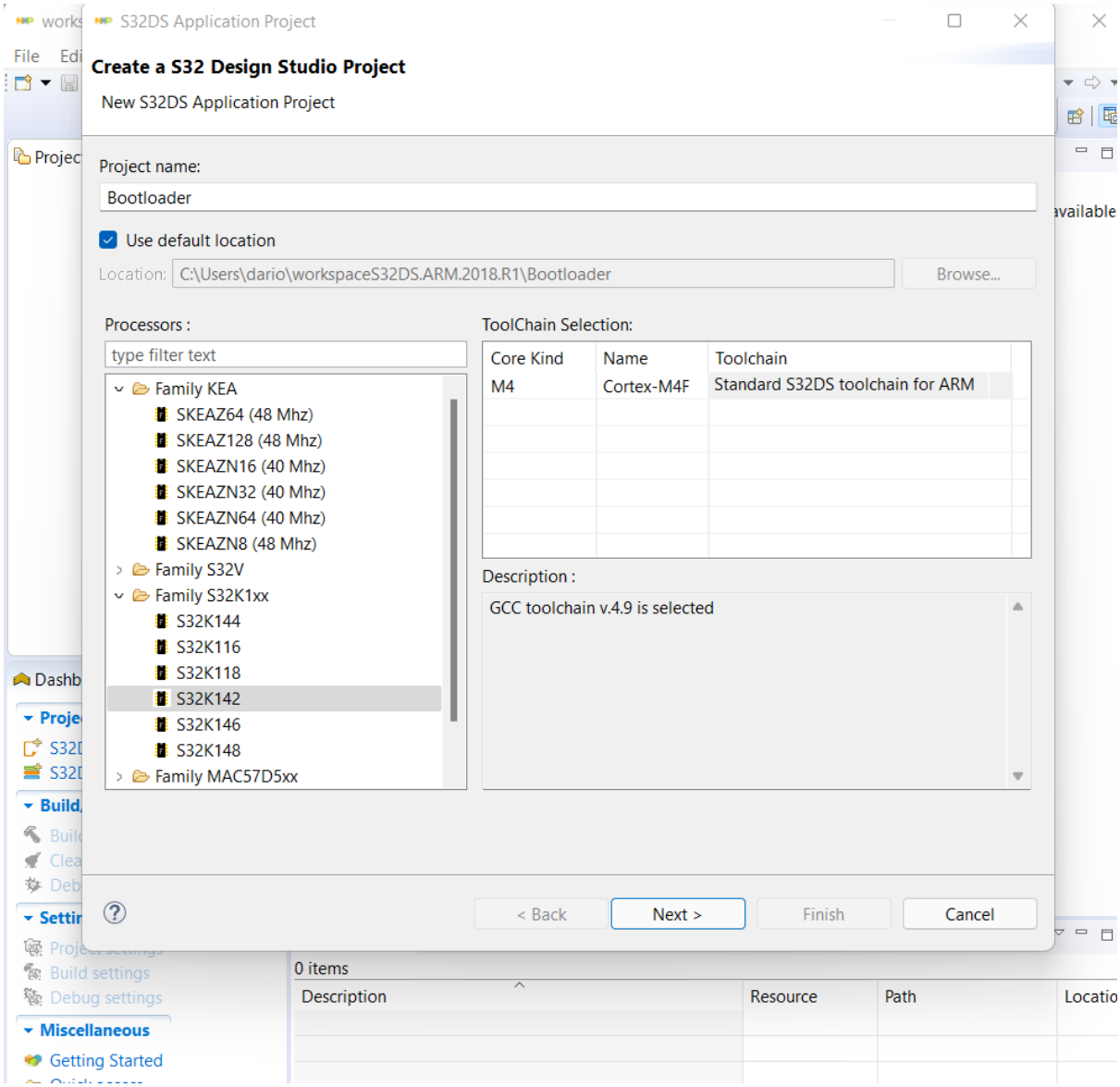
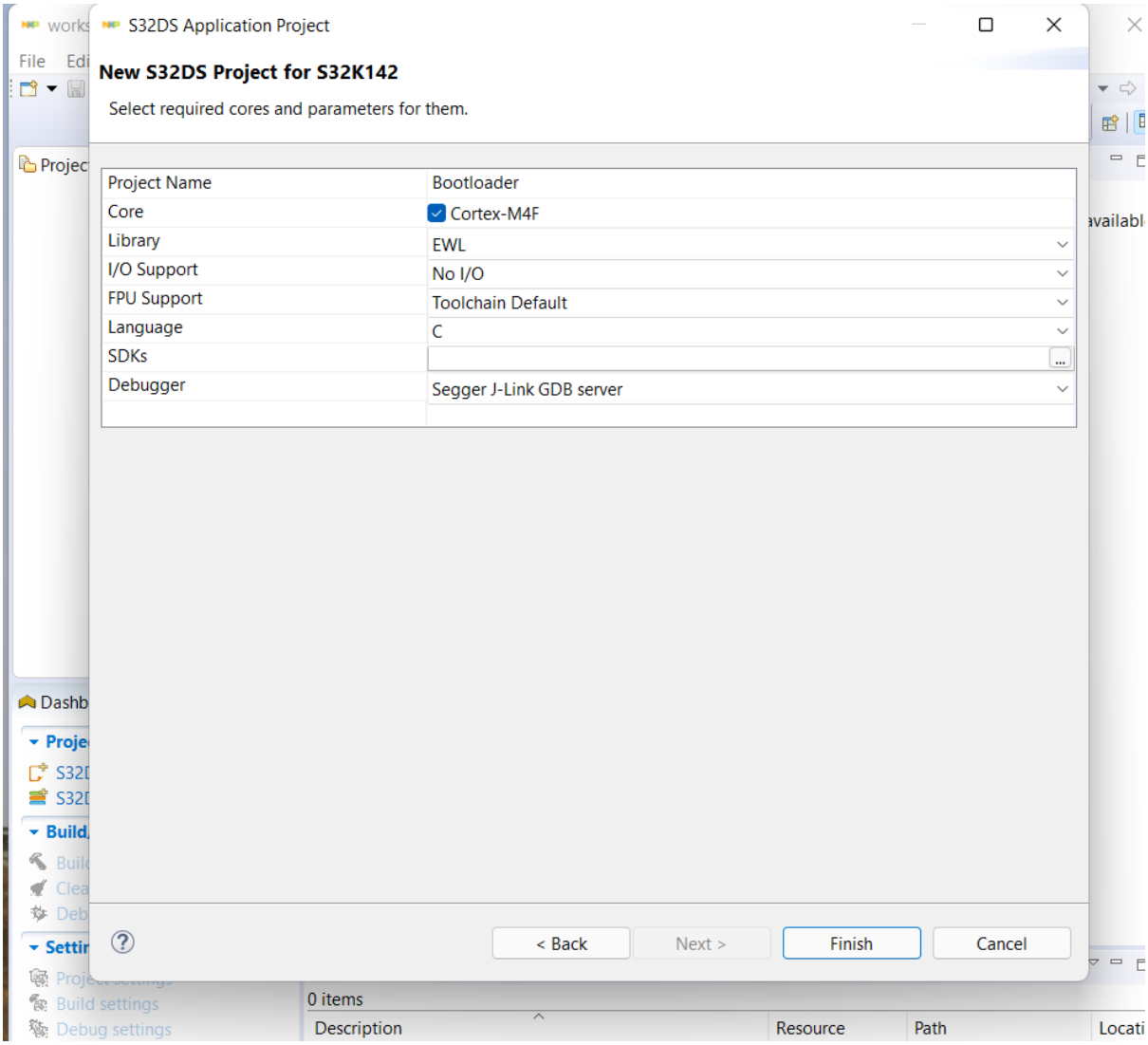
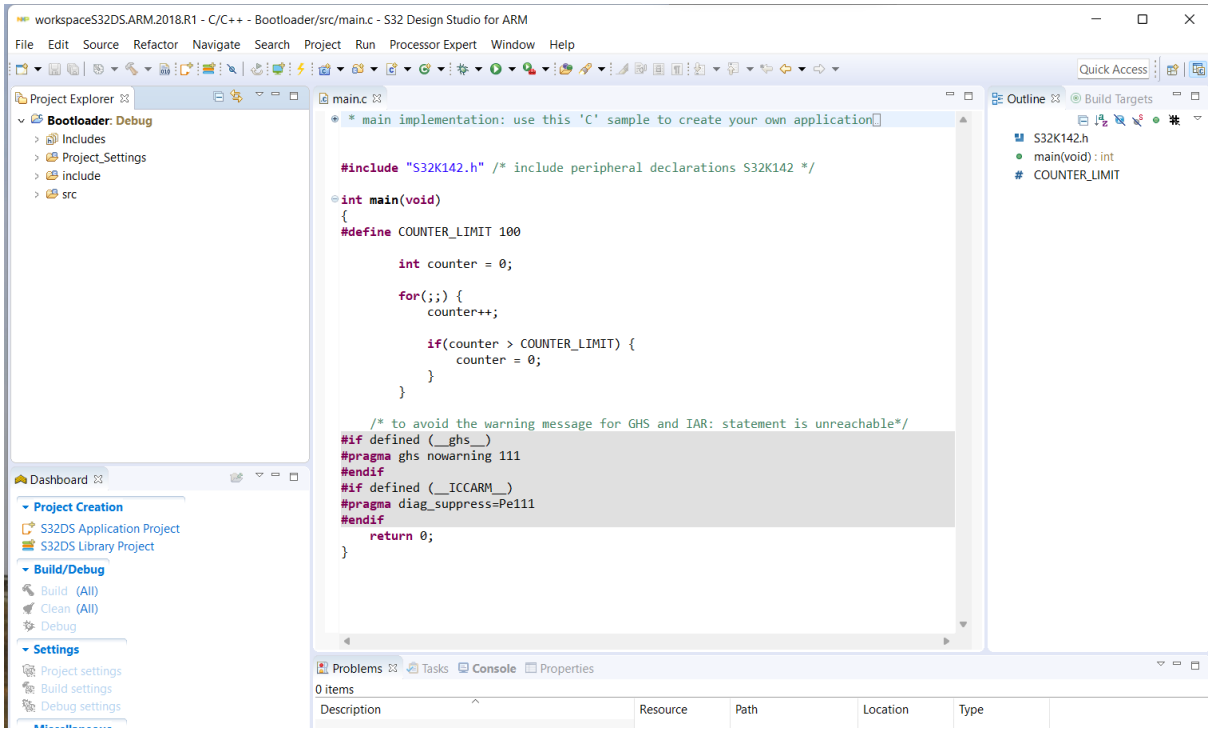
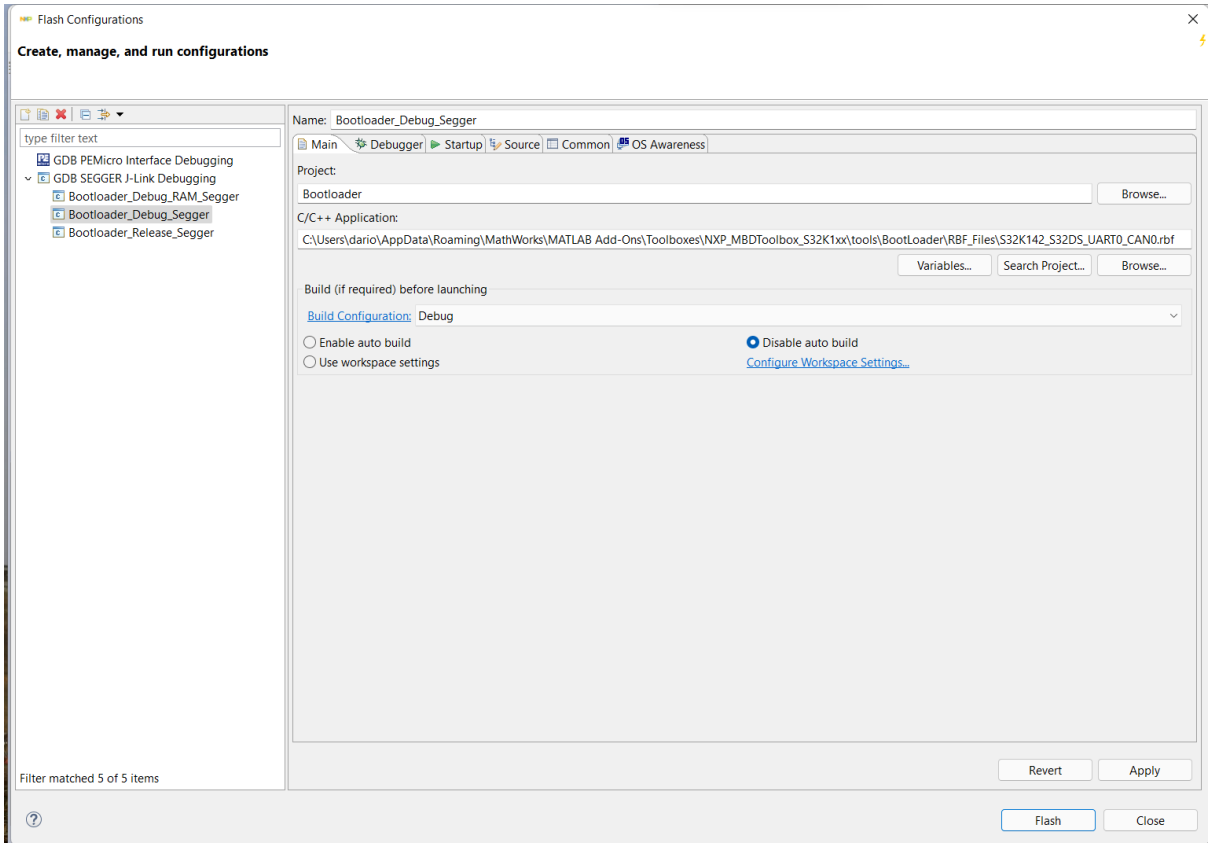
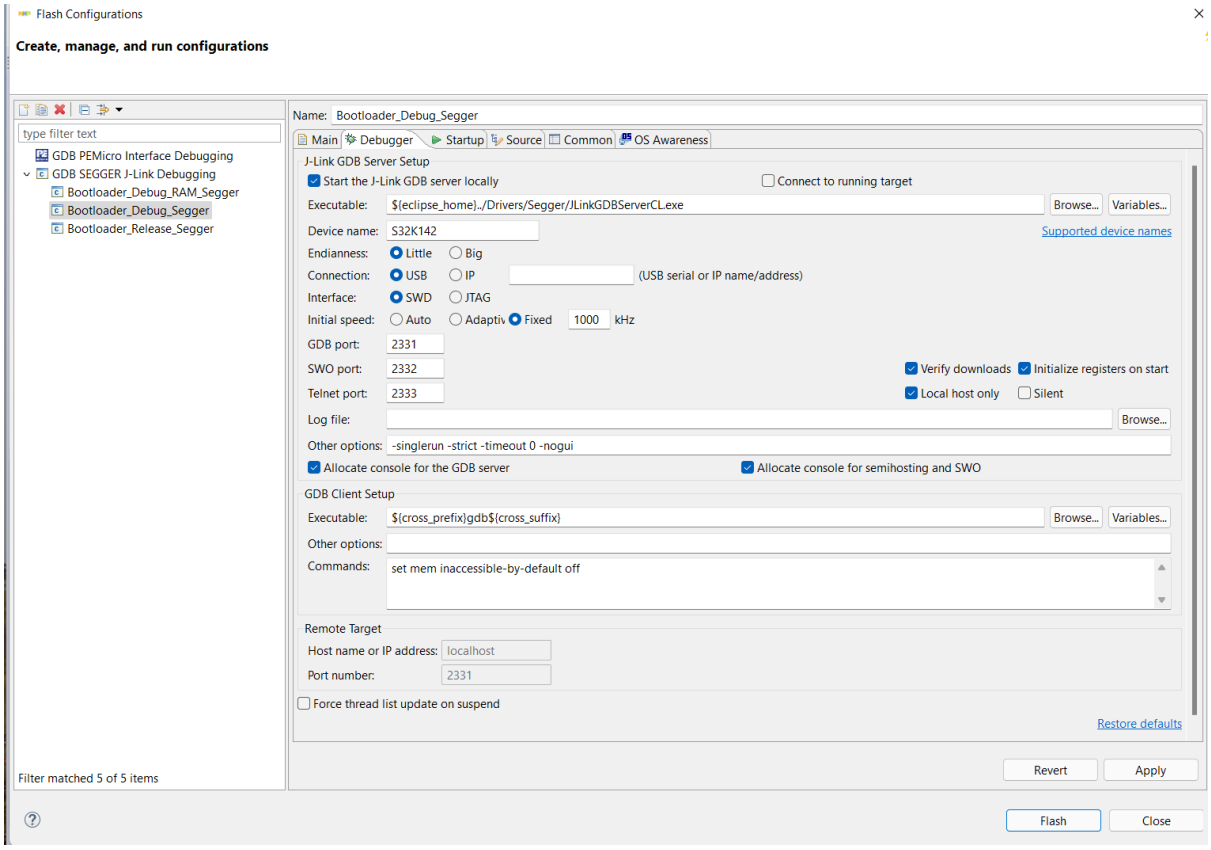
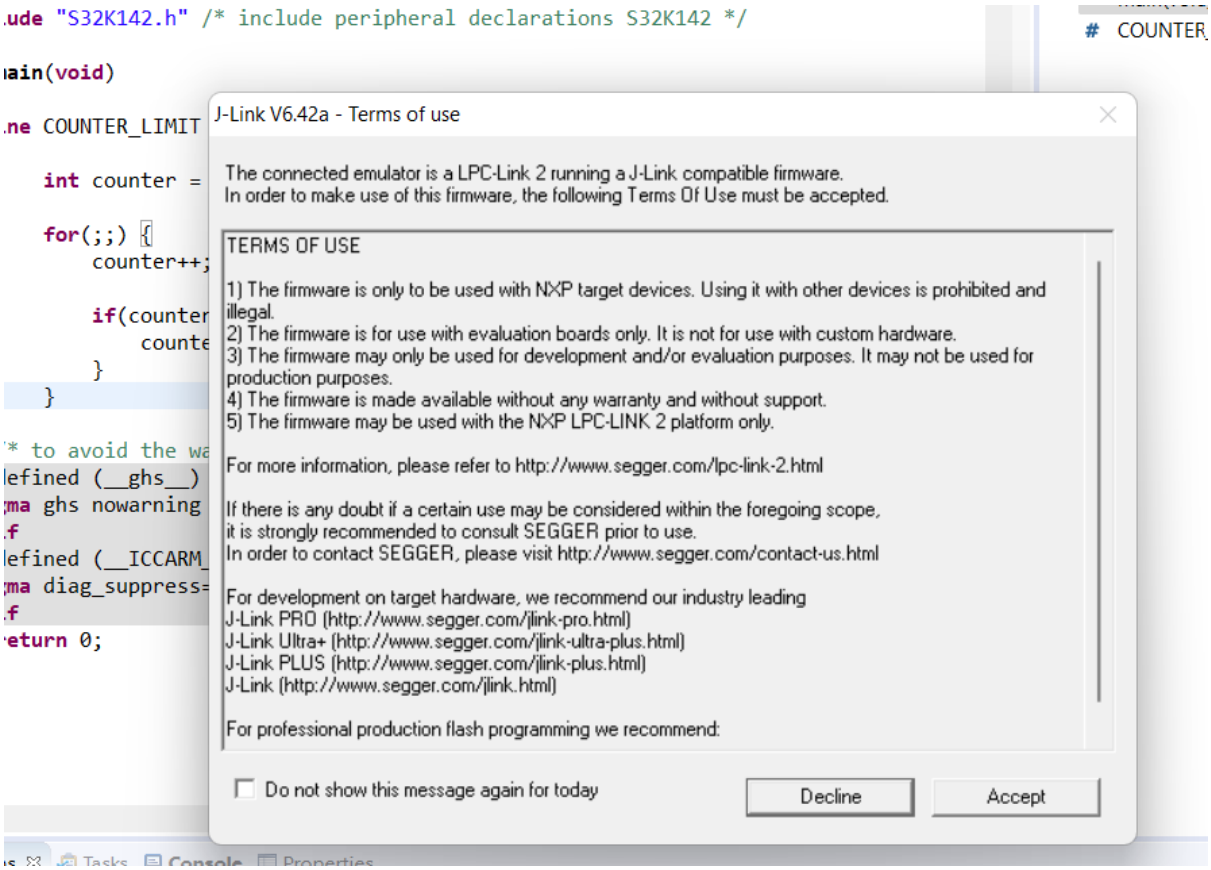

S32DS Application Project
Create a S32 Design Studio Project
New S32DS Application Project
Project name:
Bootloader
Use default location
Location: C:\Users\dario\workspaceS32DS.ARM.2018.R1\Bootloader
Browse...
Processors :
type filter text
Family KEA
SKEAZ64 (48 Mhz)
SKEAZ128 (48 Mhz)
SKEAZN16 (40 Mhz)
SKEAZN32 (40 Mhz)
SKEAZN64 (40 Mhz)
SKEAZN8 (48 Mhz)
Family S32V
Family S32K1xx
S32K144
S32K116
S32K118
S32K142
S32K146
S32K148
Family MAC57D5xx
ToolChain Selection:
Core Kind
Name
Toolchain
M4
Cortex-M4F
Standard S32DS toolchain for ARM
Description :
GCC toolchain v.4.9 is selected
< Back
Next >
Finish
Cancel
Build settings
Debug settings
Miscellaneous
Getting Started
0 items
Description
Resource
Path


S32DS Application Project
New S32DS Project for S32K142
Select required cores and parameters for them.
Project Name
Bootloader
Core
Cortex-M4F
Library
EWL
I/O Support
No I/O
FPU Support
Toolchain Default
Language
C
SDKs
Debugger
Segger J-Link GDB server
< Back
Next >
Finish
Cancel
0 items
Description
Resource
Path


workspaceS32DS.ARM.2018.R1 - C/C++ - Bootloader/src/main.c - S32 Design Studio for ARM
File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help
Quick Access
Project Explorer
Bootloader: Debug
Includes
Project_Settings
include
src
main.c
* main implementation: use this 'C' sample to create your own application
#include "S32K142.h" /* include peripheral declarations S32K142 */
int main(void)
{
#define COUNTER_LIMIT 100
int counter = 0;
for(;;) {
counter++;
if(counter > COUNTER_LIMIT) {
counter = 0;
}
}
/* to avoid the warning message for GHS and IAR: statement is unreachable*/
#if defined (__ghs__)
#pragma ghs nowarning 111
#endif
#if defined (__ICCARM__)
#pragma diag_suppress=Pe111
#endif
return 0;
}
Outline
Build Targets
S32K142.h
main(void) : int
COUNTER_LIMIT
Dashboard
Project Creation
S32DS Application Project
S32DS Library Project
Build/Debug
Build (All)
Clean (All)
Debug
Settings
Project settings
Build settings
Debug settings
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type


Flash Configurations
Create, manage, and run configurations
type filter text
GDB PEMicro Interface Debugging
GDB SEGGER J-Link Debugging
Bootloader_Debug_RAM_Segger
Bootloader_Debug_Segger
Bootloader_Release_Segger
Filter matched 5 of 5 items
Name: Bootloader_Debug_Segger
Main
Debugger
Startup
Source
Common
OS Awareness
Project:
Bootloader
Browse...
C/C++ Application:
C:\Users\dario\AppData\Roaming\MathWorks\MATLAB Add-Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\tools\BootLoader\RBF_Files\S32K142_S32DS_UART0_CAN0.rbf
Variables...
Search Project...
Browse...
Build (if required) before launching
Build Configuration:
Debug
Enable auto build
Disable auto build
Use workspace settings
Configure Workspace Settings...
Revert
Apply
Flash
Close


Flash Configurations
Create, manage, and run configurations
type filter text
GDB PEMicro Interface Debugging
GDB SEGGER J-Link Debugging
Bootloader_Debug_RAM_Segger
Bootloader_Debug_Segger
Bootloader_Release_Segger
Filter matched 5 of 5 items
Name: Bootloader_Debug_Segger
Main
Debugger
Startup
Source
Common
OS Awareness
J-Link GDB Server Setup
Start the J-Link GDB server locally
Connect to running target
Executable: ${eclipse_home}../Drivers/Segger/JLinkGDBServerCL.exe
Browse...
Variables...
Device name: S32K142
Supported device names
Endianness: Little Big
Connection: USB IP
(USB serial or IP name/address)
Interface: SWD JTAG
Initial speed: Auto Adaptiv Fixed 1000 kHz
GDB port: 2331
SWO port: 2332
Telnet port: 2333
Verify downloads
Initialize registers on start
Local host only
Silent
Log file:
Other options: -singlerun -strict -timeout 0 -nogui
Allocate console for the GDB server
Allocate console for semihosting and SWO
GDB Client Setup
Executable: ${cross_prefix}gdb${cross_suffix}
Other options:
Commands: set mem inaccessible-by-default off
Remote Target
Host name or IP address: localhost
Port number: 2331
Force thread list update on suspend
Restore defaults
Revert
Apply
Flash
Close


ude "S32K142.h" /* include peripheral declarations S32K142 */
ain(void)
ne COUNTER_LIMIT
int counter =
for(;;) {
counter++;
if(counter
counte
}
}
* to avoid the wa
efined (__ghs__)
ma ghs nowarning
f
efined (__ICCARM_
ma diag_suppress=
f
eturn 0;
COUNTER
J-Link V6.42a - Terms of use
The connected emulator is a LPC-Link 2 running a J-Link compatible firmware.
In order to make use of this firmware, the following Terms Of Use must be accepted.
TERMS OF USE
1) The firmware is only to be used with NXP target devices. Using it with other devices is prohibited and illegal.
2) The firmware is for use with evaluation boards only. It is not for use with custom hardware.
3) The firmware may only be used for development and/or evaluation purposes. It may not be used for production purposes.
4) The firmware is made available without any warranty and without support.
5) The firmware may be used with the NXP LPC-LINK 2 platform only.
For more information, please refer to http://www.segger.com/lpc-link-2.html
If there is any doubt if a certain use may be considered within the foregoing scope,
it is strongly recommended to consult SEGGER prior to use.
In order to contact SEGGER, please visit http://www.segger.com/contact-us.html
For development on target hardware, we recommend our industry leading
J-Link PRO (http://www.segger.com/jlink-pro.html)
J-Link Ultra+ (http://www.segger.com/jlink-ultra-plus.html)
J-Link PLUS (http://www.segger.com/jlink-plus.html)
J-Link (http://www.segger.com/jlink.html)
For professional production flash programming we recommend:
Do not show this message again for today
Decline
Accept