

Diese Seite verwendet eigene Cookies und Cookies von Dritten damit wir die bestmögliche Bedienbarkeit und Funktionalität bieten können. Wenn Sie auf diesen Seiten surfen, stimmen Sie der Verwendung von Cookies zu. [Mehr erfahren](#) OK

- [Home](#)
- [Sammlung](#)
- [Datenbank](#)
- [Anleitungen](#)
- [History](#)
- [Community](#)
- [Links](#)
- [Kontakt](#)
- [Info](#)

Amiga-Disketten lesen und beschreiben mit Amiga Floppy Disk Reader

Was wird benötigt?

- Verschiedene elektr. Komponenten (Siehe Text)
- Windows-Rechner
- Amiga-Disketten
- [Arduino IDE](#)
- [Firmware](#)
- [ArduinoFloppyReader](#)

Wer einen Amiga besitzt, oder auch nicht, kommt manchmal in die missliche Lage, Disketten retten zu wollen, die aber vom PC-Diskette-Laufwerk nicht gelesen werden können. Anders rum, also vom PC zum Amiga haben wir das selbe Problem, denn irgendwie muss eine Server-Software auf den Amiga gelangen die Daten empfängt und sendet. Abhilfe schafft dieses Projekt mit den ein PC-Diskettenlaufwerk mittel USB an den Rechner angeschlossen wird. Das tolle an der Sache, es können Amiga-Disketten gelesen und beschrieben werden, direkt am PC unter Windows 10.

Danke an Robert Smith der mir gestattet hat, [seine Anleitung \(amiga.robsmithdev.co.uk\)](#) auf Deutsch zu übersetzen und für Anfänger etwas anzupassen. Siehe auch das entsprechende [Github Projekt](#). ([GNU General Public License v3.0](#)). Wer den Autor mit einer Spende danken möchte, kann dies gerne auf seiner Spenden-Seite machen: [justgiving.com](#) oder [paypal.me](#).

Bevor es los geht, noch ein paar Hinweise (**Wichtig!**)

- Vermeide statische Ladungen durch das Berühren eines geerdeten Objektes, wie beispielsweise einen Heizkörper. Oder verwende ein antistatisches Armband.
- Sorge dafür, dass die elektronischen Teile isoliert sind. Ein Kurzschluss kann den Arduino oder andere Teile beschädigen.
- Ein eventuell vorhandenes Programm im Arduino wird überschrieben wenn ein neues geflasht wird.

- Aktiviere den Schreibschutz von Disketten die nicht beschrieben werden sollen.
- Ich übernehme keine Verantwortung über Schäden die direkt oder indirekt durch diese Anleitung entstanden sind.

Die Komponenten

Folgende Komponenten werden für dieses Projekt benötigt:

[Arduino Pro Mini](#) (In meinem Fall ein 5V, 16Mhz, ATmega328)

[USB zu TTL Seriell Modul FT232RL FTDI](#)

[PC Disketten-Laufwerk](#)

[1KΩ Widerstand](#)

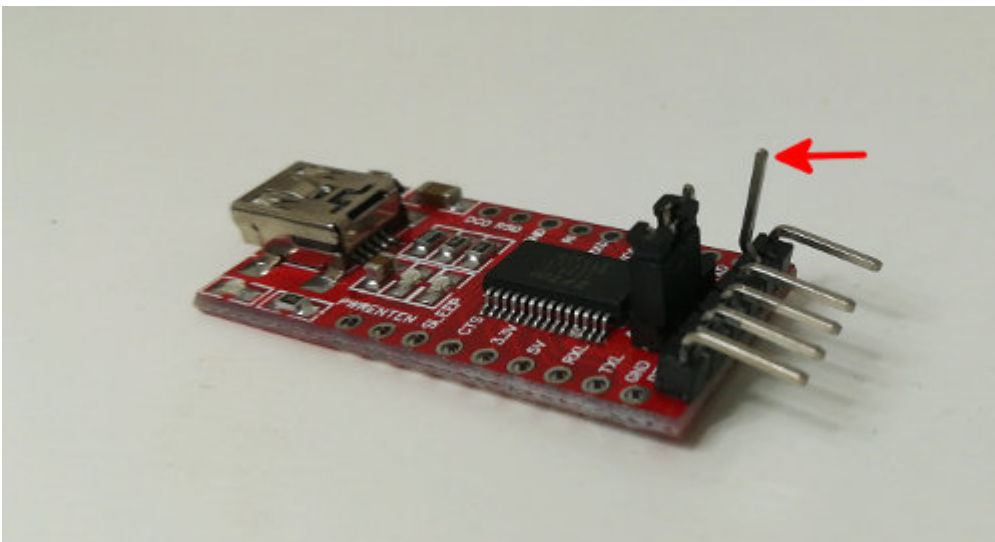
Draht-Brücken [Stecker-Buchse](#), [Stecker-Stecker](#) und [Buchse-Buchse](#)

[Experimentierplatine](#)

Für die Spannungsversorgung des Disketten-Laufwerkes wird ein PC-Netzteil oder eine andere externe Spannungsquelle mit 5V/12V empfohlen. Nicht jedes Disketten-Laufwerk benötigt die 12V, doch auch in diesem Fall sollte man sich für eine externe Spannungsversorgung entscheiden und nicht die 5V der USB-Schnittstelle verwenden, da diese möglicherweise nicht genügend Strom liefern kann.

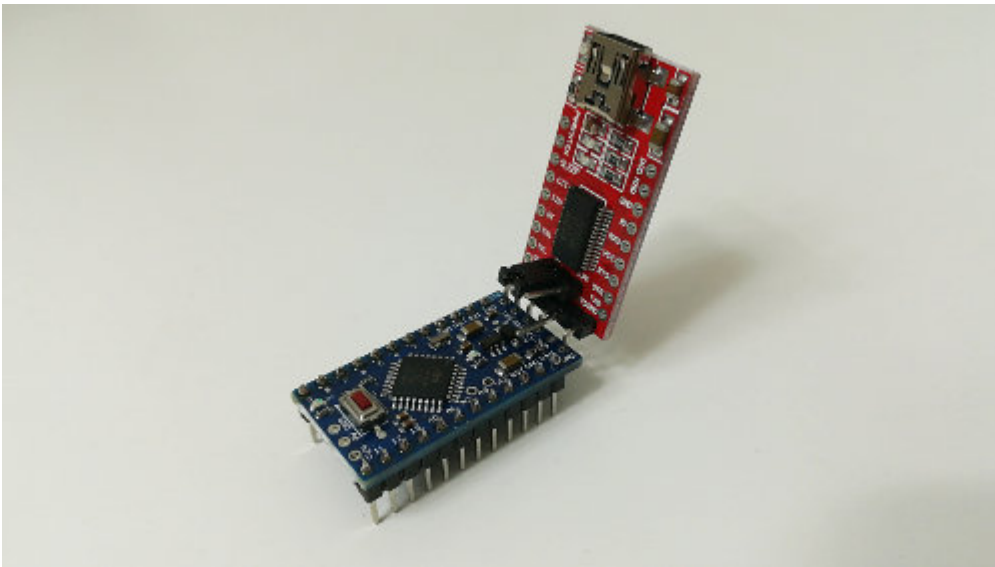
Die Arduino Firmware flashen

Damit der Arduino geflasht werden kann, muss der FTDI-Converter an den Arduino gesteckt werden und zwar so, dass die Beschriftungen der Pins (5V,GND,DTR...) übereinstimmen, abgesehen vom Pin CTS der nach oben gebogen werden muss.



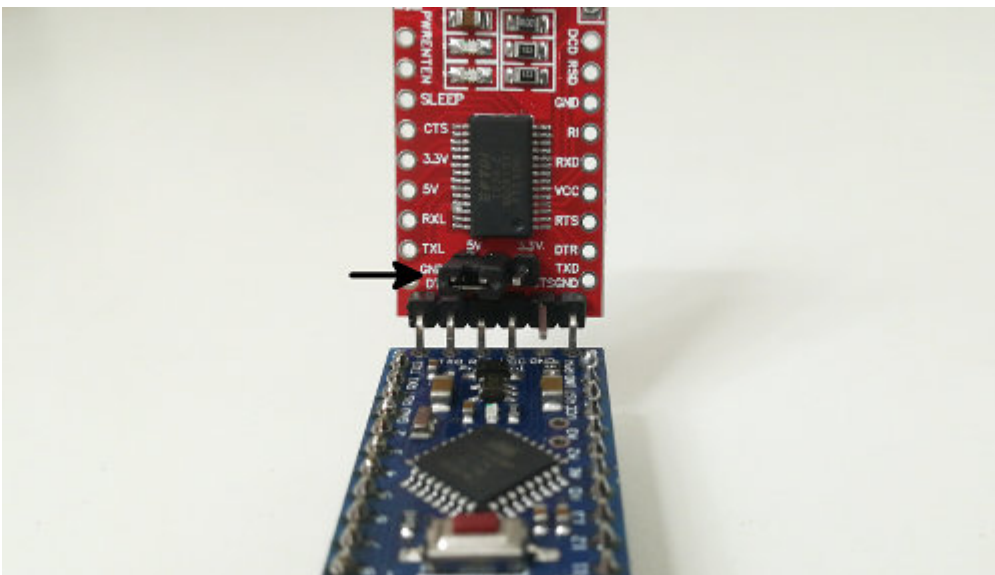
Klicke auf das Bild zum Vergrößern

Für den Flashvorgang wird CTS (Datenflusskontrolle) nicht benötigt. Die restlichen Pins des FTDI-Converters können nun an den Arduino angelötet werden.



Klicke auf das Bild zum Vergrößern

Bevor wir zur Software über gehen, sollten wir noch sicherstellen, dass der Jumper am FTDI-Converter auf 5V eingestellt ist.

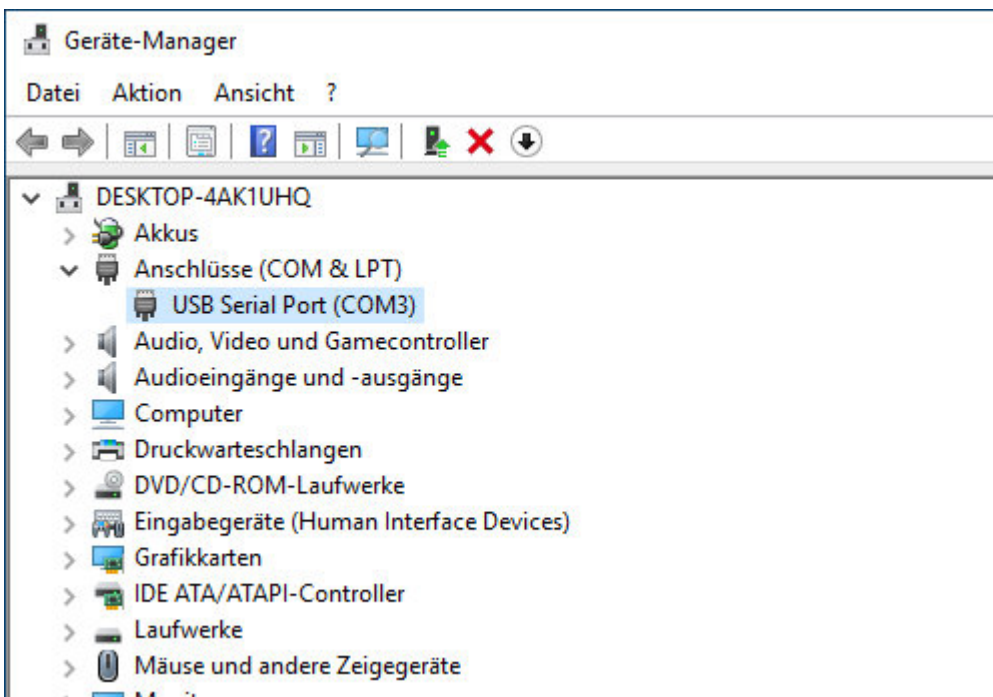


Klicke auf das Bild zum Vergrößern

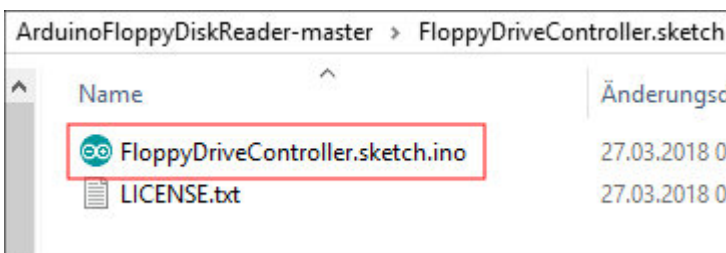
Und nun zum Windows-Treiber: Damit Windows den FTDI-Converter erkennen kann, wird ein Treiber benötigt. Welcher das ist, hängt vom verwendeten FTDI-Converter ab. Der Treiber für den FTDI-Converter der in dieser Anleitung von mir verwendet wurde ([siehe hier](#)), ist auf der folgenden Webseite zu finden: ftdichip.com. Suche nach den Windows-Treiber und lade diese runter.

Comments
<p>WHQL Certified. Includes VCP and D2XX. Available as a setup executable Please read the Release Notes and Installation Guides.</p>
<p>All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0-19 Refer to TN-101 if you need a custom VCP VID/PID in Linux VCP drivers are integrated into the kernel.</p>

Das Archiv enthält eine Setup-Datei die lediglich ausgeführt werden muss. Nach erfolgter Installation verbinden wir den FTDI-Converter mit den PC mittels USB-Kabel. Das FTDI-Modul sollte erkannt werden. Um sicherzustellen, dass alles korrekt installiert wurde, öffnen wir den Geräte-Manager von Windows und suchen den entsprechenden Treiber.



Die COM-Adresse neben den Geräte-Namen notieren wir uns. Als nächstes müssen wir die Arduino-IDE Software runterladen ([Download](#)) und installieren diese. Dann laden wir die Firmware für den Arduino runter ([Download](#)). Sie befindet sich im Ordner mit der Endung .sketch und nennt sich FloppyDriveController.sketch.ino.



Diese Datei öffnen wir nun mit Arduino IDE

```
// FloppyDriveController.sketch | Arduino 1.8.5
Datei Bearbeiten Sketch Werkzeuge Hilfe

FloppyDriveController.sketch

// This sketch manages the interface between the floppy drive and the computer as well as the //
// low-level disk reading and writing. For more information and how to connect your Arduino //
// to a floppy drive and computer visit https://amiga.rohamithdex.co.uk //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#define BAUDRATE 2000000 // The baudrate that we want to communicate over (2M)
#define BAUD_PRESCALER_NORMAL_MODE (((F_CPU / (BAUDRATE * 16UL)) - 1)
#define BAUD_PRESCALER_DOUBLESPEED_MODE (((F_CPU / (BAUDRATE * 8UL)) - 1)
#define UART_USE_DOUBLESPEED_MODE // We're using double speed mode

#define MOTOR_TRACK_DECREASE HIGH // Motor directions for PIN settings
#define MOTOR_TRACK_INCREASE LOW

// PIN 2 - INDEX PULSE PIN - used to detect a specific point on the track for sync. Not used by standard Amiga disks but
#define PIN_INDEX_DETECTED 2 // Pin used to detect the index pulse
#define PIN_INDEX_PORT PIND
#define PIN_INDEX_MASK B00000100

// PIN 3 - WRITE DATA
#define PIN_WRITE_DATA 3 // Raw triggering of writing data to the disk
#define PIN_WRITE_DATA_PORT PORTD // The actual port the above pin is on
#define PIN_WRITE_DATA_MASK B00001000 // The mask used to set this pin high or low

// PIN 4 - READ DATA
#define PIN_READ_DATA 4 // Reads RAW floppy data on this pin
#define PIN_READ_DATA_MASK B00010000 // The mask for the port
#define PIN_READ_DATA_PORT PIND // The port the above pin is on
```

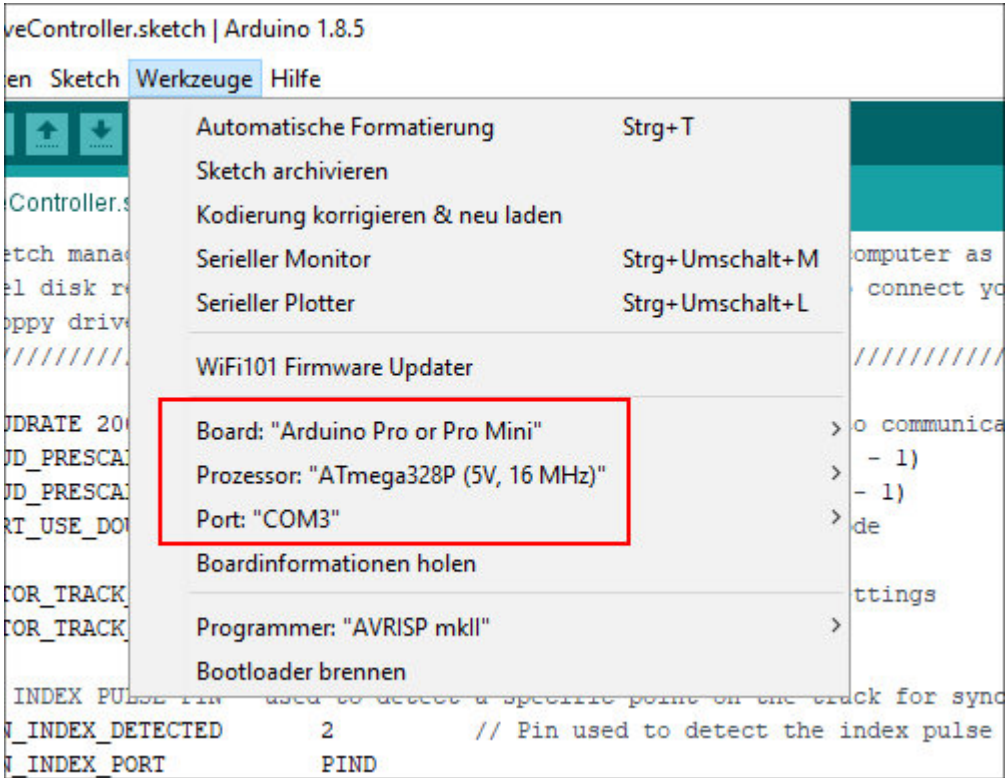
Klicke auf das Bild zum Vergrößern

Anschließen müssen folgende Einstellungen in Arduino IDE vorgenommen werden damit der Arduino Mini geflasht werden kann.

Board: Arduino Pro or Arduino Pro Mini

Prozessor: ATmega328P (5V,16 Mhz)

Port: Die Port-Nummer haben wir weiter oben im Geräte-Manager neben den Geräte-Namen gesehen. In meinem spezifischen Fall ist es COM3.

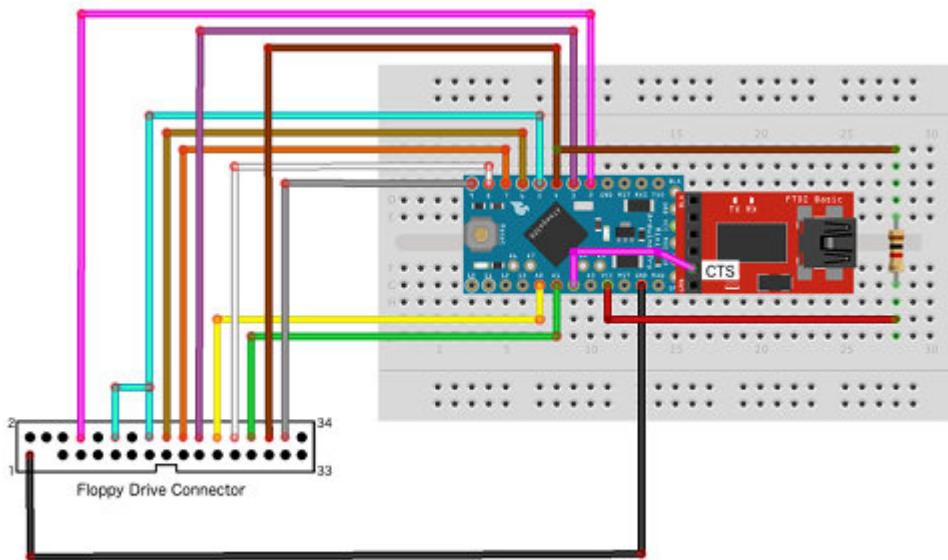


Endlich kann die Firmware geflasht werden, mit einen Klick auf die Taste "Hochladen" oder mit der Tastenkombination STRG+U.

```
Hochladen abgeschlossen.  
Der Sketch verwendet 3114 Bytes (10%) des Programmspeicherplatzes. Das  
Globale Variablen verwenden 269 Bytes (13%) des dynamischen Speichers,
```

Die Verdrahtung

Die Verdrahtung ist sehr einfach und mit wenig Draht-Brücken erledigt.

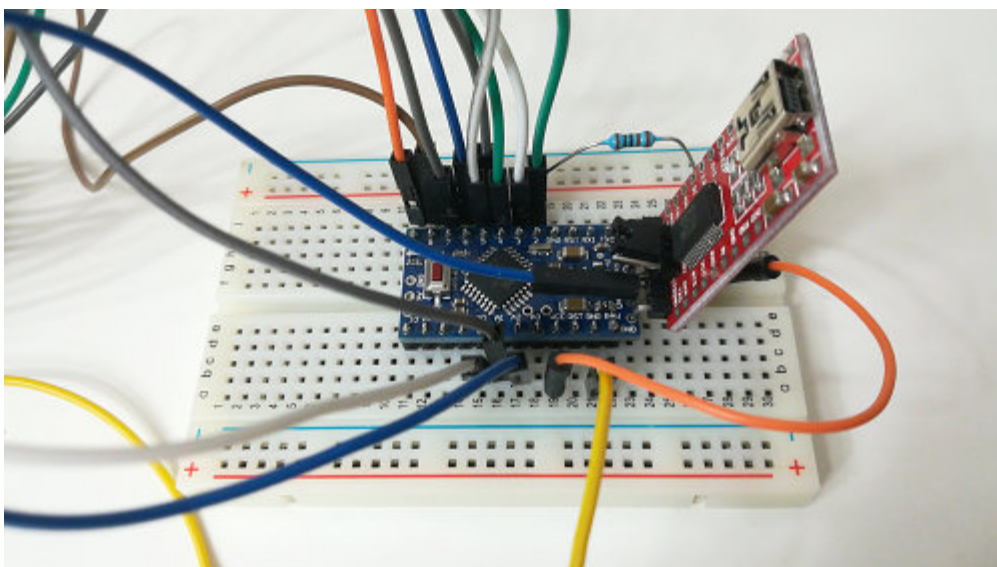


Original Verdrahtung von Robert Smith (amiga.robsmithdev.co.uk)

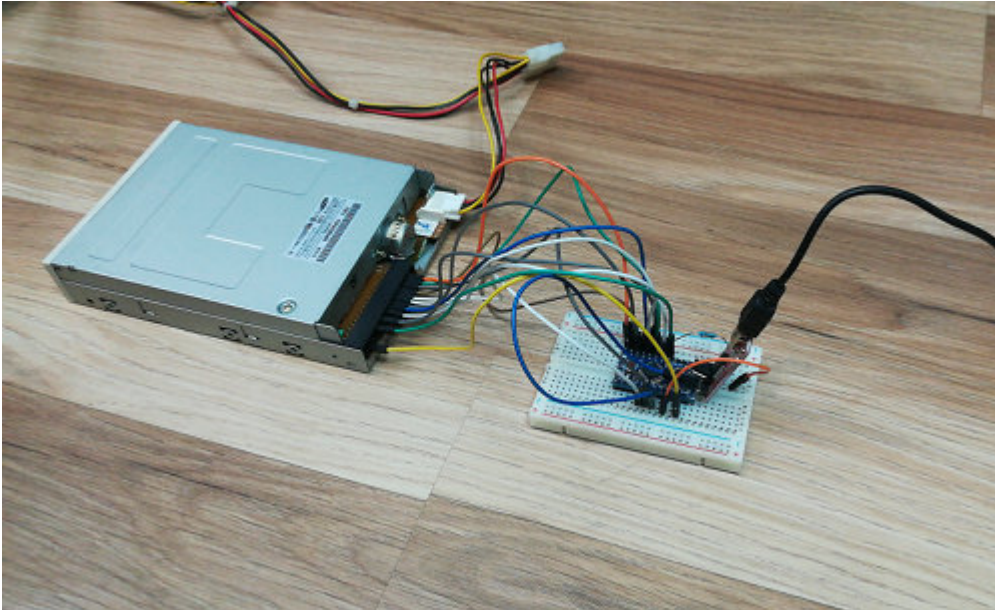
fritzing

Klicke auf das Bild zum Vergrößern

Hier kann die Verdrahtung in höherer Auflösung runtergeladen werden: [Download](#). Das könnte dann am Ende so aussehen.



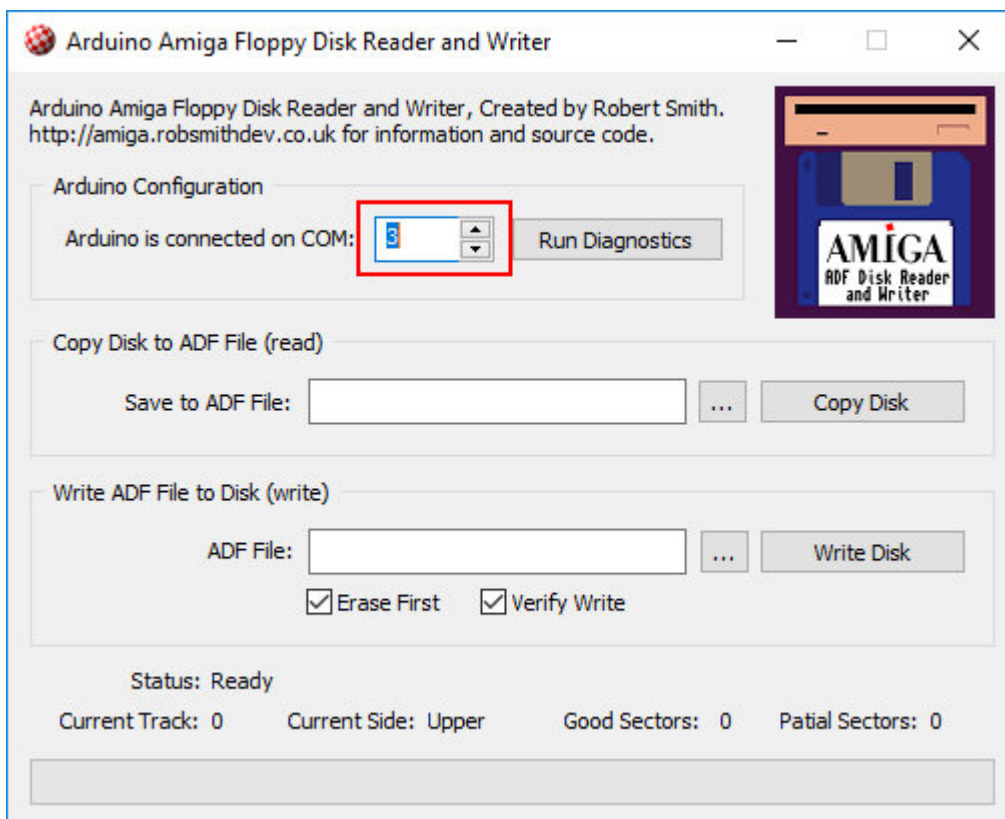
Klicke auf das Bild zum Vergrößern



Klicke auf das Bild zum Vergrößern

Die Software

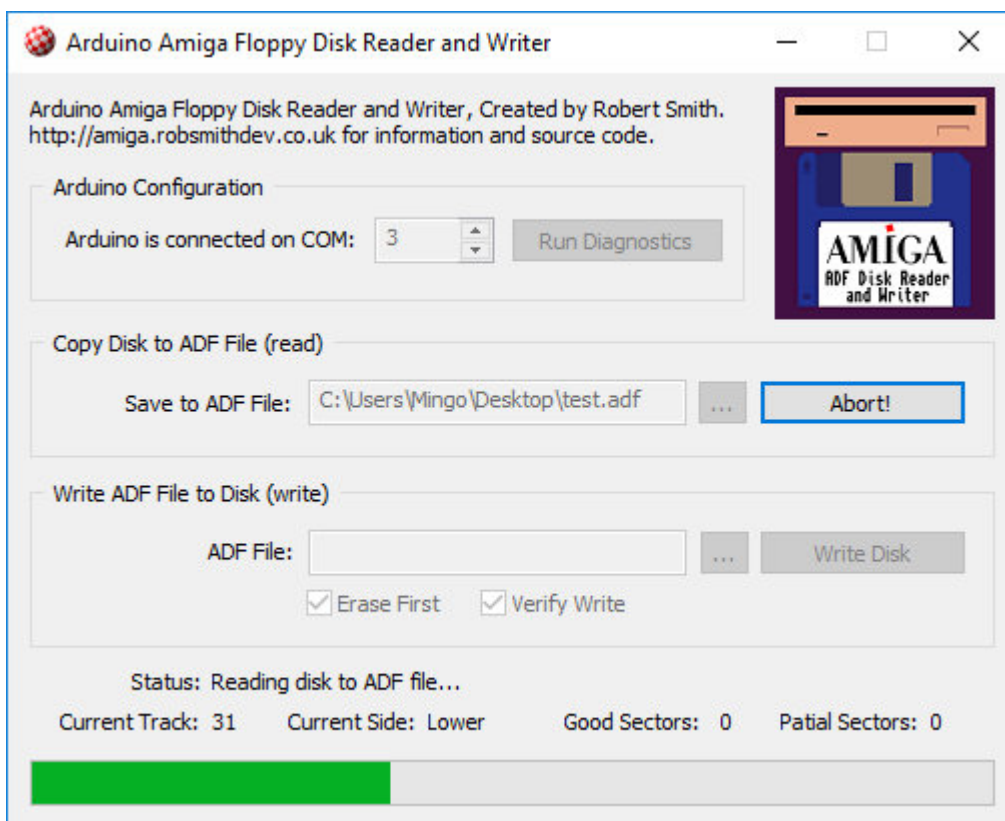
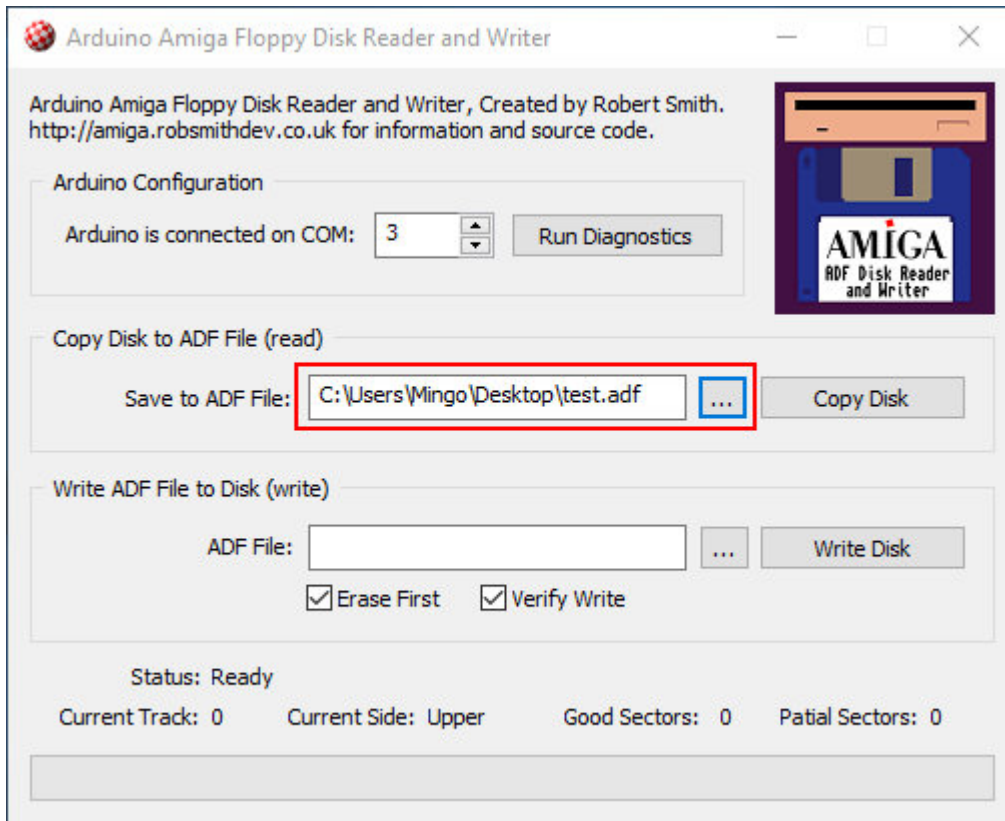
Die Software die wir zum Übertragen der Disketten-Images brauchen finden wir in bereits kompilierter Form hier: [Download](#). Wir starten das Programm "ArduinoFloppyReaderWin.exe" und geben den COM-Port an den unser Arduino hängt. In meinen Fall Port 3.



Anschließend sollte mit einem Klick auf "Run Diagnostics" das Diagnose-Programm gestartet werden. Folge den Anweisungen des Programmes. Eine Amiga-Diskette wird für den Test benötigt. Verwende am besten eine leere, formatierte und fehlerfreie Diskette.

```
Running Diagnostics on COM 3
Attempting to open and use COM 3 without CTS
Testing CTS pin
CTS OK. Reconnecting with CTS enabled
Enabling the drive (please listen and watch the drive)
Asking the Arduino to find Track 0
Track 0 was found. Asking the Arduino to find Track 70
Starting drive, and seeking to track 40.
Checking for INDEX pulse from drive
Checking for DATA from drive
Attempting to read a track from the UPPER side of the disk
Attempting to read a track from the LOWER side of the disk
Reading was successful!
Writing and Verifying Track 41 (Upper Side).
Hurray! Writing was successful. Your Arduino is ready for use! - Send us a photo!
```

Wenn alle Tests erfolgreich abgeschlossen wurden, können wir mit "Copy Disk to □ ADF file" eine Amiga-Diskette als □ ADF-Image auf der PC-Festplatte speichern. Es muss lediglich der Speicherort angegeben und dann auf "Copy Disk" geklickt werden. Die Diskette wird zwar nur gelesen aber es wird immer empfohlen den Schreibschutz zu aktivieren um Datenverluste zu vermeiden.



Genau so einfach lassen sich □ ADF-Images auf eine Amiga Diskette schreiben, mit "Write □ ADF File to Disk". Zuerst muss die gewünschte □ ADF-Datei gewählt und mit "Write Disk" der Schreibvorgang eingeleitet werden.

