FAQ zum ATMega168P

1. Welche Software und Version wird benötigt um die Firmware aufzuspielen und wo kann man sie runterladen?

Zum Download der Software auf den Kontroller wird das AVR Studio benötigt. Dieses kann kostenlos auf der Atmel Seite runtergeladen werden.

2. Welche Hardware wird benötigt um den ATMega168P zu programmieren?

Zum programmieren ist ein ISP Adapter notwendig. Mit diesem Adapter kann der Kontroller programmiert werden, wenn er in der aktuellen Schaltung eingebaut ist (In-Service-Programming). Hierfür kann der Atmel AVR ISP oder kompatibel z.Bsp. AVR ISP MKII benutzt werden, den es auch bei Reichelt gibt.

3. Wie werden die 6 poligen Kontakte mit der Controller Platine verbunden?

ISP einfach auf das Kontrollerboard stecken.

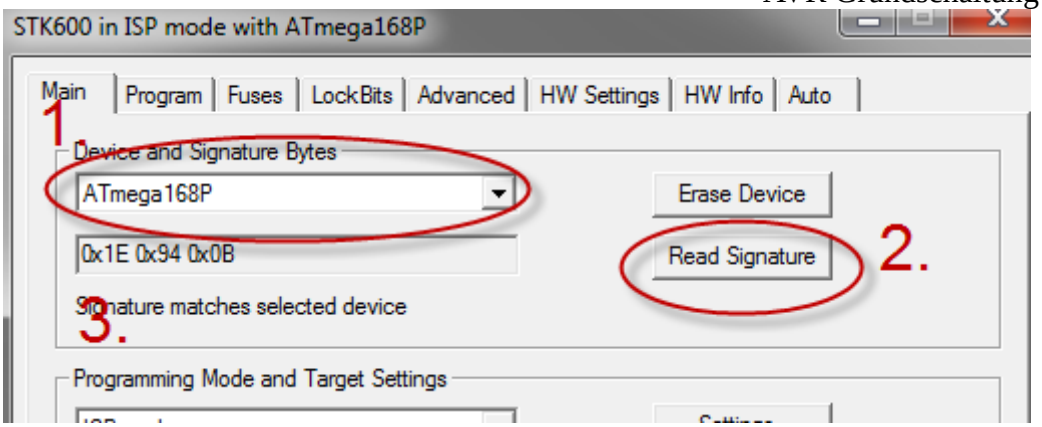
4. Kann der Chip in der Controller Platine sein wenn man ihn programmiert?

Er muss auf dem auf der Platine stecken und die Platine muss mit dem Bus verbunden sein. Nur so liegt die nötige Beschaltung/Spannung am Kontroller an.

5. Einige Hardcopys der einzelnen Schritte wie man die Software anwenden muß um die Firmware auf den 168P aufzuspielen?
(anhand der Software AVR Studio)

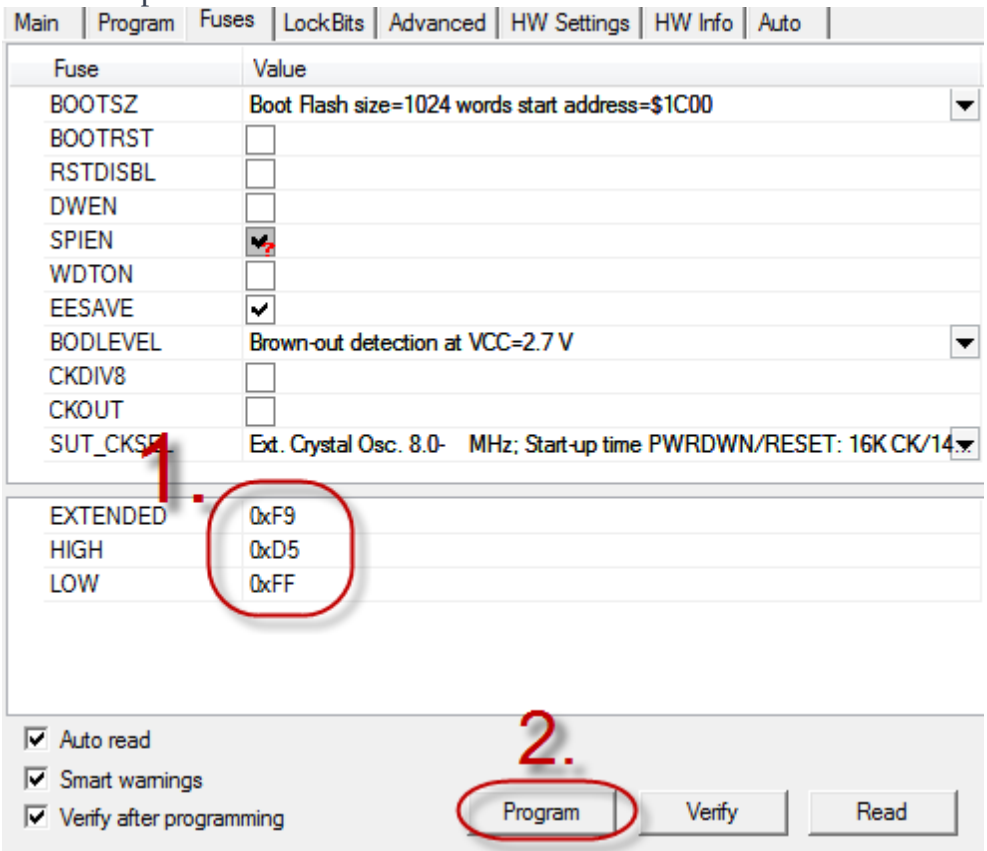
Zuerst muss man sich mit der Programmier Hardware verbinden:

Bei Select AVR Programmer das richtige Gerät auswählen. Ich habe hier ein STK600, aber es sollte bei allen anderen Gerät ähnlich aussehen.
Wenn alles geklappt hat, sollte das folgende Fenster auftauchen:



Dort muss zuerst das richtige Device ausgewählt werden (1.).
Dann auf Read Signature (2.) klicken, anschließend muss bei 3. eine Signatur auftauchen, falls das nicht klappt, ist entweder der Kontroller nicht am Bus angeschlossen, die Verbindung zwischen dem Programmer und dem Kontroller passt nicht oder die Platine ist nicht korrekt bestückt.

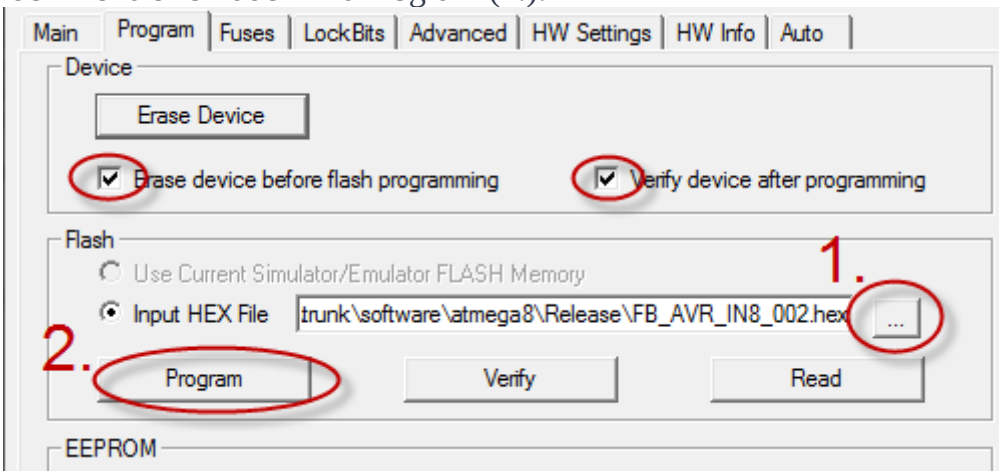
Als nächstes programmieren wir die Fuses, dafür klickt ihr auf den Tab Fuses und gebt unten bei EXTENDED, HIGH und LOW die passenden Hexwerte ein, das AVRStudio setzt die Werte oben dann entsprechend:



Dann auf Program klicken. Unten im Statusfenster sollte eine Meldung kommen, dass alles erfolgreich geschrieben worden ist:

```
Entering programming mode.. OK!
Writing fuses address 0 to 2.. 0xFF, 0xD5, 0xF9 .. OK!
Reading fuses address 0 to 2.. 0xFF, 0xD5, 0xF9 .. OK!
Fuse bits verification.. OK
Leaving programming mode.. OK!
```

Jetzt laden wir das Programm auf den Kontroller, hierfür auf den Tab Program gehen. Stellt sicher das die beiden Hacken oben aktiv sind, wählt das .hex file aus (1.) und dann das ganzen auf den Kontroller laden mit Program (2.).



Es sollte wieder die folgende Meldung zu sehen sein:

```
Erasing device.. OK!
Programming FLASH ..    OK!
Reading FLASH ..    OK!
FLASH contents is equal to file.. OK
Leaving programming mode.. OK!
```

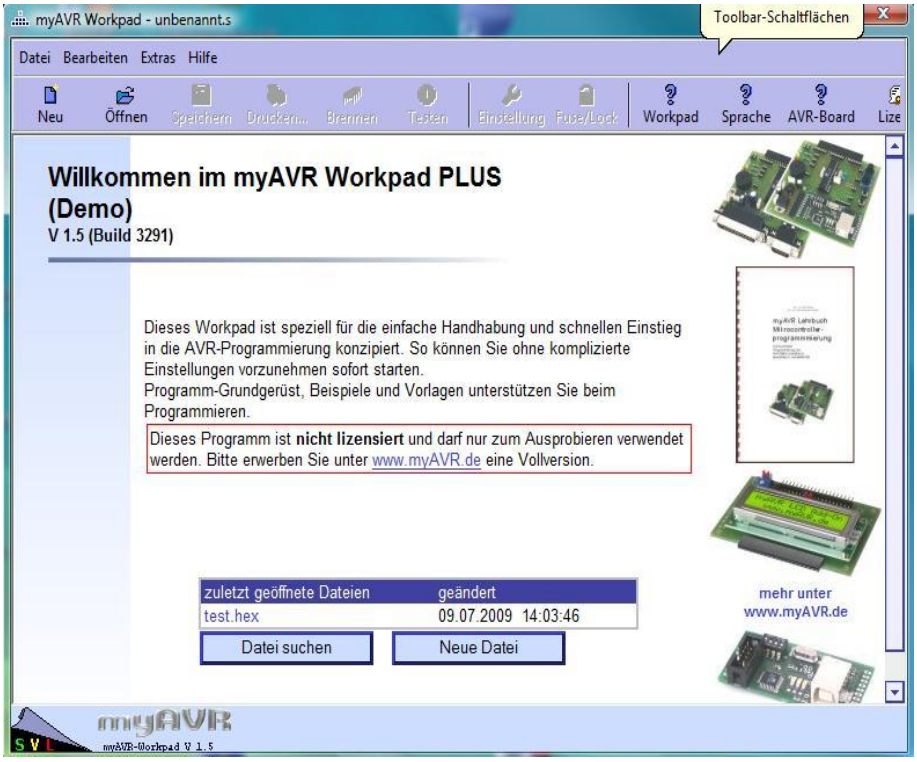
Damit ist der Kontroller komplett mir der Software bestückt und kann ganz normal aus der ETS parametrisiert werden.

6. Einige Hardcopys der einzelnen Schritte wie man die Software anwenden muß um die Firmware auf den 168P aufzuspielen? (anhand der Software myAVR Workpad)

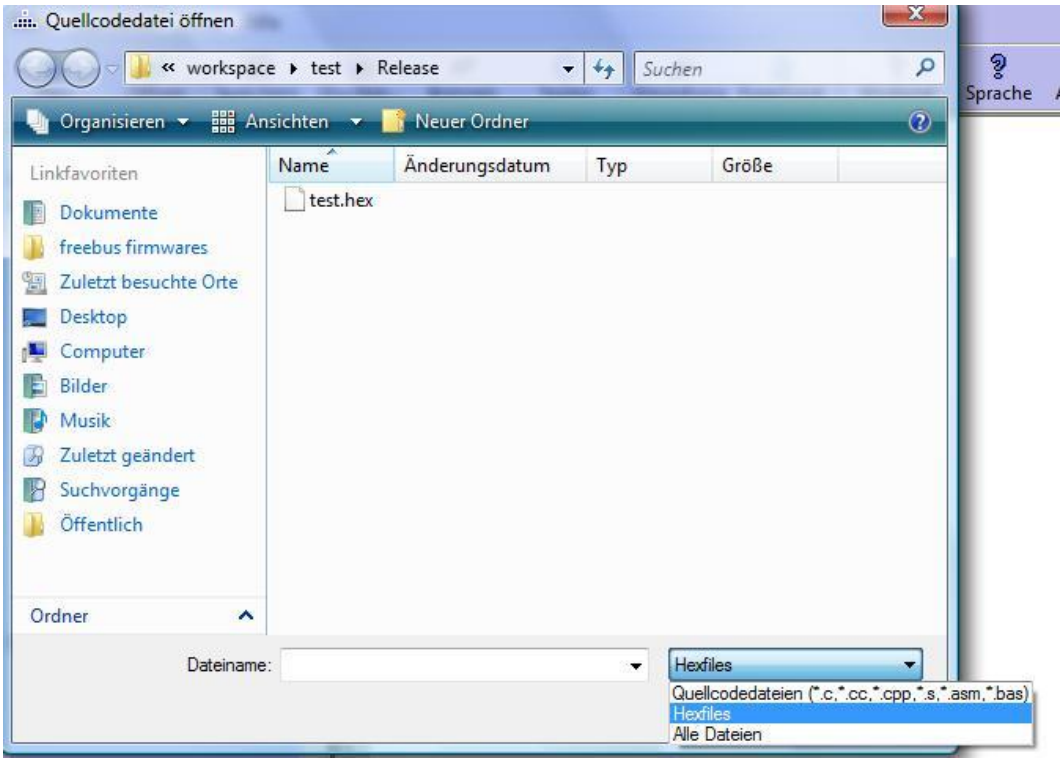
Die Software herunterladen und installieren.

Falls für den Programmer zusätzliche Treiber installiert werden müssen dann diese nach Anleitung installieren.

1. myAVR Workpad Plus starten, es erscheint folgendes Hauptfenster:

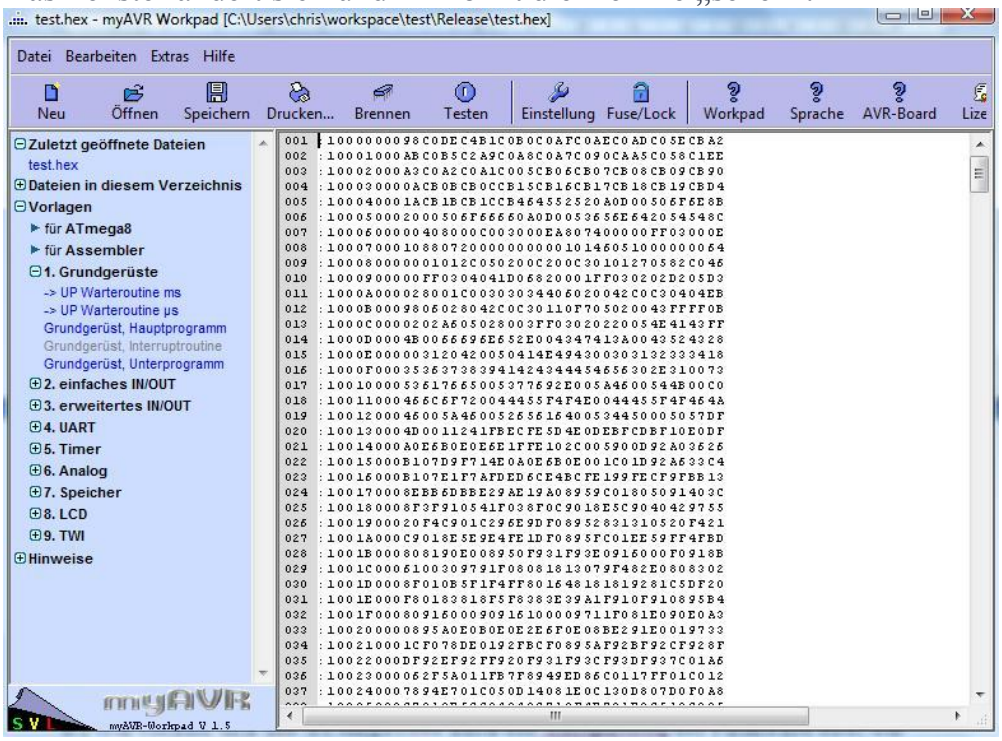


2. Auf die Toolbar Schaltfläche „Öffnen“ klicken. Es öffnet sich ein öffnen Dialog:



Hier als Dateityp „Hexfiles“ auswählen und in das Verzeichnis wechseln wo ihr die Hexfiles von der freebus Homepage gespeichert habt. Dann das Hexfile auswählen das auf den Kontroller geflasht werden soll und auf „öffnen“ klicken.

Das Fenster ändert sich und ihr könnt die Hexfile „sehen“:

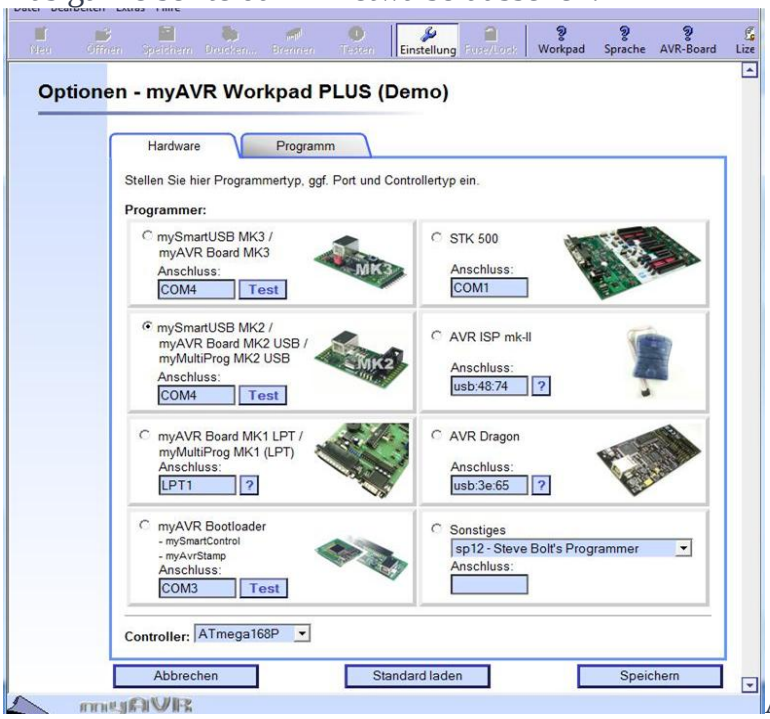


3. Jetzt auf die Schaltfläche „Einstellungen“ klicken. Es öffnet sich das Einstellungsfenster, hier müsst ihr euern Programmer auswählen:

Einfach auf den Radiobutton für euern Programmer klicken, abhängig davon kann man auch einen Kommunikationstest durchführen(auf Test klicken).

4. Wenn ihr das gemacht habt wählt ihr noch den verwendeten Kontroller, hier ATMEGA 168p.

Das ganze sollte dann in etwa so aussehen:

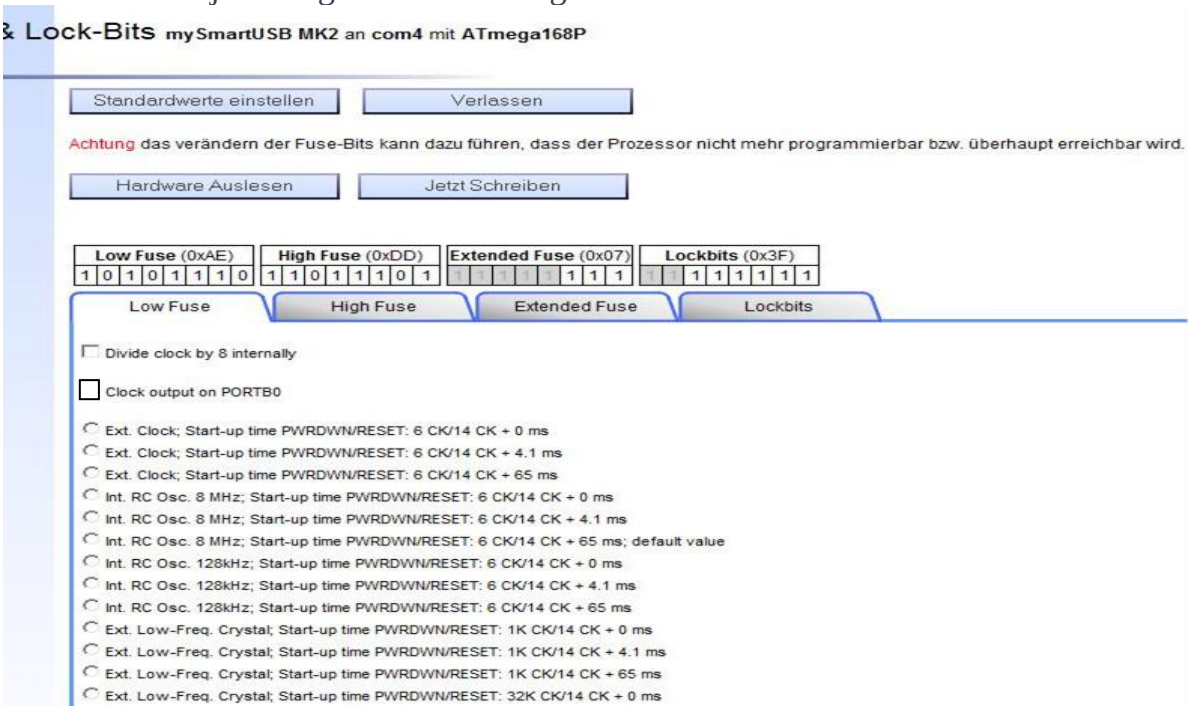


Auf „Speichern“ klicken!

5. Abhängig von der Applikation ist es notwendig am Chip bestimmte Einstellungen vorzunehmen d.h. sog. Fuses setzen(welche steht in der readme.txt welche ihr mit dem Code heruntergeladen habt)

Dazu klickt ihr auf „Fuse/Lock“

Hier müsst ihr jetzt folgende Einstellungen für den 8fach out einstellen:



Low Fuse:

"Ext. Crystal Osc.; Frequency 8.0- MHz; Start-up time PWRDWN/RESET: 16K CK/14 CK + 65 ms" auswählen

Es muss als Low Fuse 0xFF da stehen!

High Fuse:

„Serial program downloading (SPI) enabled" auswählen

"Brown-out detection level at VCC=2.7 V" auswählen

"Preserve EEPROM memory through the Chip Erase cycle" auswählen

Es muss als High Fuse 0xD5 da stehen !

Extended Fuse:

„Boot Flash section size=128 words Boot start address=0x1F80" anklicken

Jetzt auf "jetzt Schreiben" klicken und die Fuses werden im Controller gesetzt.

Als nächstes auf „Verlassen" klicken.

6. Jetzt in der Toolbar-Leiste einfach auf „Brennen" klicken und der Controller wird mit der Firmware geflasht.

Das wars nun einfach den Programmer vom Board trennen.

1. Wie verhält sich der ATMega168P nach der Aufspielung der Firmware bzw. wie

man testen kann ob das Aufspielen der Firmware erfolgreich war?
Das aufspielen der Firmware war erfolgreich, wenn die Programmiersoftware ein erfolgreich sagt. Eine weitere Möglichkeit um das zu testen ist es den Programmiertaster zu drücken, wenn das LED an bleibt wenn der Taster wieder losgelassen wird, läuft die Firmware.

2. Wird noch etwas anderes benötigt um jetzt die Controller Platine mit der entsprechenden z.B. 8 fach Aktor mit der ETS zu verwenden?
Nein, Kontroller mit dem AppBoard verbinden und alles in ein Gehäuse einbauen.