



---

# **μVersion 2**

## **Einführung**

---

**BULME**  **Höhere Technische  
Bundes-Lehr- und  
Versuchsanstalt  
BULME Graz – Götting**

V1.0

F. Wolf

Graz, September 2002

## Inhaltsverzeichnis

<b>1</b>	<b><i>Keil Mikroprozessor-Entwicklungsumgebung</i></b>	<b>1</b>
<b>2</b>	<b><i>Menübeschreibung</i></b>	<b>1</b>
2.1	Hauptmenü	2
2.2	Compilermenü	2
2.3	Debuggermenü	2
<b>3</b>	<b><i>Projekt erstellen</i></b>	<b>3</b>
3.1	Prozessorkonfiguration (Option of target)	7
3.2	Monitoreinstellungen	8
3.3	Programmsstart in der Zielhardware	9
<b>4</b>	<b><i>Simulation mit <math>\mu</math>Version2</i></b>	<b>10</b>

---

# $\mu$ Version 2

## Einführung

---

### 1 Keil Mikroprozessor-Entwicklungsumgebung

In dieser Einführung, zu der neuen integrierten Entwicklungsumgebung  $\mu$ Version 2, werden die wichtigsten Funktionen kurz beschrieben und soll Einsteigern bei der Erstellung eigener Projekte helfen

Bestandteile der Entwicklungsumgebungen sind:

- integrierte Entwicklungsumgebung (IDE)  $\mu$ Version 2
- C-Compiler
- Assembler
- Linker
- Debugger/Simulator

### 2 Menübeschreibung



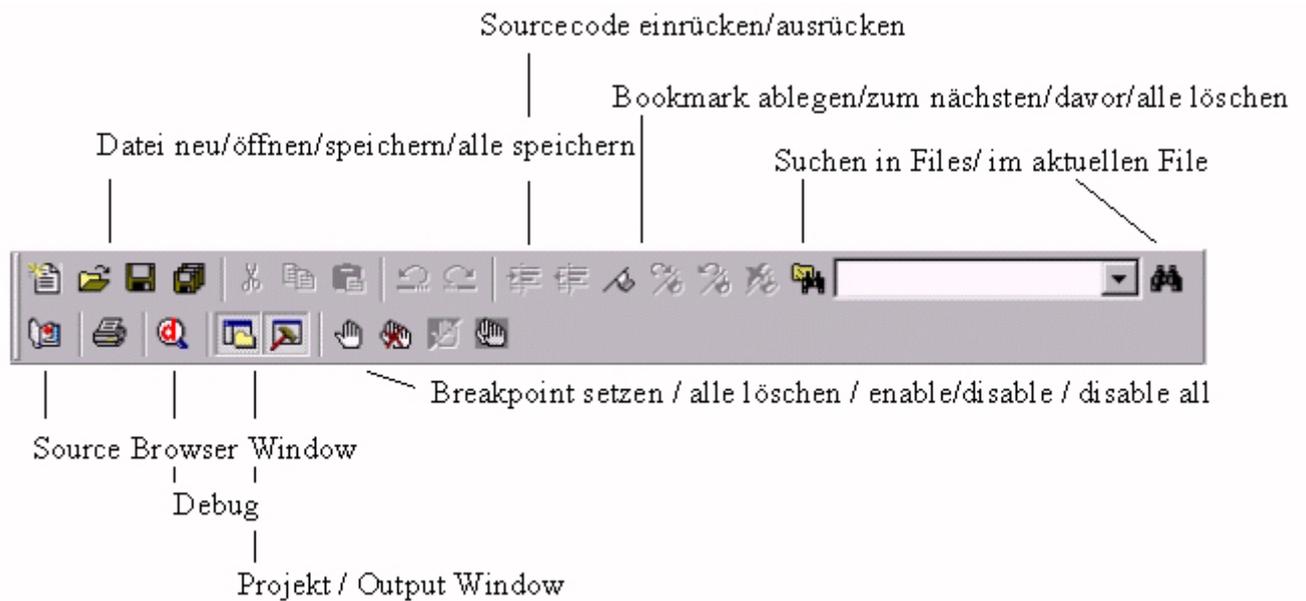
Uv2

Nach der Installation, wird das Programm mittels Doppelklick auf das Symbol gestartet.

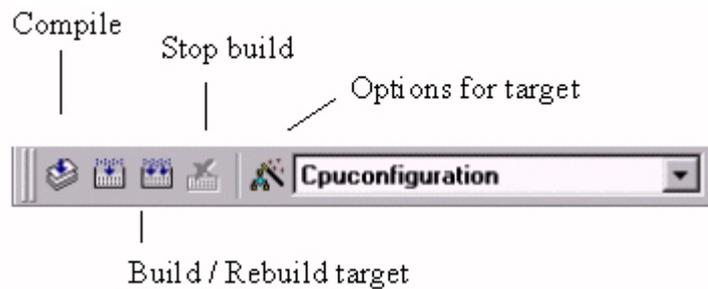
Danach wird die Oberfläche aufgebaut. Es folgt nun eine Beschreibung der wichtigsten Menüpunkte:

- Hauptmenü
- Compilermenü
- Debuggermenü

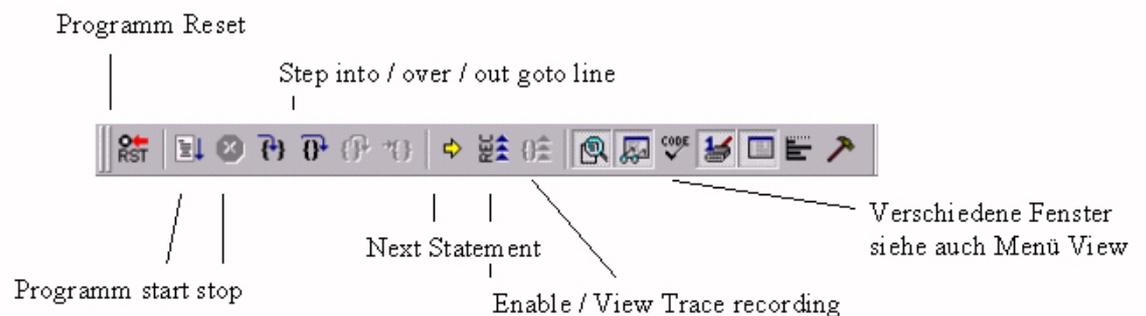
## 2.1 Hauptmenü



## 2.2 Compilermenü



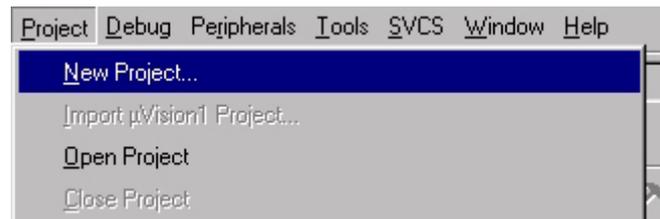
## 2.3 Debuggermenü



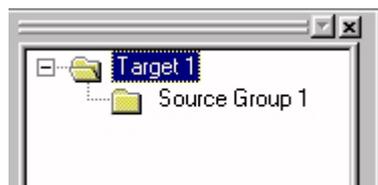
### 3 Projekt erstellen

Es wird nun ein Programm geschrieben, um den Ablauf von der Erstellung eines Projektes, bis zur Simulation von einzelnen Ports kennenzulernen.

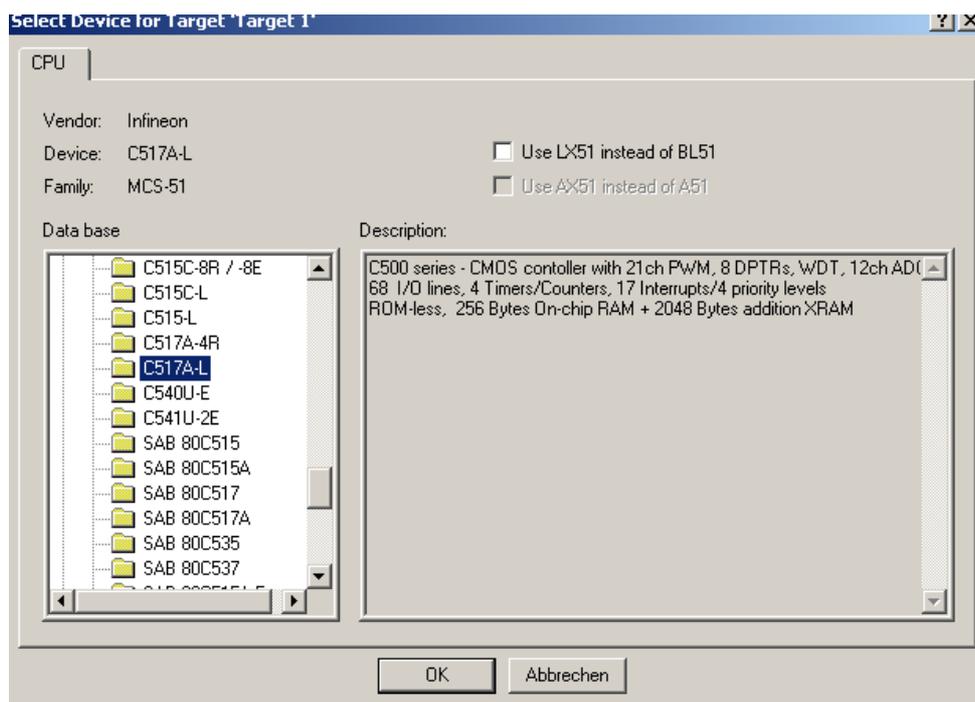
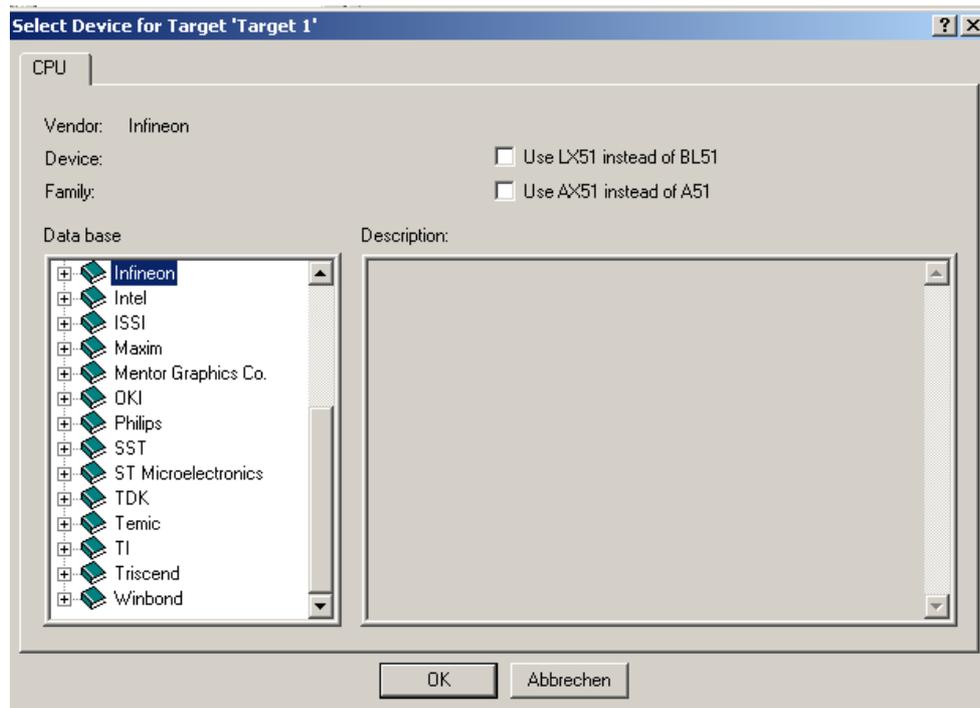
- Nachdem das Programm also gestartet wurde, öffnet man das Menü **Project** und wählt **New Project** aus



- Danach erscheint ein Fenster, indem man das neue Projekt mit einem Namen versieht und abspeichert. Man sollte erfahrungsgemäß alle Dateien, die zu einem Projekt gehören in einen Projektordner speichern, um so die Übersicht zu behalten. Nun erscheint der Name des Projekts am linken oberen Bildrand.



- Im Projekt Fenster erscheint nun Target 1 mit einer Verzweigung **Source Group1**. Falls das Projektfenster nicht sichtbar sein sollte, kann man es über **View -> Project Window** aktivieren. Als nächstes wird nun ein Name und ein Prozessor für das Beispiel gewählt.
- Dazu bewegt man die Maus im Projekt Fenster über **Target 1** und drückt die rechte Maustaste. Es erscheint ein Menü, in welchem man **Targets, Groups, Files** wählt. In dem Feld **Target to Add** gibt man einen Namen ein, z.B.: CpuConfiguration. Danach drückt man **Add** und wählt nun im Fenster darunter **CpuConfiguration**, drückt **Set as current target** und bestätigt mit **OK**. Wenn man dies erfolgreich erledigt hat, sollte nun CpuConfiguration anstatt Target1 im Projekt Fenster stehen.
- Um einen Prozessor Typ zu wählen, drückt man die rechte Maustaste über **CpuConfiguration** (vormals Target1), und wählt **Select Device for Target '....'**. Als nächstes kann man aus verschiedenen Herstellern und Typen wählen. Für die Übung wählen wir den **Infineon C517A-L**.



- Um nun ein kleines Programm zu schreiben, wählt man im Menü **File** -> **new** oder Ctrl - N. Es erscheint nun ein leeres Fenster in dem folgender C-Source Code eingegeben werden soll.

```

/* *****/
/*   Programm: port.c                               */
/*   Bitweiser Zugriff auf die LEDs an Port 4       */
/* *****/

/* Preprozessoranweisungen */

#include <reg517a.h>

/* Funktionen */

void warte(msec)
unsigned int msec;
{
    data unsigned char i,
        for (msec; msec!=0; msec--)
            for (i=0, i<255; i++);
}

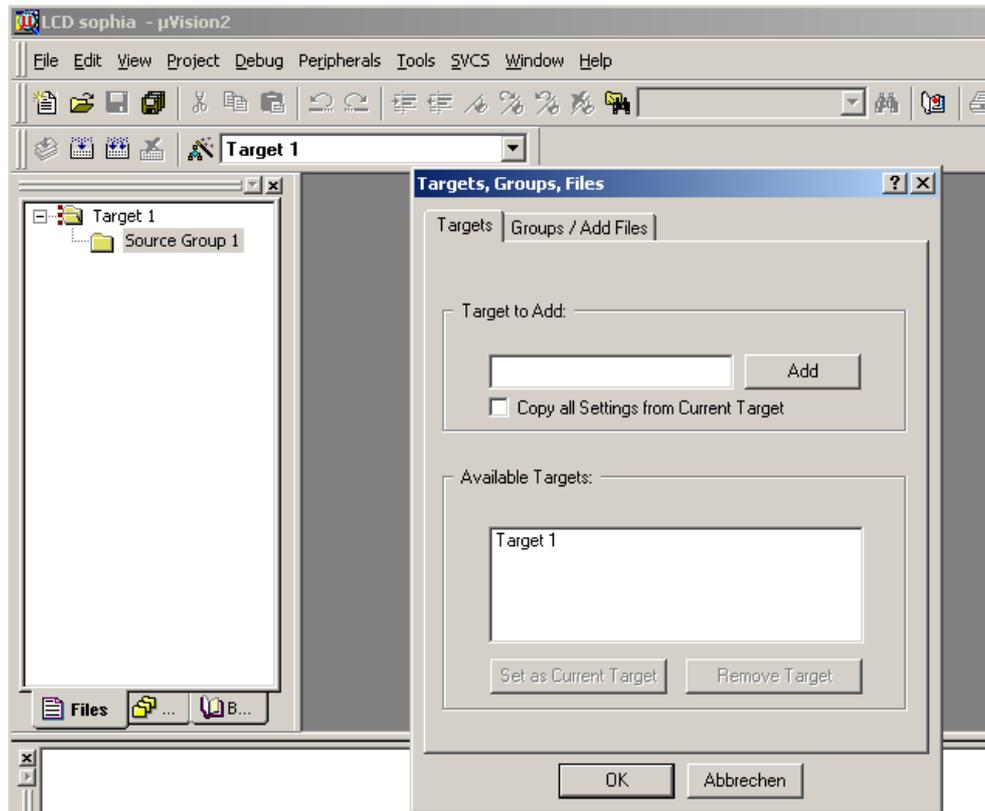
/* Hauptprogramm */

main()
{
    while(1)
    {
        P1=0;
        warte(1000);
        while (P1< 0xFF)
        {
            P1++;
            warte(500);
        }
    }
}

```

- Danach wird dieses Programm mit einem Namen und der Extension .c gespeichert (Save). z.B.: Port.c. Nachdem das Programm gespeichert wurde, wird es nun ins Projekt aufgenommen. Dazu fährt man mit der Maus auf **Source Group1** im Projekt Fenster und drückt die rechte Maustaste. Man wählt **Add Files to Source Group1** aus und im anschließendem Fenster den Namen des Files z.B.: Port.c. Abschließend drückt man **Add**, danach **Close** und das Programm ist im Projekt aufgenommen. Falls mehrere Module programmiert werden, kann man diese in verschiedenen Gruppen ablegen, um so die Übersicht bewahren und bei der Fehlersuche effizienter arbeiten.

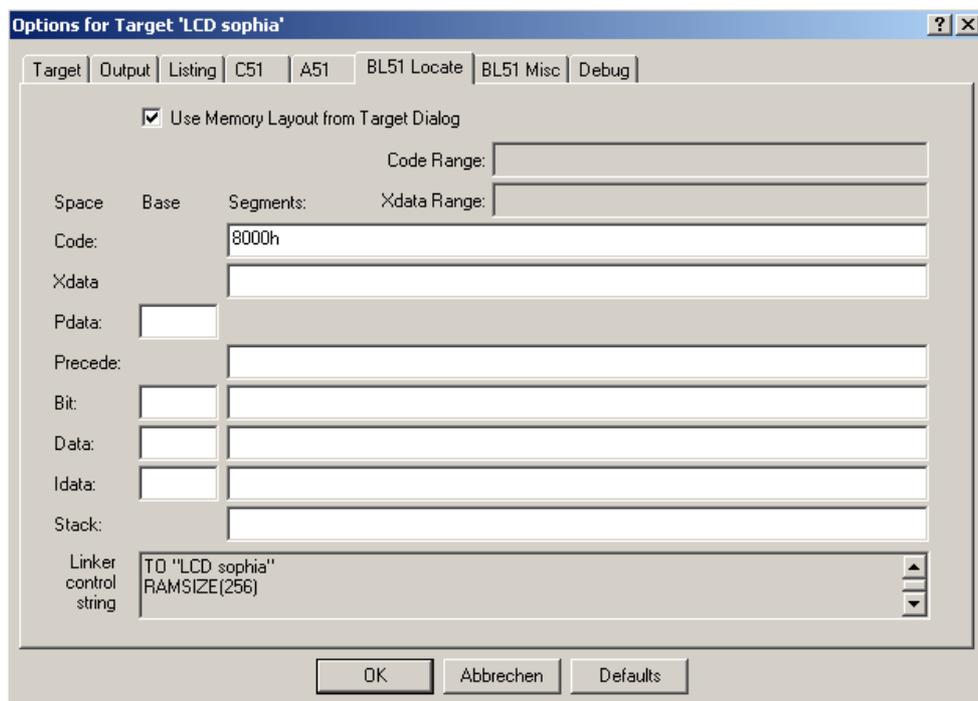
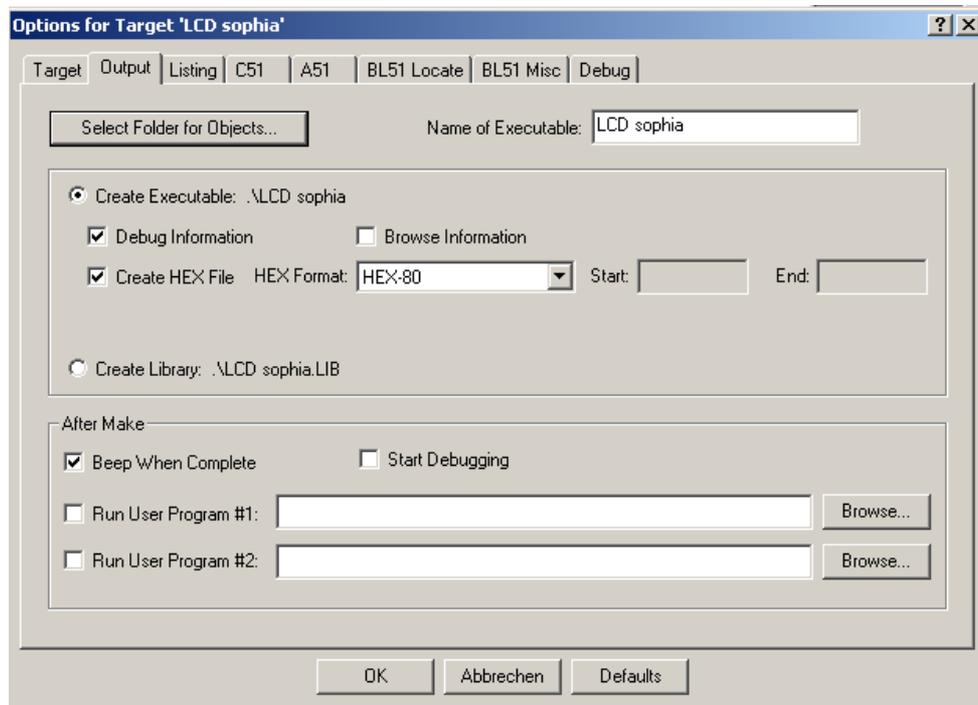
☛ Anmerkung: um z.B.: die LCD Anzeige zu benutzen muss man auch das vorgefertigte File LCD.c hinzufügen !!!



- Nach diesem Schritt, kann man das Programm mit **Build target** compilieren und linken. 
- Falls keine Fehler aufgetreten sind, kann man nun den Debugger starten, z.B.: mit Ctrl + F5, 

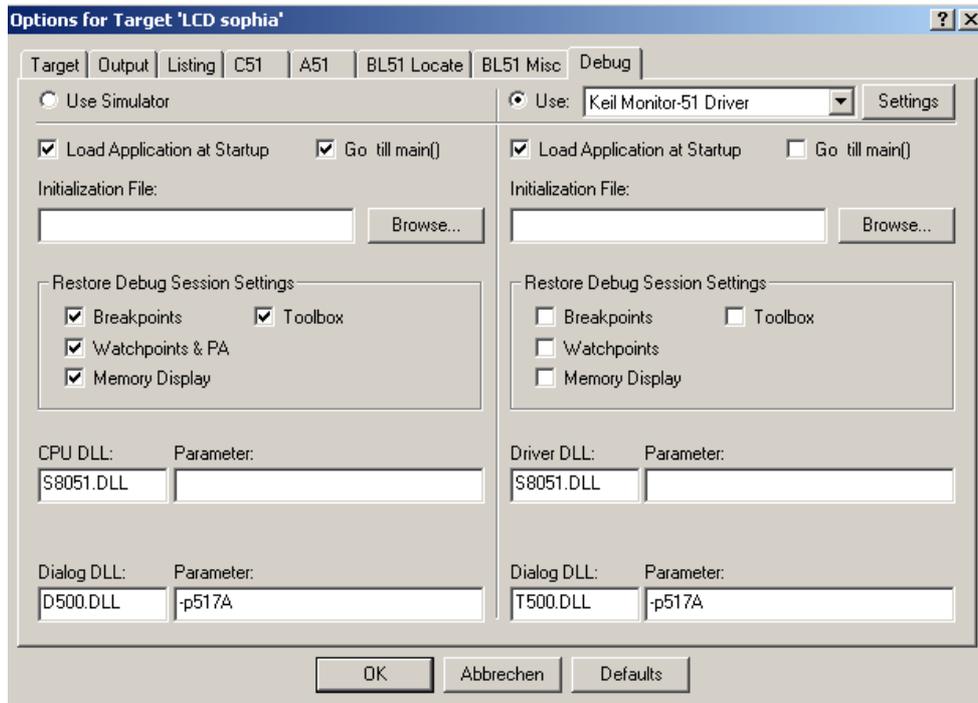
### 3.1 Prozessorkonfiguration (Option of target)

Hier sind folgende Einstellungen durchzuführen

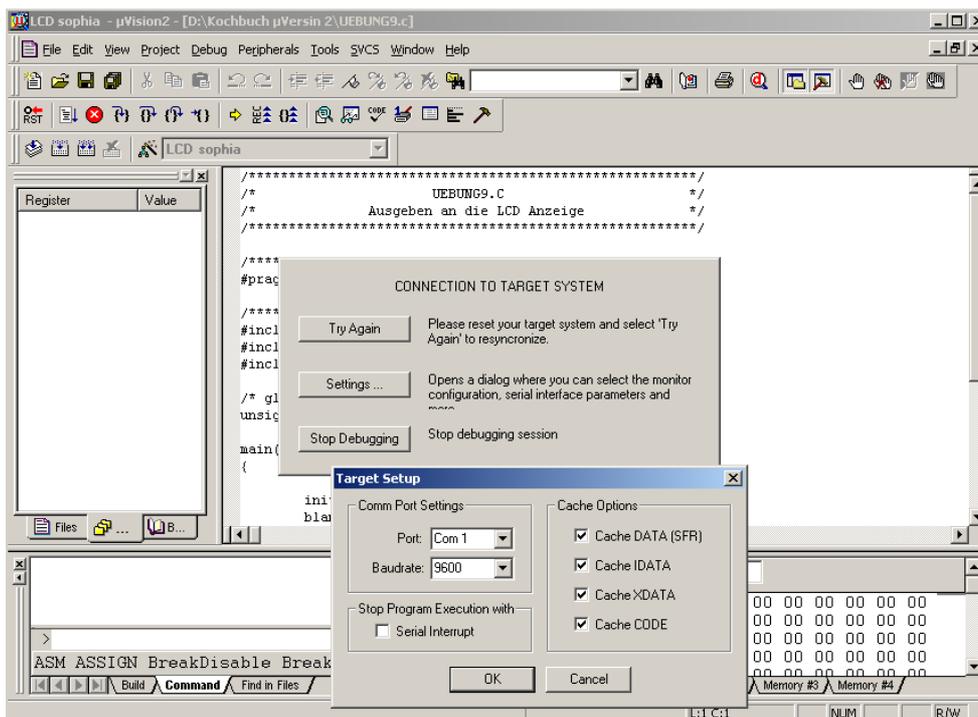


### 3.2 Monitoreinstellungen

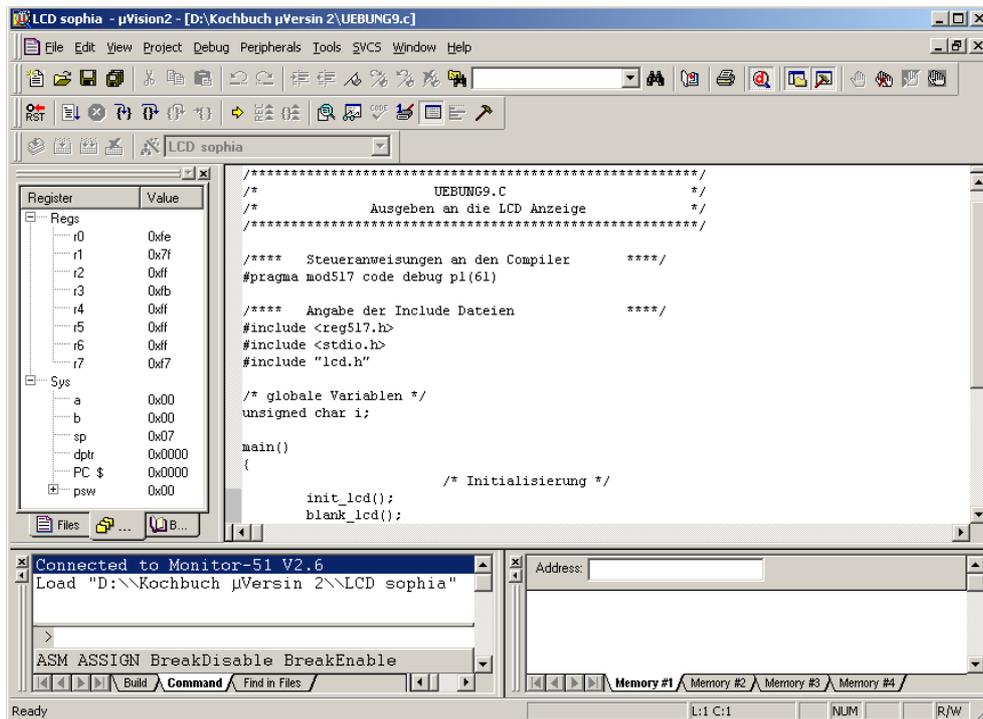
Hier wird beschrieben, wie diverse Monitoreinstellungen getroffen werden. Ein Monitor wird dazu benutzt, um z.B.: den Source Code in die Ram Chips der Hardware abzulegen. Ebenso kommuniziert µVision2 über diesen Monitor mit dem µComputer. In unsern Übungen verwenden wir den **Keil Monitor Driver**. Man aktiviert **Load Application at Startup** und **Go till main()**, um nach dem Laden des Programms, zum Beginn des Programms zu springen.



unter Settings befindet sich ein weiteres Fenster:

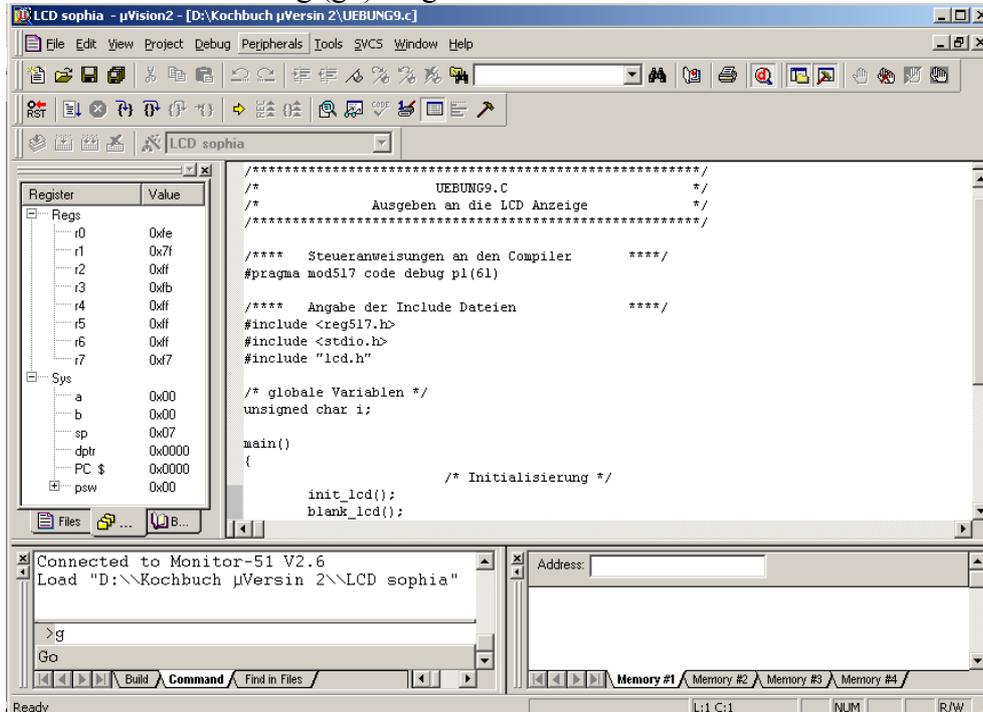


Tray Again auswählen und danach erscheint folgendes Bild



### 3.3 Programmstart in der Zielhardware

im Kommandofenster wird g (go) eingegeben

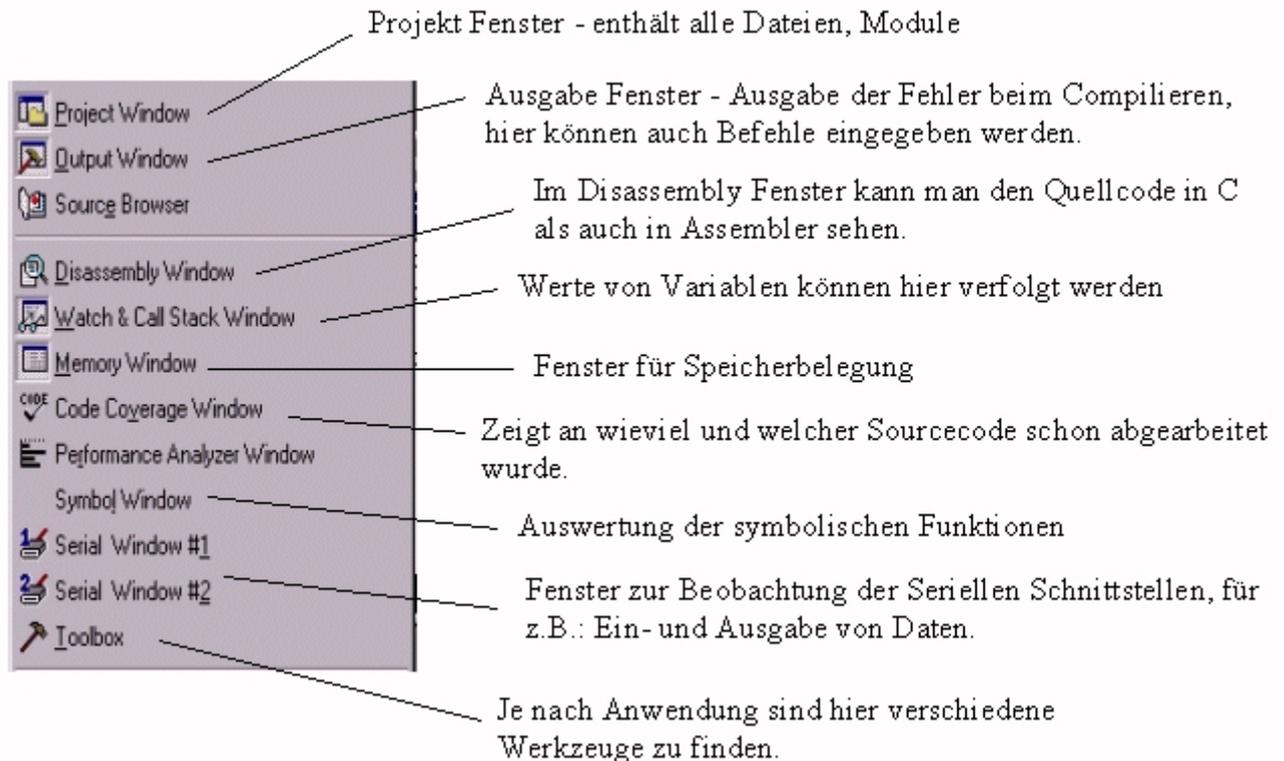


durch exit im Kommandofenster kann man den Debug Mode wieder verlassen !  
Danach kann wieder erneut der C File geändert werden.

Tipp: Vor den nächsten download in die Hardware, kann beim Verbindungsaufbau Tray Again Probleme geben. Abhilfe Netzstecker beim µC Board trennen

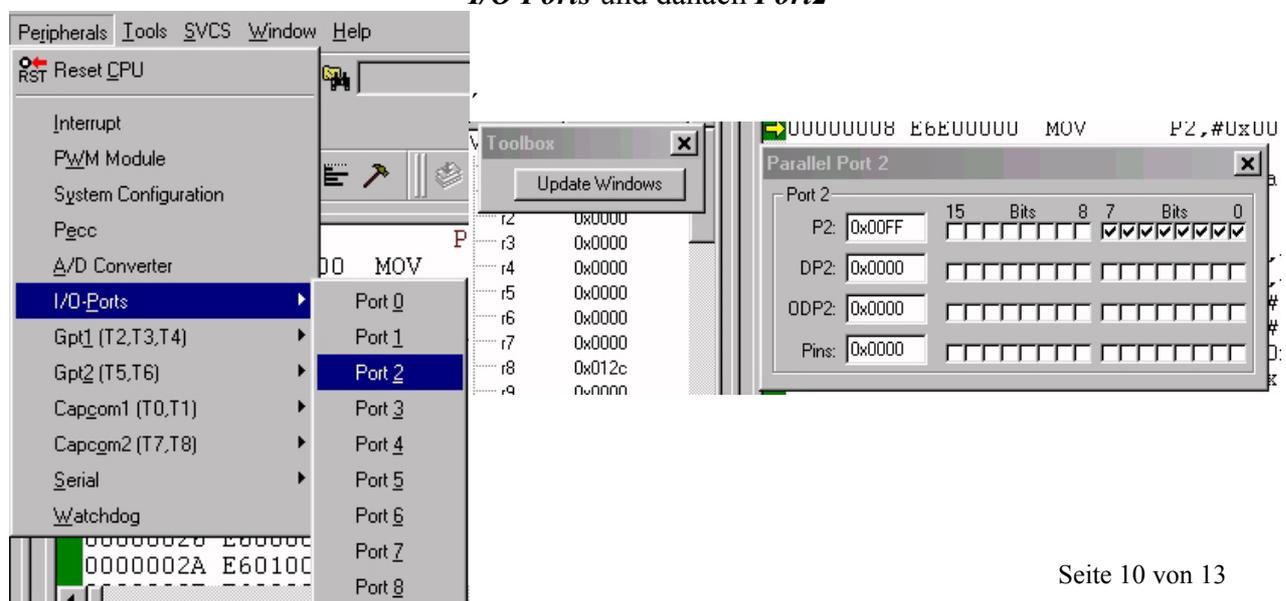
## 4 Simulation mit µVersion 2

Anders als bei µVision 1, wo zur Simulation das Programm Dscope gestartet werden mußte, erfolgt in µVision 2 die Simulation im gleichen Programm. Mit dem Starten des Debuggers, werden zahlreiche Fenster geöffnet und eine Reihe von Icons werden freigeschaltet. Diese können auch im Menü View gefunden werden. Hier eine kurze Beschreibung der Icons



Unter dem Menüpunkt **Peripherals**, sind weitere Icons integriert. Mit diesen können, je nach Programm, Timereinstellung, Portbeschaltungen, Interrupts, u.v.m. beobachtet werden. Da im Beispiel das Port 2 in einer Endlosschleife ein und ausgeschalten wird, will man dies nun auch beobachten.

Dazu öffnet man im Menüpunkt **Peripherals** den Punkt **I/O Ports** und danach **Port2**



Es erscheint ein Fenster mit dem Namen Parallel Port 2, indem der aktuelle Wert zu sehen ist.

Nun startet man das Programm mit  und stellt fest, daß sich die Werte nicht ändern. Um dem Abhilfe zu schaffen, drückt man auf das Icon mit dem Hammer (Toolbox). Es erscheint nun ein Button mit dem Namen Update Windows. Mit jedem Klick auf dieses Fenster werden nun die Werte aktualisiert. Wenn man aber im Single Step das Programm durchläuft, werden die Werte automatisch aktualisiert.

Mit  wird die Ausführung des Programms gestoppt und mit  wieder auf den Anfang zurückgesetzt.