*Octillion Power Systems India Private Limited*

302, Sector 10, Sector 17/19, MIDC, Bhosari,

Pimpri – Chinchwad, Maharastra-411026.

www.octillion.us

32 Bit Bootloader Documentation



Content

# Steps to follow to Testing the existing Bootloader v3 Pg. 2

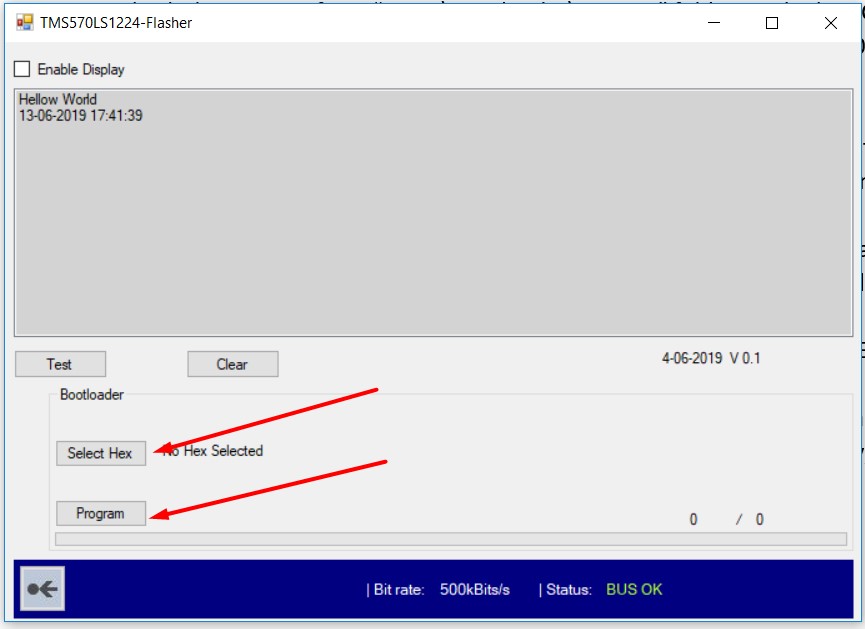
1. Steps to follow to make new Bootloader Project from Blank Page Pg. 3
2. Steps to follow for changed HalcoGen setting of Bootloader Project Pg. 9
3. Steps for setting up the Application project & Generate Hex file Pg. 11
4. Steps to change the Jump address of Application Pg. 15
5. Important Notes Pg. 17
6. Bootloader and GUI Communication State diagram Pg. 18

# A.] Steps to follow to Testing the existing Bootloader v3:

1. Download the Bootloader folder from the GitHub. And unzip the file.
2. Open both the projects from “32Bit \Bootloader\Boot v3” folder inside the CCS environment. Upload the Bootloader project named “Boot v3 Bootloader” to the TMS570 Launchpad and then Debug the project named “Boot v3 Application”. On TMS570 Launchpad Led Blink would start working.

[This will simulate the normal application running condition. For further testing the flashing activity make sure the “Memory Browser” which can be found in “View” menu inside the CCS is opened. Turn the “Continuous Refresh” on of “Memory Browser” tab.]

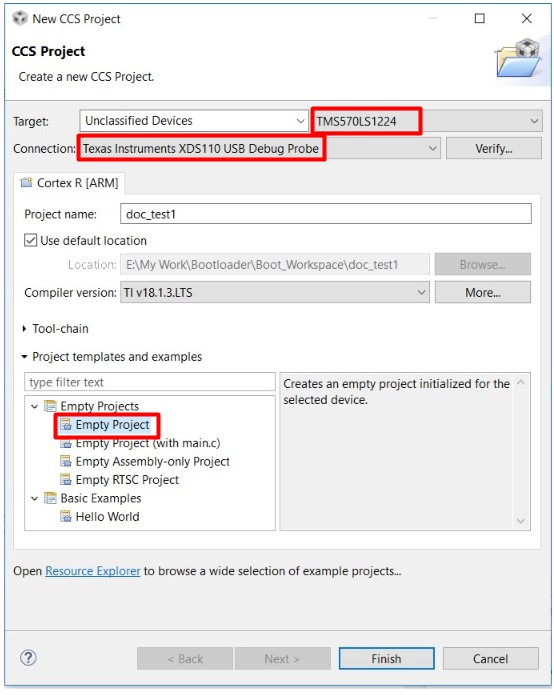
1. Make sure the “PCAN, transreceiver and TMS570 Launchpad” setup is connected and now open the GUI and make sure the GUI version is v0.1 and file is named as “32bit-Bootloader-GUI-V0.1”. The GUI executable can be found at “32Bit \GUI\32bit-Bootloader-GUI-installer-V0.1”. A PCAN initialized window will appear then click OK button which will be on the window.
2. After that GUI will open. Click the “Select Hex” button and browse to the “Boot v3 Application Int.hex” file in the folder “32Bit \Bootloader\Boot v3 \Boot v3 Application Int\Debug” and press “open” after selecting the hex file.



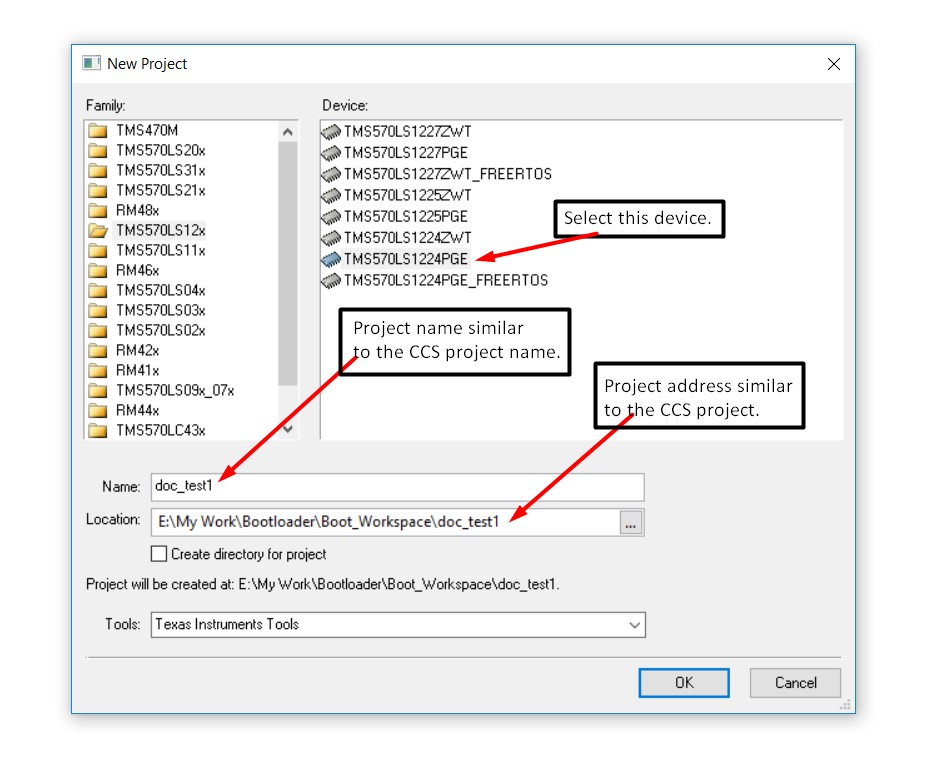
1. After selecting the hex file. Click Program button on the GUI and the flashing activity will start. The start address of application is “0x14000”. To verify the flashing activity, open debug window of CCS and go to the start address in “Memory Browser” and check whether the addresses are been programmed or not.
2. Wait until full file is been flashed and “Programming Successful” has been displayed. After that the application program of Led blink must start working normally.

# B.] Steps to follow to Make new Bootloader Project from Blank Page:

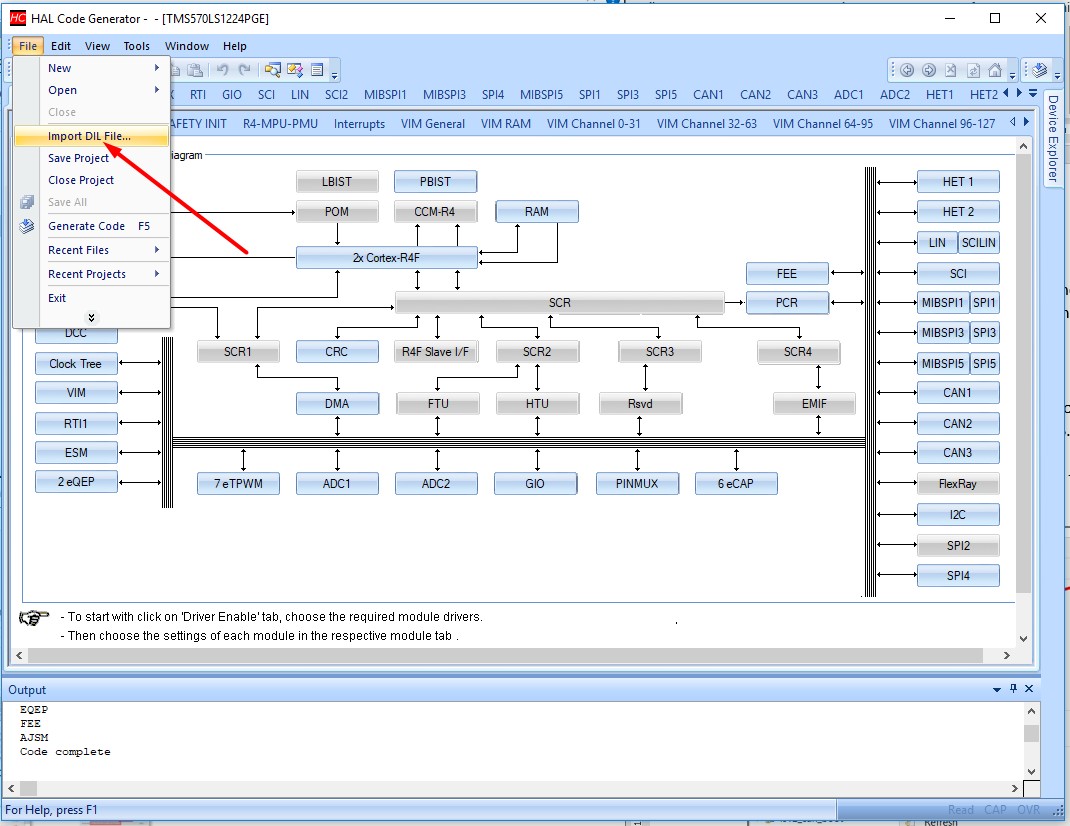
1. Open CCS and create a new project in the CCS. Make sure the red marked options are selected.



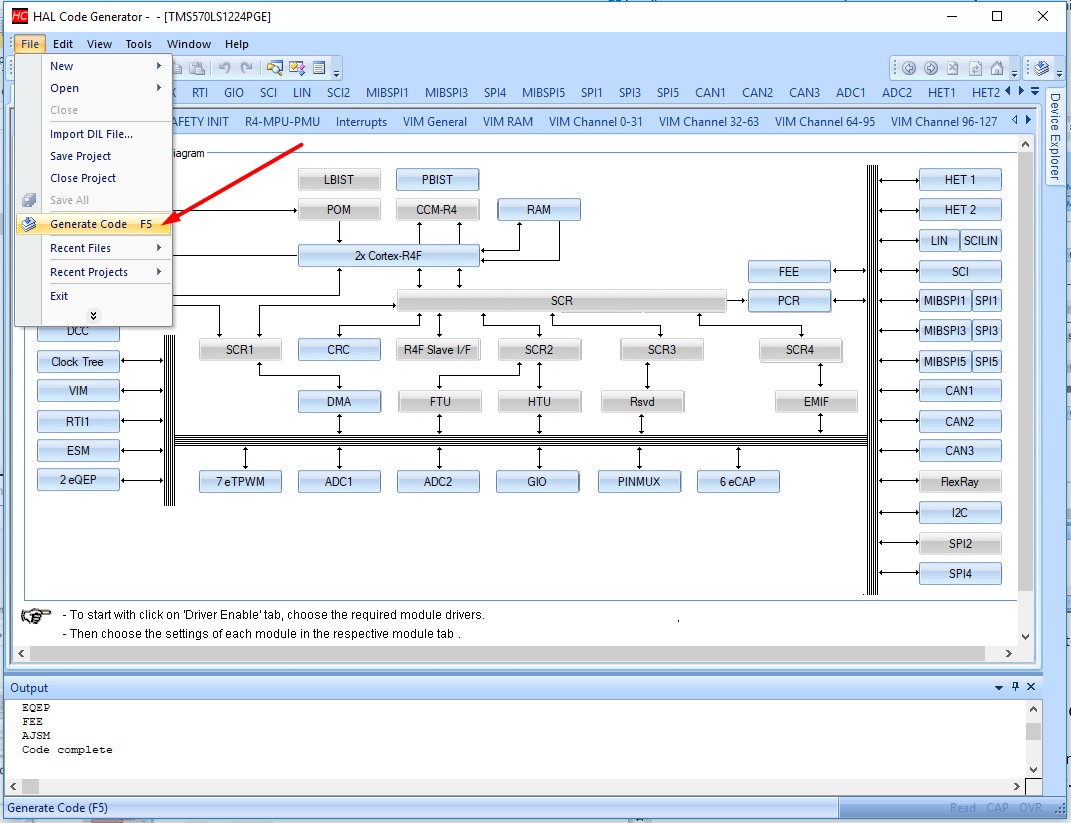
1. Open HalcoGen and create a new project and keep the name of HalcoGen project name similar to that of CCS for simplicity. Make sure the indicated things are completed.



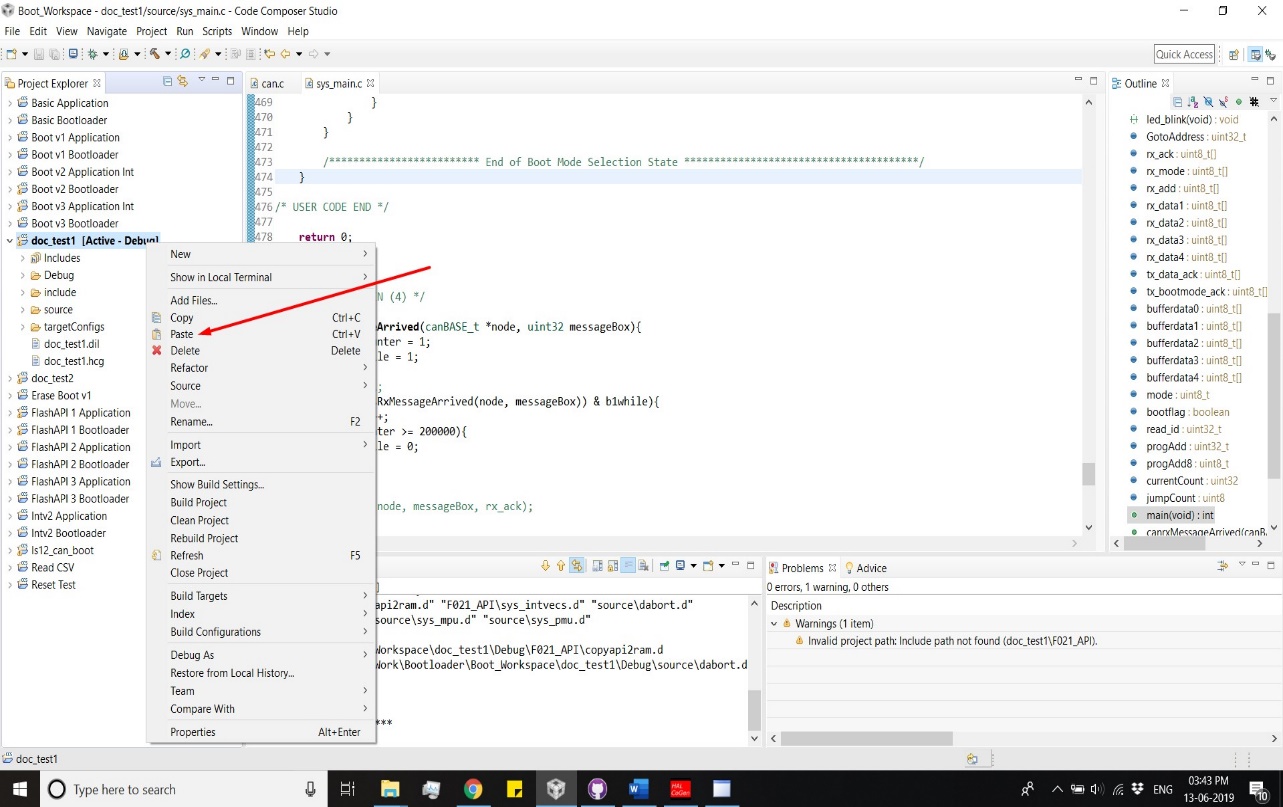
1. Now you can make any required changes in the HalcoGen project. For now, you can import the dil file named “Boot v3 Bootloader.dil” from “32Bit\Bootloader\Default Files” folder. This would add the HalcoGen settings used for “Boot v3 Bootloader” project.



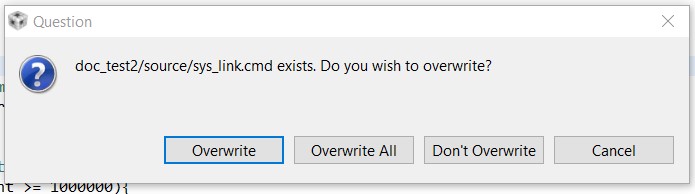
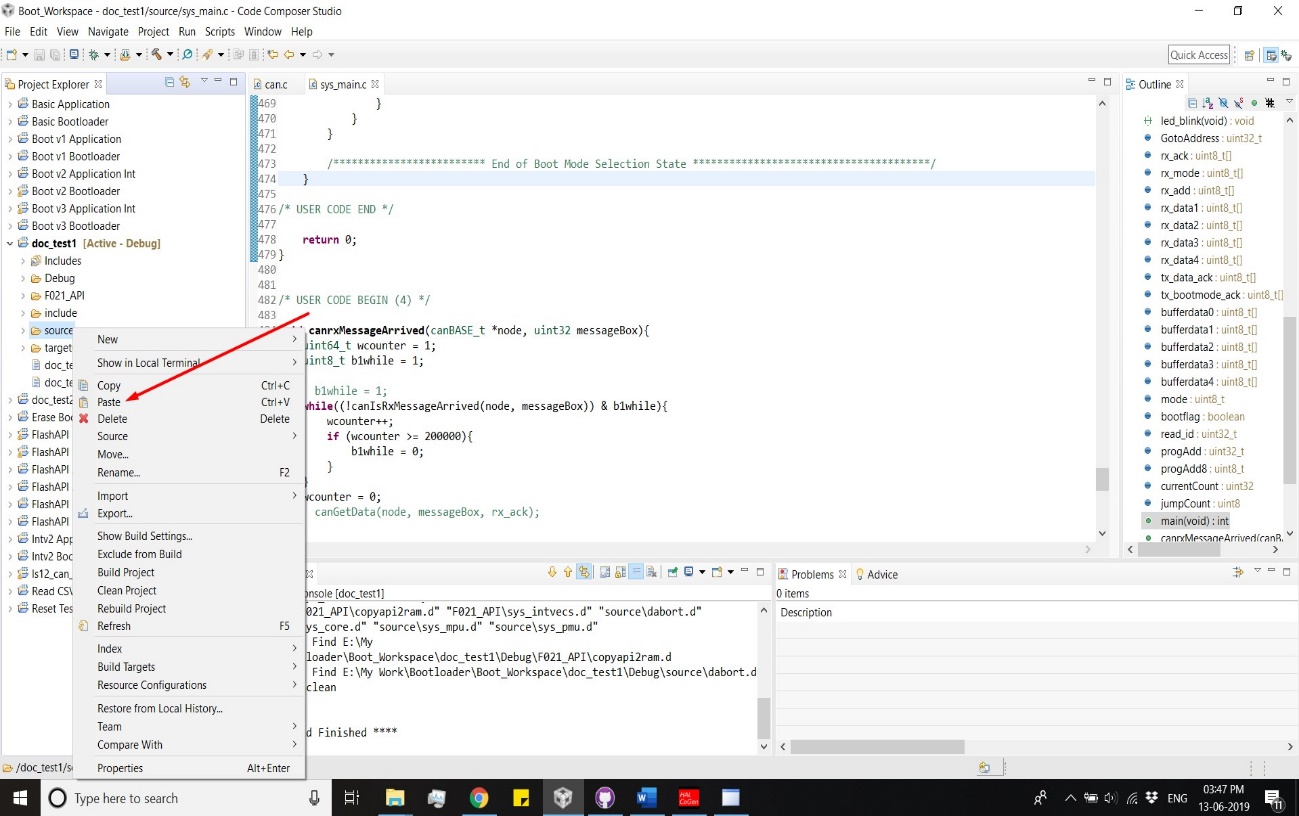
1. Generate code by going to “File-> Generate Code” or Pressing F5.



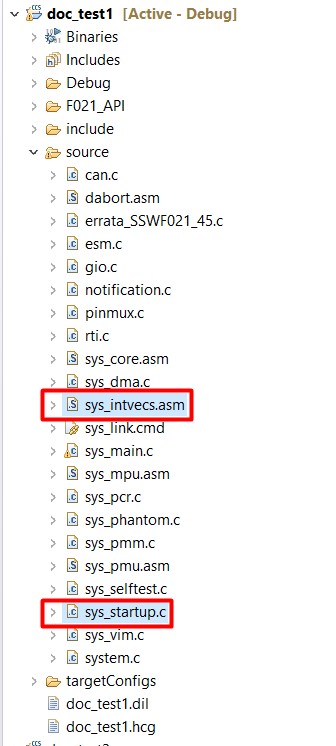
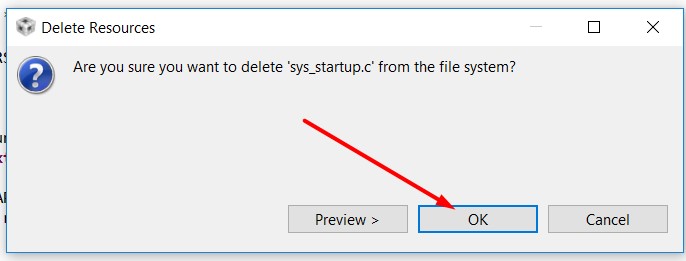
1. After code is generated. We a can shift to CCS.
2. Copy the “F021\_API” folder from “32Bit\Bootloader\Default Files” folder.
3. Now in CCS, move towards “Project Explorer” tab and right click on the current project and Paste the folder inside the current project.



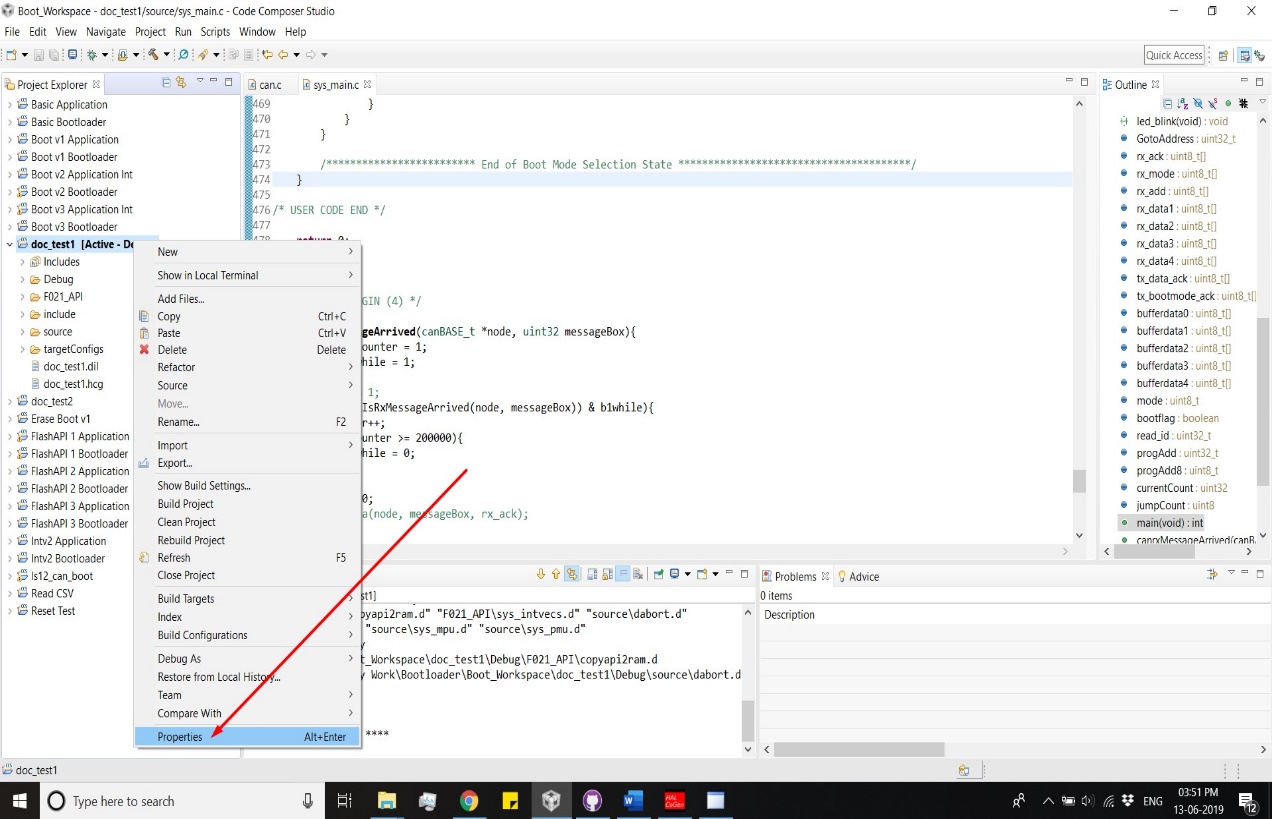
1. After that copy the “sys\_link.cmd” file from “32Bit\Bootloader\Default Files” folder.
2. Now in CCS, move towards “Project Explorer” tab inside the current project “source folder can be found. Right click on the source folder and Paste the file inside the source folder. A Question window will appear then click on Overwrite and the existing “sys\_link.cmd” file will be replaced by new.

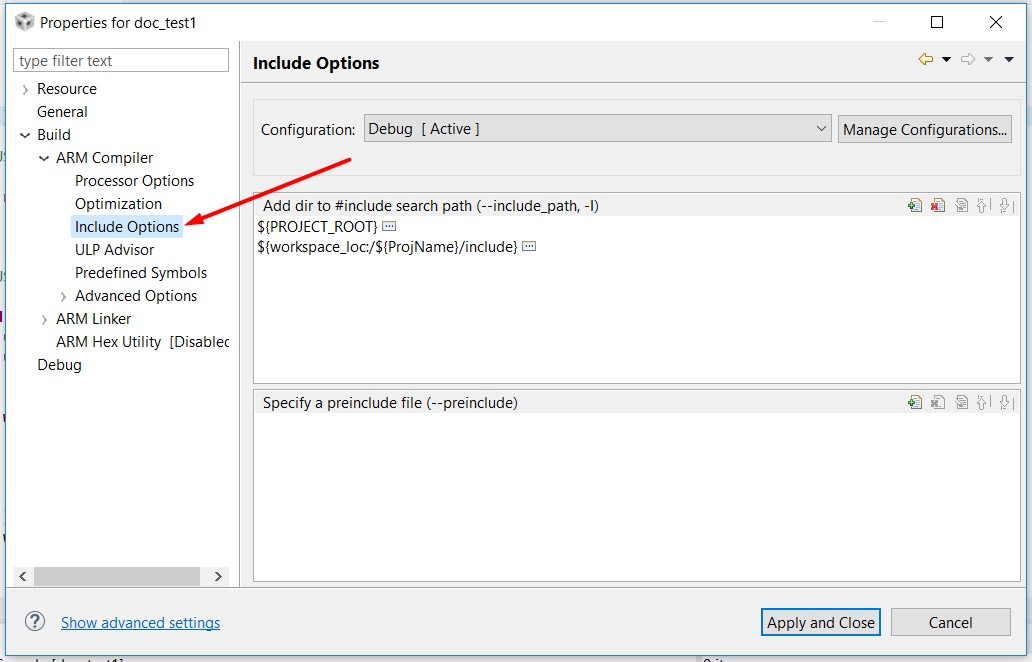
1. Delete two files from the source folder named “sys\_intvecs.asm” & “sys\_startup.c” because these files are included inside the “F021\_API” folder. The delete option can be found by right clicking on the specific file. After selecting the delete option the a “Delete resources” window will appear then click OK button to delete the file. Delete both the files using similar procedure.



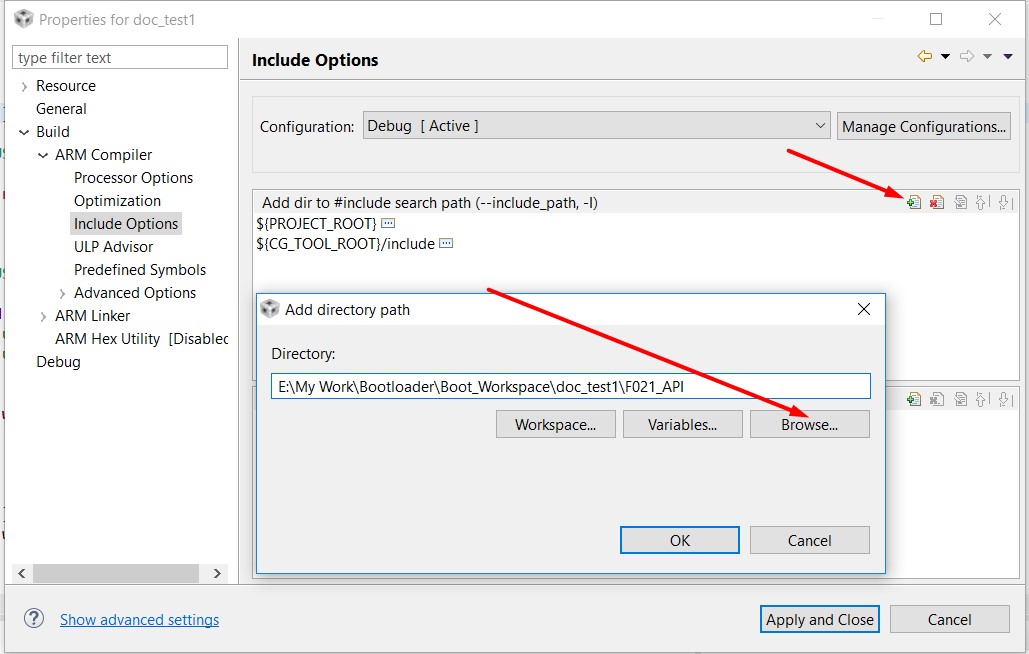
1. After that inside CCS in Project Explore Tab right click on the current project and select properties.



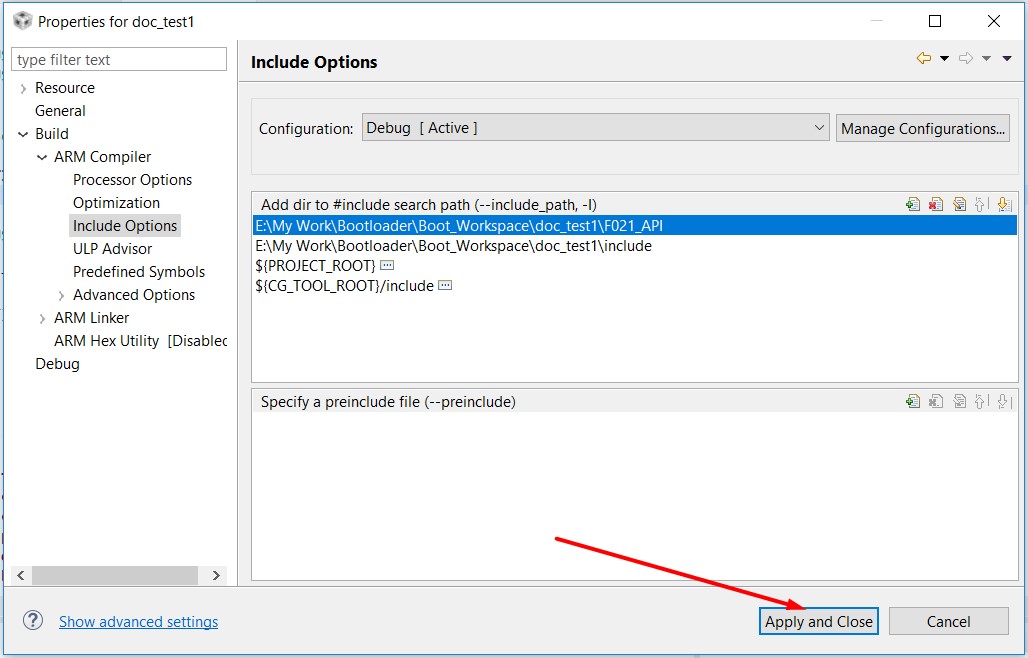
1. A properties window appears. Their inside “Build -> ARM Complier” select “Include options”. A include options configuration window must appear as shown in the below screenshot.



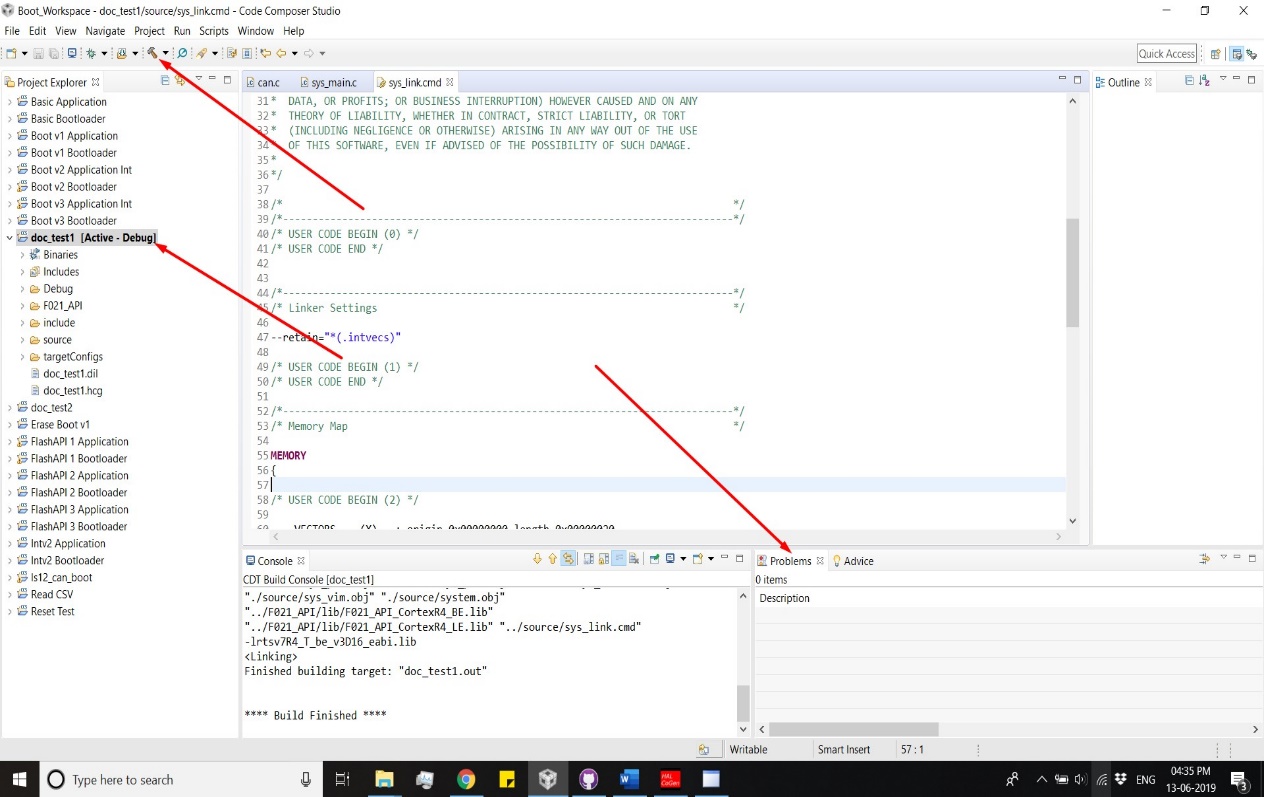
1. Then, click the “Add” button as shown in the below screenshot and after that a “Add directory path” window will appear. Their select “Browse” button and browse towards and select the “F021\_API” folder you pasted inside your current project and click “OK”.



1. Similar to “F021\_API” folder also include the path of “include” folder created by HalcoGen inside your current project folder. Click the “Add” button and after that a “Add directory path” window will appear. Their select “Browse” button and browse towards and select the “include” folder which is inside your current project and click “OK”.
2. After that Click “Apply and Close”.



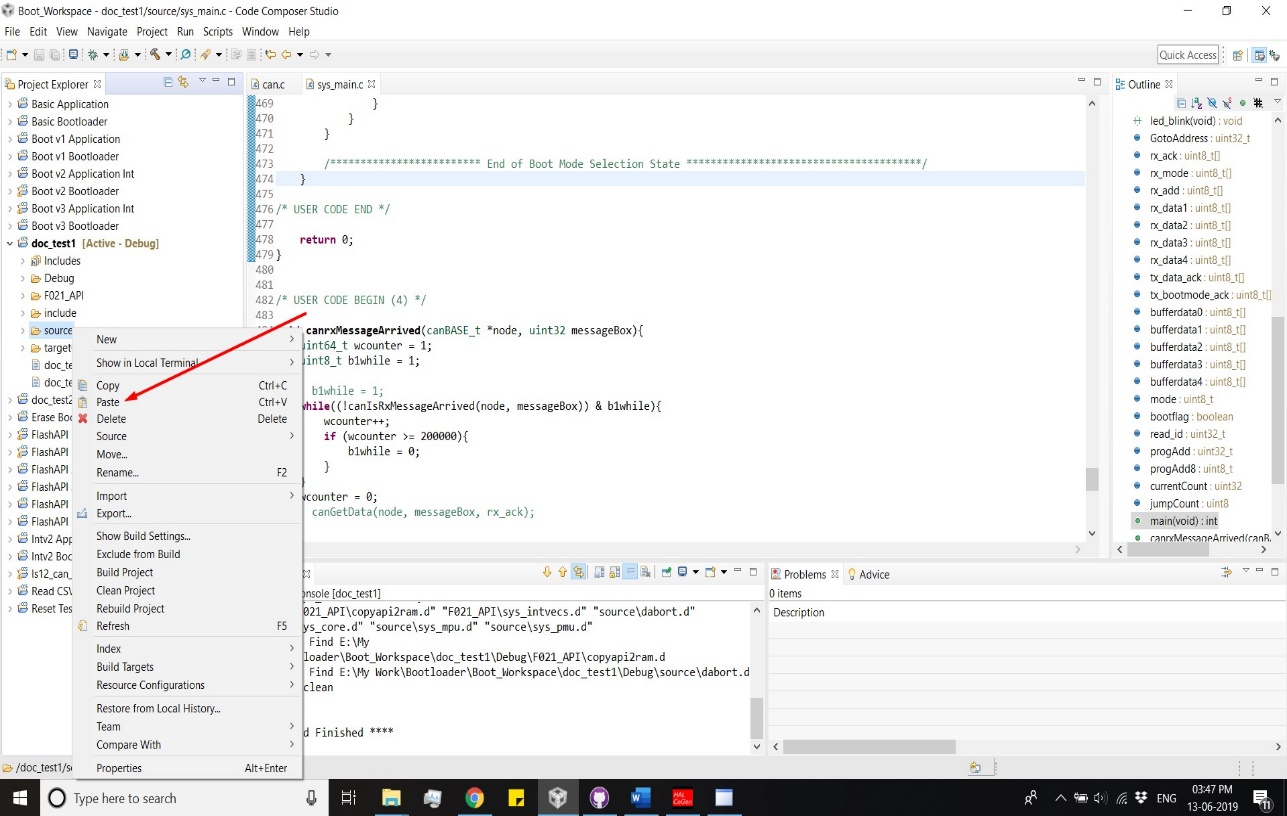
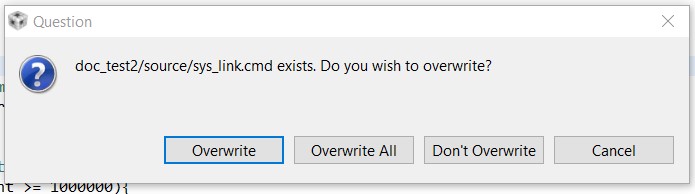
1. Now after selecting the current project you can Compile the project by clicking the hammer icon shown in the below screenshot. After compiling make sure that there are no error showing on the problems tab. There might be some warnings showing but their must not be any error. If there is any error recheck from step 6 till step 15.



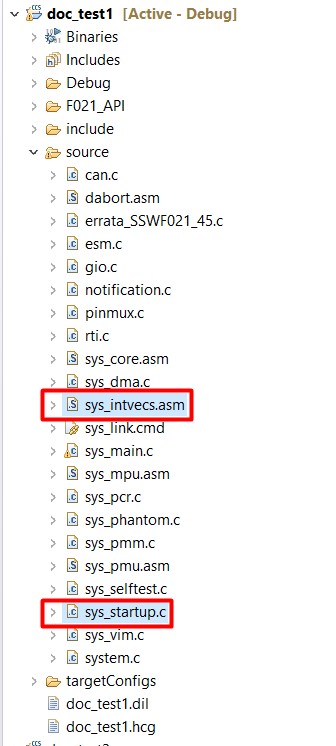
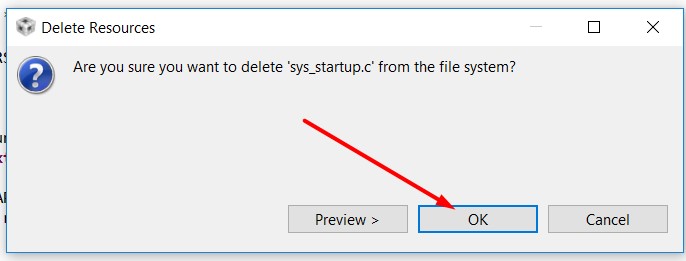
1. If project is compiled properly the you can now use this project for developing your bootloader application.
2. For purely testing purpose you can copy the “sys\_main.c” file from “32Bit\Bootloader\Default Files\Extra” folder and paste it to source folder as explained in step 9. Override the existing file. Then save the project and Compile the project as shown in step 16. Now this project is similar to “Boot v3 Bootloader” project and you can flash this project instead of “Boot v3 Bootloader” as a Bootloader for testing with “32bit-Bootloader-GUI-installer-V0.1” GUI.

# C.] Steps to follow for changed HalcoGen setting in Bootloader Proj:

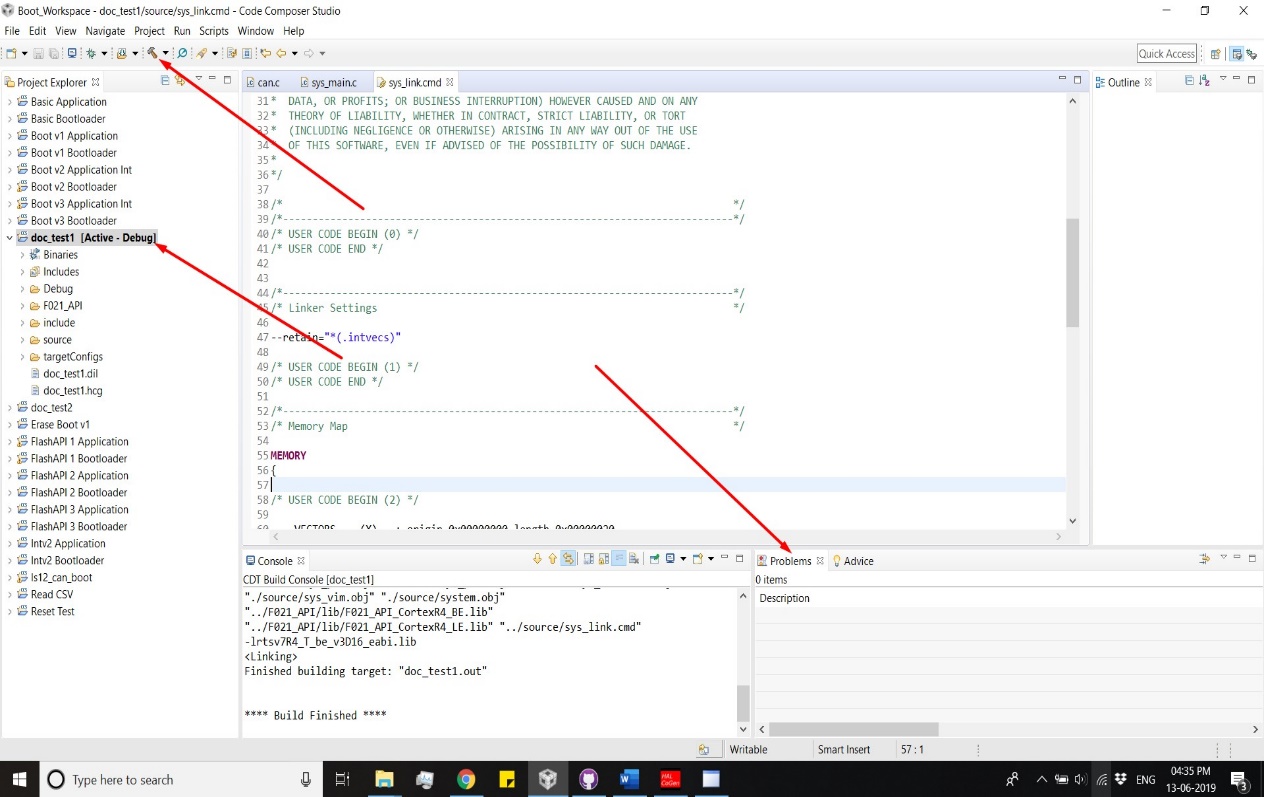
1. Open the same project in HalcoGen by opening the “.hcg” file of the project in HalcoGen.
2. Make the required changes and Generate the code for the project by going to “File-> Generate Code” or Pressing F5.
3. Open CCS, move towards “Project Explorer” tab inside the current project “source folder can be found. Right click on the source folder and Paste the file inside the source folder. A Question window will appear then click on Overwrite and the existing “sys\_link.cmd” file will be replaced by new.



1. After that delete two files from the source folder named “sys\_intvecs.asm” & “sys\_startup.c”. The delete option can be found by right clicking on the specific file. After selecting the delete option the a “Delete resources” window will appear then click OK button to delete the file. Delete both the files using similar procedure.



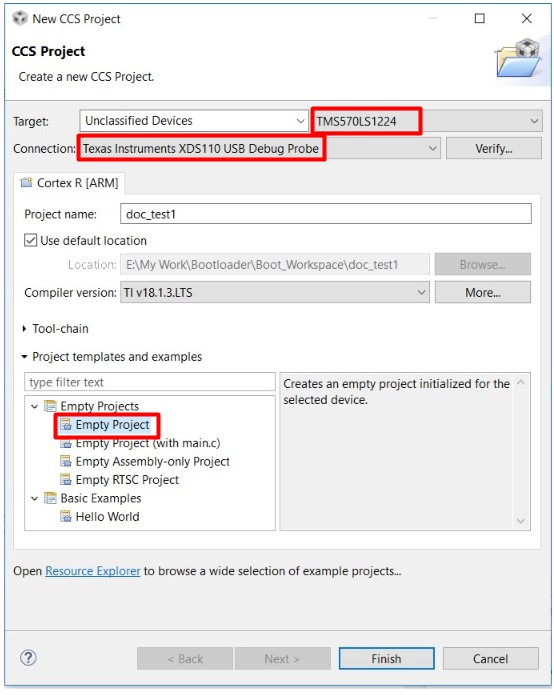
1. Now after selecting the current project you can Compile the project by clicking the hammer icon shown in the below screenshot. After compiling make sure that there are no error showing on the problems tab. There might be some warnings showing but their must not be any error.



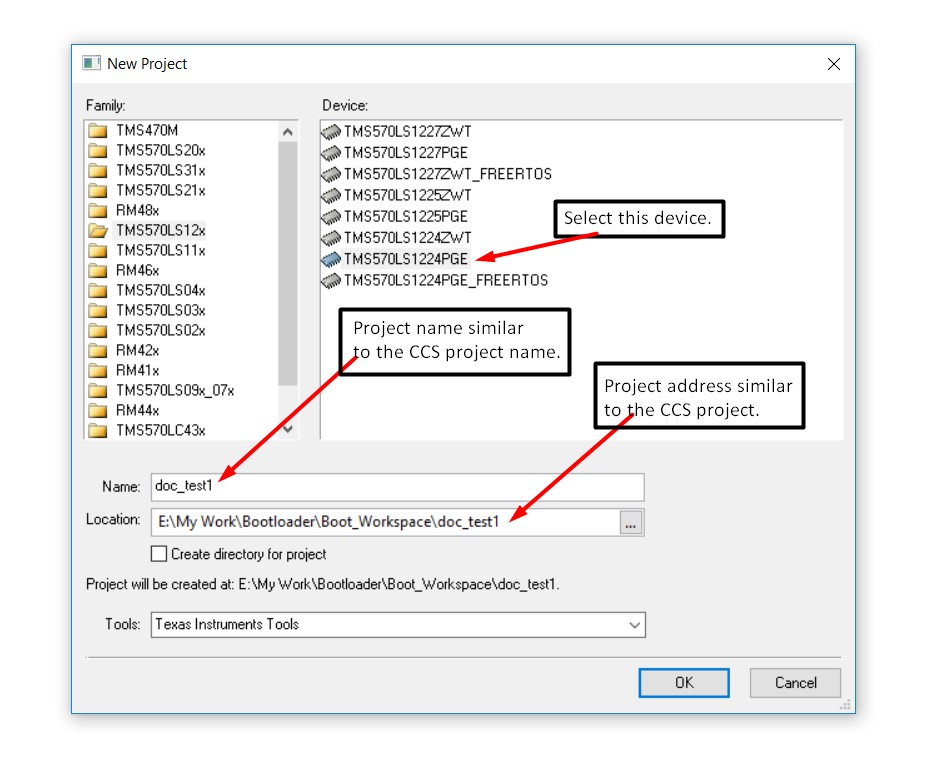
1. If project is compiled properly then you can make further changes in C code inside CCS.

# D.] Steps for setting up the Application project & Generate Hex file:

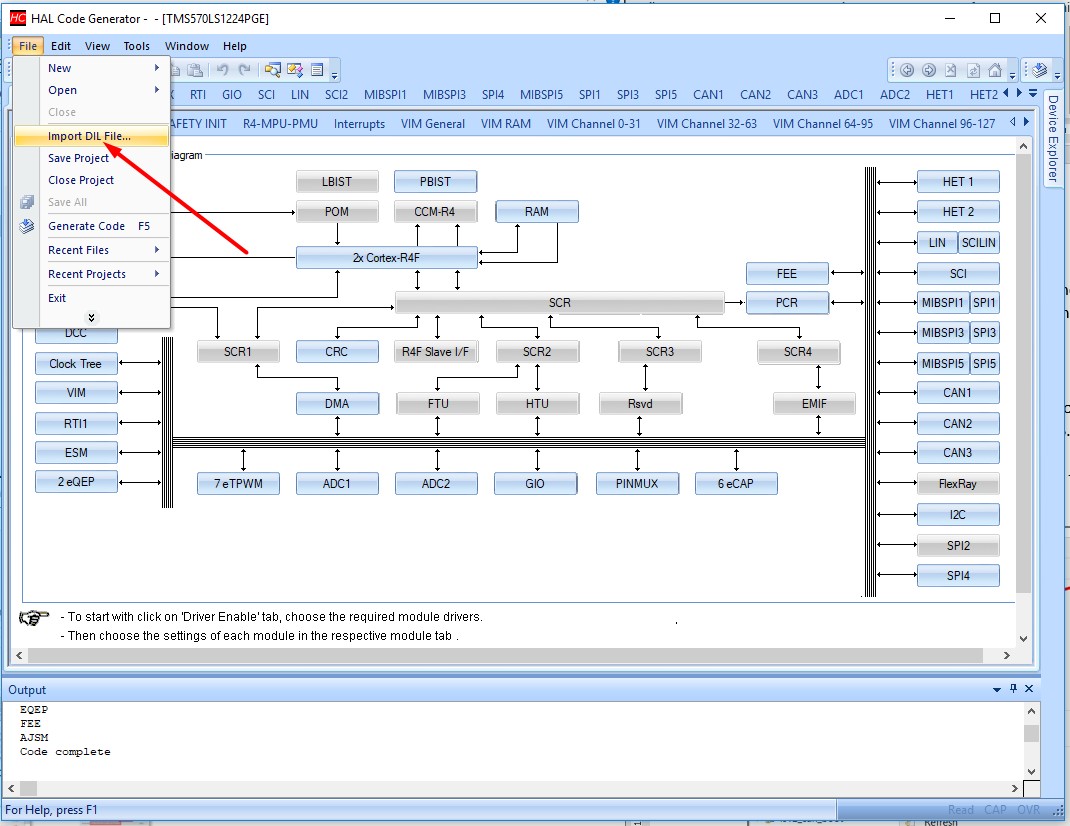
1. Open CCS and create a new project in the CCS. Make sure the red marked options are selected.



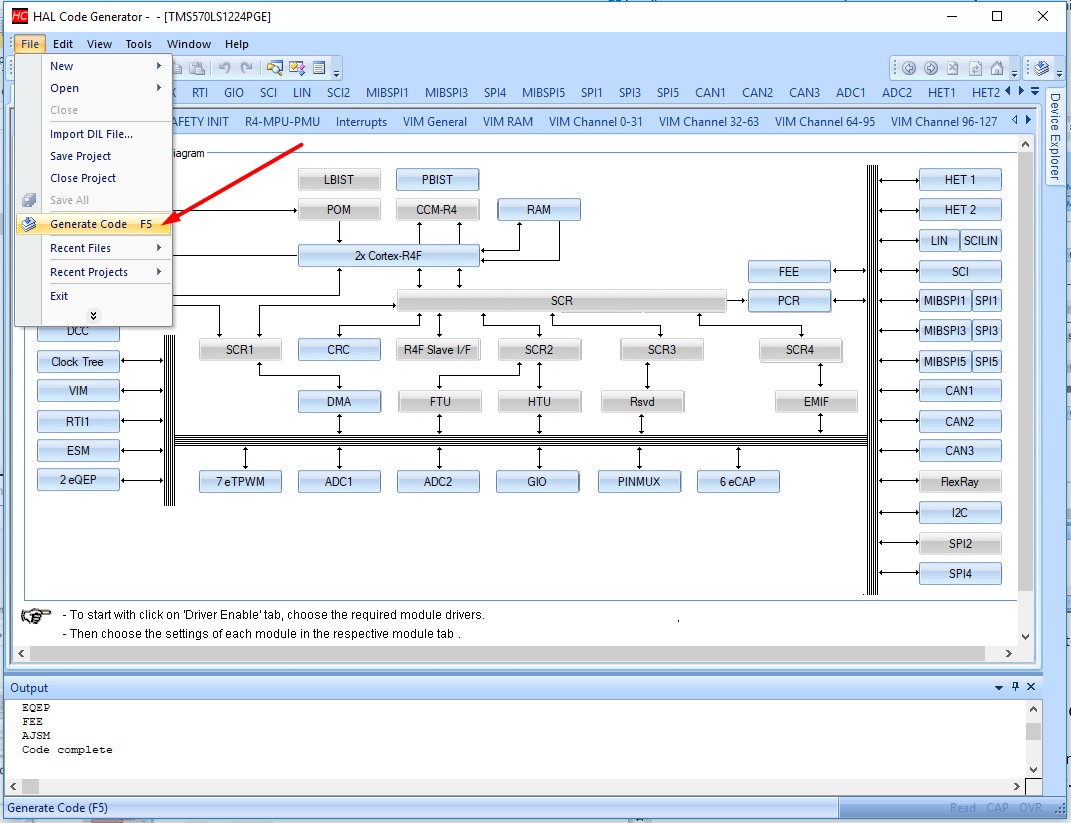
1. Open HalcoGen and create a new project and keep the name of HalcoGen project name similar to that of CCS for simplicity. Make sure the indicated things are completed.



1. Now you can make any required changes in the HalcoGen project. For now, you can import the dil file named “Boot v3 Bootloader.dil” from “32Bit\Bootloader\Default Files” folder. This would add the HalcoGen settings used for “Boot v3 Bootloader” project.



1. Generate code by going to “File-> Generate Code” or Pressing F5.



1. After code is generated. We a can shift to CCS.
2. Open the “sys\_link.cmd” file which is inside the source folder of current application project and replace the following lines

VECTORS (X): origin=0x00000000 length=0x00000020

FLASH0 (RX): origin=0x00000020 length=0x0013FEE0

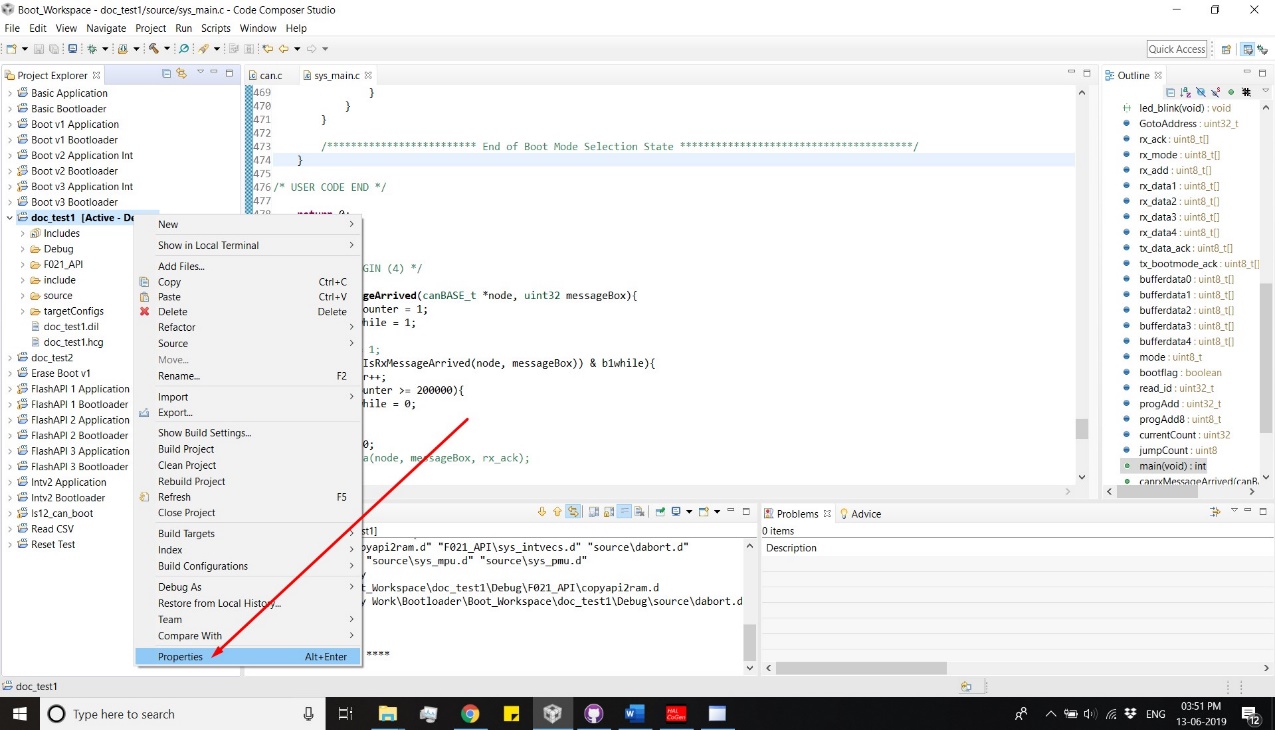
to

VECTORS (X): origin=0x00014000 length=0x00000020

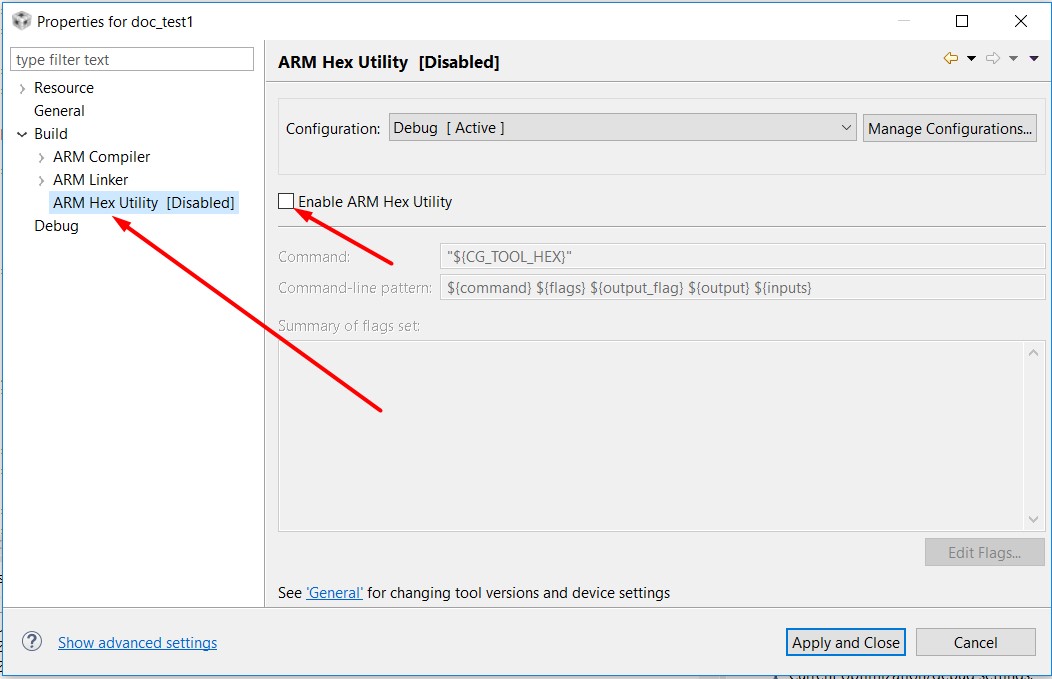
FLASH0 (RX): origin=0x00014020 length=0x0014FEE0

Here, VECTORS 0x14000 is start address & FLASH0 0x14020 is start address + 0x20. These addresses can be changed according to change in start address.

1. After that inside Project Explore Tab right click on the current project and select properties.



1. A properties window appears. Their inside “Build” select “ARM Hex Utility”. A ARM Hex Utility window will appear as shown in the below screenshot. There check the “Enable ARM Hex Utility” Box.

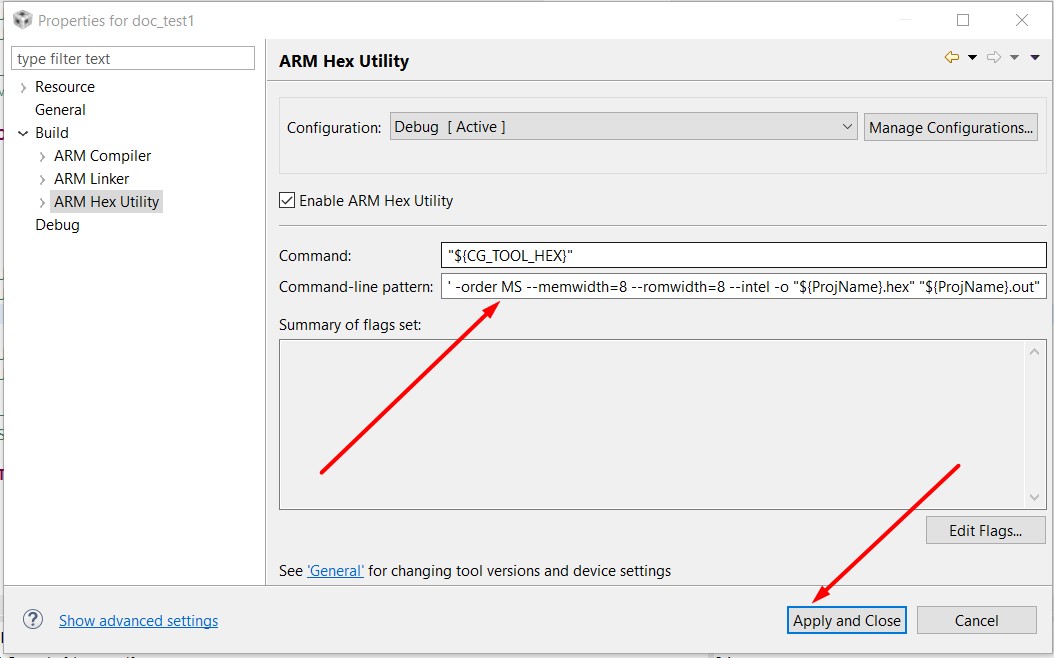


1. After that, change the text inside “Command-line pattern” box from

${command} ${flags} ${output\_flag} ${output} ${inputs}

with

"${CG\_TOOL\_HEX}" -order MS --memwidth=8 --romwidth=8 --intel -o "${ProjName}.hex" "${ProjName}.out"

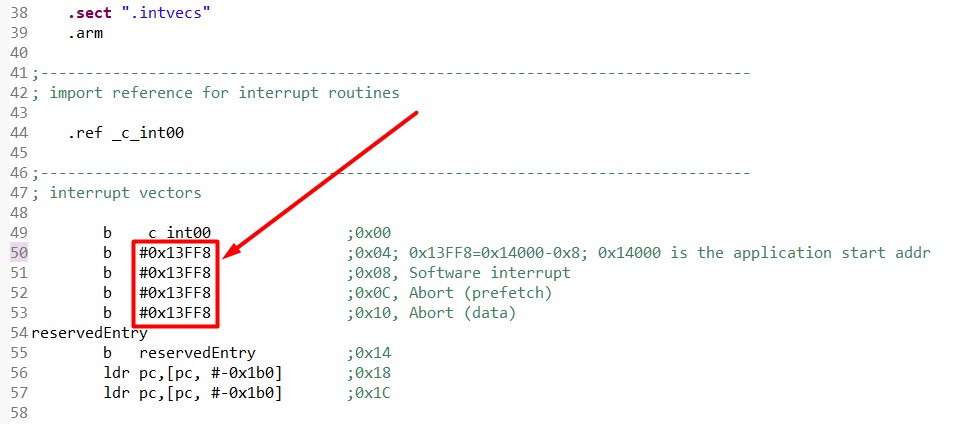


After that click “Apply and Close”.

1. Now after Compiling the project a hex file will be generate inside “Debug” folder of the current project. The name of the file will be “current project name”.hex
2. After completing all the steps, programming the application code can be carried out. After completing the application code build the project and use the hex file generated in “Debug” folder for flashing the MCU using GUI.

# E.] Steps to change the Jump address of Application:

1. Open both, the Bootloader and Application project inside the CCS environment.
2. From Bootloader project open the “sys\_intvecs.asm” file inside the “F021\_API” folder.

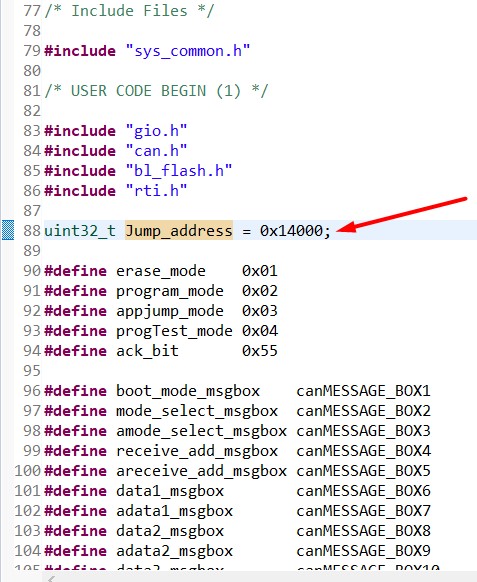


All the four numbers inside the box are depending upon the starting address of application.

Here, Number = “Starting address” – 0x08.

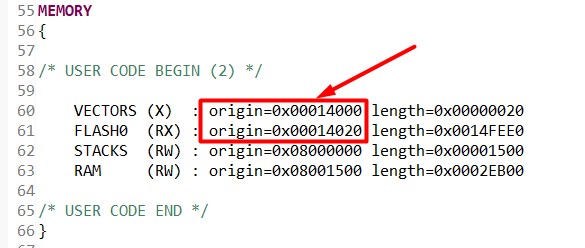
Replace all the four numbers according to the above formula.

1. From Bootloader project open the “sys\_main.c” file inside the “source” folder.



Change the Jump\_Address variable value to the new address.

1. From Application project open the “sys\_link.cmd” file which is inside the source folder of current application project and replace the following two address inside the box.



Here, VECTORS (X): 0x14000 is start address & FLASH0 (RX): 0x14020 is start address + 0x20. These addresses can be changed according to change in start address.

Where, VECTORS (X): “Start address” & FLASH0 (RX): “Start address” + 0x20.

1. Save all the changes. Making above changes will change your application start address.

# F.] Important Notes:

1. The base example referred was “CAN Bus Bootloader for TMS570LS12x MCU” the Texas document is named as [SPNA186](http://www.ti.com/lit/an/spna186/spna186.pdf). “F021\_API” files were taken from [this](http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=spna186&fileType=zip) (<http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=spna186&fileType=zip>) example project.
2. The only change made to the library was in “bl\_flash.c” file inside “**Fapi\_BlockProgram**” function line number 191 and the function call was changed from

Fapi\_issueProgrammingCommand((uint32\_t \*)dst, (uint8\_t \*)src, (uint32\_t) bytes, 0, 0, Fapi\_AutoEccGeneration);

to

Fapi\_issueProgrammingCommand((uint32\_t \*)dst, (uint8\_t \*)src, (uint32\_t) bytes, 0, 0, Fapi\_DataOnly);

As the same address was not get programmed twice. We required this feature because for blank space we are programming 0xFF initially and once the data is received for the same address, we are programming the data.

# G.] 32 Bit Bootloader and GUI Communication State diagram:

