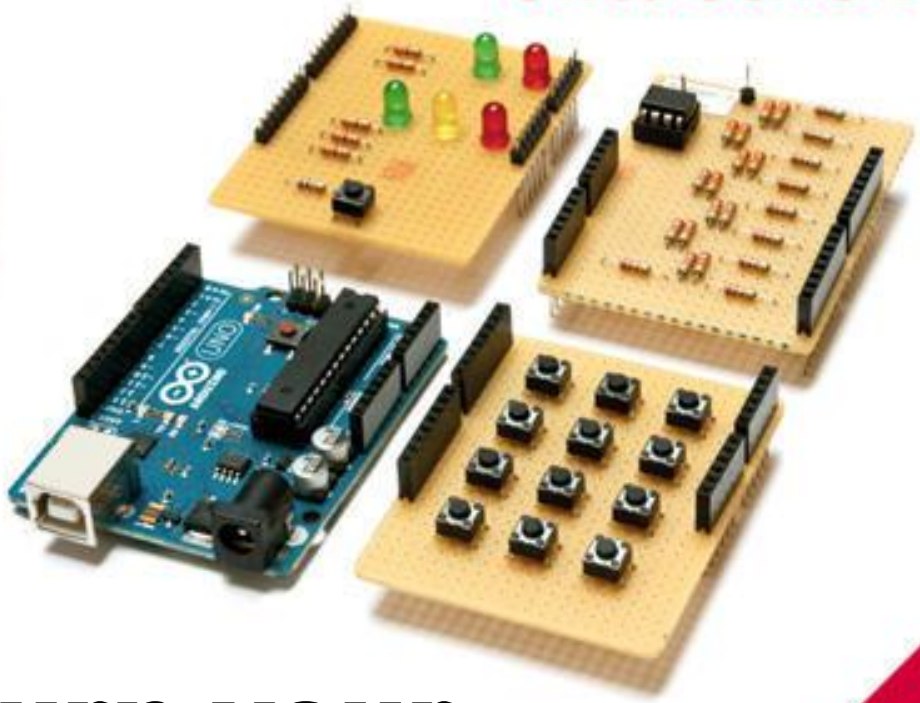
Behandelt Arduino 1.0

Die elektronische Welt mit Arduino entdecken

Erik Bartmann

o'reillys basics



O'REILLY

Burn your Bootloader

AddOn

Scope	Einen eigenen Arduino-Bootloader brennen
Version	1.0
Created	19.03.2012
Autor	Erik Bartmann
Internet	http://www.erik-bartmann.de
Email	arduino@erik-bartmann.de

Inhaltsverzeichnis

Wir brennen uns einen eigenen Arduino-Bootloader 4

Der Arduino am AVR-Programmer 4

DIP-Switches des AVR-Programmers 5

Ermitteln des COM-Ports des AVR-Programmers 6

AVR-Programmer Anpassungen 8

Bootloader brennen 9

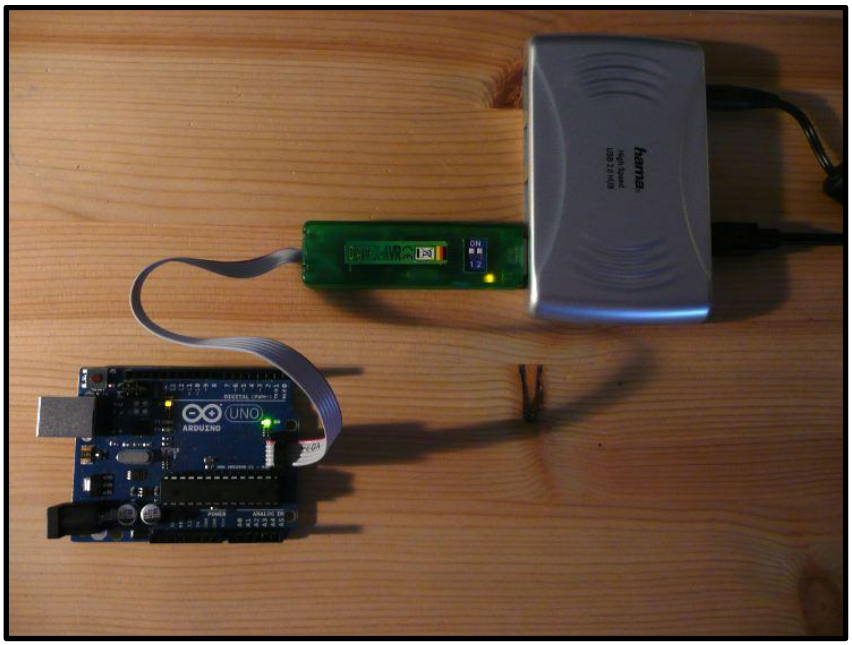
Wir brennen uns einen eigenen Arduino-Bootloader

Liebe Freunde, ich möchte euch in diesem Arduino-*AddOn* zeigen, wie Ihr euch einen eigenen Arduino-Bootloader brennen könnt.

Wer noch nicht ein elektronisches Bauteil in die ewigen Jagdgründe geschickt hat, der möge die Hand heben... Na also, dacht' ich's mir großes Grinsen

Es kann schnell passieren - aus Unachtsamkeit, aus Unwissenheit oder einfach nur, weil man mal Pech hat - und das gute Stück gibt Rauchsignale von sich. Dann ist es gut, wenn man einen sogenannten AVR-Programmer zur Hand hat. Das ist ein Gerät, das an die ISP-Schnittstelle des Arduino-Boards angeschlossen wird und mit dem wir einen brandneuen Mikrocontroller mit einem Bootloader versehen können. Ein Bootloader ist ein kleines Programm, das auf im Speicher des Mikrocontrollers dafür sorgt, dass wir unsere Sketche ganz einfach über die USB-Schnittstelle zum Board übertragen können. Also ganz ohne die Zuhilfenahme eines Programmers.

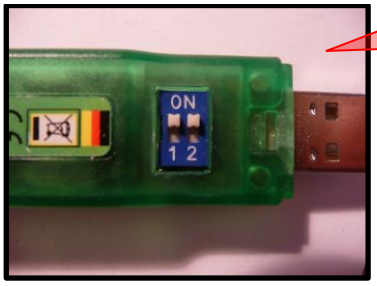
Der Arduino am AVR-Programmer



Ihr seht hier auf dem Bild, dass ich das Arduino-Board mit dem AVR-Programmer über die ISP-Schnittstelle verbunden habe, der seinerseits über meinen USB-Hub am PC hängt.

DIP-Switches des AVR-Programmers

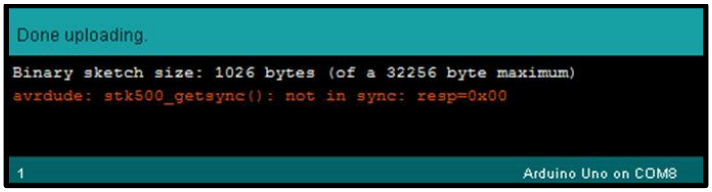
Damit das Arduino-Board mit Spannung versorgt wird, müssen sich die DIP-Switches, die sich am Programmer befinden, in der folgenden Position befinden:



Die Schalter müssen in der gezeigten Position stehen, damit der Arduino mit Spannung versorgt wird.

Wollten wir einen Sketch auf einen brandneuen Mikrocontroller ohne Bootloader schicken, dann geht das 100%ig schief. Hier seht Ihr die Reaktionen der Entwicklungsumgebung auf ein derartiges Vorhaben:

Der Compiler fängt an zu arbeiten...

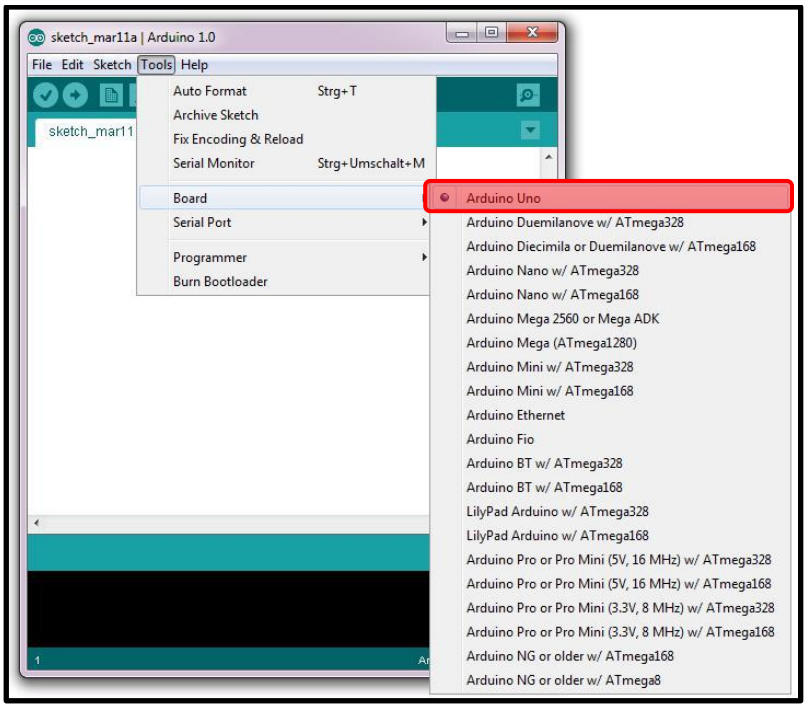


und dann kommt doch tatsächlich die Meldung "Done Uploading", was natürlich absoluter Quatsch ist. Das Nachrichtenfenster zeigt aber etwas in roter Farbe an und das kann nichts Gutes bedeuten. Das Programm *avrdude*, was im Hintergrund für die Übertragung des Sketches zum Mikrocontroller verantwortlich ist, streikt und gibt eine Fehlermeldung aus.

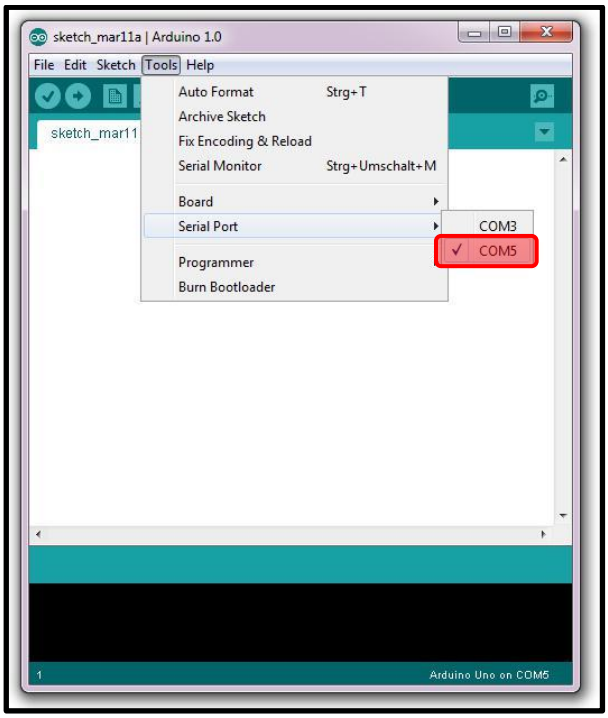
Ok, dann wollen wir mal sehen, was zu tun ist. Hast Du das Arduino-Board wir hier gezeigt mit dem AVR-Programmer verbunden, dann musst Du herausfinden, welchen COM-Port er belegt. Das kannst Du über einen Blick in den Gerätemanager herausfinden.

Ermitteln des COM-Ports des AVR-Programmers

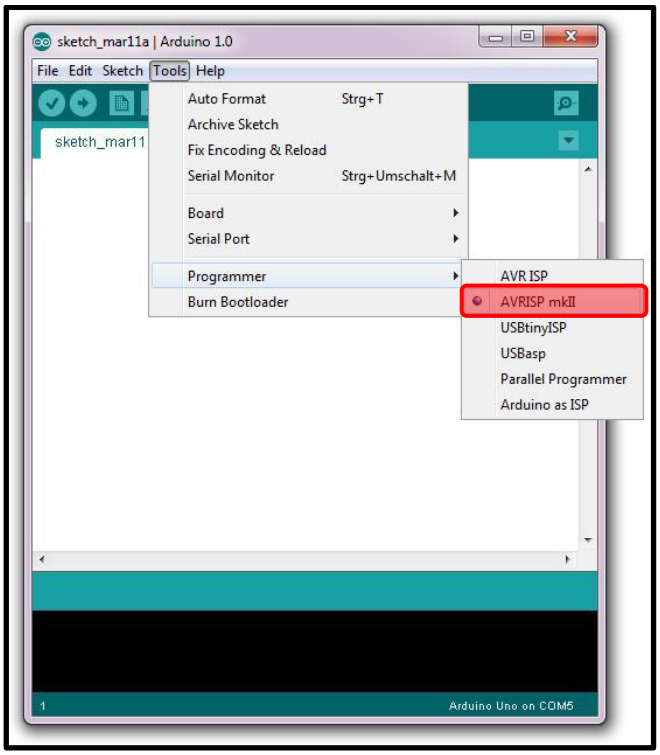
Das war in meinem Fall der Port mit der Nummer 5. Genau diesen Port musst Du Dir merken und beim Start der Arduino-Entwicklungsumgebung bei der Konfiguration eintragen. Zuerst wählst Du das richtige Board aus. In meinem Fall arbeite ich mit dem Arduino-Uno.



Dann wähle ich den besagten COM-Port aus.



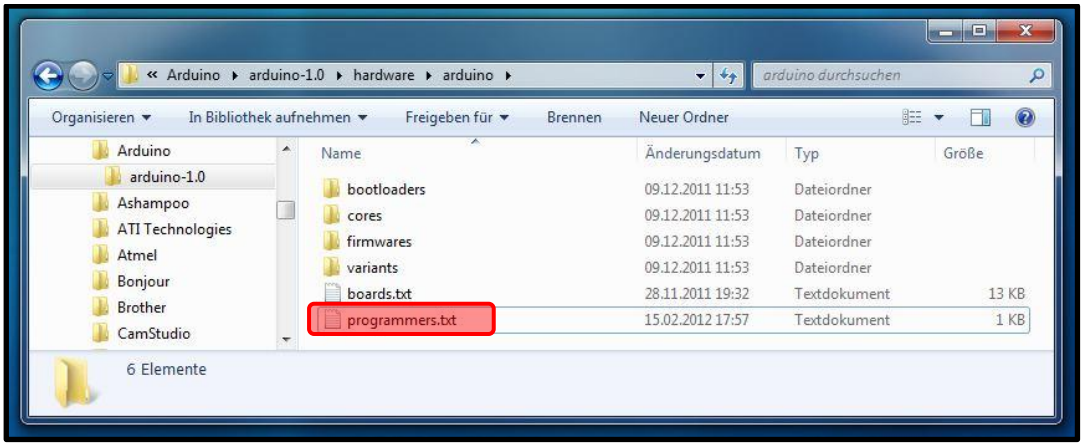
Und im Anschluss den verwendeten AVR-Programmer, der in meinem Fall *AVRISP mkII* kompatibel arbeitet.



Mein Programmer ist übrigens der DIAMEX USB ISP-Programmer für ATMEL AVR, STK500, ATmega, ATtiny, AT90, den es z.B. bei Reichelt für 21,50€ gibt. Also recht günstig, wie ich finde.

AVR-Programmer Anpassungen

Bevor Ihr jedoch den Programmer über die Entwicklungsumgebung ansprechen könnt, muss noch eine Datei angepasst werden, denn andernfalls erkennt sie den Programmer nicht.



Die Datei programmers.txt im gezeigten Verzeichnis beinhaltet einige Einträge, von denen Du die folgenden suchen musst:

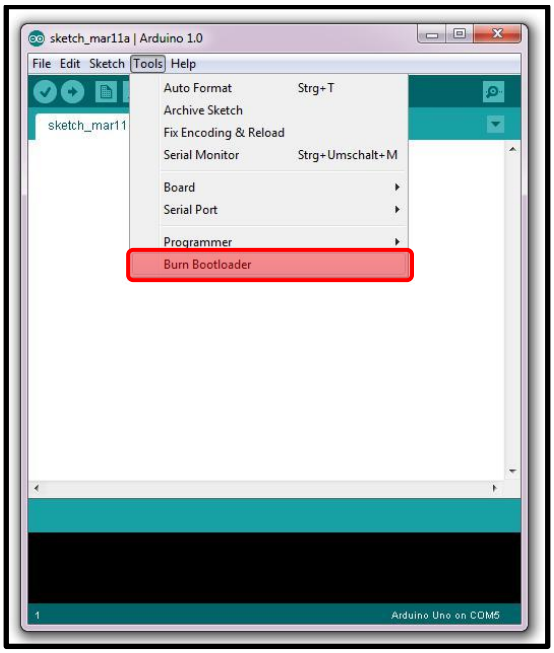
```
avrispmkii.name=AVRISP mkII
avrispmkii.communication=serial
avrispmkii.protocol=stk500v2
```

Von ***usb*** auf ***serial*** ändern.

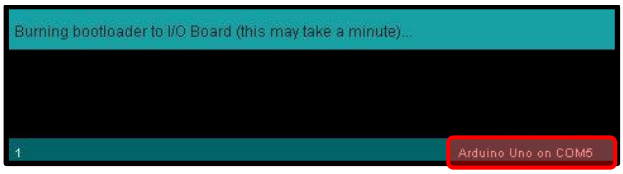
Du muss den Eintrag von ***usb*** auf ***serial*** anpassen und dann die Entwicklungsumgebung starten.

Bootloader brennen

Jetzt kannst Du den Menüpunkt *Burn Bootloader* wählen.



Danach kommen die beiden Meldungen - so Gott will - und Du hast einen neuen Bootloader auf einen frischen Mikrocontroller übertragen.



COM-Port des AVR-Programmers.

und endlich...

Ist also eigentlich ganz einfach, nicht wahr!? Ich wünsche euch viel Spaß und viel Glück dabei...

Falls Ihr noch Fragen habt, dann schreibt einfach.

Bis dann