

MIKROKONTROLLER & I²C BUS

by AS

www.makerconnect.de

<https://www.makerconnect.de/resource>

makerconnect.de

ARM Controller - SAM D21 J17A
(32 Bit Controller)
Teil 2 - Atmel Studio einrichten

SAM D21 J17A



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.

SAM D21 J17A - ein 32 Bit Contoller

Der Atmel ICE Programmer für SAM und AVR

Seit einiger Zeit gibt es neuen Programmer von der Atmel zur Programmierung von **SAM** und **AVR** mit der **Bezeichnung Atmel ICE**. Diesen Programmer gibt es in 3 verschiedenen Version

- Atmel ICE Entwicklung
- Atmel ICE Basic
- Atmel ICE PCBA

Im ersten Teil - Software 1 - Atmel Studio einrichten, möchte ich euch zeigen wie man den Atmel ICE Basic anschliesst und die Einstellungen im Atmel Studio 7 für den SAM D21 J17A vornimmt und ein erstes Programm damit erstellt.

Als Board verwende ich ein NanoSAMD21 von der Firma Rodenhausen. Diesen habe ich in mein „Modulares Board“ System angepasst.



Atmel ICE Basic mit Programmierkabel

Die Anwendung des Programmers bitte der Beschreibung von Atmel entnehmen.



NanoSAMD21 auf der Platine 110 mit 4x Wannenbuchse 10 pol. Für die Portausgänge (**3,3V**), 2x Wannenbuchse 10 pol. Zum Anschluss des Netzteiles **5V** und **12V** und **I²C Bus**, 3 LEDs zur Statusanzeige, 4 Taster und 3 LEDs als Ortsbedienung.

Der Pegelwandler für den I2C Bus und die zugehörigen Widerstände sind noch nicht bestückt.

Die Hardware kann wieder an meine anderen Boards und Module angeschlossen werden. Die Beschreibung der Hardware erfolgt im Teil SAM D21 - Hardware.

Die Portausgänge sind nur für 3,3V und 6mA (max.) zulässig. Die Verbindung zu den anderen Modulen und die Anbindung an den I²C Bus erfolgt mit 5V.

In diesem Teil wollen wir uns vorrangig um die Einstellung und Programmierung kümmern.

Voraussetzung ist, dass das Atmel Studio 7 mit allen notwendigen Treibern korrekt installiert ist. Das Programm kann kostenlos von der Seite des Herstellers geladen werden.

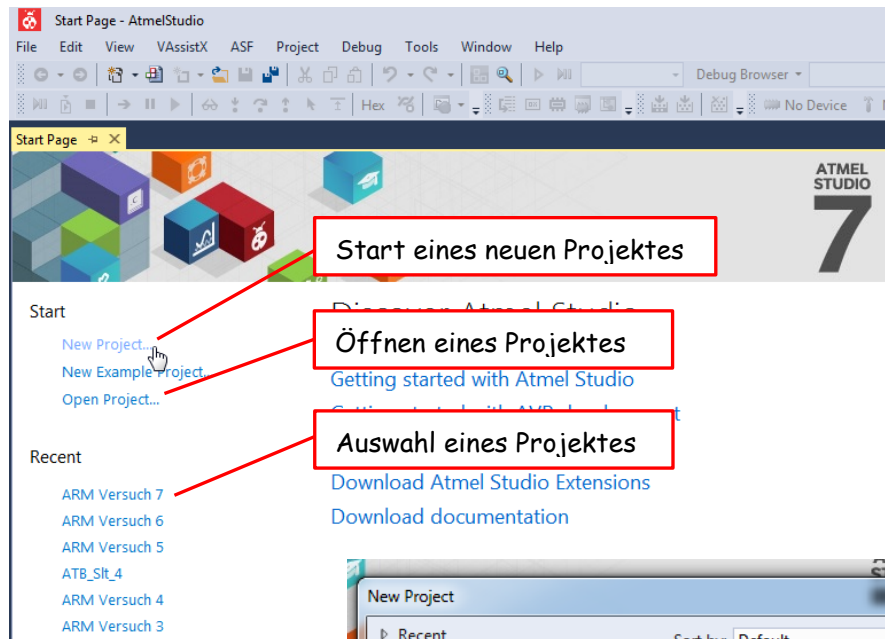
Icon des Atmel Studios 7 auf dem Bildschirm, Start durch anklicken



Startbildschirm Atmel Studio 7 mit Auswahl von

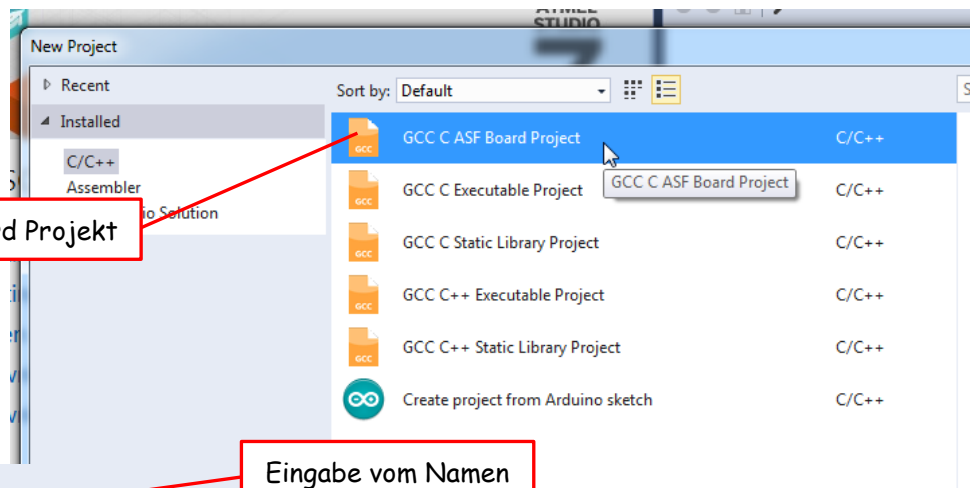
- Start eines neuen Projektes
- Öffnen eines Projektes aus dem Ordner
- Auswahl eines kürzlich verwendeten Projektes

Am oberen Rand sind verschiedene Auswahlmöglichkeiten. Die Auswahl erfolgt durch Anklicken.



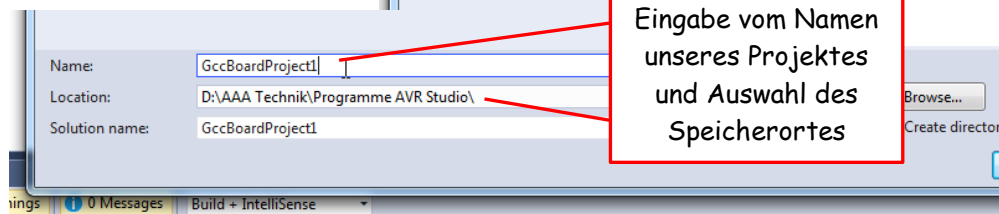
Auswahl GCC ASF Board Projekt

Auswahl eines GCC ASF Projektes durch anklicken



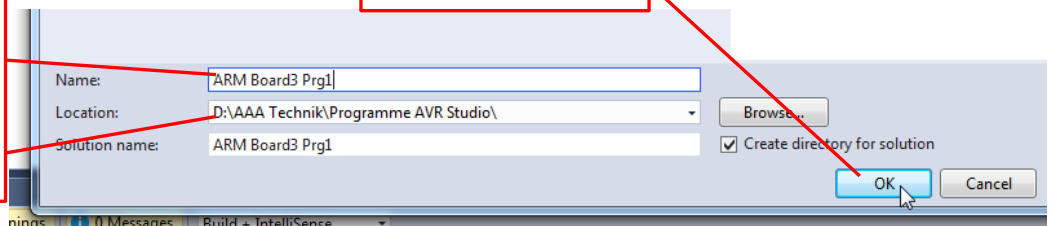
Eingabe vom Namen unseres Projektes und Auswahl des Speicherortes

Als nächste müssen wir unserem Projekt einen Namen geben und den Speicherort angeben



Bestätigen mit OK

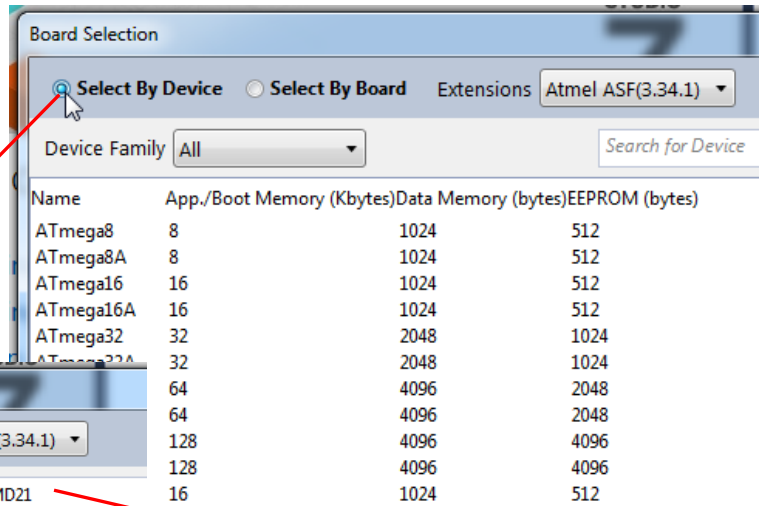
Eingabe eines neuen Namens für unser Projektes und Angabe des Speicherortes



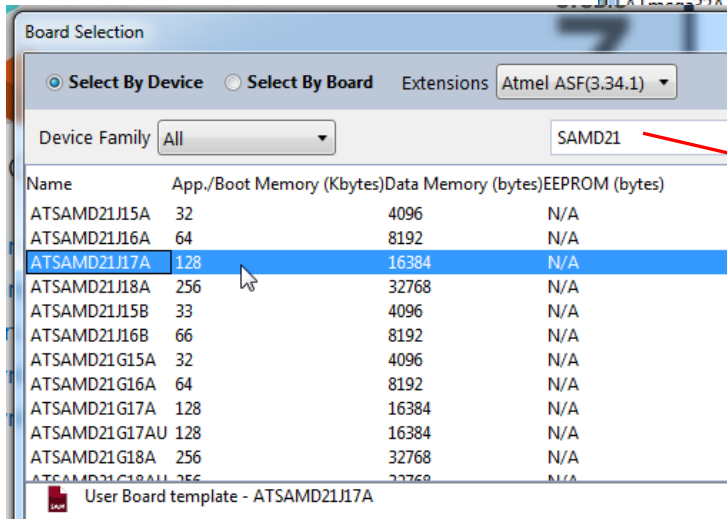
Einen **Name** für das neue Projekte angeben, den **Speicherort** angeben oder mit dem **Browser** auswählen und anschließend mit **OK** bestätigen.

Auf der nächsten Seite müssen wir **Select By Device** auswählen und einen Prozessor angeben. Beim Start dieser Seite werden uns alle möglichen Prozessoren vorgeschlagen

Select By Device einstellen



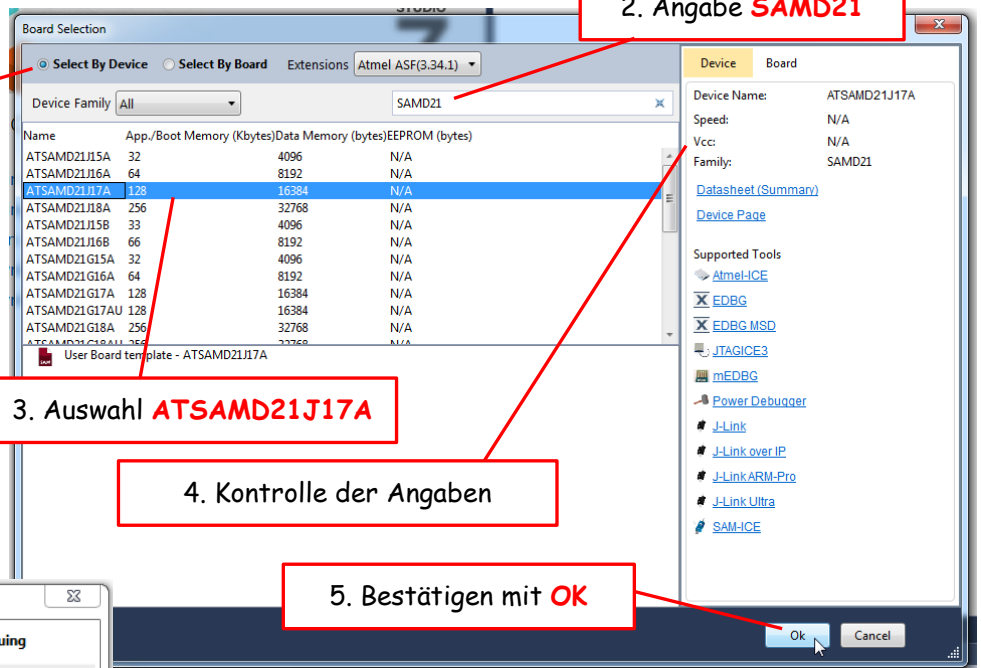
Angabe SAM D21



Ich kann auch direkt einen Prozessor angeben. Dazu kann ich ein Teil der Bezeichnung, z.B. SAMD21 angeben. Es werden mir alle möglichen Prozessoren mit der Bezeichnung SAMD21 angegeben. Durch Anklicken der korrekten Bezeichnung wähle ich den gewünschten Prozessor aus.

1. **Select By Device** einstellen

Nach der Auswahl eines Prozessors werden uns auf der rechten Seite nochmal seine technischen Daten dargestellt. Sind alle Angaben korrekt, kann ich es mit **OK** bestätigen.

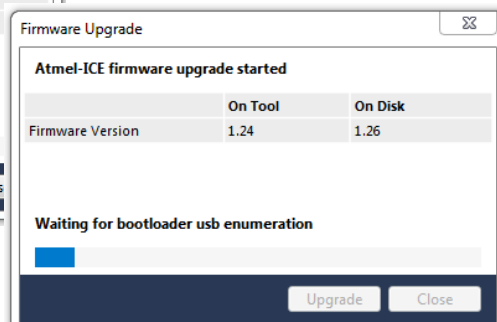
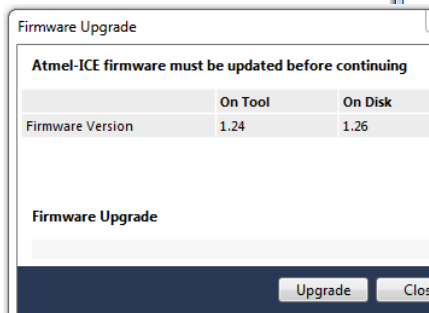


2. Angabe **SAMD21**

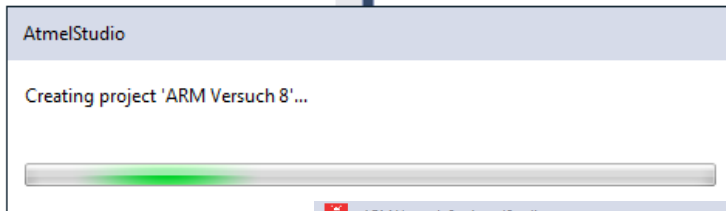
3. Auswahl **ATSAMD21J17A**

4. Kontrolle der Angaben

5. Bestätigen mit **OK**



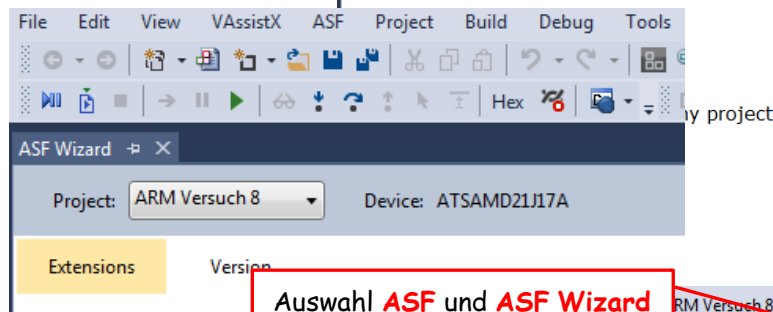
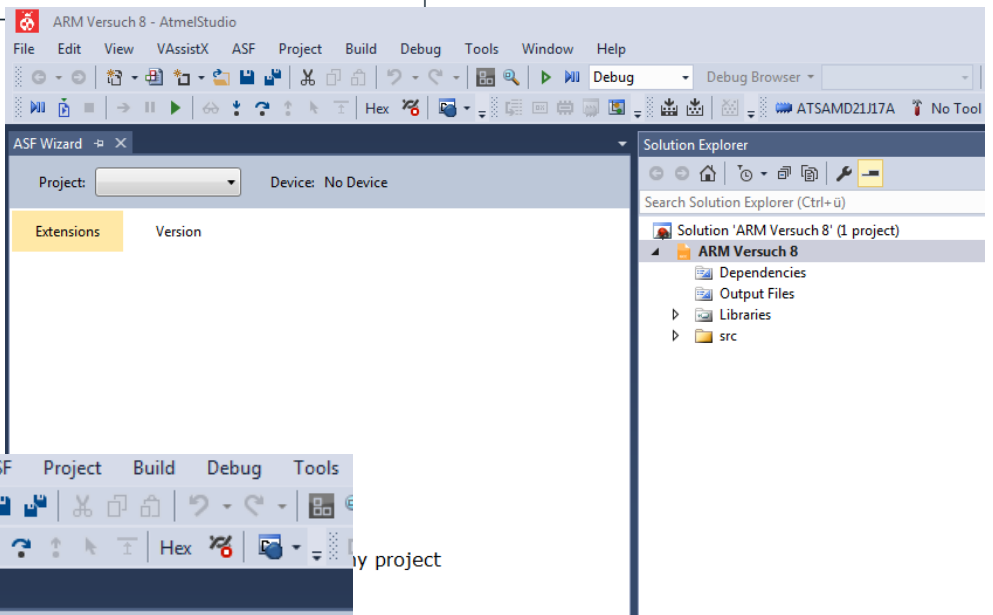
Teilweise wird das Programm von kurzen Meldungen unterbrochen, das es eine neuere Version zu einem Treiber oder Programmer gibt. Auf Upgrade klicken und warten bis er fertig ist.



Vom Programm her wird ein entsprechendes Projekt angelegt. Das kann allerdings etwas dauern.

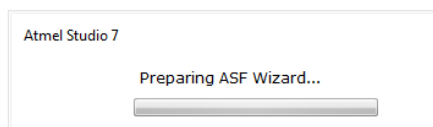
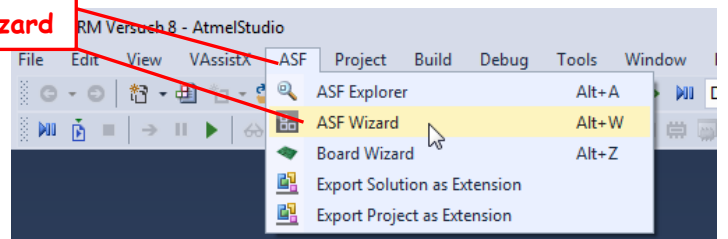
Als nächstes wird das angelegte Projekt angezeigt und der Name unseres Projektes angegeben.

Auf der rechten Seite werden die ersten notwendigen Bibliotheken angezeigt.



Auswahl **ASF** und **ASF Wizard**

Als nächstes erfolgt in der obersten Zeile die Auswahl von **ASF** und weiter **ASF Wizard**



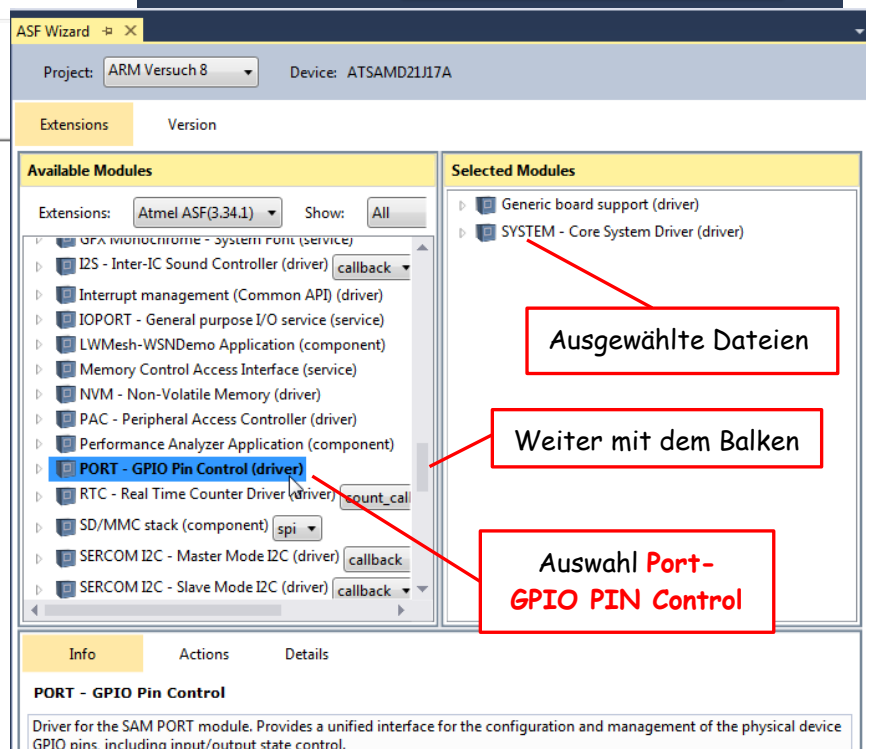
Wieder etwas warten ...

Als nächstes wird uns dieses Bild angezeigt.

Auf der linken Seite wird **Available Modules** (verfügbare Module) angezeigt. Das sind alle zur Verfügung stehenden Dateien bzw. Bibliotheken vom ASF. Auf der rechten Seite stehen bereits vom Programm ausgewählte Dateien.

Mit dem Balken kann man weiter scrollen.

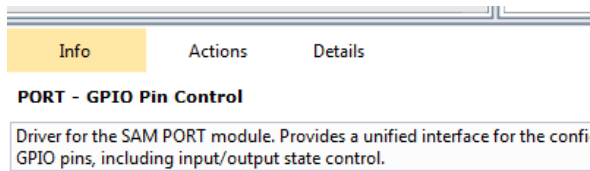
Die Auswahl der gewünschten Datei erfolgt durch anklicken.



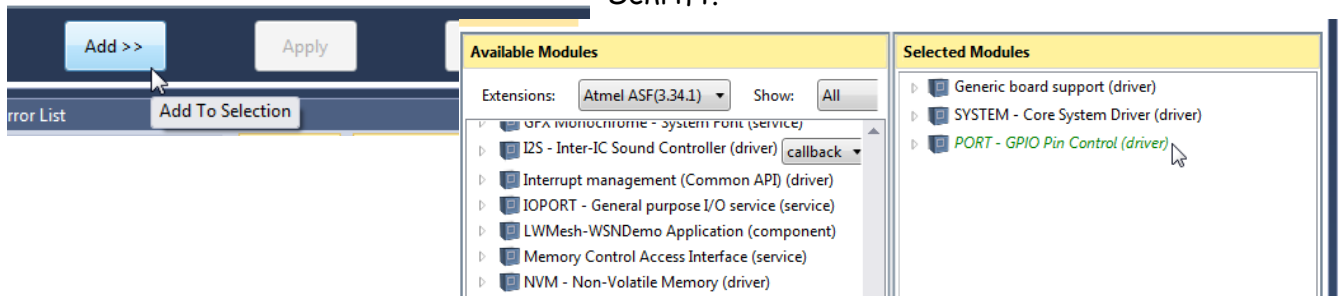
Ausgewählte Dateien

Weiter mit dem Balken

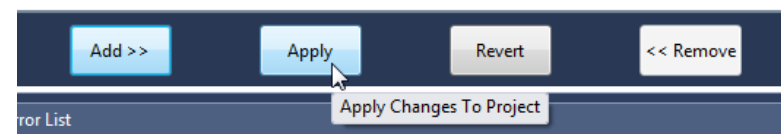
Auswahl **Port-GPIO PIN Control**



Die Auswahl der gewählten Datei erfolgt durch einen Klick auf **Add >>**. Danach erscheint die ausgewählte Datei im **Selected Modules** in grüner Schrift.

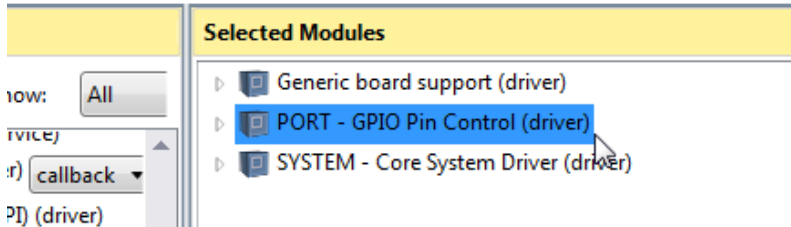
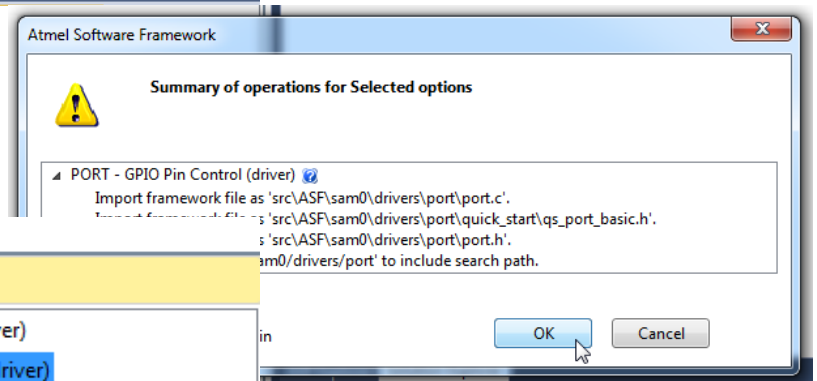


Driver for the SAM Analog Comparator module. Provides a unified interface for the configuration and device's hardware analog comparators.



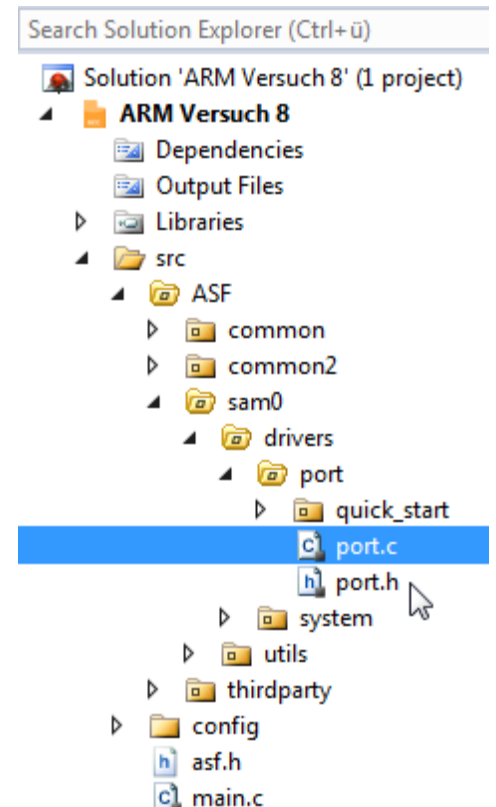
Mit einem Klick auf **Apply** wird unsere Auswahl bestätigt

Danach erfolgt noch mal eine Abfrage bzw. Anzeige unserer Ausgewählten Dateien. Diese muss ich mit **OK** bestätigen



Danach wird unsere Auswahl **PORT - GPIO Pin Control** in schwarzer Schrift dargestellt

Im **Solution Explorer** werden mir nun alle verwendeten Dateien bzw. Bibliotheken angezeigt. Damit kann ich kontrollieren ob die ausgewählten Dateien **port.c** und **port.h** zur Verfügung stehen.



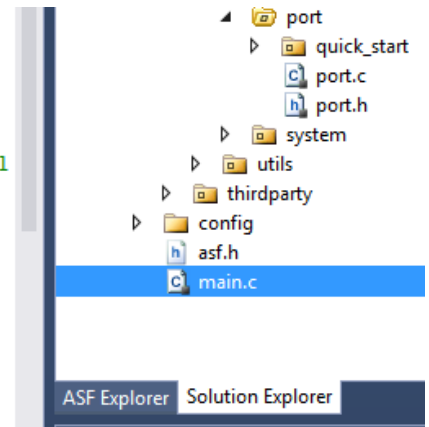
```

/*
 * Include header files for all drivers that have been imported from
 * Atmel Software Framework (ASF).
 */
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
 */
#include <asf.h>

int main (void)
{
    system_init();

    /* Insert application code here, after the board has been initialized. */
}

```



In der **main.c** wird mir das aktuelle Programm angezeigt. Da es ein neues Programm ist, besteht es nur aus einem „Grundgerüst“ und einigen Angaben zu Atmel und ASF.

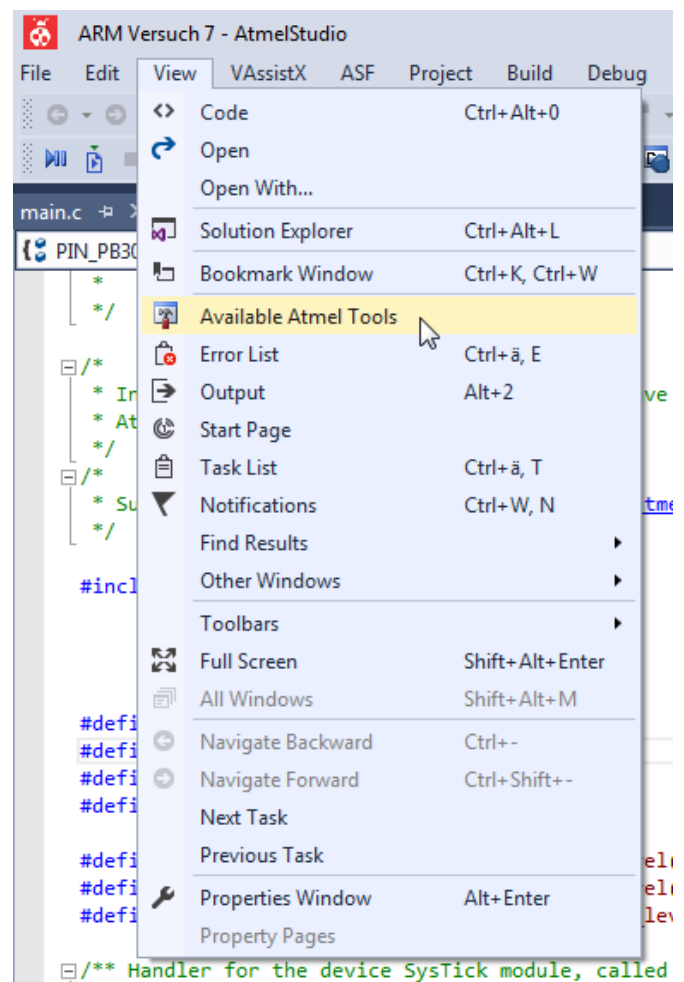
In die **main.c** kann ich nun mein eigenes Programm schreiben.

Bevor wir mit dem Programm anfangen müssen wir noch unseren Programmierer **Atmel ICE** einstellen bzw. kontrollieren.

Bei der Installation des Atmel Studios darf der Programmierer nicht über USB am Computer angeschlossen sein.

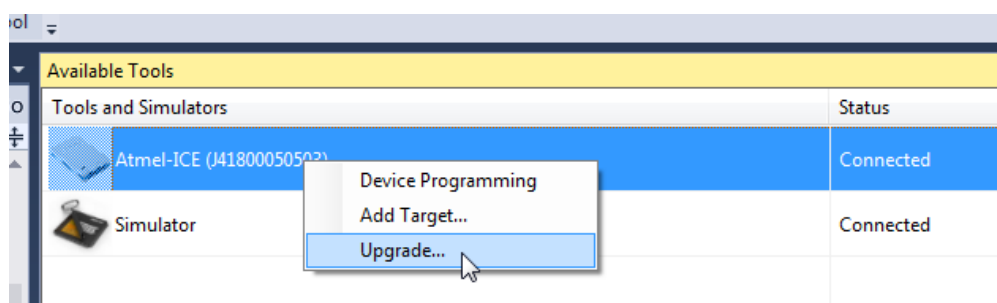
Nach der Erfolgreichen Installation kann ich den Programmierer anschliessen und er wird vom Atmel Studio erkannt.

Mit einem Klick auf **View** und **Available Atmel Tools** kann ich das kontrollieren.



Es wird mir der **Atmel ICE** mit Seriennummer angezeigt. Mit einem Klick auf die rechte Maustaste kann ich ein Update veranlassen.

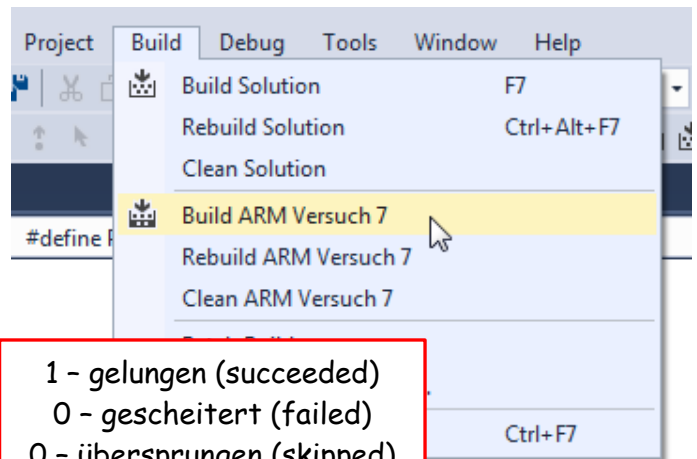
Im Allgemeinen wird der Programmierer ohne Probleme erkannt und falls nötig ein Update angeboten oder gleich durchgeführt.



Als nächstes müssen wir unser kompilieren. Dazu muss ich **Build** und danach **Build ARM Versuch7** (Beispiel Programm) anklicken.

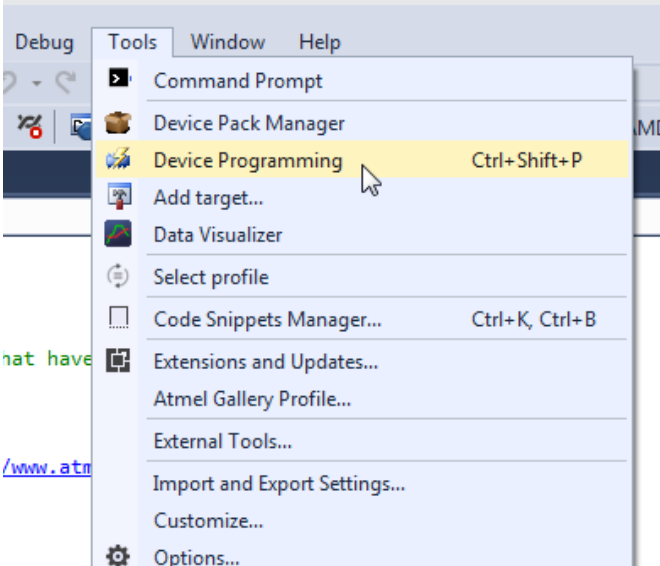
Dadurch wird das Programm in eine Sprache übersetzt, die der **SAM D21** versteht und alle notwendigen Bibliotheken bzw. Datei eingebunden. Der Erfolg wird mir am Bildschirm angezeigt oder wie viele Fehler ich gemacht habe.

Die Angaben stehen am unteren Rand:



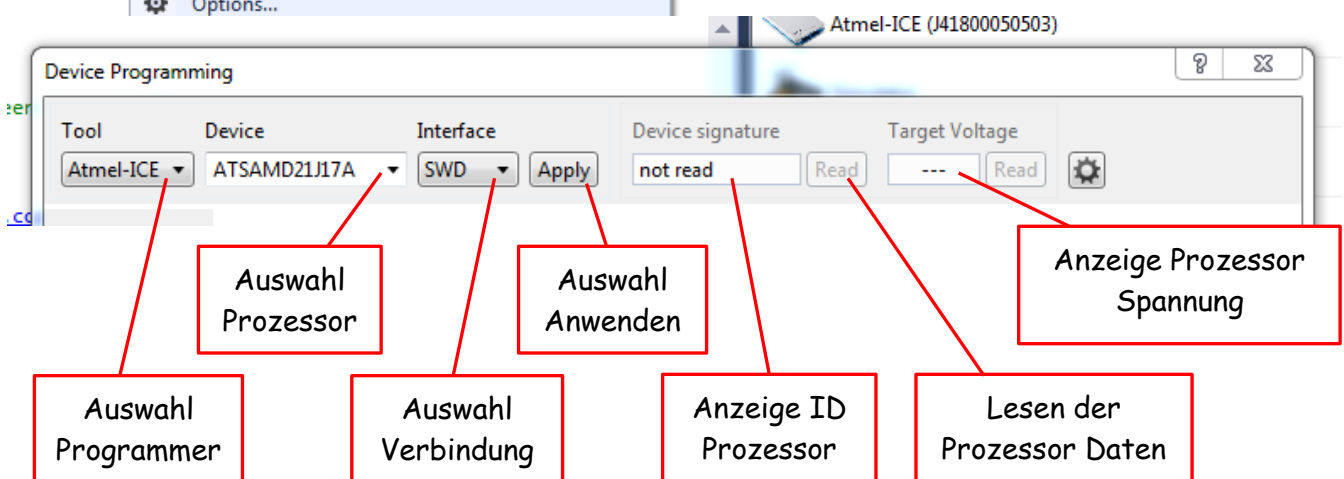
```
Done building target "Build" in project "ARM Versuch 7.cproj".
Done building project "ARM Versuch 7.cproj".

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

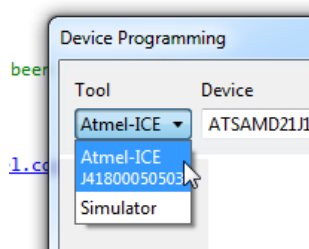


Jetzt kann ich das Programm an meinen **SAM D21** übertragen. Dazu muss ich **Tools** und danach **Device Programming** anklicken

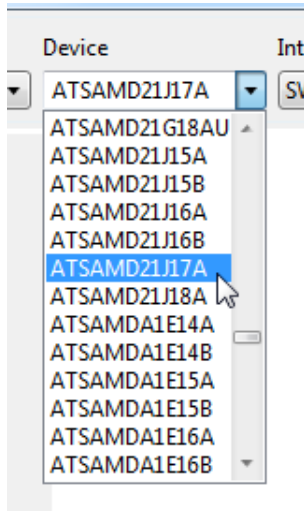
Im nächsten Bild werden mir die möglichen Einstellungen für unseren Prozessor angezeigt.



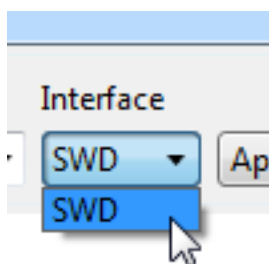
Als nächstes wollen wir uns die einzelnen Daten zu unserem Programmer und Prozessor ansehen. Es können auch verschiedene Einstellungen vorgenommen werden



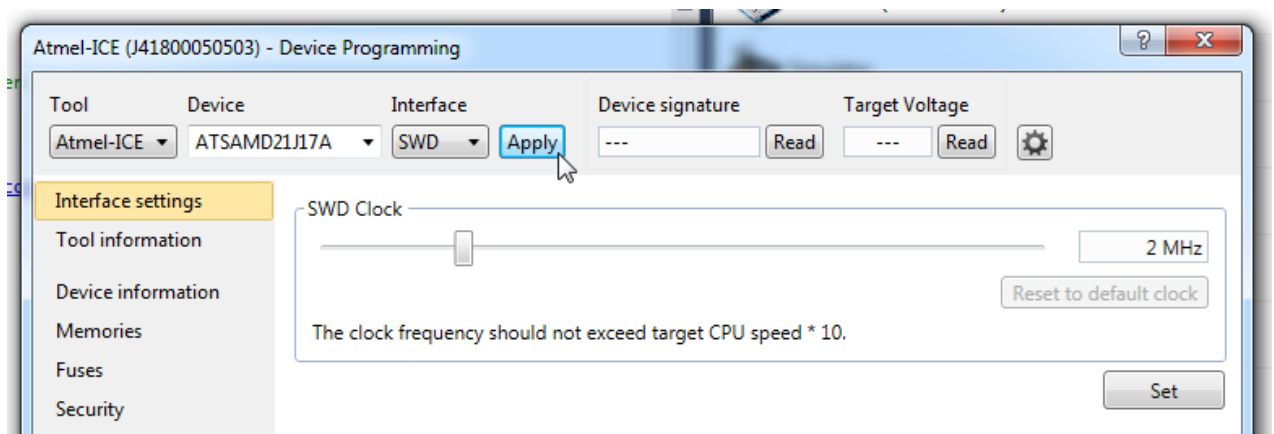
Mit **Tool** erfolgt die Auswahl des **Atmel ICE**. Wenn der Programmer nicht angezeigt wird, wurde er vom Programm nicht erkannt.



Mit **Device** kann ich verschiedene Prozessoren Auswählen. Die Einstellung muss mit dem verwendeten Prozessor auf dem Bord übereinstimmen, sonst bekomme ich eine Fehlermeldung und kann das Programm nicht übertragen.

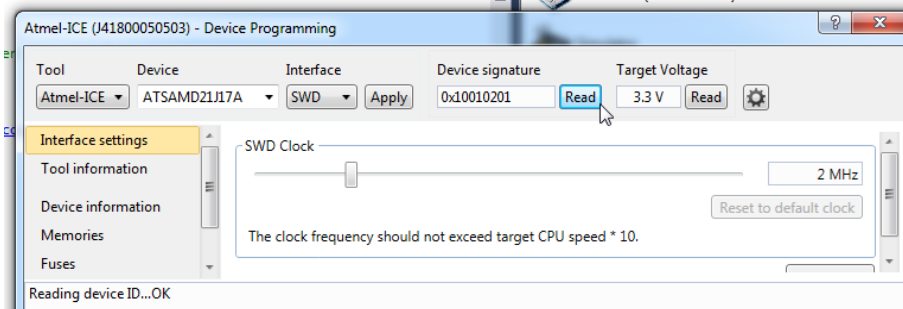
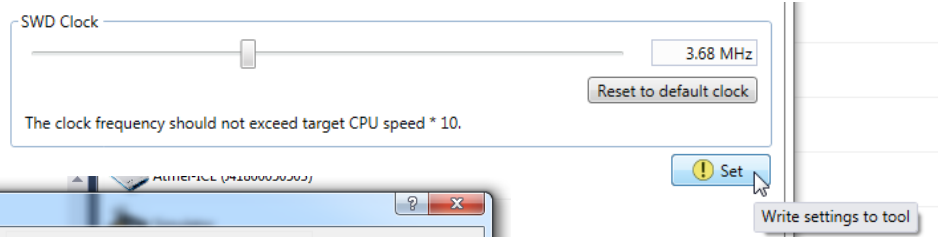


Mit **Interface** wähle ich die Verbindung zu meinem Board aus. Am Programmer **Atmel ICE** sind 2 verschiedene Steckbuchsen (SAM und AVR) vorhanden. Je nach ausgewähltem Prozessor muss diese korrekt angeschlossen sein.



Mit einem Klick auf **Apply** verwende ich die eingestellten Daten und es erscheint das **Interface settings**. Damit kann ich die Übertragungsgeschwindigkeit zu meinem Board einstellen. Dazu einfach den Block unter **SWD Clock** nach rechts ziehen und die eingestellten 2 MHz verändern sich. Die eingestellte Frequenz anschliessend mit **Set** bestätigen

So könnte es dann aussehen



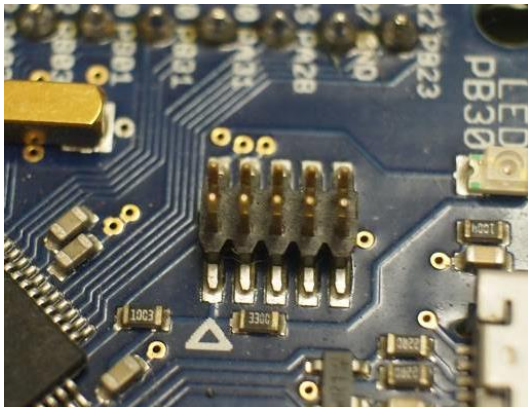
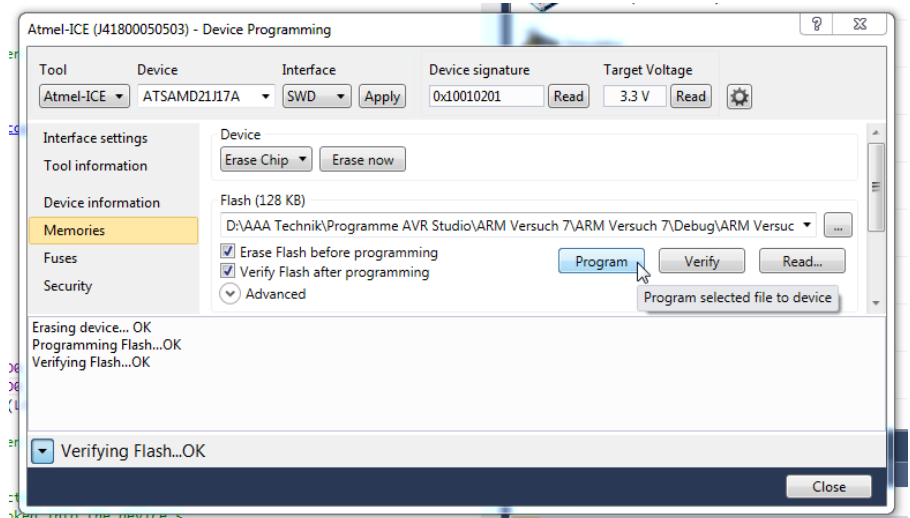
Mit einem Klick auf **Read** lese ich die **Device signature** und die verwendete Betriebsspannung des Prozessors aus.

Stimmt die Einstellung der **Device** und der **Device signature** nicht überein bekomme ich eine Fehlermeldung.

Mit einem Klick auf **Memories** kann ich die Übertragung beginnen.

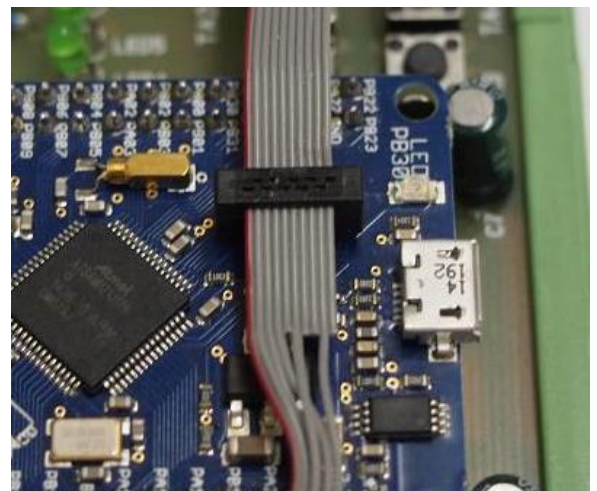
Damit habe ich alle notwendigen Einstellungen vorgenommen und ein Programm ausgewählt. Mit einem Klick auf **Programm** beginnt die Übertragung an meinen Prozessor.

Bei der Beschreibung der Einstellungen im Atmel Studio habe ich mich auf das wesentliche beschränkt. Es sind noch viele andere Einstellungen möglich.



Ein Problem auf Grund der Grösse besteht bei der SWD Verbindung. Auf der Platine ist 10 polige Steckverbindung mit 1,27mm. Der Pin 1 ist durch ein kleines Dreieck auf der Platine bezeichnet. An der Verbindungsleitung ist das Kabel 1 durch eine rote Ader gekennzeichnet.

Bitte genau auf den korrekten Sitz achten.



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