

## Der Aufbau der .dev Dateien

Die erste Information zum AVR-8 steckt schon im Dateinamen.  
Er setzt sich aus den folgenden drei Hexadezimalzahlen zusammen.

1. Der Manufacturecode (Signaturbyte 1)
2. Der Memorycode (Signaturbyte 2)
3. Der Devicecountercode (Signaturbyte 3)

Beim ATtiny13 sind das die Codes 1E, 90, 07 diese bilden dann den Dateinamen „1E9007.dev“  
Damit kann über ein verbundenen AVR-8 direkt die Device-Datei zugeordnet werden.

## Der Kopf

Der erste Teil der .dev Datei enthält grundlegende Informationen zu dem AVR-8.

```
Name = ATtiny13(A)
FLASH = 512
EEPROM = 64
SRAM = 64
FLASHblock = 16
EEPROMblock = 4
Description =
Calib0 = 9.6 MHz Base
Calib1 =
Calib2 =
Calib3 =
```

Name = Hier wird der AVR-8 Name zugewiesen, so wie er im AVR-Studio5 verwendet wird.  
Bei manchen AVR-8 gibt es unterschiedliche Bezeichnungen mit gleicher Signatur z.B. ATtiny13 und ATtiny13A so ist der Unterscheider in Klammern zusetzen. In diesem Beispiel wäre dann ATtiny13(A) einzutragen.

FLASH = Größe des FLASH-Speichers als die Anzahl der 16-Bit Worte.

EEPROM = Größe des EEPROM-Speichers als die Anzahl der Bytes.

SRAM = Größe des SRAM-Speichers als die Anzahl der Bytes.

FLASHblock = Größe einer FLASH-Page als die Anzahl der 16-Bit Worte.  
(Muss 0 gesetzt werden wenn im ISP das Page-Schreiben nicht unterstützt wird.)

EEPROMblock = Größe einer EEPROM-Page als die Anzahl der Bytes.  
(Muss 0 gesetzt werden wenn im ISP das Page-Schreiben nicht unterstützt wird.)

Description = Eine beliebige Beschreibung zum AVR-8.

Calib0 bis Calib3 wird eine Bezeichnung zu den entsprechenden Kalibrierwerten der AVR-8 zugewiesen. Für Werte die es nicht gibt, wird der Eintrag leer gelassen.

## Die Lock-Bits und Fuses

Im nächsten Teil wird den Lock- und Fuse-Bits ihre Bezeichnung zugeordnet.

LBit0 = LB1	FHighBit0 = RSTDISBL
LBit1 = LB2	FHighBit1 = BODLEVEL0
LBit2 =	FHighBit2 = BODLEVEL1
LBit3 =	FHighBit3 = DWEN
LBit4 =	FHighBit4 = SELFPRGEN
LBit5 =	FHighBit5 =
LBit6 =	FHighBit6 =
LBit7 =	FHighBit7 =
FLowBit0 = CKSEL0	FExtBit0 =
FLowBit1 = CKSEL1	FExtBit1 =
FLowBit2 = SUT0	FExtBit2 =
FLowBit3 = SUT1	FExtBit3 =
FLowBit4 = CKDIV8	FExtBit4 =
FLowBit5 = WDTON	FExtBit5 =
FLowBit6 = EESAVE	FExtBit6 =
FLowBit7 = SPIEN	FExtBit7 =

Bits die unbelegt sind erhalten eine Leereintrag. SPIEN und RSTDISBL sind zugleich auch Schlüsselworte die dem AVR FTDI ISP-Prog dazu veranlassen, diese Flags nicht änderbar zu machen.

## Das User-Friendly-Fusing

Dieser Teil ist optional. Wir er nicht angegeben so muss aber hinter dem `FExtBit7` = Eintag auf jeden Fall ein Zeilenumbruch erfolgen.

Belegt werden können 6 Dropdown-Felder `UFF1_` bis `UFF6_`. Wird ein Feld nicht belegt, so wird es vom AVR FTDI ISP-Prog ausgeblendet. Die Einträge sind dann fortlaufend nummeriert. Wird die Nummerierung unterbrochen, so endet der AVR FTDI ISP-Prog mit dem Einlesen für dieses Dropdown-Feld.

<code>UFF1_0</code> = Nicht Gelockt	#	xxxxxx11	xxxxxxxx	xxxxxxxx	xxxxxxxx
<code>UFF1_1</code> = Schreib-Lock	#	xxxxxx10	xxxxxxxx	xxxxxxxx	xxxxxxxx
<code>UFF1_2</code> = Lese-Schreib-Lock	#	xxxxxx00	xxxxxxxx	xxxxxxxx	xxxxxxxx
<code>UFF2_0</code> = External Clock	#	xxxxxxxx	xxx1xx00	xxxxxxxx	xxxxxxxx
<code>UFF2_1</code> = External Clock / 8	#	xxxxxxxx	xxx0xx00	xxxxxxxx	xxxxxxxx
<code>UFF2_2</code> = Internal RC 9,6 MHz	#	xxxxxxxx	xxx1xx10	xxxxxxxx	xxxxxxxx
<code>UFF2_3</code> = Internal RC 4,8 MHz	#	xxxxxxxx	xxx1xx01	xxxxxxxx	xxxxxxxx
<code>UFF2_4</code> = Internal RC 1,2 MHz	#	xxxxxxxx	xxx0xx10	xxxxxxxx	xxxxxxxx
<code>UFF2_5</code> = Internal RC 0,6 MHz	#	xxxxxxxx	xxx0xx01	xxxxxxxx	xxxxxxxx
<code>UFF2_6</code> = Internal RC 128 kHz	#	xxxxxxxx	xxx1xx11	xxxxxxxx	xxxxxxxx
<code>UFF2_7</code> = Internal RC 16 kHz	#	xxxxxxxx	xxx0xx11	xxxxxxxx	xxxxxxxx
<code>UFF4_0</code> = StartUp 14 CK BOD 1,8V	#	xxxxxxxx	xxxx00xx	xxxxx10x	xxxxxxxx
<code>UFF4_1</code> = StartUp 14 CK BOD 2,7V	#	xxxxxxxx	xxxx00xx	xxxxx01x	xxxxxxxx
<code>UFF4_2</code> = StartUp 14 CK BOD 4,3V	#	xxxxxxxx	xxxx00xx	xxxxx00x	xxxxxxxx
<code>UFF4_3</code> = StartUp 14 CK + 4 ms	#	xxxxxxxx	xxxx01xx	xxxxxxxxx	xxxxxxxx
<code>UFF4_4</code> = StartUp 14 CK + 65 ms	#	xxxxxxxx	xxxx10xx	xxxxxxxxx	xxxxxxxx
<code>UFF5_0</code> = BODLEVEL OFF Fast Start	#	xxxxxxxx	xxxx01xx	xxxxx11x	xxxxxxxx
<code>UFF5_1</code> = BODLEVEL OFF Slow Start	#	xxxxxxxx	xxxx10xx	xxxxx11x	xxxxxxxx
<code>UFF5_2</code> = BODLEVEL 1,8V	#	xxxxxxxx	xxxxxxxx	xxxxx10x	xxxxxxxx
<code>UFF5_3</code> = BODLEVEL 2,7V	#	xxxxxxxx	xxxxxxxx	xxxxx01x	xxxxxxxx
<code>UFF5_4</code> = BODLEVEL 4,3V	#	xxxxxxxx	xxxxxxxx	xxxxx00x	xxxxxxxx

Nach dem Schlüsselwort steht der Eintrag für das Dropdown-Feld. Nach dem # Zeichen folgen die Bitmasken. Es darf zum Trennen nur mit Leerzeichen gearbeitet werden, Tabs sind nicht erlaubt ! Es muss wie im Beispiel mindestens je ein Leerzeichen zur Trennung vorhanden sein. Die ersten 8 Zeichen stehen für die Lock-Bits. Links steht immer Bit7 und rechts immer Bit0. Ein kleines - und nur ein kleines - „x“ kennzeichnet eine Fuse, die für diesen Eintag nicht relevant ist. Eine „0“ eine gesetzte Fuse und eine „1“ eine gelöschte Fuse. Der nächste Block bildet die `FLowBits` ab, gefolgt von den `FHighBits` und den `FExtBits`.

Es kann mit mehrdeutigen Kombinationen gearbeitet werden. Angezeigt wir aber die, welche die höchsten Nummerierung hat, und zum Muster passt. Das Bit-Fusing und das User-Friendly-Fusing beeinflussen sich gegenseitig. Deshalb kann eine nicht durchdachte Eintragung den AVR FTDI ISP-Prog in eine Endlosschleife bringen.

**Da die .dev Dateien zum System gehören, werden sie vom AVR FTDI ISP-Prog nicht auf Plausibilität geprüft. Fehlerhafte .dev können den AVR FTDI ISP-Prog zum Absturz bringen und AVR-8 unbrauchbar machen.**

Nach dem letzten Eintag muss ein Zeilenumbruch erfolgen. Am Ende der Datei können noch beliebige Bemerkungen zugefügt werden.