

# Arduino Nano mit OPTIBOOT 8 Bootloader

Bei der Drehscheiben-Software haben wir zwei Probleme: einmal haben Hardi und Martin (domapi) wieder viel zu viele Funktionen eingebaut und zweitens ist der Speicher eines Nano leider begrenzt. Das erstere werden wir nicht mehr ändern können, aber dem Nano kann man zu etwas mehr Speicher verhelfen.

Sobald nach dem Kompilieren angezeigt wird „... Das Maximum sind 30720 Bytes.“, gibt es akuten Handlungsbedarf!

Um zusätzlichen Programm-Speicher freizuschaufeln, muss man dem Nano einen neuen Bootloader aufspielen. Diesen kann man hier runterladen: [Optiboot 8](#)

Das Prozedere ist hier recht ausführlich beschrieben: [Bootloader brennen](#)

Am besten macht man das gleich für mehrere Nanos nacheinander, z.B. mit einem Uno als Programmer, an dem die Nanos als Target angeschlossen werden. Das baut man sich auf dem Steckbrett auf und macht das in einem Rutsch für mehrere Nanos.

Auf den Uno muss zunächst ein Sketch aufgespielt werden („ArduinoISP“), dann kann der Uno die Nanos mit dem Bootloader versorgen.

Auf wundersame Weise haben wir dann ca. 1,5 kByte Speicher zusätzlich zur Verfügung. Zudem funktioniert nun auch der sogenannte Watchdog des Nano endlich richtig ...

## Arduino-IDE

- Brennen neuen Bootloader Optiboot 8.0 mit folgenden Einstellungen über einen Uno als ISP-Programmer (Standard-sketch „ArduinoISP“; <https://github.com/Optiboot/optiboot/releases>; Beschreibung: [Bootloader brennen – AZ-Delivery](#)

## Arduino-IDE-updaten:

1. Neue Version von Arduino.cc downloaden
2. Exe-File ausführen
3. Alte Version muss deinstalliert werden
4. Neue installieren; Sketch bleiben erhalten
5. Optiboot Bootloader ison-Link in den Boardverwalter eintragen

**Zusätzliche Boardverwalter-URLs**

Geben Sie weitere URLs ein, eine in jeder Zeile

`https://github.com/esp32boot/optiboot/releases/download/v9.0/package_optiboot_optiboot-additional_index.json`

Klicken Sie für eine Liste mit inoffiziellen Boardunterstützung-URLs.

oder so

Werkzeuge	Hilfe
Automatische Formattierung	Strg+T
Sketch archivieren	
Kodierung konfigurieren & neu laden	
Bibliotheken verwalten...	Strg+Umschalt+I
Serieller Monitor	Strg+Umschalt+M
Serieller Plotter	Strg+Umschalt+L
WiFi101 / WIFINANO Firmware Updater	>
Board: "Arduino Nano"	>
Prozessor: "ATmega328P (New Bootloader full Mem)"	>
Port: "COM3"	>
Formatvorlagen ändern	>
Formatvorlagen holen	>

The diagram illustrates the connection setup for programming an Arduino Nano. An Arduino Uno is connected to an Arduino Nano via a USB cable. The Arduino Uno is also connected to a 3.3V power source. The connections are as follows:

- USB cable: Connects the Arduino Uno to the Arduino Nano.
- 3.3V power source: Connects to the 3.3V pin on the Arduino Uno.
- 5V pin on the Arduino Uno: Connects to the 5V pin on the Arduino Nano.
- GND pin on the Arduino Uno: Connects to the GND pin on the Arduino Nano.
- TX pin on the Arduino Uno: Connects to the RX pin on the Arduino Nano.
- RX pin on the Arduino Uno: Connects to the TX pin on the Arduino Nano.
- RESET pin on the Arduino Uno: Connects to the RESET pin on the Arduino Nano.
- AVGND pin on the Arduino Uno: Connects to the GND pin on the Arduino Nano.
- 5V pin on the Arduino Uno: Connects to the 5V pin on the Arduino Nano.
- TX pin on the Arduino Uno: Connects to the RX pin on the Arduino Nano.
- RX pin on the Arduino Uno: Connects to the TX pin on the Arduino Nano.
- RESET pin on the Arduino Uno: Connects to the RESET pin on the Arduino Nano.
- AVGND pin on the Arduino Uno: Connects to the GND pin on the Arduino Nano.

Programmer Type : Arduino  
Description : Arduino  
Hardware Version: 3  
Firmware Version: 8.0  
Vtarget : 0.5 V  
Vref : 0.3 V  
Oscillator : 28.000 kHz  
SCK period : 3.3 us

Firmware-Version 8.0  
32256 Bytes III

**32256** Bytes sind irre viel (die ersten Mondraketen lagen auch in diesem Bereich).

Der Nano hat aber lediglich Programmspeicher dazu gewonnen. Der sogenannte SRAM-Speicher für die Variablen zur Laufzeit beträgt trotzdem nur 2048 Bytes. Das ist leider meistens zu wenig. Da muss

man tricksen, um z.B. Konstanten in den Programmspeicher zu bringen und damit SRAM einzusparen. Das ist aber wieder eine neue Geschichte. Weiterhin bleibt es bei den 1024 Bytes EEPROM. Den nutzt der Sketch recht ausgiebig, um die DS-Positionen zu speichern, damit sie bei nächsten Einschalten wieder verfügbar sind und nicht neu definiert werden müssen.

**Achtung:** Falls ihr die Arduino-IDE irgendwann mal updated, muss der Optiboot Bootloader json-Link erneut in den Boardverwalter eintragen werden!

From:  
<https://wiki.mobaledlib.de/> - **MobaLedLib Wiki**

Permanent link:  
[https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn\\_v10/150\\_locoturn\\_nano?rev=1675864932](https://wiki.mobaledlib.de/anleitungen/bauanleitungen/locoturn_v10/150_locoturn_nano?rev=1675864932)

Last update: **2023/02/08 14:02**

