M16C Installation und Test

Ziel

Erfolgreiche Installation der M16C Entwicklungsumgebung und Sicherstellen der Funktionsfähigkeit mit einem einfachen Testprogramm.

Umfeld

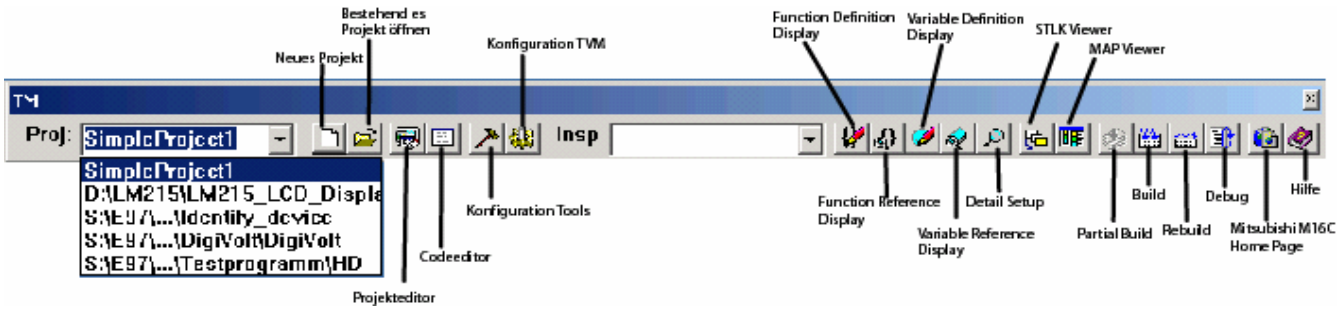
Die Embedded-System Programmierung wird M16C62-Evaluationsboard der HTA-BE durchgeführt. Es ist ähnlich dem Mitsubishi MSA0654, hat aber mehr Periferie zum Experimentieren.

Installation

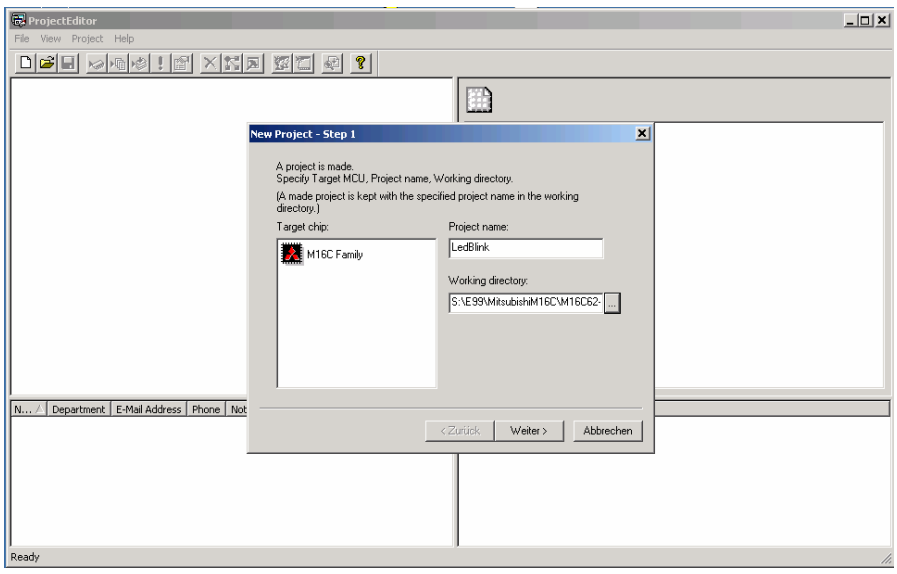
Ist gemäss der Einführung und Manual zu Evaluationsboard vorzunehmen. Auf den 8 Laborrechnern ist die Entwicklungssoftware bereits installiert.

Beispiel: LED Blinker

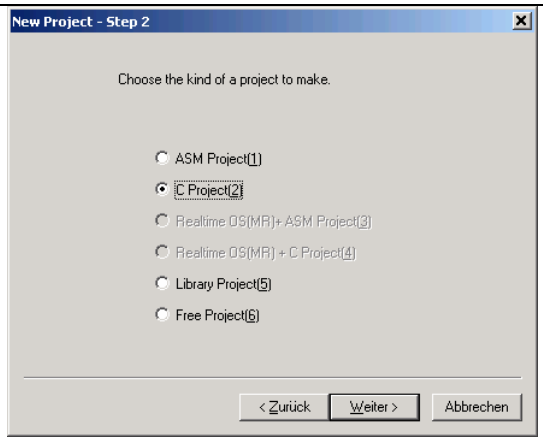
1. Projektmanager TM über Icon auf dem Desktop starten.
   In der Listbox links erscheinen die jeweils letztgeöffneten Projekte:

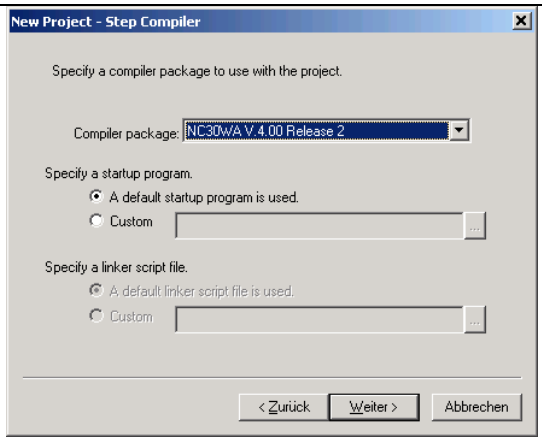


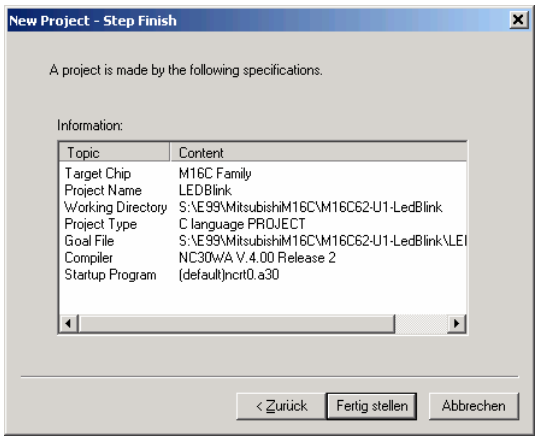
2. Neues Projekt definieren.
   Dazu zuerst Verzeichnis für die neuen Files erstellen. Nachher Projektnamen und Arbeitsverzeichnis eingeben. Nicht vergessen auf das M16C Family Icon zu klicken:



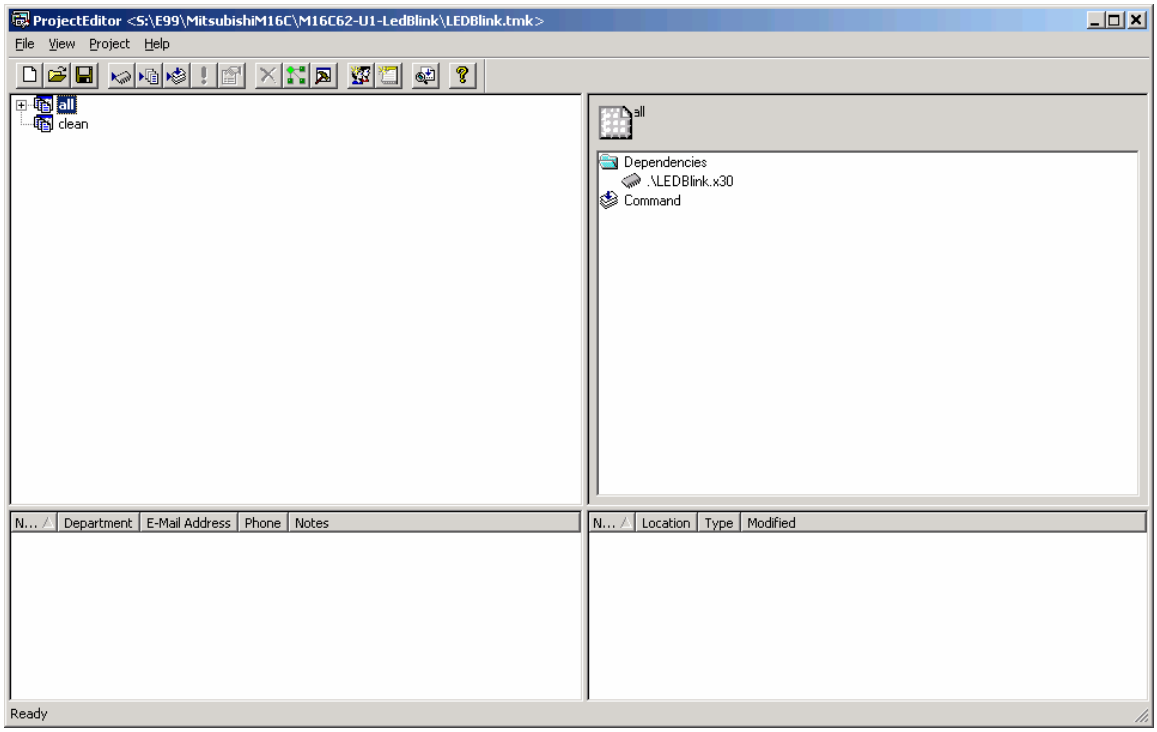
   Mit weiter die restlichen Konfigurationseinstellungen bestätigen:







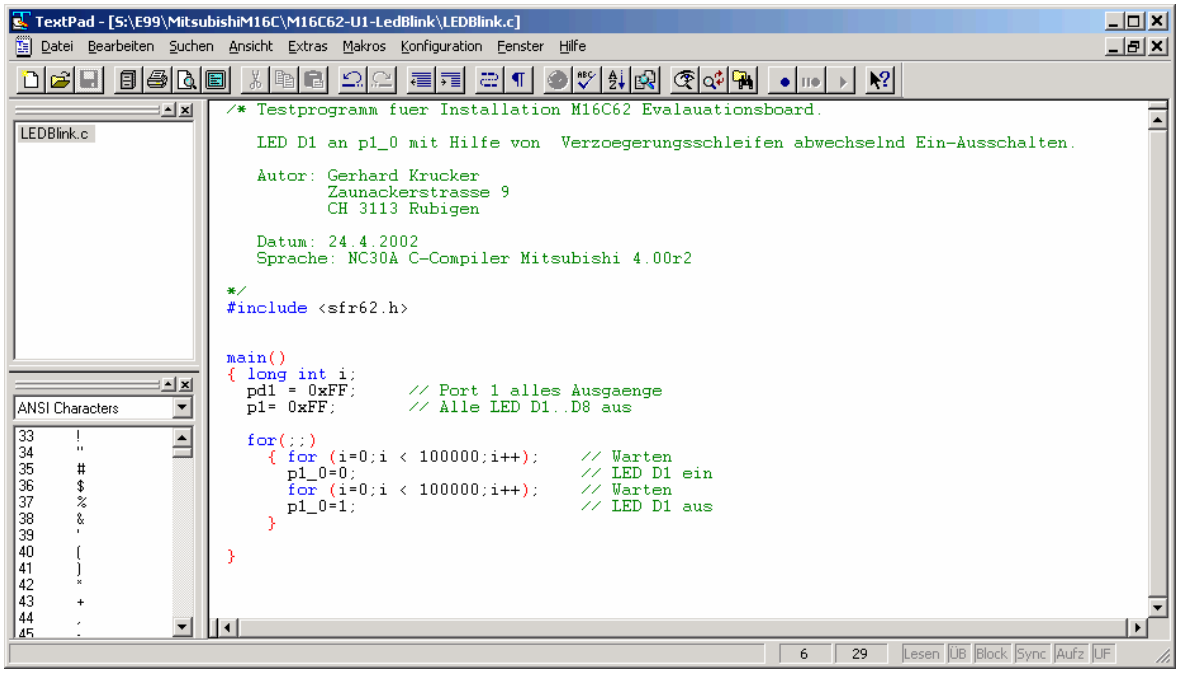
Nach „Fertig stellen“ ist das Projekt konfiguriert:



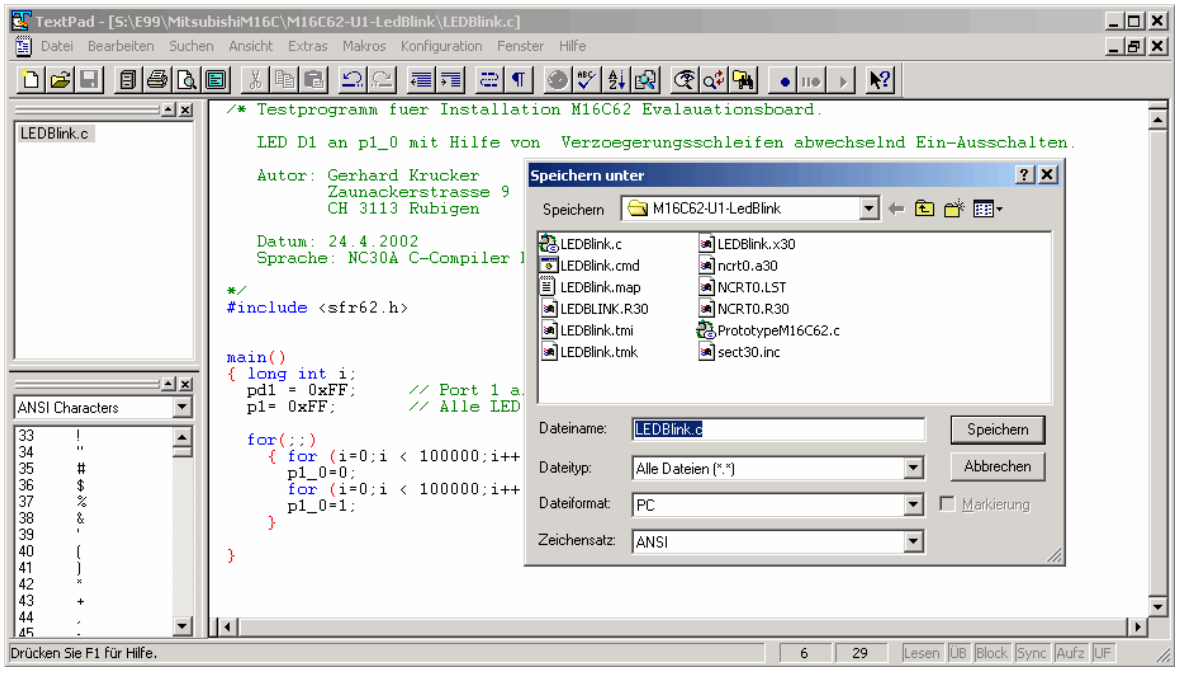
Weiterführende Informationen zur Konfiguration und Arbeit entnehme man dem ausführlichen Handbuch zum TM-Projektmanager.

3. Include- und Startup-Files:
SECT30.INC und NCRT0.A30 werden automatisch in das Arbeitsverzeichnis zum Projekt kopiert. Ev. kontrollieren, dass für die Hardware richtige Version verwendet wird.

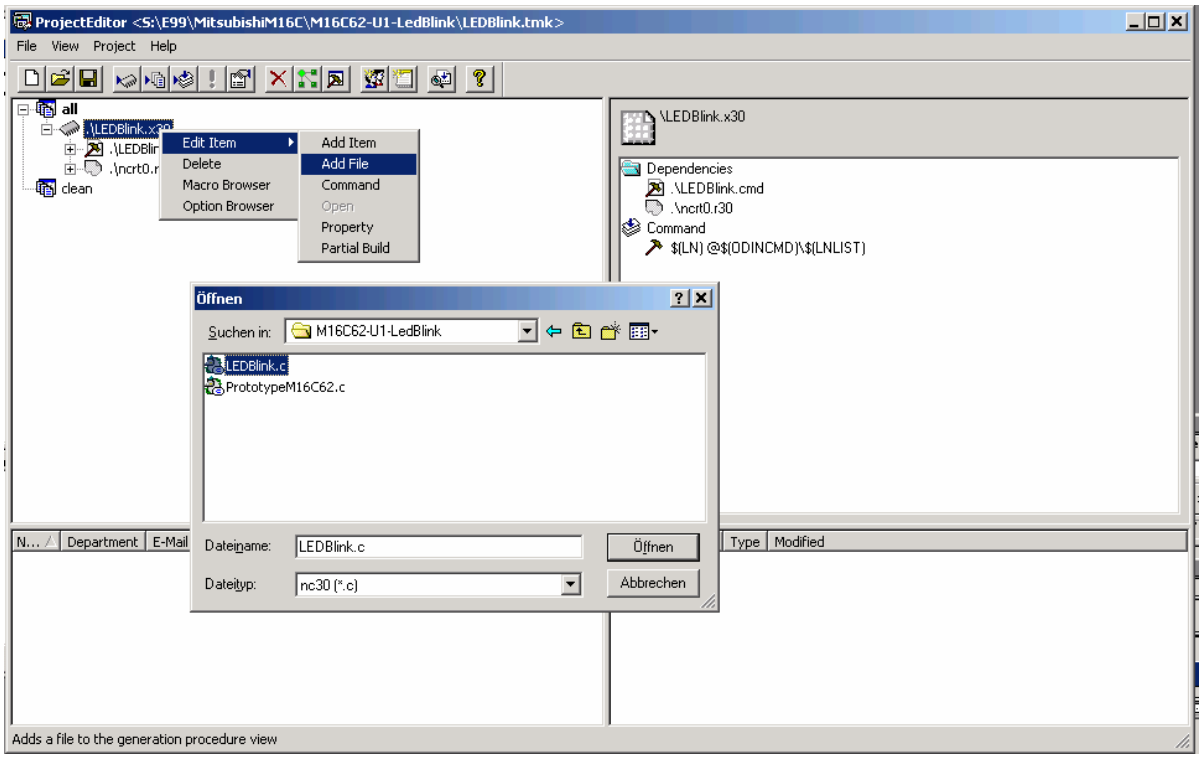
4. Eingeben des Sourcecodes.
Dazu Editor in der TM-Leiste starten und Code eingeben. Grundlage wäre PrototypeM16C62.C mit den Ergänzungen:



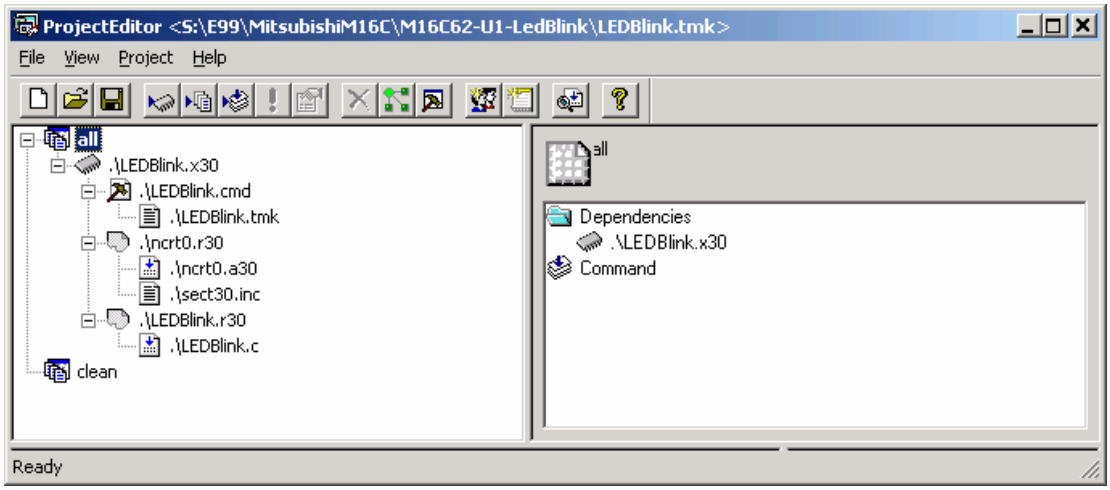
Module mit sinnvollem Namen und mit Extension .C speichern. Der Quelltext wird nach erstmaligem Speichern kontextsensitiv eingefärbt.



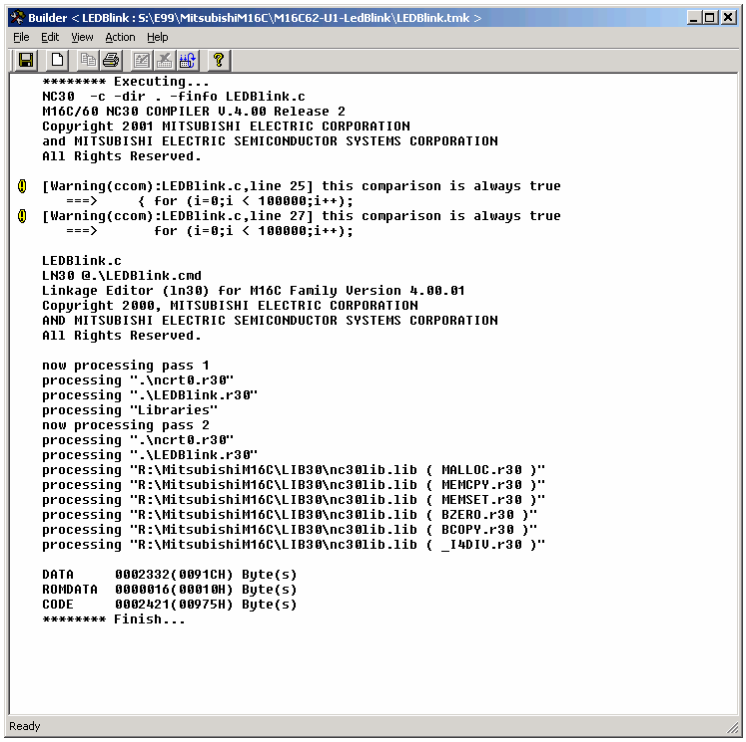
5. Quelltextmodul dem Projekt zufügen.
Dazu im Projektmanager über Add Item oder Add File das File zufügen:



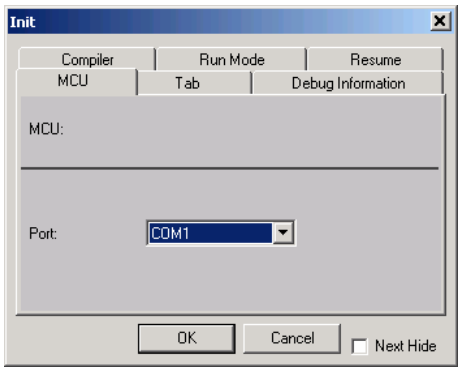
Kontrollieren, dass das zugefügte File auch wirklich im Projekt liegt: LEDBlink.tmk und NCRT0.r30 und LEDBlink.r30 müssen alle unterhalb LEDBlink.x30 liegen.



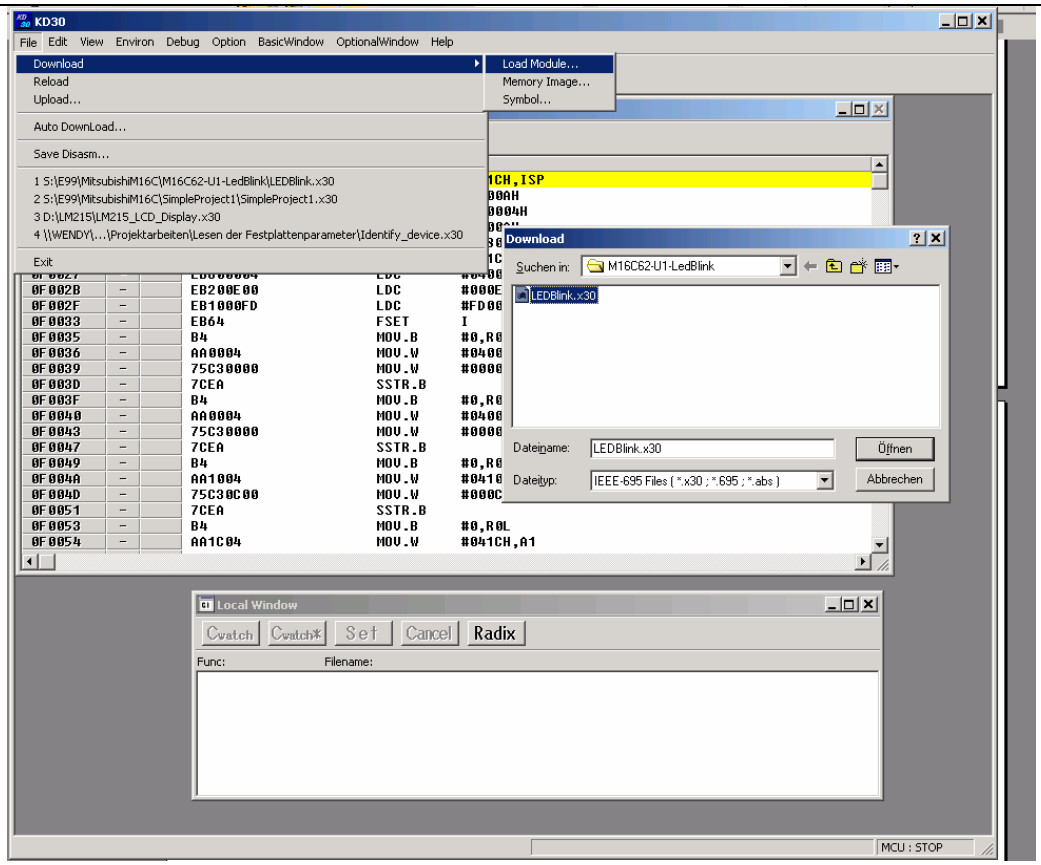
6. Kompilieren /Builden.
   Start des Compilers über die Toolbar des TM Projektmanagers. Ein neues Fenster zeigt den Kompilationserfolg. Erstmalig wird auch der Startup-Code assembliert.



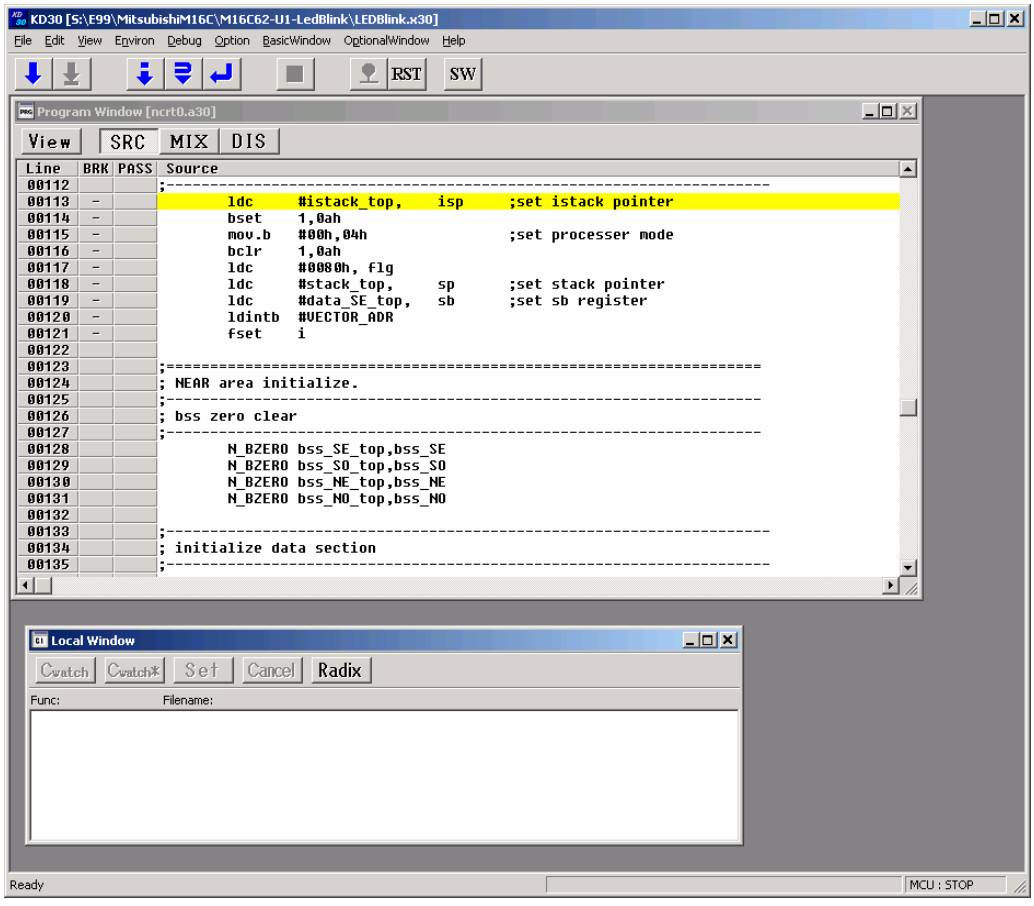
7. Test mit KD30 Debugger:
   Nach erfolgreicher Kompilation kann der Debugger aktiviert werden. Das Evaluationsboard anschliessen und einschalten. Debugger aus der TM-Leiste starten. Es erscheint ein Konfigurationsdialog. Die Schnittstellendefinition kontrollieren und starten:



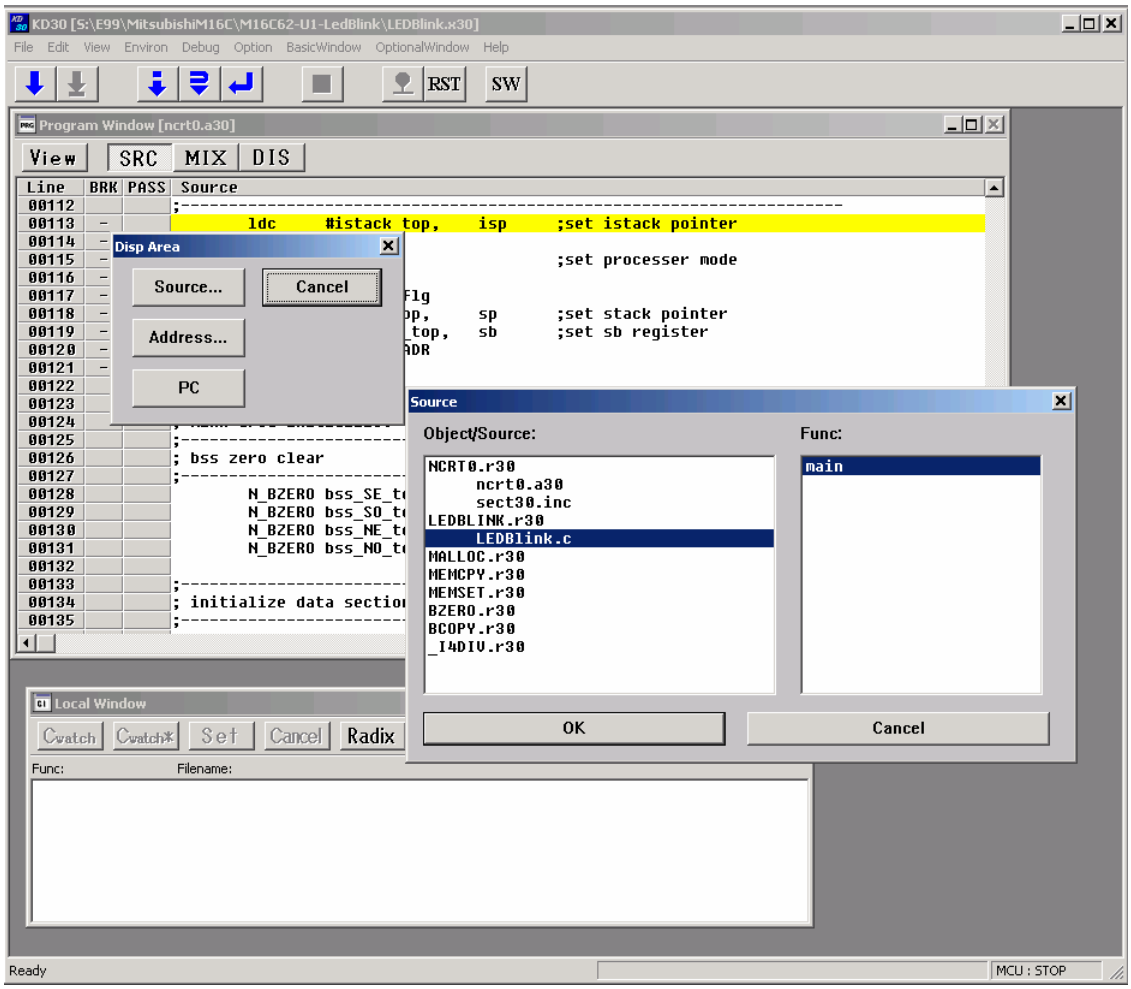
   Das Debuggerfenster wird geöffnet. Es zeigt den jeweils letztgeladenen Code an. Den Knopf RST im Debugger-Menü 1x drücken und Codemodul mit Extension `*.x30` laden:



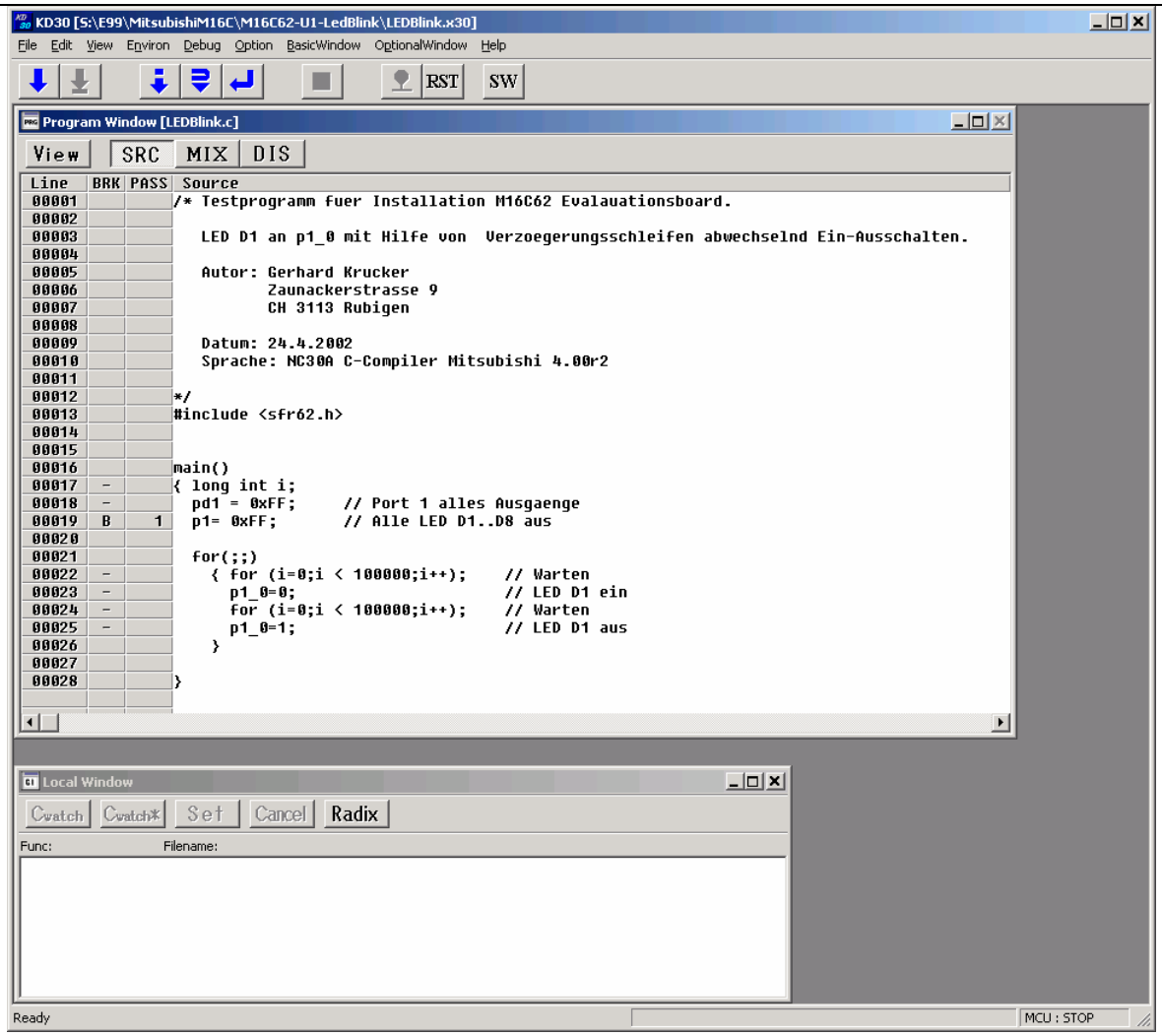
Nach dem Laden zeigt das Fenster den Startup-Code:



Das eigentliche Hauptprogramm kann über View/Source eingestellt werden:



Wir sehen nachher das eigentliche Hauptprogramm, wie auscodiert. In der Spalte BRK können über Mausclick maximal 2 Haltpunkte gesetzt werden.



8. Starten des Programmes. Über die Pfeiltaste links in der Toolbar des Debuggers

Aufgaben

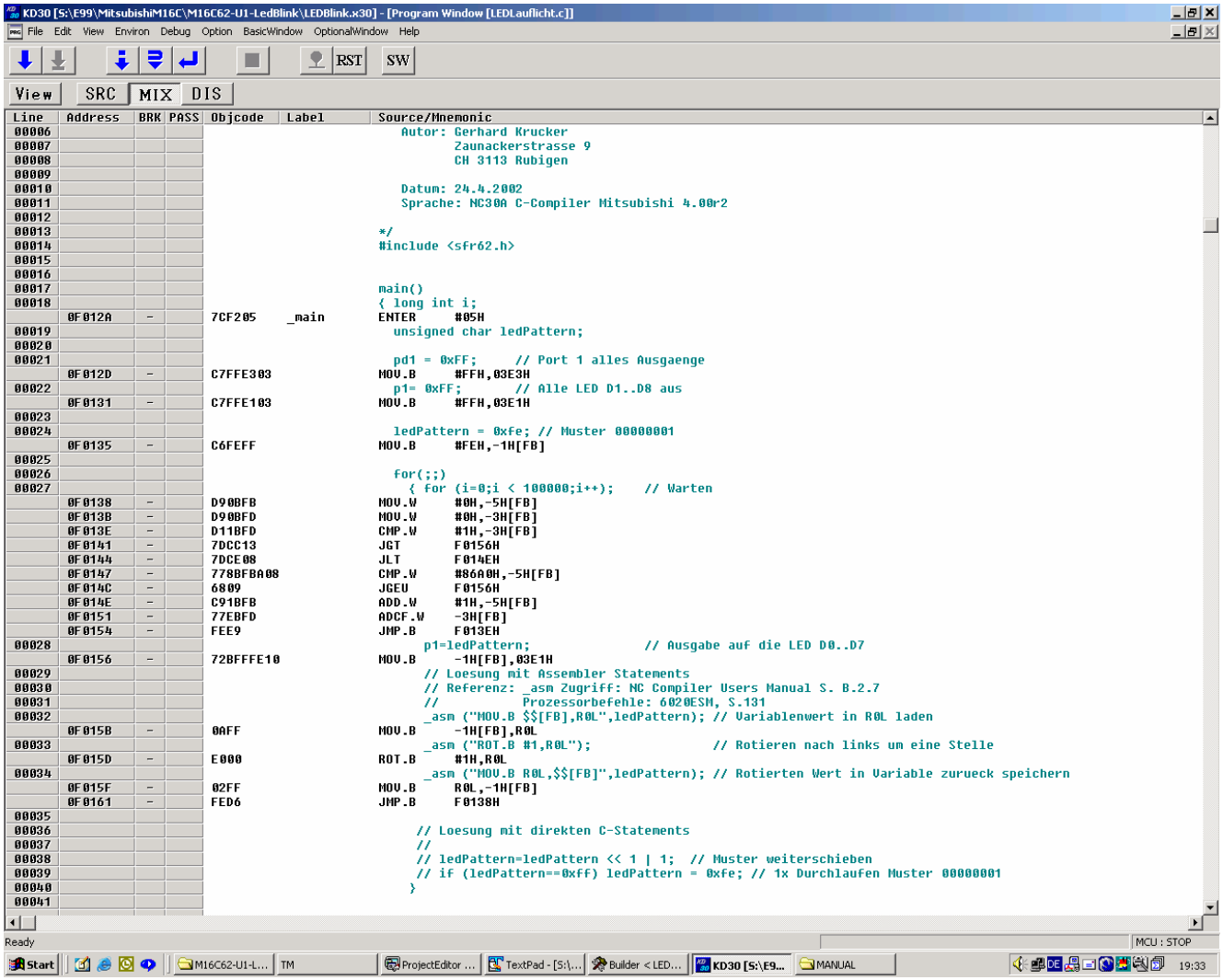
1. Installation der SW und Aufbau der HW gemäss Manual. Speisung auf 12V Einstellen anschliessen. Kabel für Debug-Schnittstelle einstecken und an der seriellen Schnittstelle am PC anschliessen.

2. Erstellen eines neuen Projektes mit TM, das den LED-Blinker nach vorherigem Beispiel beinhaltet.

3. Austesten, die LED-Blinkfrequenz beträgt ca. 2Hz.

4. Erweitern des Programmes so, dass ein Lauflicht über die LED D1..D8 entsteht.

Lösung zum Lauflicht mit direkten C-Statements und _asm()-Code:



Ablauf Lauflicht

Ports initialsieren
Muster 000'0001 laden
Do forever
warten
Muster auf Port1 ausgeben
Muster um eine Stelle linksrotieren