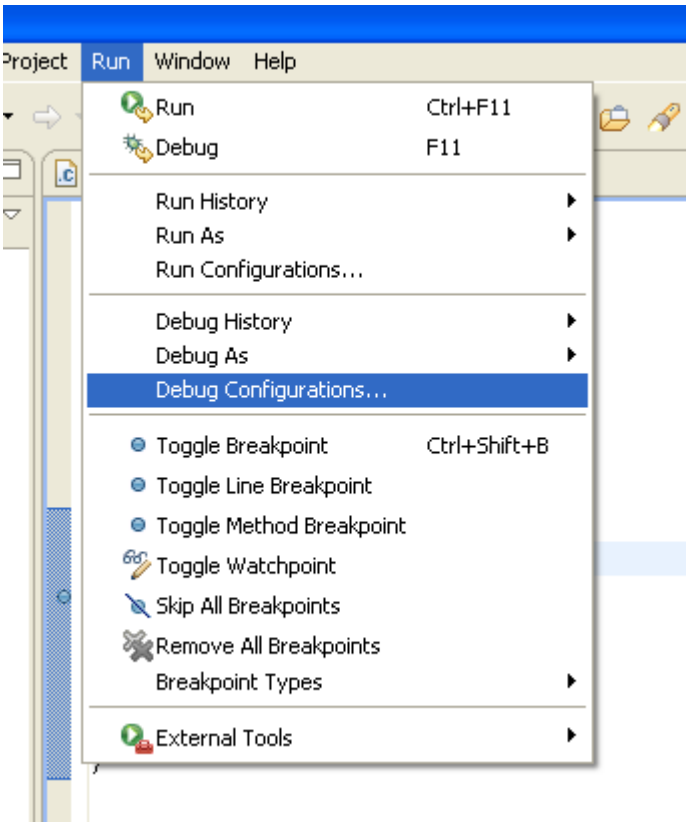
HOWTO:

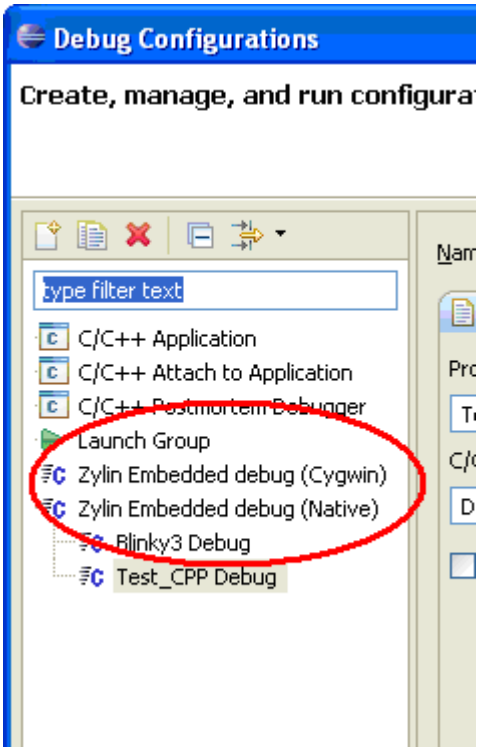
Konfiguration OpenOCD für Luminary Micro's EK-LM3S2965 Kit

Version 0.1 15.03.2010

Board - EK-LM3S2965
Eclipse 3.5 CDT
Zilyn 4.10
OpenOCD 0.4.0

Prüfen, ob Zilyn installiert:

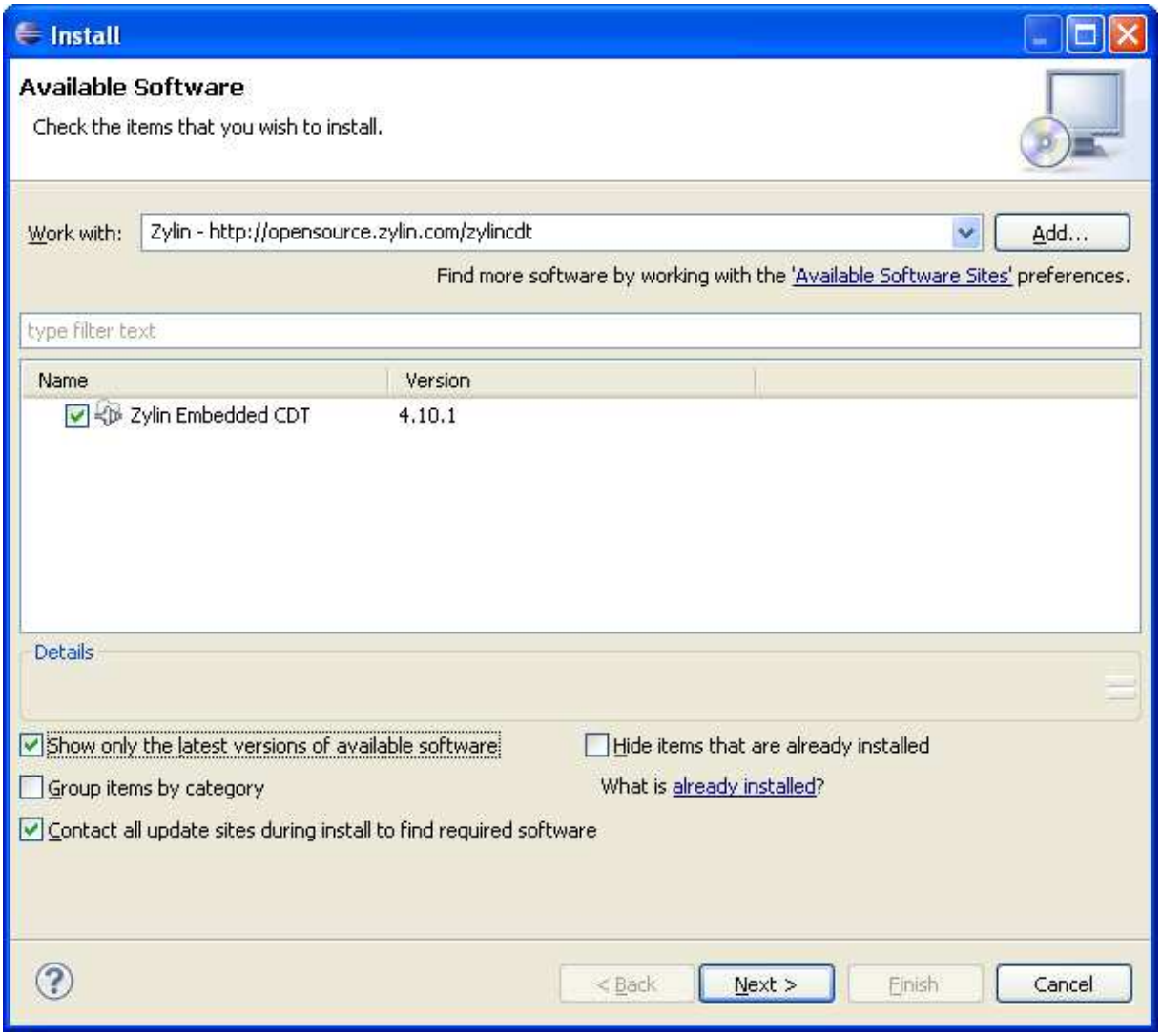




Wenn nicht, dann holen und installieren:

Help->Install New Software -> Add

Zylin - http://opensource.zylin.com/zylincdt



2. OpenOCD installieren, entweder von Quellcode

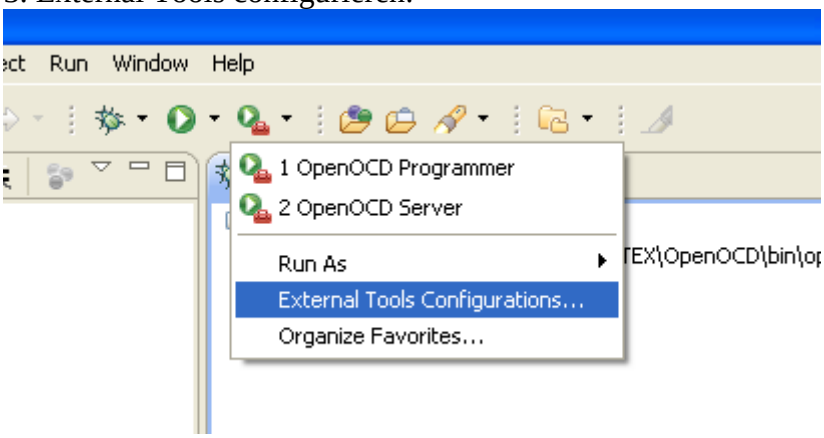
http://openocd.berlios.de (lang)

oder mit vorcompilierte Version:

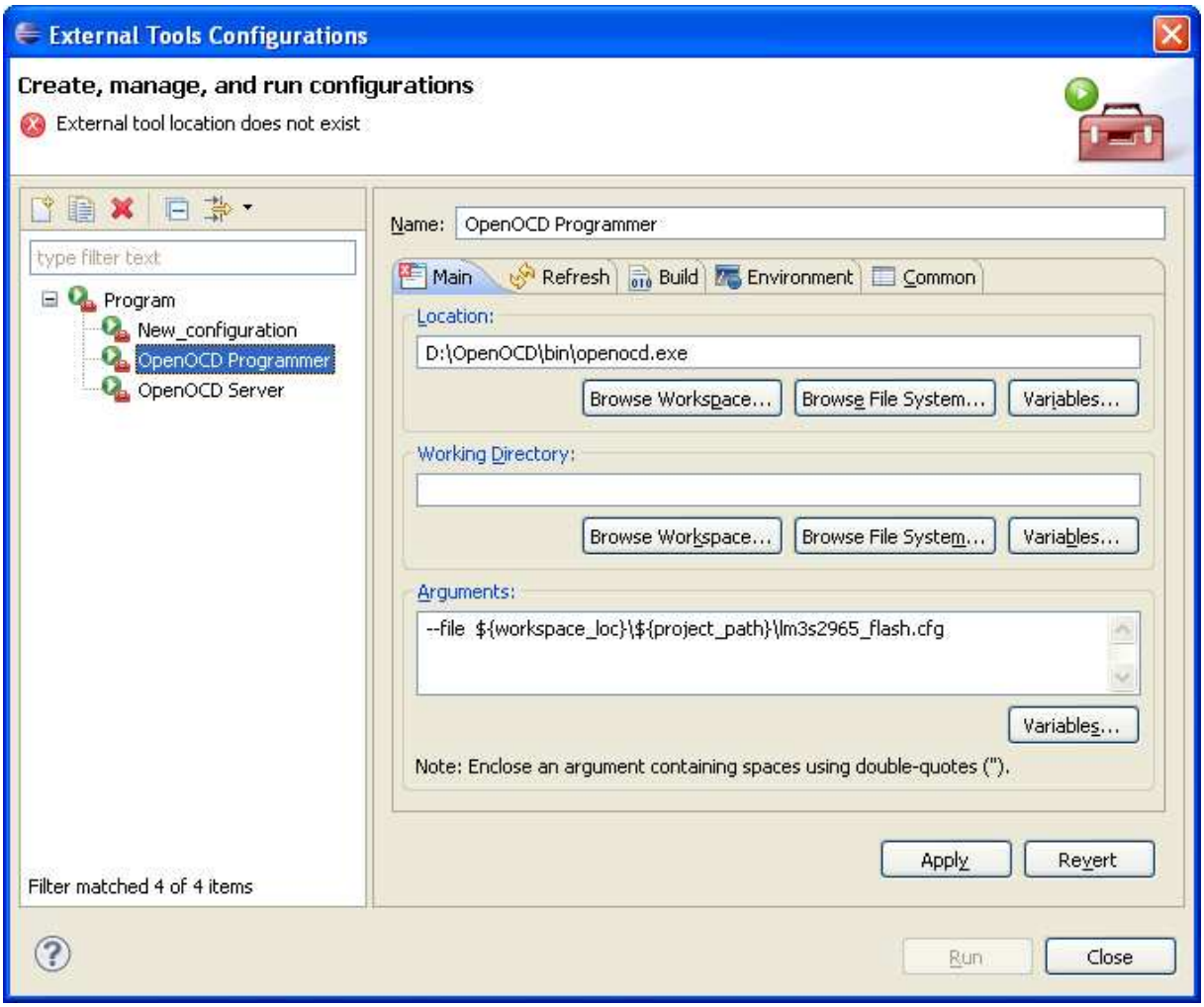
http://www.freddiechopin.info/index.php/en/download/category/4-openocd?download=34%3Aopenocd-040 (schnell)

Wichtig:
Archivdatei „libusb-win32_ft2232_driver-100223.zip“ auspacken und Treiber für Entwicklungsboard ersetzen.

3. External Tools configurieren:



3.1 OpenOCD Programmer konfigurieren



Name: OpenOCD Programmer
Location: Pfad zu openocd.exe

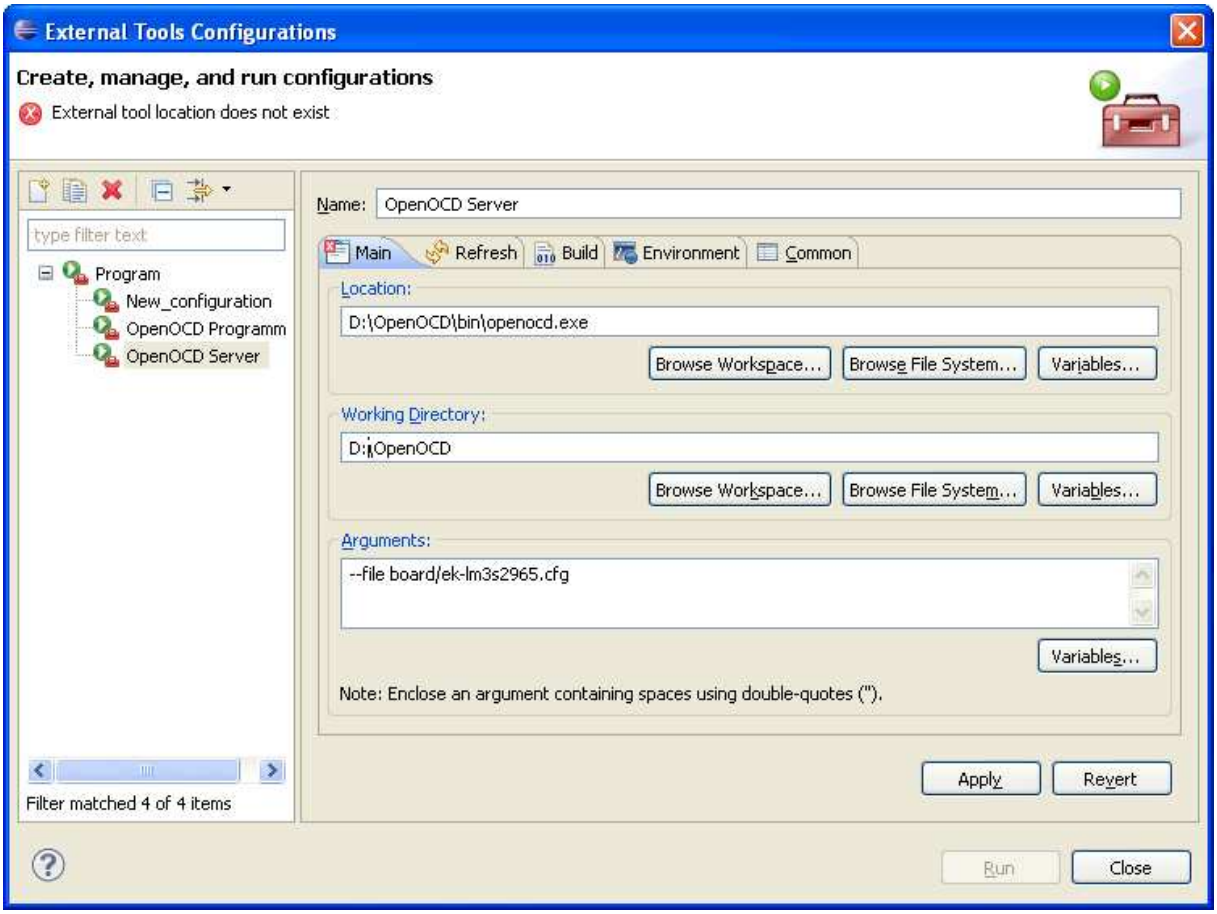
Main:
--file ${workspace_loc}\${project_path}\lm3s2965_flash.cfg

Datei lm32965_flash.cfg in Project Verzeichnis kopieren.

```
# script for luminary lm3s2965

set  _CHIPNAME lm3s2965

# this defaults to a little endian
set  _ENDIAN little

# force an error till we get a good number
set _CPUTAPID 0x3ba00477

# set interface
interface ft2232
ft2232_layout evb_lm3s811
ft2232_vid_pid 0x0403 0xbcd9

jtag_nsrst_delay 100
jtag_ntrst_delay 100

#jtag scan chain
jtag newtap $_CHIPNAME cpu -irlen 4 -ircapture 1 -irmask 0xf -expected-id
$_CPUTAPID

# the luminary variant causes a software reset rather than asserting SRST
# this stops the debug registers from being cleared
# this will be fixed in later revisions of silicon

set _TARGETNAME [format "%s.cpu" $_CHIPNAME]
target create $_TARGETNAME cortex_m3 -endian $_ENDIAN -chain-position
$_TARGETNAME -variant lm3s

# 4k working area at base of ram
$_TARGETNAME configure -work-area-virt 0 -work-area-phys 0x20000000 -work-
area-size 0x4000 -work-area-backup 0

#flash configuration
set _FLASHNAME $_CHIPNAME.flash
flash bank $_FLASHNAME stellaris 0 0 0 0 $_TARGETNAME

targets

# --------------- now do some flashing -----------------
init
reset halt
flash probe 0
stellaris mass_erase 0
flash write_image _FULLPATH_/Blinky/debug/blinky.hex 0 ihex
reset run

shutdown
```

Kartei Build:
Build before launch -> Hacken weg

3.2. OpenOCD Server:

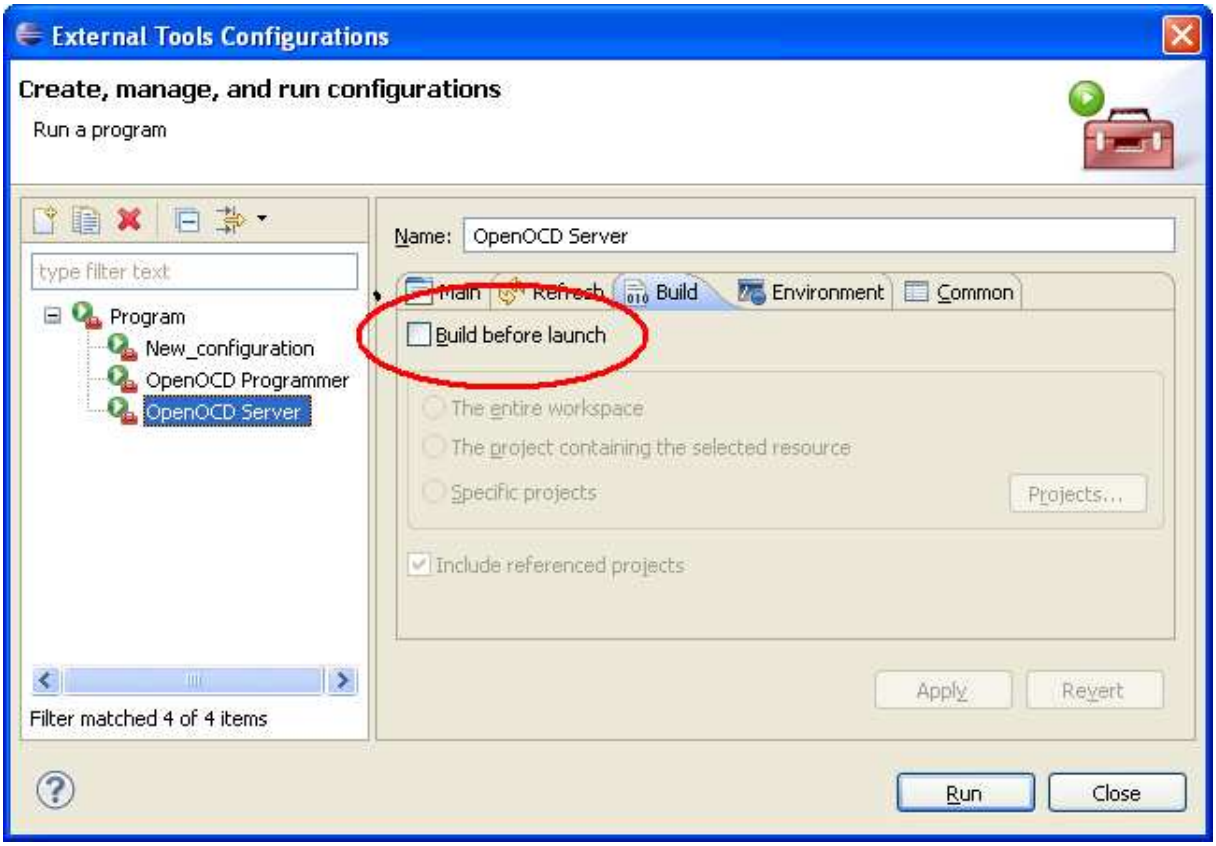


Name: OpenOCD Server
Location: Pfad zu openocd.exe
Arguments:

--file board/ek-lm3s2965.cfg

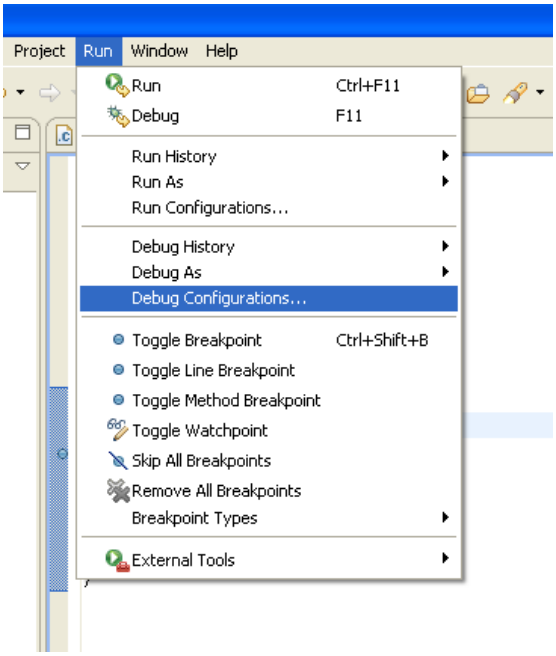
```
# TI/Luminary Stellaris LM3S2965 Evaluation Kits
# http://www.luminarymicro.com/products/lm3s2965_evaluation_kits.html

# include the FT2232 interface config for on-board JTAG interface
source [find interface/luminary.cfg]

# include the target config
source [find target/lm3s2965.cfg]

# jtag speed
jtag_khz 3000

jtag_nsrst_delay 100

reset_config srst_only
```

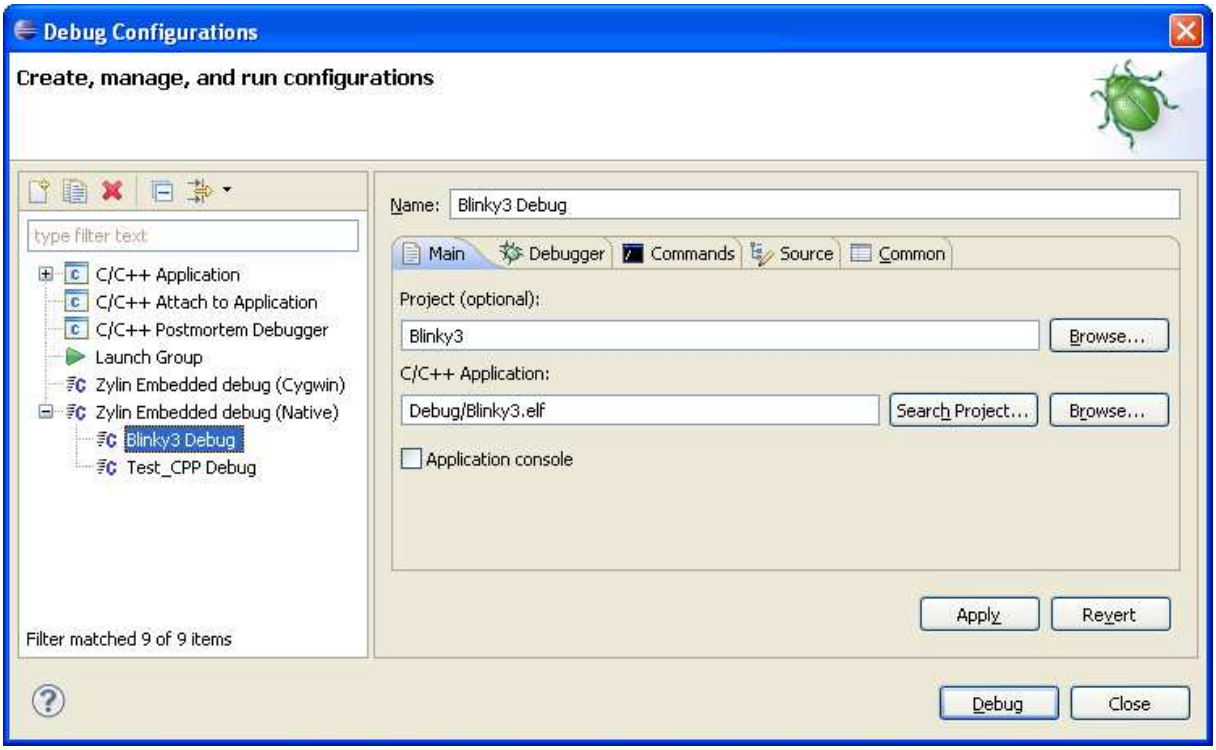
Datei target/lm3s2965.cfg ist ein Kopie von lm3s6965.cfg

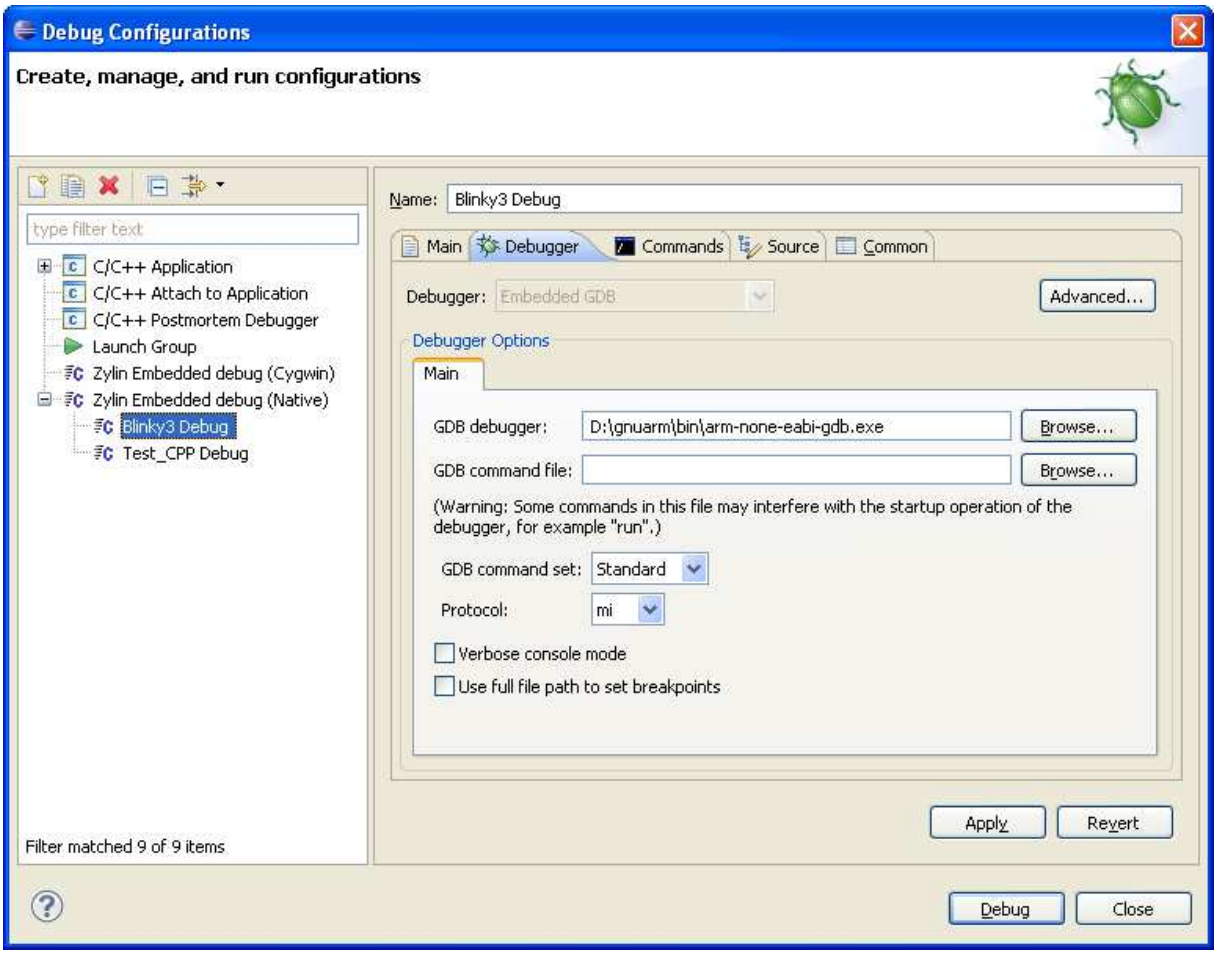


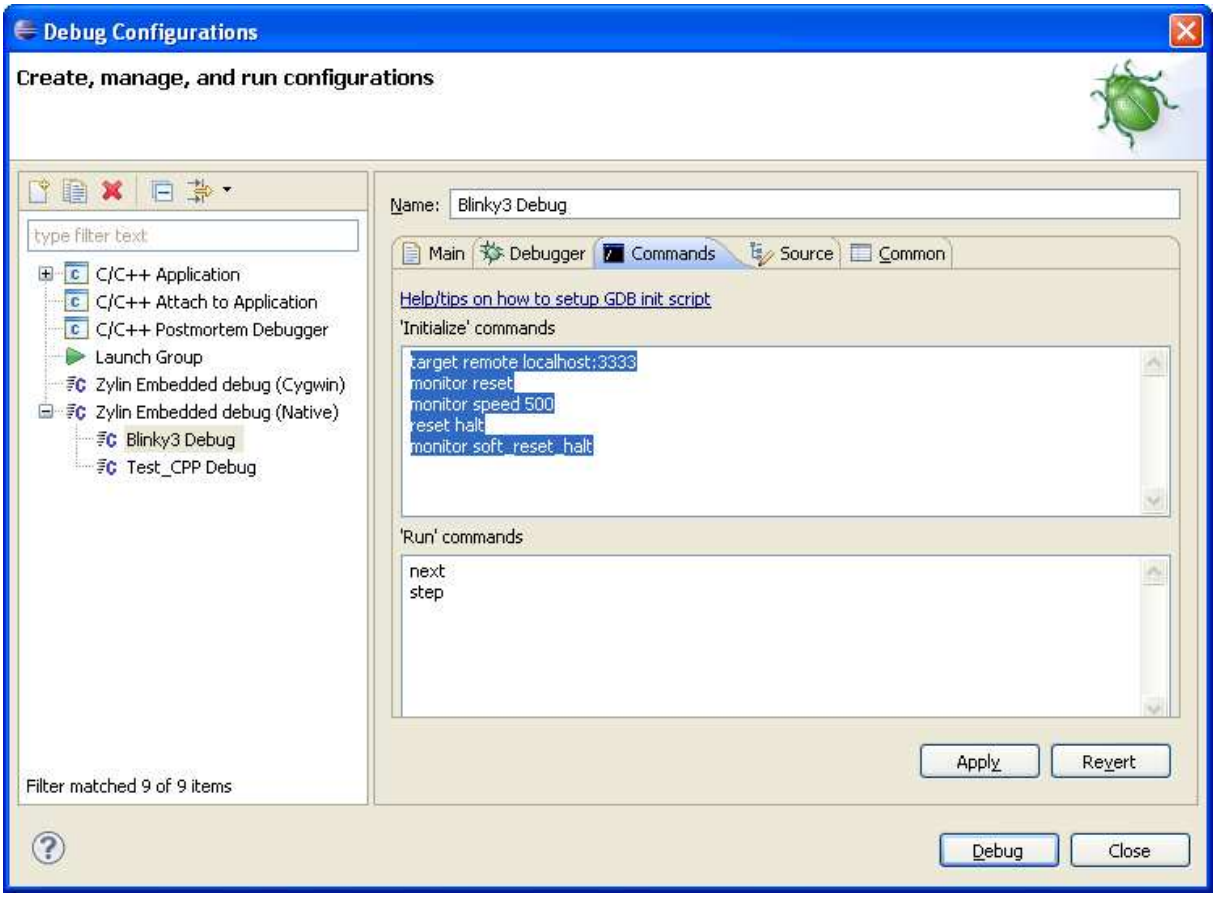
4. Debug Konfigurieren

Project auswählen, dann




Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Postmortem Debugger
Launch Group
Zylin Embedded debug (Cygwin)
Zylin Embedded debug (Native)
Blinky3 Debug
Test_CPP Debug
Filter matched 9 of 9 items
Name: Blinky3 Debug
Main
Debugger
Commands
Source
Common
Project (optional):
Blinky3
Browse...
C/C++ Application:
Debug/Blinky3.elf
Search Project...
Browse...
Application console
Apply
Revert
Debug
Close


Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Postmortem Debugger
Launch Group
Zylin Embedded debug (Cygwin)
Zylin Embedded debug (Native)
Blinky3 Debug
Test_CPP Debug
Filter matched 9 of 9 items
Name: Blinky3 Debug
Main
Debugger
Commands
Source
Common
Debugger: Embedded GDB
Advanced...
Debugger Options
Main
GDB debugger: D:\gnuarm\bin\arm-none-eabi-gdb.exe
Browse...
GDB command file:
Browse...
(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)
GDB command set: Standard
Protocol: mi
Verbose console mode
Use full file path to set breakpoints
Apply
Revert
Debug
Close



„Initialize“ commands:

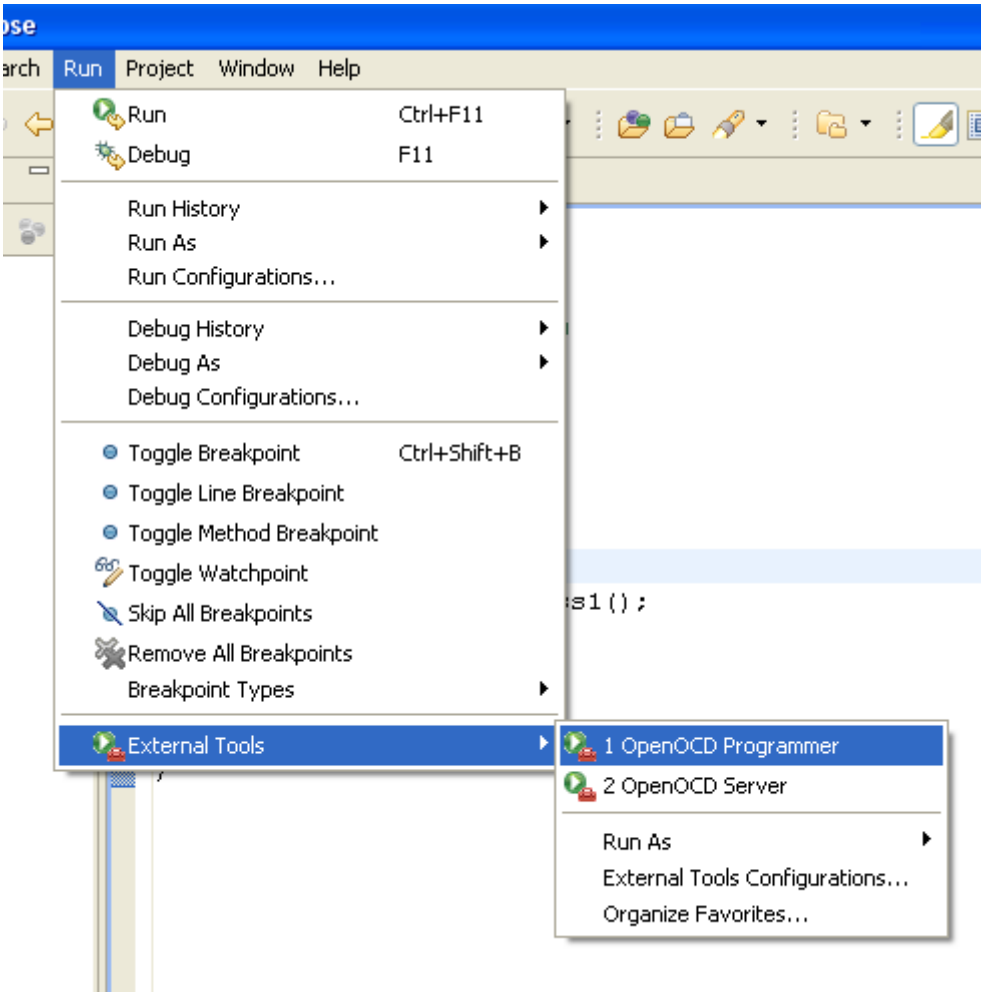
```
target remote localhost:3333
monitor reset
reset halt
monitor soft_reset_halt
```

„Run“ commands:

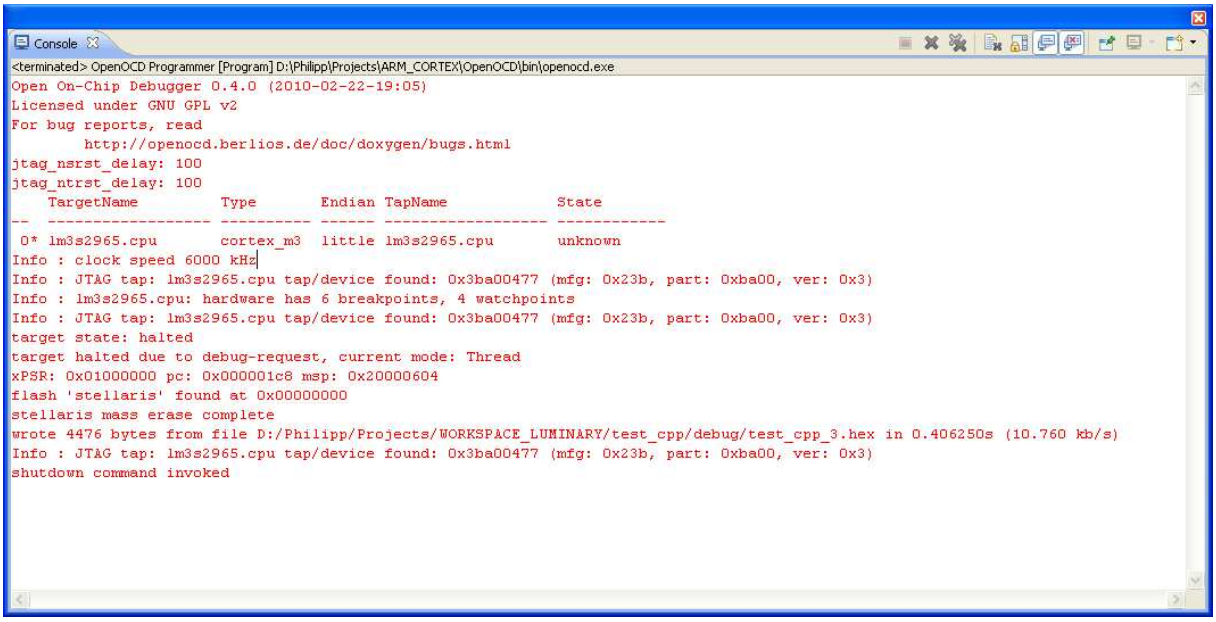
```
next
step
```

5. Arbeitsschritte:

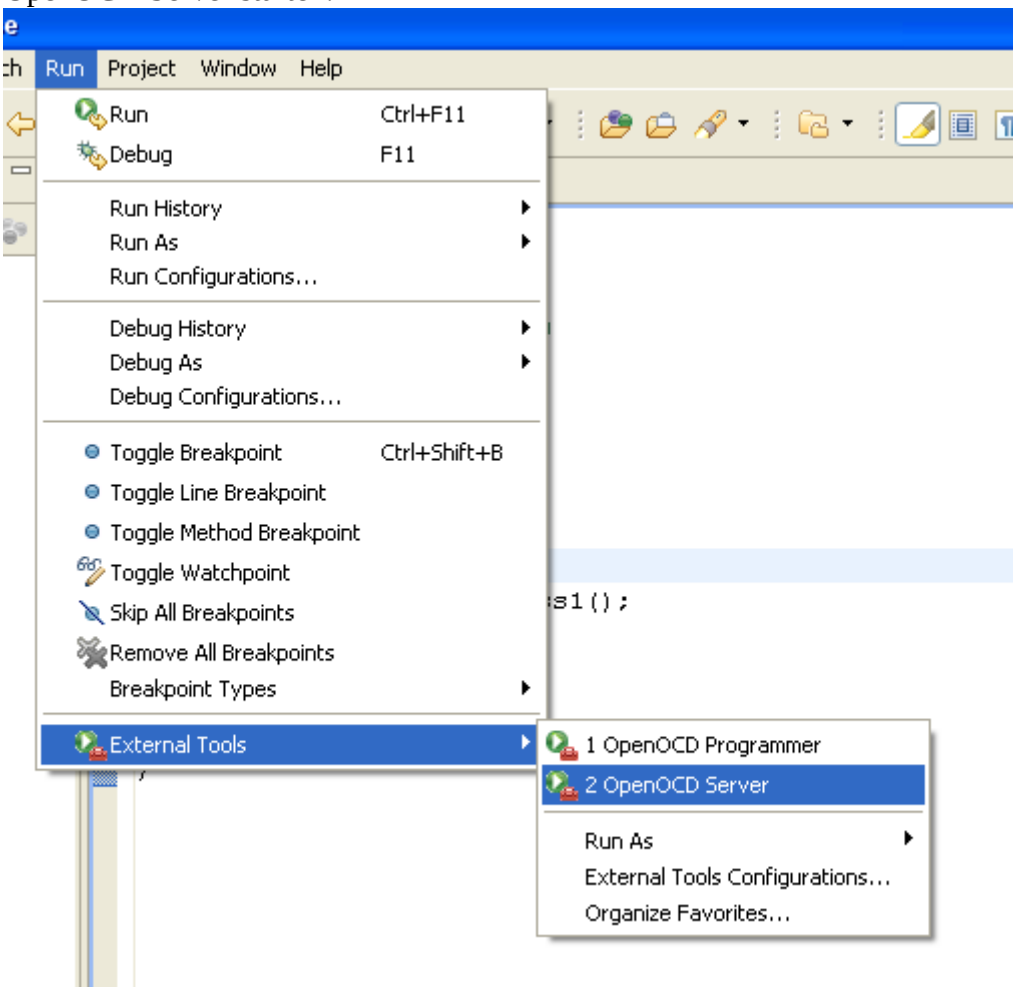
Compilieren.
Flashen:



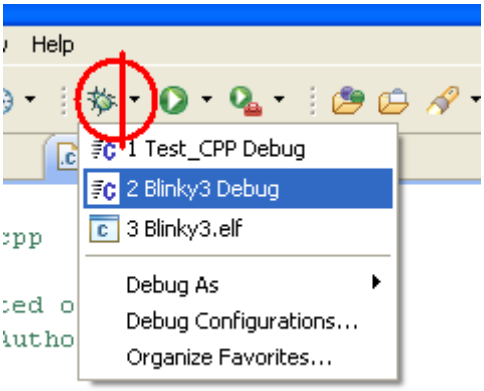
Ergebnis:

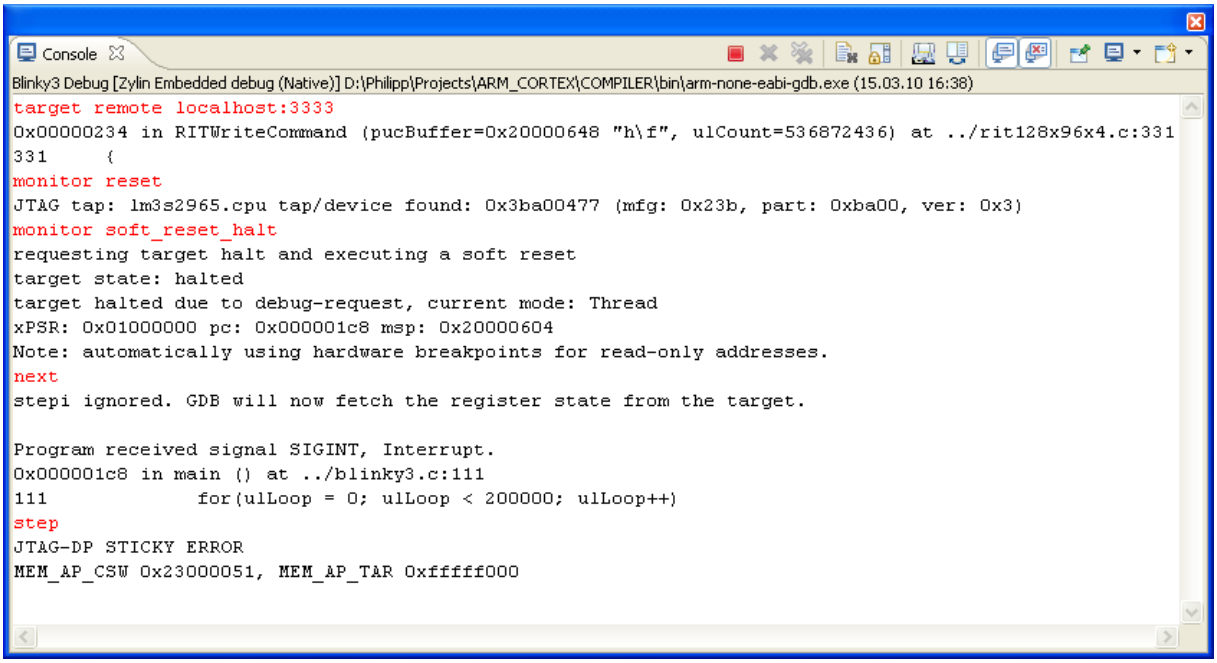


OpenOCD Server starten:



Debug starten:


Help
1 Test_CPP Debug
2 Blinky3 Debug
3 Blinky3.elf
Debug As
Debug Configurations...
Organize Favorites...


Console
Blinky3 Debug [Zylin Embedded debug (Native)] D:\Philipp\Projects\ARM_CORTEX\COMPILER\bin\arm-none-eabi-gdb.exe (15.03.10 16:38)
target remote localhost:3333
0x00000234 in RITWriteCommand (pucBuffer=0x20000648 "h\f", ulCount=536872436) at ../rit128x96x4.c:331
331 {
monitor reset
JTAG tap: lm3s2965.cpu tap/device found: 0x3ba00477 (mfg: 0x23b, part: 0xba00, ver: 0x3)
monitor soft_reset_halt
requesting target halt and executing a soft reset
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x000001c8 msp: 0x20000604
Note: automatically using hardware breakpoints for read-only addresses.
next
stepi ignored. GDB will now fetch the register state from the target.
Program received signal SIGINT, Interrupt.
0x000001c8 in main () at ../blinky3.c:111
111 for(ulLoop = 0; ulLoop < 200000; ulLoop++)
step
JTAG-DP STICKY ERROR
MEM_AP_CSW 0x23000051, MEM_AP_TAR 0xfffff000

That’s all!