MODULE TESTING PROCESS

Heated Glass Stage 1000
Version 1.0

02 May 2016

Project Code	Document No	Author	Client Name
PN15100106	TD16042901	DAK/MJC;TSJ	

REVISION HISTORY

Date	Version	Description	Author
24/01/2016	1.0	Initial Draft	DAK/MJC
29/04/2016	2.0	With python script	TSJ

Integrated Systems Development Limited

Table of Contents

1. MODULE TESTING OVERVIEW

Need to check stack pointer.

Data memory.

1.1 Code review

Will be conduct by both the testing engineer and coder together

1.2 Functional testing

1.2.1 Correct operation

Will be conduct by testing engineer. Will be run on the required correct setup with a debugger.

The testing engineer will test all the functions written in the module except lower level functions. Needs to cover all the lines.

1.2.2 Out of boundaries check

Will be conduct by testing engineer. All lower level drivers will be mocked up.

1.2.2.1 Things to mock

- Hardware drivers like SPI, I2C – Just return with a constant value all the time. No need to vary the return value according to the request.
- Replace <util/delay.h> with mock up delay – Comment "#include <util/delay.h>" and insert "#include <moc_delay.h>"

Will be simulated within the Atmel Studio.

1.2.2.2 Things to cover

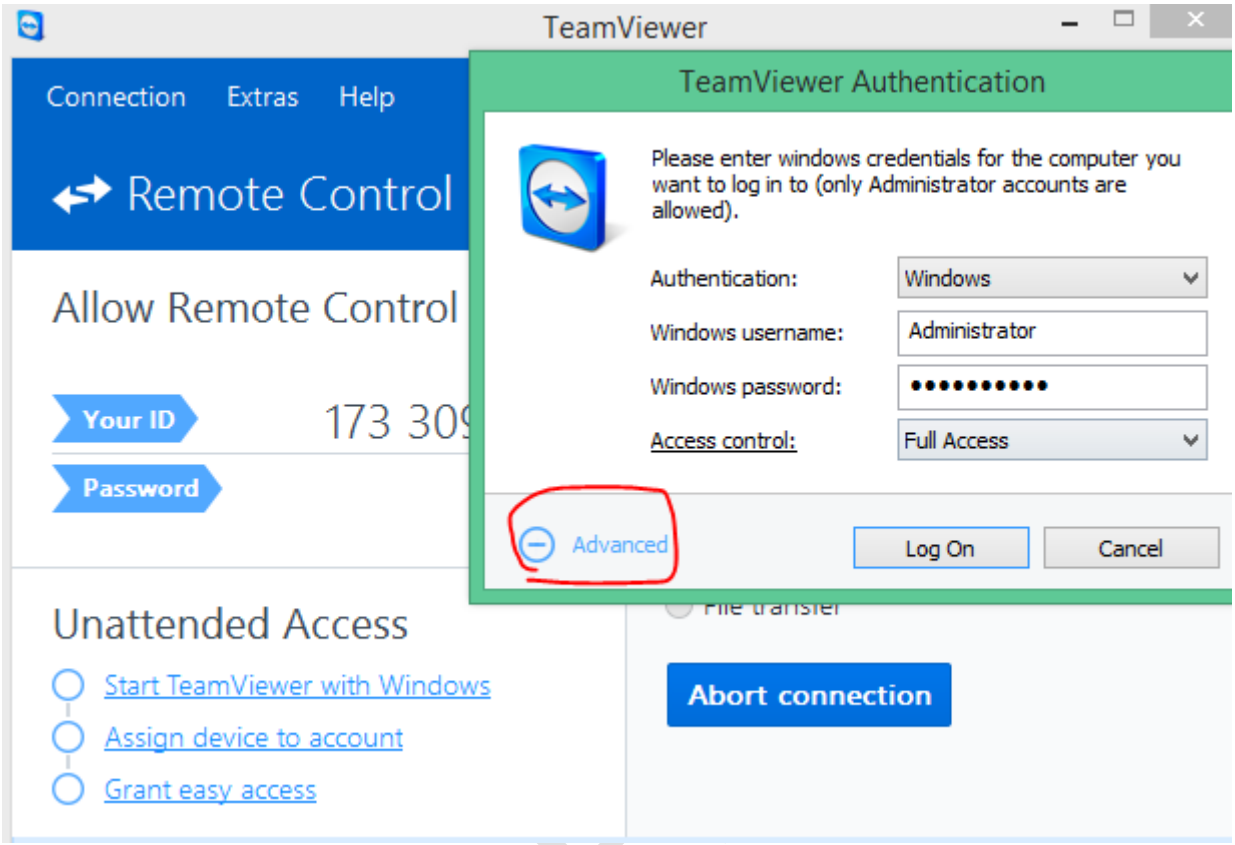
1. Overloading parameters
2. Unexpected inputs
3. Not responding to the functions calls
4. Unexpected responses even without calling

2. CTC TESTWELL

2.1 Initial setup

2.1.1 Logging into the server

1. Open TeamViewer in the host machine
2. Enter server IP address for "partner ID" and click "Connect to Partner"
3. Select "Advanced", set Authentication to "Windows" and enter credentials



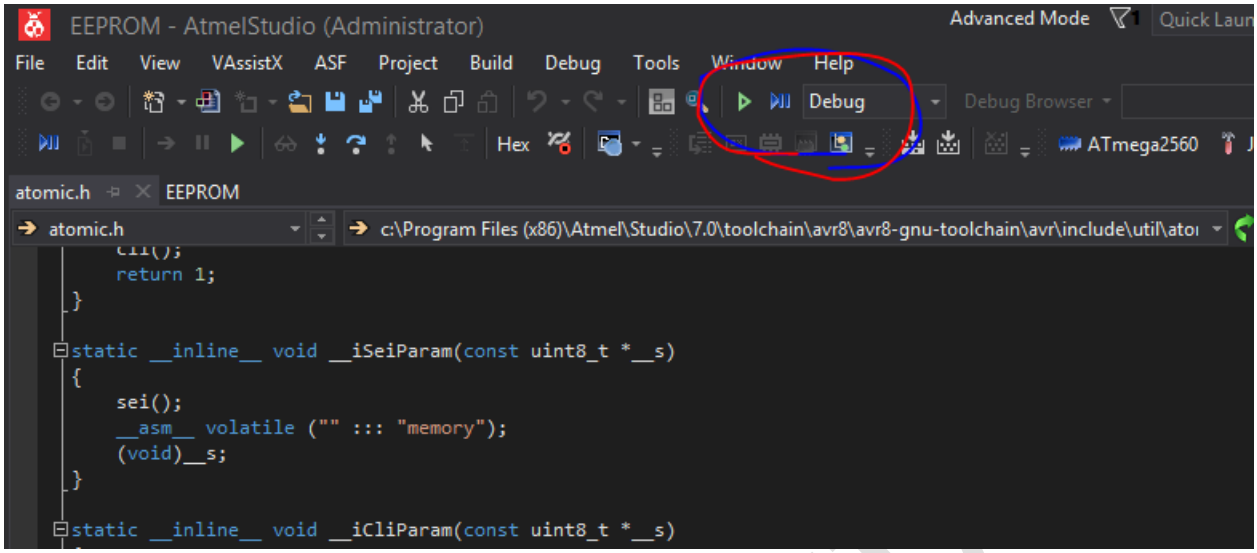
4. Log in to the server

2.1.2 Running the python script

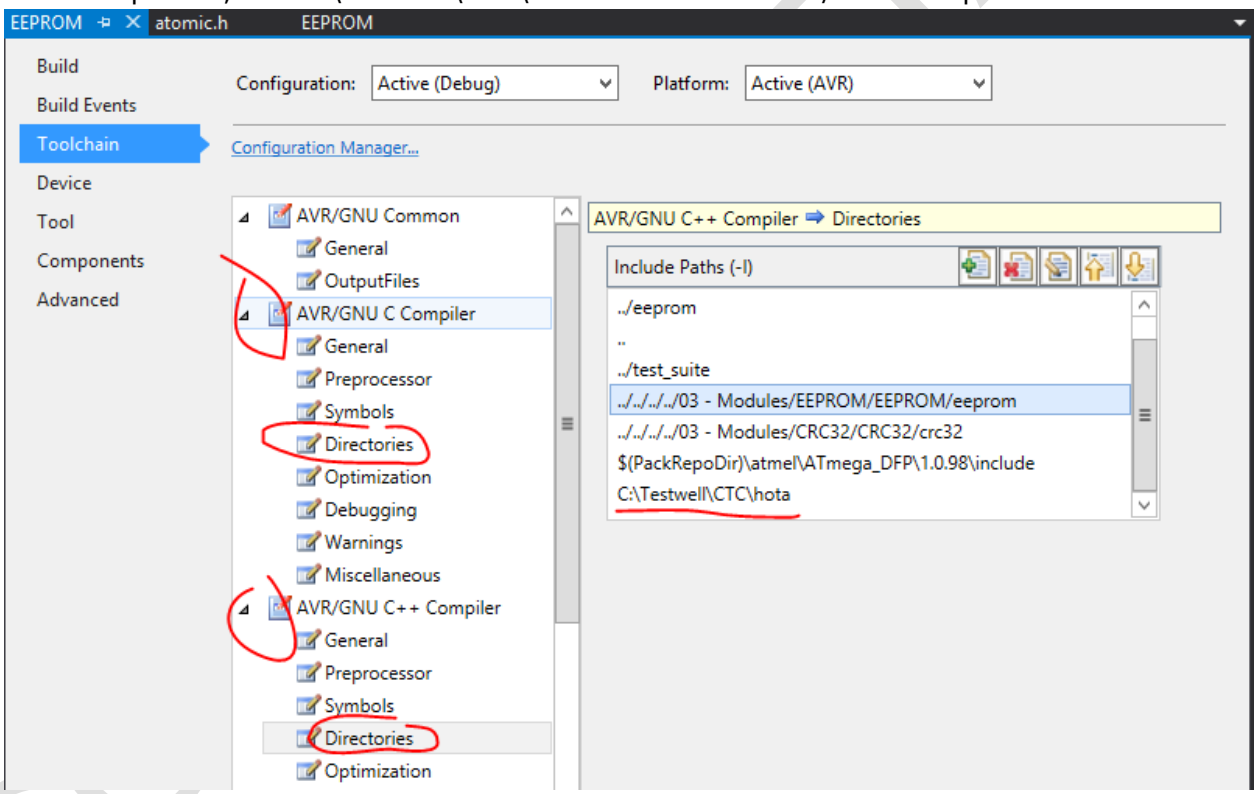
1. Browse to the project inside the server.
2. Copy the project to be tested to "08 – CTC++".
3. Copy the latest version of python script "CTC_Module_Testing.py" from "F:\01 - R & D\05 - CTC Testwell\01 - Run Coverage\Python Script" to inside of the copied project folder
4. Run the script and follow instructions carefully
5. Come to the document again when prompted "**** FOLLOW STEP 1 ****"

2.2 Step 1 – In the server

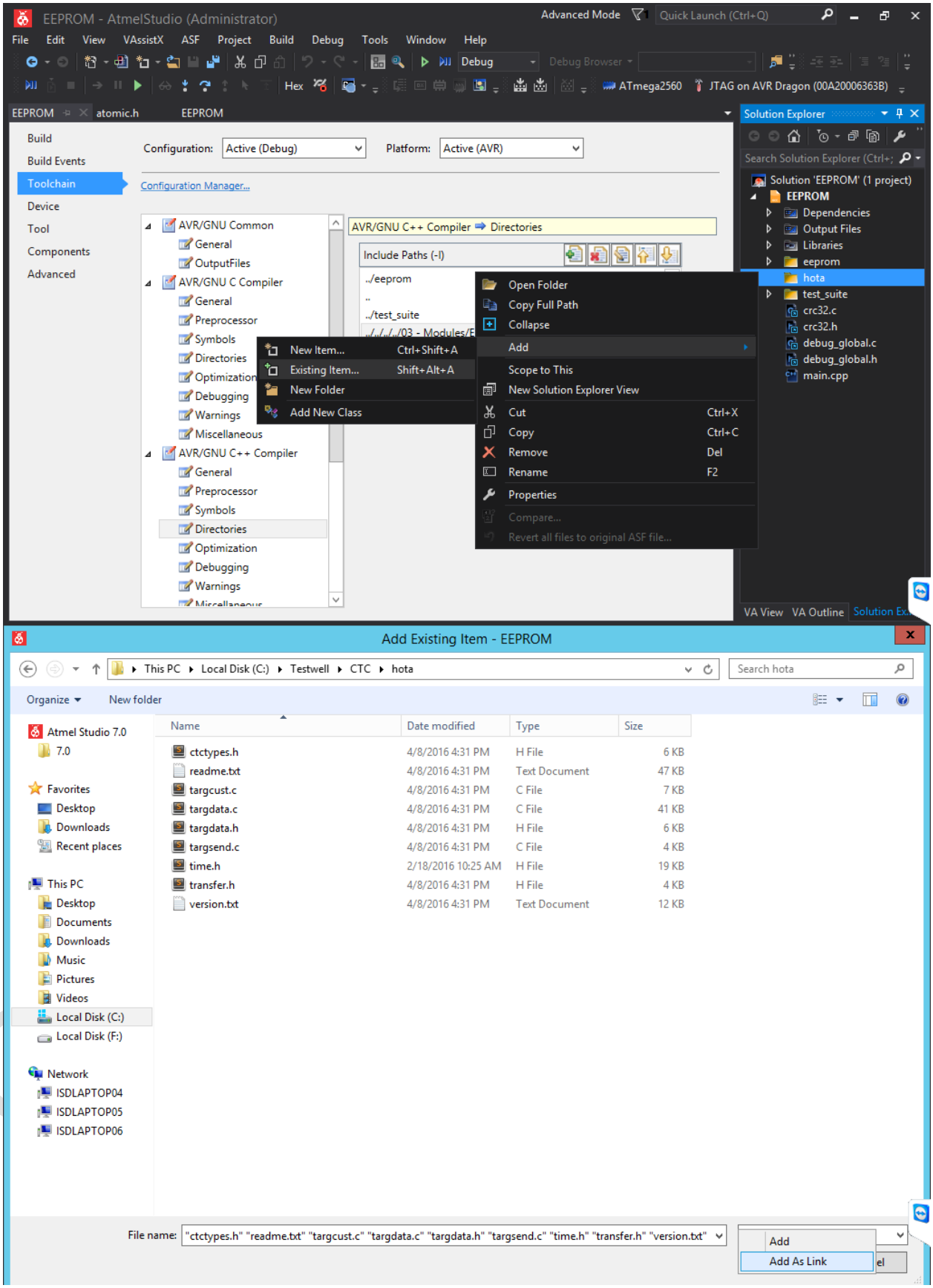
1. Open project in Atmel Studio
2. Make sure the build happens in debug mode



3. Go to Properties, Add c:\Testwell\CTC\hota to tool chain > C/C++ compiler directories

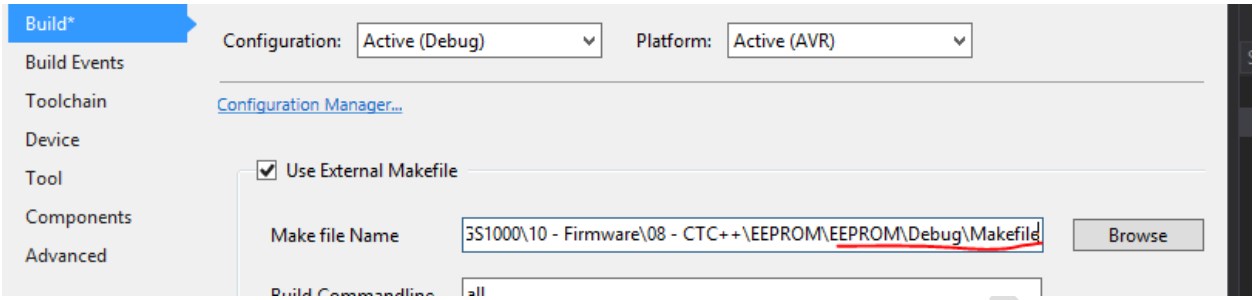


4. Under Solution explorer, add a new folder named "hota" and select add existing item and select all inside Add c:\Testwell\CTC\hota and add as a link.



5. Run "Rebuild solution"
6. (Should be no errors)

7. Go to Properties, under Build tab, select use external makefile and choose the makefile present inside the debug folder of the project.

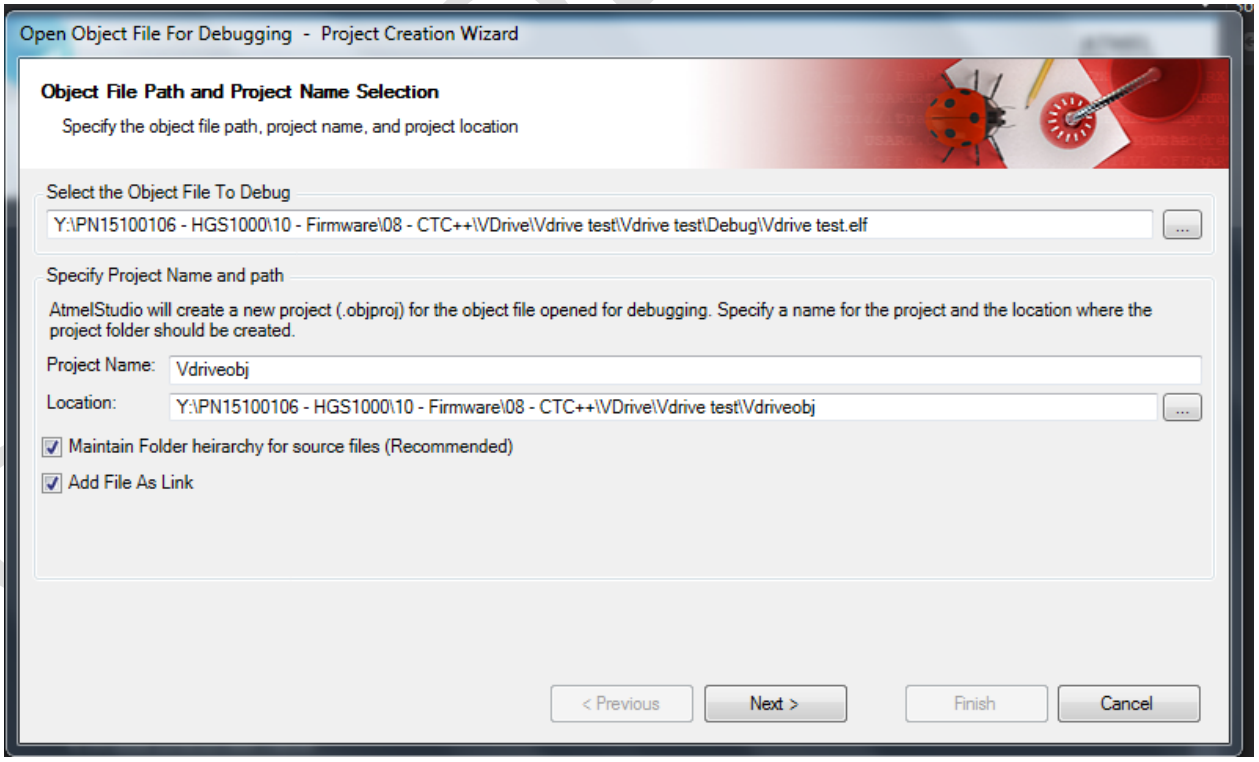


8. Under Solution explorer, add existing item and select add **CTCArray.h** and **CTCArray.c** found in F:\01 - R & D\05 - CTC Testwell\01 - Run Coverage\CTC Array Source files.
9. Inside "main.cpp" add the line "#include "CTCArray.h"

******STEP1 is over – go to TeamViewer and continue the python script.**

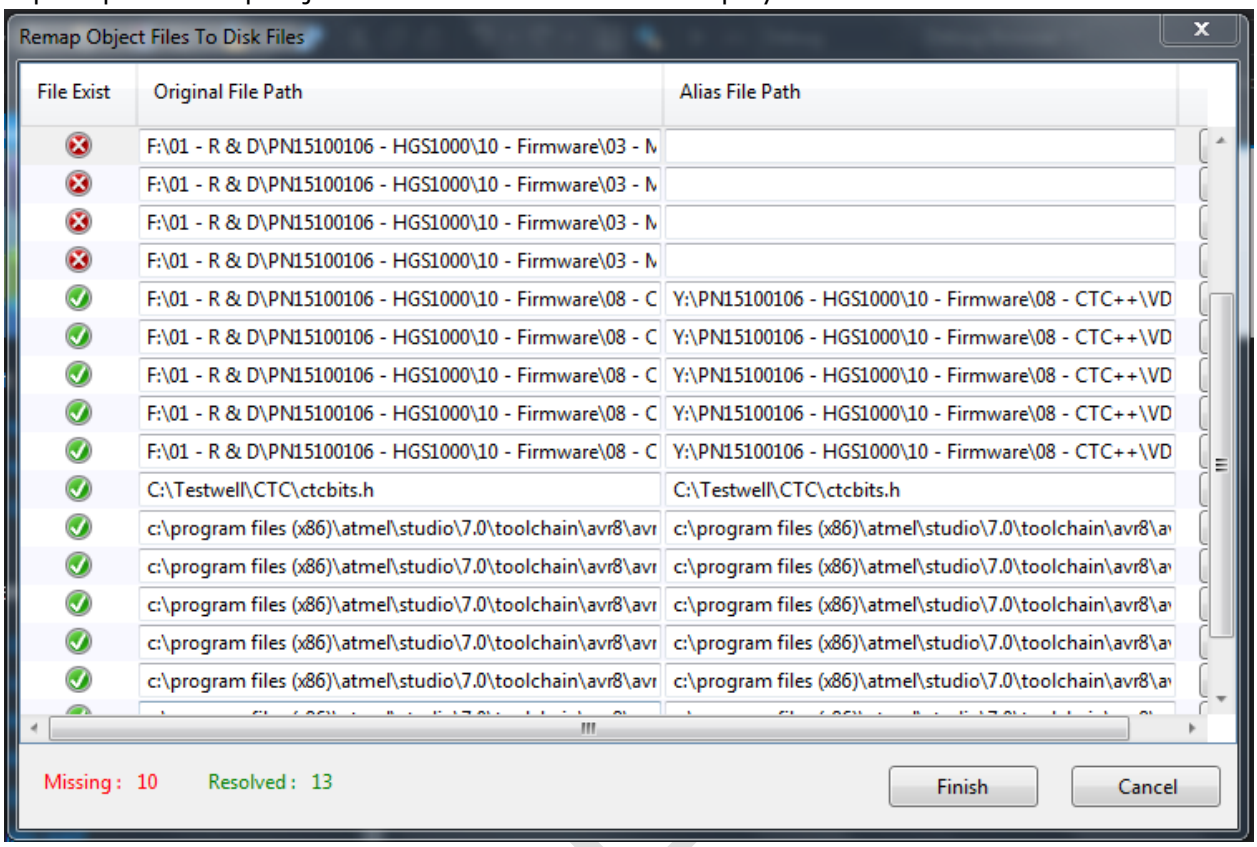
2.3 STEP 2 – On Host PC

1. Open Atmel Studio
2. Goto FILE>OPEN> Open object file for debugging

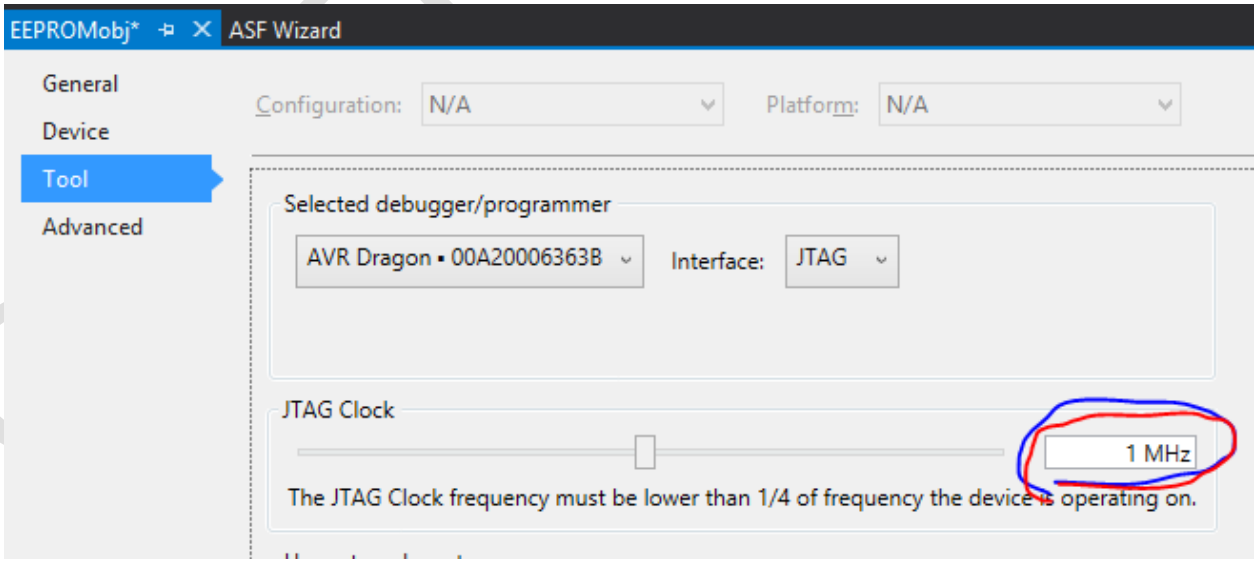


3. For the Object file to debug, browse the "**PROJECTNAME**.elf" file in the debug folder of the compiled project in the server
4. Projectname should be **PROJECTNAMEobj**
5. For Location, Choose same directory as the project directory inside 08 - CTC++
6. Select the MCU for the project

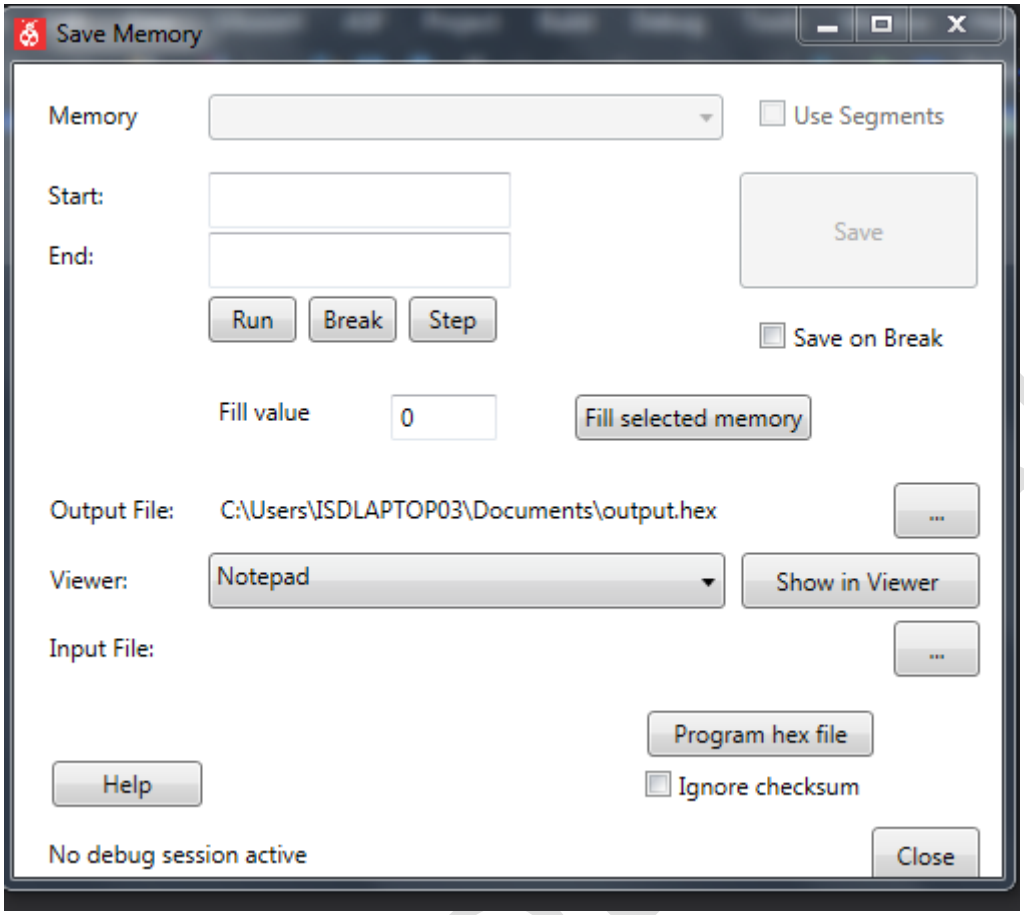
7. A prompt to remap object files to disk files will be displayed.



8. Browse only the "**CTCarray.h**" file in the project directory. Some other files would be linked at this moment.
9. Create a watch variable for CTC_array[].
10. Download the code to the MCU using the AVRDragon
11. Give 1MHz for the JTAG clock



12. Execute the test routine till the end
13. Pause if necessary
14. Observe the hex data in the array.
15. Under tools open memory logger.



16. Select data memory.
17. Start address should be as same as the address of the 0th element of the CTC_array[]
18. End address should be as same as the address of the last element of the CTC_array[]. Add the array size to initial address to obtain this.
19. Save the dump as "**output .hex**" inside the debug folder of the project.

STEP2 is over – go to teamviewer, run the "generate_HTML.bat" in the Project's Debug folder