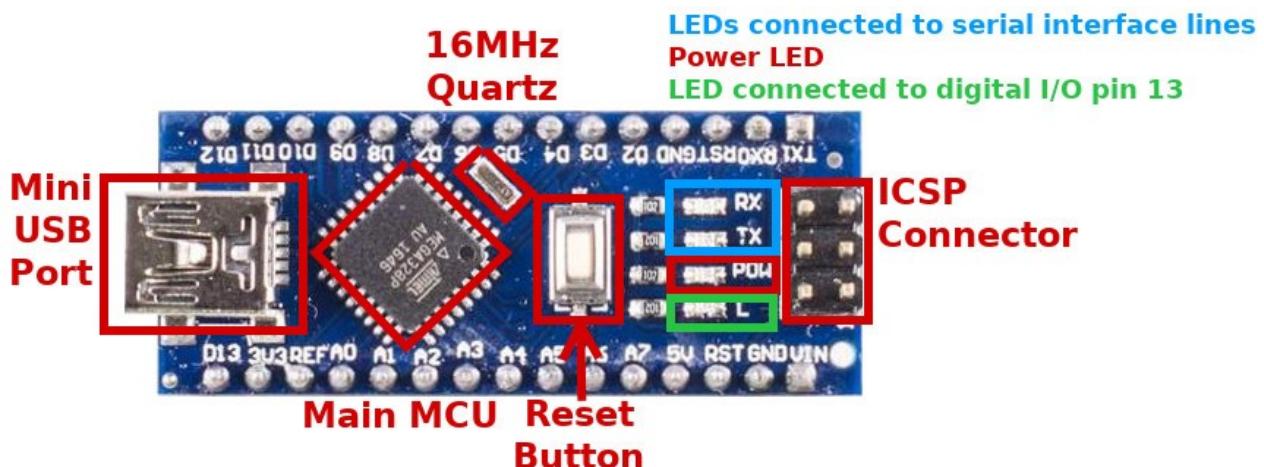
Der Nano V3.0 ist ein kleines, kompatibles, flexibles und Breadboard-freundliches Mikrocontroller-Board, das auf dem ATmega328P basiert. Es besitzt fast die gleiche Funktionalität wie der Arduino Uno, aber in kompakter Größe.

# Az-Delivery

**Dieses Nano V3.0-Board hat einen CH340-Chip und wird mit einem USB-Kabel zur Verbindung mit dem PC geliefert.**

## Technische Daten

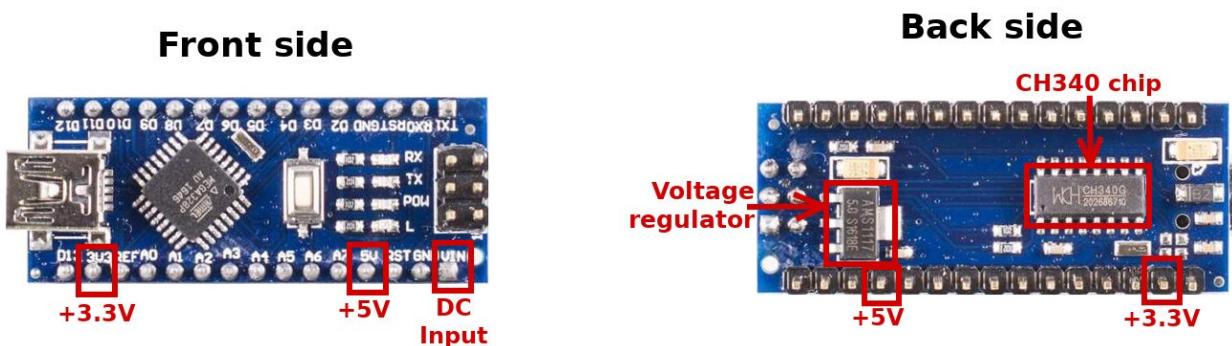
Mikrocontroller:	ATmega328
Betriebsspannung:	5V
Eingangsspannung (Empfehlung):	7÷12V
Eingangsspannung (Limit):	6÷20V
Digitale I/O-Pins:	22 (6 sind PWM)
PWM-Ausgang:	6
Analoge Input-Pins:	8
Gleichstrom pro I/O-Pin:	20mA
Gleichstrom für 3.3V-Pin:	50mA
Flash-Speicher:	32KB, 2KB von bootloader verwendet
SRAM:	2KB (ATmega328P)
EEPROM:	1KB (ATmega328P)
Taktfrequenz:	16MHz
LED_BUILTIN:	verbunden mit digital I/O-Pin 13
Länge:	18mm
Breite:	45mm
Gewicht:	7g



Ein Nano V3.0 verfügt über den Hauptmikrocontroller "ATMega328p" mit einem 16MHz Quarzoszillator. Er hat einen Mini-USB-Port, der zur Programmierung des Hauptmikrocontrollers, zur Stromversorgung des Boards, und zur Datenaustausch, z.B. über den Seriellen Monitor der IDE) verwendet werden kann. Er hat auch einen ICSP-Anschluss (In Circuit Serial Programming), wenn Sie den Hauptmikrocontroller extern programmieren möchten. Er hat auch 4 SMD-LEDs, zwei, die an die Empfangs- und Sendeleitungen der seriellen Schnittstelle angeschlossen sind, eine, die für die Leistungsanzeige verwendet wird, und eine, die an den digitalen Ein-/Ausgangs-Pin 13 angeschlossen ist.

Er hat einen Reset-Knopf (Bild oben) und zwei Reset-Pins (mit RST gekennzeichnet).

# Az-Delivery

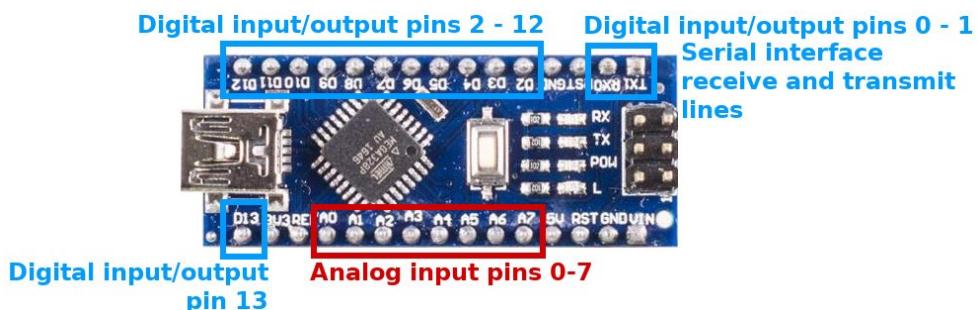


Ein Nano V3.0 verfügt über einen DC-Spannungsregler für +5V, **VIN**-Pin (DC-Eingang auf dem Bild links). Sie können eine externe Gleichstromversorgung an den **VIN**-Pin auf der Platine mit einer Spannung im Bereich von 7V bis 12V anschließen. Der Spannungsregler senkt und stabilisiert sie auf +5V. Die Ausgangsspannung von +3,3V können wir vom CH340-Chip erhalten. Sie ist NICHT genau und somit NICHT stabil.

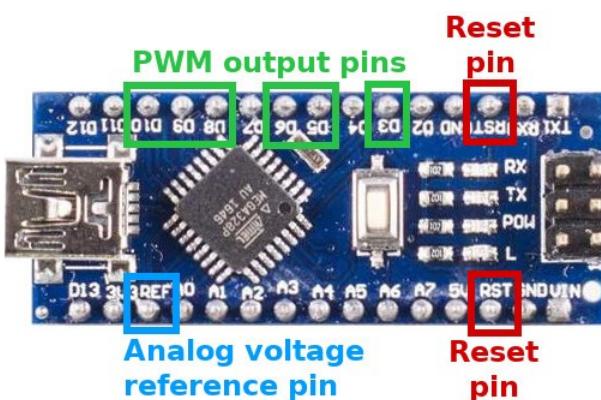
Der Nano V3.0 verwendet den CH340-Chip um zu kommunizieren, wie eine serielle Schnittstelle USART, allerdings über USB. Der CH340-Chip verbindet den USB-Port des PCs mit der seriellen USART-Schnittstelle des Mikrocontrollers und ermöglicht uns so die Programmierung des Mikrocontrollers über den USB-Port.

Sie können auch den Mini-USB-Port zur Stromversorgung des Nano V3.0-Boards verwenden!

# Az-Delivery



Der Nano V3.0 ist so aufgebaut, dass digitale Eingangs-/Ausgangspins von analogen Eingangspins getrennt sind. Es gibt 8 analoge Eingangspins (Die Pins A6 und A7 sind nicht im Arduino Uno enthalten) und davon getrennt 14 digitale Eingangs-/Ausgangspins.



6 von 14 digitalen Ein-/Ausgangspins können als PWM-Ausgänge (Pulsweitenmodulation) verwendet werden. Diese Pins sind D3, D5, D6, D9, D10 und D11.

Um den Hauptmikrocontroller zurückzusetzen, müssen Sie den Reset-Knopf drücken oder einen der Reset-Pins mit GND (0V) verbinden.

# Az-Delivery

Wenn Sie einen Analog-Digital-Wandler verwenden, wird standardmäßig eine analoge Spannungsreferenz von VCC (=+5V) verwendet, aber Sie können jede beliebige analoge Spannungsreferenz verwenden. Sie müssen nur diese Spannungsreferenz mit dem REF-Pin verbinden - Analoger Spannungsreferenzpin (Bild oben). Der Arduino Uno hat diesen Pin nicht.

Die digitalen I/O-Pins D0 und D1 sind mit den Empfangs- und Sendeleitungen der seriellen Schnittstelle verbunden. In der obigen Abbildung sind diese Pins mit RX bzw. TX beschriftet. 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 dem Nano V3.0 hochladen. Eigentlich überall dort, wo Sie Ihren Nano V3.0 neu programmieren oder wann immer Sie die serielle Schnittstelle verwenden. Wenn Sie diese Pins als digitale Ein- oder Ausgänge verwenden, können Sie bei der Neuprogrammierung viele Fehler erhalten oder es führt dazu, dass Ihre Elektronikteile oder daran angeschlossene Geräte falsch funktionieren.

# Az-Delivery

## Kommunikationsschnittstellen

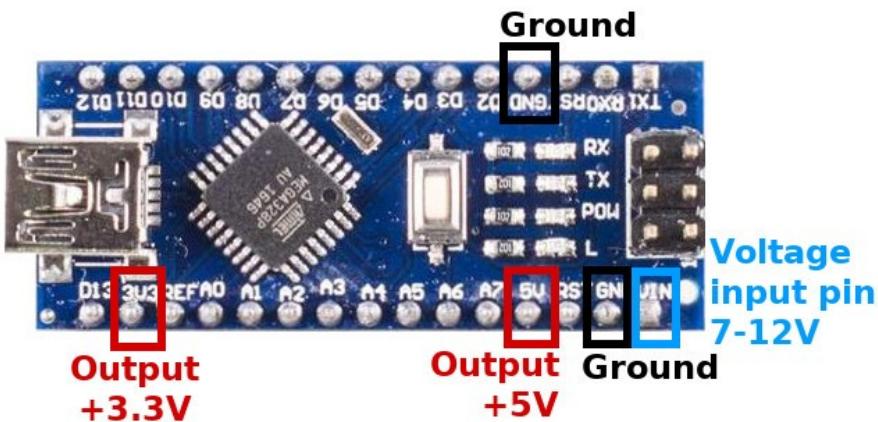
Wir haben bereits erwähnt, dass die digitalen I/O-Pins D0 und D1 alternative Funktionen haben. Sie sind mit den Empfangs- und Sendeleitungen der seriellen Schnittstelle verbunden.

Es gibt zwei weitere Kommunikationsschnittstellen, die vom ATMega328P-Mikrocontroller unterstützt werden, nämlich die serielle Peripherieschnittstelle - SPI und die Inter-Integrated-Circuit-Schnittstelle - 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 jeweils SS, MOSI, MISO und SCK.

Für die I2C-Schnittstelle werden die analogen Eingangspins A4 und A5 verwendet. Ihre Funktionen sind SDA bzw. SCL.

## Power-Pins



Der Nano V3.0 hat keine Power-Stiftleiste wie der Arduino Uno. Er hat +3,3V und +5V Spannungsausgangspins, zwei Masse- (GND) Pins und einen VIN-Pin. Wir haben den VIN-Pin bereits erwähnt, der als DC-Eingangspin für eine externe Stromversorgung verwendet werden kann. Ebenfalls haben wir erwähnt, dass das Nano V3.0 über einen Mini-USB-Anschluss mit Strom versorgt werden kann.

Der Nano V3.0 kann Spannungen von +5V (die geregelt und stabil sind) und +3,3V ausgeben, was nicht sehr genau ist, und somit nicht stabil. Seien Sie also vorsichtig mit empfindlichen elektronischen Geräten, die für die Stromversorgung stabile +3,3V benötigen.



## Nano V3.0-Leistungs-, Strom- und Spannungsbegrenzungen

### Spannungseingangsgrenzen:

Eingangsleistung: Um den Nano V3.0 mit Strom zu versorgen, schließen Sie es entweder an einen USB-Port an, oder eine Spannungsquelle über seine "VIN"- und "GND"-Pins. Wenn das Arduino über diese Pins mit Strom versorgt wird, gibt es folgende Eingangsspannungsbegrenzungen:

» **Empfohlene Eingangsspannungsgrenzen: 7~12V.**

Diese Spannungen können auf unbegrenzte Zeit gehalten werden.

» **Absolutes Limit für die Stromversorgung des Arduino: 6~20V**

- Unterhalb von 7V können die 5V-Pegel auf der Platine schwanken oder sinken, was zu Instabilität der Platine und weniger genauen analogen Messwerten bei der Verwendung von analogRead( ) führt.
- Anhaltende Spannungspegel über 12V führen zu einer zusätzlichen Erwärmung des linearen Spannungsreglers des Nano V3.0, was zu einer Überhitzung des Nano V3.0 führen kann. Kurze Perioden sind jedoch in Ordnung. Fühlen Sie den Spannungsregler mit Ihrem Finger. Wenn er sich zu heiß anfühlt, um ihn komfortabel zu berühren, müssen Sie eine Spannungsquelle innerhalb der empfohlenen Grenzen verwenden, um die Wärmeentwicklung zu reduzieren.

**Vorsicht!!!**



**Bevor Sie irgendwelche ESD-empfindlichen Teile des Nano V3.0 (gesamter Nano V3.0 ) berühren, berühren Sie zuerst den Metallteil des USB-Steckers, um sich an der Platine zu erden und die aufgebaute statische Spannung sicher zu entladen.**

» **Spannungsgrenzen an Ein-/Ausgangspins: -0,5 bis +5,5V max.**

Wenn Sie eine Spannung an einem digitalen oder analogen Nano V3.0-Eingangspin lesen müssen, stellen Sie sicher, dass sie zwischen 0 und 5V liegt. Liegt sie außerhalb dieser Grenzen, können Sie die Spannung mit Hilfe eines Spannungsteilers herabsetzen. Dieser skaliert die Eingangsspannung, um analoge oder digitale Messungen von Spannungen zu ermöglichen, die sonst außerhalb des zulässigen Bereichs liegen würden. Wenn Ihr Eingangssignal digital ist und Sie keine skalierten analogen Messungen vornehmen müssen, besteht eine andere Technik darin, die Eingangsspannung zu kürzen (die Oberseite abzuschneiden), anstatt sie zu skalieren. Da AVR- Mikrocontroller über interne Klemmdiode verfügen, kann dies durch einfaches Hinzufügen eines einzelnen Widerstandes zum Pin erfolgen. Durch Hinzufügen eines  $10k\Omega$  Widerstandes zum Eingangspin (jeder Eingangspin) können Eingangsspannungen bis zu -10,5V oder bis zu +15,5V erreicht werden.

# Az-Delivery

## **Limit der Stromausgabe:**

- » **Maximale Gesamtstromaufnahme des Nano V3.0 bei Versorgung über einen USB-Port: 500mA**
- » **Der Nano V3.0 hat eine "zurücksetzbare Polysicherung, die die USB- Ports Ihres Computers vor Kurzschlägen und Überstrom schützt".**
- » **Maximale Gesamtstromaufnahme bei Versorgung über ein externes Netzteil: 1A**

**Hinweis:** Wenn der Nano V3.0 nicht über USB mit Strom versorgt wird, wird die Gesamtstrombegrenzung von 5V, die aus dem Nano V3.0 kommt, durch den Spannungsregler auf Ihrem speziellen Board und/oder Ihr Eingangsnetzteil begrenzt, je nachdem, was weniger Strom liefert. Nehmen wir an, Ihre Stromversorgung für den Nano V3.0 kann  $7\div 12V$  und  $\geq 1A$  liefern. Wenn dies der Fall ist, wird die 5V-Leistung durch den Spannungsregler Ihres Nano V3.0-Boards streng begrenzt.

- » **Maximale Gesamtstromaufnahme über den Nano V3.0 "5V"-Pin und "GND": wie oben angegeben.**
- » **Maximaler Gesamtstrom pro Ein-/Ausgangspin: 40mA**



» **Summe der Strömflüsse aus ALLEN Ein-/Ausgangspins zusammen: 200mA!!!**

**Hinweis:** Dies wird am häufigsten missverstanden! Trotz der Tatsache, dass Ihr Spannungsregler auf dem Nano V3.0 bis zu 1A über die "5V" und "GND" Pins beziehen kann, kann die Summe aller Ströme, die in die oder aus den Eingangs-/Ausgangspins (alle analogen und digitalen Pins zusammen) des ATMega328P Mikrocontrollers selbst gehen, 200mA nicht überschreiten. Wenn Sie also 10 LEDs mit jeweils 20mA über Ihre Analog- oder Digital-Pins mit Strom versorgen, dann haben Sie gerade Ihr Limit erreicht! Mehr als das und Sie könnten den Mikrocontroller auf dem Nano V3.0-Board beschädigen. Wenn Sie mehr Strom benötigen, ist die Verwendung von Transistoren möglich. Die Nano V3.0-Eingangs-/Ausgangs-Pins können dann einen geringen Stromfluss verwenden, um einen Transistor zu aktivieren, der dann einen höheren Stromfluss direkt vom 5V-Pin (der direkt mit dem Ausgang des linearen Spannungsreglers auf dem Board verbunden ist) zum Gerät, das Sie steuern wollen, ein- und ausschaltet. Auf diese Weise halten Sie die Gesamtausgangssumme der analogen/digitalen Nano V3.0-Pins unter 200mA, während das Erreichen des Grenzwertes von 500mA~1A vom 5V-Pin möglich ist.



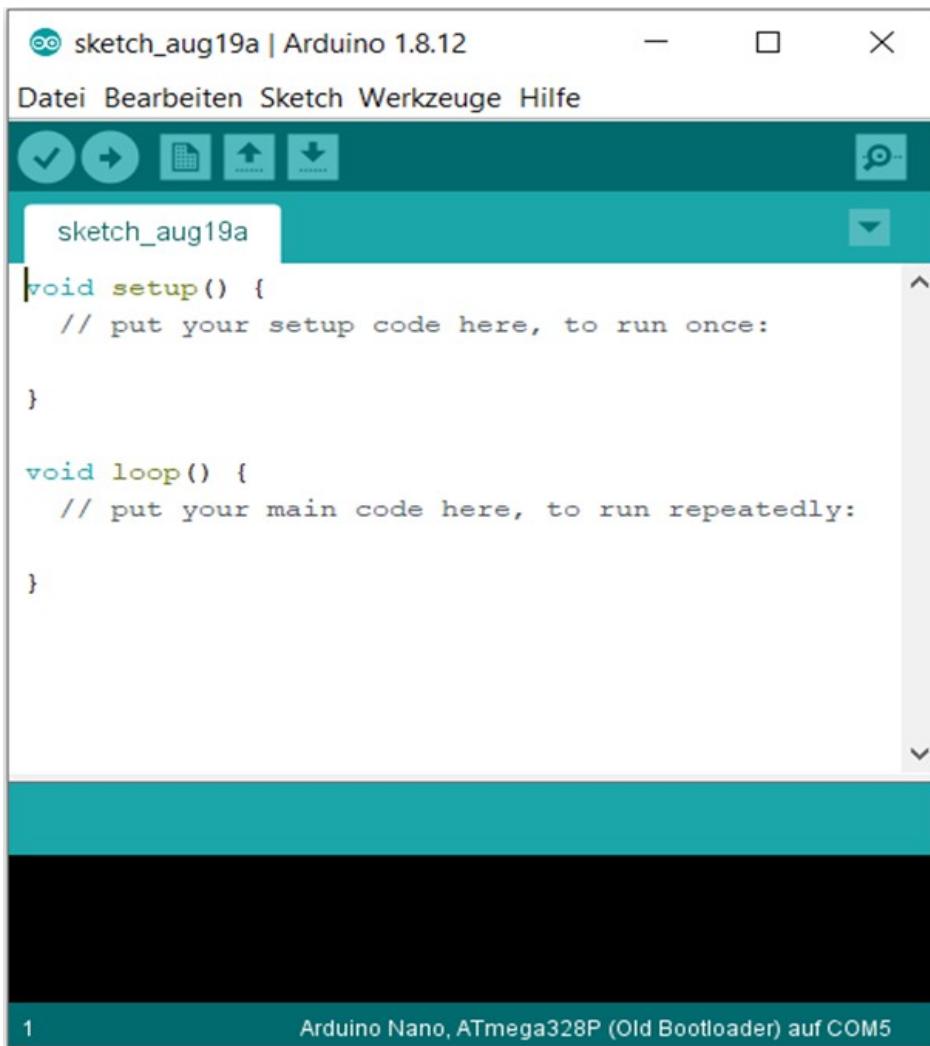
## Arduino IDE

Um ein Nano V3.0-Board zu programmieren, benötigen Sie die IDE-App (Integrated Development Environment). Arduino verfügt über eine Arduino IDE, die auf <https://www.arduino.cc/en/Main/Software> heruntergeladen werden kann.

Suchen Sie einfach Ihr Betriebssystem, laden Sie es herunter und installieren Sie es.

Wenn Sie es installieren und die Anwendung öffnen, wird dies das Startfenster sein.

# Az-Delivery



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_aug19a | Arduino 1.8.12". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". Below the menu is a toolbar with icons for file operations. The main code editor window contains the following code:

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
}
```

At the bottom of the IDE, it says "1" and "Arduino Nano, ATmega328P (Old Bootloader) auf COM5".

Das

geöffnete Programmbeispiel wird als leerer Sketch bezeichnet. Ein Sketch ist ein Programmbeispiel, in das wir unseren Code schreiben. Es hat zwei wesentliche Teile, die `setup()`-Funktion und die `loop()`-Funktion. Es kann auch eine beliebige Anzahl anderer Funktionen haben.

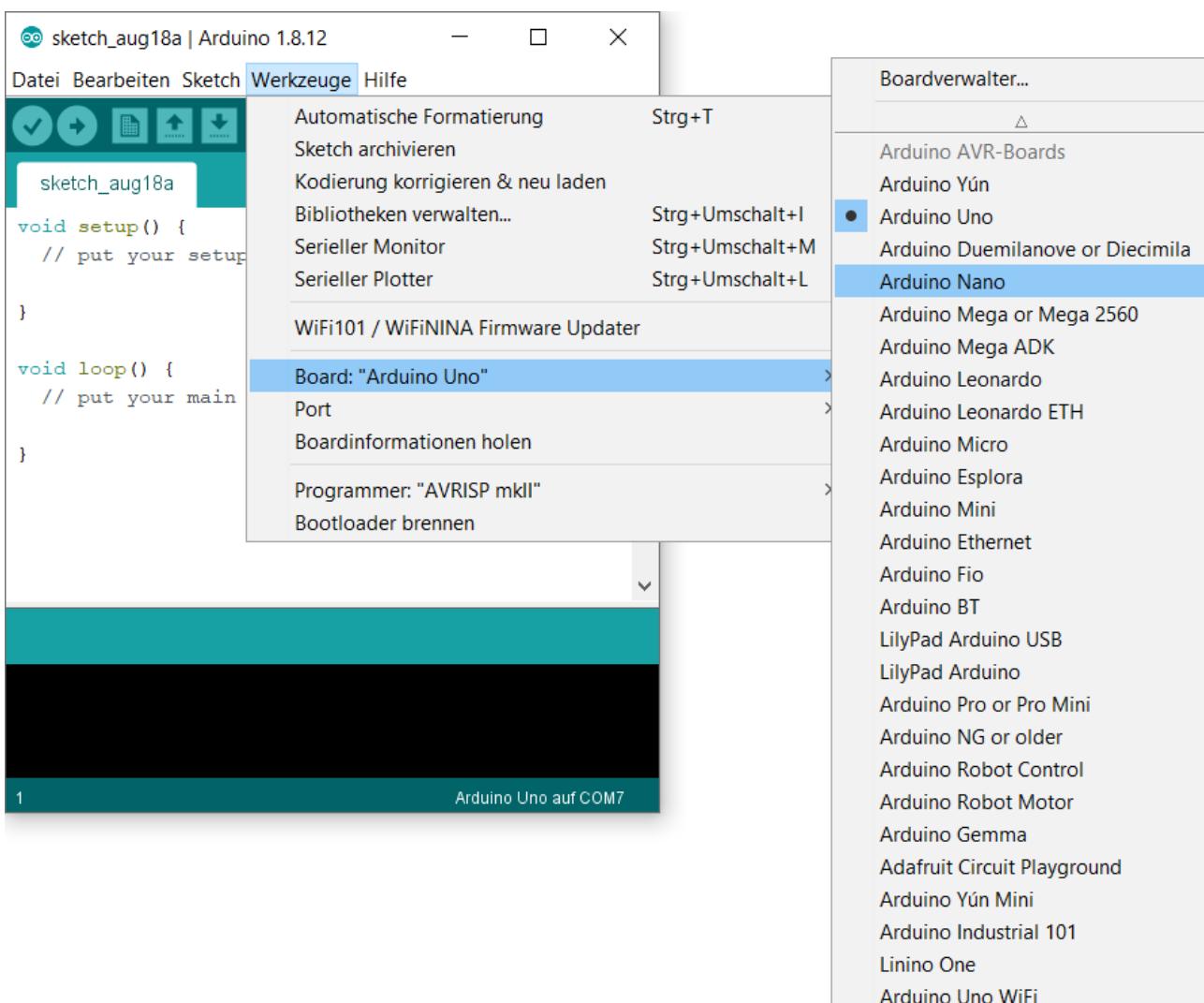
Die `setup()`-Funktion wird nur einmal zu Beginn der Programmausführung durchgeführt. Beim Einschalten der Platine oder beim Zurücksetzen der Platine. In dieser Funktion richten wir alle Initialisierungen ein, z.B. deklarieren wir den Zustand der digitalen Ein-/Ausgangspins oder richten analoge Eingangspins ein, oder richten die serielle Schnittstelle für die serielle Kommunikation ein, usw.

# Az-Delivery

Die `loop()`-Funktion läuft nach dem `setup()` und läuft auf unbestimmte Zeit, immer und immer wieder, die sogenannte "endless loop"-Funktion. Eigentlich läuft sie die ganze Zeit, während die Platine an die Stromversorgung angeschlossen ist. Das liegt daran, dass Programme in elektronischen Geräten niemals das Ende erreichen sollten, denn wenn das passiert, bedeutet das, dass Ihr Gerät so gut wie ausgeschaltet ist. Hier schreiben wir die Logik und die Algorithmen, mit denen unsere Anwendung für das Microcontroller-Board arbeitet.

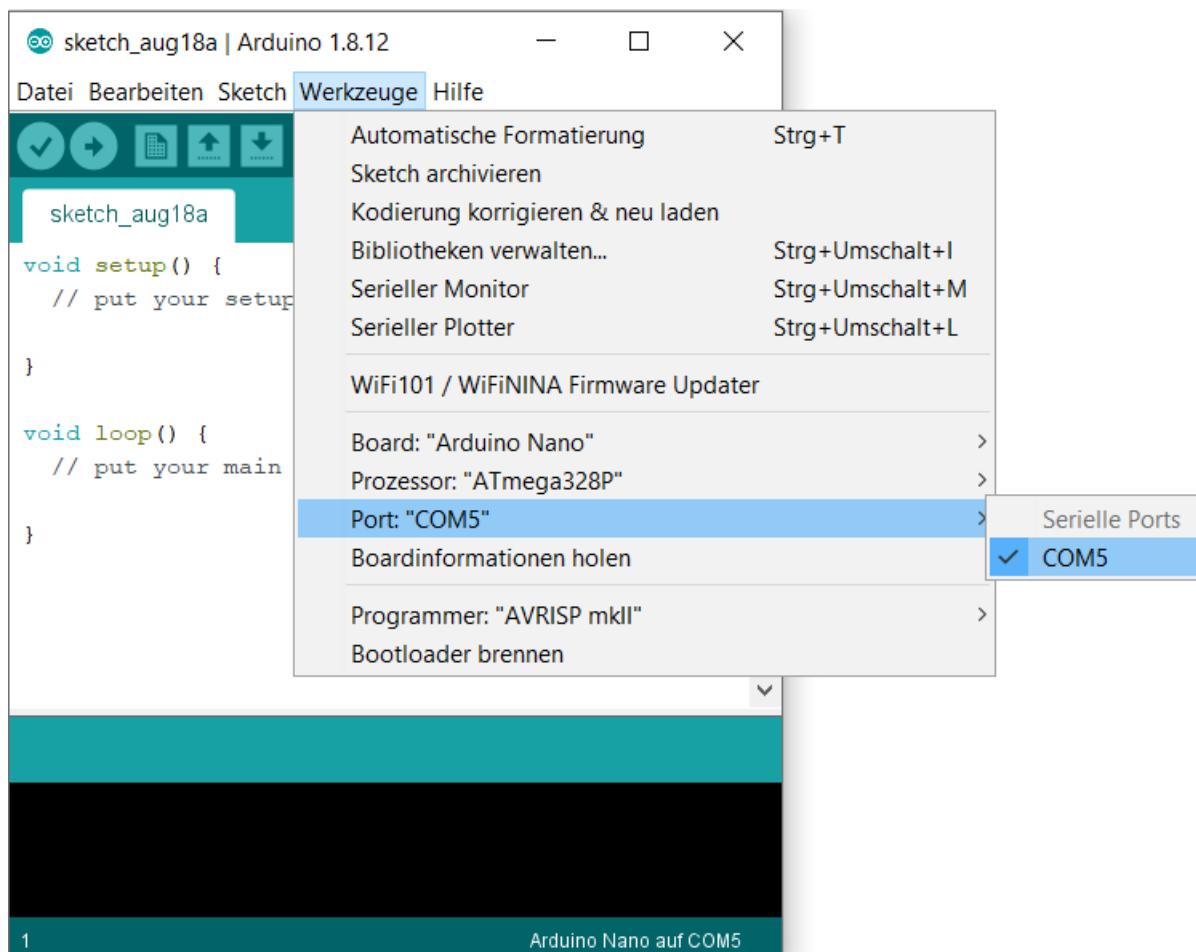
Wenn wir unseren Nano V3.0 über ein USB-Kabel an den PC anschließen, sollten wir in der Arduino IDE als erstes den Nano V3.0 im Menü unter Werkzeuge / Board auswählen.

# Az-Delivery



Als nächstes wählen Sie im Menü unter Werkzeuge / Port die COM-Schnittstelle für die Datenverbindung zwischen PC und Arduino Nano aus.

# Az-Delivery

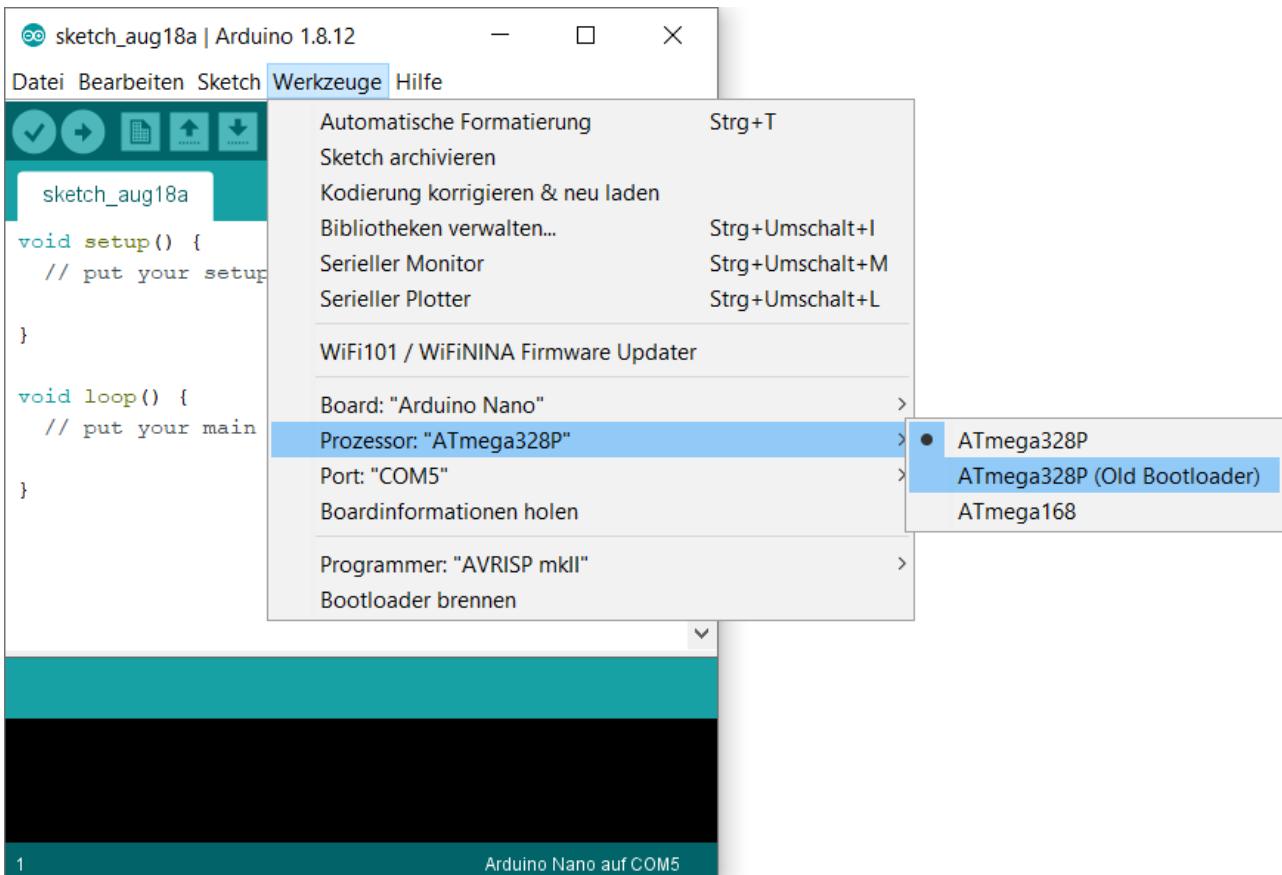


Bei fast allen modernen Betriebssystemen wird diese Schnittstelle automatisch erkannt. Sollte dies bei Ihnen nicht der Fall sein und Sie bekommen keine COM-Schnittstelle angezeigt, müssten Sie zunächst die Treiber-Software für den CH340 (so heißt der Chip auf der Unterseite des Nano V3.0) für Ihr Betriebssystem installieren. Einfach Im Browser-Suchfenster CH340 und Ihr Betriebssystem eingeben. Download, Installation und Neustart dauern maximal fünf Minuten.

Als drittes wählen Sie im Menü unter Werkzeuge / Prozessor den Atmega328P (Old Bootloader) aus.

Damit sind alle Vorbereitungen für den Upload Ihres Sketches erledigt.

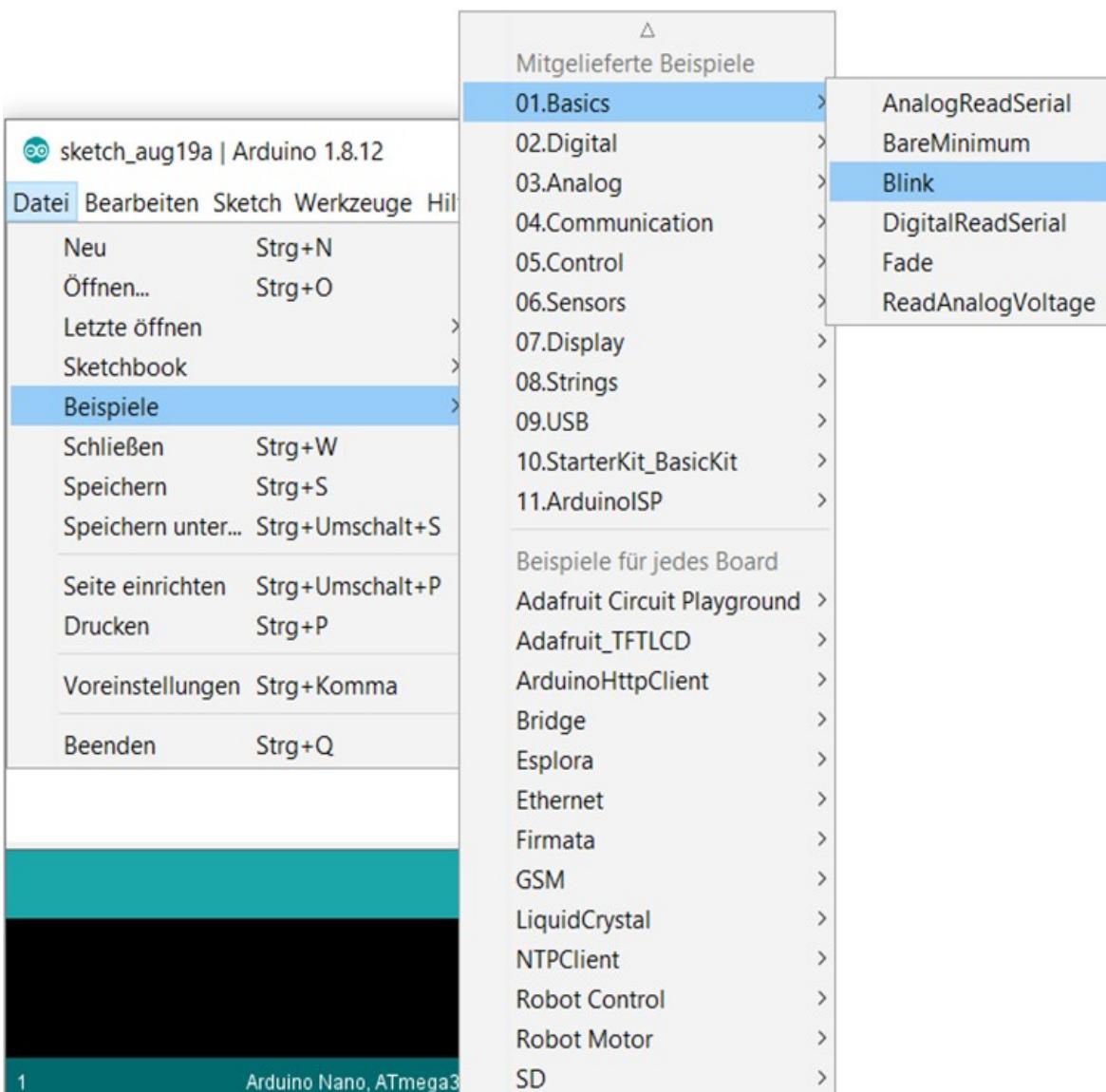
# Az-Delivery



## Anwendungsbeispiel

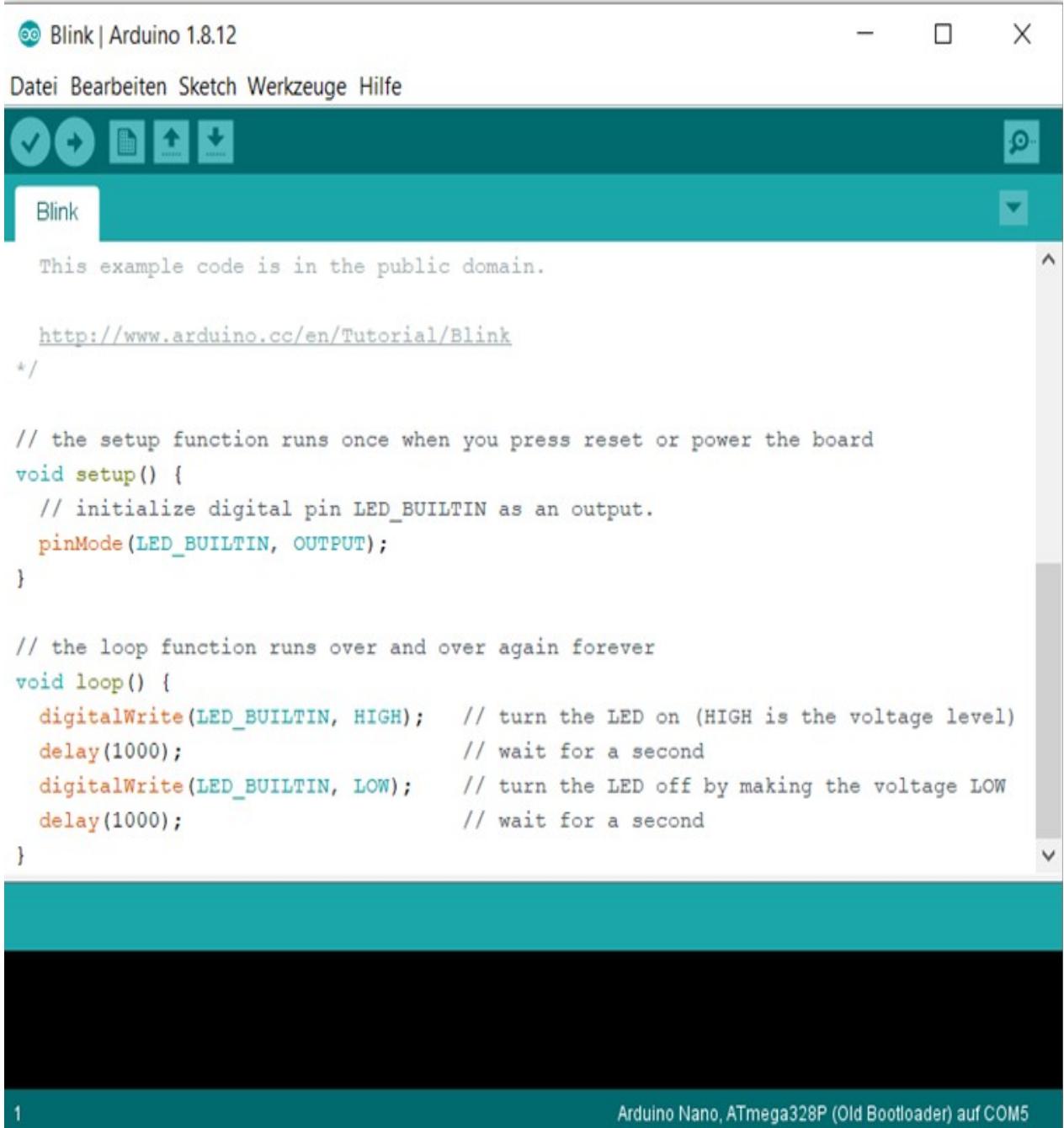
Und jetzt können wir mit dem Programmieren beginnen. Das Arduino IDE kommt mit unzähligen, vorgefertigten Sketchbeispielen, die Sie verwenden können. Hier werden wir das BLINK-Sketchbeispiel verwenden. Gehen Sie zu Datei > Beispiele > 01.Basics > Blink.

# Az-Delivery



Ein neues Fenster mit dem Sketchbeispiel wird sich öffnen:

# Az-Delivery



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.12". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". Below the menu is a toolbar with icons for file operations. The main window displays the "Blink" sketch. A note at the top states "This example code is in the public domain." followed by a link to the Arduino website. The code itself is the standard Blink sketch:

```
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);
}

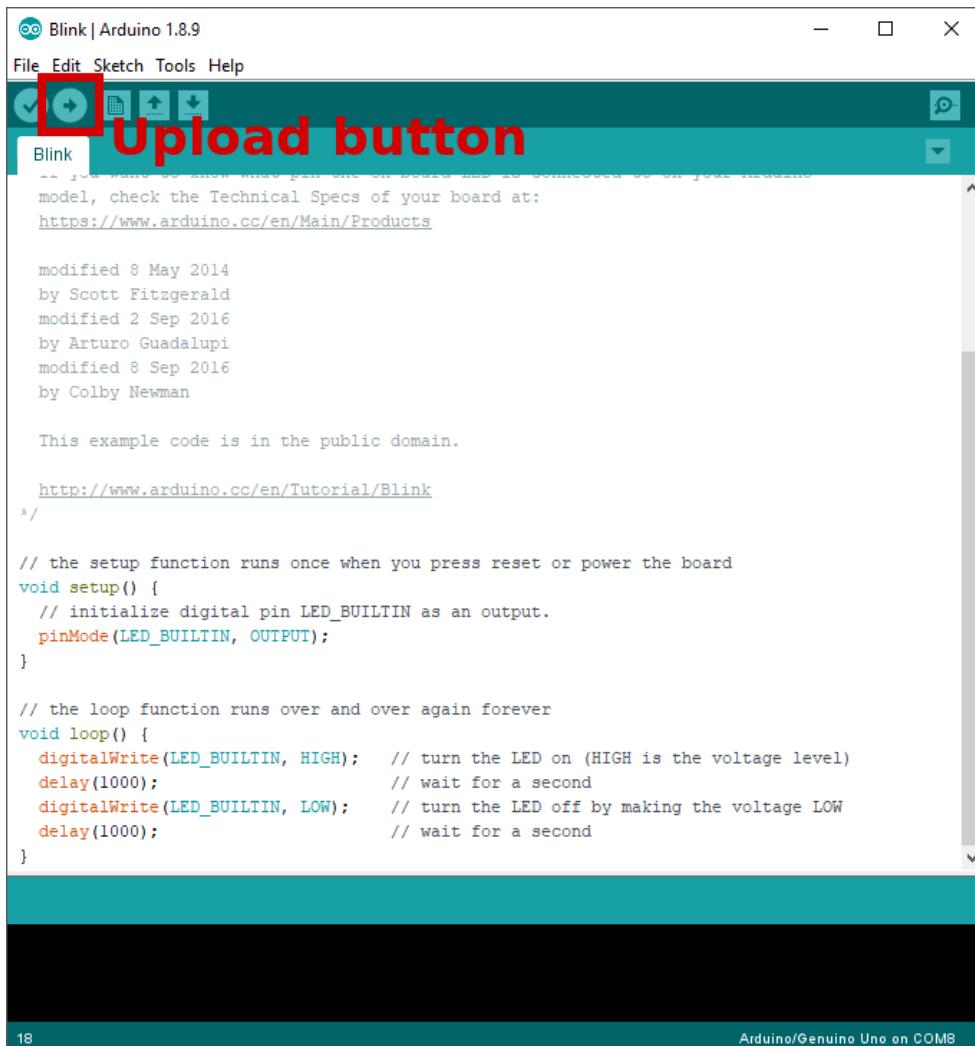
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage LOW
  delay(1000);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

The status bar at the bottom indicates "1" and "Arduino Nano, ATmega328P (Old Bootloader) auf COM5".

In diesem Sketch wird eine an den digitalen I/O-Pin 13 angeschlossene LED auf der Platine eine Sekunde lang eingeschaltet und dann eine Sekunde lang ausgeschaltet. Dieses Ein- und Ausschalten wird als Blinken bezeichnet, daher der Name dieses Sketches.

# Az-Delivery

Wenn wir mit der Programmierung fertig sind, drücken Sie die Upload-Taste, um die Skizze zu kompilieren und auf Ihr Nano V3.0-Board hochzuladen.



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

Danach sollte die eingebaute LED anfangen in einem Zeitintervall von einer Sekunde zu blinken

**Sie haben es geschafft. Sie können jetzt unser Modul für Ihre Projekte nutzen.**



Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

**Falls Sie noch nach weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.**

<https://az-delivery.de>

Viel Spaß!

Impressum

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