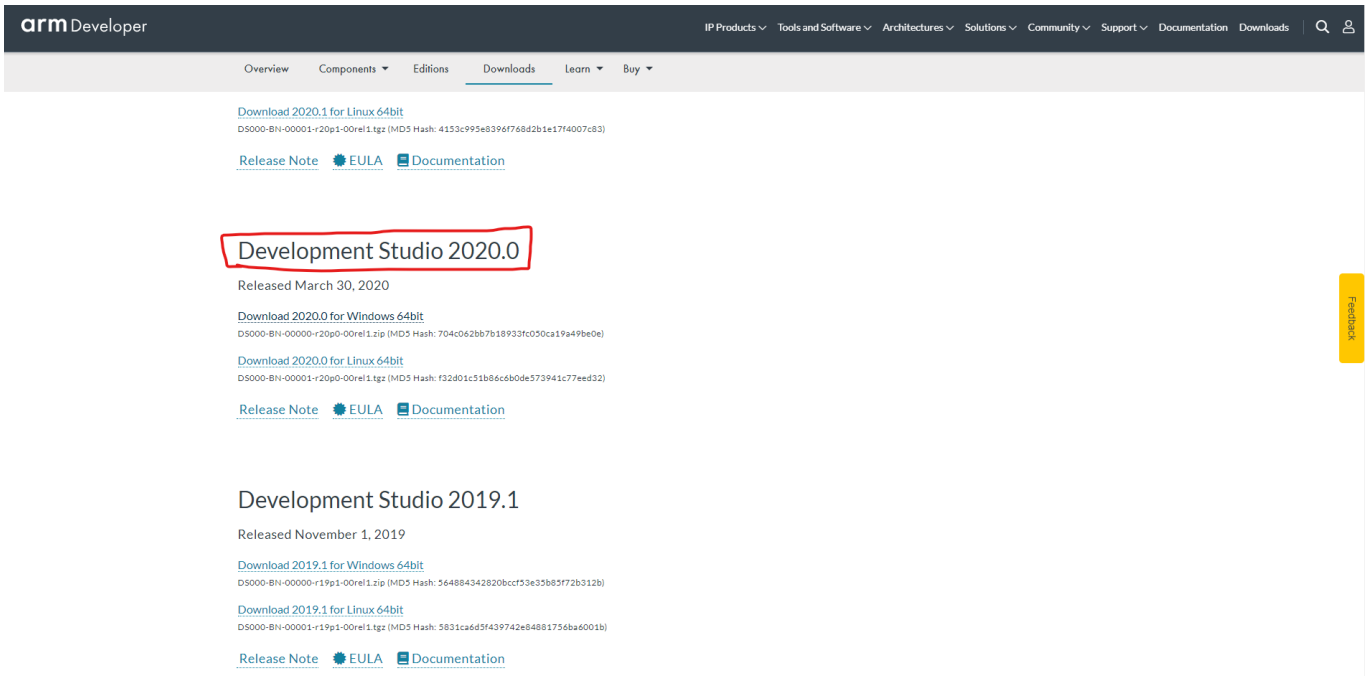
Arm Development Studio Tutorial

Download Arm DS

Download the software (Arm Development Studio).
→ Choose the version 2020.0 (highlighted on the following image) with the build: 202000915

https://developer.arm.com/tools-and-software/embedded/arm-development-studio/downloads



Install the software.

Server Configuration

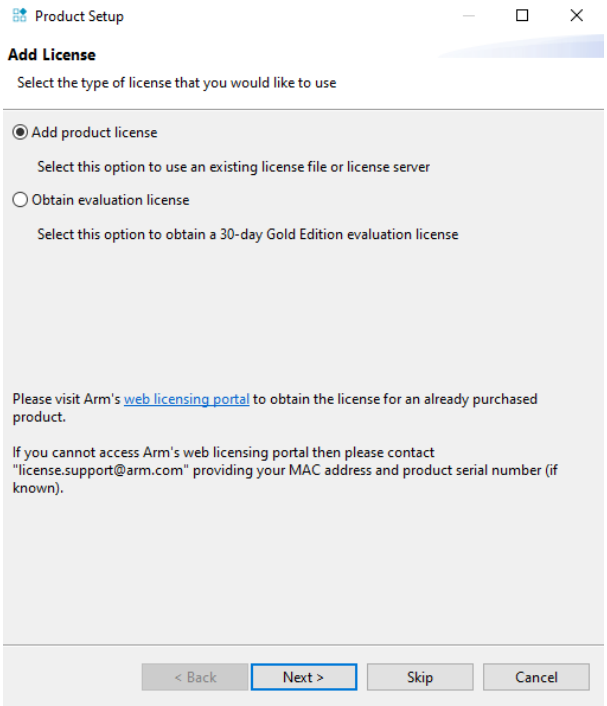
Password of the machine: **ninjaCERTIFICADE**

1. Open lmtools.exe ("search" or in C:/FlexNET/lmtools.exe).
2. Open the tab "*Start/Stop/Reread*" and click in "*Start Server*".
3. After starting it, on the tab "*Server Status*" click in "*Perform Status Enquiry*" to assure that the server is active.

When it's active the console has the following output:
"*License server status: 8224@ESRGV3-SERVER*"

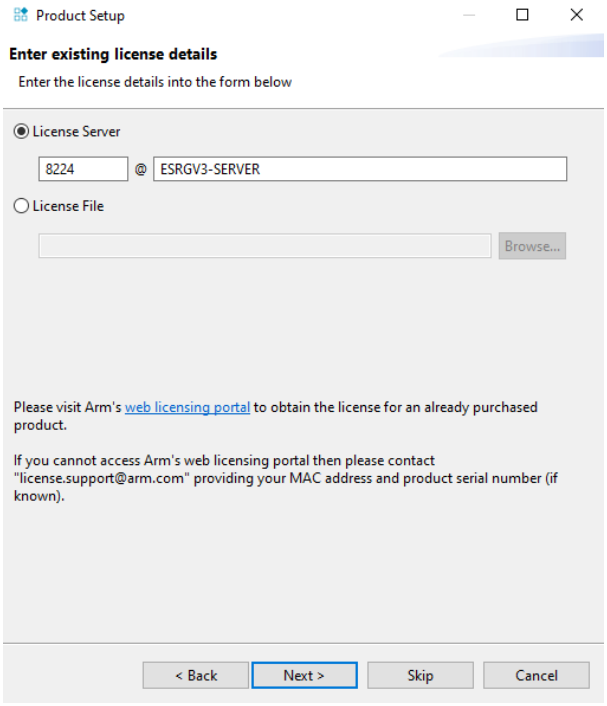
License

After installing the sorftware, open it. The *Product Setup* window should open. Select the *Add product license* option.

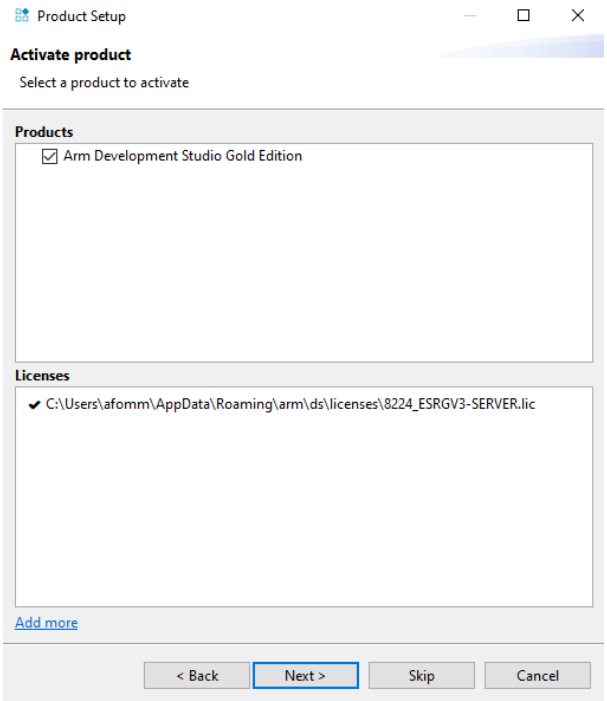


Select the *License Server* options and introduce the following server port and hostname:

8224 @ ESRGV3-SERVER

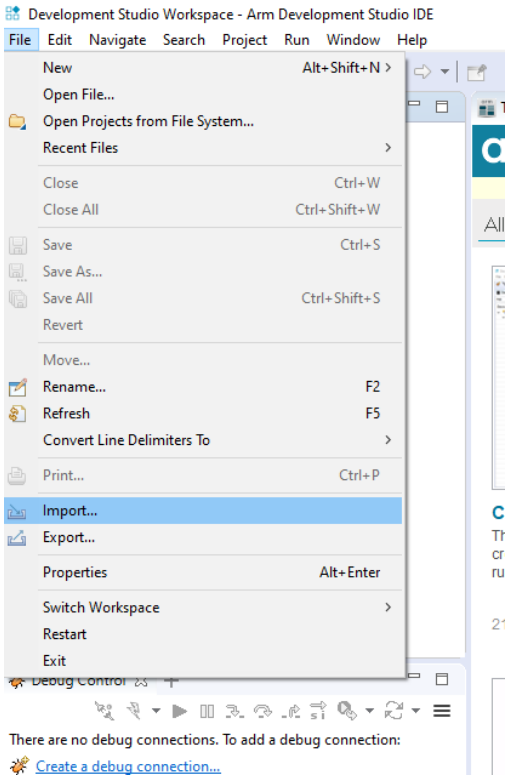


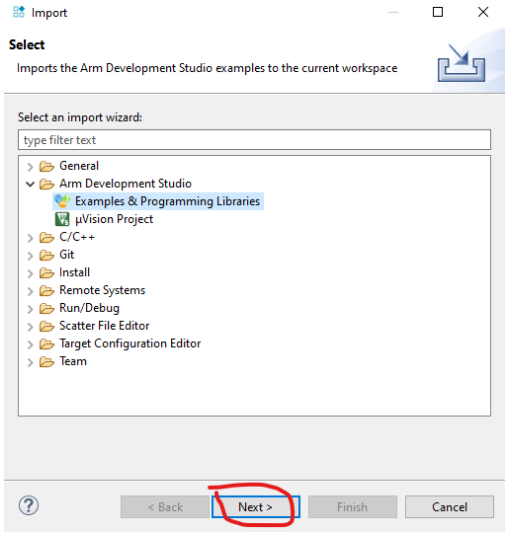
Click *Next* to activate the product



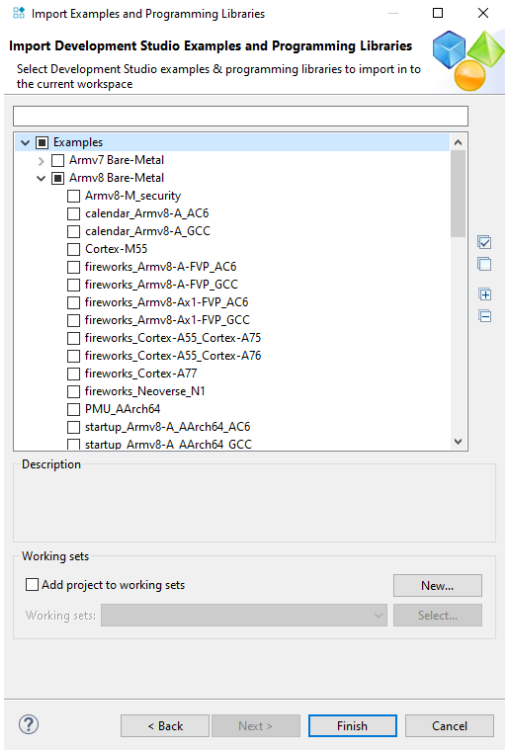
Setup Example Project

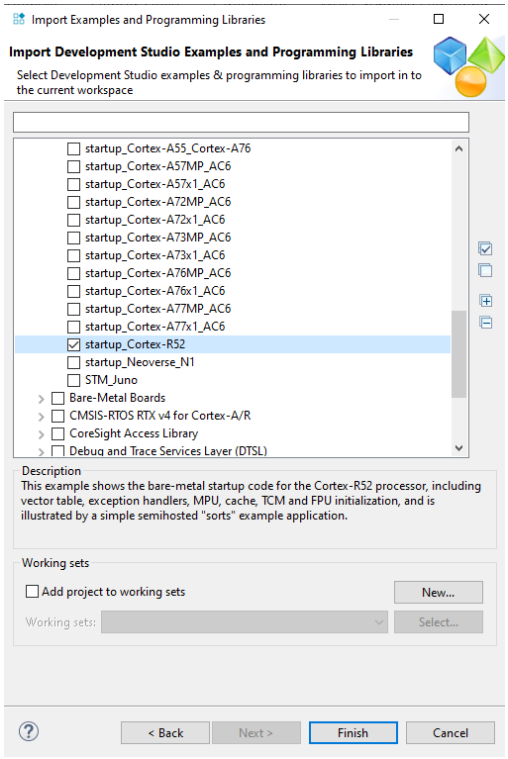
After activating the license select *File* → *Import* and on the folder *Arm Development Studio* choose the *Examples & Programming Libraries*



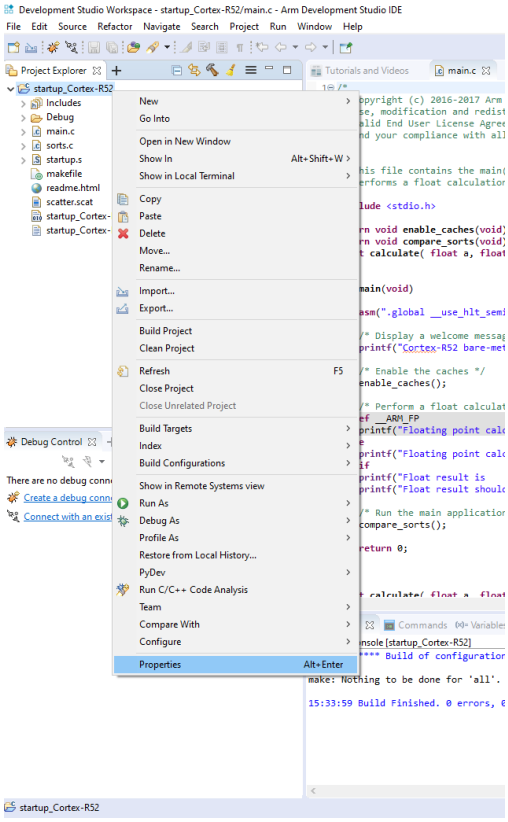


Then, to select the Cortex-R52 example, select *Examples* → *Armv8 Bare-Metal*.
Scroll-down to find the *startup_Cortex-R52* project.
Click *Finish* to select the project.





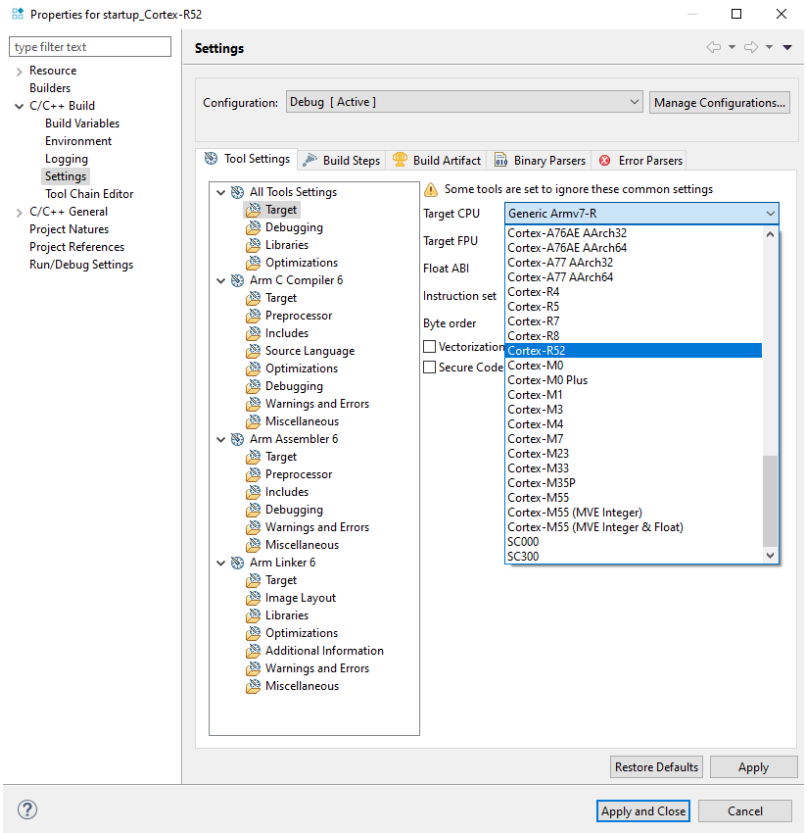
Right-click on the project folder and open its priorities.



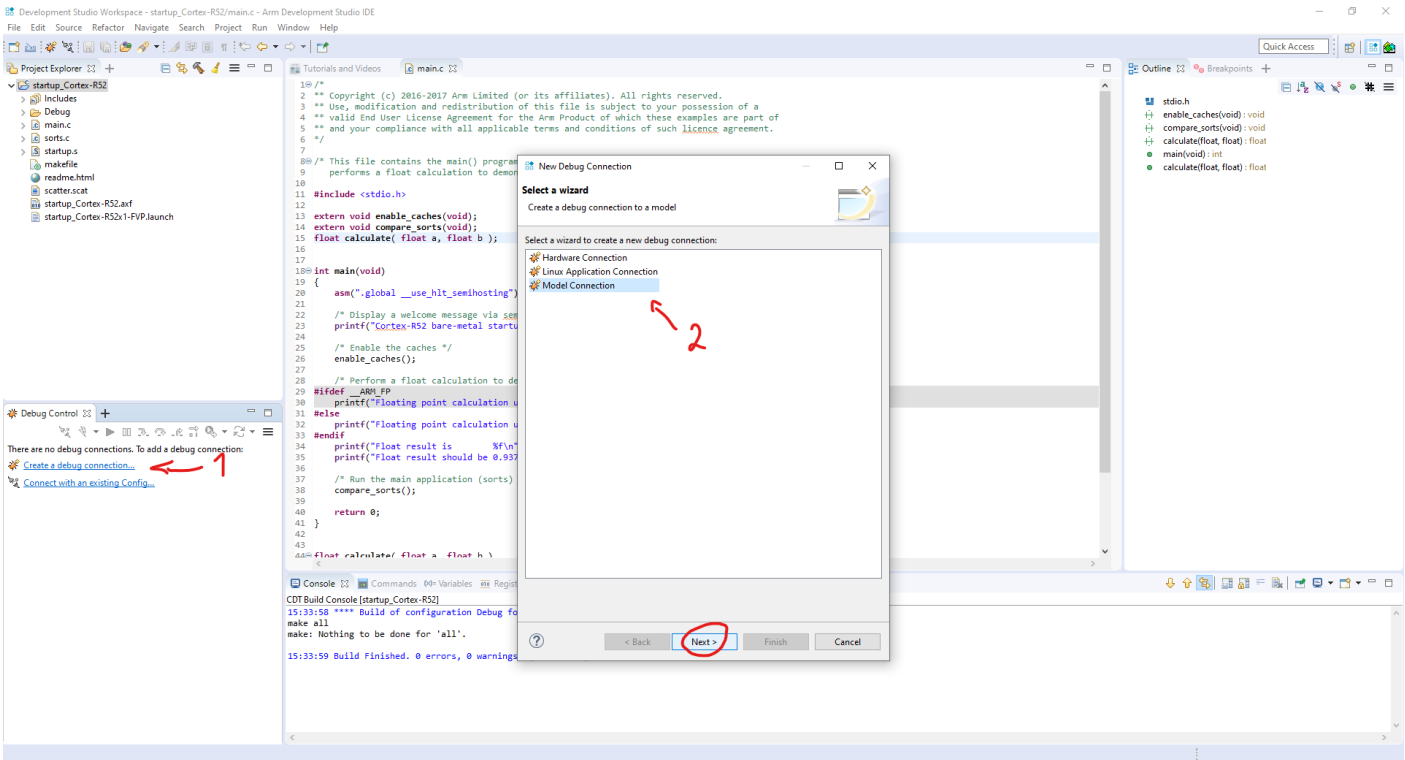
On the left settings select *C/C++ Build → Settings* to change the target CPU to the one used on the project.

To do so, click on "All Tools Settings → Target *and change the* Target CPU* to the

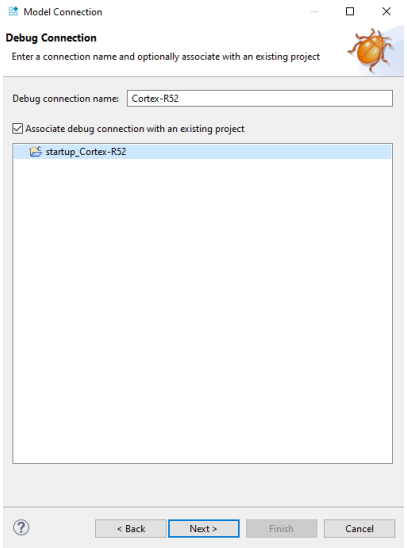
Cortex-R52.
Then click Apply and Close.



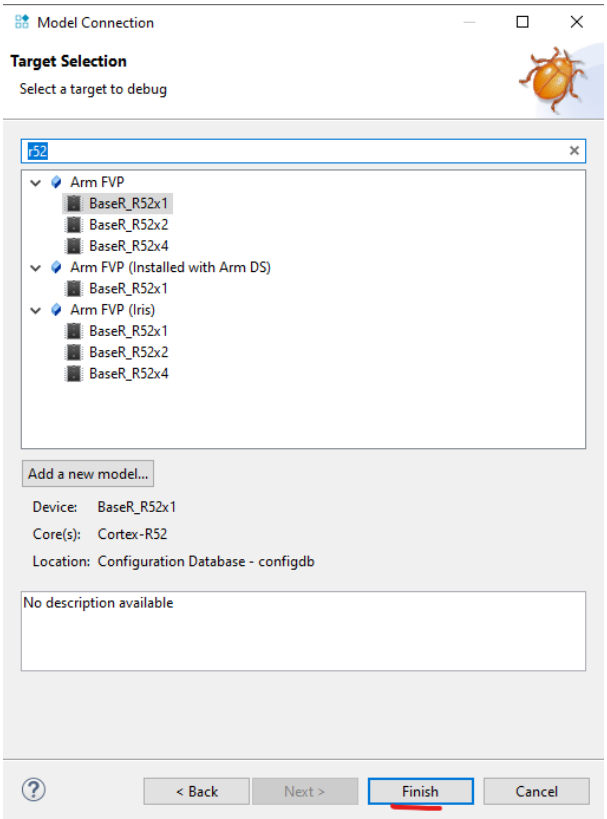
Now, to create the connection to the CPU, select the *Create a debug connection...* under the Debug control (1).
Then select the *Model Connection* (2) and click *Next*



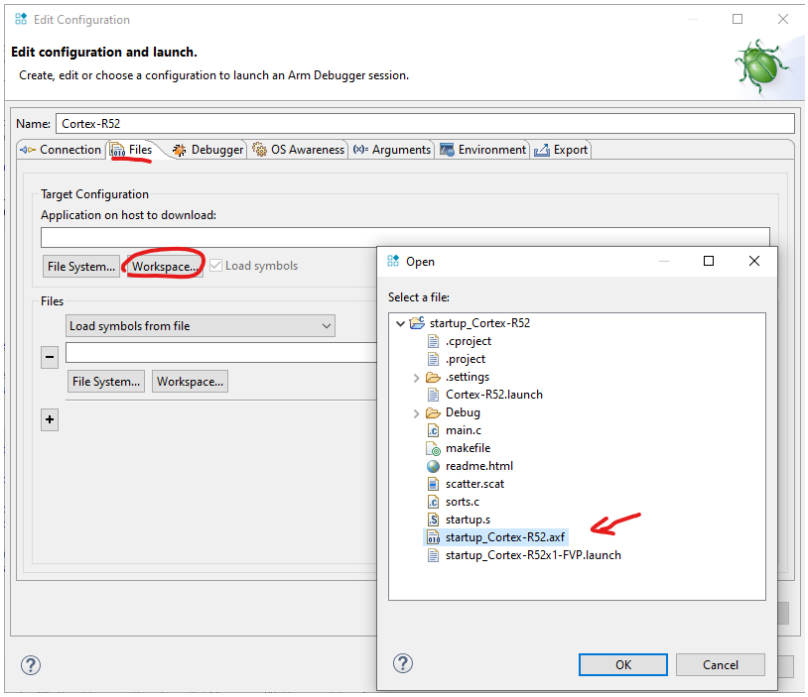
Associate the debug connection with the existing project and choose a name for this connection



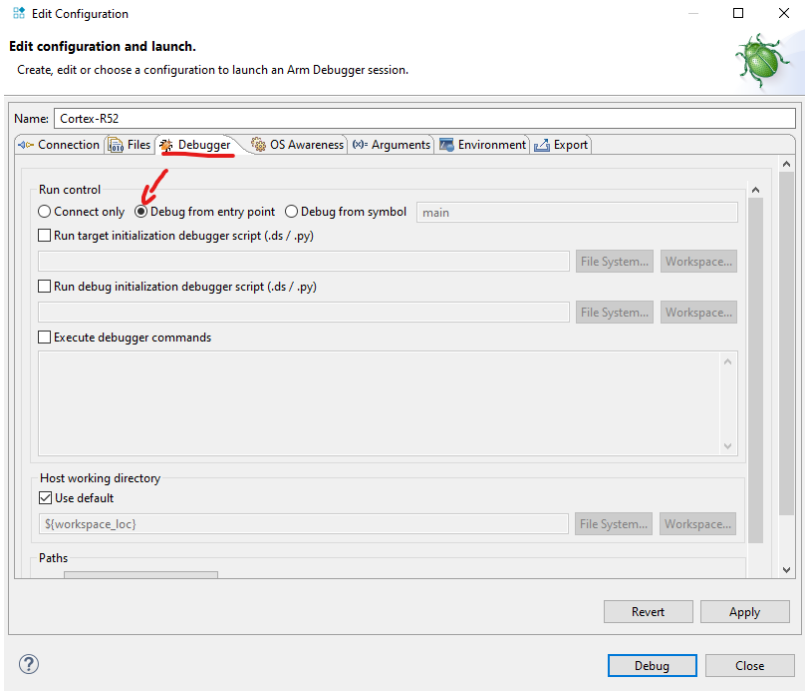
Search for the Cortex-R52 cpu and select it



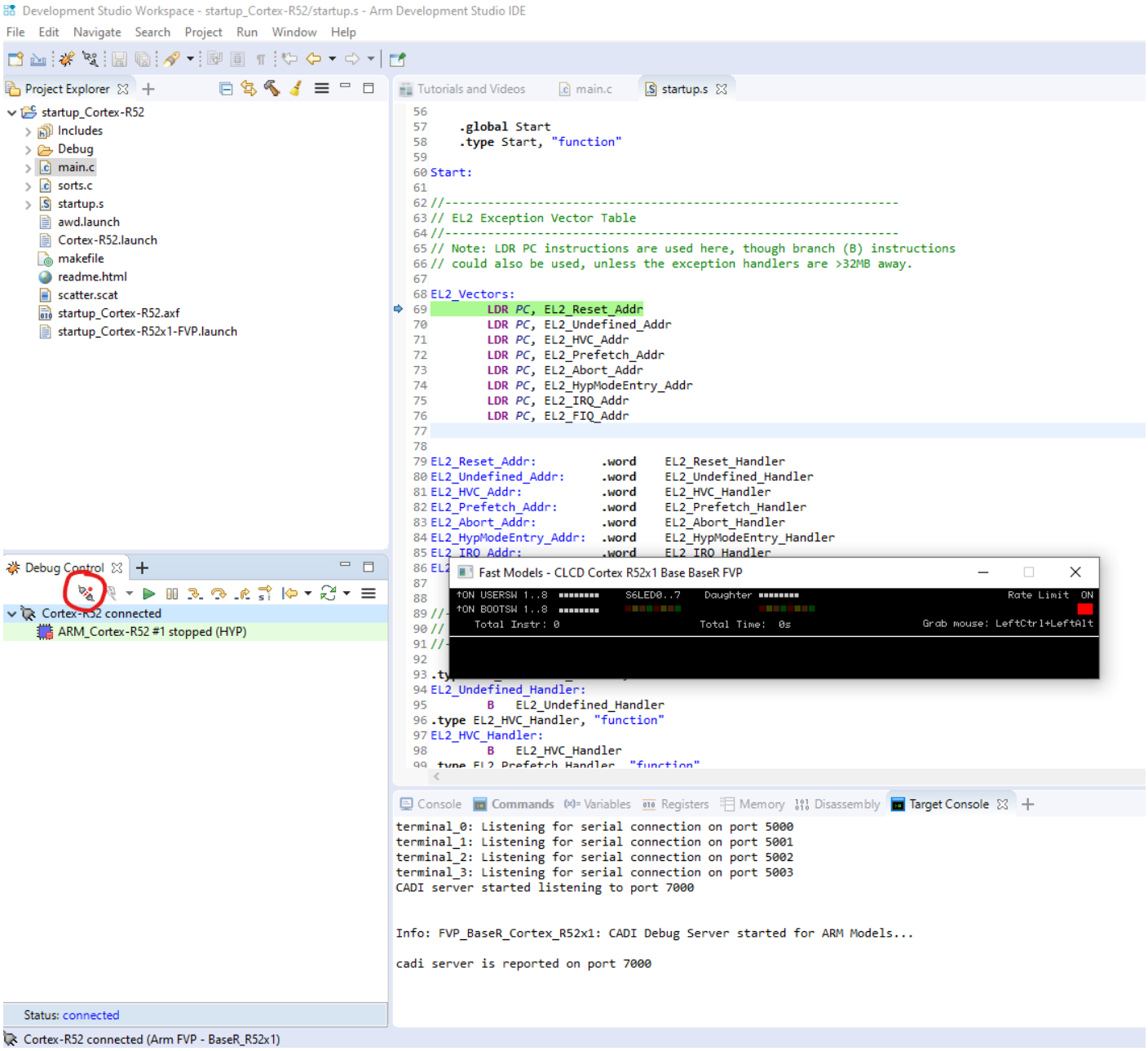
The window to configure the connection should appear.
On the *Files* tab select the *Workspace* button under the *Target Configuration*.
Select the *[...].axf* file from the Debug folder of the project



On the *Debugger* tab select the *Debug from entry point* option. Apply the changes and click *Debug*.



With the connection configured, click the plug icon to start a debug session.


Development Studio Workspace - startup_Cortex-R52/startup.s - Arm Development Studio IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
startup_Cortex-R52
Includes
Debug
main.c
sorts.c
startup.s
awd.launch
Cortex-R52.launch
makefile
readme.html
scatter.scat
startup_Cortex-R52.axf
startup_Cortex-R52x1-FVP.launch
Tutorials and Videos
main.c
startup.s
.global Start
.type Start, "function"
Start:
// EL2 Exception Vector Table
// Note: LDR PC instructions are used here, though branch (B) instructions
// could also be used, unless the exception handlers are >32MB away.
EL2_Vectors:
LDR PC, EL2_Reset_Addr
LDR PC, EL2_Undefined_Addr
LDR PC, EL2_HVC_Addr
LDR PC, EL2_Prefetch_Addr
LDR PC, EL2_Abort_Addr
LDR PC, EL2_HypModeEntry_Addr
LDR PC, EL2_IRQ_Addr
LDR PC, EL2_FIQ_Addr
EL2_Reset_Addr: .word EL2_Reset_Handler
EL2_Undefined_Addr: .word EL2_Undefined_Handler
EL2_HVC_Addr: .word EL2_HVC_Handler
EL2_Prefetch_Addr: .word EL2_Prefetch_Handler
EL2_Abort_Addr: .word EL2_Abort_Handler
EL2_HypModeEntry_Addr: .word EL2_HypModeEntry_Handler
EL2_IRQ_Addr: .word EL2_IRQ_Handler
EL2_Undefined_Handler:
B EL2_Undefined_Handler
.type EL2_HVC_Handler, "function"
EL2_HVC_Handler:
B EL2_HVC_Handler
Debug Control
Cortex-R52 connected
ARM_Cortex-R52 #1 stopped (HYP)
Fast Models - CLCD Cortex R52x1 Base BaseR FVP
Total Instr: 0
Total Time: 0s
Grab mouse: LeftCtrl+LeftAlt
Console
Commands
Variables
Registers
Memory
Disassembly
Target Console
terminal_0: Listening for serial connection on port 5000
terminal_1: Listening for serial connection on port 5001
terminal_2: Listening for serial connection on port 5002
terminal_3: Listening for serial connection on port 5003
CADI server started listening to port 7000
Info: FVP_BaseR_Cortex_R52x1: CADI Debug Server started for ARM Models...
cadi server is reported on port 7000
Status: connected
Cortex-R52 connected (Arm FVP - BaseR_R52x1)