Die Programmierung eines 8051ers (am Beispiel des CC03ers) unter µC/51

V 1.0 / 07.03.2006 / LGE

Zur Programmierung eines 8051er-Mikrocontrollers mit Hilfe der Programmiersprache ´C´ benötigen Sie, **rein softwaremäßig**, drei verschiedene Programme:

1) Die Entwicklungsumgebung (IDE ≡ Integrated Development Environment) ***µC/51*** der Firma Wickenhäuser Elektrotechnik, mit der u.a. das ´C´-Programm komplett geschrieben, verbessert und natürlich auch erweitert werden kann.
   Sie können hierbei die kostenlose Demo-Version (max. 8 kByte Umfang für das 8051er-Programm) oder die kostenpflichtige Vollversion (64 kByte max. Programmumfang für den 8051er) von µC/51 verwenden.

2) Das Download ´***Flip***´, mit dem Sie das endgültig erzeugte 8051er-Intel-HEX-File von Ihrem Entwicklungs-PC in Ihr 8051er-Zielsystem transferieren.
   Flip ist kostenlos von der Homepage der Firma Atmel download-bar.

3) Das Terminal-Programm ´***HyperTerm***´, das zur Kommunikation mit dem 8051er-Zielsystem über eine der seriellen COM-Schnittstellen des Entwicklungs-PC´s dient.
   Mit Hilfe von HyperTerm können Sie somit Ihr 8051er-Programm ausführlich austesten und verbessern.
   HyperTerm ist Bestandteil jeder Windows-Betriebssystem-Umgebung und ist daher ebenfalls kostenfrei verfügbar.
   Falls es nicht bei der Windows-Grundinstallation auf Ihrem PC mit installiert wurde, so kann es einfach nachinstalliert werden.

Und nun im Detail:

Möchten Sie mit Hilfe der ***IDE µC/51*** ein lauffähiges ´C´-Programm für Ihr 8051er-Mikrocontroller-System erstellen und austesten, so sind dazu im wesentlichen die nachfolgenden **sechs einfachen Schritte** notwendig:

1) Erstellung einer geeigneten **Arbeitsumgebung** in der die Randbedingungen für das neue 8051er-Projekt festgelegt werden, wie z.B. der Name des ´C´-Programms, das die Grundlage des Projektes bildet, das also den

gewünschten ´C´-Programmtext enthält, die Startadressen des Programm- und des Datenspeichers für das zu Grunde liegende 8051er-Zielsystem, etc.
Solch eine Arbeits- bzw. Projektumgebung wird unter µC/51 auch ´**Workspace**´ genannt und die darin enthaltenen projektspezifischen Grundfestlegungen werden vorab mit Hilfe des Programms ´**MakeWiz**´ (Teil von µC/51) definiert und dann automatisch in zwei Dateien abgelegt.

2) Nachdem Sie Ihren Workspace (also Ihre Arbeitsumgbebung) festgelegt haben, wird dieser in den ´**C´-Programm-Text-Editor** namens ´**jfe**´ geladen und Sie können beginnen, Ihren gewünschten C-Quell-Text einzugeben.

3) Vom Editor ´jfe´ aus können Sie dann unmittelbar den ´**C´-Compiler** und den ´**C´-Linker** mit einem ´Klick´ aufrufen, sich also Ihr Programm übersetzten lassen und alle (syntaktischen) Fehler korrigieren.

4) Sind alle Fehler beseitigt, so erhalten Sie als Endergebnis den **downloadbaren Intel-HEX-File** für Ihren Mikrocontroller bzw. für Ihr Mikrocontroller-System.

5) Mit Hilfe des Download-Programms ´**Flip**´ transferieren Sie jetzt den Intel-HEX-File von Ihren Entwicklungs-PC auf das Mikrocontroller-Ziel-System.

6) Mit Hilfe des Terminal-Programms ´**HyperTerm**´ können Sie nun abschließend Ihr 8051er-Programm austesten und verbessern.

Diese einzelnen Aktionen werden wir Ihnen nachfolgend ausführlich darlegen und beschreiben.

1. Die Schaffung der Projekt- bzw. Arbeitsumgebung, also des projektspezifischen Workspaces, für Ihr neues 8051er-Projekt mit Hilfe von ´MakeWiz´.

Die Erstellung eines lauffähigen 8051er-´C´-Programms unter der **IDE µC/51** läuft, wie allgemein bei IDE´s üblich, projektorientiert ab, d.h. man muß sich zu aller erst eine geeignete, passende **Projektumgebung** (**Workspace**) schaffen, in der man dann alle nachfolgenden Arbeiten ausführt, in der man also sein ´C´-Programm erstellt, es übersetzt und es solange korrigiert bis es fehlerfrei ist und die gewünschten Aufgaben erfüllt.
Dieses Workspace besteht unter µC/51 im wesentlichen aus **drei Elementen**:

1) dem eigenen, **selbst geschriebenen ´C´-Programm**,
2) der ***.mak-Datei** mit Anweisungen zur Steuerung des Compilierungs- und Link-Prozesses,
3) der ***.wsp-Datei** mit allen weiteren notwendigen Informationen zum Workspace für dieses Projektes.

Nachfolgend werden wir zuerst alle notwendigen Schritte zur Erstellung eines µC/51er-Workspace ausführlich erläutern, denn dieses Workspace dient später als Grundlage für die Erstellung, die Compilierung und die Linkung des eigenen C-Programms.

1. Das selbst geschriebene ´C´-Programm für den 8051er

bildet natürlich die Basis für das **8051er-Workspace** und zu dessen Erstellung empfehlen wir die folgende Vorgehensweise:
Wenn Sie µC/51 ordnungsgemäß auf Ihrem Rechner installiert haben, so befindet sich i.a. auf dem Laufwerk C das **Verzeichnis ´uC51´** mit seinen verschiedenen Unterverzeichnissen:

```
uC51
 |--------bin
 |--------docu
```

|--------include
|--------lib
|--------src
|--------uninst

In diesem uC51-Verzeichnis erzeugen Sie sich nun mit Hilfe des Windows-Explorers ein neues Verzeichnis für Ihre eigenen Seminar-´C´-Projekte, das z.B. ´**Seminar**´ heißt (der Name kann beliebig gewählt werden, achten Sie nur darauf, das keine Leerzeichen im Namen enthalten sind, denn das verträgt µC/51 nicht).
Die neue Verzeichnisstruktur sieht dann wie folgt aus:

uC51
|--------bin
|--------docu
|--------include
|--------lib
|--------**Seminar**
|--------src
|--------uninst

In diesem Verzeichnis werden nun alle Ihre neuen Projekte ablegt.
Kopieren Sie sich daher zunächst von unserer CD aus dem Verzeichnis ´**Seminar-Programme**´ die noch leere ´C´-Programm-Datei ´**leer.c**´ in das neu angelegte Verzeichnis ´**Seminar**´.
Dieses ´C´-File bildet jetzt den Ausgangspunkt für alle unsere weiteren Programmierungen.
Wenn Sie daher jetzt ein neues Projekt beginnen, so erstellen Sie sich von der Datei ´leer.c´ eine weitere Kopie und benennen diese dann nach dem neuen Projekt, also z.B. Umbennung der Kopie von ´leer.c´ in ´**prog_1.c**´ für unser erstes Projekt bzw. Programm.
Das sollten Sie jetzt schon einmal durchführen.

2. Die Anweisungen zur automatischen Steuerung des Compilierungs- und des Link-Prozesses (die *.mak-Datei)

Als nächstes müssen Sie sich Ihre individuelle Projektumgebung, d.h. Ihr notwendiges ´**Workspace**´, schaffen.
In einem ersten Schritt sind dazu z.B. Festlegungen zu treffen, ab welcher Adresse der Programm- und der Datenspeicher in Ihrem 8051er-System beginnen soll, welche Treiberroutinen für die serielle Schnittstelle des 8051ers verwendet werden sollen, ob ein Intel-Hex-File erzeugt werden soll, etc..
Diese Projekt-spezifischen Einstellungen werden in einer besonderen Datei abgelegt, der so genannten **Make-Datei** (Datei-Endung *.mak) ≡ „Mache-Datei“ mit Regeln für den **´C´-Compiler und den ´C´-Linker** zur Erstellung des endgültigen, download-baren Intel HEX-Files.
Dieses *.mak-File wird einmalig zu Beginn eines neuen Projektes zusammengestellt und bildet danach die Grundlage für die gesamten Übersetzungs- und Link-Arbeiten mit bzw. zu diesem Projekt (´*´ steht hier für den frei wählbaren Namen des Mak-Files und muß nicht identisch mit dem Namen des ´C´-Programms sein).

Zur einfachen und benutzerfreundlichen Erstellung dieser *.mak-Datei gibt es unter µC/51 ein **Hilfsprogramm** namens ´**MakeWiz**´, in dem der Anwender einfach seine Festlegungen eintragen kann (Wiz ≡ Wizard ≡ Zauberer).

Bevor wir nun aber ´MakeWiz´ aufrufen, müssen wir noch den nächsten Punkt betrachten:

3. Die restlichen Festlegungen für den Projekt-spezifischen Workspace (die *.wsp-Datei)

´MakeWiz´ erzeugt nämlich, neben der ´*.mak-Datei´, noch eine weitere Datei mit der Endung ´***.wsp**´ (wsp ≡ Workspace), in der nun **zusammen gefaßt alle restlichen Festlegungen** für die komplette Arbeitungsumgebung zum aktuellen Projekt enthalten sind.
Durch die spätere Abarbeitung dieses *.wsp-File

- wird z.B. bei der Programmübersetzung automatisch das zuvor erstellte *.mak-File zur Steuerung des ´C´-Compilers und des ´C´-Linkers aufgerufen und abgearbeitet,

- wird, bei Fehlerfreiheit, das endgültige download-bare Intel-HEX-File für den 8051er erstellt,

- werden die Festlegungen zur rein optischen Gestaltung des ´C´-Editors ausgeführt (s. nachfolgend)

- etc.

SEHR WICHTIG:
Dieses von ´MakeWiz´ erstellte ´*.wsp-File´ (Workspace-File) bildet die Grundlage für alle Einstellungen und Festlegungen zu Ihrem Projekt und das bedeutet:
Wenn ein Projekt mit Hilfe des ´C´-Editors (weiter)bearbeitet werden soll (s. nachfolgend), so muß als Erstes das zu dem Projekt zugehörige *.wsp-File in den Editor geladen werden !

Durch diese Aktion wird dann nämlich bereits automatisch das richtige ´C´-File geöffnet, die richtige *.mak-Datei bei der Programmübersetzung gestartet und somit das gesamte Projekt korrekt bearbeitet.

Wird gar keine oder die falsche *.wsp-Datei im Editor geöffnet, so kann das Projekt nicht korrekt abgearbeitet werden !

Kommen wir daher nun zur Erstellung der grundlegenden ´*.wsp-´ und ´*.mak-Dateien´ mit Hilfe des Programms ´**MakeWiz**´:
´MakeWiz´ ist ein Teil der IDE µC/51 und wird ganz einfach durch Anklicken des zugehörigen Ikons aufgerufen, **Abb.1**:

Abb.1: Das Aufruf-Ikon zu ´MakeWiz´

Es erscheint das Übersichtsfenster von ´MakeWiz´, **Abb.2**:

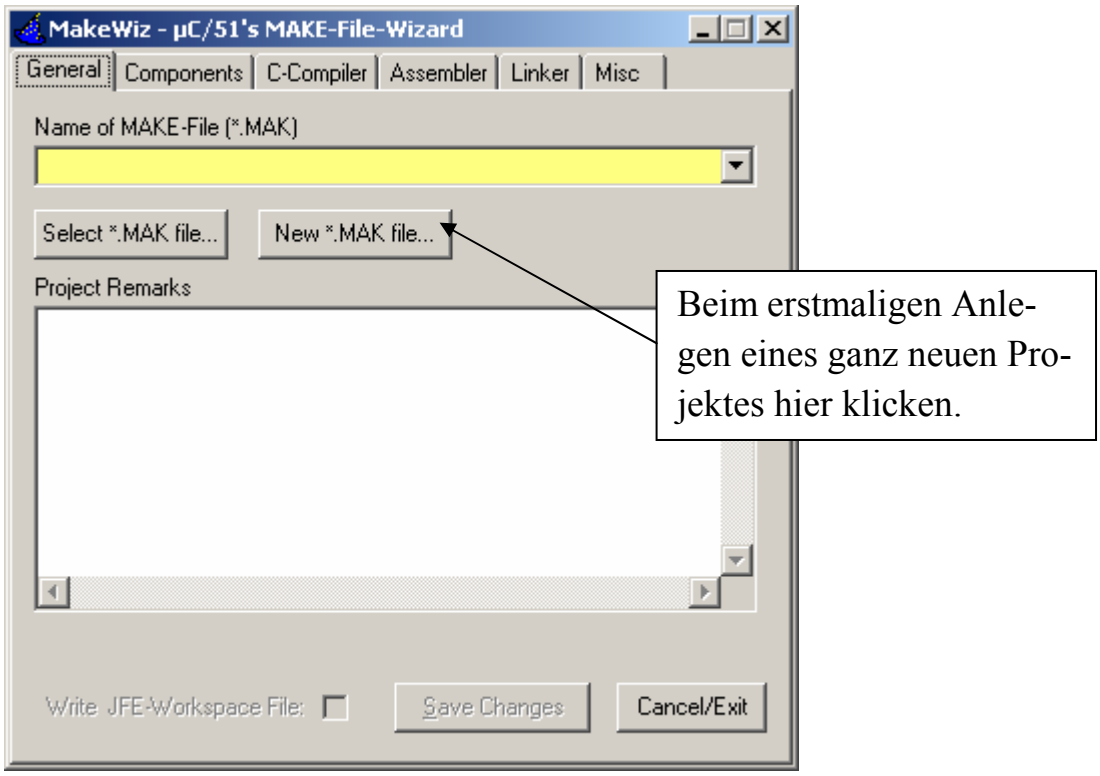

Abb.2: Das Übersichtsfenster von ´MakeWiz´

Da wir nun **erstmalig** ein ganz neues Projekt eröffnen, müssen Sie auf ´**New *.MAK file...**´ klicken.
Es erscheint ein Auswahlmenü, in dem Sie eintragen müssen, wie das neue Projekt bzw. die neue Projektumgebung (der neue Workspace) heißen soll und wo (in welchem Verzeichnis) die Projekt-Informationen abgelegt werden sollen, **Abb.3**:

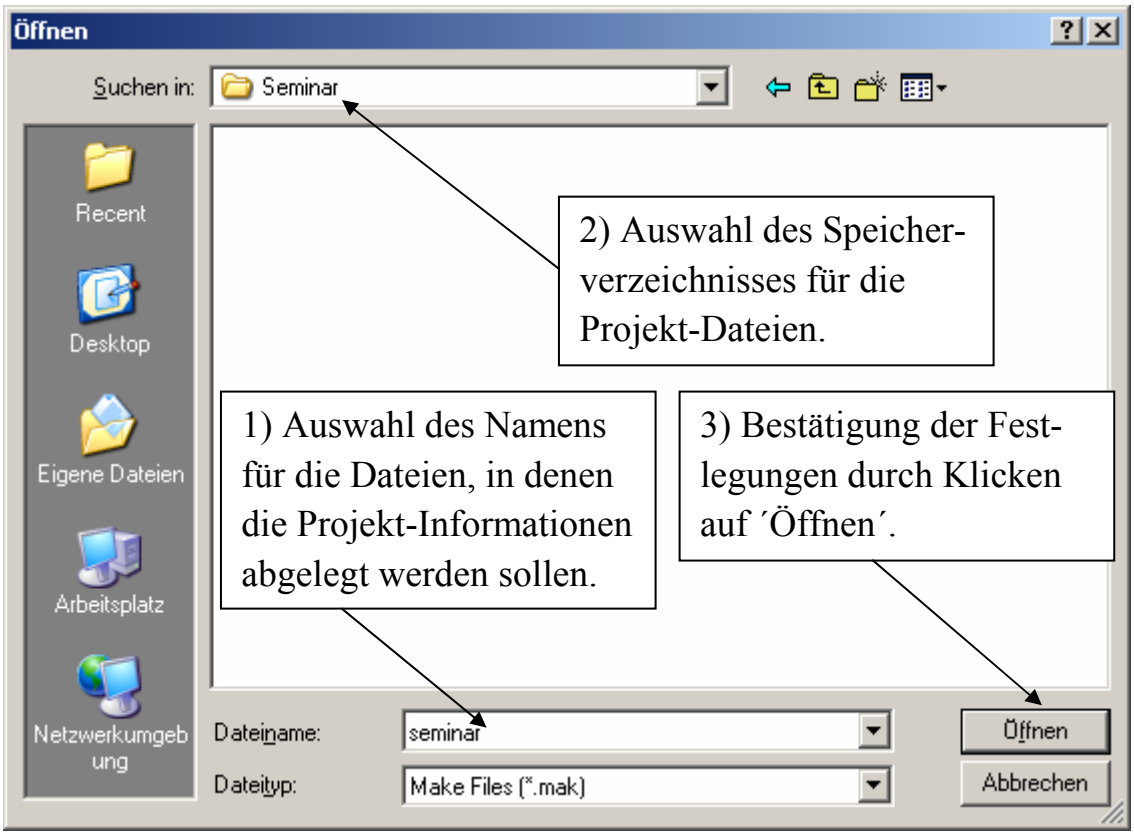

Abb.3: Die Auswahl von Projektverzeichnis und Projektnamen

Wir wählen hier:

1) Als (Vor)Name der Dateien, in denen die Projekt-Informationen gespeichert werden sollen: ´**seminar**´.
´MakeWiz´ legt darauf hin später die wichtigen Dateien ´**seminar.mak**´ und ´**seminar.wsp**´ an (s. Beschreibungen zuvor).

2) Als Speicherort für alle Projektdateien wählen wir das neu angelegte Verzeichnis ´**Seminar**´ unter µC/51 aus.

Bestätigen Sie nun die Festlegungen durch Klicken auf den Button ´Öffnen´. ´MakeWiz´ meldet jetzt, daß die zu erzeugende *.mak-Datei (hier: seminar.mak) noch nicht existiert und fragt daher, ob sie neu erstellt werden soll, **Abb.4**:

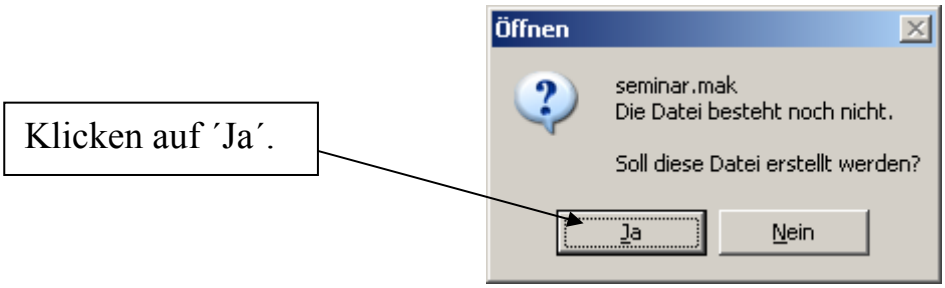

Abb.4: Die Neuerstellung der Datei seminar.mak

Diese Frage beantworten Sie durch klicken auf ´Ja´.
Es erscheint wieder das Grundfenster von ´MakeWiz´, aber: es wurde nun die Datei ´**seminar.mak**´ angelegt und im Text-Fenster ´**Projekt Remarks**´ sind schon einige Kommentartexte automatisch eingetragen worden, **Abb.5**:

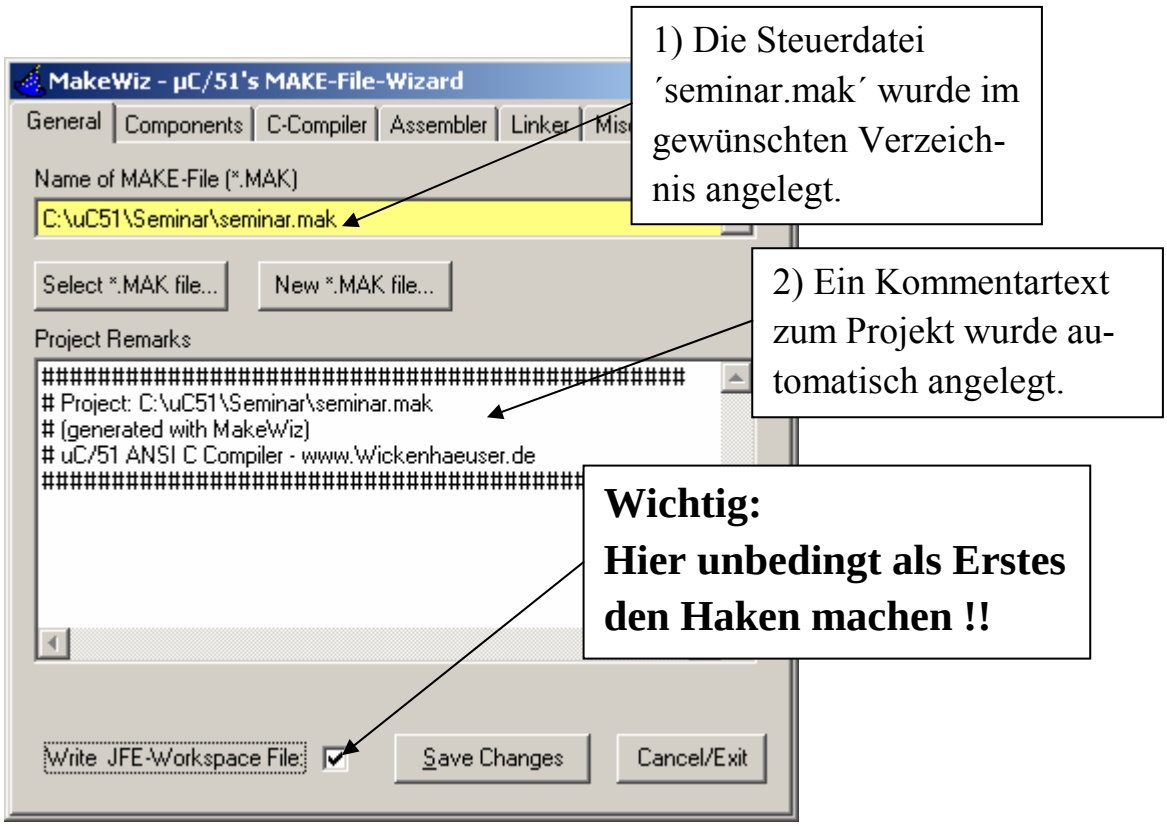

Abb.5: Die Eintragungen der ersten Projektdaten

Diese Texteintragungen geben einige Auskünfte über das erzeugte Projekt und werden später automatisch von ´MakeWiz´ als Kopf-Kommentarzeilen in die Datei ´**seminar.mak**´ eingetragen.
Sie können diesen Text durch Anklicken mit der Maus jederzeit beliebig Ihren eigenen Bedürfnissen und Vorstellungen anpassen, aber zuvor:

EXTREM WICHTIG:

Setzen Sie unbedingt, bevor Sie irgend etwas anderes machen, einen Haken in das Kästchen bei:

´Write JFE-Workspace File´.

Nur wenn Sie diese Schreib-Option auswählen, werden alle nachfolgenden Änderungen auch wirklich in Ihre Projekt-Workspace-Datei ´*.wsp´, also in die ´seminar.wsp-Datei´ mit übernommen. Wenn Sie diesen Haken vergessen, können Sie zwar beliebige Änderungen machen, diese werden aber nicht in die ´*.wsp-Datei´ mit übernommen und Sie wundern sich später, warum Ihr Projekt nicht ordnungsgemäß bearbeitet bzw. übersetzt wird.

Dieser Fehler wird gerade am Anfang sehr häufig gemacht !

Nachdem Sie den Haken gesetzt haben wird auch das Feld ´**Save Changes**´ aktiv geschaltet.

Nun müssen noch einige Eintragungen in die verschiedenen Register-Karten von ´MakWiz´ gemacht werden und achten Sie bei allen Änderungen darauf, daß ´**Write JFE-Workspace File**´ angehakt ist und es auch bleibt !
Die nachfolgenden Eintragungen stellen nun letztendlich Ihre individuellen, Projekt-spezifischen Festlegungen dar.

Eintragungen unter ´Components´, Abb.6:

Hier tragen Sie ein, welches ´C´-Source-File zu Ihrem Projekt gehört, m.a.W. welches ´C´-File später geöffnet und bearbeitet werden soll..
Bei unserem ersten Projekt ist das die ´C´-Datei ´**prog_1.c**´ (s. Punkt 1)).

Klicken Sie daher jetzt ´**Add Source File...**´ an und wählen Sie im nachfolgenden Auswahlmenü unsere Datei ´**prog_1.c**´ aus dem Verzeichnis ´**Seminar**´ aus. Bestätigen Sie die Auswahl durch Klicken auf ´**Öffnen**´, **Abb.7**:

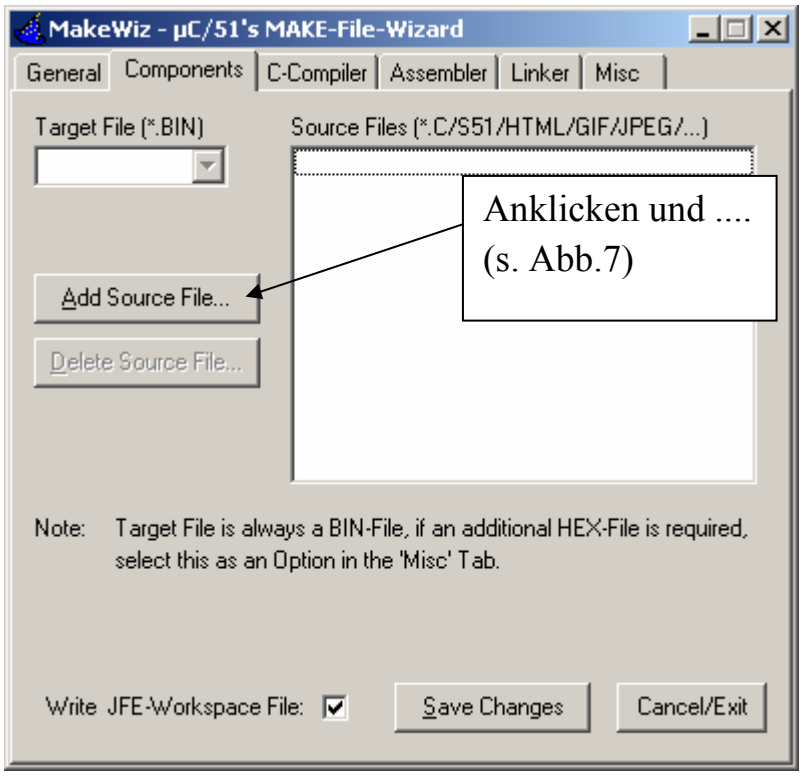

Abb.6: Eintragungen unter ´Components´

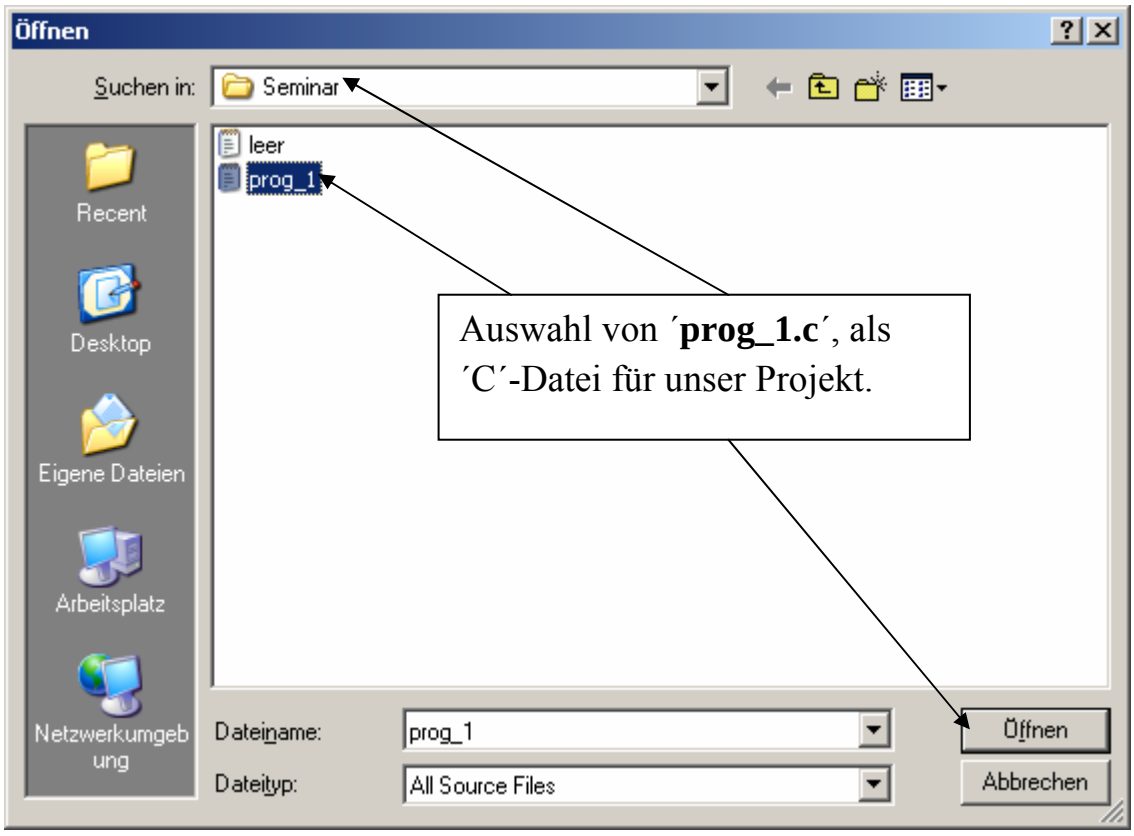

Abb.7: Auswahl von prog_1.c´

Weiterhin müssen Sie noch angeben, wie das Ziel(Target)-File heißt, wie also das endgültige Intel-HEX-File heißt, das Sie später auf den Mikrocontroller herunter laden.
Der Name dieses Intel-HEX-Files ist i.a. identisch mit dem Namen des ´C´-Files, das ja die Grundlage Ihres Projektes bildet.
Also klicken Sie den Auswahl-Kasten ´**Target File (*.BIN)**´ an und wählen dann als Namen ´**prog_1**´ aus, **Abb.8**:

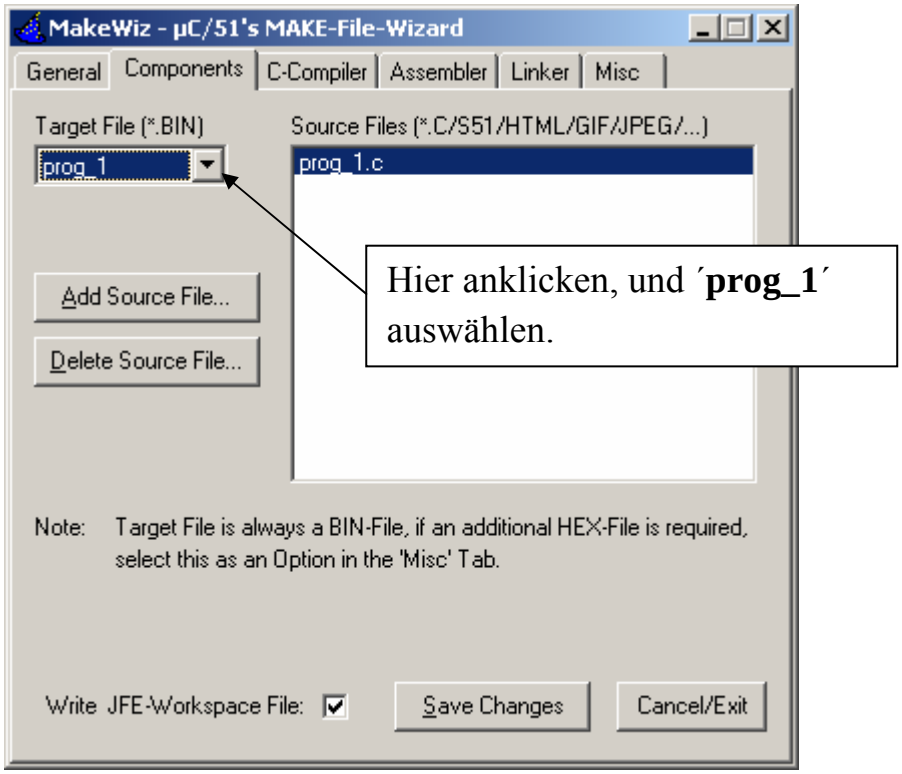

Abb.8: Die Auswahl des Target(Ziel)-Namens

Hinweise:

1) Hier steht zwar als Datei-Ergänzung ´*.BIN´, die Auswahl des Target-Namens bezieht sich aber auf alle bei der Compilierung und bei der Linkung entstehenden (Zwischen)Datei-Namen, also auch auf den Namen des Intel-HEX-Files.

2) Vergessen Sie die Auswahl des ´Target-Files´, so erscheint später, wenn Sie Ihre Änderungen/Eingaben abspeichern wollen, immer folgende Fehlermeldung, **Abb.9**:

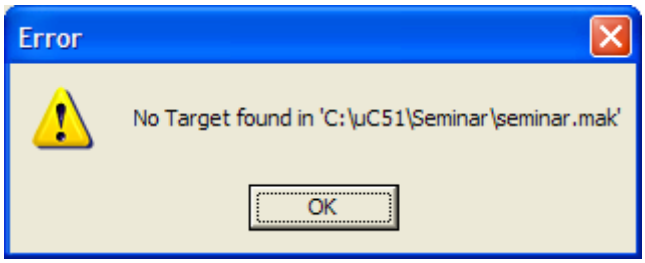

Abb.9: Fehlermeldung, wenn kein ´Target File´ ausgewählt wurde

Damit sind die Eintragungen bei den ´Components´ beendet und Sie wählen die nächste Register-Karte aus:

Eintragungen unter ´C-Compiler´, Abb.10:

Hier ist zur Zeit nur eine einzige Eintragung notwendig:
Die Einstellung der CPU-Geschwindigkeit (≡ CPU-Speed ≡ Maschinenzyklus-Zeit des 8051er) in ns, wobei gilt:

$$\textbf{Maschinenzyklus-Zeit} = 12 * (1 / f_q)$$

mit f_q ≡ Frequenz des externen 8051er-Quarzes.

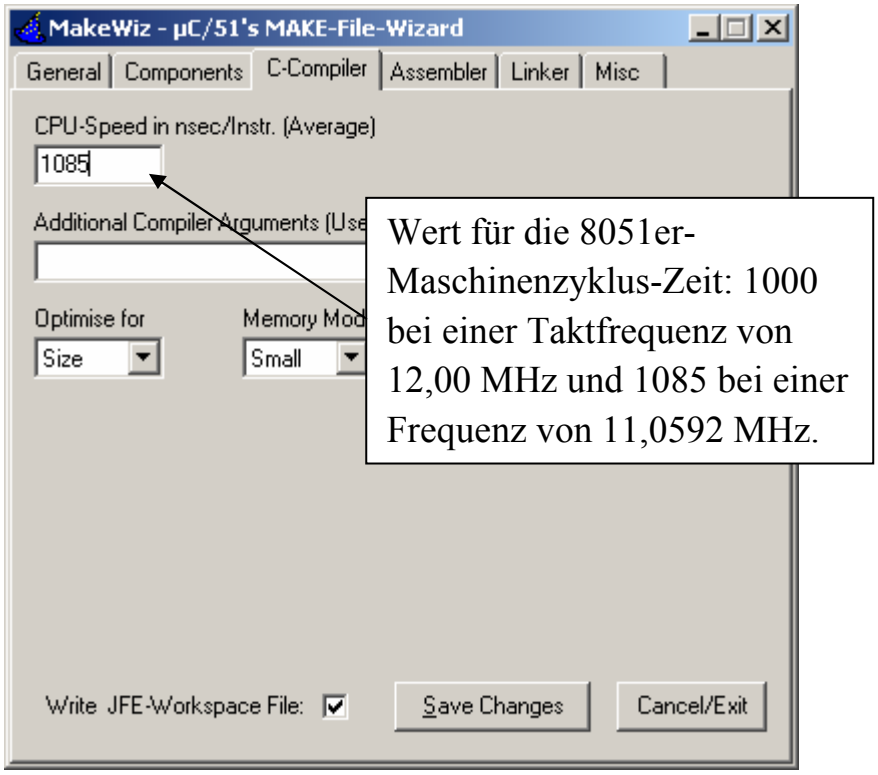

Abb.10: Eintragungen unter ´C-Compiler´

Bei einer Quarz-Frequenz von 12,00 MHz ergibt sich somit der Wert von 1 µs = 1000 ns und bei einer Taktfrequenz von 11,0592 MHz ein Wert von 1,085 µs = 1085 ns.
Für den Betrieb des CC03ers im TFH-Board ONE tragen wir daher den letzt genannten Wert ein.

Die anderen Auswahlmöglichkeiten auf dieser Register-Karte beziehen sich auf die Optimierungsarbeiten des ´C´-Compilers und des ´C´-Linkers und auf das verwendetet (Daten)Speicher-Modell.
Wir lassen diese Einstellungen so wie vorgegeben und wählen als letztes die Register-Karte ´**Misc**´ aus (auf den Register-Karten ´**Assembler**´ und ´**Linker**´ sind zur Zeit keine weiteren Einstellungen bzw. Eintragungen notwendig).

Eintragungen unter ´Misc´, Abb.11:

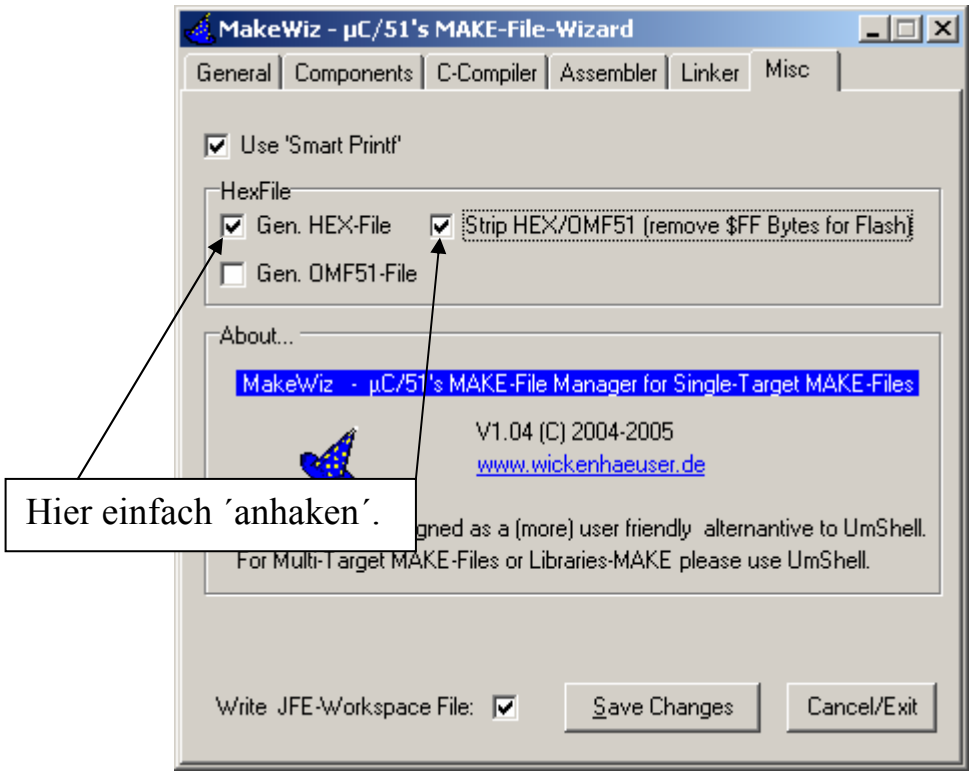

Abb.11: Eintragungen unter ´Misc´

Auf dieser Register-Karte sind lediglich zwei Haken notwendig, die dafür sorgen, daß der Linker als Abschluß seiner Arbeit auch wirklich einen download-baren Intel-HEX-File erzeugt.

Damit haben Sie alle Workspace-Einstellungen für Ihre Projektumgebung erledigt und können diese jetzt durch Anklicken von ´**Save Changes**´ abspeichern, **Abb.12**:

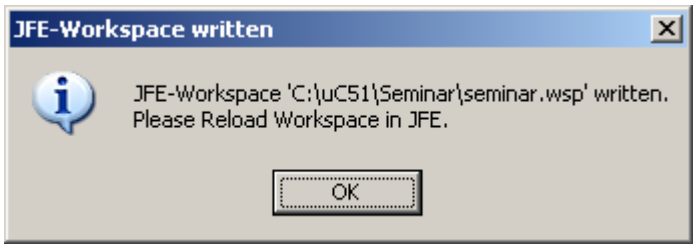

Abb.12: Die Speicherung der Workspace-Einstellungen

Damit ist nun die Arbeit mit ´MakeWiz´ beendet und Sie können das zugehörige Programm(fenster) schließen (´MakeWiz´ also beenden).
Wenn Sie sich jetzt den Inhalt des Verzeichnisses ´**Seminar**´ ansehen, so erkennen Sie, daß ´MakeWiz´ die zwei notwendigen Projektdateien automatisch angelegt hat: einmal ´**seminar.mak**´ und einmal ´**seminar.wsp**´.
Das war´s schon und nun können wir uns die Erstellung und die Compilierung des eigentlichen ´C´-Source-Files näher ansehen.
Aber zunächst noch ein

WICHTIGER HINWEIS FÜR DIE TÄGLICHE PRAXIS:
Wenn Sie nun ein weiteres neues Projekt anlegen / bearbeiten wollen, d.h. wenn Sie ganz einfach ein anderes neues ´C´-Programm schreiben wollen, so brauchen Sie die vorherigen Schritte nicht immer wieder neu durchzuführen, sondern es reicht aus, wenn Sie

1) sich das neue ´C´-Source-File erzeugen : entweder durch Kopieren und Umbenennung eines alten ´C´-Files oder durch Verwendung unseres ´C´-Leer-Files

und

2) wenn Sie in der Register-Karte ´**Componentes**´ (s. Abb.8) das alte Source-File löschen (markieren und dann ´**Delete Source File...**´anklicken) und danach das neue Source-File einfach einbinden: ´**Add Source File...**´ anklicken, File aussuchen, markieren und öffnen.
Vergessen Sie aber nur nicht die Eintragungen im Auswahl-Kasten ´**Target File (*.BIN)´ und den Haken bei ´Write JFE-Workspace File´.**
In **Abb.13** sind die vorzunehmenden Änderungen noch einmal dargestellt: das neue ´C´-Source-File (Projekt) heißt hier ´**prog_2.c**´.
Danach klicken Sie einfach auf ´**Save Changes**´ und haben damit Ihre neue Projektumgebung fertig gestellt.
Mehr Änderungen sind i.a. nicht notwendig, da alle anderen Eintragungen erhalten bleiben und übernommen werden können.

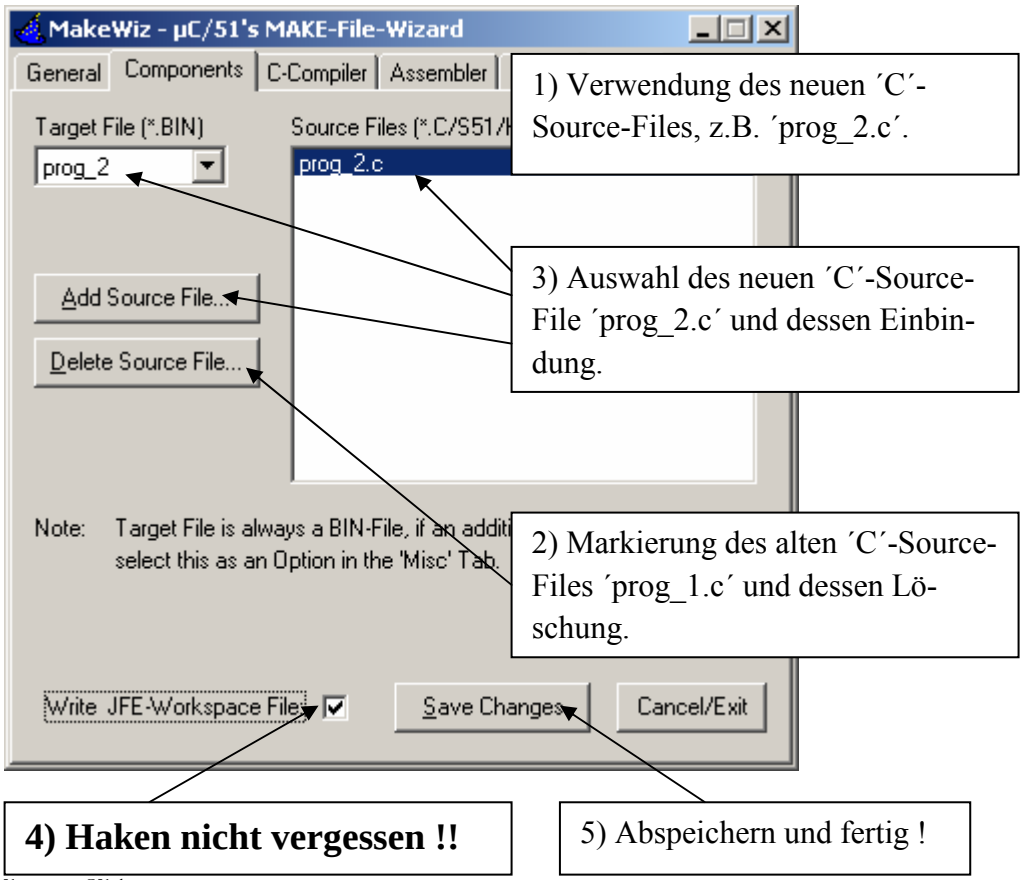

Abb.13: Die Einrichtung eines neuen Projektes

2. Die Erstellung und die Änderung des eigentlichen ´C´-Programm-Textes mit Hilfe des Editor ´jfe´.

Als Editor zur Erstellung bzw. Eingabe und Korrektur eines ´C´-Quell-Textes kann im Prinzip jeder normale ASCII-Editor verwendet werden.
Der in der IDE µC/51 bereits enthaltene Editor „**Jens´ File Editor (jfe)**“ ist jedoch schon optimal auf die Bedürfnisse eines ´C´-Programmierers (unter µC/51) eingerichtet und das heißt:

- ´jfe´ bietet eine sehr hilfreiche Syntaxcolorierung bei der Erstellung des ´C´-Quell-Textes.
- ´jfe´ kann sofort die von ´MakeWiz´ erstellten beiden Projektdateien verarbeiten und somit kann die gesamt Projekt-Ablaufsteuerung von ´jfe´ aus erfolgen.
- Von ´jfe´ aus kann sofort der ´C´-Compiler und der ´C´-Linker ´auf Knopfdruck´ aufgerufen werden.
- Fehlermeldungen werden in einem Fenster innerhalb der ´jfe´-Arbeitsumgebung angezeigt und so kann optimal die Fehlerstelle gefunden und korrigiert werden.
- ´jfe´ ist somit die ideale Arbeitsoberfläche für die tägliche Programmierarbeit unter µC/51.

Zum Start von ´jfe´ klicken Sie einfach auf das zugehörige Ikon im uC51-Ordner, **Abb.14**:

JFE

Abb.14: Das Aufruf-Ikon zu ´jfe´

Es erscheint das Grundfenster von ´jfe´, das Sie zunächst einmal auf maximale Größe schalten sollten, **Abb.15**:

Abb.15: Das Grundfenster von ´jfe´

Als erstes müssen Sie nun Ihre Arbeitsumgebung, also Ihren zuvor angelegten ´**Workspace**´, in ´jfe´ einladen.
Das machen Sie, indem Sie auf die Register-Karte ´**File**´ klicken und dann ´**Open Workspace**´ anwählen, **Abb. 16**:

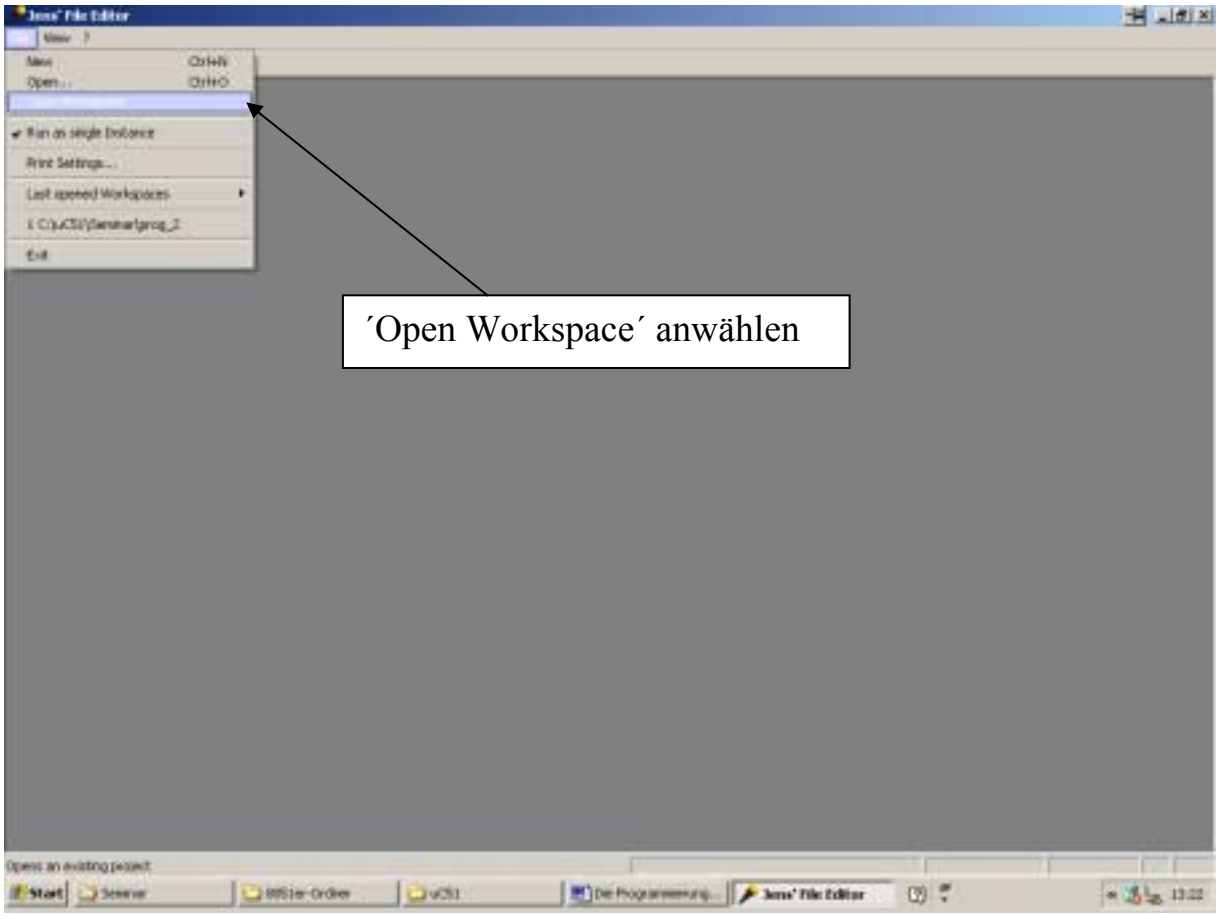

Abb.16: Die Auswahl eines Workspaces, I

Sie wählen nun den Workspace ´**seminar**´ aus dem Verzeichnis ´Seminar´ unter µC/51 aus und öffnen diesen, **Abb.17**:

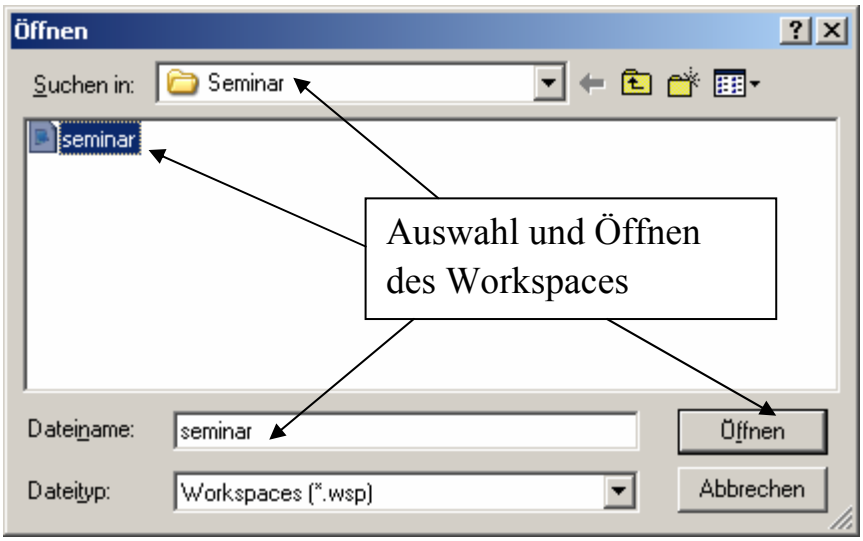

Abb.17: Die Auswahl eines Workspaces, II

Nach dem Klicken auf ´**Öffnen**´ erscheint dann sofort **Ihr komplettes Projekt** mit allen getätigten Einstellungen und dem eingebundenen zugehörigen ´C´-Source-File in der ´jfe´-Arbeitsumgebung, **Abb.18**:

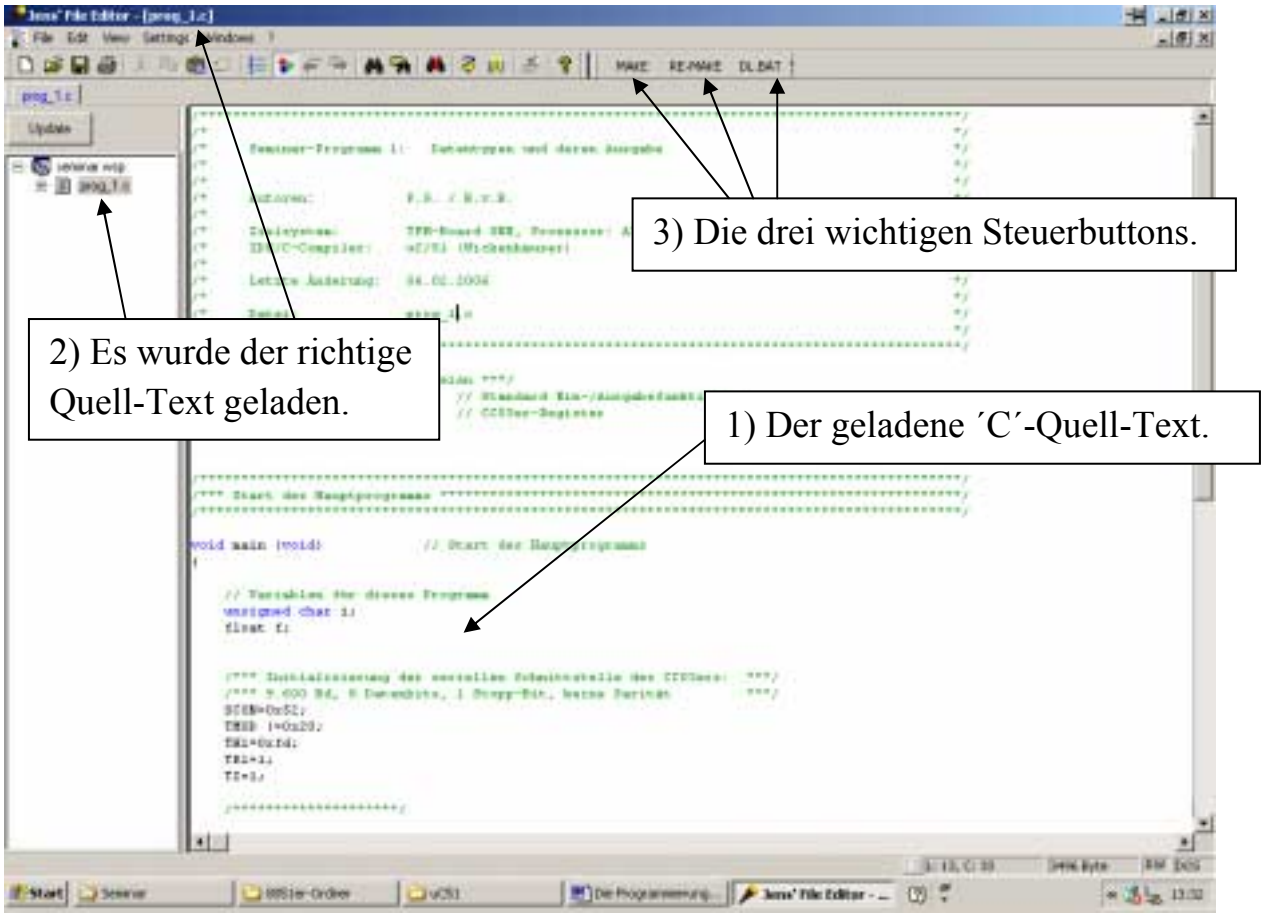

Abb.18: Das geladene Projekt (die geladene Workspace-Umgebung)

Wichtig sind hierbei, daß Sie die folgenden drei Punkte immer kurz überprüfen:

1) Wurde wirklich ein ´C´-Quell-Text geladen ?

2) Wurde auch der **richtige** Quell-Text geladen, sind also Ihre Eintragungen in ´MakeWiz´ richtig gewesen und wurden diese auch korrekt abgespei-

chert ?

3) Es müssen unbedingt die drei wesentlichen Steuer-Buttons ´**MAKE**´, ´**RE-MAKE**´ und ´**DL.BAT**´ vorhanden sein, sonst haben Sie einige Eintragungen in ´MakeWiz´ vergessen oder nicht korrekt abgespeichert.

Wenn all diese Punkte O.K. sind, wurde der Workspace korrekt geladen und nun kann´s richtig los gehen mit der eigentlichen Erstellung bzw. Änderung des Ihres ´C´-Quell-Textes.

Dazu können Sie jetzt das ´C´-Prorgramm ganz normal mit Hilfe der Editor-Funktionen von ´jfe´ eingeben, abändern, ergänzen und korrigieren.

UND NUN DAS BESTE:
direkt von ´jfe´ aus können Sie die Compilierung und die Linkung Ihres Programm-Textes starten.

Dazu klicken Sie einfach den Steuer-Button ´**RE-MAKE**´ an und die Übersetzung und Linkung bis hin zum fertigen, download-baren Intel-HEX-File läuft automatisch ab, **Abb.19**:
(Grundlage für den gesamten Übersetzungs- und Link-Vorgang ist der zuvor mittels MakeWiz erstellte Inhalt der Datei ´**seminar.mak**´, die nun von ´**jfe**´ aus automatisch, im Hintergrund, abgearbeitet wird)

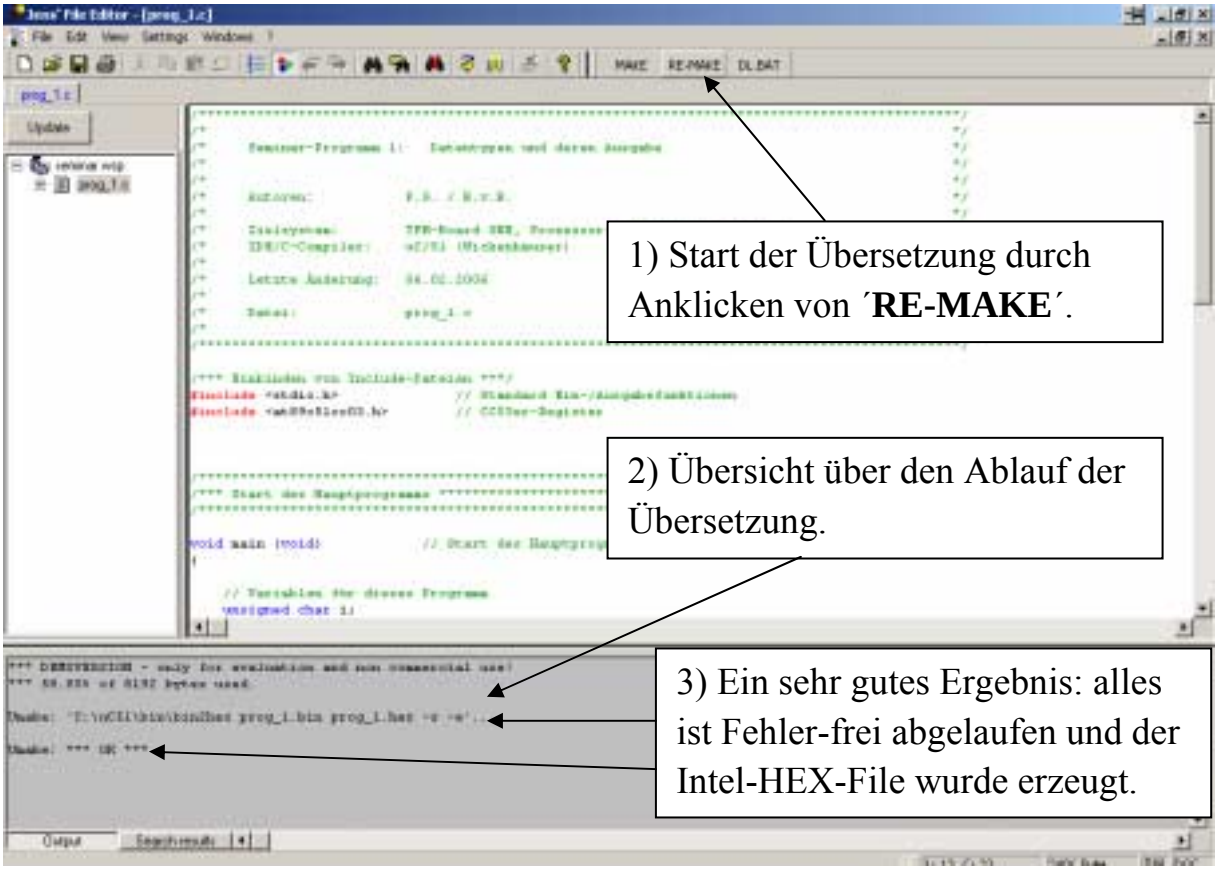

Abb.19: Die automatische Übersetzung und Linkung des ´C´-Programm-Textes

Im unteren, neu erscheinenden Fenster von ´jfe´ können Sie nun sehr gut den Ablauf der Übersetzungs- und Linkungsarbeiten bis hin zum fertigen Endergebnis verfolgen.
Haben Sie keinen Fehler bei der Programmerstellung gemacht, so zeigen Ihnen die letzten 4 Zeilen in diesem Fenster den Erfolg Ihrer Arbeit an:

```
*** DEMOVERSION - only for evaluation and non commercial use!
*** 58.85% of 8192 bytes used.

Umake: 'C:\uC51\bin\bin2hex prog_1.bin prog_1.hex -s -e'...

Umake: *** OK ***
```

1) Wenn Sie (noch) die Demo-Version von µC/51 benutzen, so wird Ihnen angezeigt, wie viel Speicherplatz Sie bereits verbraucht haben.
   Die einzige Einschränkung bei der Demo-Version besteht ja bekannter Maßen darin, daß Sie nur Programme mit einer maximalen Größe von 8 kByte (anstatt von 64 kByte) erzeugen können.
   Hier haben wir also mit unseren Programm bereits 58,85% von 8192 Byte (≡ 8 kByte) verbraucht.

2) Es wurde ein downlaod-barer Intel-HEX-File aus unseren ´C´-Programm ´**prog_1.c**´ erzeugt:
   ´ .....prog_1.hex .....´

3) Es sind keine gravierenden Fehler aufgetreten:
   ´*** OK ***´

Sind dagegen noch (syntaktische) Fehler im Programm enthalten, so sehen die Meldungen leider nicht ganz so optimistisch aus, z.B. so:

```
Error: prog_1.s51 122: unresolved reference to '_printxf'
Error: 1 errors found
Umake: Error: returncode: 2

Umake: Error: Errors found: *** TARGET NOT GENERATED ***
```

Bei dieser Übersetzung wurde also ein Fehler gefunden und daher wurde kein download-barer Intel-HEX-File erzeugt:

```
*** TARGET NOT GENERATED ***
```

Daher bleibt Ihnen in solch einem Fall nicht anderes übrig, als mit den bekannten Methoden nach dem Fehler oder sogar nach den Fehlern zu suchen !

Gehen wir nun aber einmal davon aus, daß alles fehlerfrei übersetzt worden ist und Sie einen **download-baren Intel-HEX-File** vorliegen haben.
Wie kommt dieser jetzt im nächsten Schritt nun in Ihren Mikrocontroller (den **CC03er**) bzw. auf das Mikrocontroller-Board, das **TFH-Board ONE** ?

3. Der Download des Intel-HEX-Files auf das TFH-Board ONE

Um den Intel-HEX-File auf den CC03er ´down zu loaden´, starten Sie als erstes das Down-Load-Programm ´**Flip**´ durch Klicken auf das zugehörige Ikon, **Abb.20**:

Abb.20: Das Aufruf-Ikon zu ´Flip´

WICHTIGER HINWEIS:
Bevor Sie ´**Flip**´ nun starten bzw. bevor Sie generell mit ´**Flip**´ arbeiten, müssen Sie sicherstellen, daß das Terminal-Programm ´**HyperTerm**´ die serielle Schnittstelle des Entwicklungs-PCs zur Kommunikation mit dem CC03er freigegeben hat.

Wenn ´**Flip**´ gestartet ist, sehen Sie das Grundfenster von ´Flip´, **Abb.21**:

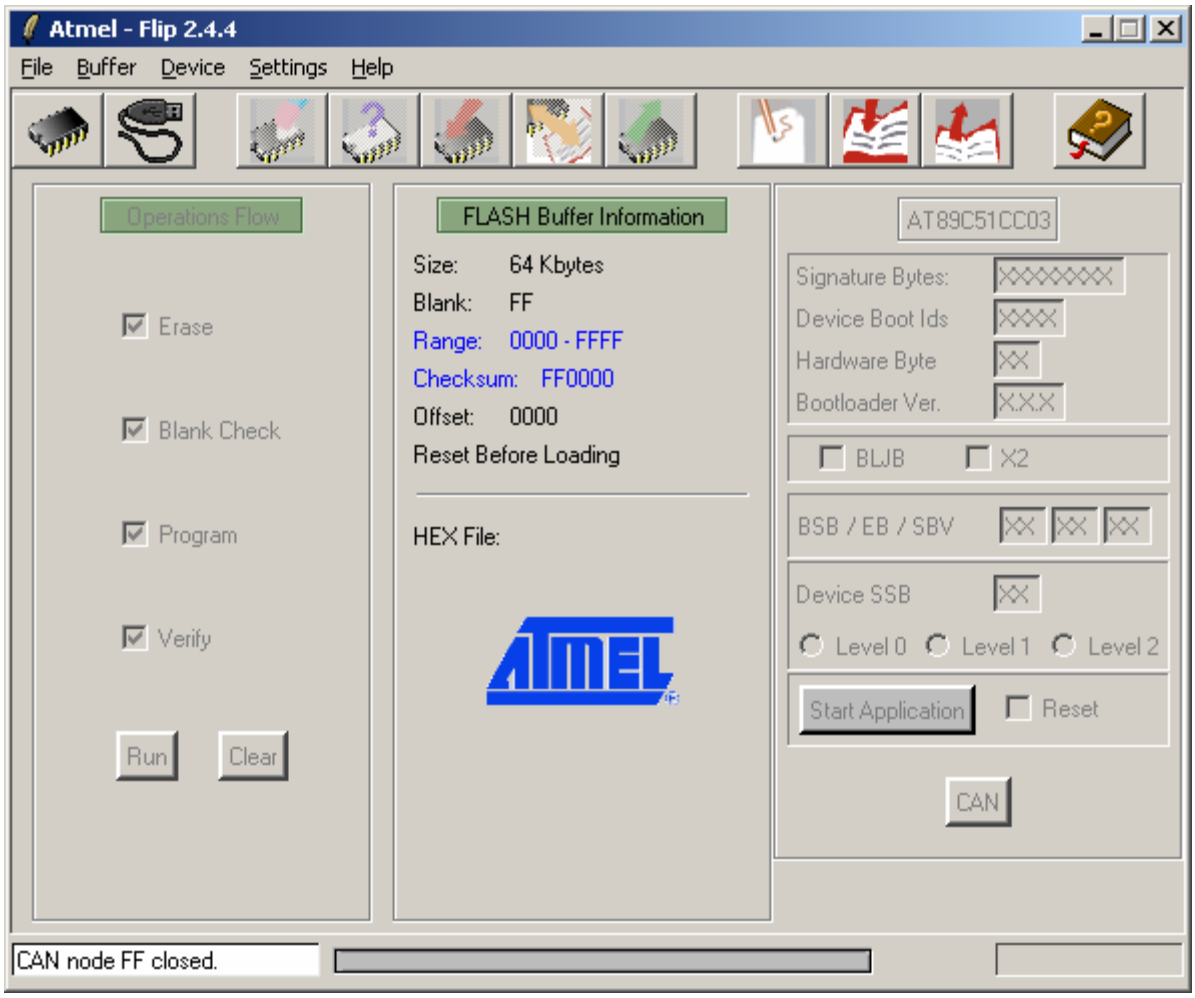

Abb.21: Das Grundfenster von ´Flip´

Bevor Sie mit ´Flip´ weiter arbeiten, müssen Sie zuerst den **CC03er** auf dem TFH-Board ONE erst in den so genannten ´**Boot-Modus**´ bringen:

DER BOOT-MODUS DES CC03ERS:
Um den CC03er in den ´Boot-Modus´ zu bringen, drücken Sie gleichzeitig den Boot- und den Reset-Taster.

Dann lassen Sie den Reset-Taster wieder los, warten 2 Sekunden lang und lassen dann den Boot-Taster los.
Der CC03er ist jetzt im Boot-Modus, d.h. er ist jetzt in der Lage, einen Intel-HEX-File vom Entwicklungs-PC zu empfangen und seinen internen Flash-Programmspeicher damit zu programmieren.
Und

Daher Ganz wichtig:

Bevor Sie mit ´Flip´ irgendein Programm in den CC03er laden wollen, muß der CC03er immer in den Boot-Modus gebracht worden sein, sonst erscheint eine Fehlermeldung (s. **Abb.22**) und nichts passiert !

Abb.22: ´Flip´ kann keine Verbindung mit dem CC03er aufnehmen, da dieser nicht im Boot-Modus ist

Kommen wir nun zurück zu ´Flip´: wenn Sie ´Flip´ zum aller ersten Mal aufrufen, müssen Sie als Erstes (einmalig) angeben, mit welchem Mikrocontroller der Firma Atmel Sie arbeiten, **Abb.23**:

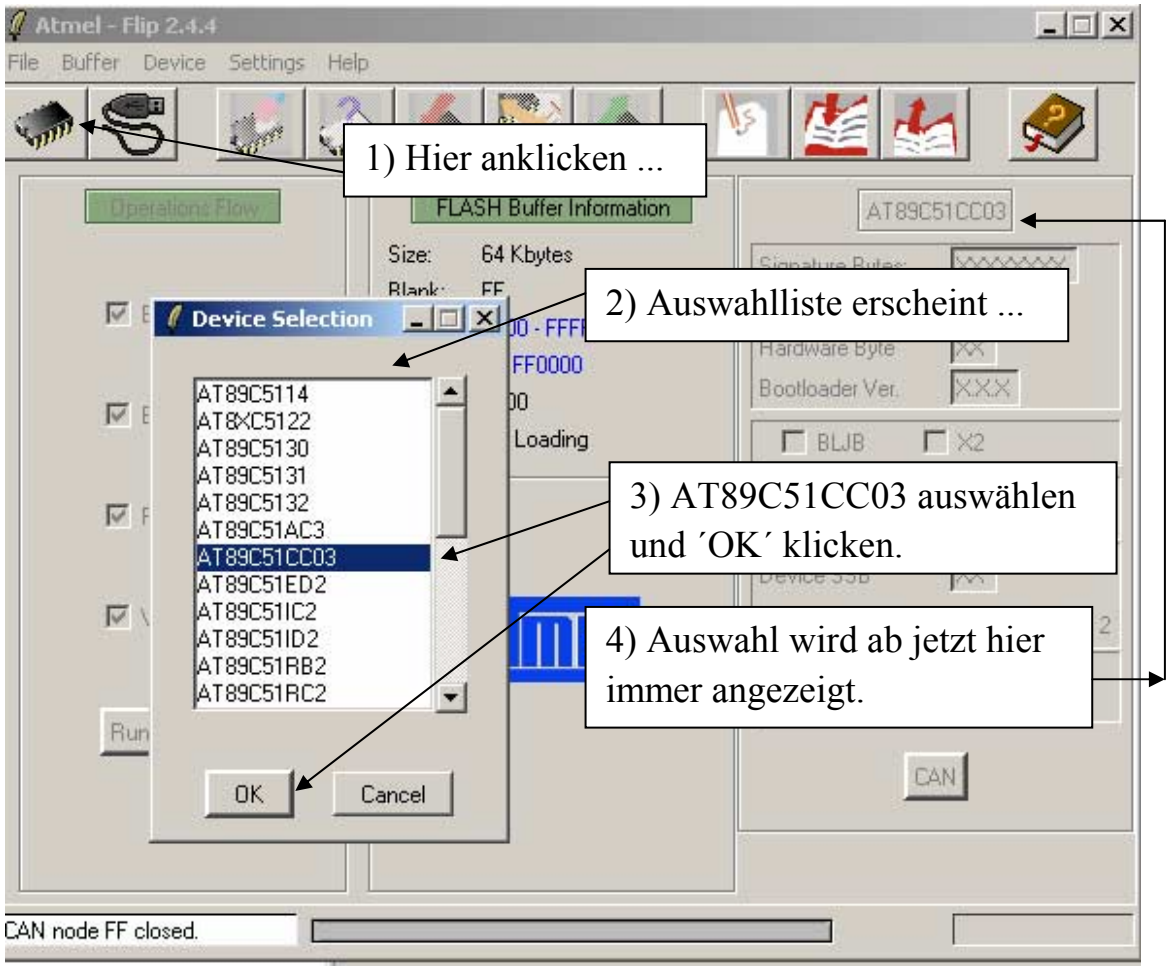

Abb.23: Die Auswahl des Mikrocontrollers

Zur **Auswahl des Mikrocontrollers** klicken Sie auf das Mikrocontroller-Symbol ganz links oben in der Symbolleiste.
Darauf hin erscheint eine Liste (´**Device Selection**´) aller auswählbaren 8051er-Mikrocontroller der Firma Atmel. Sie wählen den **AT89C51CC03** aus und bestätigen Ihre Auswahl mit ´OK´. Das war´s schon.
In der rechten oberen Ecke vom Flip-Fenster ist nun der gewählt Mikrocontroller eingetragen und da ´Flip´ diese Auswahl bis zur nächsten Änderung selbständig abspeichert, brauchen Sie diesen Schritt ab jetzt nicht mehr durch zu führen: der CC03er wird bei jedem Aufruf von ´Flip´ immer automatisch ausgewählt.

Und noch einmal:
Vor dem nächsten Schritt müssen Sie unbedingt sicherstellen, daß das Terminal-Programm ´HyperTerm´ die serielle Schnittstelle des Entwicklungs-PC´s zur Kommunikation mit dem CC03er auch wirklich freigegeben hat, d.h. Sie müssen unbedingt beachten, daß ´HyperTerm´ ´getrennt´ bzw. ´disconnected´ ist, sonst meldet ´Flip´ einen Fehler (s. nachfolgend) !

Als nächstes müssen Sie die COM-Schnittstelle auswählen, über die Ihr Entwicklungs-PC mit dem TFH-Board ONE (mit dem CC03er) verbunden ist, **Abb.24**:

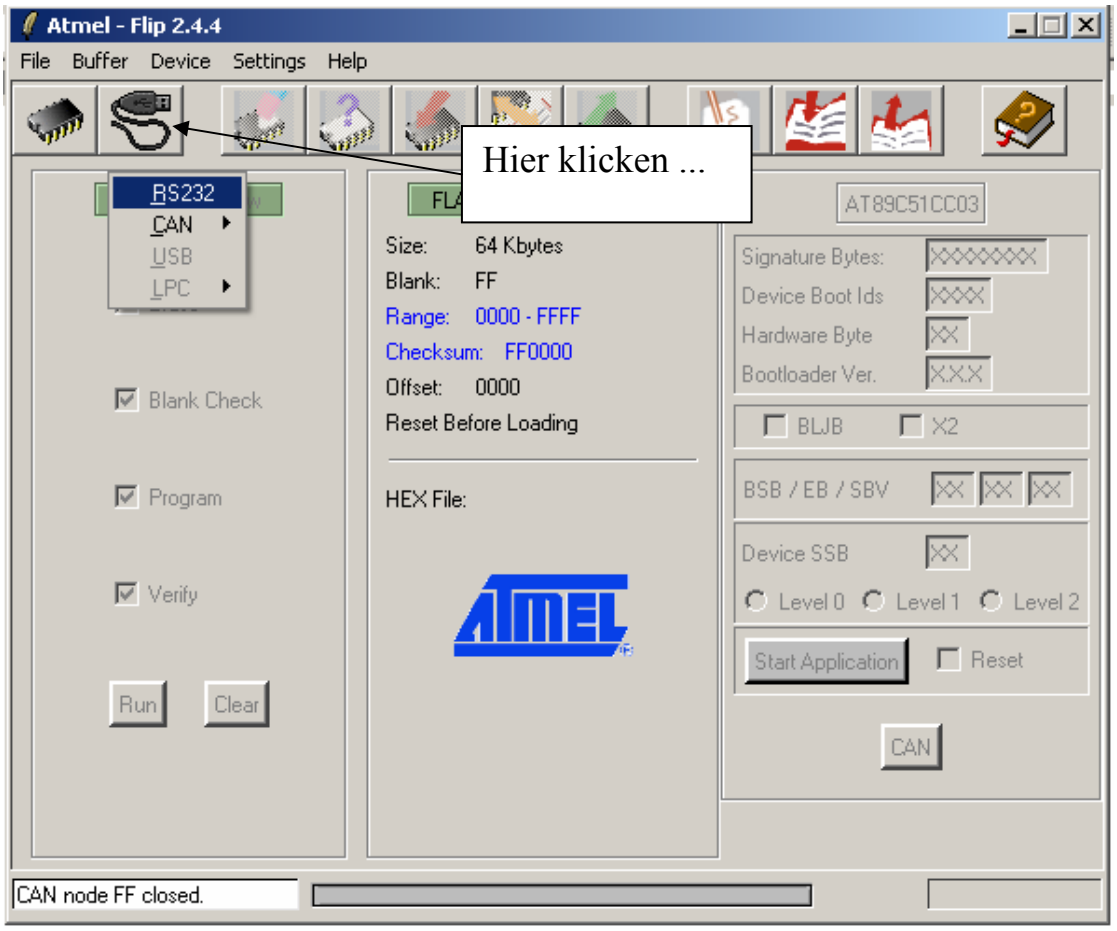

Abb.24: Die Auswahl der COM-Schnittstelle

Es erscheint ein kleines Auswahl-Fenster in dem Sie ´**RS232**´ anwählen.
Nun erscheint ein weiteres Fenster, in dem Sie ganz konkret die COM-Schnittstelle auswählen müssen, **Abb.25**:

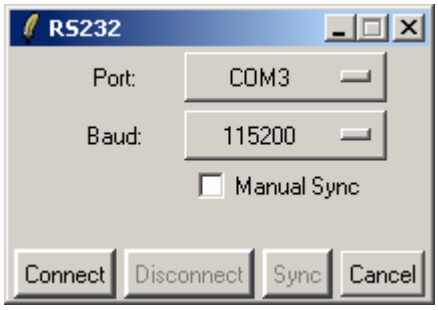

Abb.25: Die konkrete Auswahl der COM-Schnittstelle des Entwicklungs-PC´s

In unserem Fall haben wir die Schnittstelle **COM3** ausgewählt, weil dort das TFH-Board ONE angeschlossen ist.
Andere Anpassungen als diese Auswahl des Kommunkations-Ports brauchen Sie nicht durch zu führen, insbesondere die maximale Baudrate von 115200 Bit/s braucht nicht geändert zu werden: beim Download paßt sich Flip der maximal möglichen Geschwindigkeit des jeweils ausgewählten Mikrocontrollers automatisch an, so daß der Intel-HEX-File schnellstmöglich übertragen wird.
An diesem Punkt kann es Ihnen passieren, daß Sie, gerade am Anfang, mit zwei **Fehlermeldungen** konfrontiert werden:

1) ´HyperTerm´ hält den zugehörigen COM-Port noch besetzt, d.h. Sie haben vergessen, ´HyperTerm´ vom COM-Port zu trennen. Dann erscheint die nachfolgende Fehlermeldung (oder eine ähnliche), weil ´Flip´ nicht auf den gewünschten COM-Port zugreifen kann **Abb.26**:

Abb. 26: HyperTerm hält den COM-Port noch besetzt

2) Sie haben den CC03er nicht in den Boot-Modus gebracht und ´Flip´ kann keine Verbindung mit dem Mikrocontroller aufnehmen, es entsteht ein ´Timeout Error´, s. Abb.22:

Haben Sie aber alles richtig gemacht,

- es sind keine Fehlermeldungen aufgetreten und
- bestimmte „Farbmerkmale“ sind im Fenster vorhanden,

so kann ´Flip´ die Verbindung mit dem CC03er aufnehmen und Sie können den Intel-HEX-File zum Mikrocontroller transferieren, **Abb.27**:

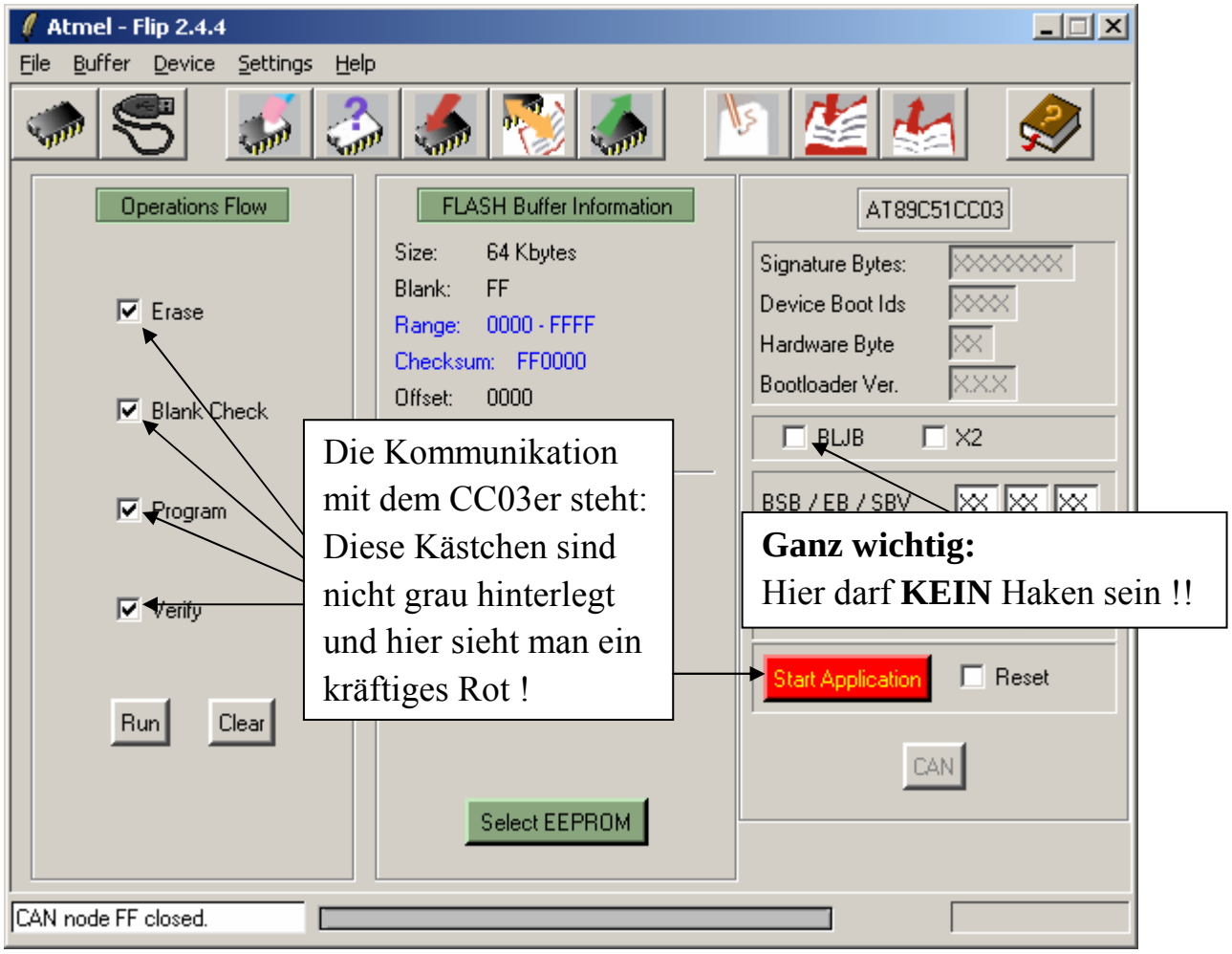

Abb.27: Die Verbindung mit dem CC03er steht

GANZ WICHTIG:

Sie müssen unbedingt darauf achten, dass im Flip-Fenster **KEIN** Haken im Kästchen ´**BLJB**´ steht, sonst läuft das herunter geladene Programm später nicht richtig an !

Das ist auch dann wichtig, wenn ein ´nagel-neuer´ CC03er das erste Mal programmiert werden soll ! (denn im Auslieferungszustand eines neuen CC03ers ist dieses Steuerbit gesetzt, näheres s. Datenblatt zum CC03er)

Nun muß als nächstes der in den CC03er zu programmierende Intel-HEX-File nach ´Flip´ hineingeladen und danach zum CC03er gesendet werden.
Zum Laden des HEX-Files nach ´Flip´ klicken Sie auf das Lade-Symbol, **Abb.28**:

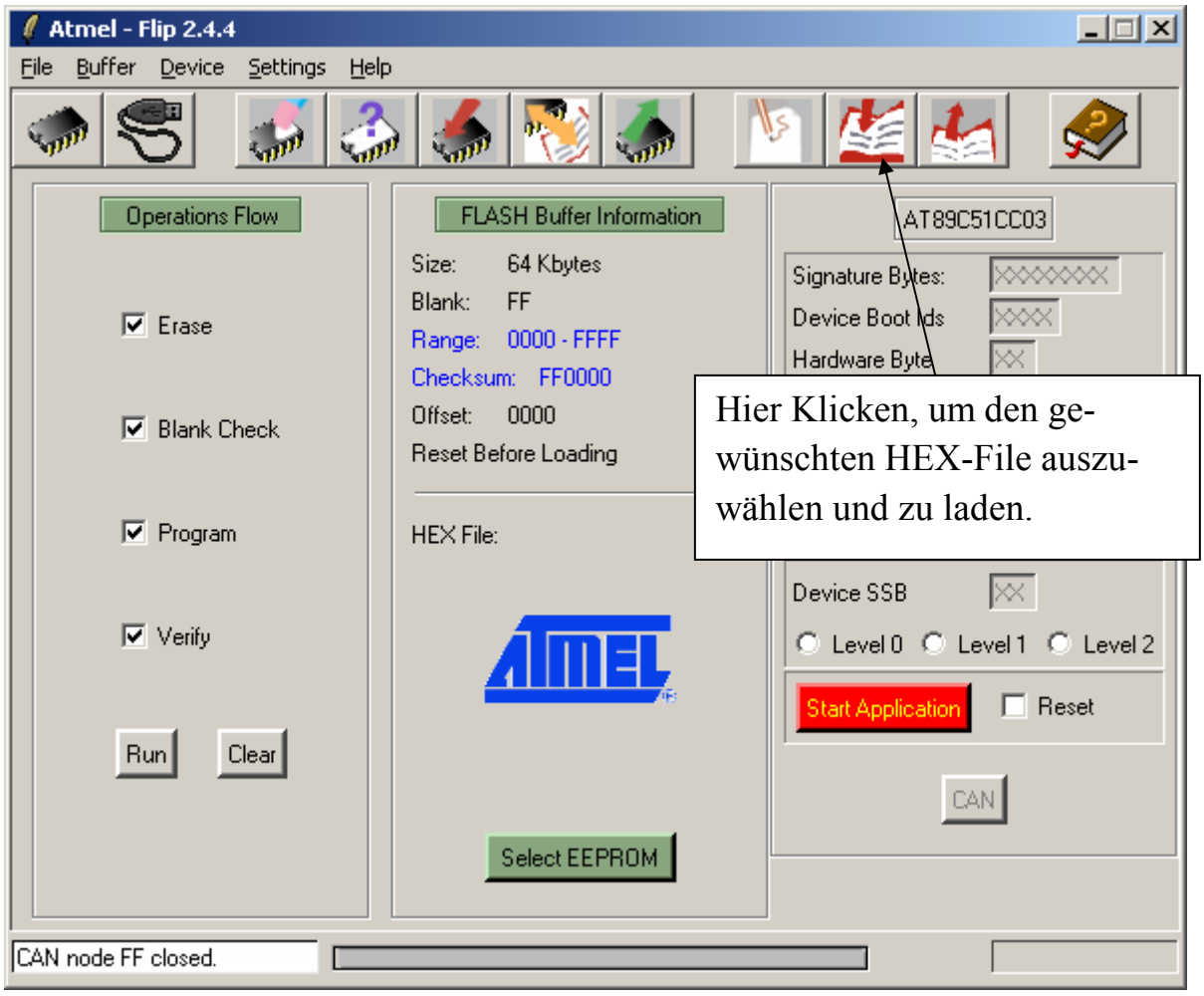

Abb.28: Das Laden des gewünschten Intel-HEX-Files

Im nun erscheinenden Auswahlfenster wählen Sie unseren HEX-File ´**prog_1.hex**´ aus dem Verzeichnis ´**Seminar**´ aus und bestätigen diese Auswahl mit ´**Öffnen**´.

´prog_1.hex´ wird nun nach ´Flip´ eingeladen und im ´Flip-Fenster´ können Sie bereits einige wichtige Daten dieses HEX-Files erkennen, **Abb.29**:

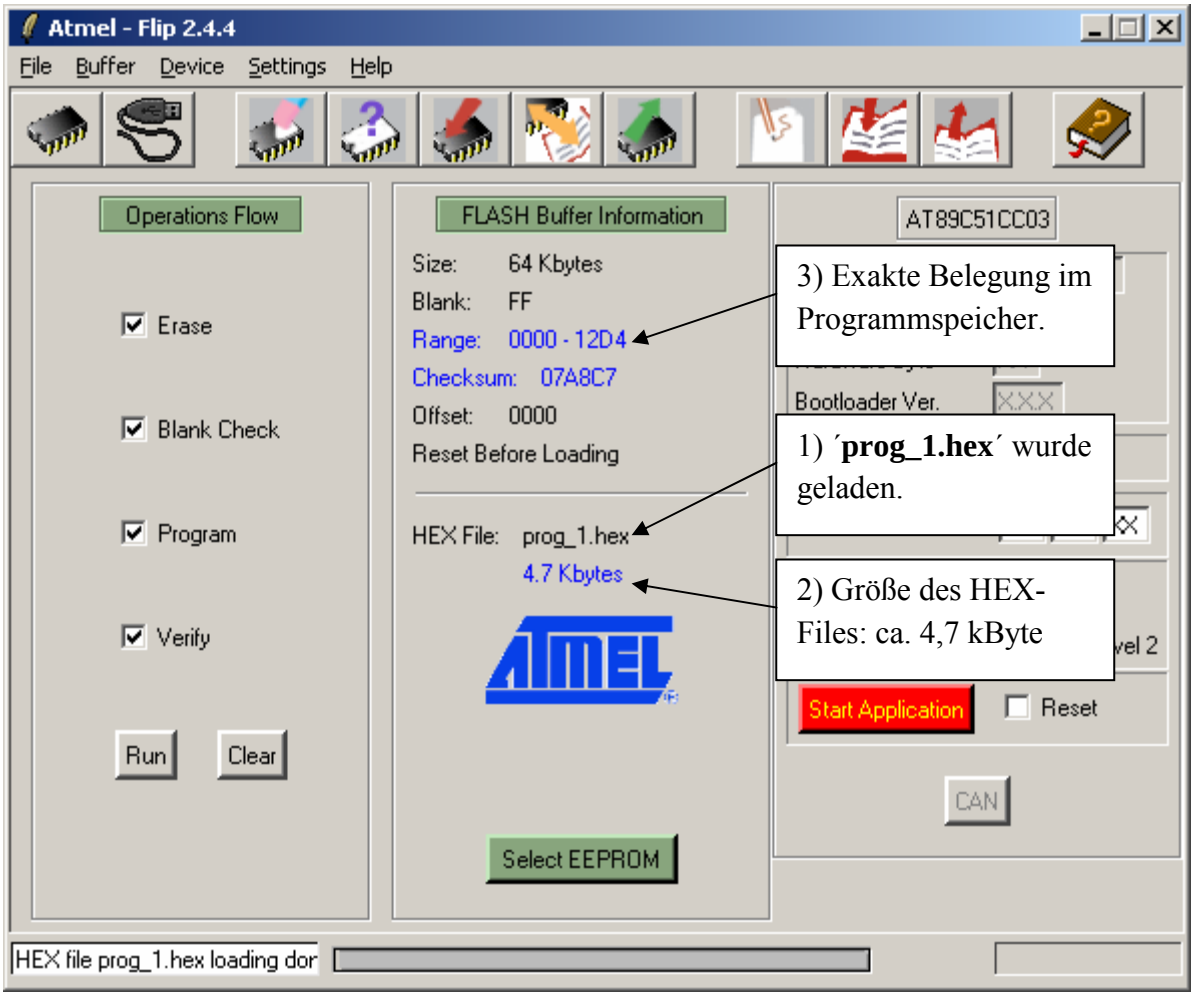

Abb.29: Der geladene Intel-HEX-File ´prog_1.hex´

Nun müssen Sie diesen Intel-HEX-File als Letztes noch in den CC03er laden und das machen Sie, indem Sie auf ´**Run**´ klicken, **Abb.30**:

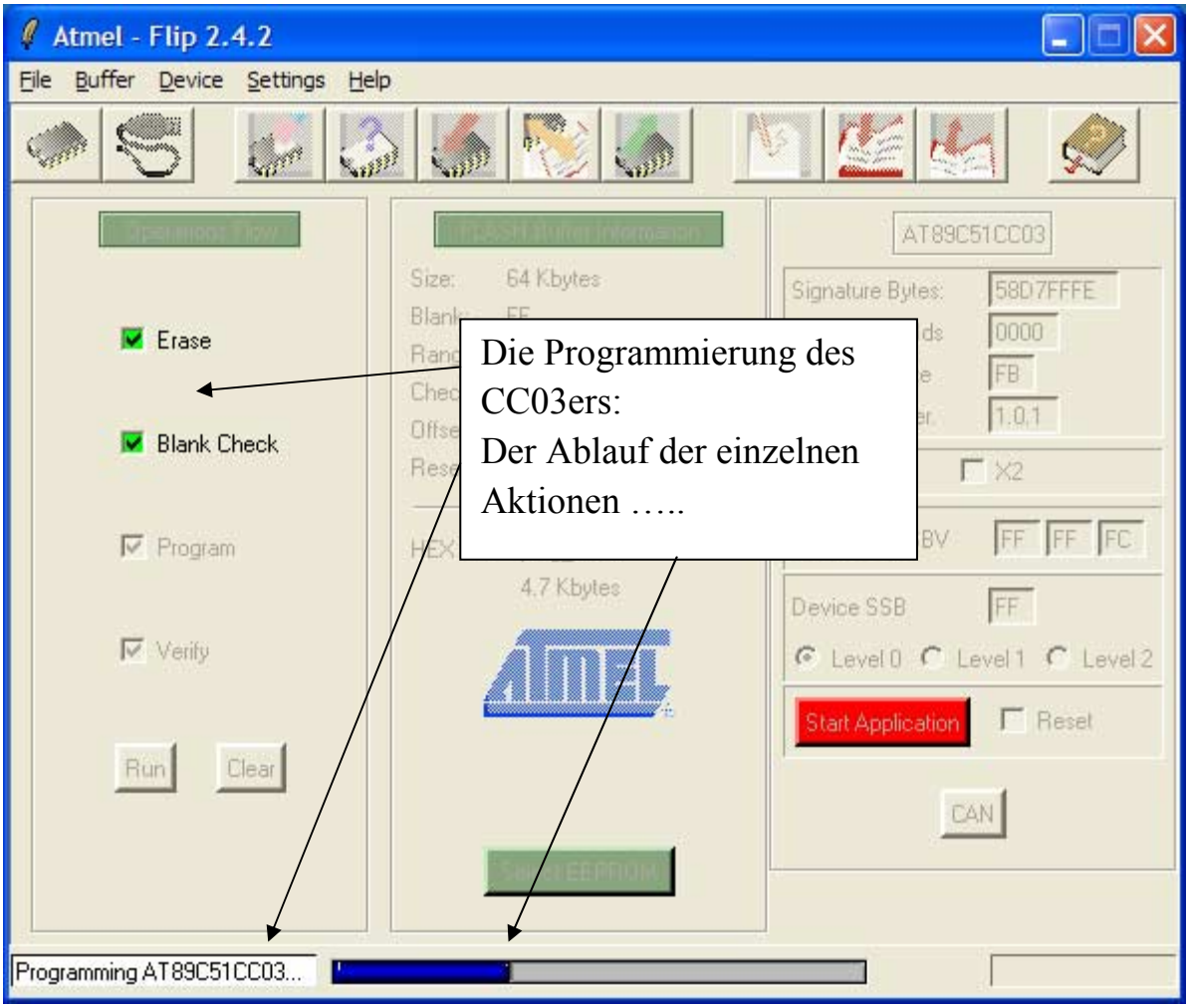

Abb.30: Die Programmierung des CC03ers mit dem Intel-HEX-File

Es werden nun die einzelnen notwendigen Schritte zur Programmierung des CC03ers durchgeführt und durch die farblich grüne Hinterlegung der entsprechenden Kästchen angezeigt:

- das Löschen des internen CC03er-Programmspeichers: ´**Erase**´
- das Überprüfen, ob der Programmspeicher auch wirklich leer ist: ´**Blank Check**´

- das Programmieren des Programmspeichers: ´**Program**´ und
- die Überprüfung, ob die Programmierung auch korrekt erfolgt ist: ´**Verify**´

Wenn diese vier Schritte ohne Fehlermeldung abgelaufen sind, ist der CC03er korrekt programmiert worden und

als letzten wichtigen Schritt müssen Sie auf den roten Button ´Start Application´ klicken, damit ´Flip´ die serielle Schnittstelle des Entwicklungs-PC´s wieder frei gibt.

Dann kann nämlich ´HyperTerm´ wieder auf diese Schnittstelle zugreifen und das ist die grundlegende Voraussetzung, wenn Sie im letzten Punkt Ihr Programm austesten möchten.

Sie können nun ´Flip´ beenden, d.h. ´Flip´ zunächst nach unten in die Task-Leiste weg klicken.

4. Der Test des Programms mit dem Terminal-Programm ´HyperTerm´

Nun können Sie das Terminal-Programm ´HyperTerm´ aufrufen und die Verbindung mit dem Mikrocontroller-Board herstellen, **Abb.31**:

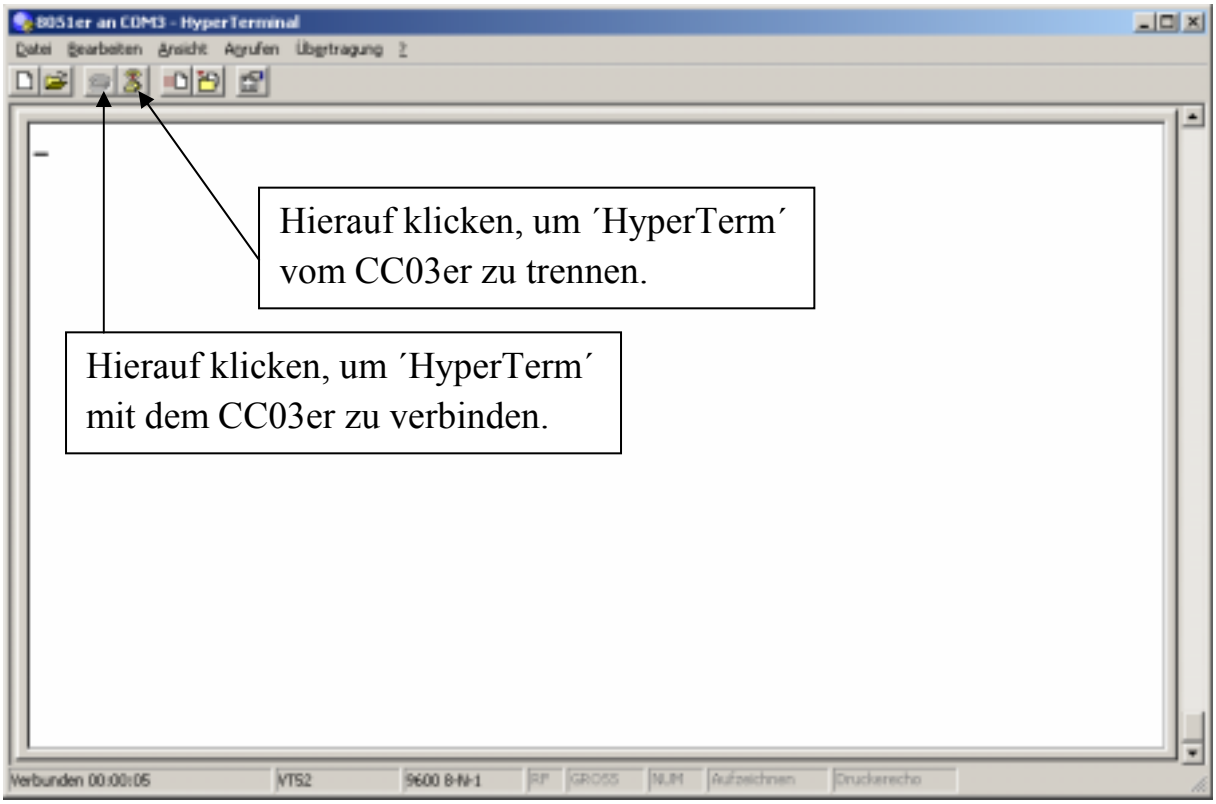

Abb.31: Der Aufruf von ´HyperTerm´ und die Verbindung mit dem CC03er

Wenn Sie ´HyperTerm´ aufgerufen haben bzw. wenn Sie ´HyperTerm´ mit dem CC03er verbinden wollen, kann noch eine Fehlermeldung auftreten, **Abb.32**:

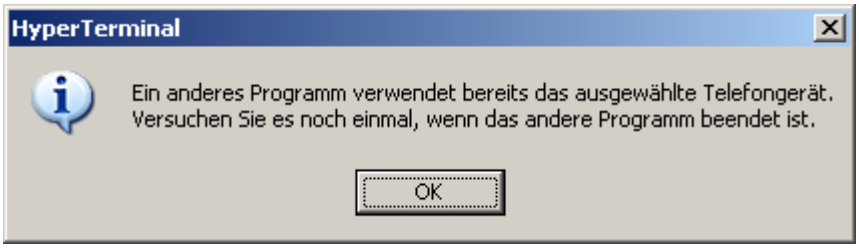

Abb.32: Das Programm ´Flip´ hält die serielle Schnittstelle noch besetzt

In diesem Falle haben Sie bei ´Flip´ vergessen, auf den Button ´Start Application´ zu klicken und ´Flip´ hält daher die serielle Schnittstelle noch für sich besetzt.
Öffnen Sie daher ´Flip´ noch einmal, klicken Sie auf diesen Button und schalten Sie dann ´Flip wieder in die Tast-Leiste zurück.
Nun können Sie ´HyperTerm´ erneut und problemlos mit dem TFH-Board ONE verbinden.

Einem ersten Test Ihres 8015er-Programms steht nun nichts mehr im Wege: drücken Sie daher auf den Reset-Knopf des TFH-Board ONE und verfolgen Sie den Ablauf Ihres Programms (die printf-Ausgaben Ihres Programms) auf dem Bildschirm.

Hinweis:
Sie müssen natürlich ´HyperTerm´ zuvor richtig eingerichtet haben (wie in unseren vorangehenden Ausführungen bereits erläutert), **Abb.33**:

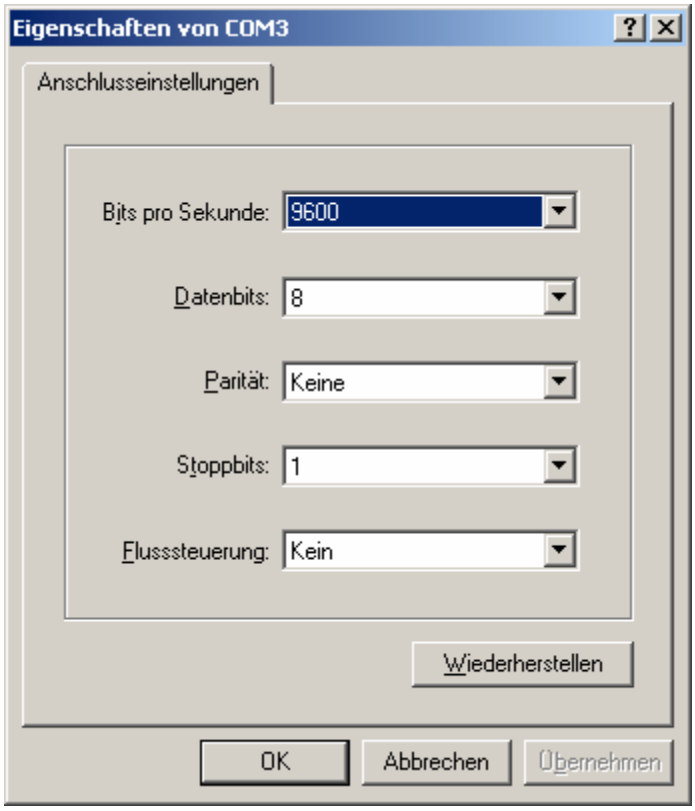

Abb.33: Die korrekte Einrichtung von ´HyperTerm´

Das war´s eigentlich schon mit unseren ausführlichen Erläuterungen zur Programmierung eines 8051ers, speziell des CC03ers, unter µC/51.

Als Letztes haben wir noch einmal alle wichtigen Schritte in Kurzfassungen zusammen gefaßt, so daß Sie diese zwei Seiten für Ihre tägliche Arbeit als kleine Gedächtnisstütze ausdrucken und an Ihrem Programmierplatz hinlegen können.

Die Programmierung eines 8051ers unter µC/51

1. Die Einrichtung des Workspaces (der Projektumgebung) mit ´*MakeWiz*´

Die Grundeinrichtung eines Workspaces ist i.a. nur erstmalig notwendig. Spätere Änderungen bei der Einrichtung von neuen Projekten (neuen ´C´-Programmen) beschränken sich lediglich auf neue Eintragungen auf der Registerkarte ´Components´ unter ´MakeWiz´ (≡ Löschung des alten und Einbindung des neuen ´C´-Files).

2. Die ´C´-Programm-Erstellung mit ´*jfe*´

1) Nach jedem Aufruf von ´jfe´: Laden des gewünschten Workspaces.

2) Programm-Eingabe, Änderung, Erweiterung mit Hilfe von ´jfe´.

3) Programm übersetzen mit ´Make´ bzw. ´Re-Make´.

4) Fehler korrigieren, bis alles ´O.K.´ ist.

5) Download-barer Intel-HEX-File wurde erzeugt.

6) ´jfe´ in die Taskleiste wegklicken.

3. Programm-Transfer mit ´*Flip*´

1) **´HyperTerm´ deaktivieren ´Offline´**.

2) Mikrocontroller in den Boot-Modus bringen (Tasten ´Reset´ und ´Boot´).

3) ´Flip´ aufrufen und Schnittstelle verbinden.

4) Hex-File in ´Flip´ einladen.

5) Auf ´Run´ klicken und abwarten bis fertig.

6) Wenn fertig: ´Start Application´ anklicken.

7) ´Flip´ in Taskleiste wegklicken.

4. Programm-Test mit ´***HyperTerm***´

1) ´HyperTerm´ aktivieren.

2) Reset auf dem Board drücken, F E R T I G !!

3) HyperTerm deaktivieren vor erneutem Aufruf von Flip.

Bei jeder Änderung, Korrektur und Erweiterung des Programms: weiter bei ´2. Die ´C´-Programm-Erstellung mit ´***jfe***´´.