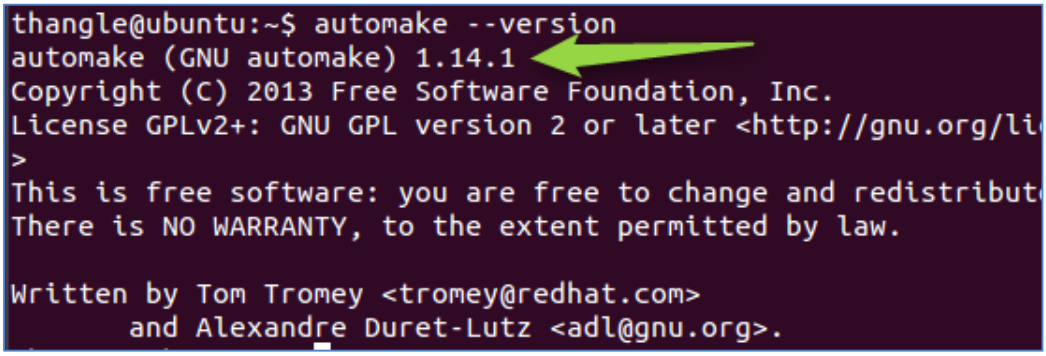
Create Mixed C/C++ project for ARM with Eclipse Autotools

www.letrungthang.blogspot.com

1. Environment setup

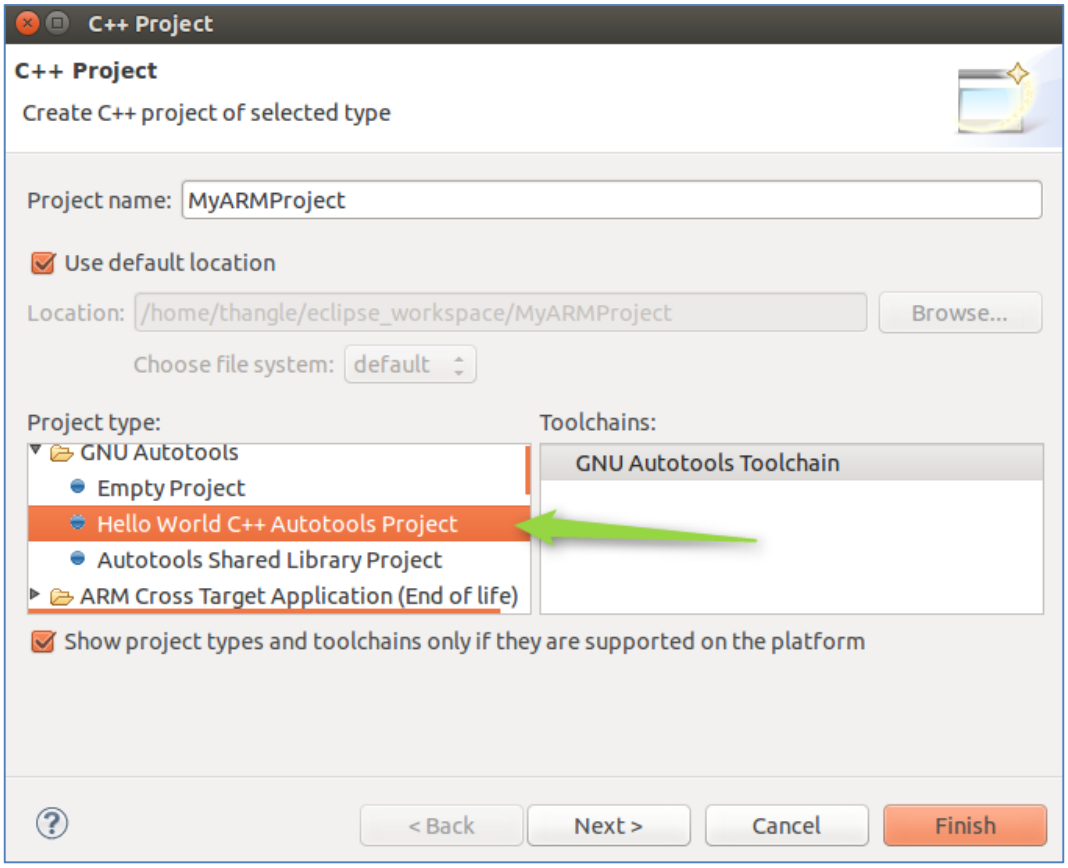
Assuming that you installed and setup GCC for ARM (Linux x86). Else, reference here.
Be sure that you installed automake, autoconf, autoreconf. Else, you need install it.
To install Automake: *sudo apt-get install automake*
This also will install Autoreconf and Autoconf for you.
To check version of automake: *automake --version*.



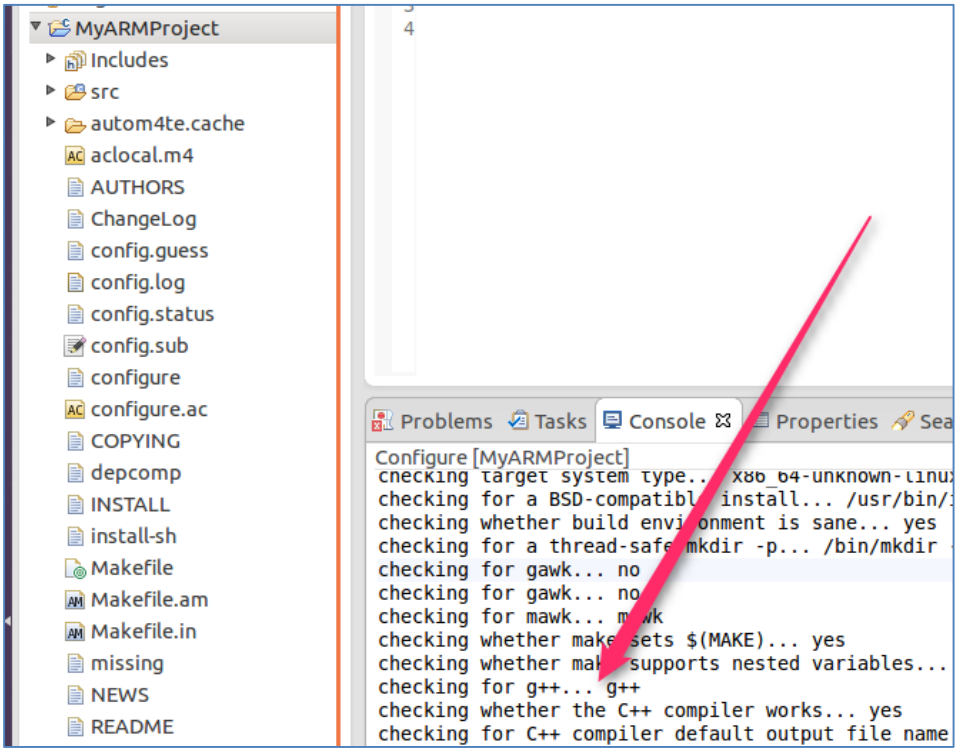
2. Create an C++ Autotools project

Assuming that your GCC for ARM is installed at:
/Home/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_GNU_Linux/
Also assuming you are creating a mixed C/C++ project for AT91SAM9260 SoC (Core arm926ej-s).

- In Eclipse, select **File -> New -> C++ Project**. Select *GNU Autotools*.



Click **Finish** to complete project creation. You will have a new created project as below.



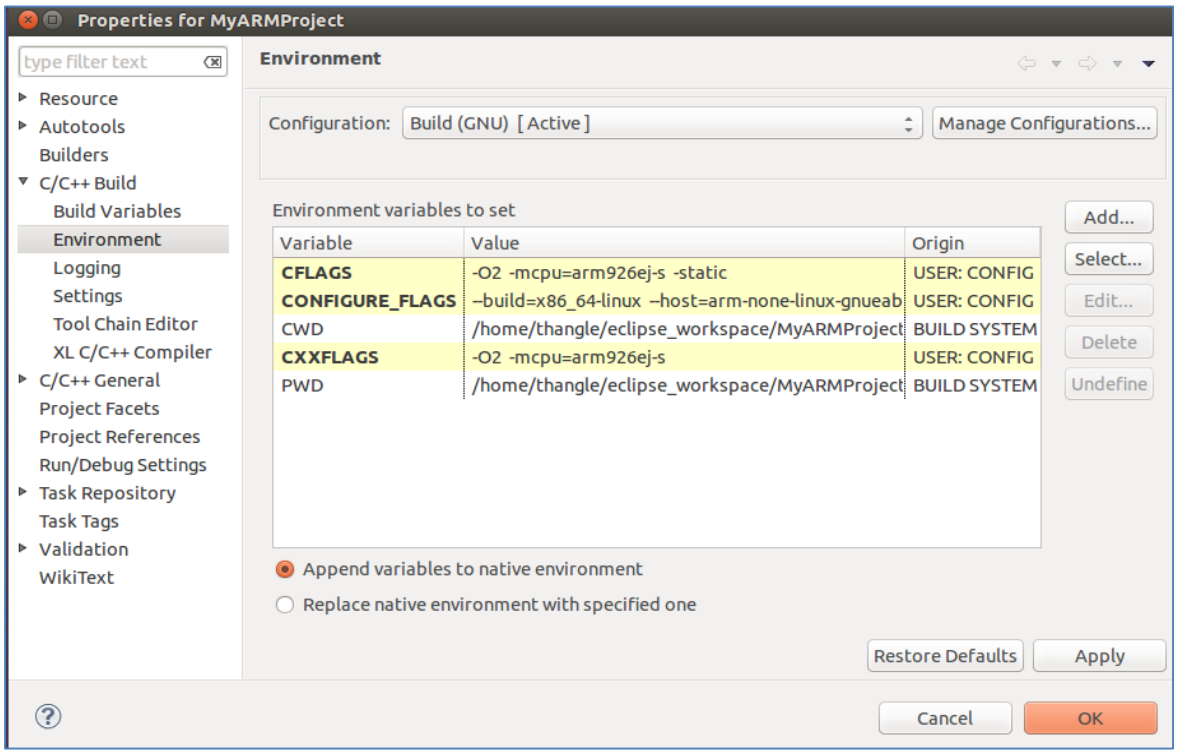
- Now, Autotools still takes g++ (x86) as default compiler. So, you need tell Autotools to take GCC for ARM when it builds your project. Go to **C/C++ Build Environment** and configure variables as below.

CONFIGURE_FLAGS = `--build=x86_64-linux --host=arm-none-linux-gnueabi --target=arm-none-linux-gnueabi CROSS_COMPILE=arm-none-linux-gnueabi- ARCH=arm`

CXXFLAGS = `-O2 -mcpu=arm926ej-s`
CFLAGS = `-O2 -mcpu=arm926ej-s -static`

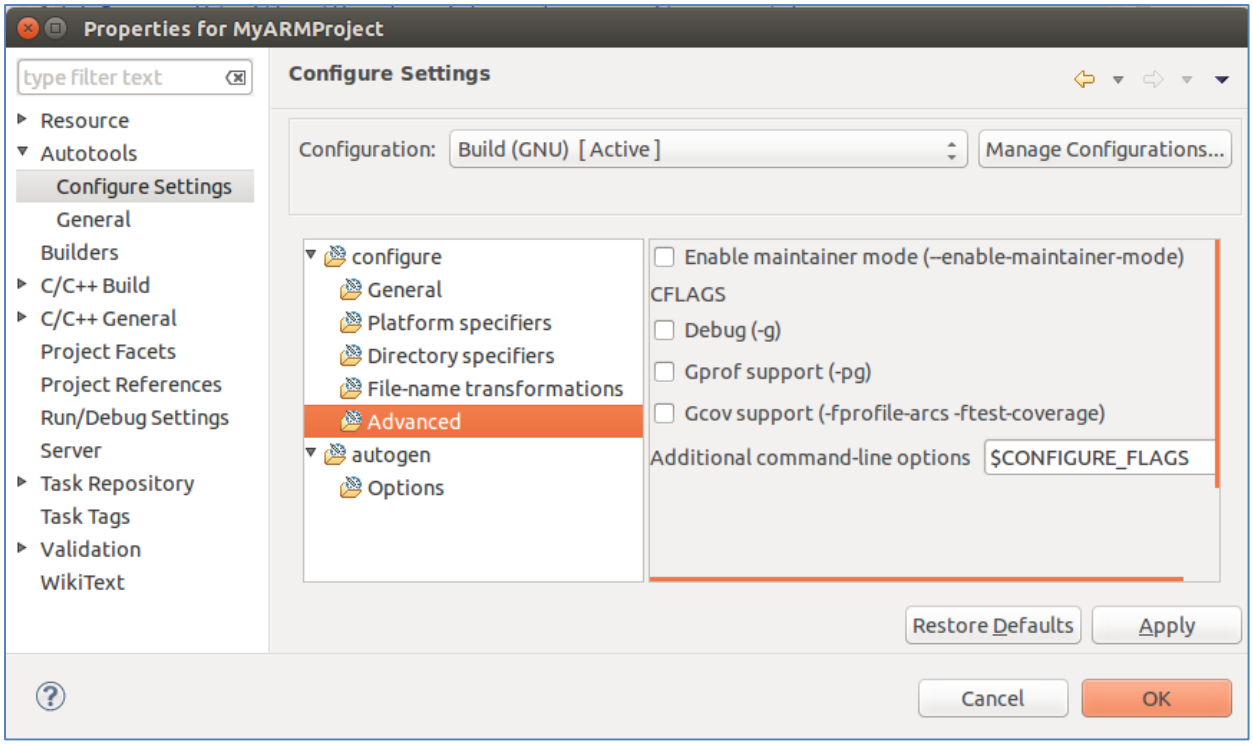
Note: You need to change the option –mcpu to respective CPU in your target.
For example: `-mcpu=cortex-a15` for ARM Cortex A15 core. More info reference in GCC page here.
Below is list of CPUs supported by Sourcery CodeBench Lite 2013 (arm-2013.05-24-arm-none-linux-gnueabi).

arm1020e arm1020t arm1022e arm1026ej-s arm10e arm10tdmi arm1136j-s arm1136jf-s arm1156t2-s arm1156t2f-s arm1176jz-s arm1176jzf-s arm2 arm250 arm3 arm6 arm60 arm600 arm610 arm620 arm7 arm70 arm700 arm700i arm710 arm7100 arm710c arm710t arm720 arm720t arm740t arm7500 arm7500fe arm7d arm7di arm7dm arm7dmi arm7m arm7tdmi arm7tdmi-s arm8 arm810 arm9 arm920 arm920t arm922t arm926ej-s arm940t arm946e-s arm966e-s arm968e-s arm9e arm9tdmi cortex-a15 cortex-a5 cortex-a7 cortex-a8 cortex-a9 cortex-m0 cortex-m1 cortex-m3 cortex-m4 cortex-r4 cortex-r4f cortex-r5 ep9312 fa526 fa606te fa626 fa626te fa726te fmp626 generic-armv7-a iwmmxt iwmmxt2 mpcore mpcorenovfp native strongarm strongarm110 strongarm1100 strongarm1110 xscale



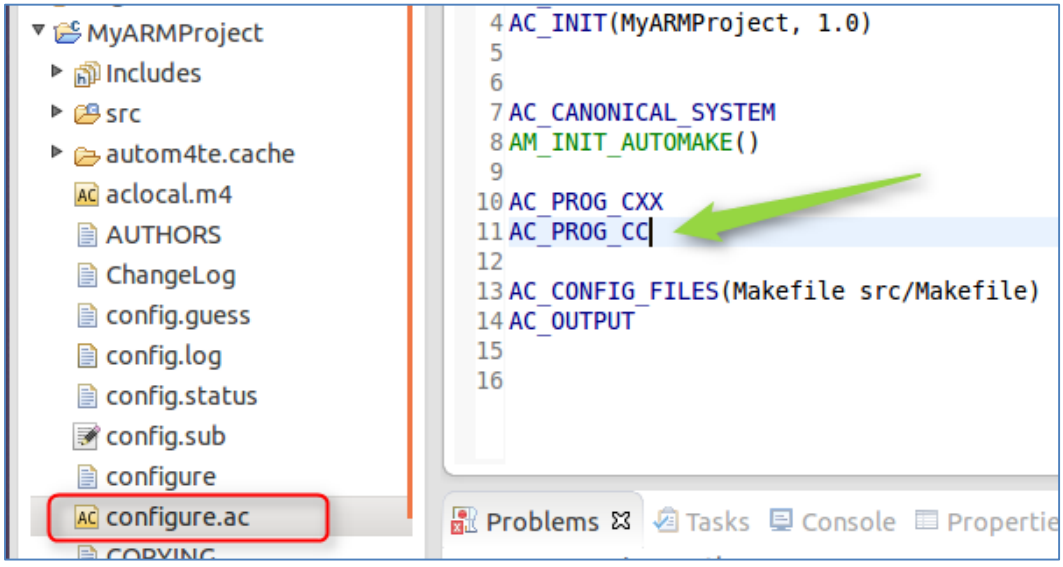
- Next is to setup configure for the project.

In Eclipse, go to project **Properties -> Autotools -> Configure Settings -> Configure -> Advanced**
Put **$CONFIGURE_FLAGS** to *Additional command-line options* box.

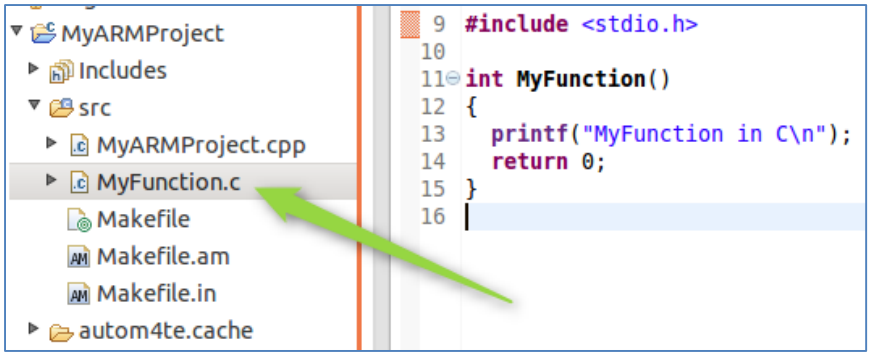


- Because the project mixed both C and C++ code, so to add 2 instructions below to the file **configure.ac**

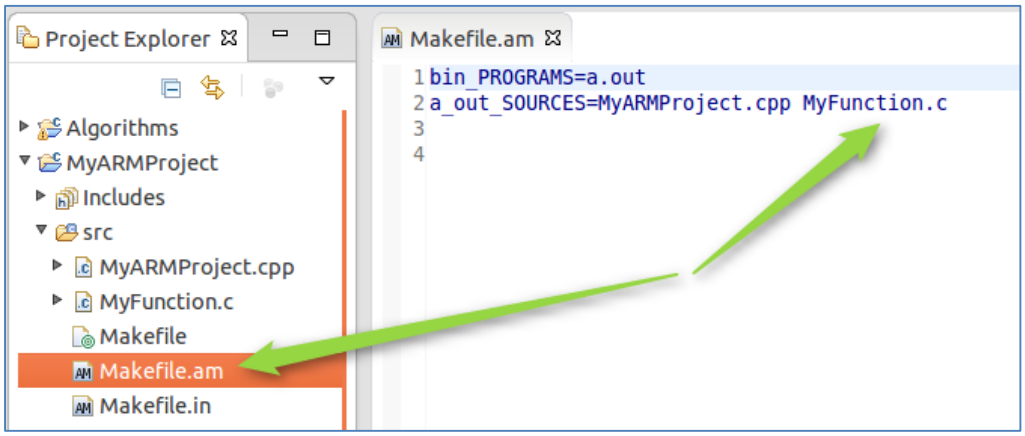
```
AC_PROG_CXX
AC_PROG_CC
```



So far, you still don't have any C source file in your project, so just add it. For simple, I added a C source file **MyFunction.c** with only a simple function as below.

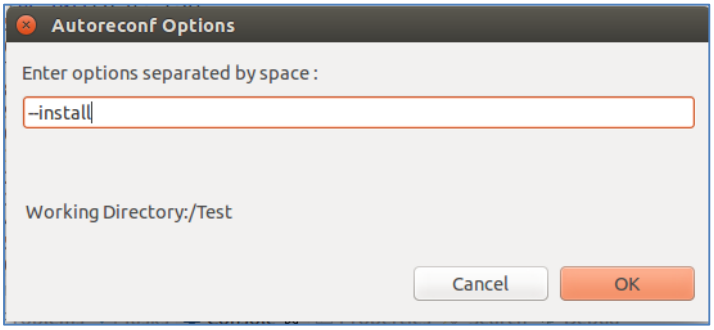


You need inform Autotools about this new C source file by adding its name to file **makefile.am**



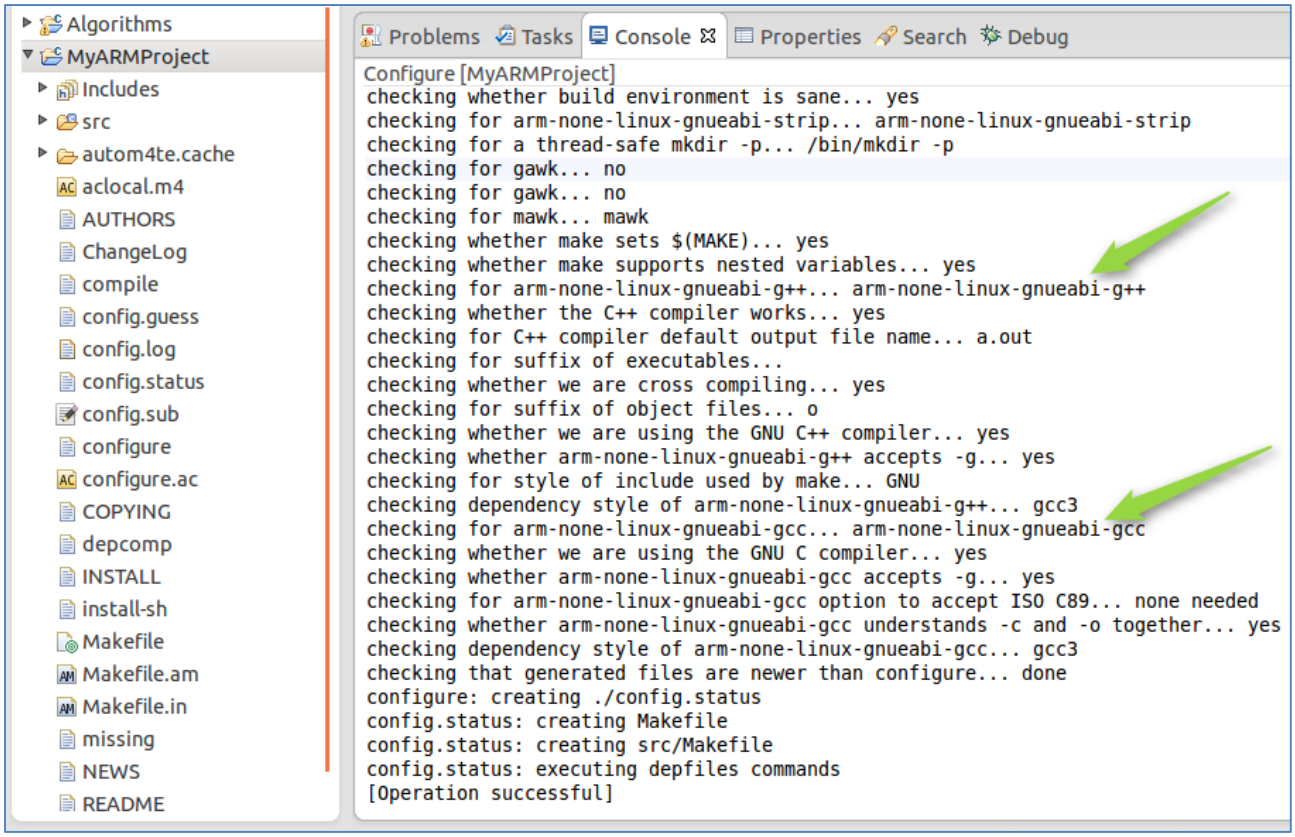
3. Build project

- In Eclipse, right click and run **Invoke Autotools -> Invoke Autoreconf** and type `--install` in **Autoreconf** box. This is requested only to do **one time** when you create your project. Automake will install necessary things for your project.

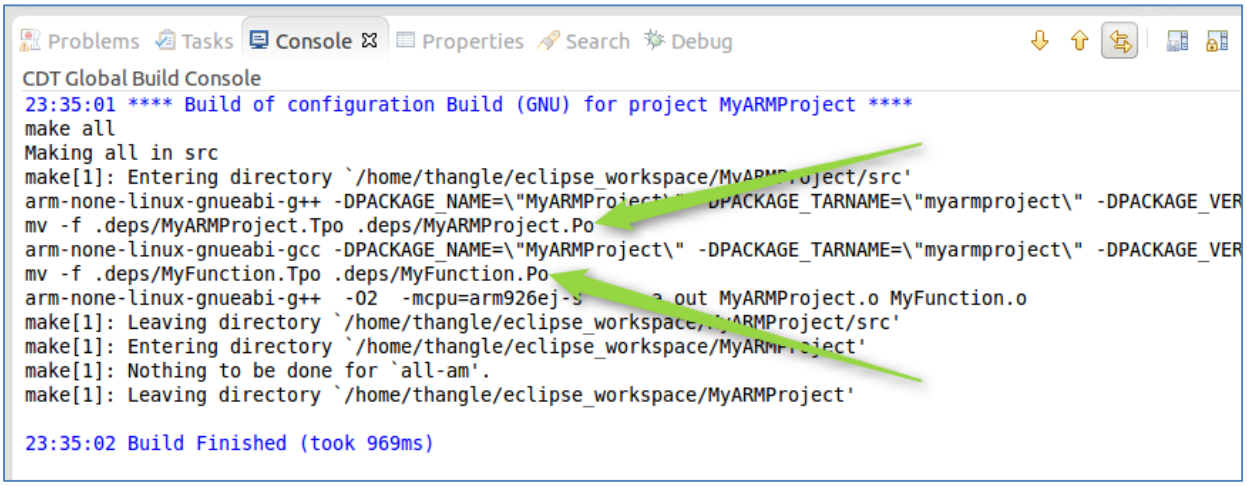


- Next, In Eclipse right click and run **Reconfigure Project**. This only need do once when you create new project and you change something in makefile.am, configure.ac.

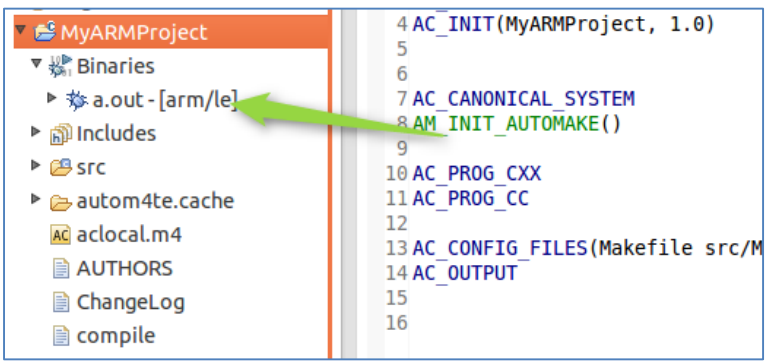
If build successful, you could see messages as below



- Last, in Eclipse right click on top of project and select **Build Project**.
  If build successful, you will see messages below.

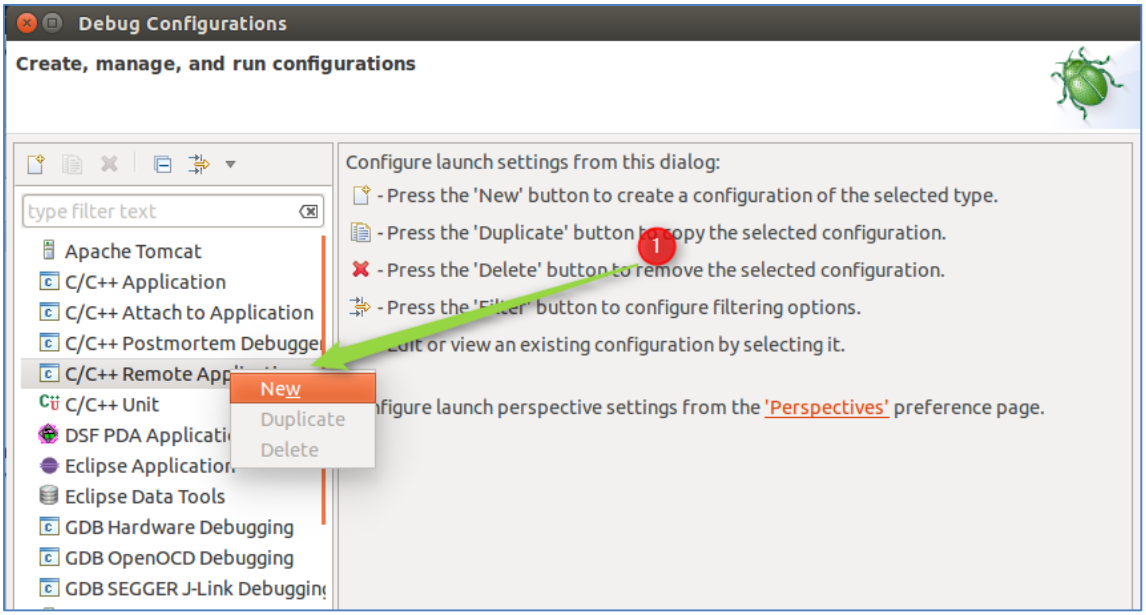


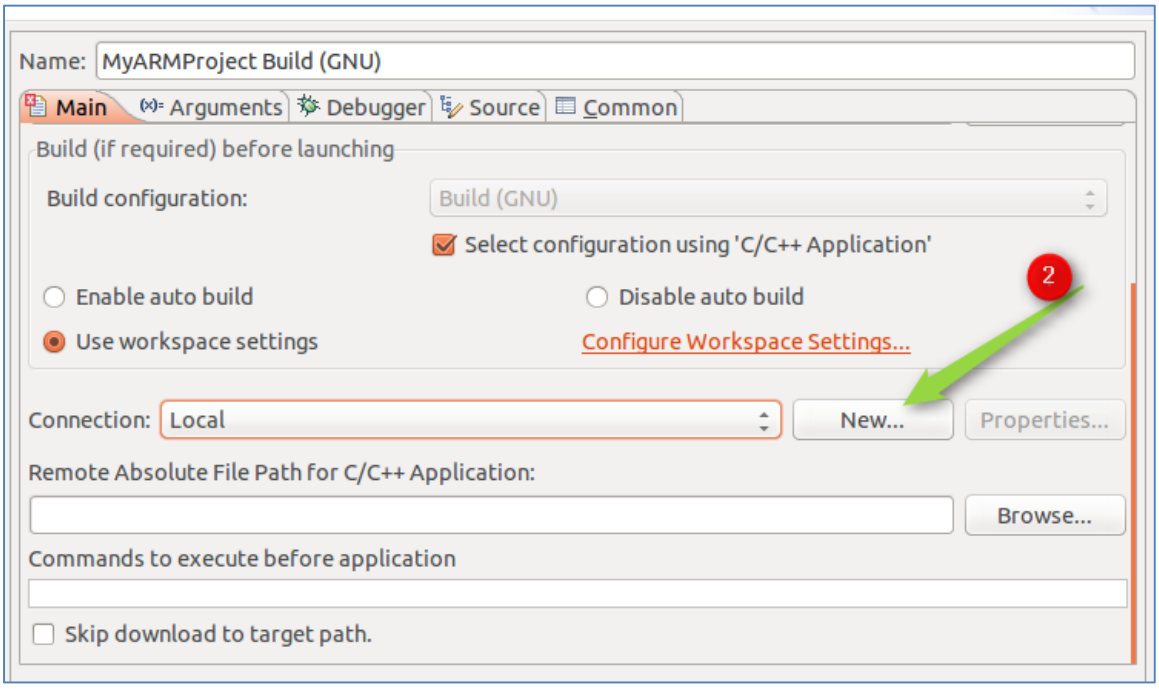
A binary (**a.out**) is generated. It is executable file that can run on your target device.

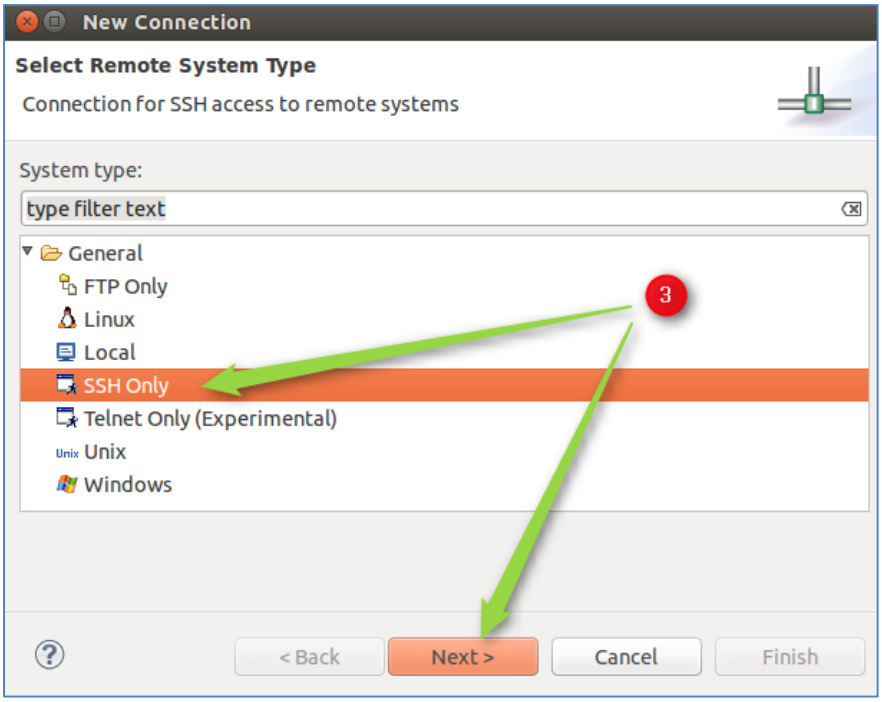


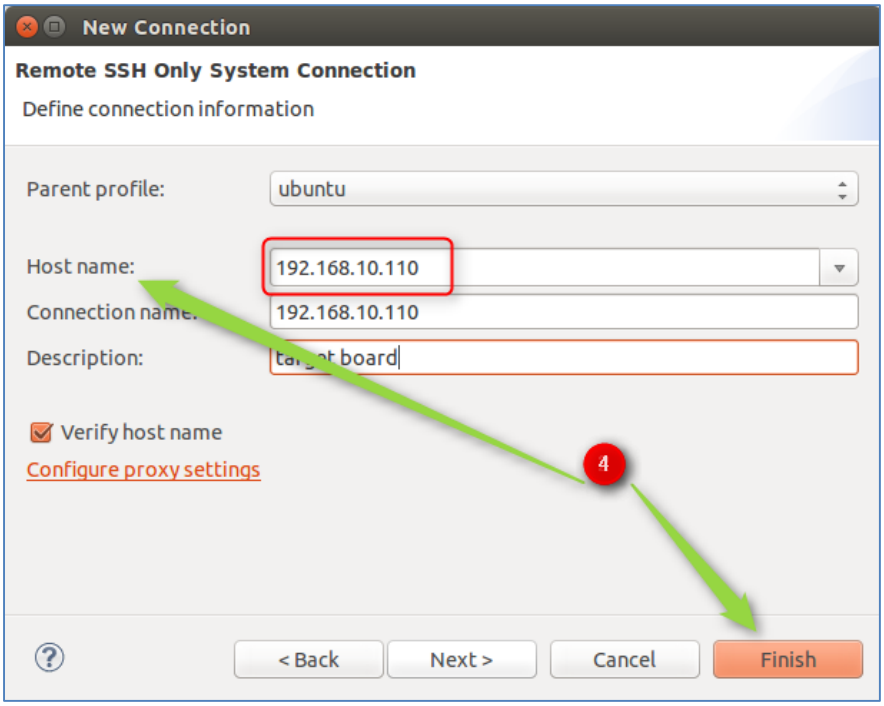
4. **Run and Debug your program**

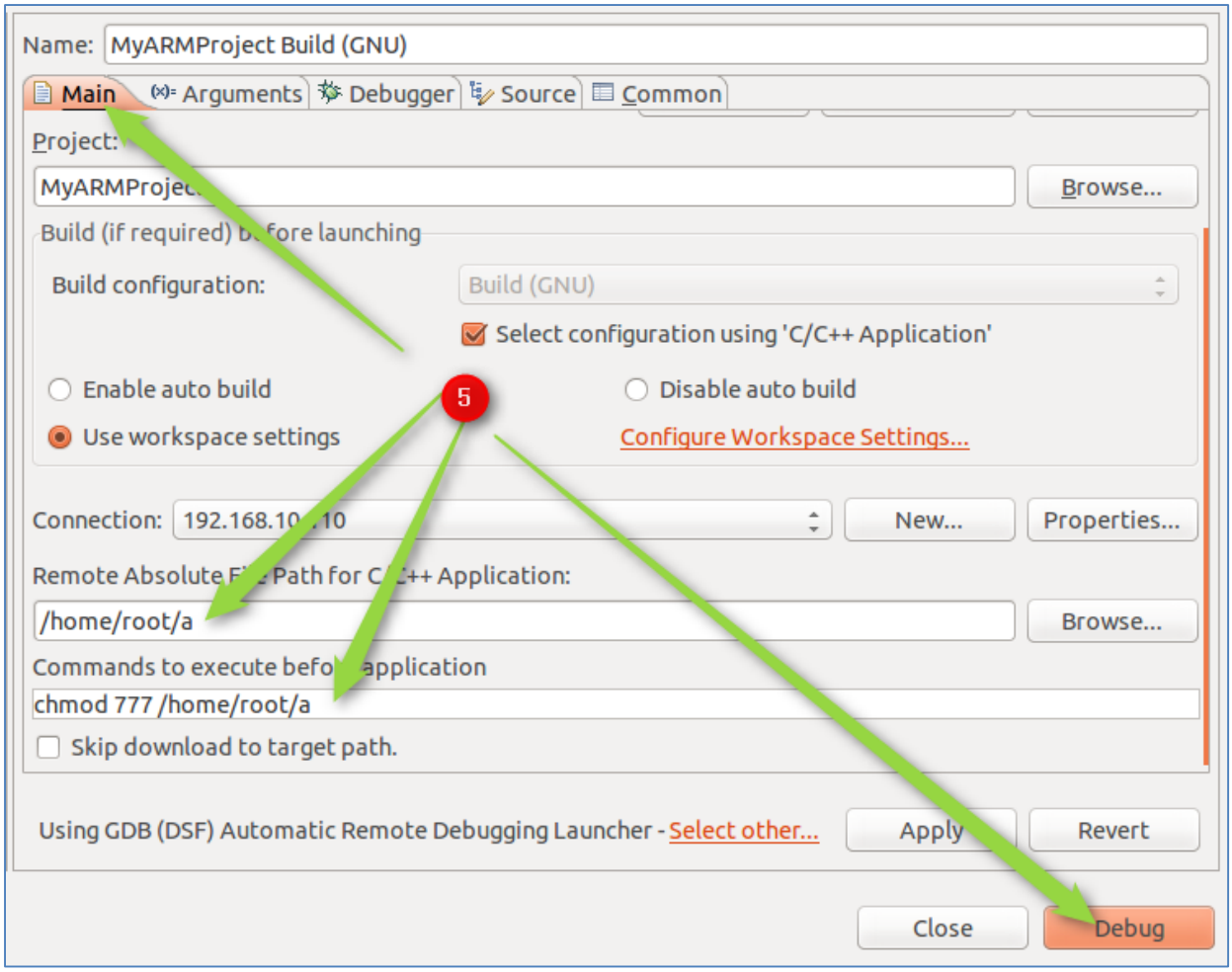
You can copy generated binary to target and run it manually from your target or you can run directly it from Eclipse by configuring as below if you are able to connect to your target via Ethernet. You can debug only if your target installed gdb-server. Below is steps to configure Eclipse debugger

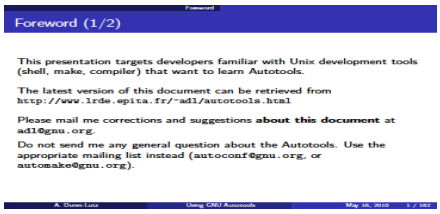





New Connection
Select Remote System Type
Connection for SSH access to remote systems
System type:
type filter text
General
FTP Only
Linux
Local
SSH Only
Telnet Only (Experimental)
Unix
Windows
3
< Back
Next >
Cancel
Finish


New Connection
Remote SSH Only System Connection
Define connection information
Parent profile:
ubuntu
Host name:
192.168.10.110
Connection name:
192.168.10.110
Description:
target board
Verify host name
Configure proxy settings
4
< Back
Next >
Cancel
Finish



5. References

- Example source code in Git hub (Host PC is Ubuntu 64bit).
- Autotools manual