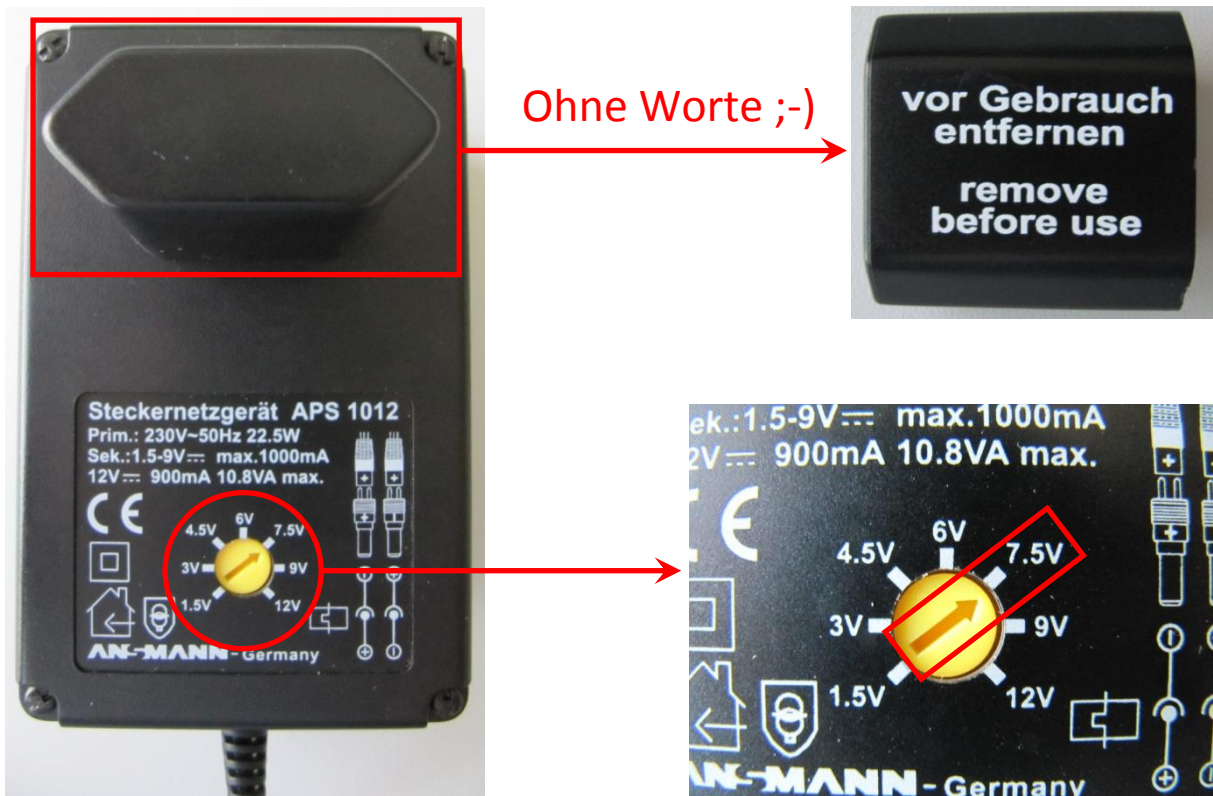
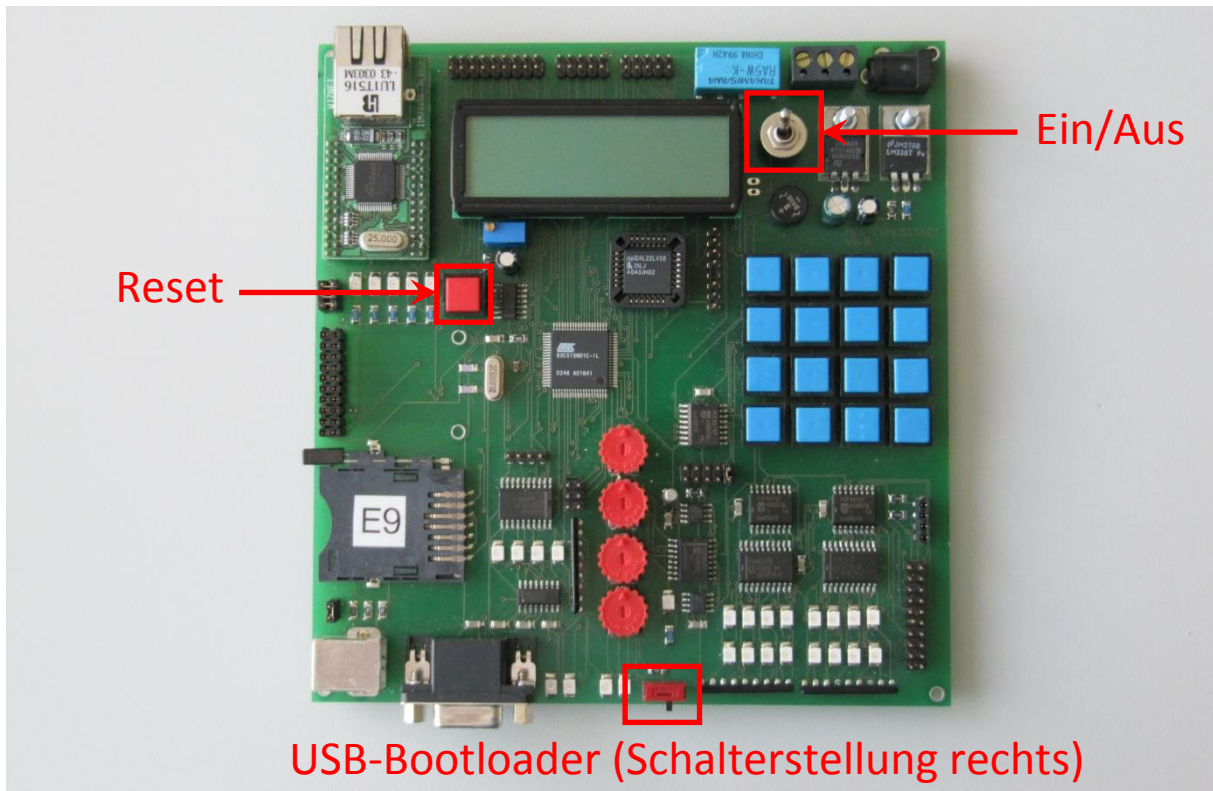


ANLEITUNG EVALUATIONSBOARD MKS-EVAL51SND1



PROGRAMMANPASSUNG

Zunächst wird das **HelloWorld**-Programm (siehe **µVision2-Guide**) für den Betrieb auf dem Mikrocontroller angepasst. Dazu wird der bestehende Quellcode durch folgenden ersetzt:

```
#include <REGSND1.H>

void wait(unsigned int in_nTime);

void main(void)
{
    bit bToggleFlag = 0;

    while(1)
    {
        if(bToggleFlag)
        {
            P5=0x05;
            bToggleFlag = 0;
        }
        else
        {
            P5=0x0A;
            bToggleFlag = 1;
        }
        wait(1000);
    }
}

void wait(unsigned int in_nTime)
{
    unsigned int nTimeCount = 0;
    unsigned int nCount = 0;

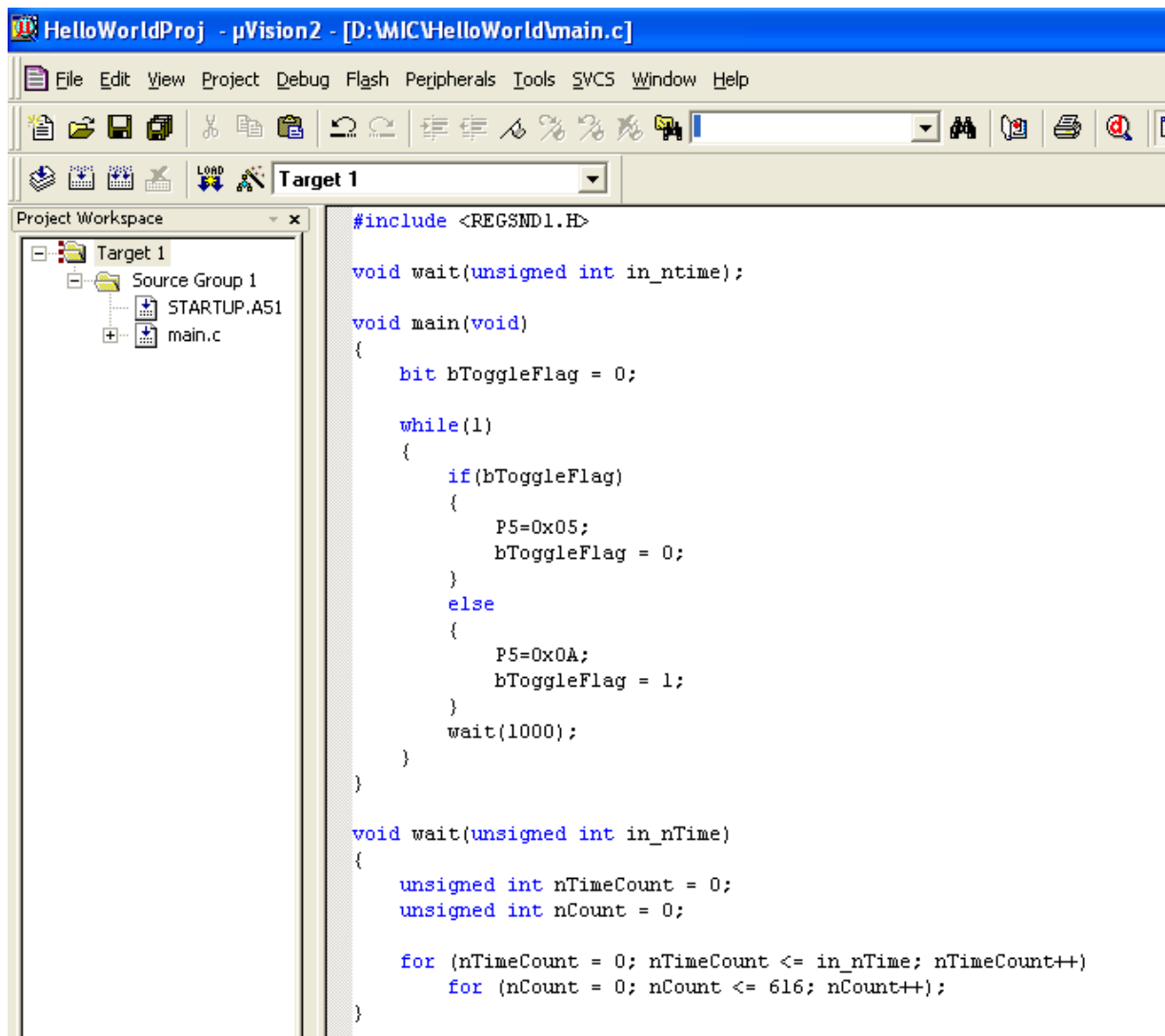
    for (nTimeCount = 0; nTimeCount <= in_nTime; nTimeCount++)
        for (nCount = 0; nCount <= 616; nCount++);
}
```

Das Toggeln des Ports wird nicht mehr direkt nacheinander durchgeführt, sondern durch ein Bit unterschieden. Zudem wird nicht wie im ursprünglichen Programm auf den Port1, sondern auf Port5 zugegriffen. Port5 ist auf dem Evaluationsboard die einzige Schnittstelle zur direkten Ansteuerung von Leuchtdioden. Die anderen LEDs sind nur über I2C-Bus ansteuerbar. Es ist zu beachten, dass es sich bei Port5 um einen 4 Bit breiten Port und nicht wie sonst üblich, um einen 8-Bit breiten handelt. Das bedeutet, dass die oberen Bits keinesfalls gesetzt werden dürfen. Gegebenenfalls sollte zur Sicherheit bei Setzen eines Wertes auf Port5 die oberen 4 Bit maskiert werden (**P5 = 0x0F & /*Zu setzender Wert*/**)

Um das Toggeln des Ports für das menschliche Auge sichtbar zu machen, wird eine Verzögerungsfunktion verwendet. Diese besteht aus zwei ineinander verschachtelten For-Schleifen,

wobei die äußere Schleife die Verzögerung in Millisekunden festlegt. Die innere Schleife stellt nur eine Annäherung zur Angleichung der internen Abarbeitungstaktschritte an eine Millisekunde dar.

Anmerkung: Für präzise Verzögerungen sollte ein **Timer** verwendet werden.



```
#include <REGSND1.H>

void wait(unsigned int in_nTime);

void main(void)
{
    bit bToggleFlag = 0;

    while(1)
    {
        if(bToggleFlag)
        {
            P5=0x05;
            bToggleFlag = 0;
        }
        else
        {
            P5=0x0A;
            bToggleFlag = 1;
        }
        wait(1000);
    }
}

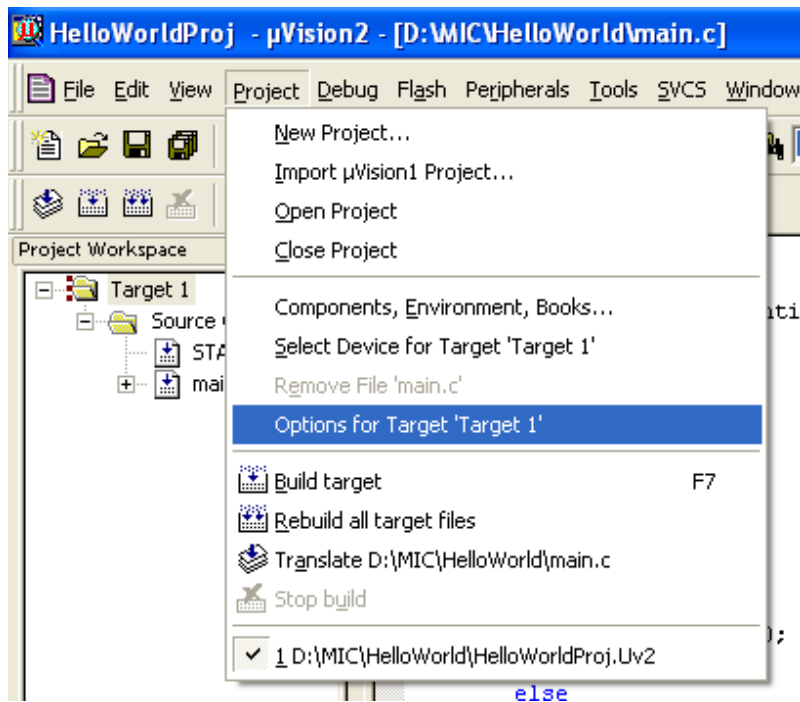
void wait(unsigned int in_nTime)
{
    unsigned int nTimeCount = 0;
    unsigned int nCount = 0;

    for (nTimeCount = 0; nTimeCount <= in_nTime; nTimeCount++)
        for (nCount = 0; nCount <= 616; nCount++);
}
```

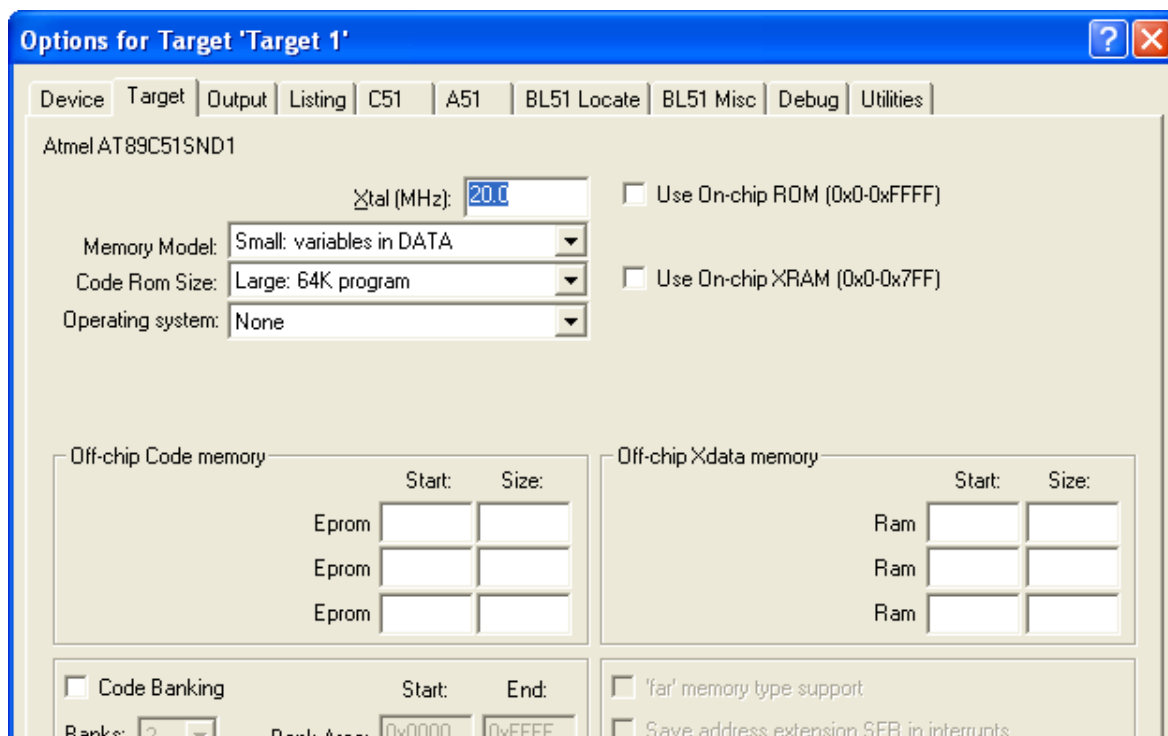
Für die Überspielung des Quellcodes auf den Mikrocontroller wird vom Hersteller **Atmel** ein entsprechendes Tool namens **FLIP** bereitgestellt. Dieses benötigt zur Programmierung des Mikrocontrollers ein Hex-File. Keil µVision2 erzeugt beim Kompilieren standardmäßig kein Hex-File. Dies muss durch folgende Einstellung explizit festgelegt werden.

PROJEKTOPTIONEN FÜR DAS EVALUATIONSBOARD

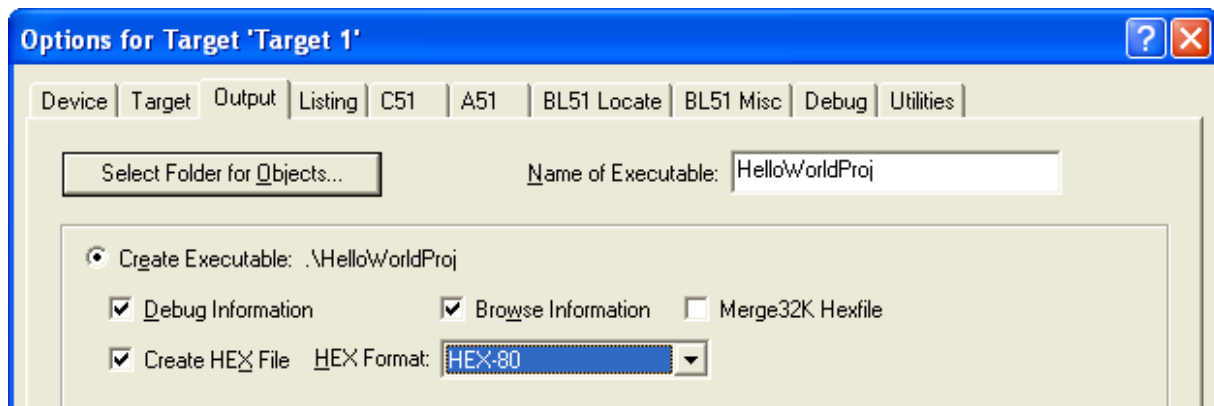
Unter dem Punkt **Project** → **Options for Target** öffnet sich ein Dialogfenster zur Parametrisierung des Projektes.



Im Tab **Target** wird standardmäßig eine Oszillatorfrequenz von 24MHz verwendet. Das Evaluationsboard wird hingegen mit einem **20MHz-Quarz** betrieben. Daher sollte der Parameter **Xtal** auf **20** abgeändert werden.



Im Reiter **Output** wird durch Anhängen der Option **Create HEX File** die Kompilierung des Quellcodes in ein Hex-File festgelegt.



Die Änderungen werden mit einem Klick auf **OK** übernommen.

Durch Kompilieren des Projektes (Funktionstaste **F7**) wird im Projektverzeichnis nun auch ein Hex-File mit dem Namen des Projektes (Bsp.: HelloWorldProj.hex) erzeugt.

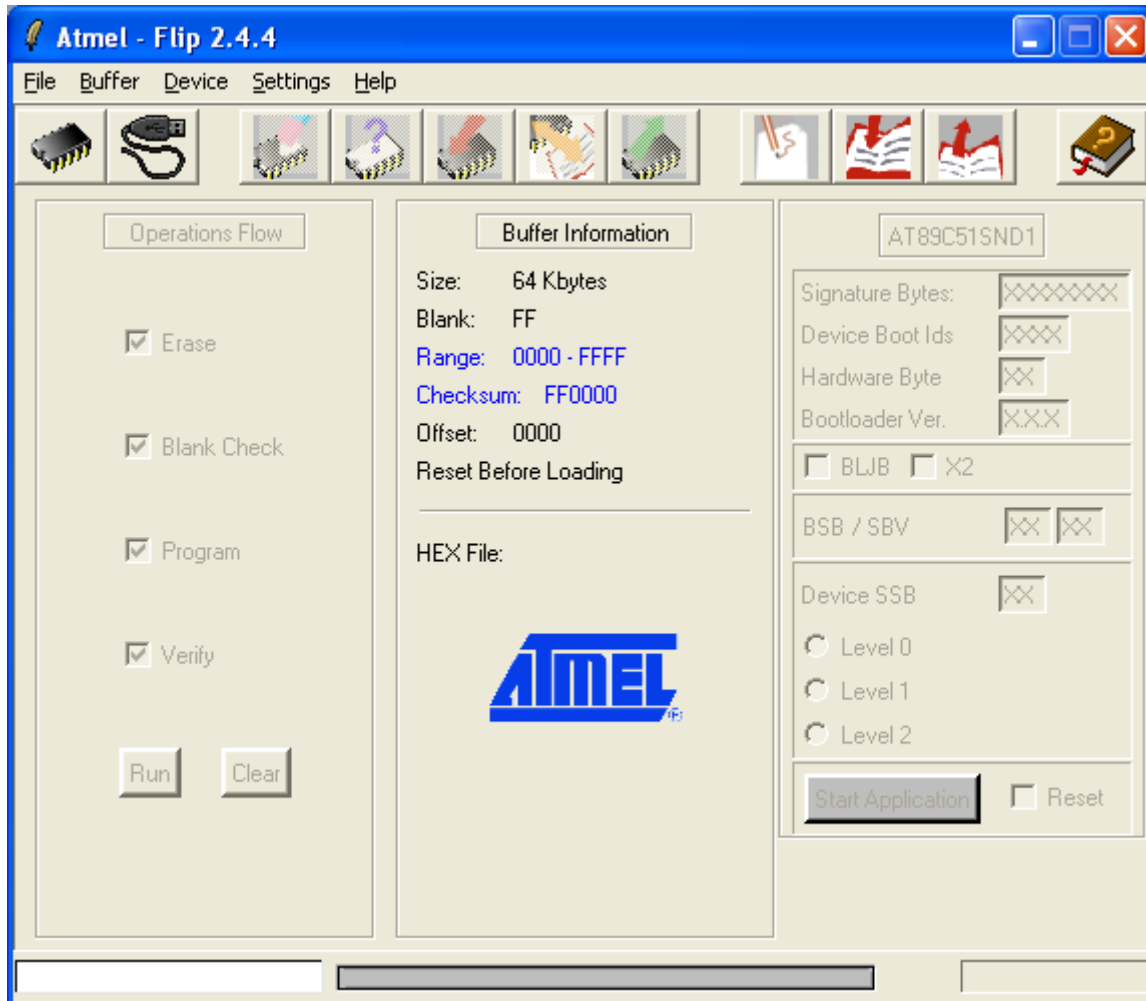
Dies kann nun in Folge mit Hilfe der Anwendung **Atmel FLIP** auf den Mikrokontroller gespielt werden.

ATMEL FLIP

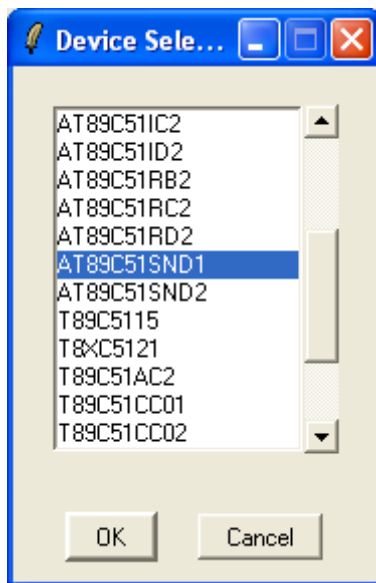


FLIP 2.4.4

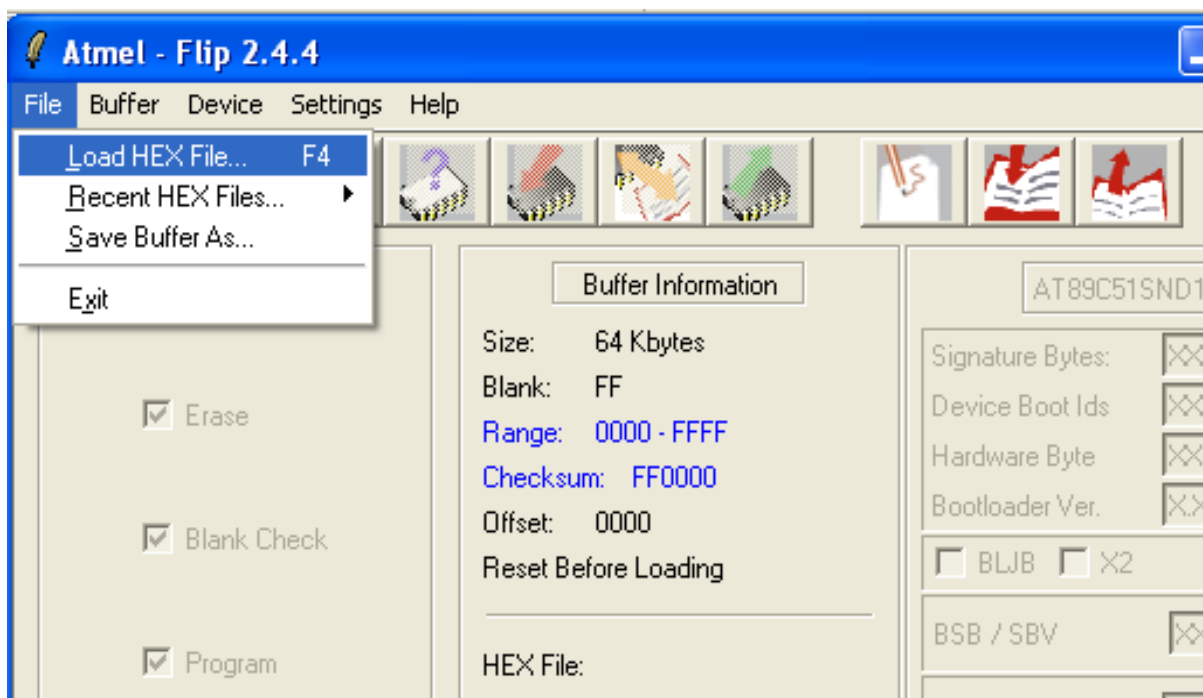
Zunächst muss angegeben werden, um welchen Prozessortyp es sich handelt. Dies geschieht durch einen Klick auf das gekennzeichnete Symbol:



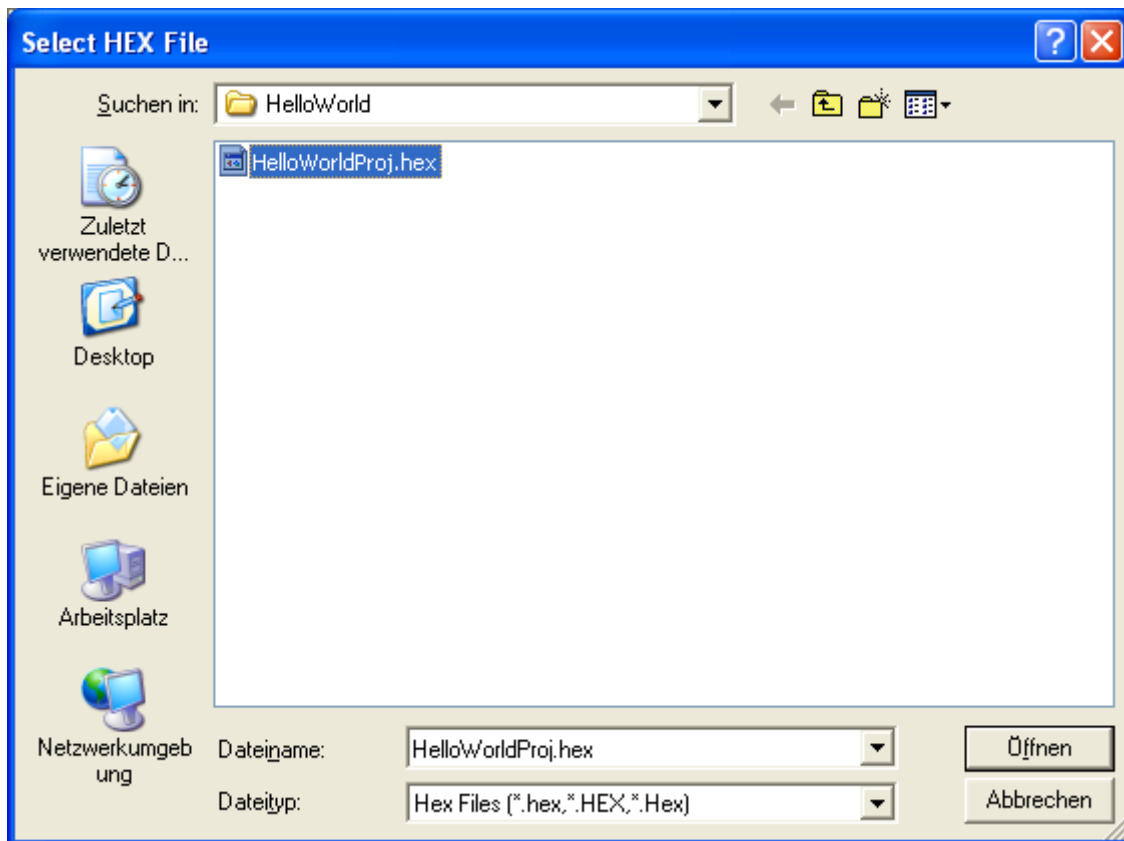
Hierbei ist der Typ **AT89C51SND1** auszuwählen:



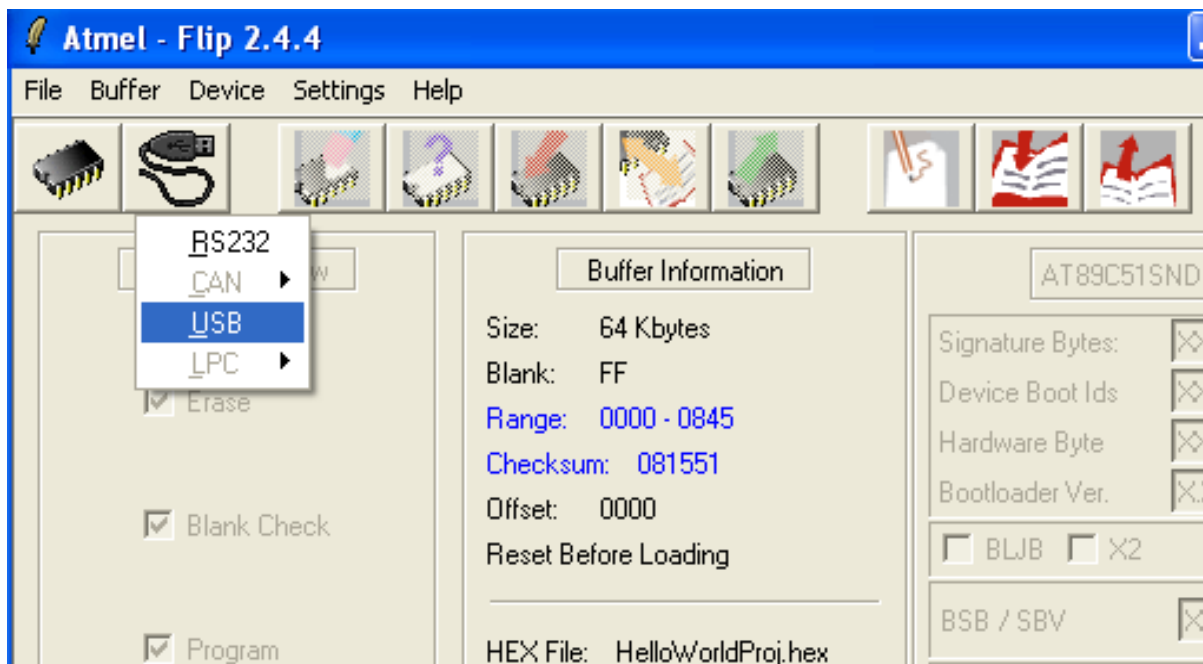
Im nächsten Schritt wird das zuvor erstellte Hex-File geladen:



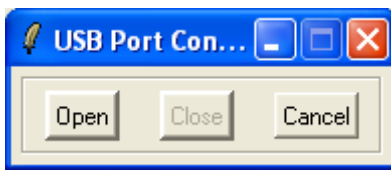
Im Dialogfenster das entsprechende Hex-File auswählen:



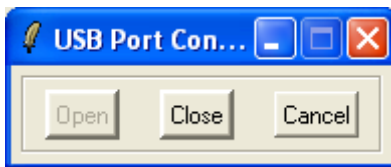
Der Prozessortyp und das zu überspielende Programm sind somit festgelegt. Im Folgenden muss eine Verbindung zum Evaluationsboard hergestellt werden. Die Verbindung zum Mikrocontroller erfolgt mittels **USB**. Das Evaluationsboard muss dabei mit einem USB-Kabel mit dem Computer verbunden sein, das Board spannungsversorgt und eingeschaltet sein und der Schalter für den Bootloader muss sich in der rechten Stellung befinden.



Über das USB-Dialogfenster kann der Verbindungsstatus gesteuert werden:

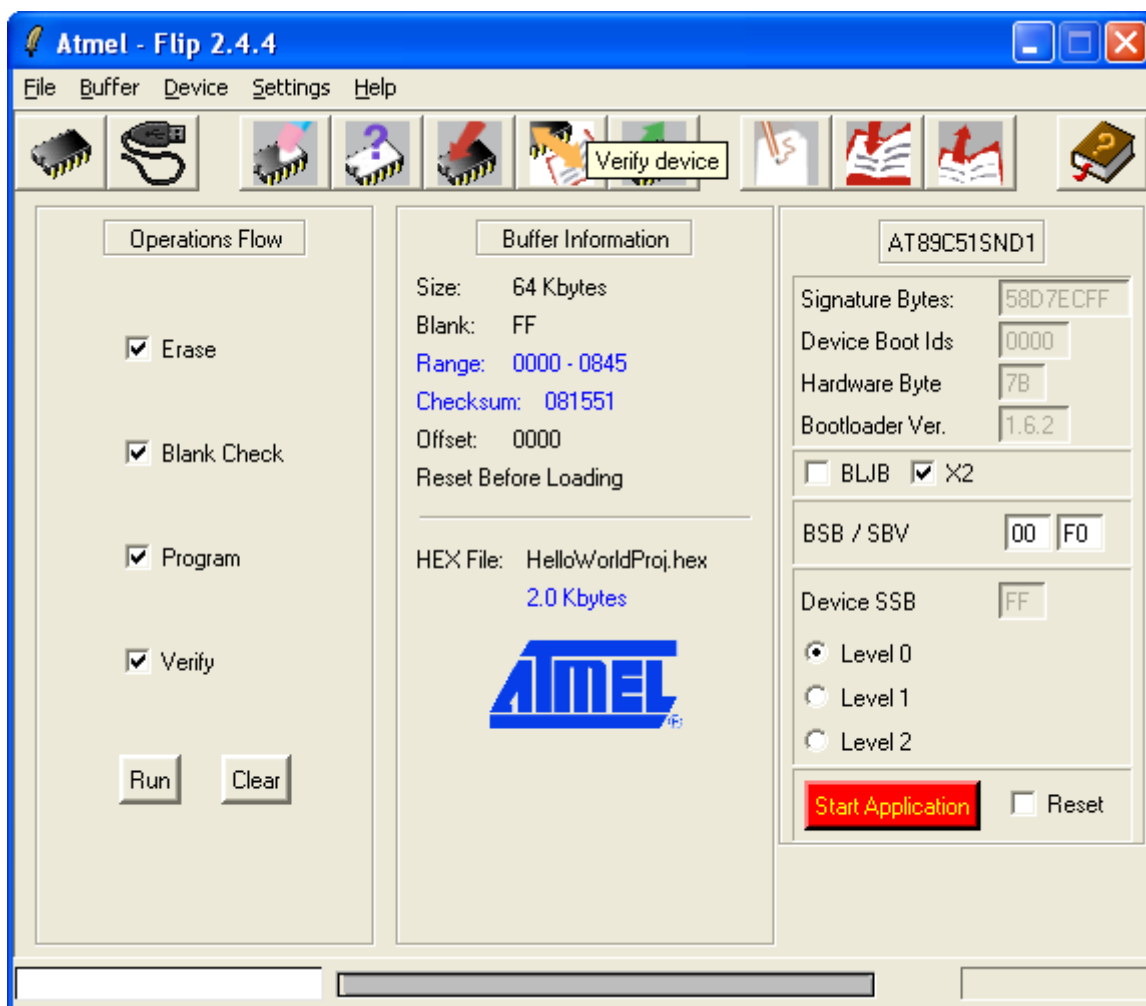


Durch Klick auf **Open** wird die Verbindung hergestellt:



Dadurch werden im Hauptfenster weitere Optionen freigeschaltet. Durch Klick auf **Run** wird der Programmspeicher gelöscht, der geleerte Programmspeicher verifiziert, das Programm überspielt und das überspielte Programm verifiziert.

Durch Klick auf den Button **Start Application** wird das am Evaluationsboard befindliche Programm gestartet.



Durch Starten der Applikation am Mikrokontroller verliert der PC die Verbindung zum Evaluationsboard, da das Bootloaderprogramm nicht mehr aktiv ist.

Durch Drücken des roten Knopfes am Evaluationsboard erfolgt ein Reset und der Bootloader ist wiederum aktiv, um mit dem Programm Atmel FLIP eine Verbindung aufzubauen. Dadurch kann ein neues Programm überspielt werden bzw. das aktuelle Programm neu gestartet werden.

Befindet sich der Schalter für den Bootloader am Evaluationsboard hingegen auf der linken Stellung, so ist das Bootloaderprogramm generell deaktiviert und der aktuelle Programmcode wird abgearbeitet. Durch Drücken des Reset-Knopfes kann dabei die Applikation (neu) gestartet werden.