

## **AN011: AVR-Programmierung**

### **Inhalt**

- Einleitung
- Hardware
- Installation
- Setup
- Flashen von AVR
- Setzen der fuse bits
- board.hex

### **Einleitung**

Diese Application Note beschreibt die Programmierung von AVR's mit Hilfe des ATMEL Studios. Sie ist auf den DMX-Transceiver zugeschnitten, sollte sich aber auch allgemein anwenden lassen.

## Hardware

Als USB-Programmer ist der AVRISPmkII von ATMEL (jetzt Microchip) empfehlenswert. Es gibt auch zahlreiche andere ISP-Programmer für AVR Mikrocontroller, die ebenfalls kompatibel mit dem ATMEL Studio sein sollen. Bitte beachten Sie die Installationsanleitung des jeweiligen Herstellers.



Abbildung 1: ISP-Programmer (AVRISPmkII)

Ein DMX-Transceiver ab der Rev. 3.2 kann direkt angeschlossen werden.

Der Anschluss an einen DMX-Transceiver der Rev. 3.01 und kleiner erfolgt über ein Adapterkabel. Die Versorgungsspannung VCC wurde nicht auf der ISP-Steckverbindung herausgeführt und muss daher separat von der Platine bezogen werden.

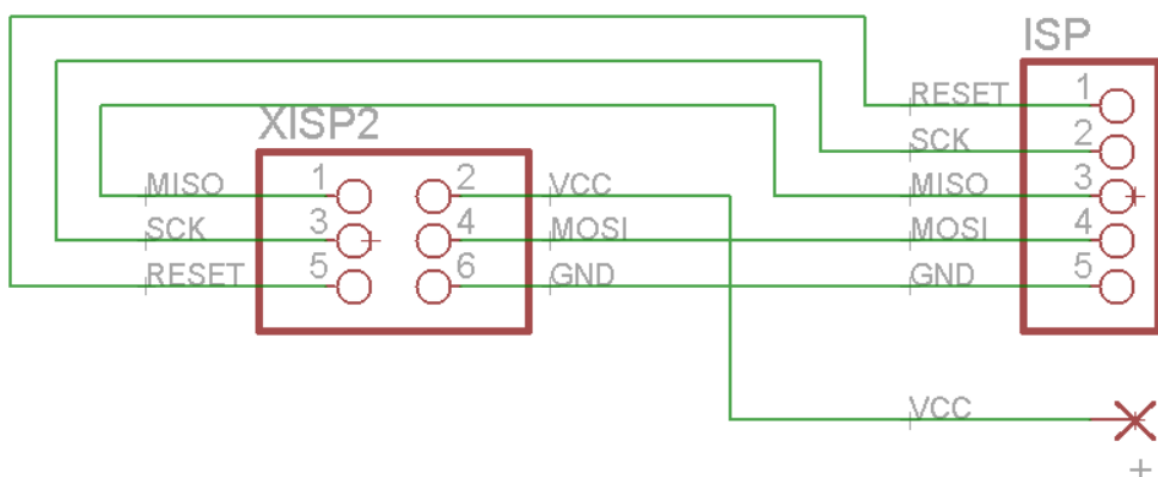


Abbildung 2: Adapterkabel für DMX-Transceiver Rev. 3.01

## Installation

Laden Sie zunächst eine aktuelle Version des ATMEL Studios herunter und folgen Sie den Installationsanweisungen:

<https://www.microchip.com/mplab/avr-support/avr-and-sam-downloads-archive>

## Setup

Verbinden Sie zunächst den ISP-Programmer mit dem Computer. Die LEDs des AVRISPMkII sollten wie auf Abbildung 1 dargestellt leuchten – andernfalls war die Installation des Treibers nicht erfolgreich.

Verbinden Sie anschließend den ISP-Programmer – gegebenenfalls über das Adapterkabel - mit dem DMX-Transceiver und verbinden den DMX-Transceiver mit einer Betriebsspannung zwischen 9V und 12V. Beide LEDs des AVRISPMkII sollten nun grün leuchten.

Leuchtet die eine LED weiterhin rot, wurde VCC nicht erkannt: Bitte prüfen Sie, ob 5V zwischen + und GND anliegen (vgl. Abb. 3):

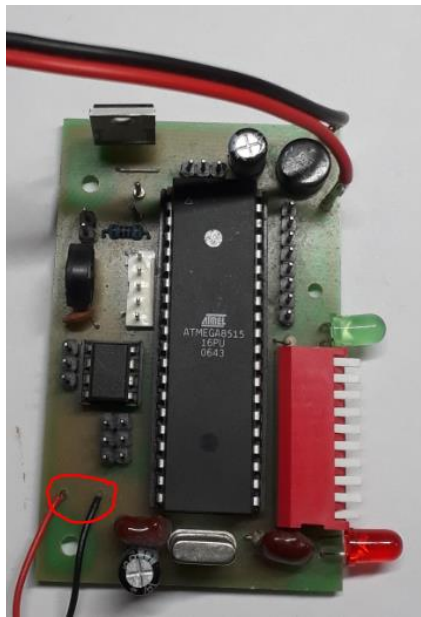


Abbildung 3: Prüfen von VCC

Blinkt die eine LED orange, hat der AVRISPMkII einen Fehler in der Verbindung festgestellt. Bitte überprüfen Sie ggf. nochmals sorgfältig das Adapterkabel.

Eventuell wurde der Programmer auch verkehrt herum aufgesteckt.

Starten Sie nun das ATMEL Studio:

Möchten Sie eine bestehende Firmware auf den AVR übertragen, wählen Sie direkt „Device Programming“ (vgl. Abb. 4).

Möchten Sie eine Firmware anpassen, wählen Sie „Open Project...“.



Abbildung 4: Device Programming

Wählen Sie als Tool Ihren ISP-Programmer, den ATmega8515 als Device und ISP als Interface aus und bestätigen durch Klick auf Apply:



Abbildung 5: Settings

Da bei den meisten AVR ab Werk eine Betriebsfrequenz von 1 MHz eingestellt ist, sollte die ISP Clock zunächst mit 125 kHz arbeiten:

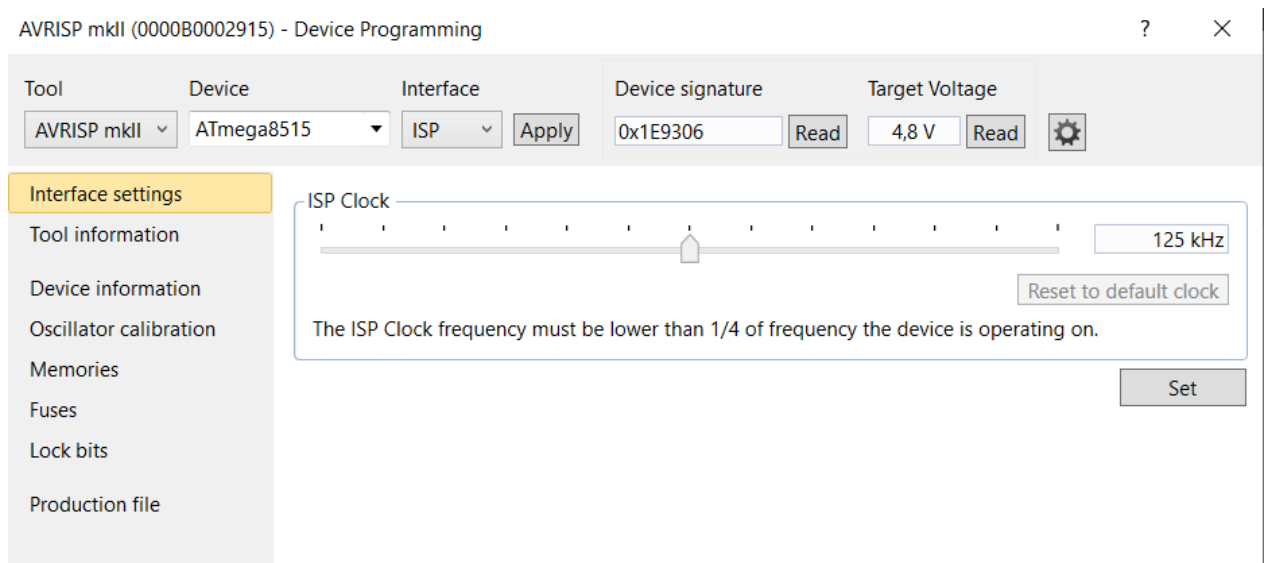


Abbildung 6: ISP Clock

Prüfen Sie abschließend die Verbindung durch Auslesen der Device signature. Die Target Voltage sollte beim DMX-Transceiver ca. 5,0V betragen.

## Flashen von AVR

1. Gehen Sie links auf „Memories“.
2. Laden Sie das gewünschte Projekt von Henne's Sites und entpacken das Archiv.
3. Öffnen Sie für den Flash-Speicher die gewünschte .hex-Datei aus dem Debug-Ordner des Projektes.
4. Flashen Sie den AVR durch Klick auf „Program“.

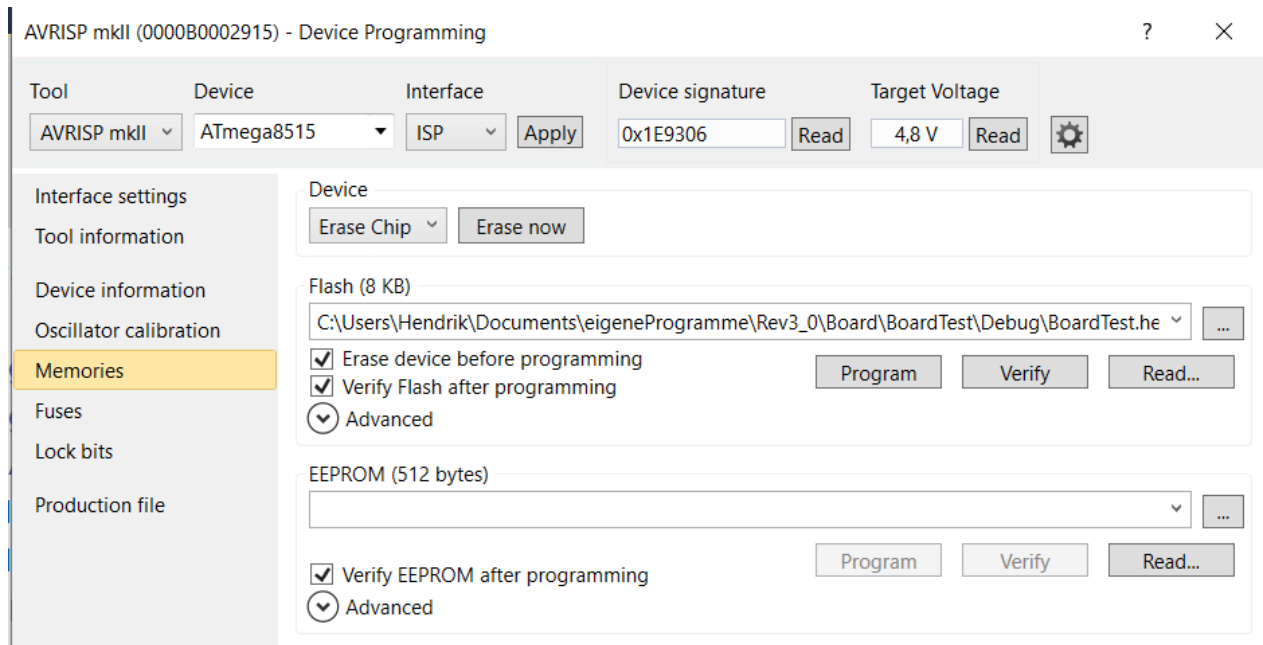


Abbildung 7: Flashen der Firmware

### Setzen der fuse bits

Ab Werk ist der interne RC-Oszillator (1MHz) als Taktquelle ausgewählt. Da dies für DMX zu langsam ist, muss der Quarz (8MHz) ausgewählt werden. Dies geschieht über folgende Anpassung der fuse bits:

1. Gehen Sie links auf „Fuses“
2. Lesen Sie die aktuellen Einstellungen aus.
3. Verändern Sie die bits gemäß Abbildung 8 (0x9D; 0x3F).
4. Schreiben Sie die fuse bits zurück.

AVRISP mkII (0000B0002915) - Device Programming ? X

Tool	Device	Interface	Device signature	Target Voltage
AVRISP mkII	ATmega8515	ISP	0x1E9306	4,9 V

	Fuse Name	Value
<input checked="" type="checkbox"/>	HIGH.S8515C	<input type="checkbox"/>
<input checked="" type="checkbox"/>	HIGH.WDTON	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	HIGH.SPIEN	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	HIGH.EESAVE	<input type="checkbox"/>
<input checked="" type="checkbox"/>	HIGH.BOOTSZ	Boot Flash size=256 words Boot address=\$0F00
<input checked="" type="checkbox"/>	HIGH.BOOTRST	<input type="checkbox"/>
<input checked="" type="checkbox"/>	HIGH.CKOPT	<input type="checkbox"/>
<input checked="" type="checkbox"/>	LOW.BODLEVEL	Brown-out detection at VCC=4.0 V
<input checked="" type="checkbox"/>	LOW.BODEN	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	LOW.SUT_CKSEL	Ext. Crystal/Resonator High Freq.; Start-up time: 16K CK + 64 ms

Fuse Register	Value
HIGH	0x9D
LOW	0x3F

Auto read Copy to clipboard  
 Verify after programming Program Verify Read

Abbildung 8: fuse bits (Settings für externen Quarz, Brown-Out Überwachung und Watchdog)

## **board.hex**

Hierbei handelt es sich um ein Testfile für den DMX-Transceiver.

1. Laden Sie das Projekt herunter.
2. Übertragen Sie die hex-Datei in den Flash-Speicher Ihres AVR.
3. Selektieren Sie durch Anpassen der fuse bits den externen Quarz.

Testfunktionen:

Blinken der roten LED zeigt an, dass das Testfile in den Flash geladen wurde.

### **Alle DIPs off:**

Leuchten der grünen LED zeigt an, dass sie korrekt eingebaut wurde.

### **Alle DIPs an:**

Leuchten der grünen LED zeigt an, dass alle DIP-Schalter korrekt verlötet sind.

### **Nur DIP1 an:**

Leuchten der grünen LED zeigt an, dass die fuse bits angepasst wurden.

### **Nur DIP2 an:**

Leuchtet die grüne LED, liegt irgendein Signal am DMX-Eingang an.

Es mag sein, dass D+ und D- verpolt sind.

### **Nur DIP3 an:**

Leuchtet die grüne LED, wurde ein korrektes DMX-Signal erkannt.

### **Nur DIP4 an:**

Leuchtet die grüne LED, funktioniert die angeschlossene ZC-Detection.