

MIKROKONTROLLER & I²C BUS



by AS

www.boxtec.ch

playground.boxtec.ch/doku.php/tutorial

Attiny 841 - ein IC und
5 verschiedene Module
Teil 4 - AVR Studio

I²C Bus und der
Attiny 841



Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung-NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



Sicherheitshinweise

Lesen Sie diese *Gebrauchsanleitung*, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die Gewährleistung / Garantie. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfwerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehler muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

Attiny 841 – AVR Studio

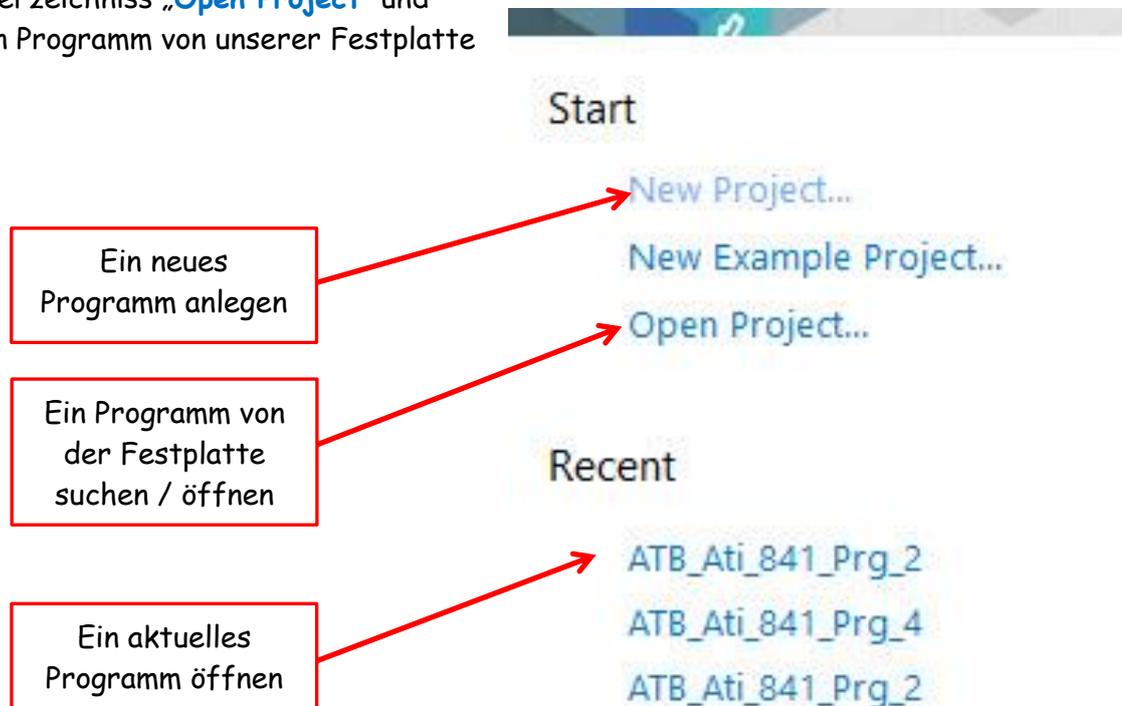
In einem anderen Tutorial habe ich die erste Inbetriebnahme für einen Atmega 1284p (Board1) beschrieben. Im Grund wird der Attiny 841 fast genau so programmiert bzw. in Betrieb genommen. Ein Grundlegender Unterschied besteht bei den Fuse-Bits. In Abhängigkeit der verwendeten Bauteile erfolgt eine Einstellung mit Quarz und ohne Quarz. In den beiden ersten Teilen habe ich bereits 5 verschiedene Module mit der Hardware und die erste Inbetriebnahme vorgestellt. Da der Attiny 841 sehr komplex ist, werde ich in den folgenden Teilen spezielle Einstellungen und Funktionen vorstellen.

Was muss ich machen um meinen Attiny 841 zum ersten Mal in Betrieb zu nehmen?

- Atmel Studio 7 starten
- Ein vorhandenes Programm angeben oder ein neues Programm anlegen
- Name und Speicherort angeben
- Prozessor auswählen
- Programm eingeben
- Programm kompilieren
- Programm an den Attiny 841 übertragen

Als nächste wollen wir uns den genauen Ablauf ansehen und Schritt für Schritt durchführen. Die Installation des Atmel Studio 7 habe ich bereits in einem anderen Beitrag beschrieben.

Wir starten als erstes unser Atmel Studio 7, wählen ein Projekt aus der Liste aus oder öffnen das Verzeichniss „**Open Project**“ und wählen ein Programm von unserer Festplatte aus.



Wir können auch ein neues Programm anlegen, was wir in diesem Tutorial machen werden. In unserem Beispiel wählen wir „**New Project**“.

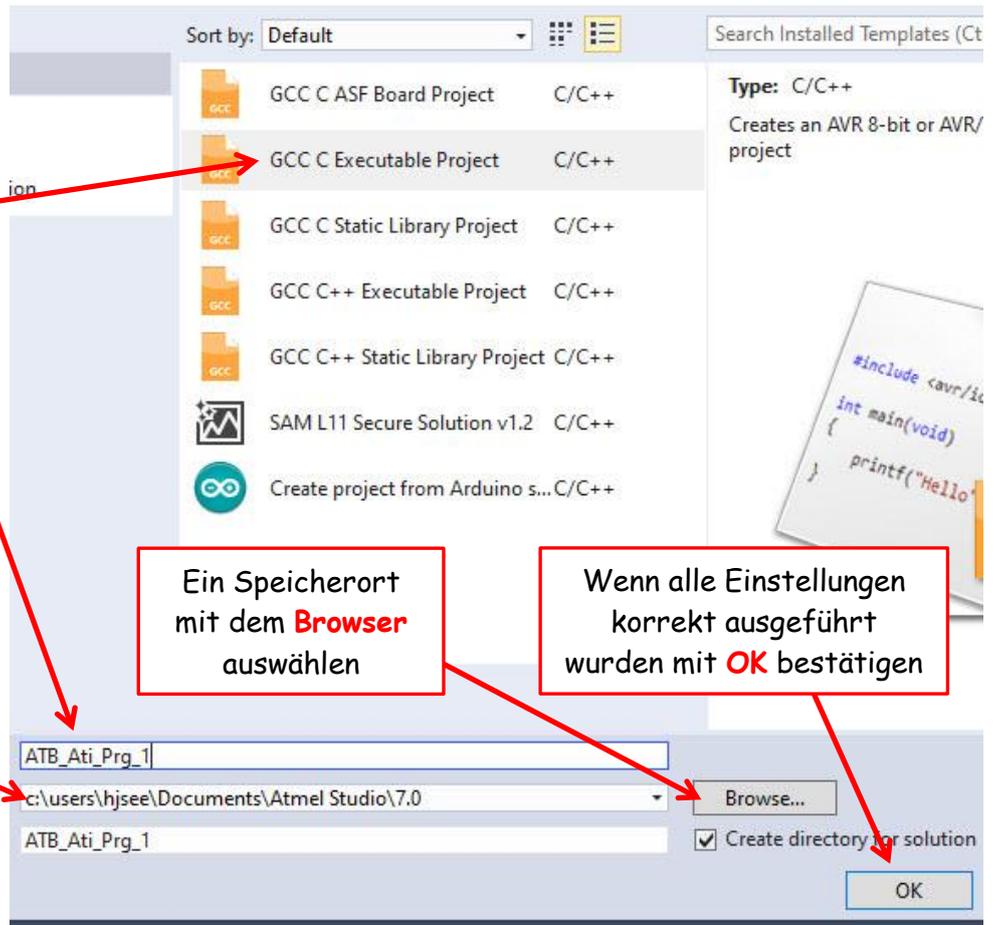
Für die Programmierung verwende ich einen Atmel-ICE. Die Belegung Der ISP Stecker bitte einem anderen Tutorial entnehmen. Die Programmierung erfolgt in C.

Im nächsten Bild wir wählen „GCC C Executable Project C/C++“ aus.

Auswahl „GCC C Executable Project“

Angabe eines Programmnamens

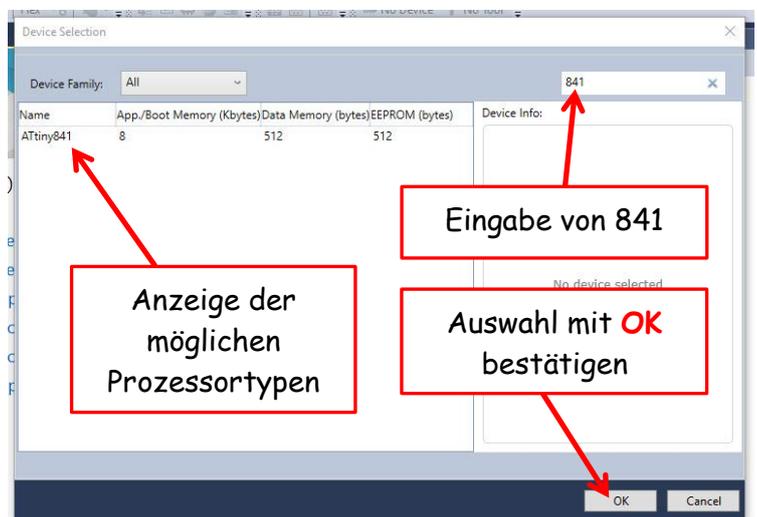
Auswahl eines Speicherorts



Danach muss der Name des Programmes angegeben werden, Speicherort angeben oder mit dem **Browser** einen Ort auswählen und danach müssen die Angaben mit **OK** bestätigt werden.

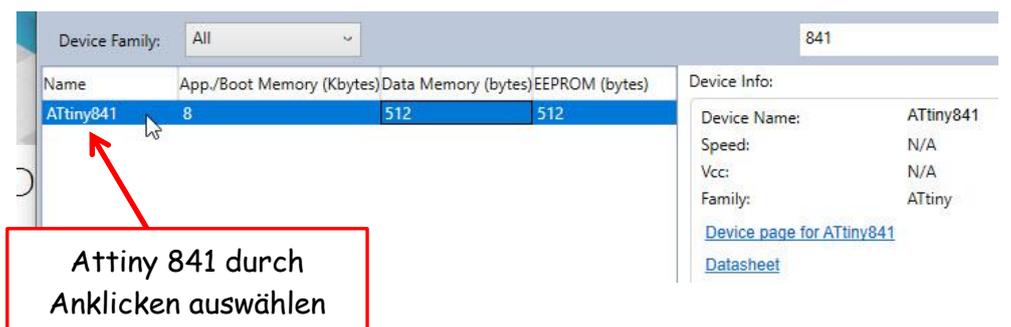
Im nächsten Bild muss der Prozessor ausgewählt werden.

Wenn ich im oberen rechten Feld einen Teil der Bezeichnung eingebe, werden mir alle möglichen Prozessoren mit dieser Angabe angezeigt. Bei der Eingabe von „841“ wird mir nur ein Prozessortyp vorgeschlagen, den Attiny 841.

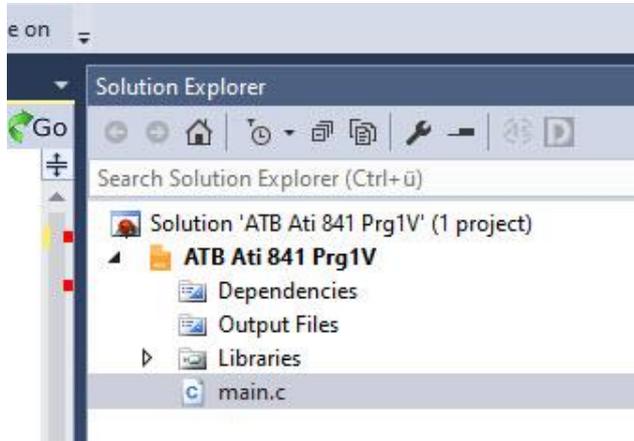


Durch Anklicken wird der **Attiny 841** ausgewählt.

Durch Anklicken von **OK** wird der Attiny 841 bestätigt.



Als nächste öffnet sich das eigentliche Programmfenster. Das Atmel Studio gibt uns hier bereits eine Programm vor mit dem Namen „main.c“.



```

main.c* x
main.c
D:\AAA Technik\Programme AV
/*
 * ATB Ati 841 Prg1.c
 *
 * Created: 22.03.2020 11:58:15
 * Author : hjsee
 */

#include <avr/io.h>

int main(void)
{
    /* Replace with your application code */
    while (1)
    {
    }
}
    
```

Im „Solution Explorer“ werden mir noch verschiedene Option angezeigt oder kann verschiedene Einstellungen ändern.

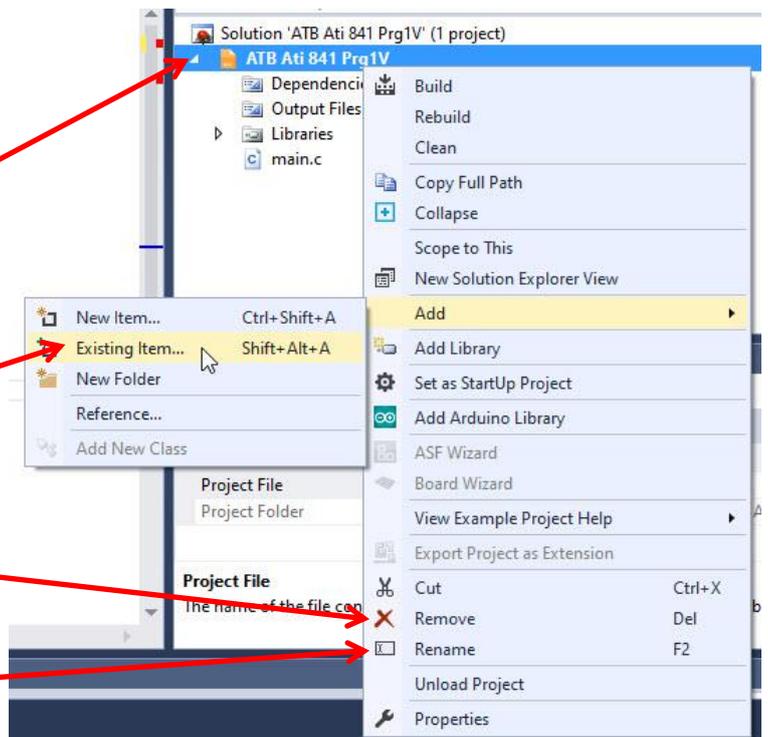
Auswählen und Einstellungen im „Solution Explorer“:

Mit der rechten Taste der Maus das Menü aufrufen

Dateien von anderen Programmen auswählen und einfügen

Löschen von Programmen oder Dateien

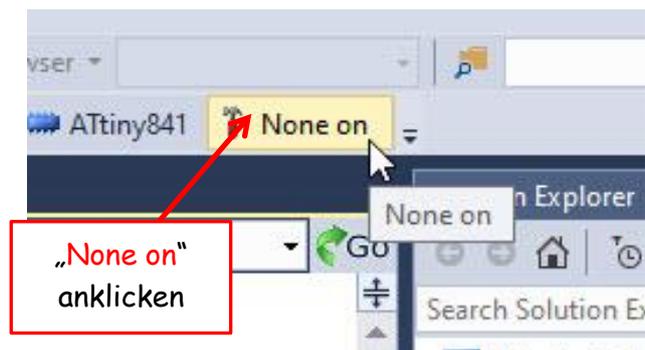
Dateien umbenennen



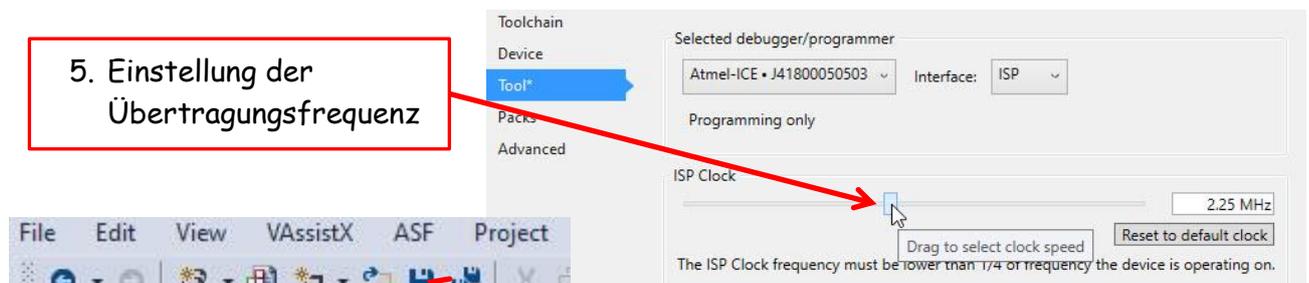
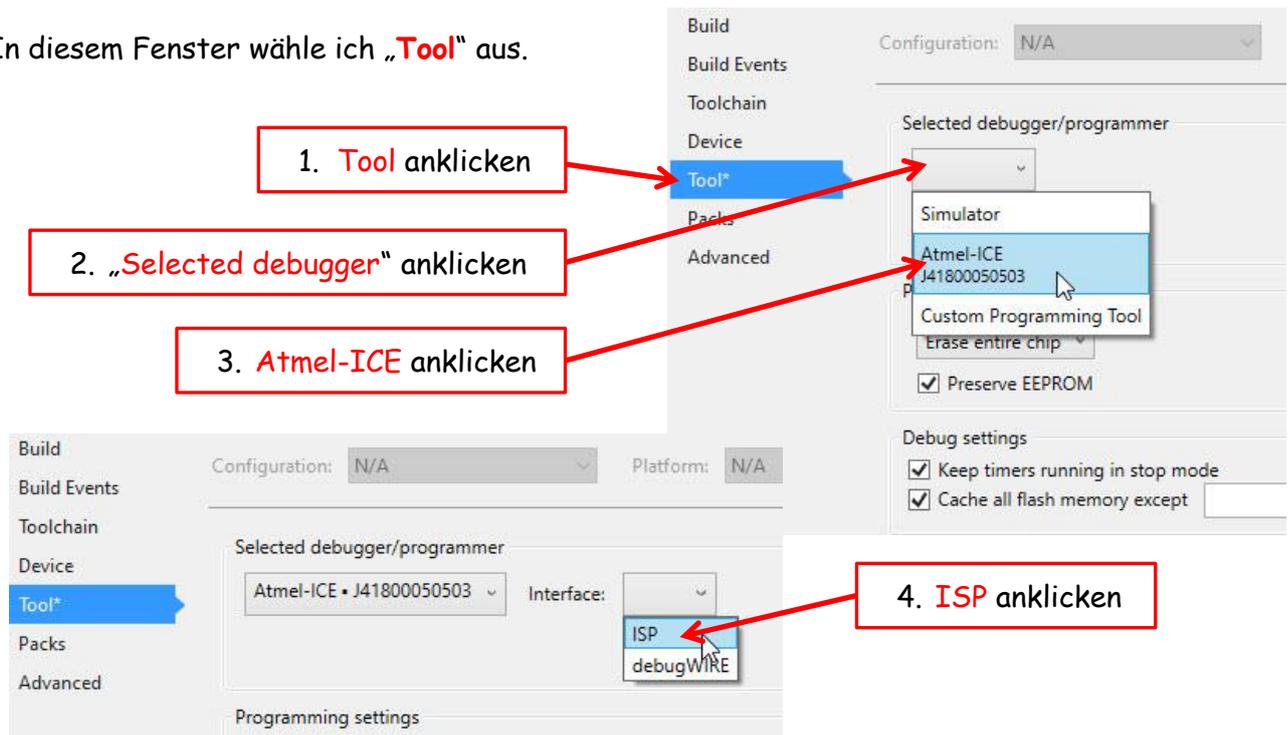
In diesem Menü sind noch andere Einstellungen möglich, auf die nicht eingehen werde.

In der oberen Reihe kann der Punkt „None on“ angeklickt werden.

Es erscheint eine weitere Seite auf der ich wieder Einstellungen vornehmen kann. Auch hier sind noch andere Einstellungen möglich, auf die ich nicht eingehe.



In diesem Fenster wähle ich „Tool“ aus.

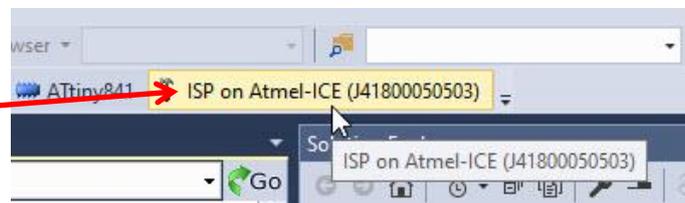


Solange die Einstellungen nicht abgespeichert wurden, wird „ * “ angezeigt

6. Abspeichern der Einstellung

Beenden mit „ x “

In meiner Leiste wird die Einstellung angezeigt



Die Einstellungen die wir auf dieser Seite vornehmen werden später bei der Programmierung des Attiny 841 verwendet und angezeigt. Bei der Einstellung des **ISP Clock** dürfen beim ersten Mal maximal 1/4 der eingestellten Frequenz verwendet werden. Später kann diese Frequenz erhöht werden. Nach jeder Veränderung der Einstellung bitte neu speichern. Nach dem beenden geht es weiter mit der **main.c**.

Sehen wir uns noch mal die Hardware dazu an:

Platine P173 (mit Quarz)

Belegung: **PA0 - LED 2**
 PA1 - LED 3
 PA2 - Vcc/2
 PA3 - Regler
 PB2 - Taster 2
 PA7 - Taster 3

Platine P177 (ohne Quarz)

Belegung: PA0 - Taster T1
 PA1 - A0
 PA2 - A1
 PA3 - A2
 PA5 - Taster T4
 PA7 - Taster T3
 PB0- Taster T2
 PB1 - Taster T5
 PB2 - LED 2

Zum Test der Ausgänge und der Funktion der Platine **P173** habe ich dieses kleine Programm geschrieben. Die Funktion besteht darin, die **LED2** in einer Frequenz von **1 Sekunde** blinken zu lassen. P177 werde ich mit den I/O Pins beschreiben.

```
#define F_CPU 16000000UL // Angabe der Frequenz, wichtig für die Zeit
#include <util/delay.h> // Einbindung Datei Pause
#include <avr/io.h> // Einbindung Datei Ausgänge

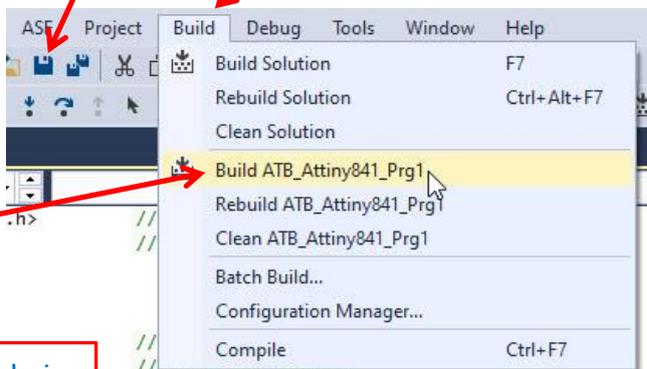
int main(void)
{
    DDRA=0b00000011; // DDRA PIN 0 und 1 auf Ausgang schalten
    PORTA=0b00000011; // PortA PIN 0 und 1 auf aus schalten
    while(1) // Programmschleife
    {
        PORTA |= (1<<PINA0); // LED 2 ein
        _delay_ms(500); // Pause 500ms
        PORTA &=~(1<<PINA0); // LED 2 aus
        _delay_ms(500); // Pause 500ms
    }
}
```

1. Diskette anklicken zum Abspeichern

2. Button „Build“ anklicken

Als nächstes wollen wir unser Programm abspeichern und kompilieren um es anschließend zum Attiny 841 zu übertragen.

3. Button „Build“ anklicken oder mit „F7“



Wenn keine Fehler sind kann das Programm übertragen werden, sonst beginnt die Fehlersuche.

Anzeige kein Fehler

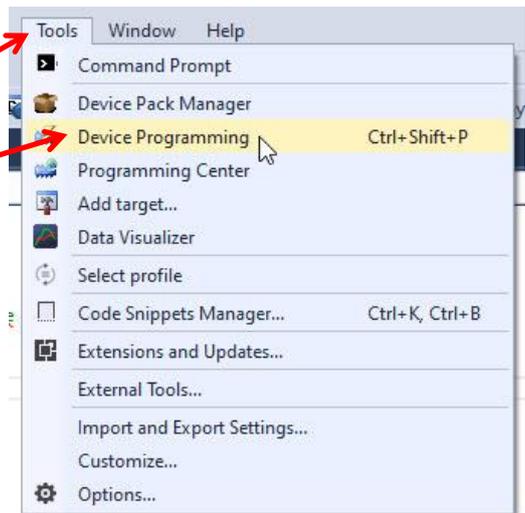
```
Done building target "Build" in project "ATB_Attiny841_Prg1.cproj".
Done building project "ATB_Attiny841_Prg1.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

Als nächstes wollen wir das Programm zum Attiny 841 übertragen.

„Tools“ anklicken

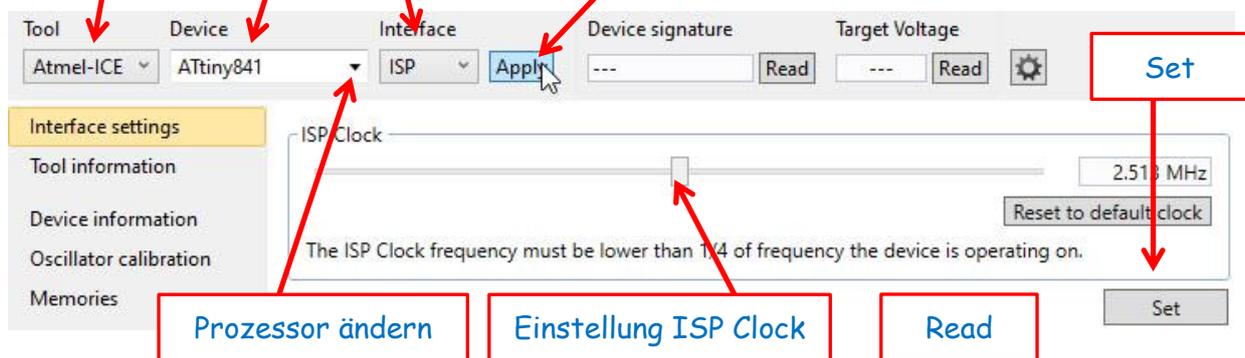
„Device Programming“ anklicken



Im nächsten Fenster werden mir Daten des Zielprozessors angezeigt und die Daten die ich bereits eingestellt hatte. Nach dem anklicken von „Apply“ wird mir der „ISP Clock“ angezeigt

Angabe von Atmel_ICE, Attiny841, ISP

Apply



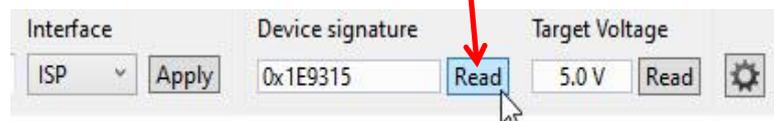
Prozessor ändern

Einstellung ISP Clock

Read

Set

Den **ISP Clock** kann ich durch ein verschieben des Reiters verändern und mit **Set** bestätigen.

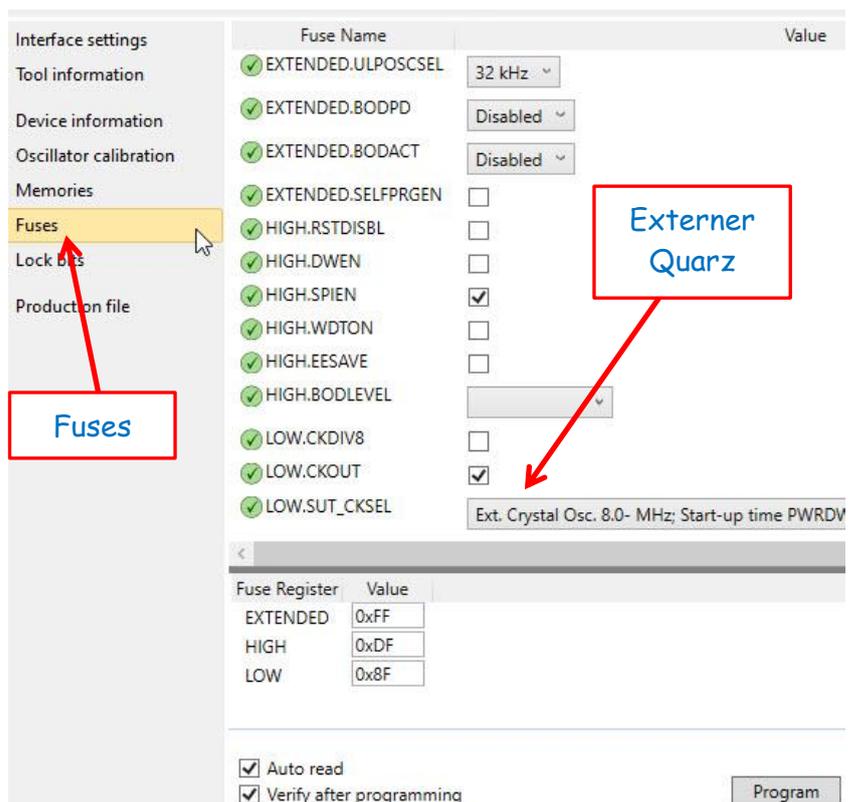


Als nächste wird der Button „**Read**“ angeklickt. Damit wird die Signatur aus dem Prozessor ausgelesen und mit dem eingestellten Prozessor verglichen. Falls diese nicht übereinstimmt kann mit „**Device**“ die Einstellung verändert werden.

Auf der nächsten Seite bitte „**Fuses**“ auswählen.

Die Einstellungen die ich hier mache müssen beim ersten mal auf den Prozessor übertragen werden. Beim nächsten Programmieren sind die Daten auf dem Prozessor gespeichert und brauchen nicht noch einmal übertragen werden.

Im ersten Bild ist die Einstellung für einen externen Quarz dargestellt.



Fuses

Externer Quarz

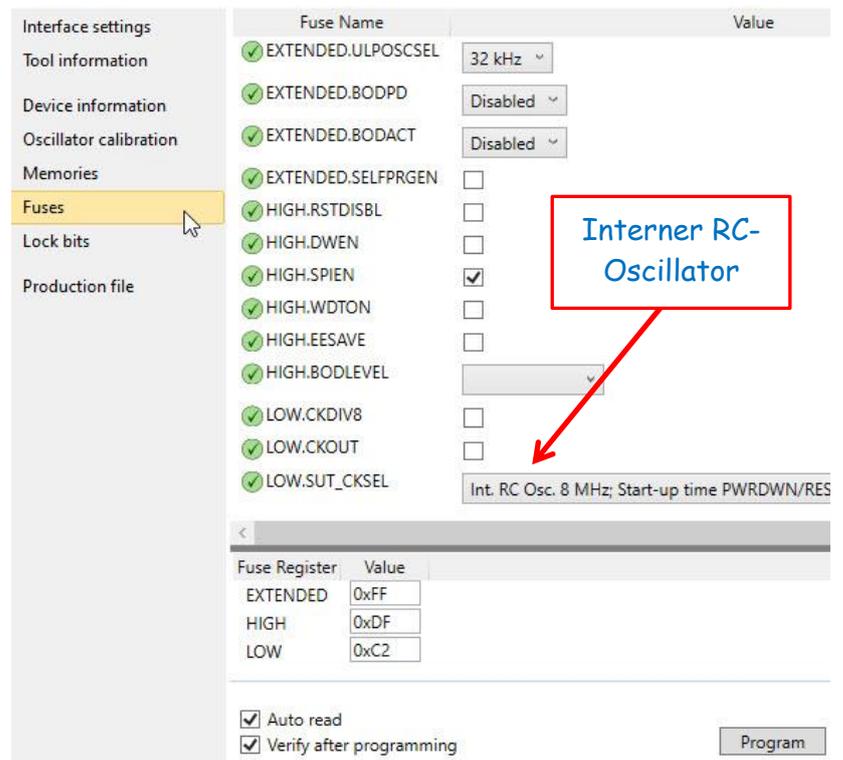
Im zweiten Bild habe ich die Einstellung mit dem internen RC-Oszillator dargestellt.

Einstellung mit Quarz

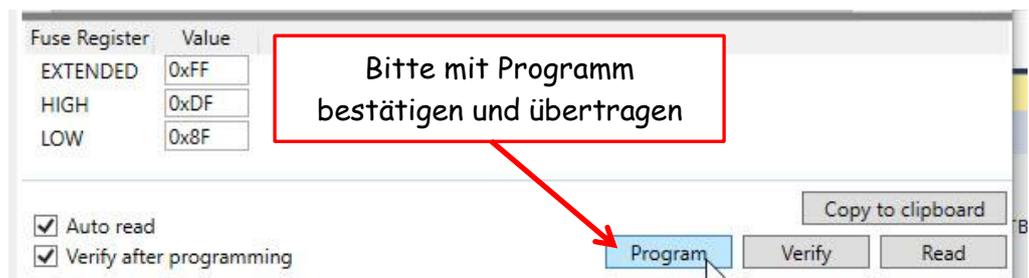
EXTENDED **0xFF**
HIGH **0xDF**
LOW **0x8F**

Einstellung ohne Quarz

EXTENDED **0xFF**
HIGH **0xDF**
LOW **0xC2**



Wenn die Einstellungen korrekt sind bitte mit Programm bestätigen und übertragen.



Bitte genau die Einstellungen überprüfen. Falsche Einstellungen können später unter Umständen nicht mehr korrigiert werden und der Prozessor kann nicht mehr programmiert werden.

Der Prozessor wird verlust !



Als letzten wird das Programm auf unseren Prozessor übertragen. Dazu bitte „Memories“ anklicken. Das korrekte Programm müsste bereits angezeigt werden. Danach noch „Programm“ anklicken und unser Programm wird auf den Attiny 841 übertragen.

Einige Teile des Textes wurden zur besseren Übersicht farblich gestaltet.

Die Nutzung erfolgt auf eigenes Risiko. Ich wünsche viel Spaß beim Bauen und programmieren
Achim

myroboter@web.de