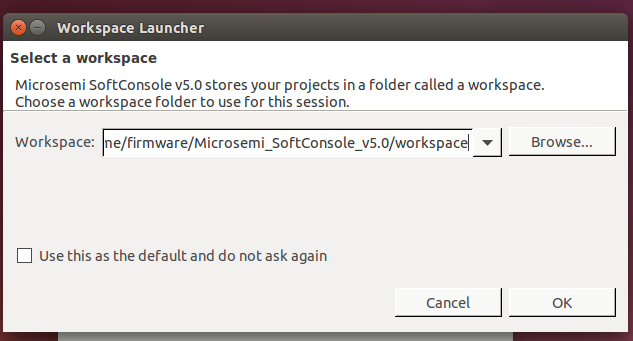
**U-Boot – Build and Debugging Setup on Softconsole**

1. Get the RISC-V U-Boot source from the githup.

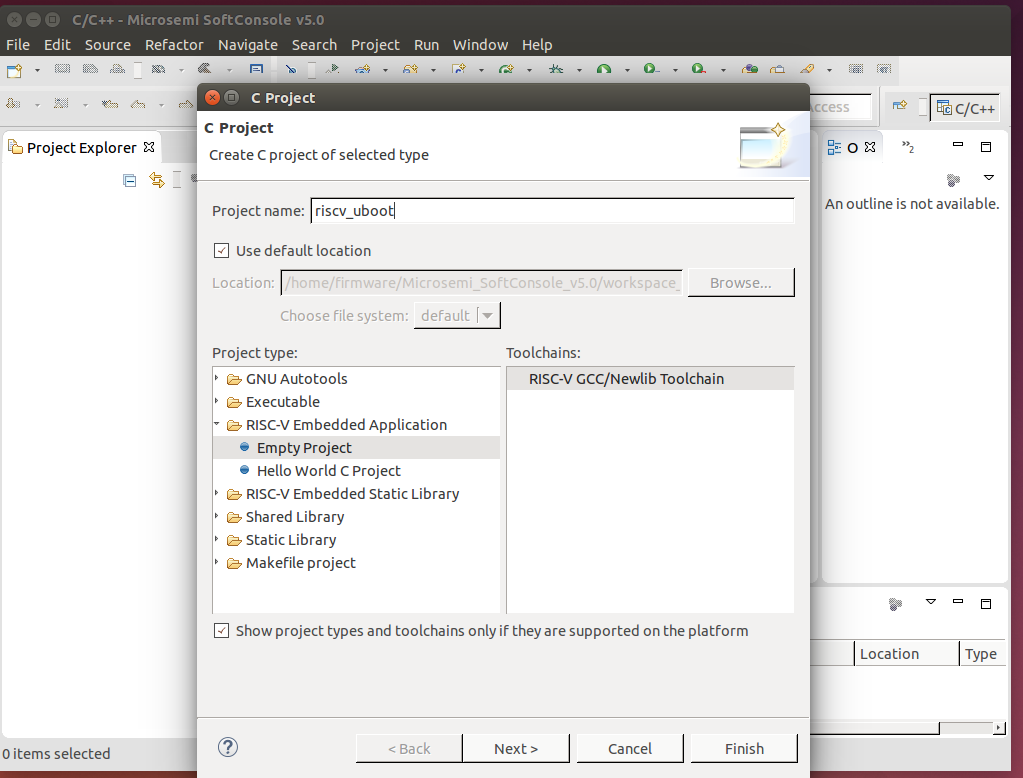
$ git clone https://github.com/RISCV-on-Microsemi-FPGA/RISC-V\_U-Boot

$ cd RISC-V\_U-Boot

2. Open Softconsole and create a workspace for building and debugging U-Boot.

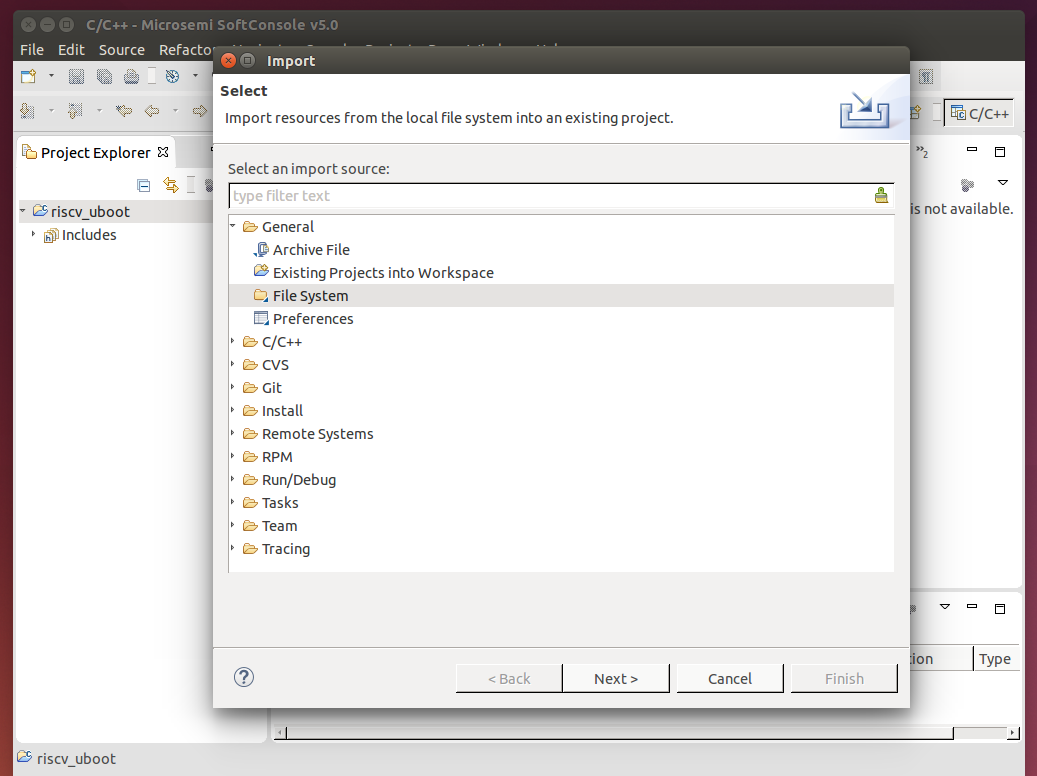


3. Create a project – Select **File -> New -> C Project**

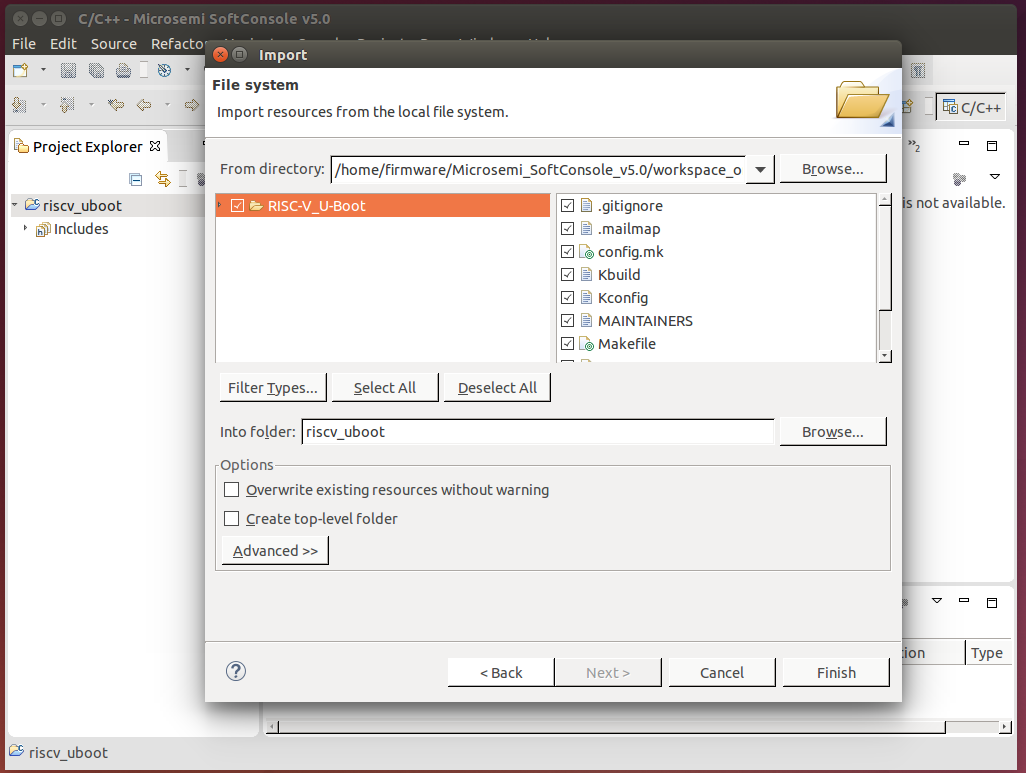


Write the project name, select RISC-V Embedded Application Empty Project and click on Finish.

4. Import U-Boot sources into the project – Select **File -> Import -> General -> File system**

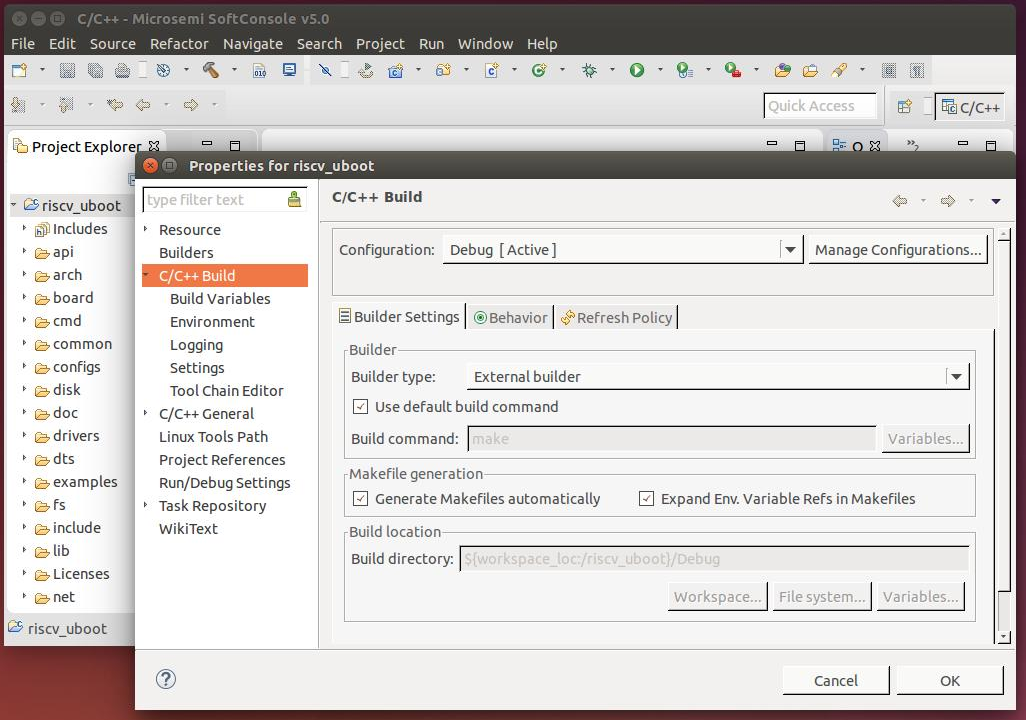


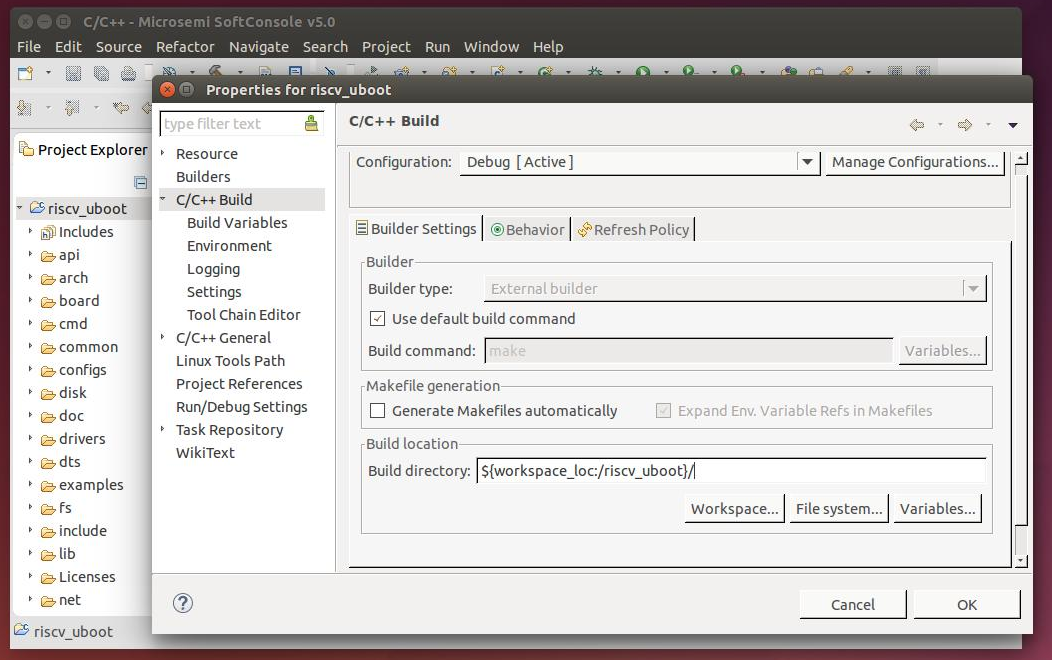
Press next.



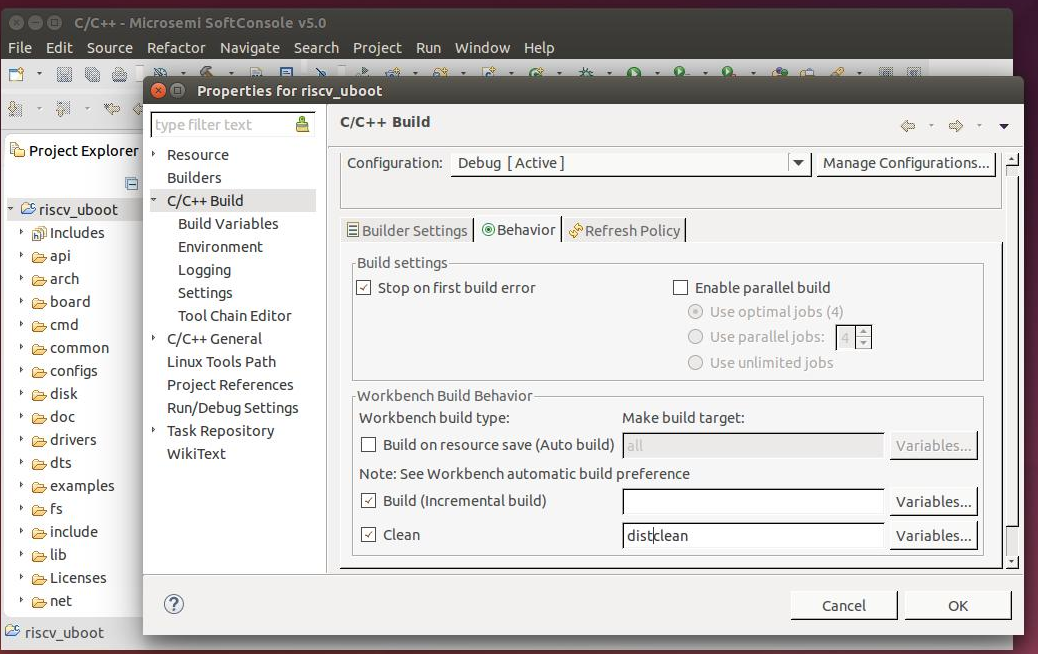
Browse for RISC-V U-Boot source directory on local machine and select all files and Click on Finish.

5. Set the Build process – **Right click on Project -> Properties -> C/C++ Build**





C/C++ Build -> **Builder** S**ettings**

* Uncheck “Generate Makefiles Automatically”
* Build directory to the project, remove 'Debug'

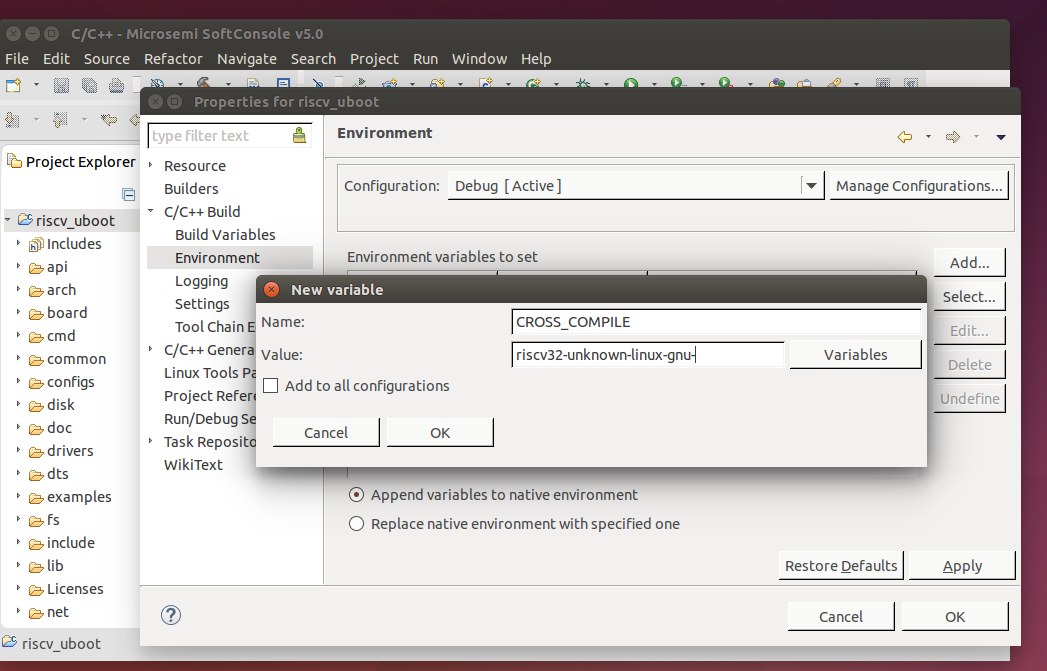
Click on Apply.

C/C++ Build -> **Behavior** settings

* Uncheck “Build on resource save”
* Clear “Build(incremental build)”
* Write 'distclean' on “Clean”

Click on Apply.

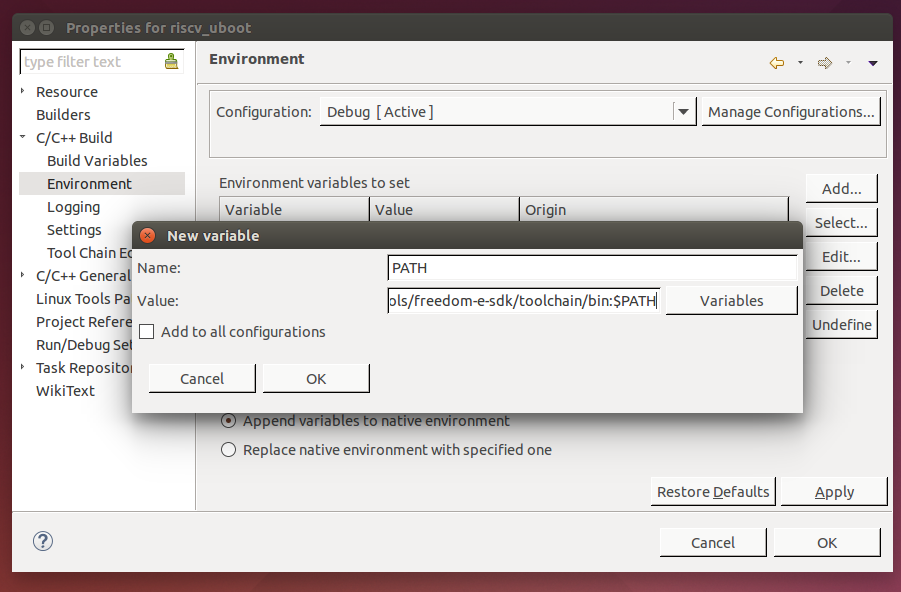
5.1. **Set the RISC-V cross tools -** C/C++ Build -> Environment -> add (twice)

****

Name : CROSS\_COMPILE

Value : riscv32-unknown-linux-gnu-

Click on OK

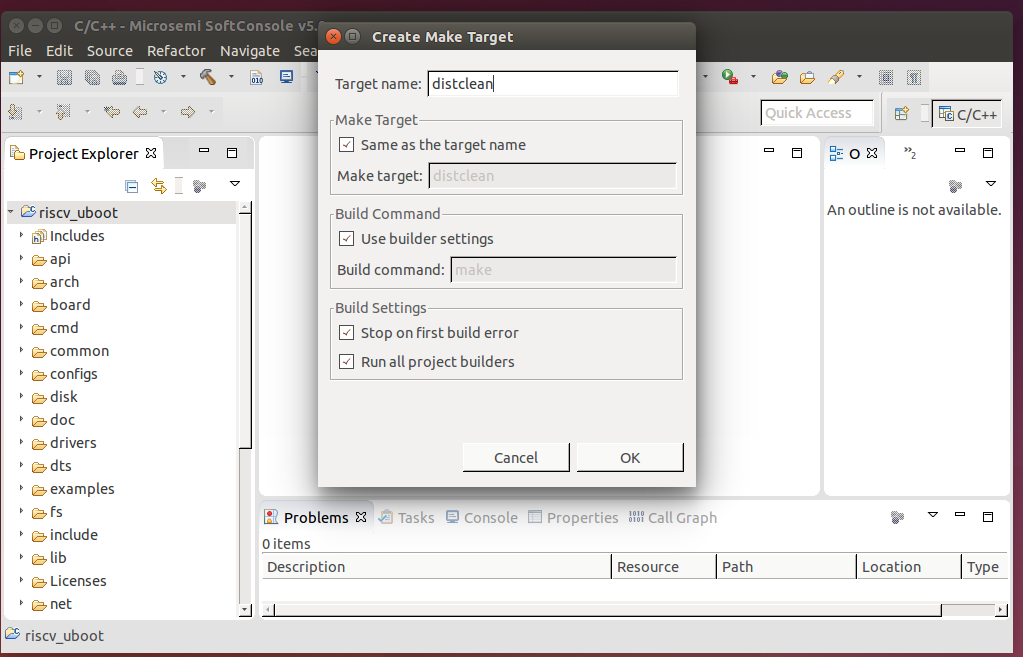
****

Name: PATH

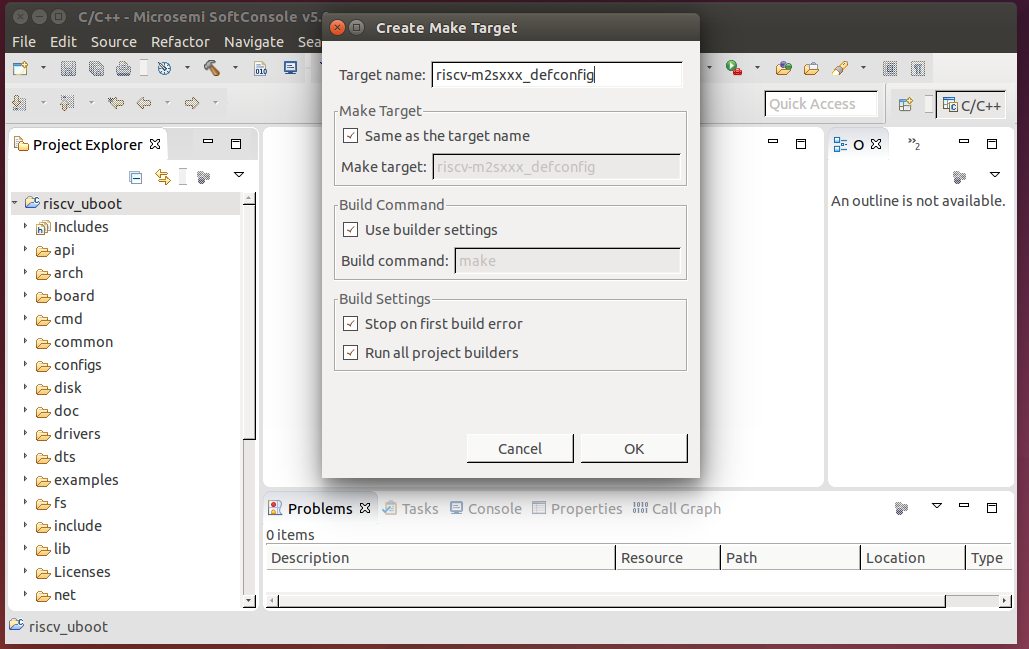
Value : [risc-v cross tool installation directory]/bin:$PATH

Click on OK and Apply... OK.

5.2. **Create Make Targets – Right Click on Project -> Make Targets -> Create** (twice)

****

Target Name: distclean

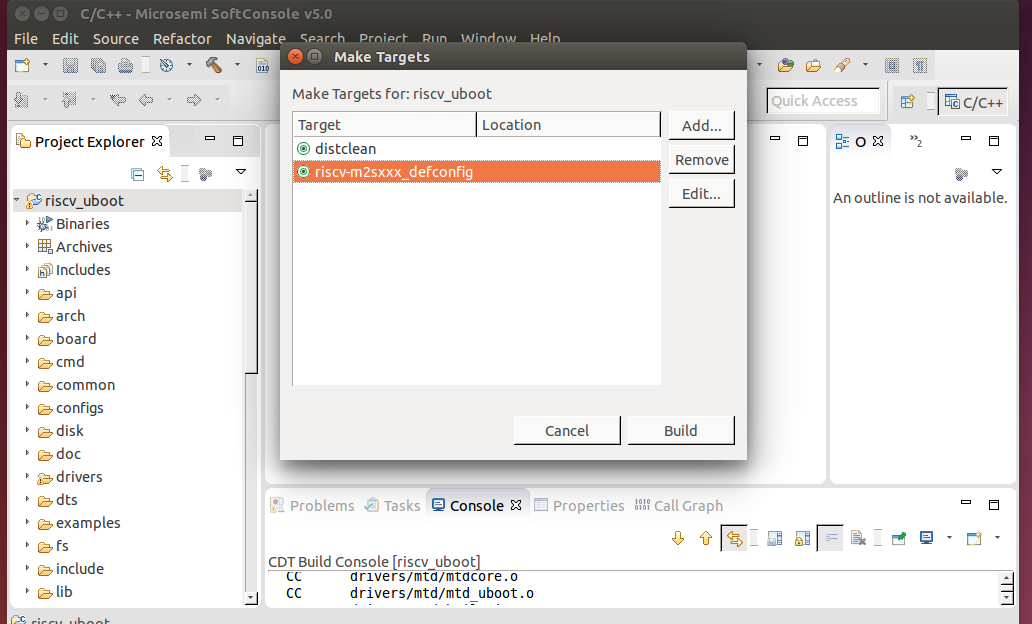
****

Target Name : riscv-m2sxxx\_defconfig

Click on OK

5.3. **Build U-Boot**

* Set macro CONFIG\_SYS\_TEXT\_BASE to SDRAM base address(0x80000000) in include/configs/riscv-m2sxxx.h
* Right Click on Project -> Clean Project
* Right Click on Project -> Make Targets -> Build

Select 'riscv-m2sxxx\_defconfig' and press Build

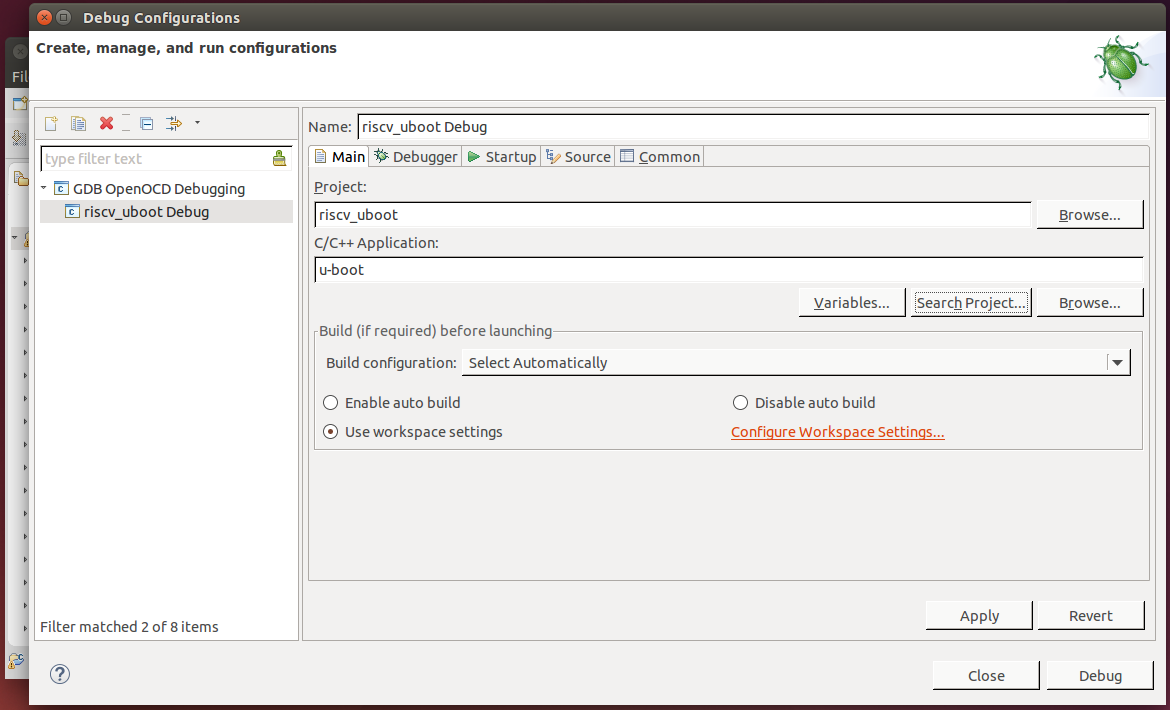
* Right Click on Project -> Build Project
* The u-boot output files available in project root directory(u-boot.bin, u-boot...etc)

**6. U-Boot Debugging Setup**

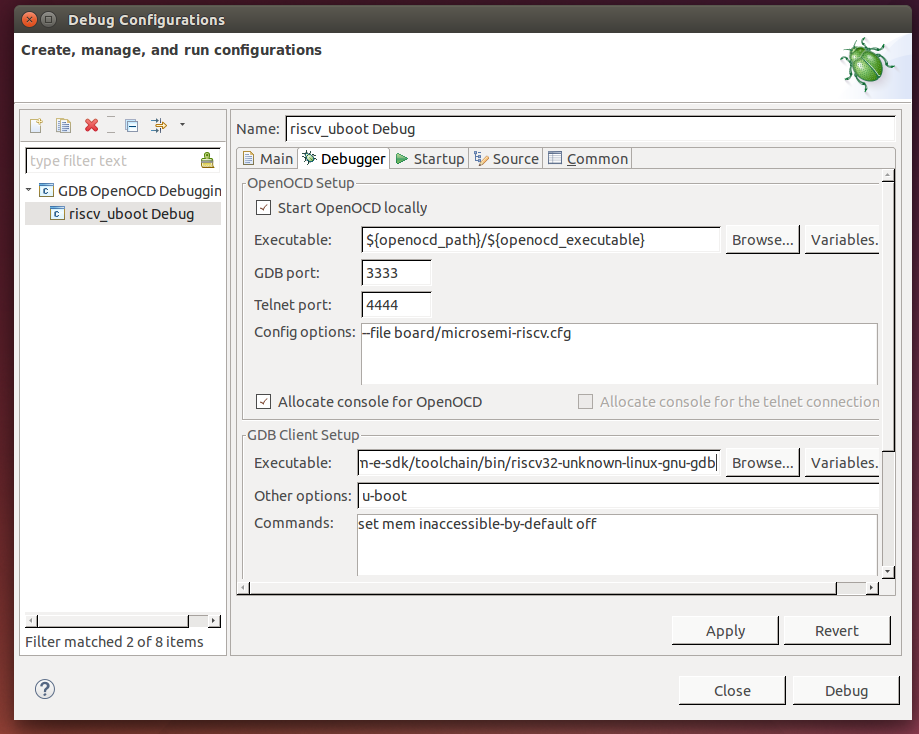
* Right Click on Project -> Debug As -> Debug Configurations

On GDB OpenOCD Debugging - Right Click and select New

GDB OpenOCD Debugging -> **Main**

C/C++ Application : u-boot

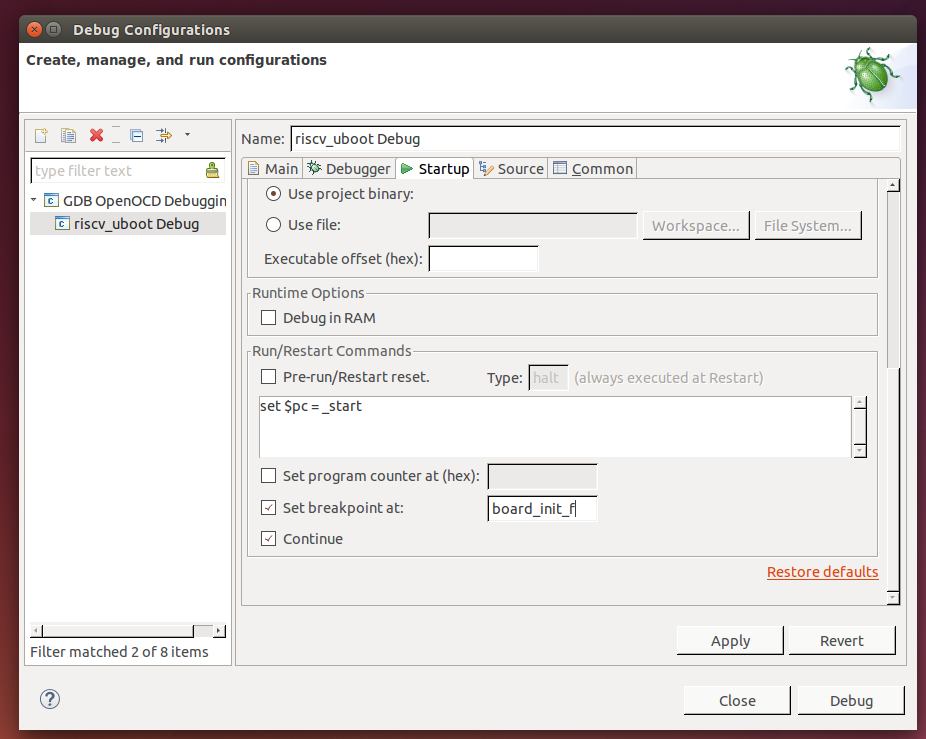
GDB OpenOCD Debugging -> **Debugger**



Edit GDB Client Setup section

* Executable : [risc-v cross tool installation directory]/bin/riscv32-unknown-linux-gnu-gdb
* Other options : u-boot

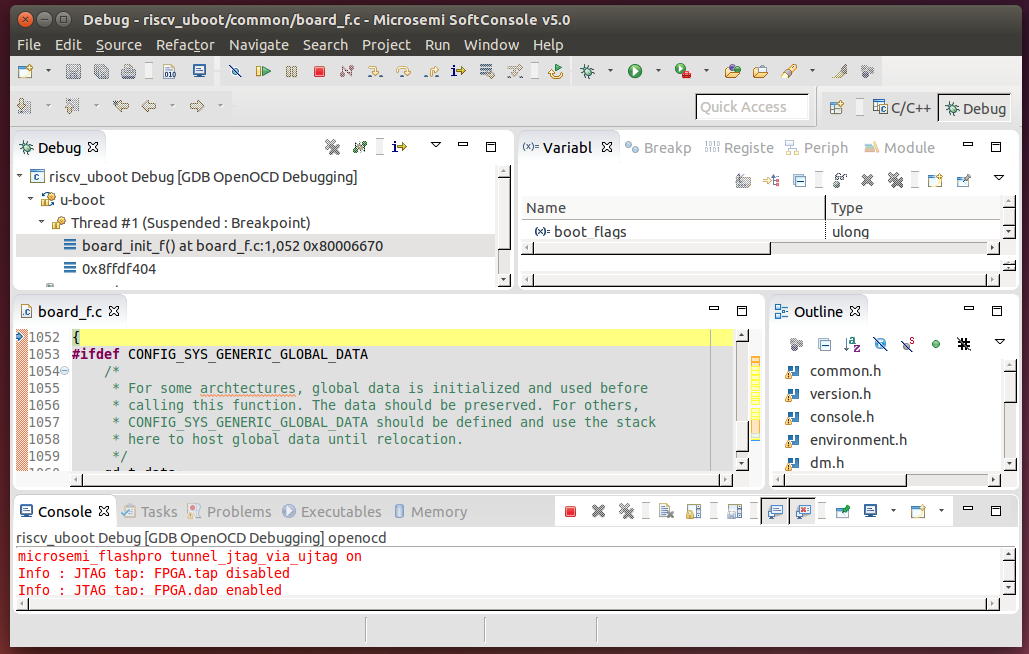
GDB OpenOCD Debugging -> **Startup**



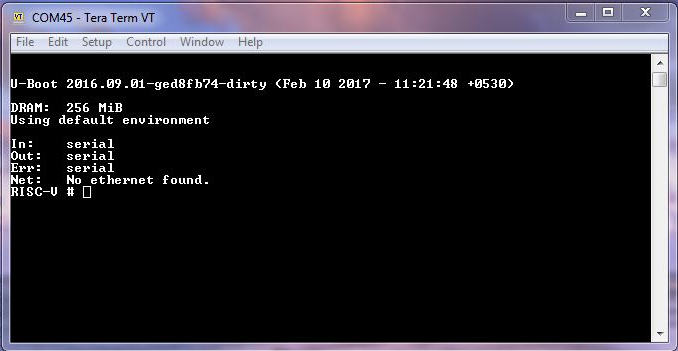
Set breakpoint at : board\_init\_f

Click on Apply and to start debugging Press “Debug”.

Before Pressing “Debug” - SmartFusion2 M2S150 Advance Dev Kit is powered ON and FlashPro5 and USB-serial cable is connected between the Board and PC. USB-serial baud rate is 115200.



Run the U-Boot and see the messages on serial terminal.



7. **U-Boot Run From eNVM**

* Set macro CONFIG\_SYS\_TEXT\_BASE to eNVM base address(0x60000000) in include/configs/riscv-m2sxxx.h
* Right Click on Project -> Clean Project
* Right Click on Project -> Make Targets -> Build

Select 'riscv-m2sxxx\_defconfig' and press Build

* Right Click on Project -> Build Project
* The u-boot output files available in project root directory(u-boot.bin, u-boot-nodtb.bin...etc).
* Generate ihex file from binary file of u-boot.

**$ riscv32-unknown-linux-gnu-objcopy -I binary -O ihex u-boot-nodtb.bin u-boot.ihex**

* Change u-boot.ihex to u-boot.ihx
* Upload u-boot.ihx file in eNVM using Libero design(available on R**ISC-V-on-Microsemi-FPGA Github**) and generate staple file.
* Program staple file on Smartfusion2 M2S150 Adv Dev Kit using FlashPro5.
* On reset, the serial terminal displays the u-boot messages.