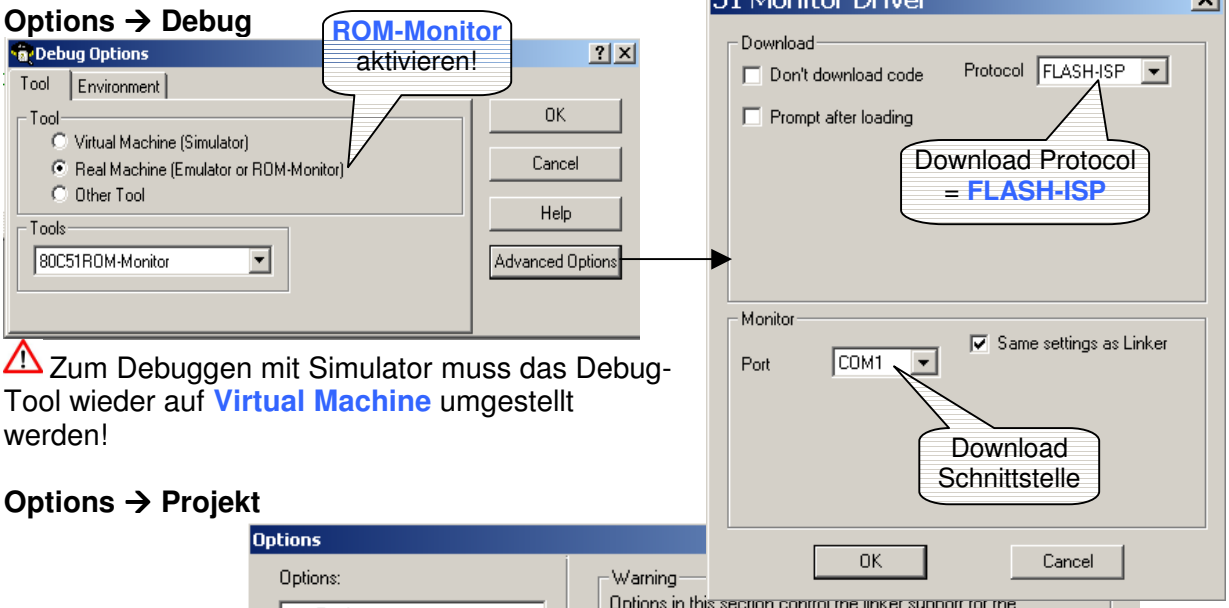
⚠ Das Im-System-Debuggen funktioniert nicht mit den USB-Controllern ⚠

1. Einstellungen in RIDE

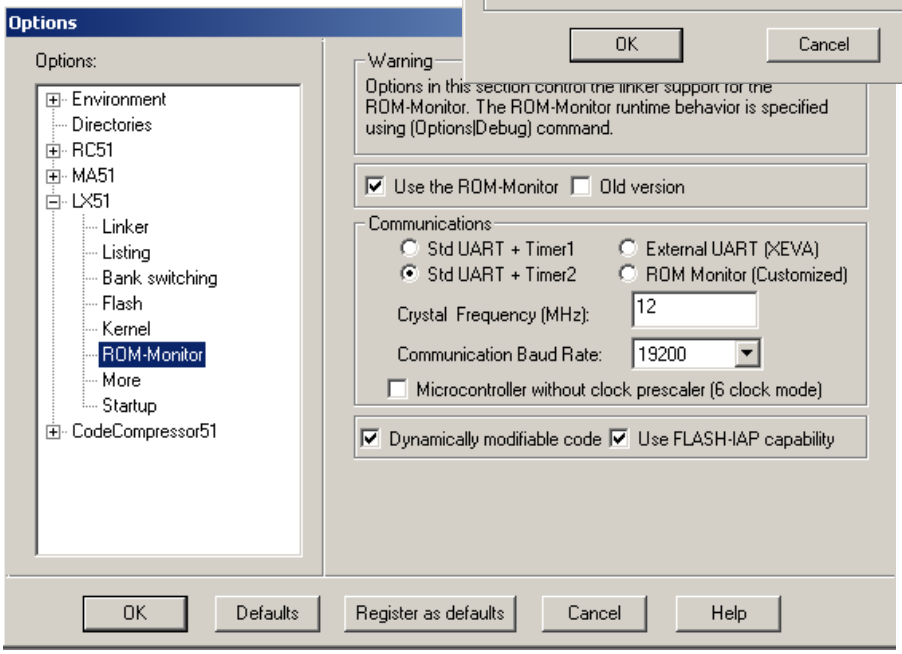
Das Debuggen ermöglicht ein Programmteil, der **ROM-Monitor** genannt wird. Der Linker fügt den ROM-Monitor dem eigenen Projekt hinzu.

Options → Debug

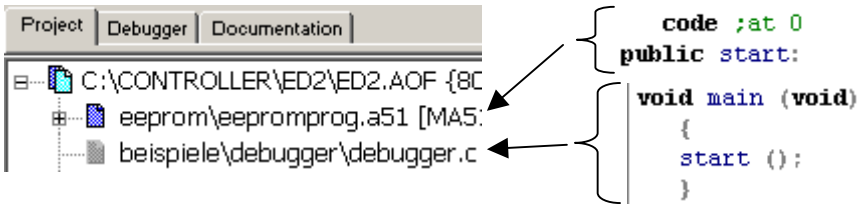


⚠ Zum Debuggen mit Simulator muss das Debug-Tool wieder auf **Virtual Machine** umgestellt werden!

Options → Projekt



Zum Debuggen von Assembler-Programmen muss dem Projekt ein C-Programm hinzugefügt werden, welches das Assemblerprogramm aufruft.



Im Assemblerprogramm muss at 0 (bei code at 0) auskommentiert werden. Das eigene Programm beginnt nicht mehr bei Adresse 0, sondern der ROM-Monitor, der dann das Assemblerprogramm aufruft.
Außerdem muss am Programmbeginn das Label **public start:** eingefügt werden. Diese kann dann vom C-Programm aufgerufen werden.

2. Debuggen

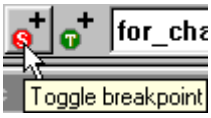
Beim Debuggen können die meisten Funktionen genutzt werden wie im Simulator. Nicht zu benutzen ist z.B. der Trace Befehl!

Vorgehensweise:

- Platine empfangsbereit schalten, gelbe LED muss leuchten. Reset drücken
- Debugger starten

- Bei Assembler-Programmen: einmal „Step into“, nun befindet man sich am Beginn des Assemblerprogramms.
- step over → Unterprogramme in einem Schritt ausführen

- step into → ins Unterprogramm springen

- Breakpoints setzen ist möglich. 

- Start des Programms (bis zum Breakpoint) in Echtzeit mit Go.

- Alle sichtbaren Fenster werden in Abhängigkeit von der Hardware aktualisiert.
- In den Fenstern geänderte Werte werden beim nächsten step der Hardware mitgeteilt.
- Alle Ports werden in der Hardware beeinflusst und abgefragt!
- Das Programm wird in der Hardware ausgeführt und nicht am PC simuliert.
- Sämtliche auf der Platine befindliche Hardware außer der seriellen Schnittstelle kann untersucht werden.