

# Software-Entwicklungsumgebung

Dipl.-Inf. (FH) Rainer Höckmann, Hochschule Osnabrück

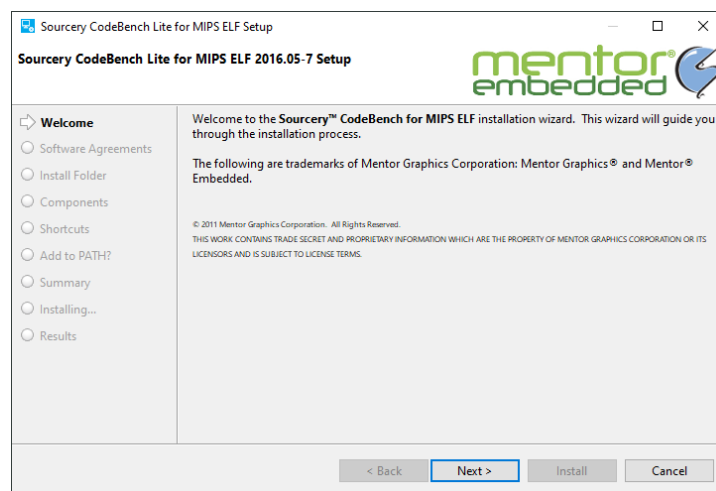
Diese Anleitung wurde erstellt für:

- Sourcery CodeBench Lite 2016.05-7
- Eclipse IDE for C/C++ Developers, Version 2020-12 (4.18.0)

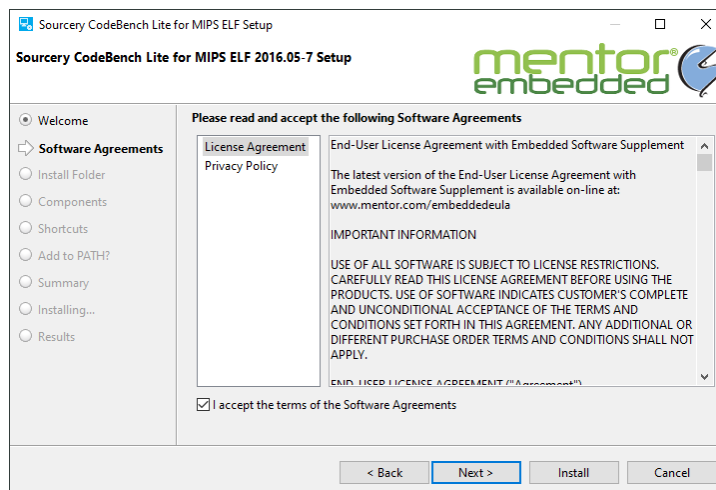
## Installation der GNU Cross-Compile-Toolchain

Die GNU Toolchain umfasst mehrere Programme, die bei der Erstellung und beim Debugging von Programmen verwendet werden. Dazu gehören zum Beispiel Assembler, Compiler, Linker und Debugger. Sie können sich die verwendete Toolchain bei der Firma Mentor Graphics herunterladen: <https://sourcery.mentor.com/GNUToolchain/package14479/public/mips-sde-elf/mips-2016.05-7-mips-sde-elf.exe>

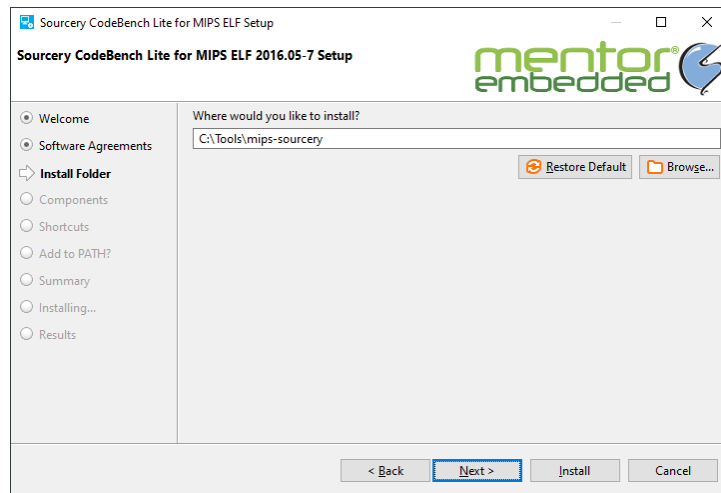
Nach dem Start des Installationsprogramms erscheint das folgende Fenster:



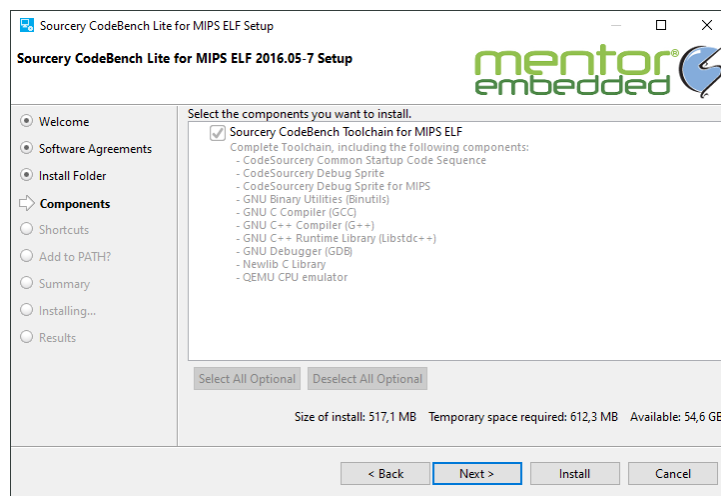
Klicken Sie auf [Next >].



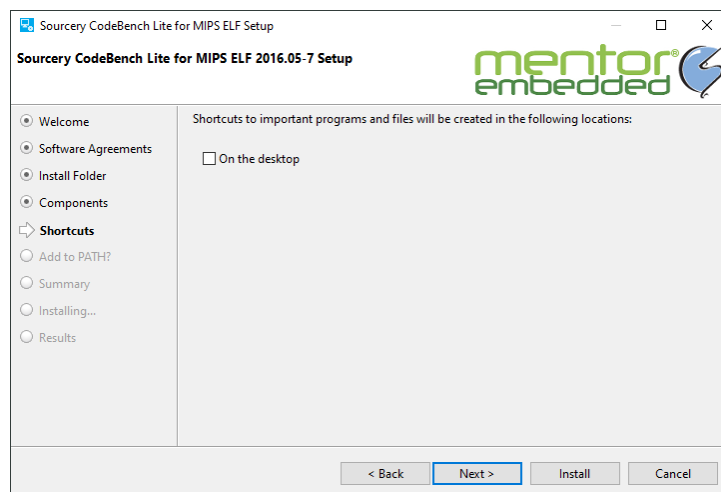
Setzen Sie den Haken bei „I accept the terms oft the Software Agreements“ klicken Sie auf [Next >].



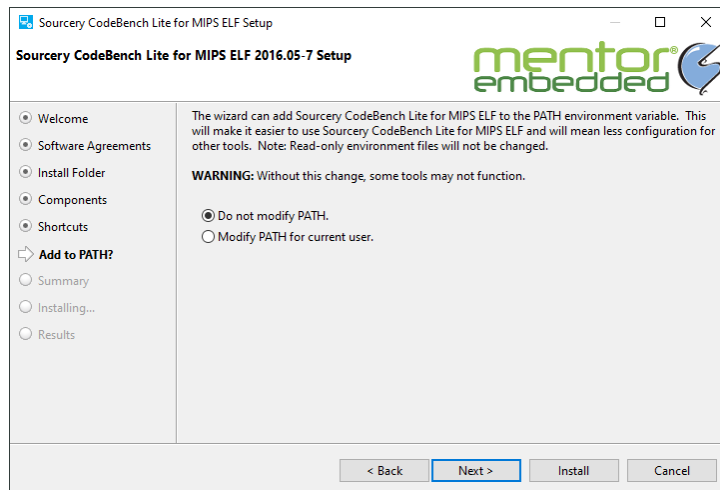
Geben Sie als Installationspfad „C:\Tools\mips-sourcery“ an. Klicken Sie auf [Next >].



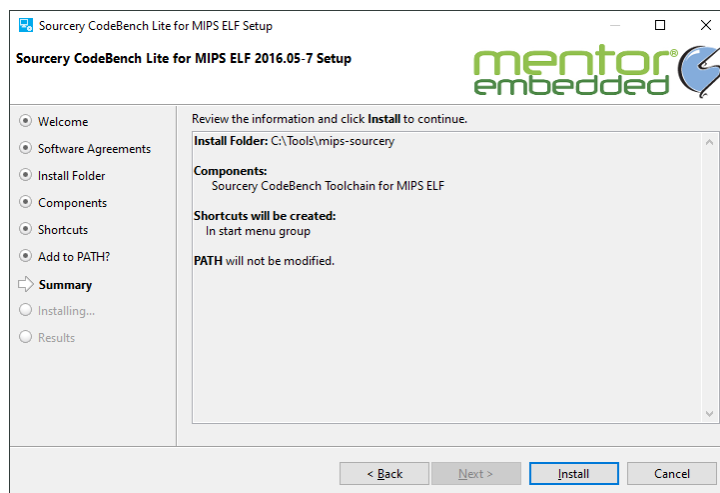
Klicken Sie auf [Next >].



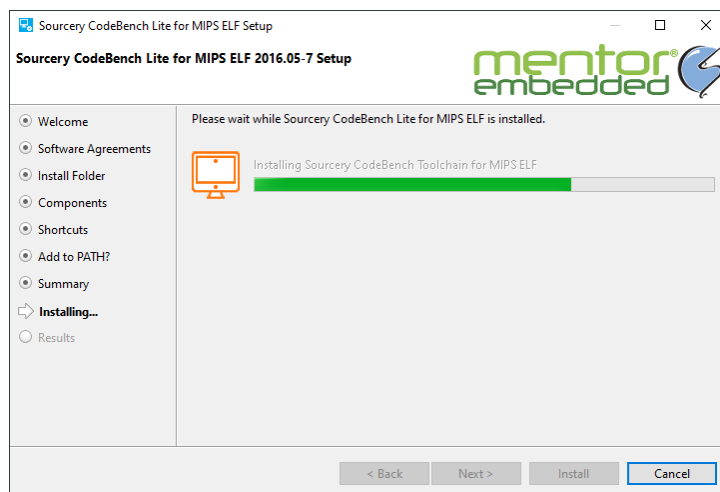
Entfernen Sie das Haken bei „On the Desktop“ und klicken Sie auf [Next >].



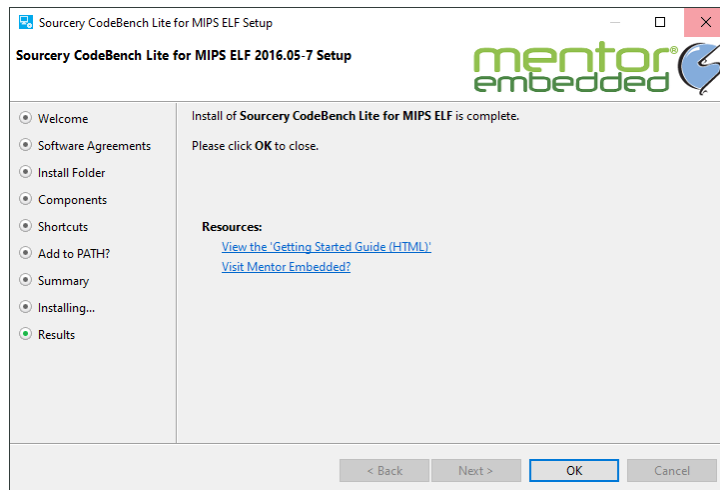
Setzen Sie den Punkt vor „Do not modify PATH“ und Klicken Sie auf [Next >].



Klicken Sie auf [Install].

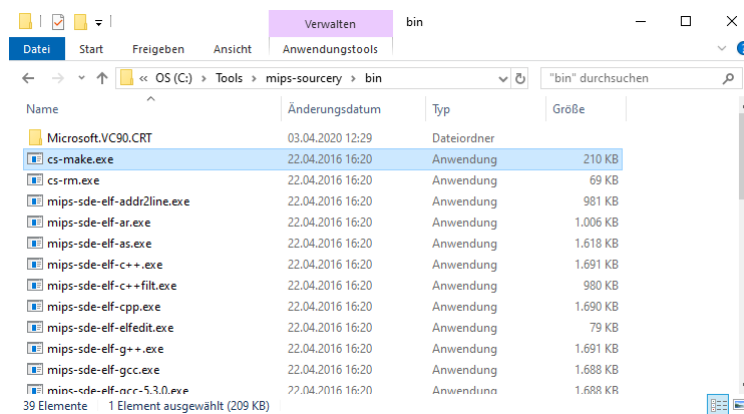


Warten Sie auf das Ende der Installation.

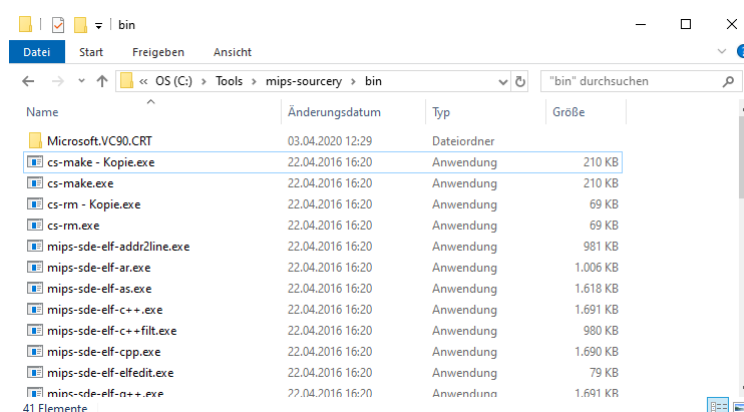


Klicken Sie auf [OK].

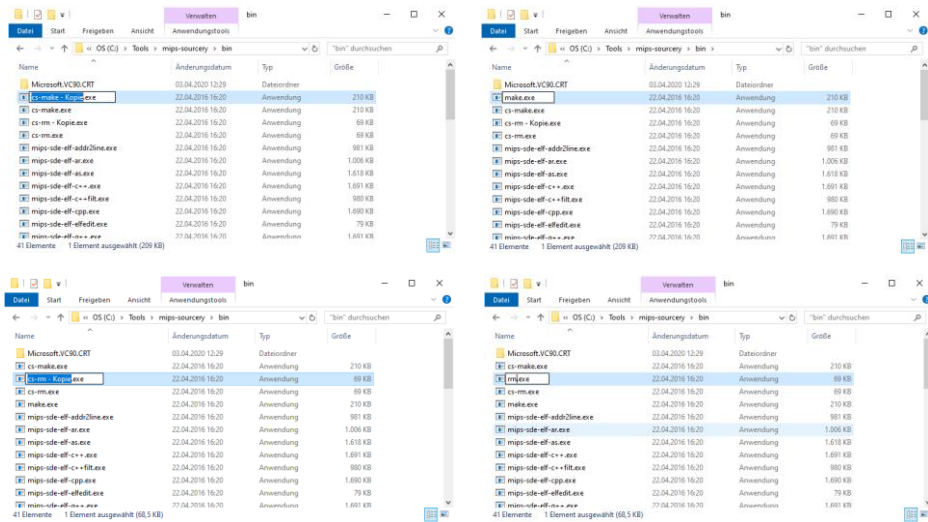
Öffnen Sie ein Windows-Explorer Fenster und wechseln Sie damit in das Verzeichnis „C:\Tools\mips-sourcery\bin“.



Erstellen Sie Kopien der Dateien „cs-make.exe“ und „cs-rm.exe“ indem Sie mit der rechten Maustaste auf die Dateien klicken, [Kopieren] klicken und anschließend mit der rechten Maustaste auf den freien Bereich hinter der letzten Spalte klicken und [Einfügen] klicken.



Benennen Sie nun die Datei „cs-make – Kopie.exe“ um zu „make.exe“ und die Datei „cs-rm – Kopie.exe“ zu „rm.exe“, indem Sie auf die Dateien mit der rechten Maustaste klicken und [Umbenennen] wählen.



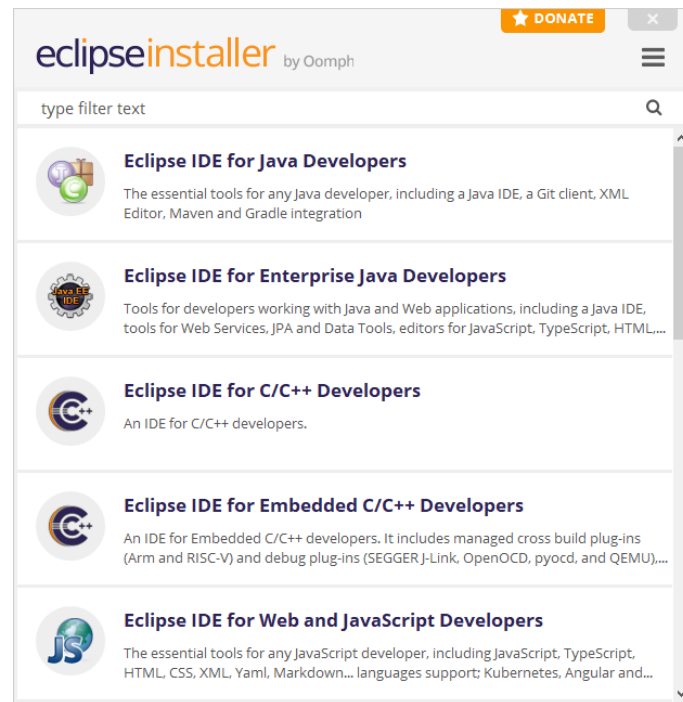
# Installation der Eclipse IDE for C/C++ Developers

Sie können sich die verwendete Version bei der Eclipse Foundation herunterladen:

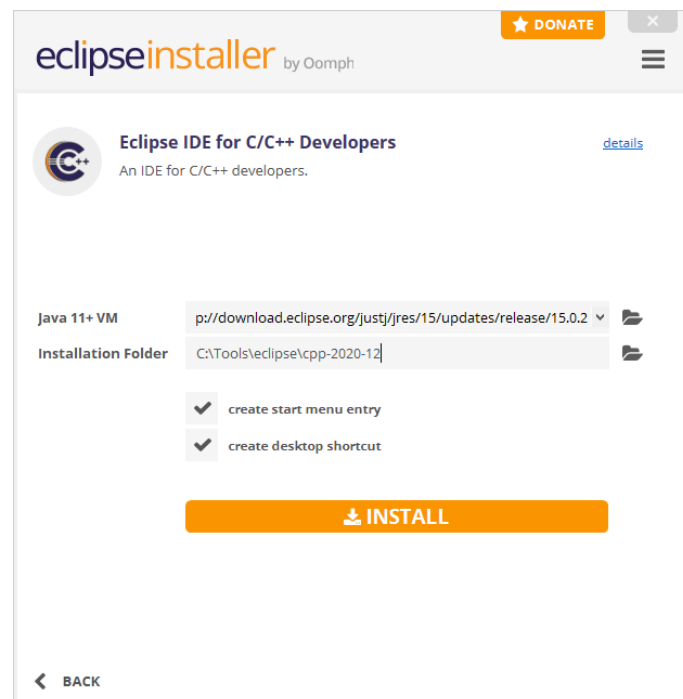
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2020-12/R/eclipse-inst-jre-win64.exe>

Der hier verlinkte Installer enthält eine zur Anwendung passende JAVA-Laufzeitumgebung (JRE).

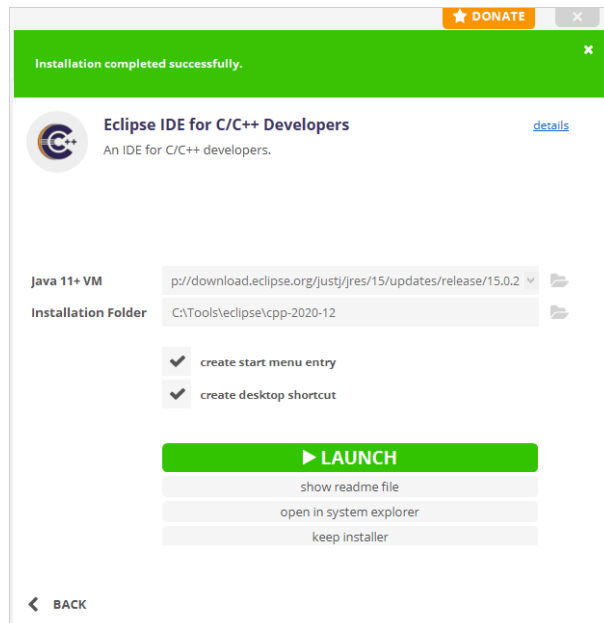
Nach dem Start des Installationsprogramms erscheint das folgende Fenster:



Wählen Sie das Paket „Eclipse IDE for C/C++ Developers“.

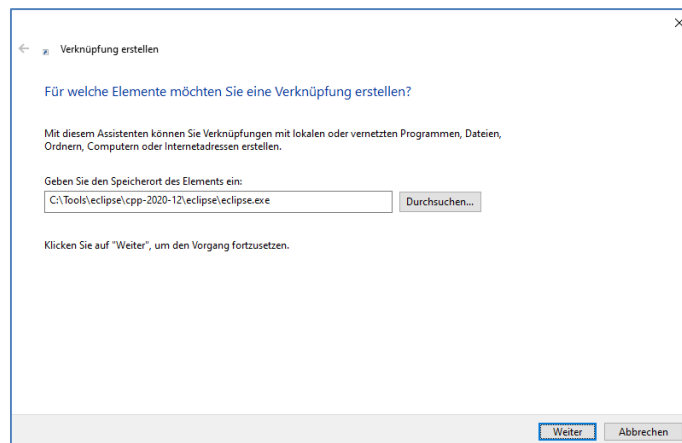


Setzen Sie den Installationspfad auf „C:\Tools\eclipse\cpp-2020-12“ und klicken Sie auf [INSTALL].

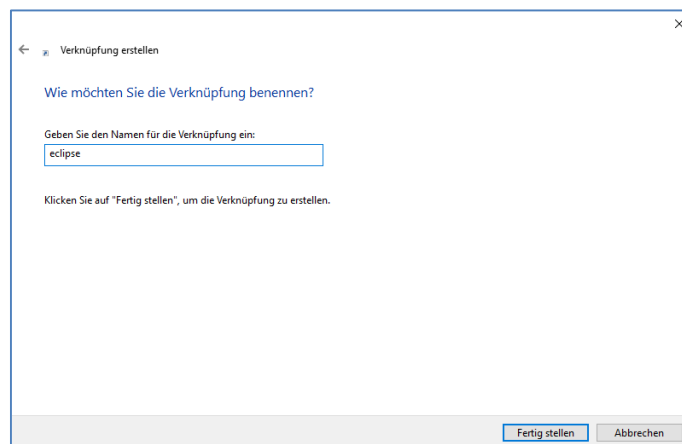


Schließen Sie das Fenster indem Sie auf die Schaltfläche [X] klicken.

Falls das Installationsprogramm keine Verknüpfungen angelegt hat, können Sie diese auch noch manuell erstellen. Klicken Sie dazu mit der rechten Maustaste auf den Desktop und wählen Sie [Neu]→[Verknüpfung].



Tragen Sie als „Speicherort des Elements“ ein: „C:\Tools\eclipse\cpp-2020-12\eclipse\eclipse.exe“ und klicken Sie auf [Weiter].



Tragen Sie als „Namen für die Verknüpfung“ ein „eclipse“ und klicken Sie auf [Fertig stellen].

# Anhang

## In den Projekten verwendete Einstellungen

- File > New > Project...
- Select a wizard
  - C/C++ - C/C++ Project
- Templates for New C/C++ Project
  - All - C Managed Build
- C Project
  - Project name: {Projektname ohne Leerzeichen}
  - Project type - Executable: Empty Project
  - Toolchains: Cross GCC
- Select Configurations
  - Debug
- Cross GCC Command
  - Cross compiler prefix: mips-sde-elf-
  - Cross compiler path: C:\Tools\mips-sourcery\bin
- Rechtsklick – New – Source Folder: Folder name: source
- Source-Dateien und ldscrip.ld in den Source Folder kopieren
- Rechtsklick – Properties
- C/C++ Build
  - Behavior
    - Häkchen bei „Build on resource save (Auto build)“
  - Refresh Policy
    - Projektverzeichnis unter Resources eintragen
  - Environment
    - PATH - Edit...:
    - Am Anfang einfügen: C:\Tools\mips-sourcery\bin;
  - Settings – Tool Settings:
    - Cross GCC Compiler - Miscellaneous – Other flags:  
-mips1 -EL -fno-delayed-branch -mno-check-zero-division -ggdb
    - Cross GCC Compiler - Includes:  
"\${workspace\_loc}/\${ProjName}/source}"
    - Cross GCC Linker – General:  
Haken setzen:
      - Do not use standard start files (-nostartfiles)
      - Do not use default libraries (-nodefaultlibs)
      - No startup or default libs (-nostdlib)
      - No shared libraries (-static)
    - Cross GCC Linker – Miscellaneous – Other options:  
-T\${ProjDirPath}/source/ldscrep.ld -EL
    - Cross GCC Assembler – General – Assembler flags:  
-mips32r2 -EL -g -gstabs+
  - Build Steps:
    - Post-build-steps – Command:  
mips-sde-elf-objdump.exe -h -j.text -j.data -j.bss \${ProjName}.elf & mips-sde-elf-objdump.exe -h -t -j.text -j.data -j.bss -S \${ProjName}.elf > \${ProjName}\_diss.txt & mips-sde-elf-objcopy.exe -O ihex \${ProjName}.elf \${ProjName}.hex
    - Build Artifact:  
Artifact name: \${ProjName}  
Artifact extension: elf

Bei Build-Fehlern Bei den kopierten Optionen prüfen, ob wirklich '-'-Zeichen verwendet wurden!