

EyeQ CDD Integration

1. Scope	2
2. Introduction to EYEQ, EYEQ CDD and Application Interaction.....	2
3. Understanding ME Software Release	6
4. Converting ME binary file .bin to .S37 files.....	10
5. Understanding the Core Software(EyeQ CDD) Release	12
6. Understanding the Core Software(EyeQ CDD) Folder Structure	12
7. Core Software Integration Steps and Procedure	14
8. ARXML Generation according to Project Architecture	15
9. Core Software Integration Steps in detail Static files:	20
9.1 CDD Component Integration (Source code)	20
9.2 App Config Integration.....	21
9.3 IoHwAb Integration	22
9.4 Startup code integration	22
9.5 AoUSafe Integration:	23
9.6 MCAL Static code Integration	24
9.7 Other Static code Integration	24
10. ARXML import and BSW updates	25
10.1 Importing Arxmls to the Davinci Developer	25
10.2 NvM Integration.....	32
10.3 DEM Integration	40
10.4 Integration of Proxy modules	47
10.5 Os Configuration.....	49
10.6 CDD Component Integration(Port Connections)	53
10.7 Source Code Generation and Compilation	54
11. MCAL Integration XDM changes and Dynamic code:	55
11.1 XDM integration and Dynamic code generation:.....	56
11.2 Things to be taken care while check-in MCAL changes :	57
12. NVM default values updates:.....	58
13. SConstruct and linker changes	60
13.1 SConstruct integration:	60
13.2 Linker changes:	60
14.General Issues, debugging steps and validating EyeQ integration.....	61

EyeQ CDD Integration

1. Scope

The scope of this document is to learn procedure and steps to be followed during integration of the EyeQ CDD with Application software. This doc will also give an InSite of regular issues faced and care to be taken during integration along with some debug steps.

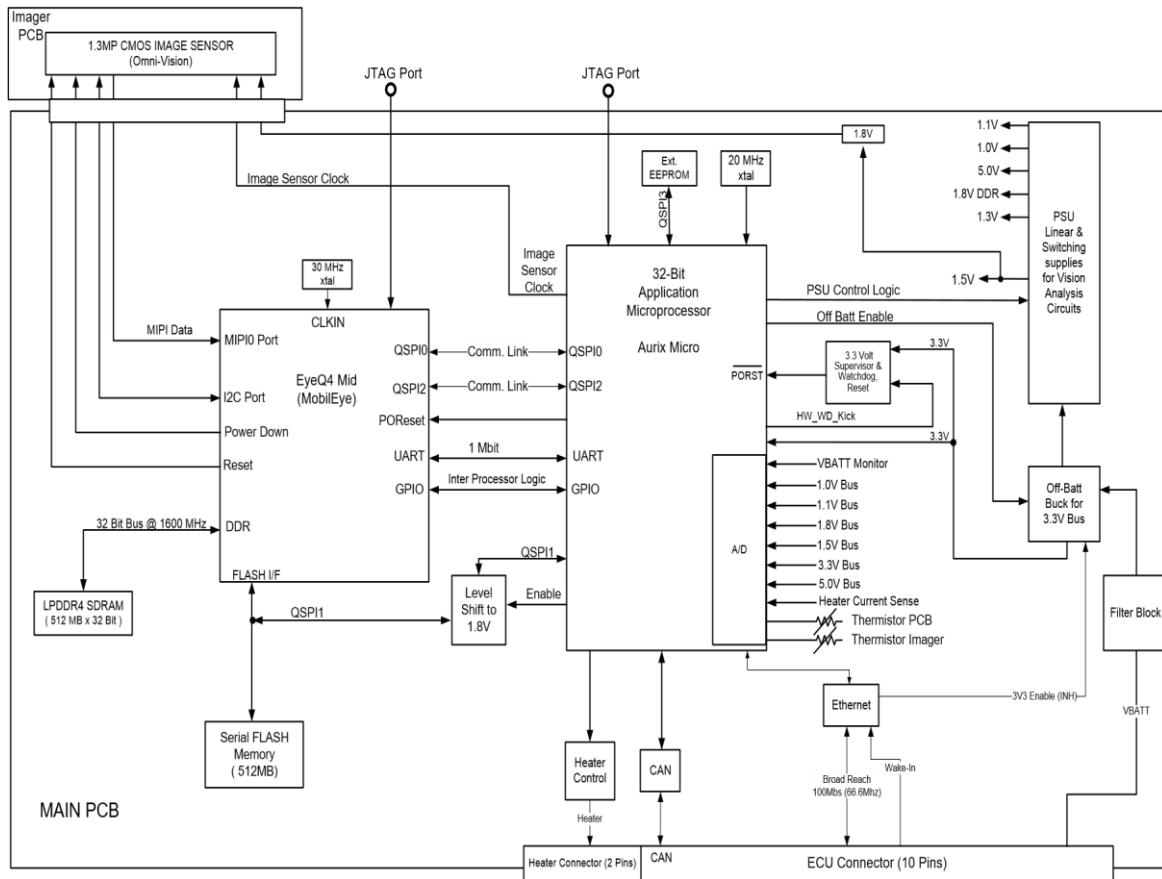
NOTE: This document gives all steps in integration process, some of steps can be ignored if there are no changes in particular step in core release.

2. Introduction to EYEQ, EYEQ CDD and Application Interaction

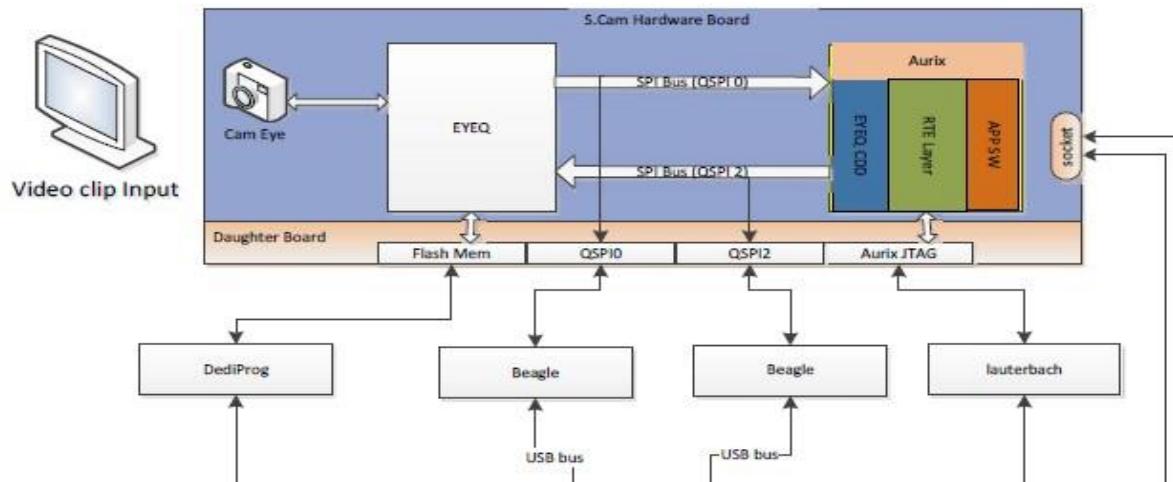
Front Camera Module consist of EYEQ on EyeQ chip (external processor or ASIC),with EYEQCDD application and application algo components on AURIX .EYEQ is generally delivered by MobilEye(ME) hence EyeQ binaries are called ME files.

ME consist of core image acquisition along with processing according to features supported by particular ME. ME interacts with EyeQ CDD using QSPI half-duplex communication on two different SPI channels usually (QSPIO and QSPI2) as shown in below diagrams.

EyeQ CDD Integration



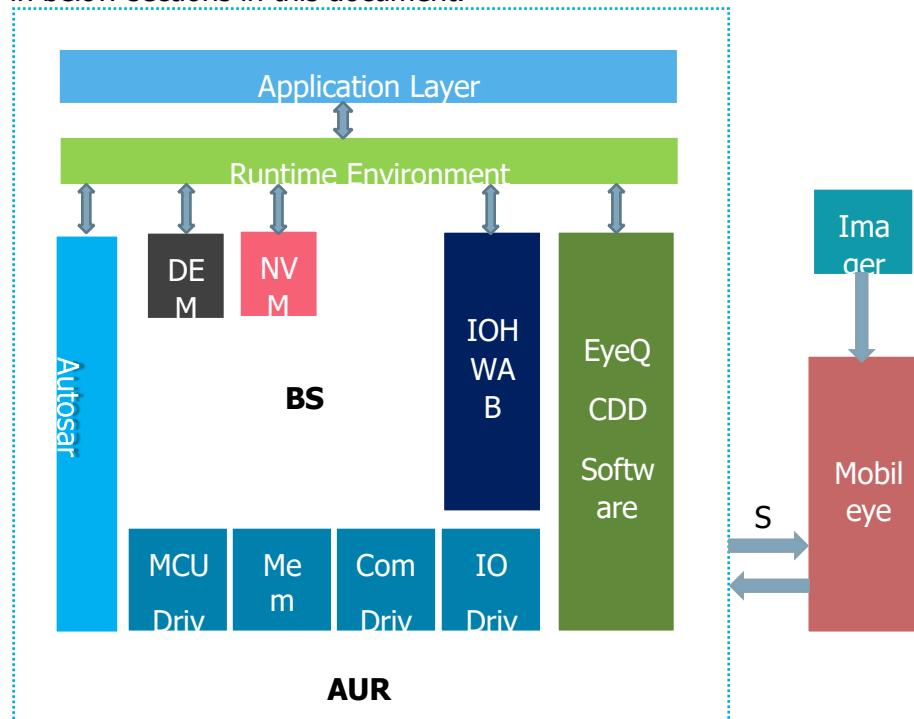
Detail Front Camera module



Front camera module block diagram with various connection

EyeQ CDD Integration

The below diagram shows the EyeQ CDD interactions with Application,BSW and the interaction between ME and Aurix. ME and Aurix are connected through SPI communication as discussed earlier to exchange the information. EyeQ CDD software will run on Aurix and handles the interactions between Aurix and ME ,it controls the power and state management of ME according to application needs. EyeQ CDD software architecture is designed based on OSI layered architecture. The power management of ME is handled through IOHWAB. DEM and NVM are configured for EyeQ CDD specific faults and blocks. EyeQ CDD integration is done with the interface connections between Application software and EyeQ CDD, NvM and Dem interface connections and mapping the schedulable entities in EyeQ CDD component to Autosar OS. The EyeQ CDD Integration process is explained in detail in below sections in this document.



EyeQ CDD Integration

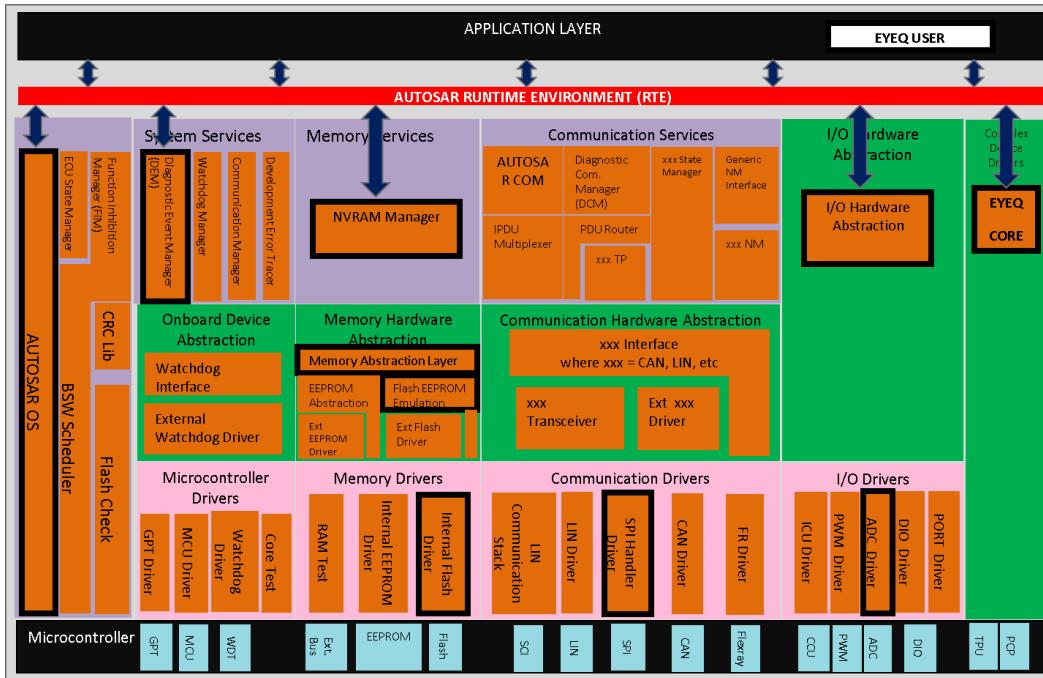


Figure 1 DAS_SCAM_4.8 Platform Architecture

The above figure shows the DAS_SCAM_4.8 platform architecture. It follows AUTOSAR_4.0.3 classic platform. EyeQ CDD component interacts with application software, IoHwAb, NvM and DEM modules. It uses the services provided by NVM, DEM and IOHWAB. It uses MCAL services to interact with ME chip. Os will trigger the periodic runnables in EyeQ CDD component and manages the execution of ISR's configured for EyeQ CDD component.

The EyeQ CDD core software contains the configuration of NVM, DEM and IOHWAB specific to EyeQ CDD module. NVM configuration consists of NVM Blocks configured for EyeQ CDD. DEM configuration contains DEM events configured for EyeQ CDD faults.

The EyeQ CDD core software integration with application software carries integration of EyeQ CDD source code, EyeQ CDD Configuration, NVM configuration specific to EyeQ CDD, DEM configuration specific to EyeQ CDD, IOHWAB, MCAL, AOUSAFA ,OS changes from core ,linker and Startup code.

EyeQ CDD Integration

3. Understanding ME Software Release

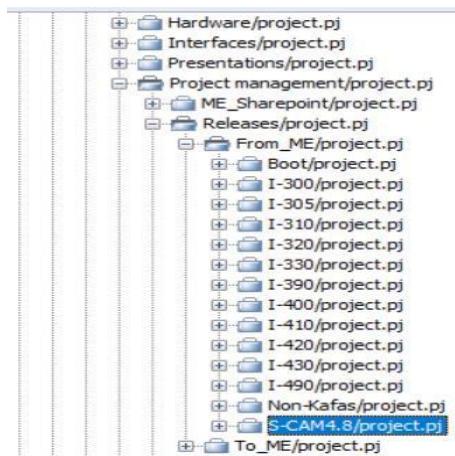
ME Software release contains binary file which will be programmed into EyeQ chip and a Release notes along with SPI protocol sheet(excel) for the current release.

Usually ME software release will be done by MobilEye when there is an update in ME software or changes suggested by the user or if there is a bug reported by the user in ME software.

ME releases can be find in following integrity path:

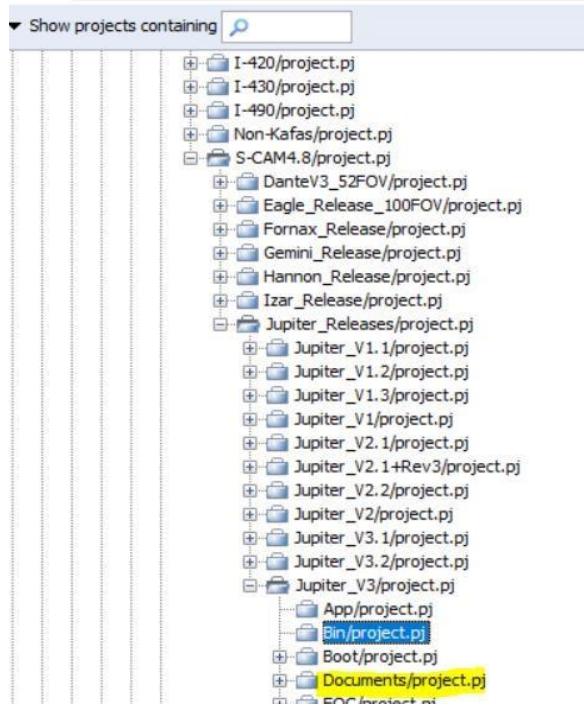
<url:integrity://skobde-mks.kobde.trw.com:7001/si/viewproject?project=/DAS%2dME/020%5fCore/EyeQ4/Project%20management Releases/From%5fME/project.pj>

By clicking the above integrity link the integrity view will be as shown in below figure:



Select the variant according to project and project specific for example norma xx.

EyeQ CDD Integration



Folder structure of ME release as below

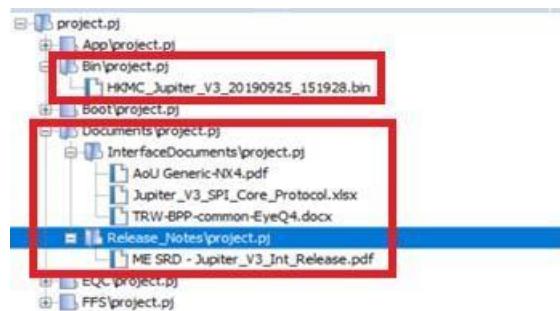
Name	Date modified	Type
App	10-03-2021 10:54	File folder
Bin	10-03-2021 11:05	File folder
Boot	10-03-2021 11:05	File folder
dbc	10-03-2021 11:06	File folder
Documents	10-03-2021 11:16	File folder
EQC	10-03-2021 11:37	File folder
FFS	10-03-2021 11:37	File folder

Inside variant folder ,bin folder contains the binary file in .bin format to be flashed to

EyeQ chip. Generally .bin files are flashed using Dediprog tool if port is available. In GWM and G35 project variants these .bin files are converted into .s37 which will be explained in document and flashed using MFT tool .

Documents folder contains TRW BPP document, SPI protocol sheet (Excel) and a release notes for the current release. Sandbox view shown below.

EyeQ CDD Integration



Release information is present in \Documents\Release Notes

Release Notes

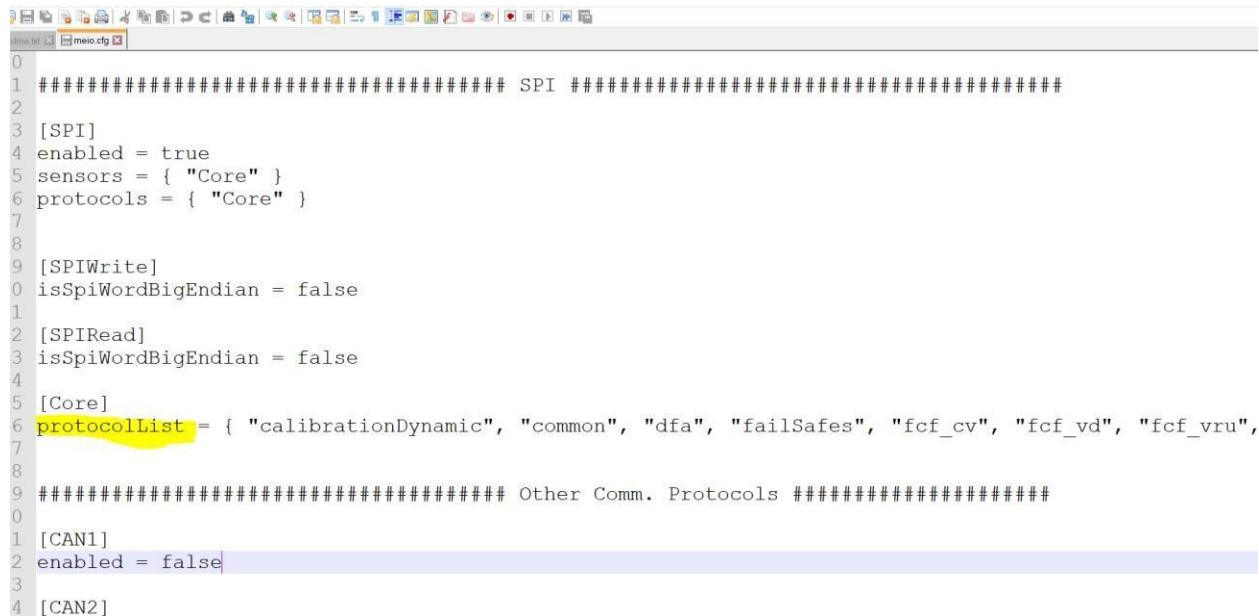
Name	Date modified	Type	Size
 Mars_V2.2_SRD	06-10-2020 02:23	Adobe Acrobat D...	93 KB

This document gives version of Boot, MEST and SPI protocol version as below

EyeQ CDD Integration

Release Catalog Entry	
Category	Value
Base Date Released:	20-090-03
Base Release Name:	Mars_V2.1
Base Major Release:	MR_20_5
Base MEST (application):	20.5.2.1
Base Boot:	8.7.1
Base SPI Protocol:	20.5.11.8
Base CF Notation:	0
Base FFS Version	0.2.0.1

Other important file in the ME release software is meio which is present in \FFS\etc folder .This file contains important information about features supported by particular ME and other details as below .Protocol list gives all the feature supported and released in particular ME .



```
0 ##### SPI #####
1 [SPI]
2 enabled = true
3 sensors = { "Core" }
4 protocols = { "Core" }
5
6 [SPIWrite]
7 isSpiWordBigEndian = false
8
9 [SPIRead]
10 isSpiWordBigEndian = false
11
12 [Core]
13 protocolList = { "calibrationDynamic", "common", "dfa", "failSafes", "fcf_cv", "fcf_vd", "fcf_vru",
14
15 ##### Other Comm. Protocols #####
16
17 [CAN1]
18 enabled = false
19
20 [CAN2]
```

EyeQ CDD Integration

4. Converting ME binary file .bin to .S37 files .

In GWM and other variants of project .S37 of ME are flashed using MFT tool.
Single binary file of ME is converted into three .s37 files BOOT,FFS and MEST files

There are tools (batch files)files present in tools folder of GWM application project in path Application\Tools\EyeQ_S37_Gen as below highlighted.

_ETC > Application > Tools > EyeQ_S37_Gen				
Name	Date modified	Type	Size	
Output	09-04-2021 17:28	File folder		
64MB_EyeQ	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeBOOT	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeFFS	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeMEST	09-04-2021 17:28	Windows Batch File	1 KB	
srec_cat	09-04-2021 17:28	Application	2,931 KB	

Place the binary file of ME which need to be converted into S37 files in below folder as shown

_ETC > Application > Tools > EyeQ_S37_Gen				
Name	Date modified	Type	Size	
Output	09-04-2021 17:28	File folder		
64MB_EyeQ	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeBOOT	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeFFS	09-04-2021 17:28	Windows Batch File	1 KB	
64MB_NisMakeFlashFileMobileyeMEST	09-04-2021 17:28	Windows Batch File	1 KB	
FD.Norma2-V5.1.GWM.bin	11-05-2021 11:36	BIN File	65,536 KB	
srec_cat	09-04-2021 17:28	Application	2,931 KB	

Drag and Drop the .bin on Batch files for example on
64MB_NisMakeFlashFileMobileyeBOOT.A command window will pop out as
below enter Y to generate Boot s37 file

```
C:\WINDOWS\system32\cmd.exe
ould Not Find D:\GWM_SCAM4.8_APPL_P05_ETC\Application\Tools\EyeQ_S37_Gen\BOOT.bin
ould Not Find D:\GWM_SCAM4.8_APPL_P05_ETC\Application\Tools\EyeQ_S37_Gen\data.s19
:\GWM_SCAM4.8_APPL_P05_ETC\Application\Tools\EyeQ_S37_Gen\Output\*, Are you sure (Y/N)? Y
```

Boot files is generated in Output folder as below

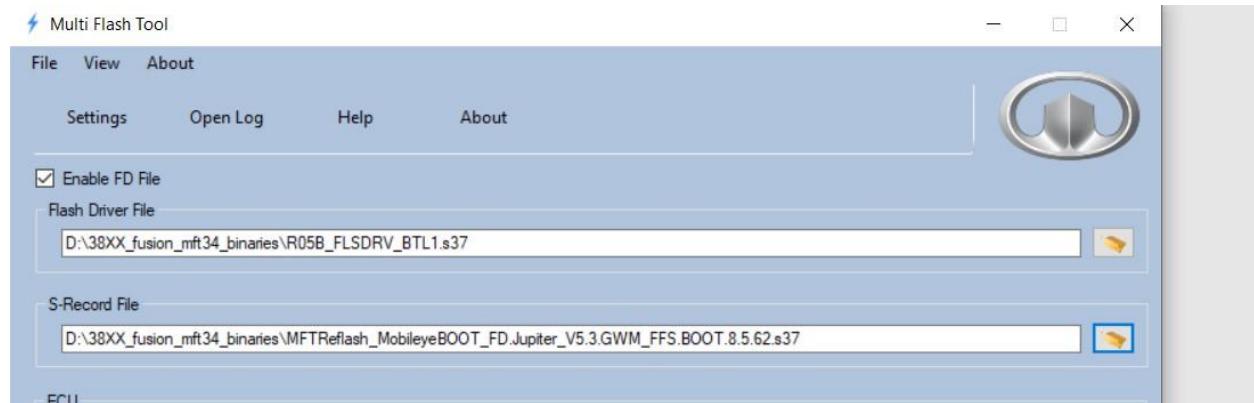
EyeQ CDD Integration

ETC > Application > Tools > EyeQ_S37_Gen > Output			
Name	Date modified	Type	Size
MFTReflash_MobileyeBOOT_.s37	20-06-2021 19:46	File	3,840 KB

Rename file with .S37 extension and move to some other location . Similarly drag and drop FFS and MEST files on respective batch files , rename with S37 extension and move to separated folder as below.

④ MFTReflash_MobileyeBOOT_norma	11-05-2021 11:58	S37 File	3,840 KB
④ MFTReflash_MobileyeFFS_NORMA_GWM	11-05-2021 11:59	S37 File	1,280 KB
④ MFTReflash_MobileyeMEST_NORMA	11-05-2021 12:00	S37 File	1,58,720 KB

We can flash s37 files one after other using MFT tool by selecting each at once as below one after other.



EyeQ CDD Integration

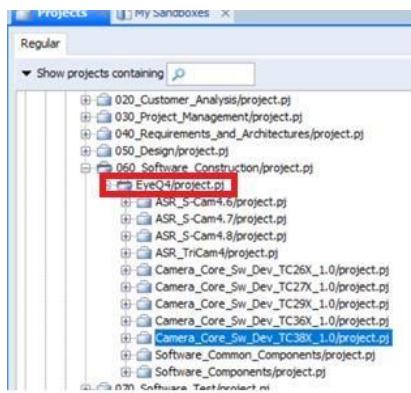
5. Understanding the Core Software(EyeQ CDD) Release

Core Software will be released by Core team. The Core Software release contains configurations and source code of EyeQ CDD, MCAL, IOHWAB and contains integration code that is useful for the platform integration.

The below link will navigate to core software releases in integrity:

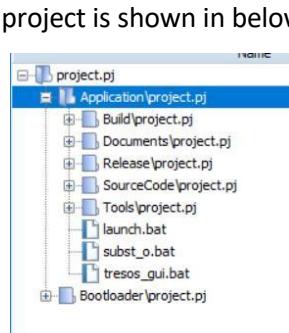
<url:integrity://skobde-mks.kobde.trw.com:7001/si/viewproject?project=/DAS/040%5fTest%5fProjects/%3cCore%5fFC%3e%20Core%20Forward%20Camera%20Development/060%5fSoftware%5fConstruction/EyeQ4/projetc.t.pj>

By clicking the above link the integrity view is shown in below picture:



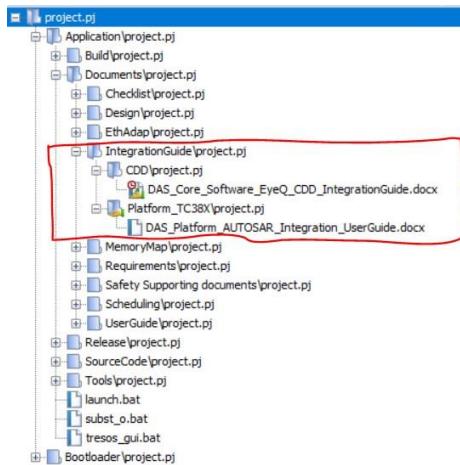
6. Understanding the Core Software(EyeQ CDD) Folder Structure

Sandbox view of Core software project is shown in below figure..

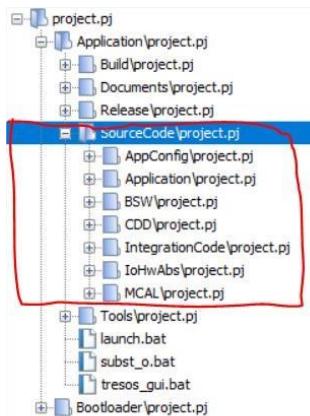


In **Documents** folder **Integration Guide** folder contains CDD integration guide and Platform integration guide.

EyeQ CDD Integration



The SourceCode folder view shown in below figure:



- AppConfig --> This folder contains EyeQ CDD, IOHWAB and PTM component configuration files(.arxml and cfg.h) which can be modified by application team.
- CDD --> This folder contains source code for EyeQ CDD Component, Production mode services, MCU_JTAG_Lock, PTM and code generators for DEM, NVM Proxy modules.
- IntegrationCode --> This folder contains source code for SafetyLib(AoUSafe), Startup code(AURIX).
- IoHwAbs --> This folder contains the IoHwAbs source code specific to EyeQ CDD component.
- MCAL --> This folder contains third party MCAL driver source code, generated configuration files(.arxml, cfg.c, cfg.h).

EyeQ CDD Integration

7. Core Software Integration Steps and Procedure

Below are steps involved in EyeQ CDD Core software integration. Details of each steps will be explained where ever it is required in documents in upcoming chapters.

Step1:

- Integrate the EyeQ_CDD ,IoHwAb, AOUSAFe files and other file from core release.
- Integrate the MCAL Changes.
- Identify & Configure the NVM blocks if any changes.
- Identify & Configure the Dem Events if any changes.
- Identify & Configure the OS if any changes.
- Identify linker changes, make files changes if any and integrated to application linker and Sconstruct file.

Step2:

- Convert ARXMLS according to project architecute and Import EyeQ_CDD ,IoHwAb Proxy(wrapper) ,PTM related Arxml in to Developer.
- Map the EyeQ Interface related changes in Developer SWC's.
- With above changes build the SW.
- If Build is successful Test and verify the Basic Communication & EyeQ Vision Mode.

EyeQ CDD Integration

8. ARXML Generation according to Project Architecture.

ARXML delivered by core cannot be directly imported into developer as the core architecture and Application architecture may not be same. Components like NVM, DEM, EYEQCDD, IoHwAb and PTM may not be in same core in core and application. Integrator need to configure certain files and follow below steps to generate there architecture specific ARXMLS.

To install tools related to ARXML we need to follow installation guide in following path of core delivered software.

norma_1.24 > Tools > ARXML_Generators				
	Name	Date modified	Type	Size
	📁 _BSWs	09-02-2021 11:32	File folder	
	📁 _Dependencies	09-02-2021 11:33	File folder	
	📁 _Generators_ARXML	09-02-2021 11:33	File folder	
	📁 _Generators_ECUC	09-02-2021 11:33	File folder	
	📁 _Generators_SBS	09-02-2021 11:33	File folder	
	📁 _Generators_SourceCode	09-02-2021 11:33	File folder	
	📁 _Parsers	09-02-2021 11:34	File folder	
	📁 _Release	09-02-2021 11:34	File folder	
	📁 _Template	09-02-2021 11:34	File folder	
	📄 InstallationGuide	11-04-2019 19:43	Microsoft Word D...	799 KB

After installation of tools, generate o drive for core project by clicking on subst_o.

Configuration

of a component in particular core can be done in Config.ini file at location

O:\norma_1.24\SourceCode\ AppConfig as shown below

Data2 (O:) > norma_1.24 > SourceCode > AppConfig				
	Name	Date modified	Type	Size
	📁 _Scripts	09-02-2021 10:40	File folder	
	📁 Common	09-02-2021 10:40	File folder	
	📁 Config_AoUSafe	09-02-2021 10:40	File folder	
	📁 Config_EthAdap	09-02-2021 10:40	File folder	
	📁 Config_EyeQcdd	09-02-2021 10:41	File folder	
	📁 Config_IoHwAb	09-02-2021 10:41	File folder	
	📁 Config_Proxy	09-02-2021 10:41	File folder	
	📁 Config_PTM	09-02-2021 10:41	File folder	
	📁 Config_TestComp	09-02-2021 10:42	File folder	
	📄 App_Config	06-04-2020 15:55	Makefile	2 KB
	📄 Config	25-01-2021 18:53	Configuration setti...	8 KB
	📄 LoadConfiguration	14-11-2019 18:29	Windows Batch File	1 KB

Example of Config.ini is as shown below, we can see options to configure according to project architecture in core 1.2 or 3 for NVM,DEM,CDD,IOHW and PTM components.

EyeQ CDD Integration



```
MULTICORE_SUPPORT=TRUE

; NAME: NvM core assignment
; DESCRIPTION: This parameter specifies which core the NvM will reside on
; USER: Application and Core
; POSSIBLE VALUE(S): {0,1,2}
; EXAMPLE: NVM_CORE_ID=0
NVM_CORE_ID=1

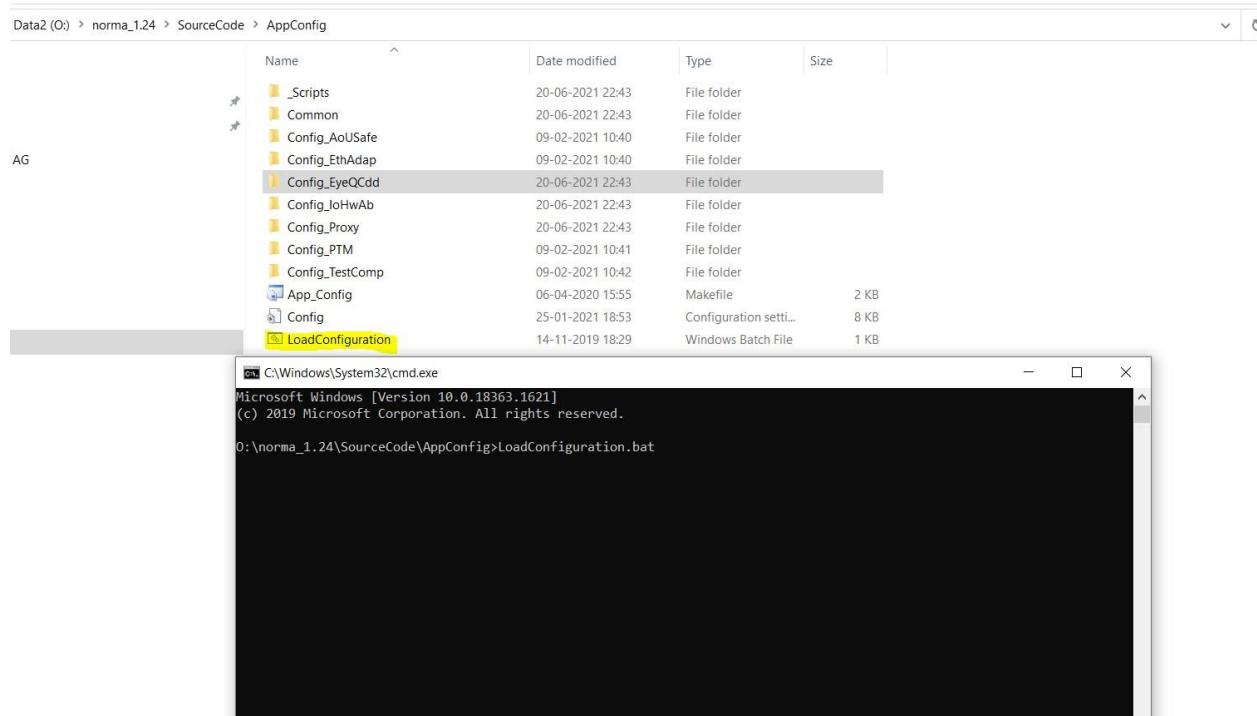
; NAME: Dem core assignment
; DESCRIPTION: This parameter specifies which core the Dem will reside on
; USER: Application and Core
; POSSIBLE VALUE(S): {0,1,2}
; EXAMPLE: DEM_CORE_ID=0
DEM_CORE_ID=1

; NAME: Composition Reference (x-path)
; DESCRIPTION: The reference to the AUTOSAR composition into which the component will be integrated
; USER: Application and Core
; POSSIBLE VALUE(S): any valid AUTOSAR x-path expression
; EXAMPLE: COMPOSITION_REF=/CoreCam/ECU_Composition
COMPOSITION_REF=/CoreCam/ECU_Composition
```

Save the file after configuring according to project architecture. We need to generate now excel sheets for CDD, PROXY, IOHW and PTM. Proxy component has NVM and DEM related ports and interfaces.

To generate excel sheets we need to run LoadConfiguration batch file in command prompt as below

EyeQ CDD Integration



Excels for EYEQCDD,IOHWAB etc are generated and can be as seen below

```
C:\Windows\System32\cmd.exe - LoadConfiguration.bat
Begin customization process... | 06/20/2021 | 22:42:59 | User: Z651061
ARXML generation tool was found at path = C:\Users\z651061\Desktop\ARXML_Generators
The configuration file: 0:\norma_1.24\SourceCode\AppConfig\Config.ini was found.
Loading configuration file and parsing...
Validating user inputs...
All user parameters are valid.
Deleting old generated files...
...Files were deleted
Creating EyeQCDD Generator...
...EyeQCDD Generator created!
Customization of Satellite Core 1...
...Satellite Core1 customization complete!
Creating IoHwAb Generator...
...IoHwAb Generator created!
Creating Proxy Generator...
...Proxy Generator created!
Creating PTM Generator...
!!!!!!ERROR!!!!!
Unable to make a substitution for placeholder <insert_HilAdapter_ConfigDoc>
Row: 6
Worksheet: RevisionHistory
Workbook: 0:\Application\SourceCode\CDD\PTM\Config\PTM.xlsx
Press any key to continue . . .
```

Paths of excel sheets are folders of configuration like for EYEQCDD at O:\norma_1.24\SourceCode\AppConfig\Config_EyeQCdd as below

EyeQ CDD Integration

Data2 (O:) > norma_1.24 > SourceCode > AppConfig > Config_EyeQCdd				
AG	Name	Date modified	Type	Size
	ARXML	20-06-2021 22:43	File folder	
	include	09-02-2021 10:41	File folder	
	EyeQCdd	20-06-2021 22:43	Microsoft Excel M...	142 KB
	EyeQCddSatCore1	20-06-2021 22:43	Microsoft Excel M...	46 KB

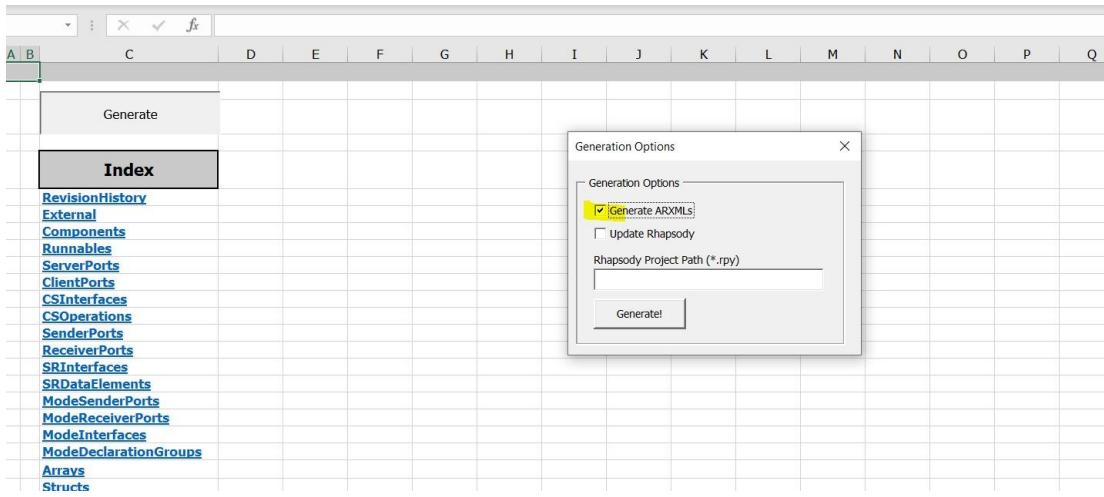
Now we need to generate arxmls by opening EyeQCDD Excel and clicking on generate option as below .

The screenshot shows a Microsoft Excel spreadsheet titled "EyeQCdd - Excel". The spreadsheet has two columns, A and B. Row 1 contains column headers. Row 2 has a yellow box around cell B2 containing the word "Generate". Row 3 has a black box around cell A3 containing the word "Index". Rows 4 through 29 list various components and ports, each preceded by a blue link icon. The rows are numbered from 1 to 29.

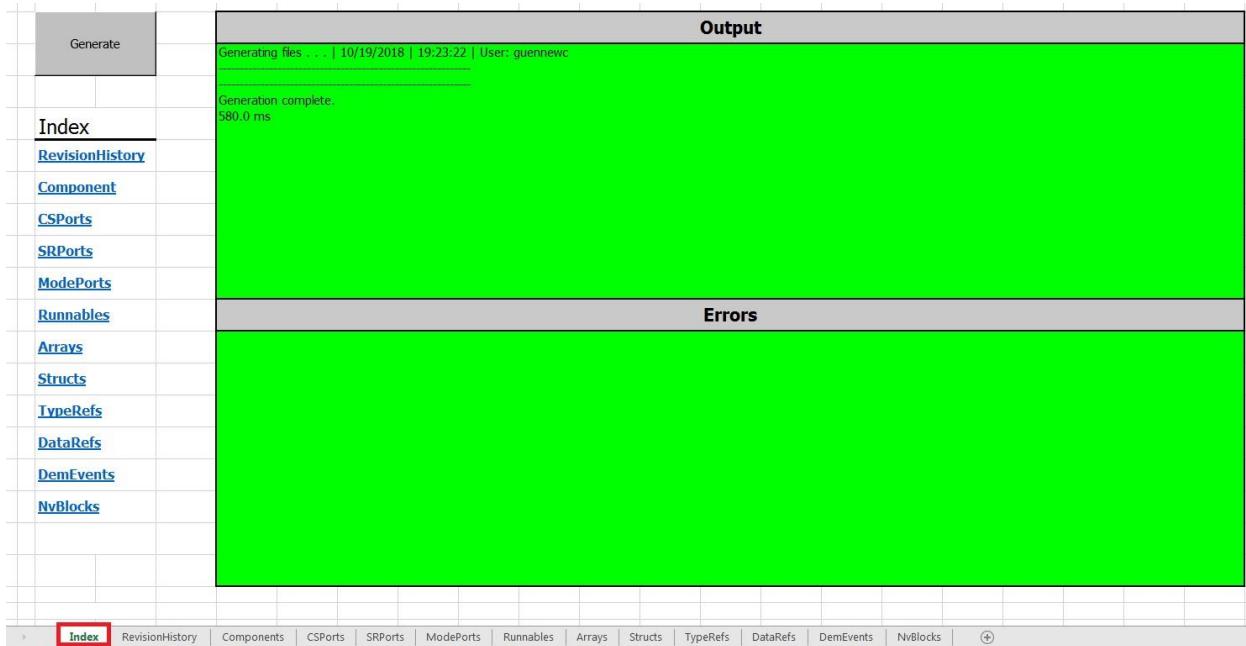
A	B
1	
2	Generate
3	Index
4	RevisionHistory
5	External
6	Components
7	Runnables
8	ServerPorts
9	ClientPorts
10	CSInterfaces
11	CSOperations
12	SenderPorts
13	ReceiverPorts
14	SRInterfaces
15	SRDataElements
16	ModeSenderPorts
17	ModeReceiverPorts
18	ModeInterfaces
19	ModeDeclarationGroups
20	Arrays
21	Structs
22	TypeRefs
23	DataRefs
24	DemEvents
25	NvBlocks
26	AlivenessCheckPoints
27	
28	
29	

Now window will pop up select , Generate ARXML and click generate

EyeQ CDD Integration



A window will now open with generation completed. We can now see ARXML at path O:\norma_1.24\SourceCode\ AppConfig\Config_EyeQCdd\ARXML for EyeQCDD .



Similarly we can generate ARXML for IOHWAB,PROXY,PTM etc components .Now ARXMLS are ready to be integrated .

EyeQ CDD Integration

9. Core Software Integration Steps in detail Static files:

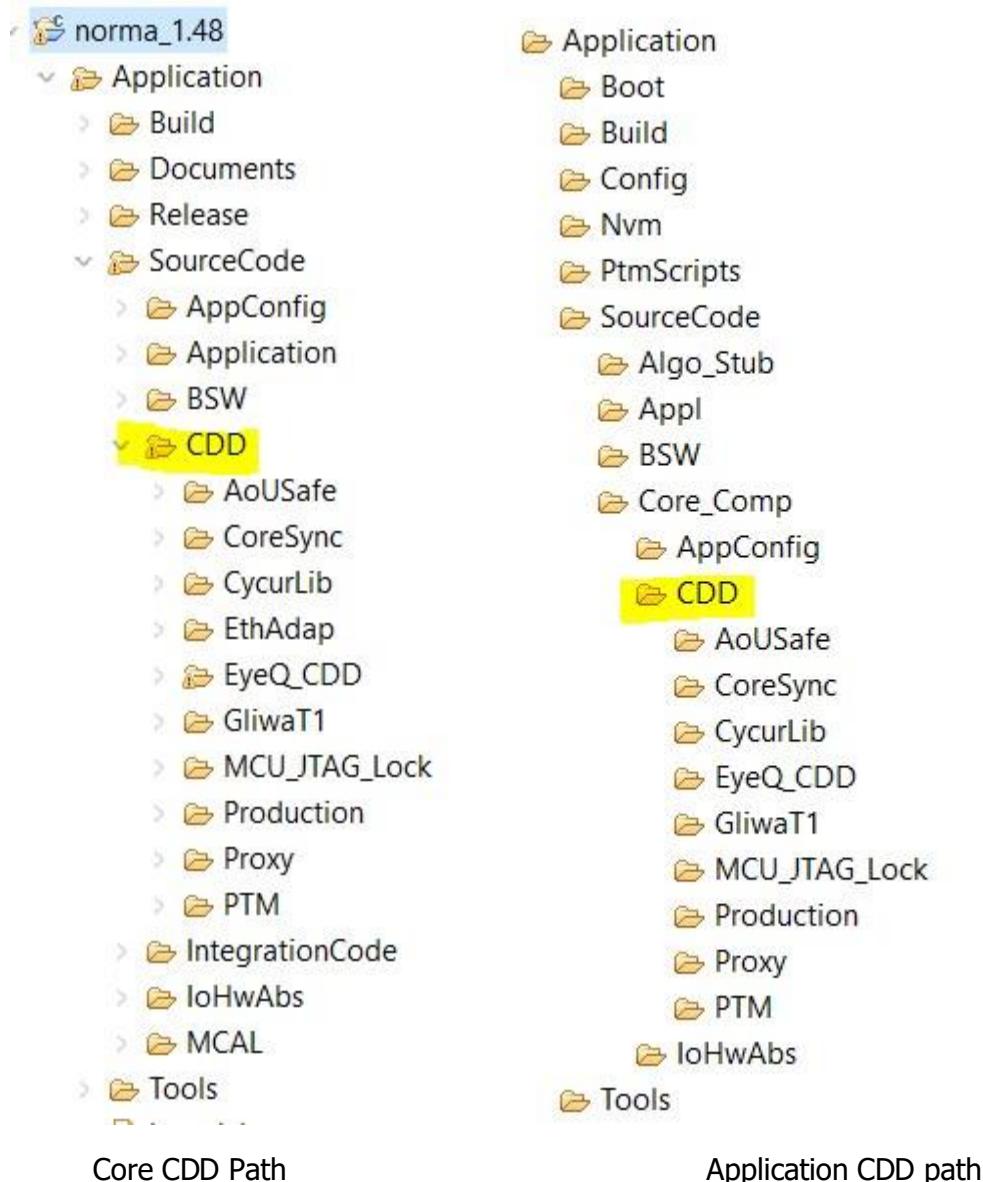
9.1 CDD Component Integration (Source code)

In core sandbox we can find CDD in below path as shown below

Application\SourceCode\CDD -- Core CDD path

Application sandbox of GWM camera we have CDD folder in below path as shown

Application\SourceCode\Core_Comp\CDD --GWM CDD path



Core CDD Path

Application CDD path

EyeQ CDD Integration

This folder contains the source code for EyeQ CDD, Production, CoreSync, Cycurlib, GliwaT1, PTM, MCU_JTAG_LOCK and AOUSAFA. We need to integrate these static files from core to application.

Below are points need to be considered while integration.

- 1) Before merging changes from core please compare CDD changes from earlier checkpoint to identify changes.
- 2) We need to merge the changes which are done in current check point. Please do not overwrite any changes done in application from core .If we not sure about any point to merge ,we need to drop mail and get confirmation from application owner who has done changes(Ideally we need to take from core but in GWM this is how it is followed.)
- 3) PTM source code can be adapted from Application software. The destination folder in Application software may vary across the projects. PTM,GliwaT1 integration is taken care at STC.
- 4) EyeQ CDD, Production, CoreSync and Cycurlib need to be merged as it is from core by following above points. MCU_JTAG_LOCK feature is not used in GWM and variants. We will discuss about AOUSAFA in detail in upcoming chapters.

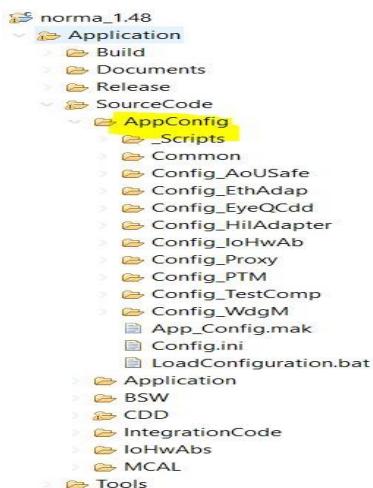
9.2 App Config Integration

Below is path of App config folder in core

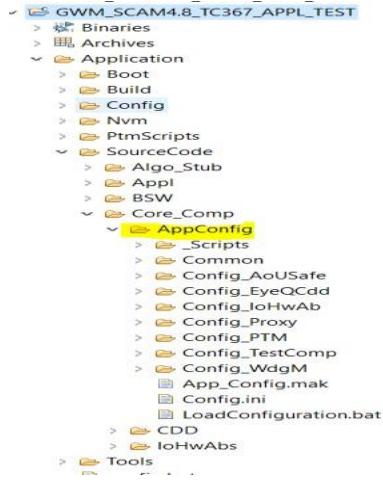
norma_1.48\Application\SourceCode\ AppConfig --Core path

Below is path of App config folder in GWM application project

Application\SourceCode\Core_Comp\ AppConfig --GWM application Path



AppConfig path in core



AppConfig path in GWM application

EyeQ CDD Integration

This configuration can be modified according to project needs. In this folder Config_AoUSafe, Config_EyeQCDD, Config_IoHWb,Config_WdgM need to copied from core and can be modified according to project needs.

Config_Proxy contain code from ARXML generator tool which we discussed earlier chapter of ARXML generator. We need to copy generated code to application folder. Test components is not required to copy and PTM can be ignored as done at STC.

NOTE: Component configurations may need to adapt from Application software.

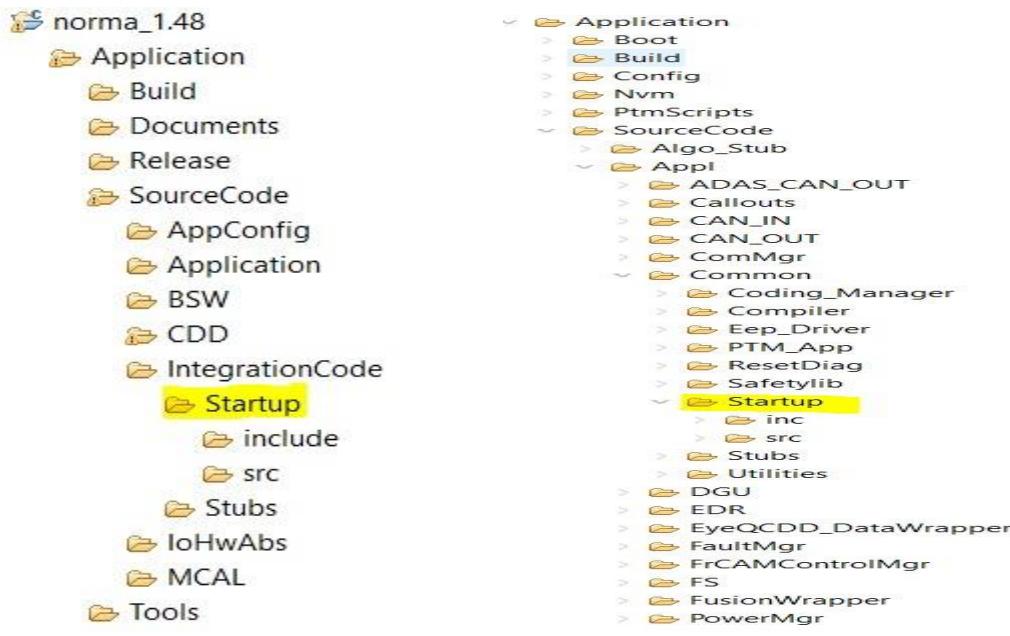
9.3 IoHwAb Integration

This folder contains the source code for IOHWAB component as explained in **Heading 5**.

The folder **IoHwAb** should be copied from Core software(refer to **Heading 5**) to Application software. The destination folder in Application software may vary across the projects.

9.4 Startup code integration

The folder **Startup** should be copied from Core software to Application software.



Core start up path

GWM start up path

EyeQ CDD Integration

Application\SourceCode\IntegrationCode\Startup --core path
Application\SourceCode\Appl\Common\Startup – GWM path

NOTE : Some of the files may be adapted from application software(Project specific).

9.5 AoUSafe Integration:

AoUsafe folder path is in CDD folder as discussed earlier. We need to take all files as it is from core .For AoUSafe to work without issues we need to configure certain files according to project .In path "Application\SourceCode\Core_Comp\Config_AoUSafe\include" we need to configure AoUSafe_Application_Config.h file according to project architecture .We can enable safety or disable from this file .There are many other configuration in this file which can be modified.

```
33
35* /*
37
38 #include "EcuM.h"
39 #include "AoUsafe_Std_Types.h"
41*/
45* /* @details Enable/disable Safety
47 #define AOUSAPE_TEST_ENABLE (AOUSAPE_SUPPORT_OK)
49* /* @details Enable/disable SMU
53 #define AOUSAPE_SMU_DISABLE (AOUSAPE_SUPPORT_NOK)
55* /* @details Enable/disable SMU
59 #define AOUSAPE_DEBUG_SUPPORT (AOUSAPE_SUPPORT_NOK)
60
61* /* @details Enable/disable STM debug information
65 #define AOUSAPE_STM_DEBUG_SUPPORT (AOUSAPE_SUPPORT_NOK)
66
67* /* @details Enable/disable STM debug information
71 #define AOUSAPE_MBIST_DEBUG_SUPPORT (AOUSAPE_SUPPORT_NOK)
72
73* /* @details Enable/disable STM debug information
77 #define AOUSAPE_SFRTEST_DEBUG_SUPPORT (AOUSAPE_SUPPORT_NOK)
78
79* /* @details Enable/disable ACCPROT debug information
83 #define AOUSAPE_ACCPROT_DEBUG_SUPPORT (AOUSAPE_SUPPORT_NOK)
84
85* /* @details Enable/disable FEE
89 #define AOUSAPE_SUPPORT_FEE (AOUSAPE_SUPPORT_NOK)
90
92* /* @details Enable/disable Last SMU Reset reason
96 #define AOUSAPE_LAST_SMU_RESET_REASON_API (AOUSAPE_SUPPORT_OK)
97* /* @details Enable/disable safety test
101 #define AOUSAPE_LOCKSTEP_ENABLE_TEST_SUPPORT (AOUSAPE_SUPPORT_OK)
102
103* /* @details Enable/disable safety test
107 #define AOUSAPE_ACCPROT_SUPPORT (AOUSAPE_SUPPORT_OK)
108
109* /* @details Enable/disable safety test
113 #define AOUSAPE_MONBIST_SUPPORT (AOUSAPE_SUPPORT_OK)
114
115* /* @details Enable/disable safety test
119 #define AOUSAPE_SMU_REGMON_COREALIVE_TEST_SUPPORT (AOUSAPE_SUPPORT_OK)
```

The folder **AoUSafe** should be copied from Core software(refer to **Heading 5**) to Application software.

NOTE : When the core platform integration is done for the first time,it is suggested that **AoUSafe** integration can be done at last.

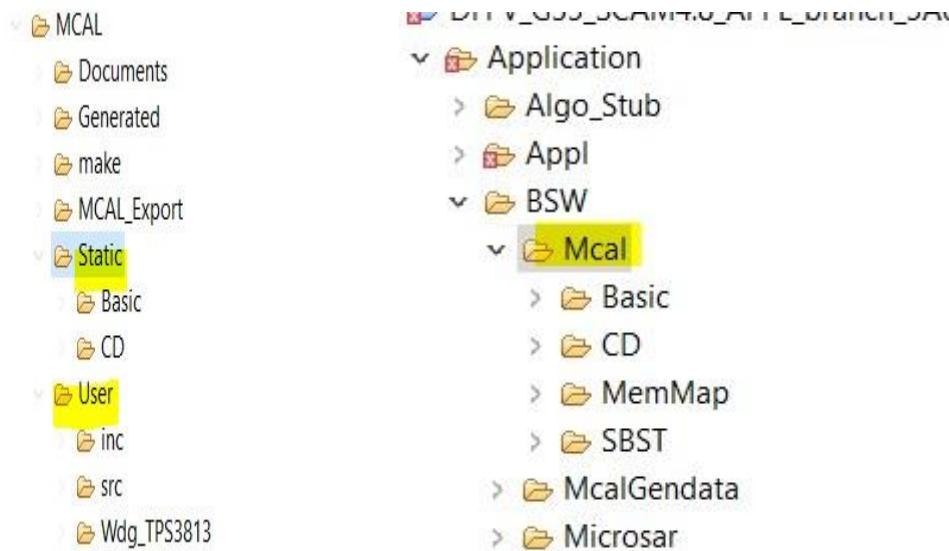
EyeQ CDD Integration

9.6 MCAL Static code Integration

The folder **MCAL(static and user should not be missed)** should be copied from Core software to Application software. The destination folder in Application software may vary across the projects.

Application\SourceCode\MCAL -core path

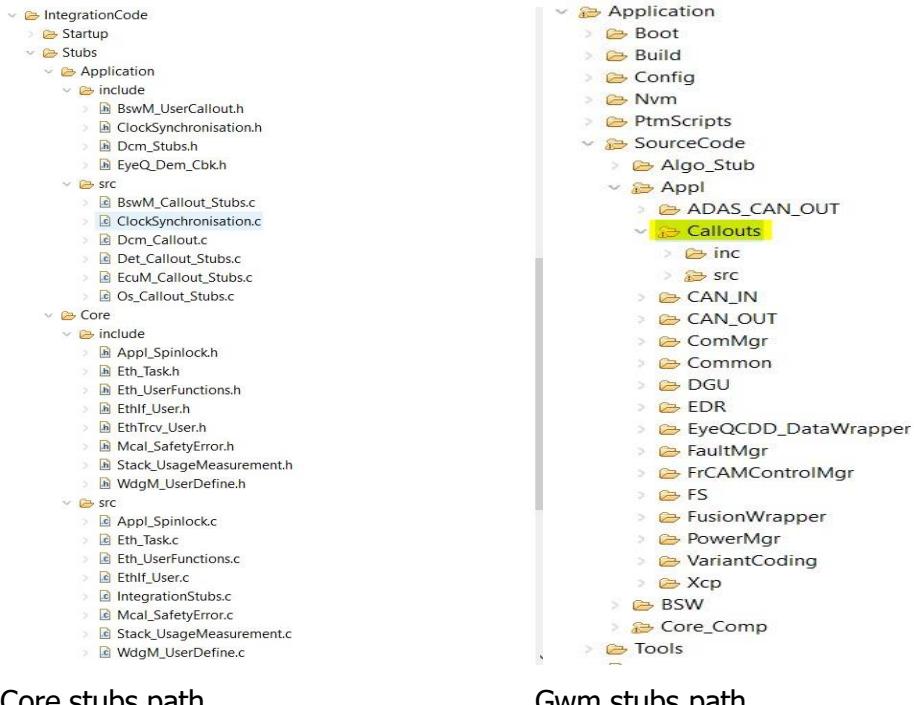
Application\BSW\Mcal -GWM path



9.7 Other Static code Integration.

There are other files in Application\SourceCode\IntegrationCode\Stubs should be integrated EcuM_Callouts_Stubs,BswM_Callouts_Stubs should be carefully integrated which as satup and shutdown strategy.Stubs in GWM are at Application\SourceCode\Appl\Callouts.

EyeQ CDD Integration



Core stubs path

Gwm stubs path

10. ARXML import and BSW updates

Prerequisites

- Davinci Tools shall compatible with Vector SIP version.
- Platform compatible Tasking version should be installed.

10.1 Importing Arxmls to the Davinci Developer

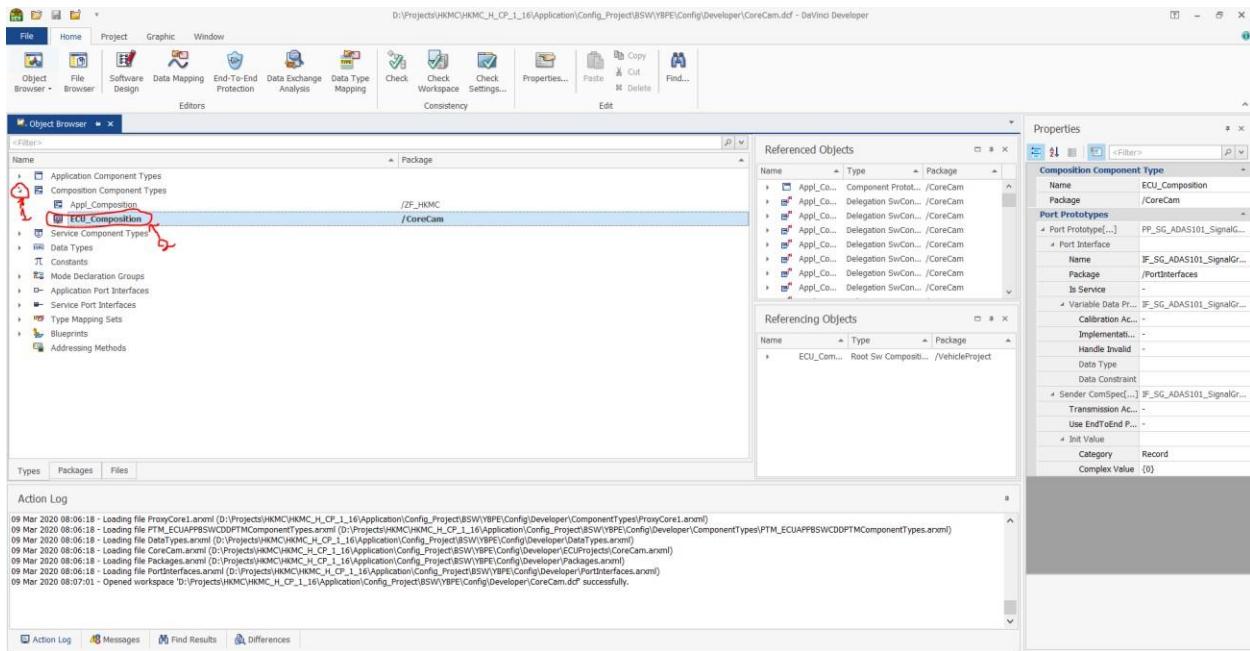
The first step in integration procedure is to import necessary Arxml files which are generated using Arxml generator procedure of EyeQ CDD, Proxy component and IOHWAB component which contains the description of the components.

These components can be overwritten on existing components or can be deleted from and imported into the project.

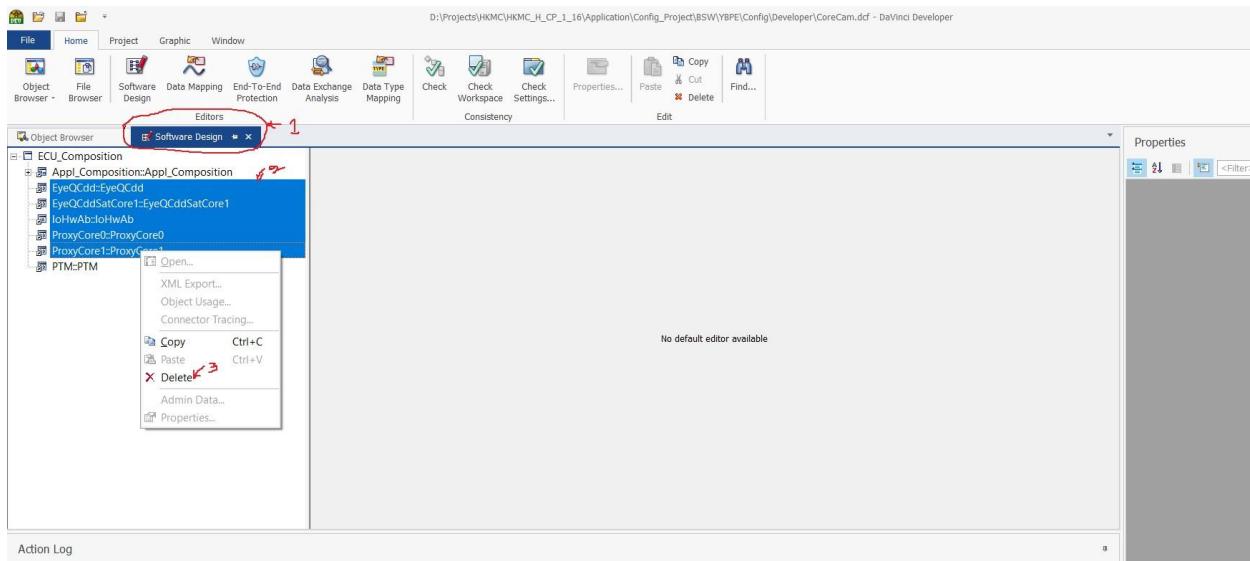
To delete components from project first delete components from ECU Composition and then delete from Application component types.

Click on Composition Component Types and then double click on
Ecu_Composition

EyeQ CDD Integration

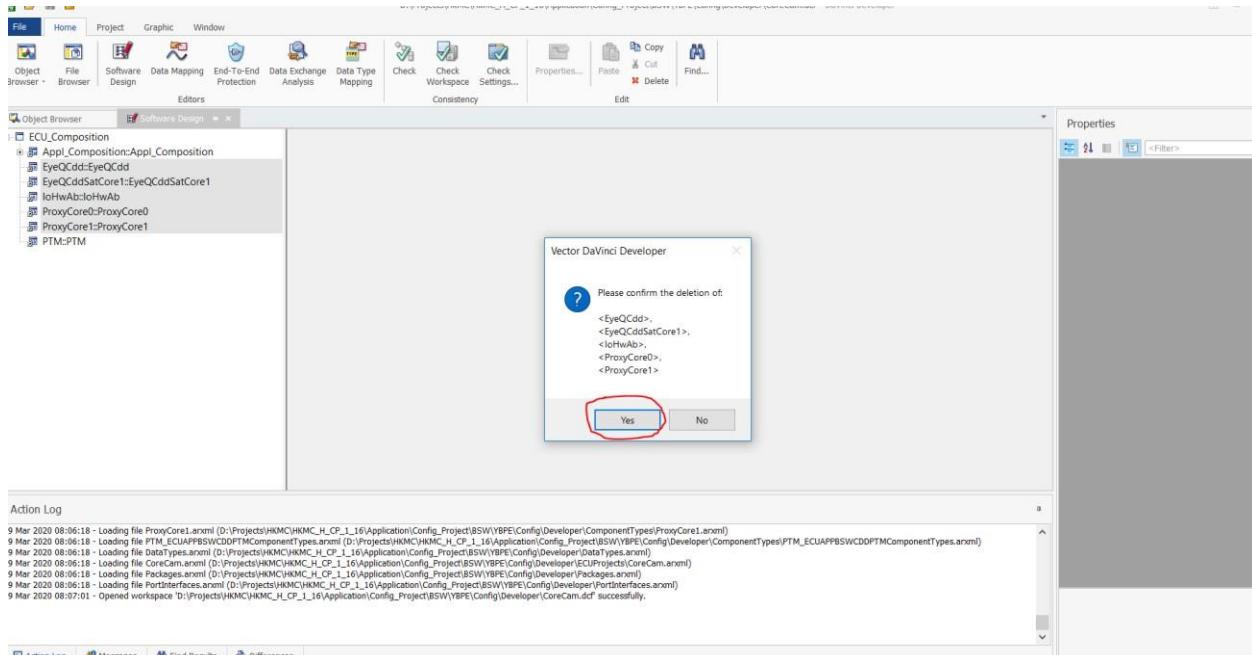


Select the EyeQ CDD core components from software design tab and right click and select delete.

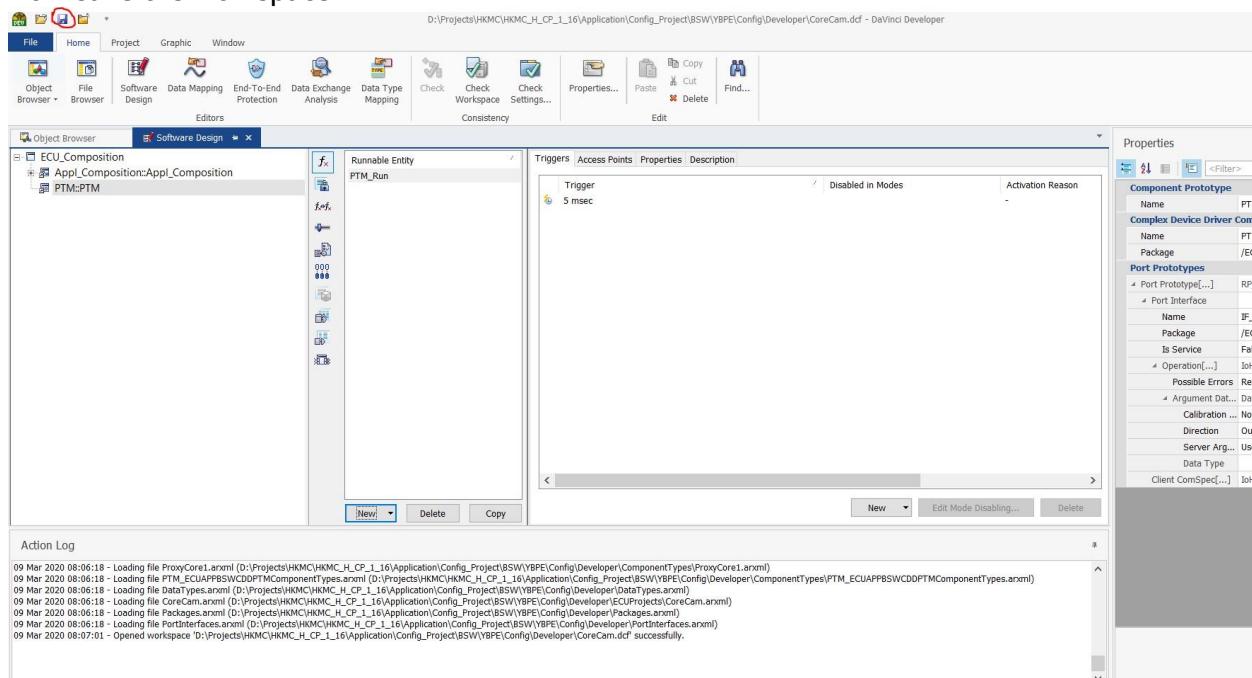


Click on Yes

EyeQ CDD Integration

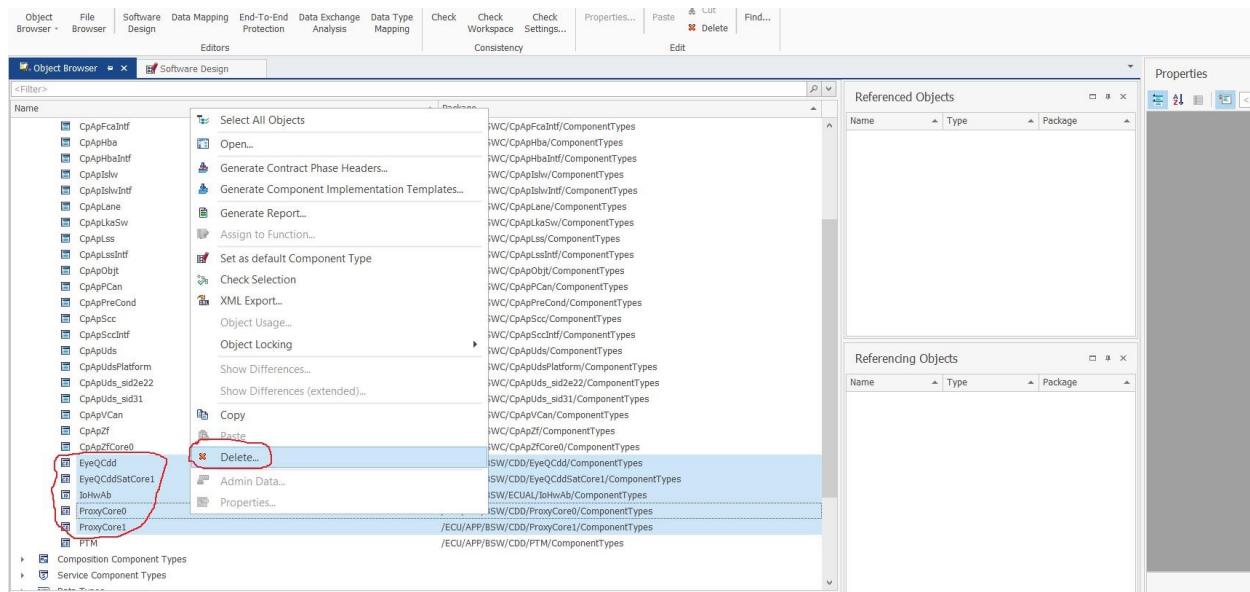


Now save the workspace.



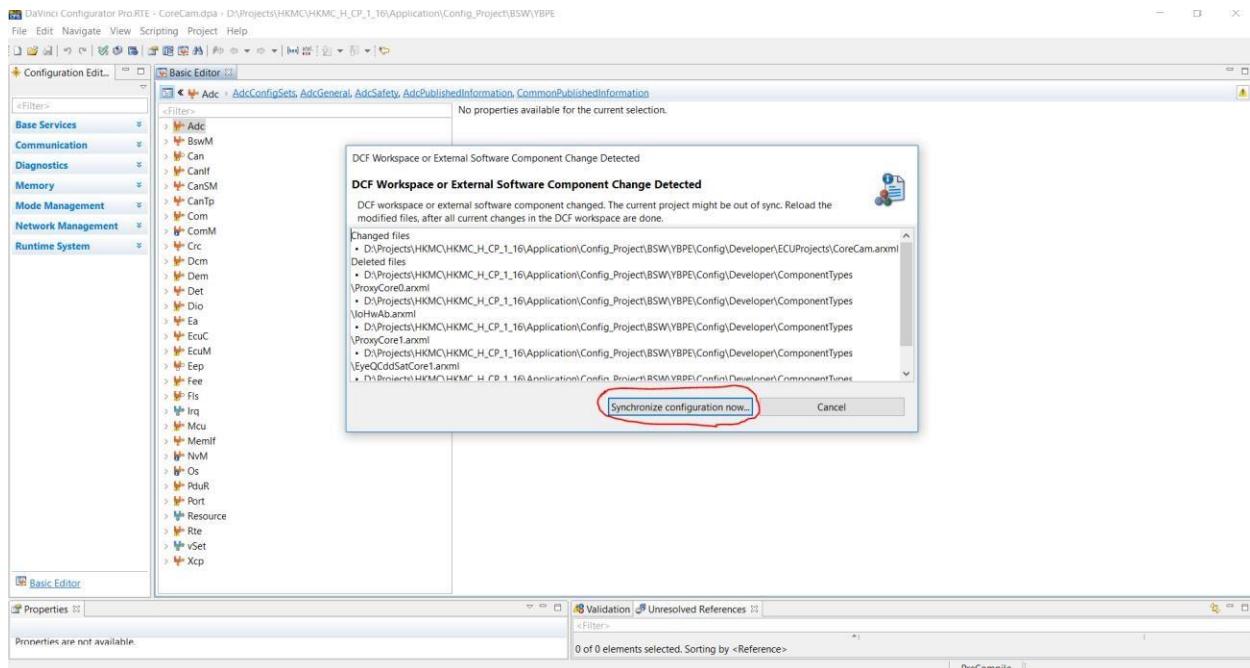
Now delete Application component types of EyeQ CDD Core from Application component types in object browser tab.

EyeQ CDD Integration



Now save the project.

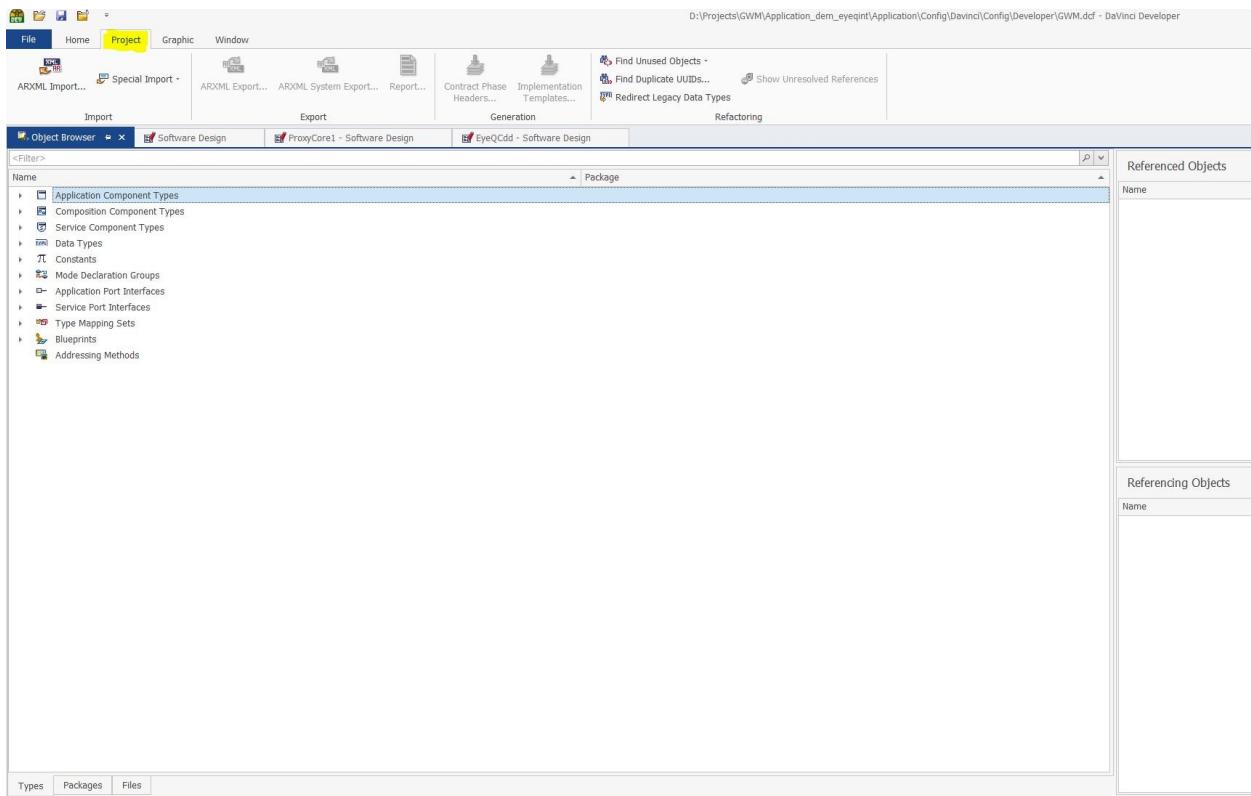
Now open Configurator and the screen should appear as shown, then click on **Synchronize configuration now** and the click on save.



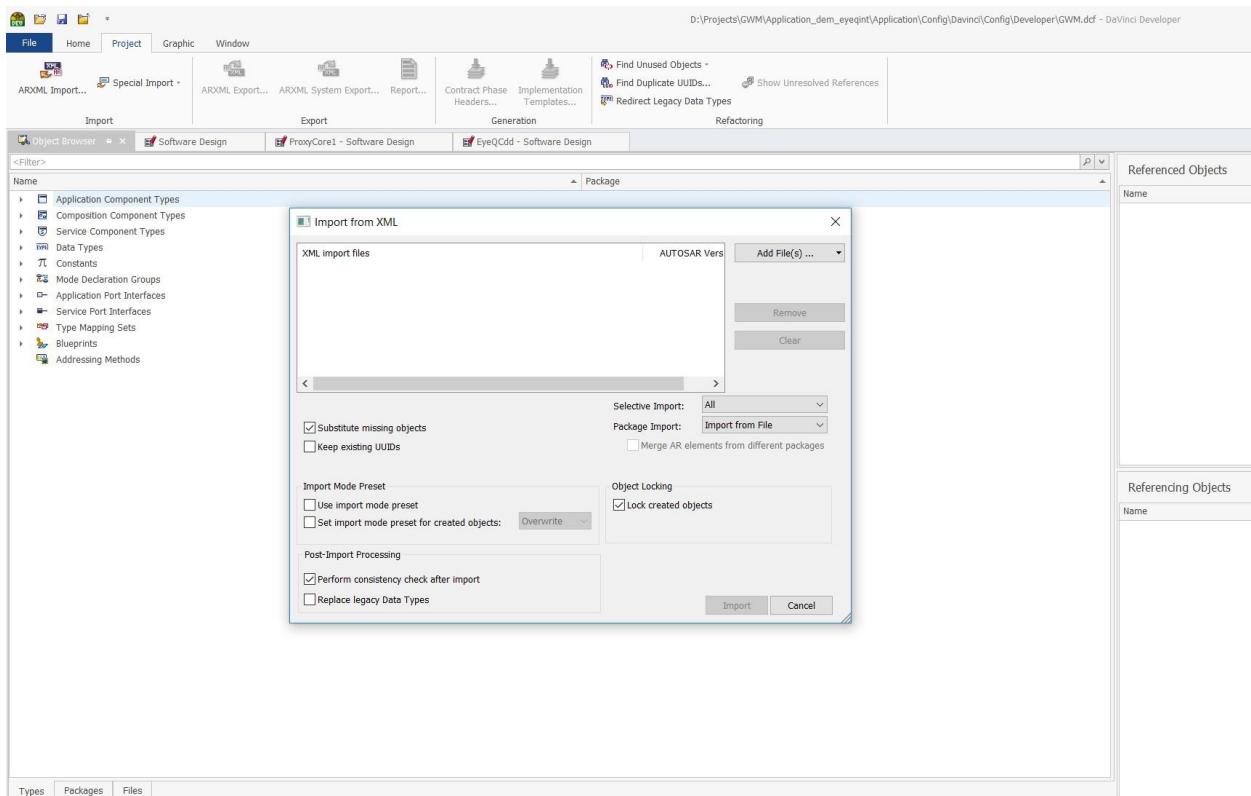
-->Open Davinci Developer project

-->Select project tab

EyeQ CDD Integration

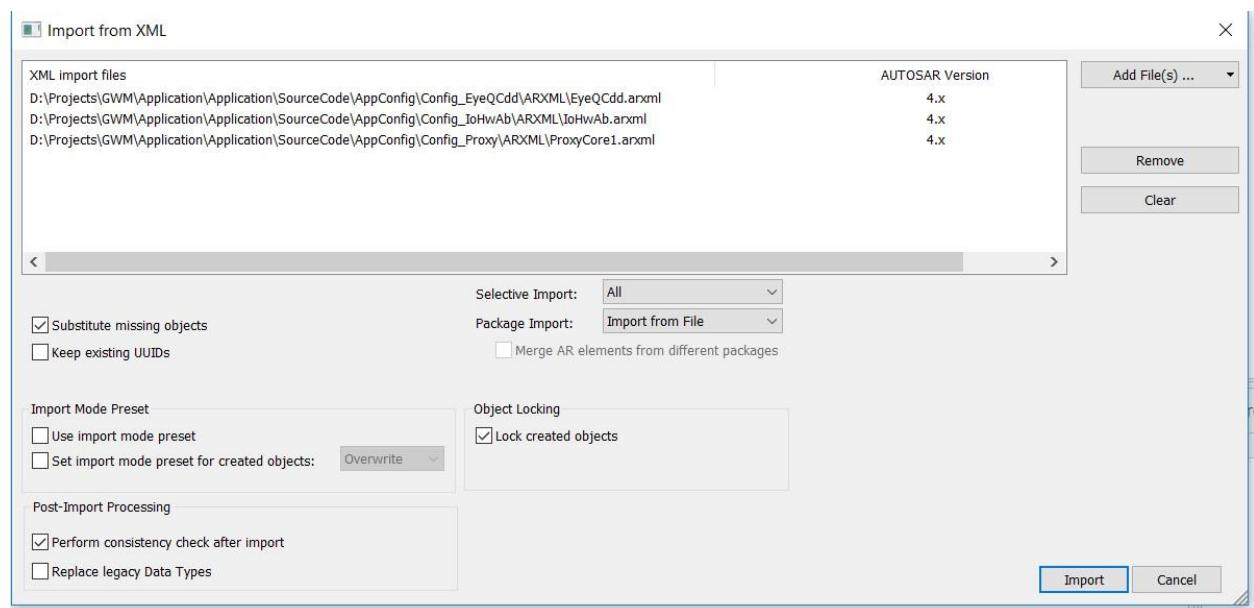
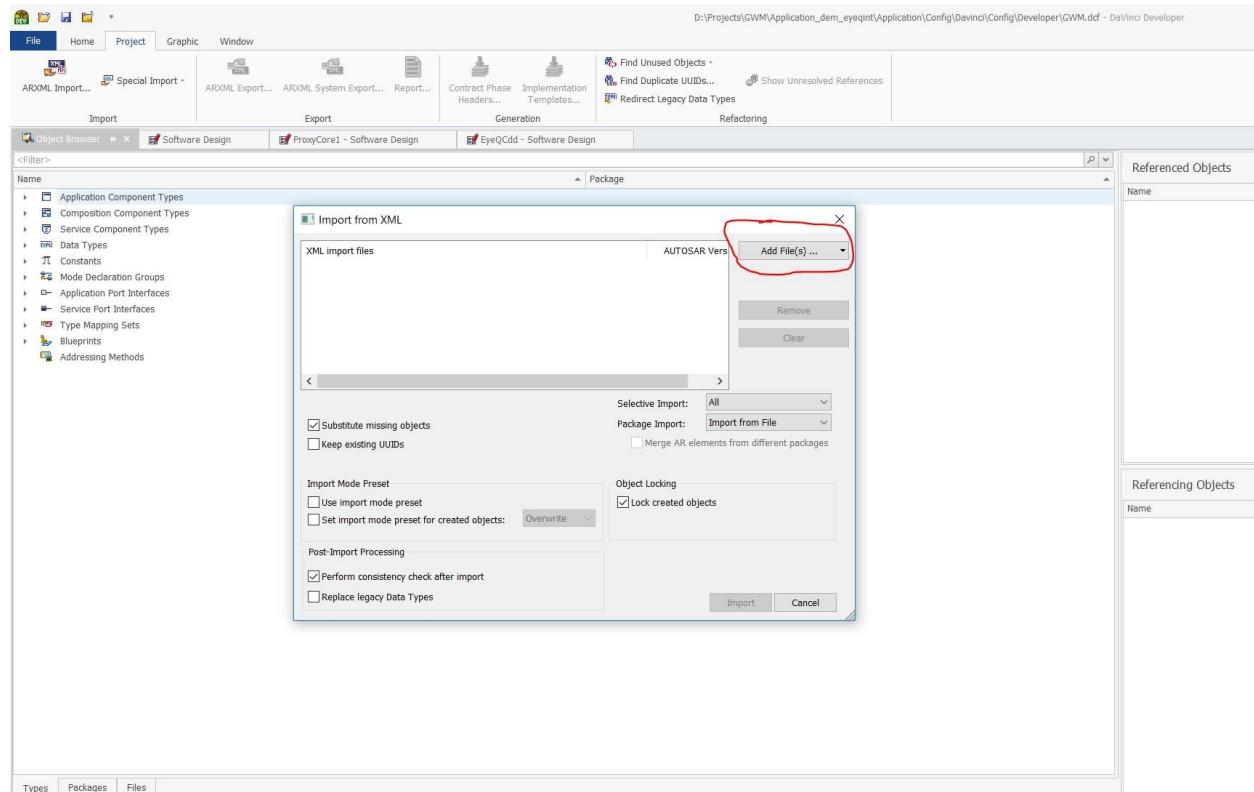


-->Click on ARXML Import



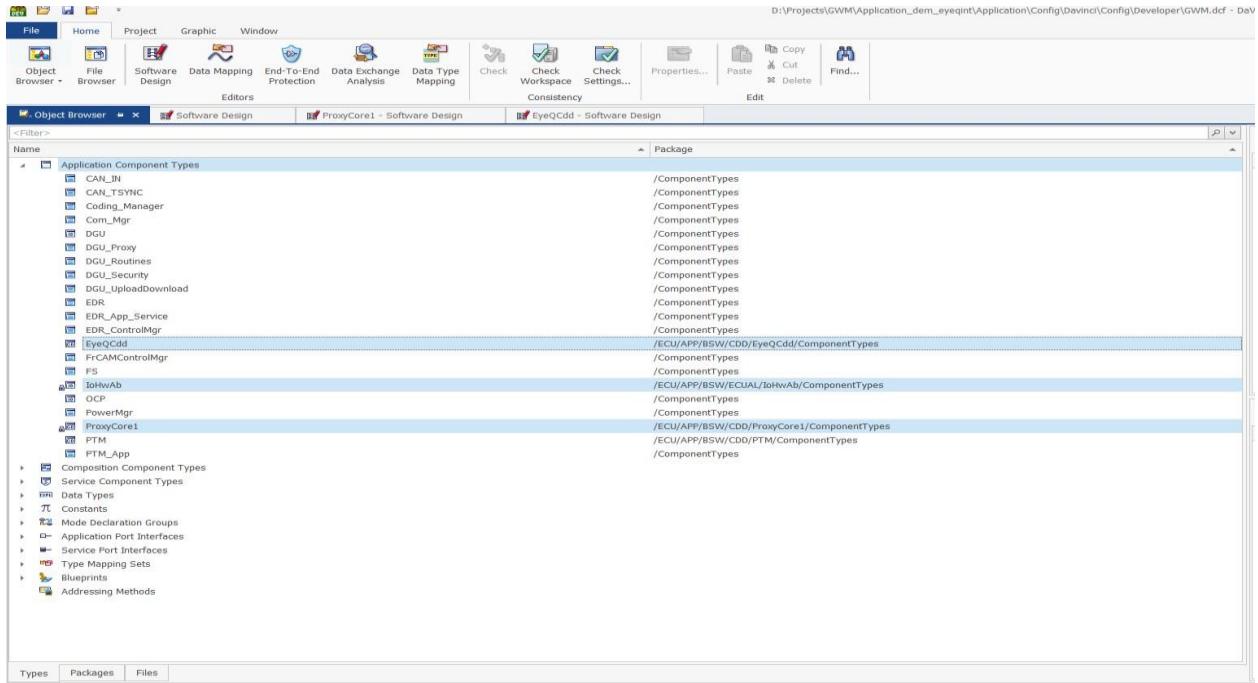
EyeQ CDD Integration

-->Click on Add Files and then select EyeQCdd.arxml, IoHwAb.arxml, ProxyCore1(or ProxyCore0 depends on architecture).arxml from AppConfig folder which are generated as per project needs by Arxml generator tool in the project and then click on import.



EyeQ CDD Integration

Once imported successfully these components should reflect in Home-->Object Browser->Application Component Types



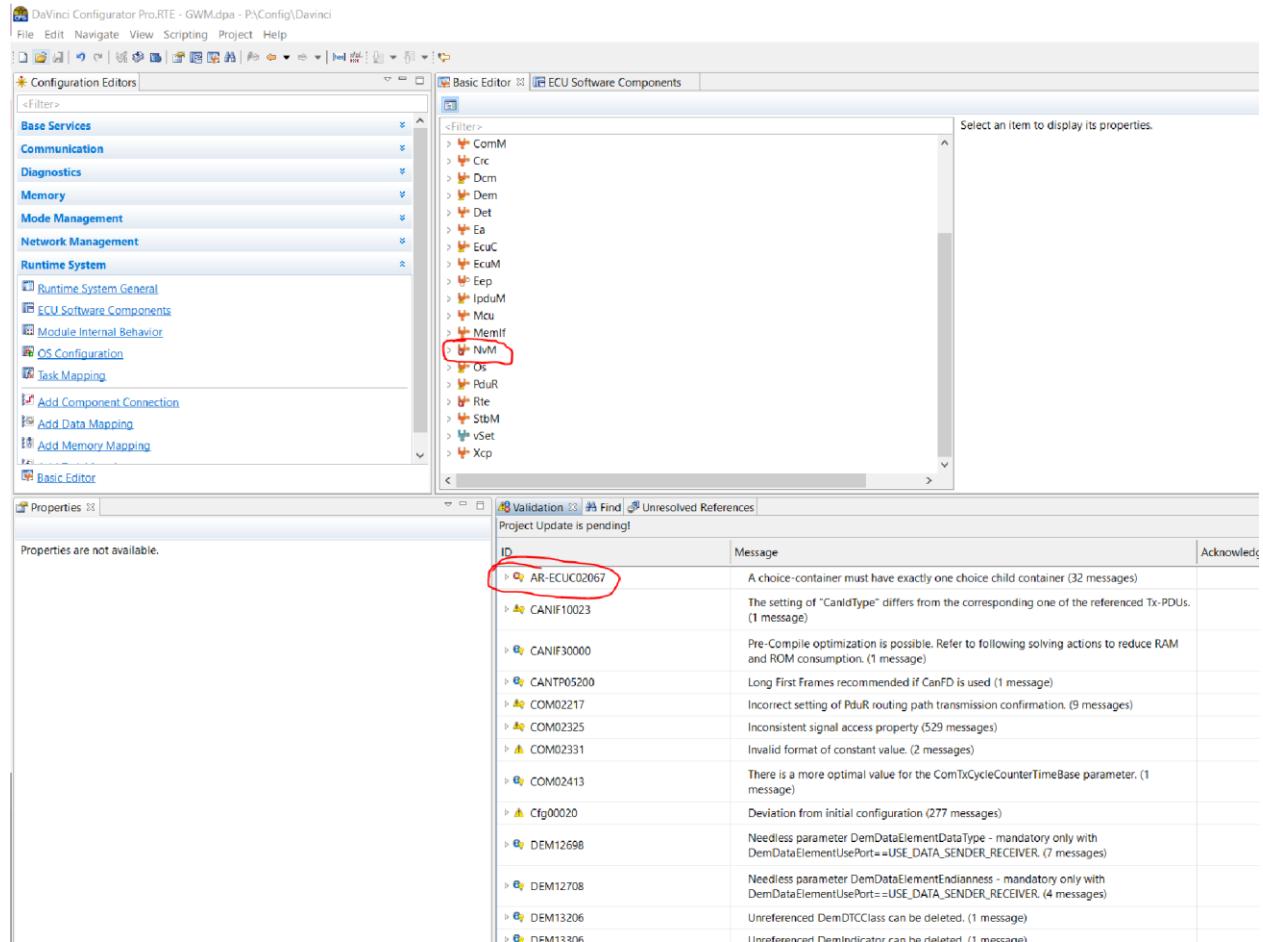
NOTE: If components are newly added the components should appear in **Application Component Types**. If components are already exists then components can be overwritten.

EyeQ CDD Integration

10.2 NvM Integration

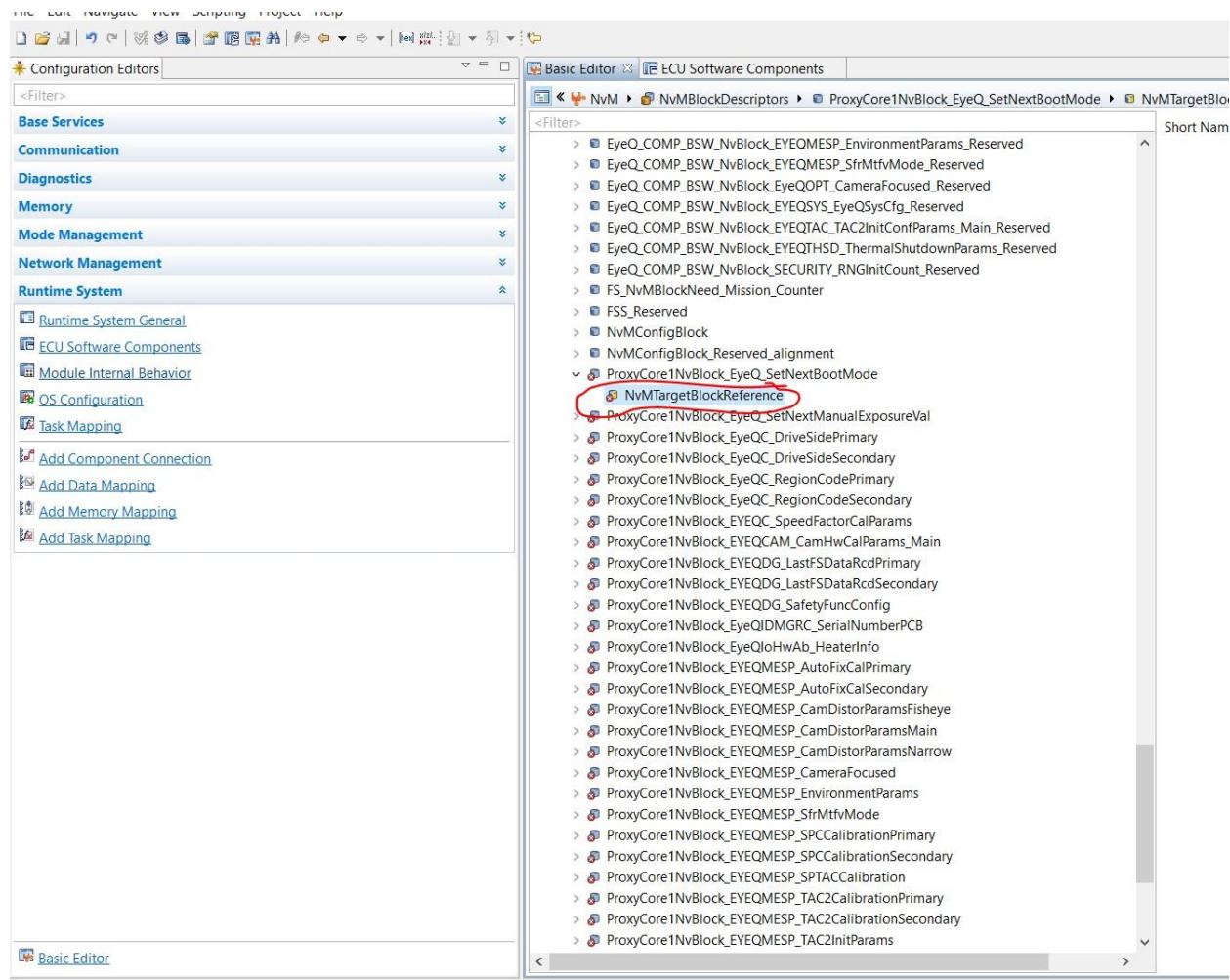
Once the arxmls are imported Davinci configurator will synchronise with the Davinci developer project. After synchronization the NvM configuration needs will be updated. Follow the below procedure to configure NvM.

Check for NvM errors in **validation** tab.



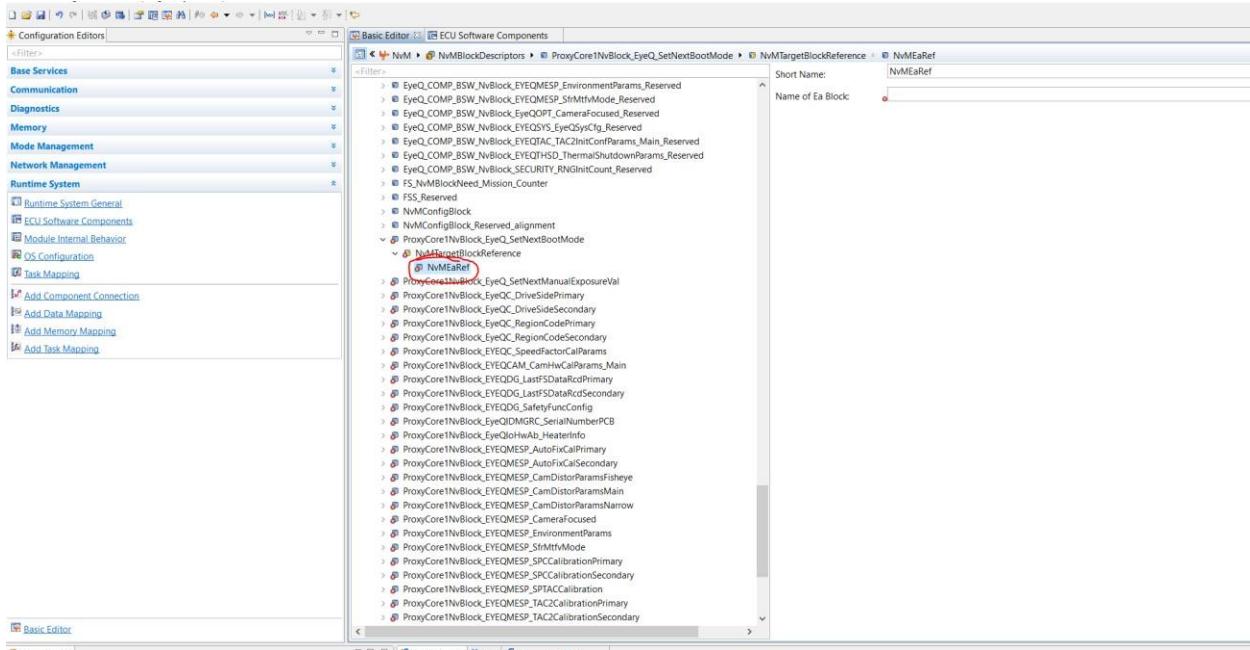
Go to faulty blocks in NvM Component and

EyeQ CDD Integration

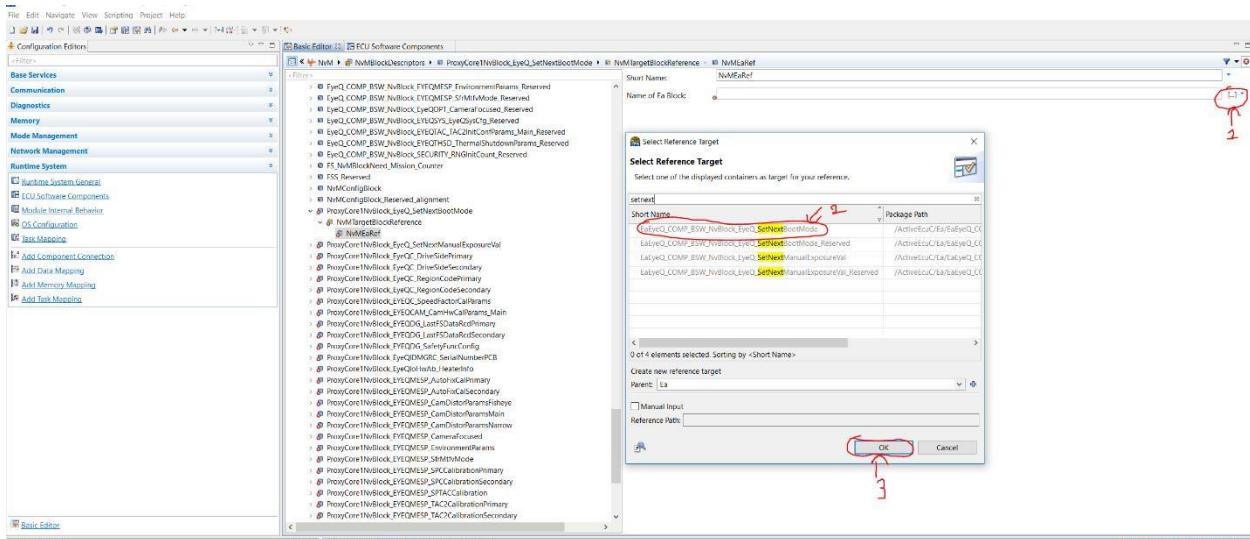


Right click on the Block reference and Choose Ea Block or FEE block based on project requirement.

EyeQ CDD Integration



Select an Ea block by clicking on select a reference target and then select the respective block.(Ea block configuration is taken for example)



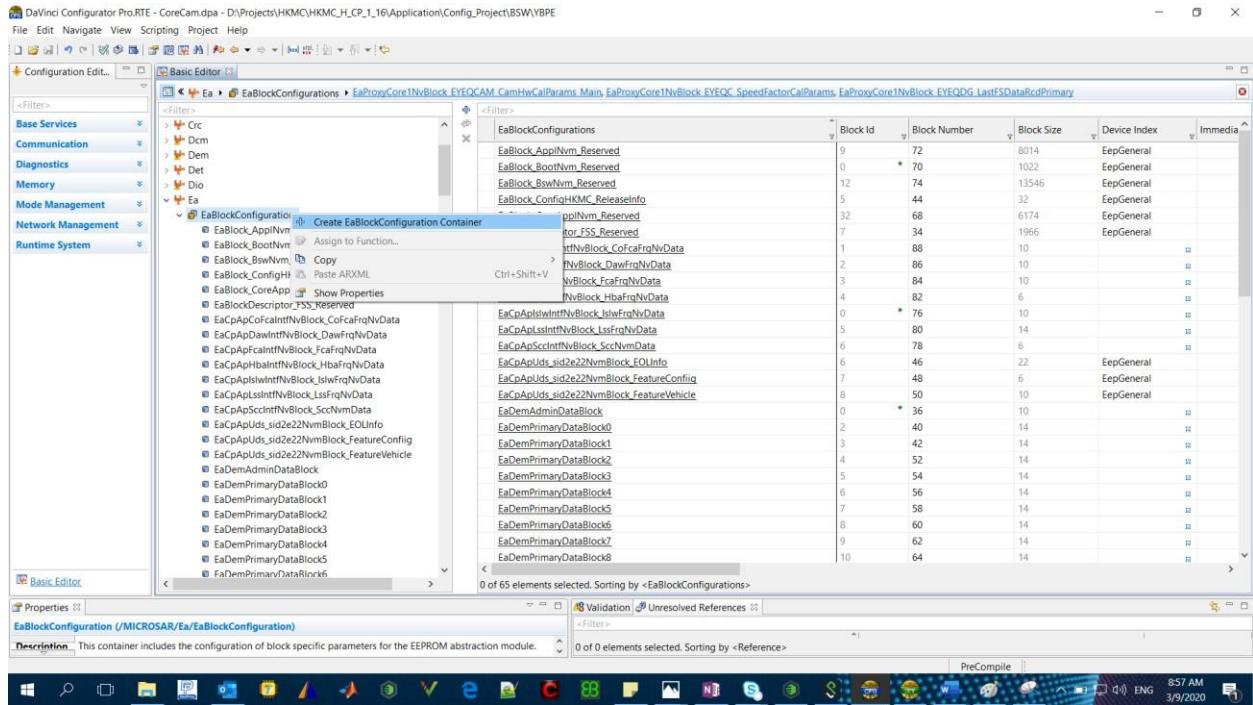
Repeat same steps for all faulty blocks and validate once finished.

If some blocks are newly added/removed in core software then create/delete Ea/Fee blocks first in Ea/Fee in EyeQ CDD CORE partition in Application software and then give the Ea/Fee block reference in NvM Configuration. Update the NvM block lengths as per CORE software requirement and adjust the Core NvM Reserved space in EyeQ CDD CORE partition

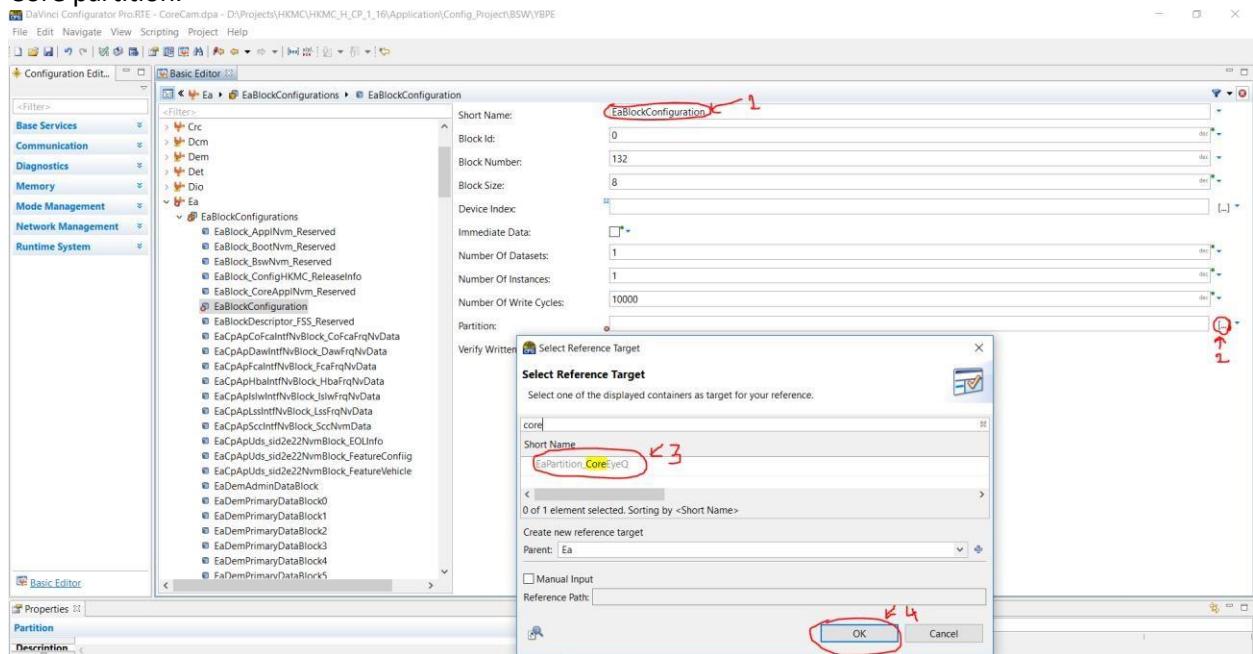
Creating and deleting Ea/Fee Blocks

EyeQ CDD Integration

In configurator open Ea module and the right click on **EaBlockConfigurations** and the select **CreateEaBlockConfigurationContainer**.



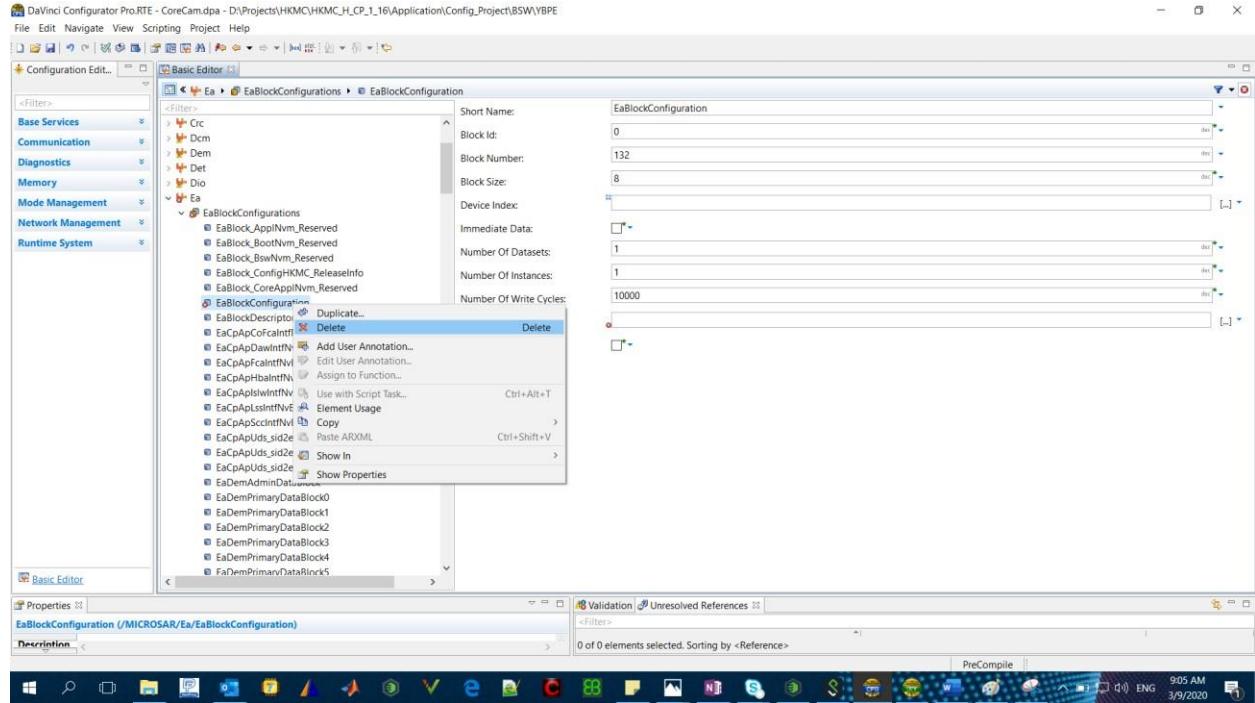
Click on created EaBlock and rename the block with Core specific block name and the select Core partition.



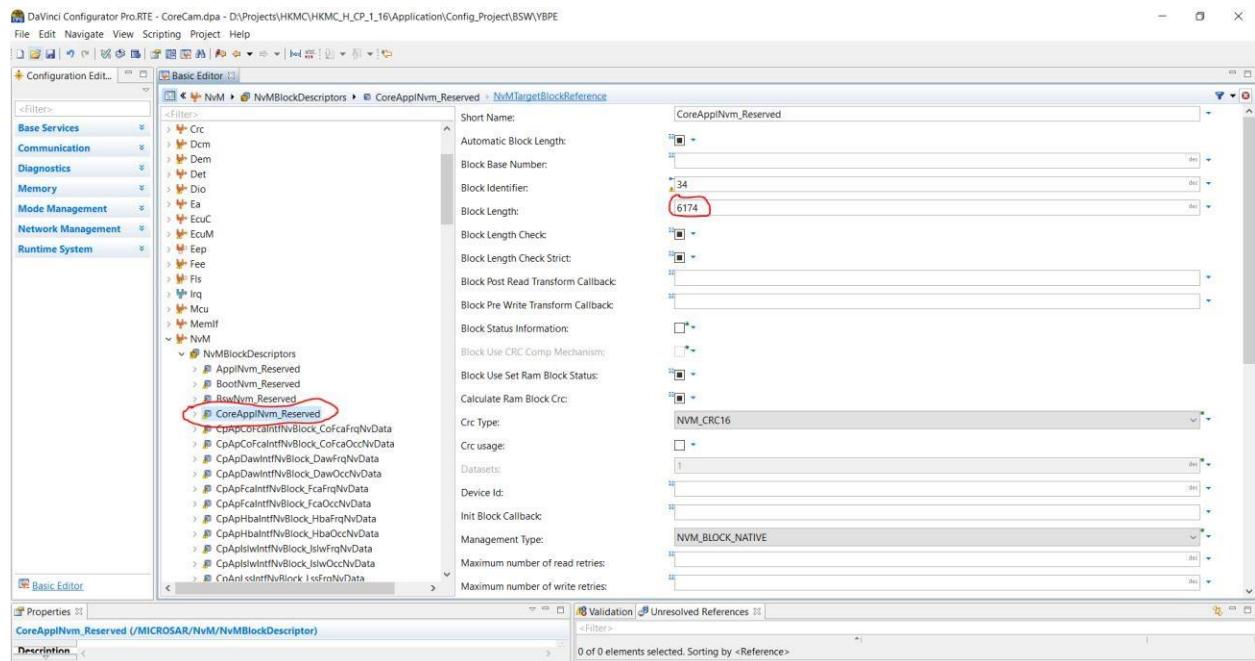
Repeat the above step for all newly created core blocks.

EyeQ CDD Integration

To Delete Ea block configuration which are removed from Core, Right click on the block to be removed and select delete.



Now adjust the Core reserved space according to new NvM layout of the Core.



If this step is not done then the size NvM partition for Core and the sum of configured core block lengths along with Core reserved block will not be same. Then Nvm and Ea modules

EyeQ CDD Integration

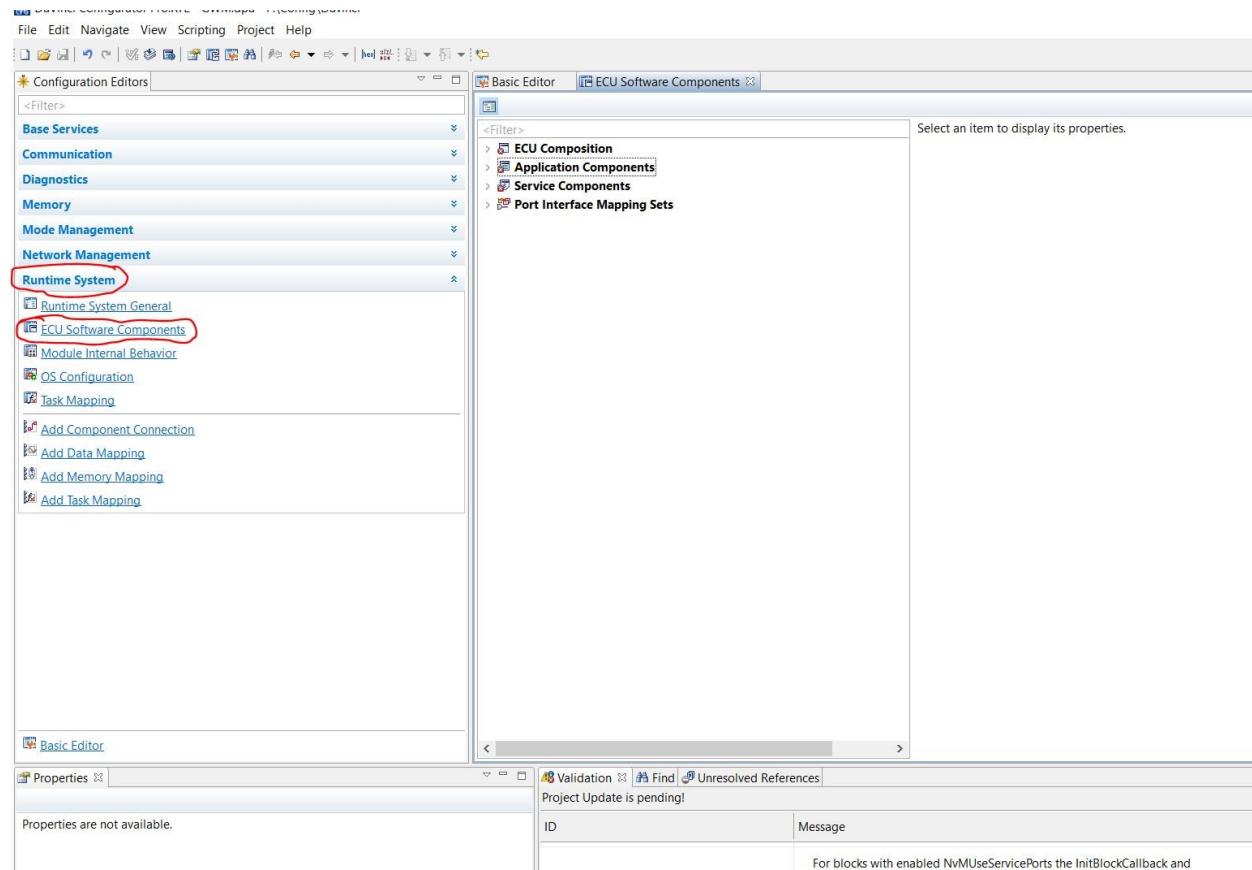
will throw an error. To calculate the Core partition length, Subtract the lengths of blocks which are newly added from Core reserved block length and add the lengths of the blocks which are removed/deleted to Core reserved block length. The lengths of CRC and Start and End markers of the blocks should be included.

Note: To refer Core NvM Blocks the information will be given in EyeQCDD_Platform_Integration_Guide in Core software.

After this respective changes in layout and .cfg should be done.

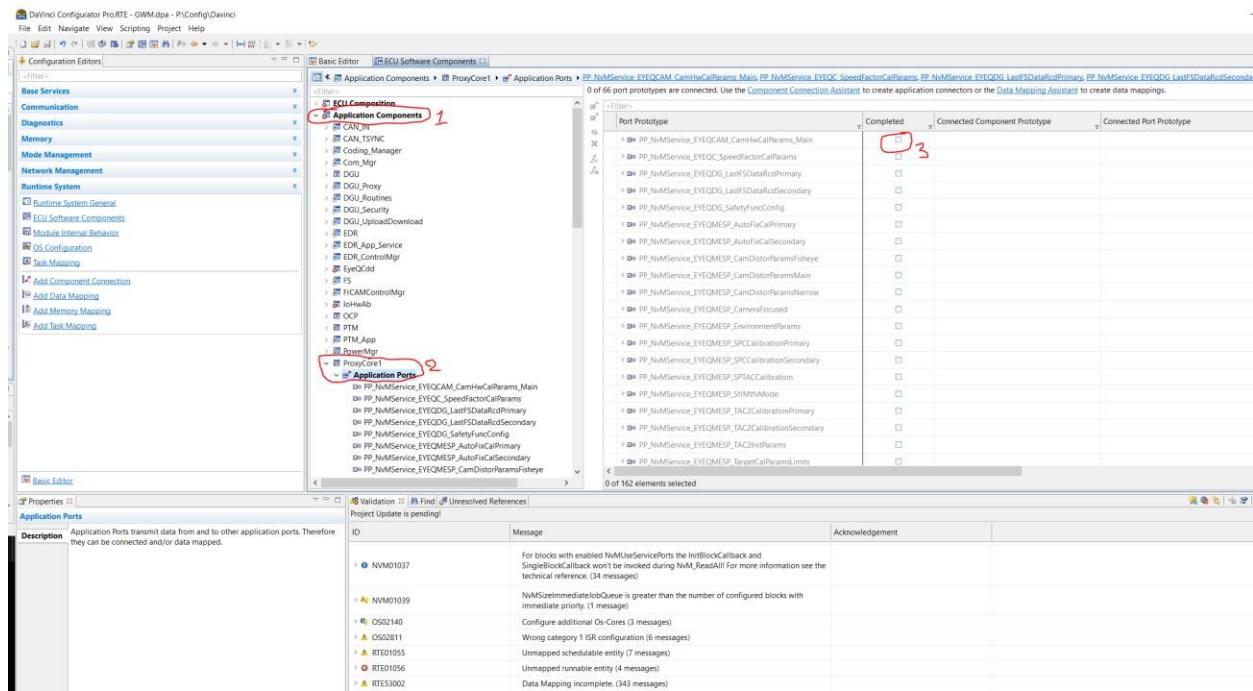
Connections to EyeQ CDD:

Click on Runtime system and then click on Ecu Software components.

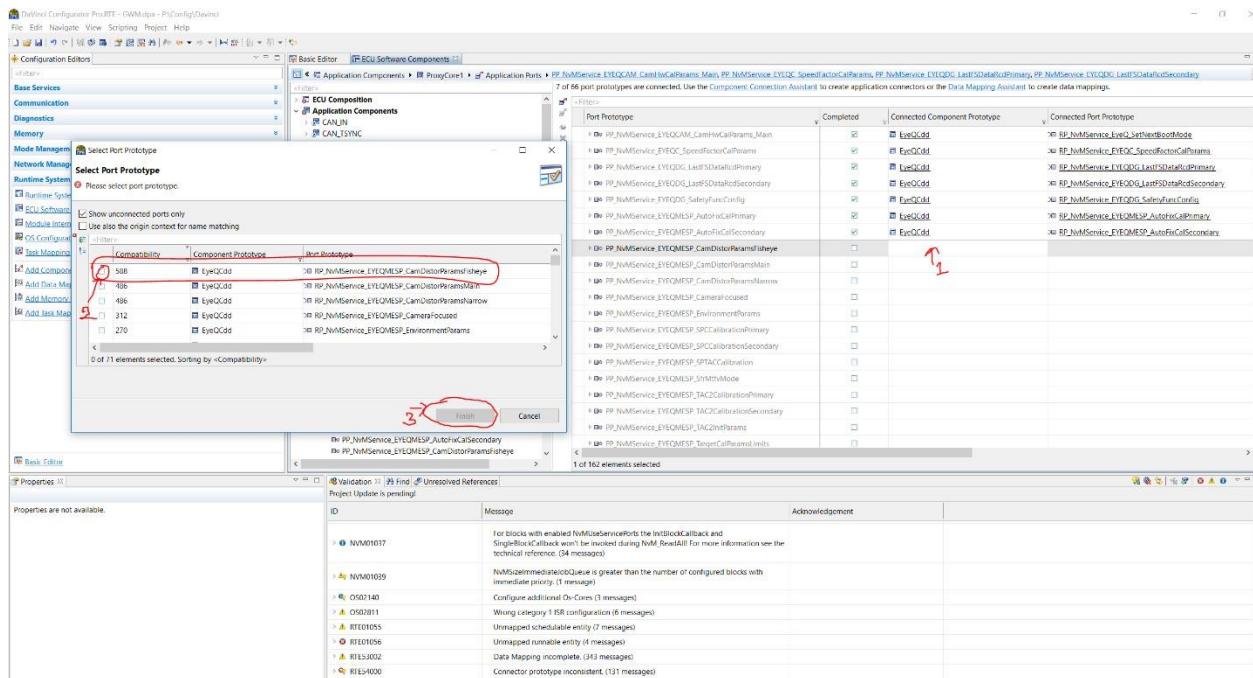


Click on Application components and then click on proxycore1(or proxycore0 depends on architecture) and the application ports and check for the connection status.

EyeQ CDD Integration



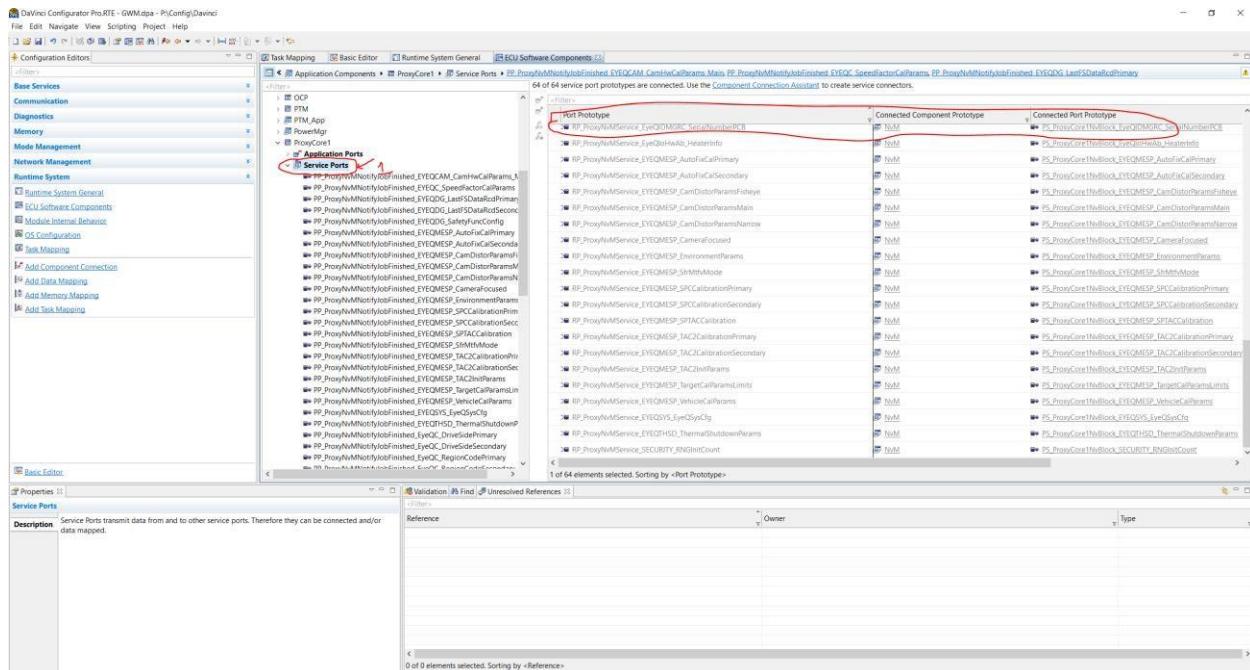
Compare with the port connections provided in core project and do the needful connections. Double click on Connected component prototype and select the port to connect and click on finish.



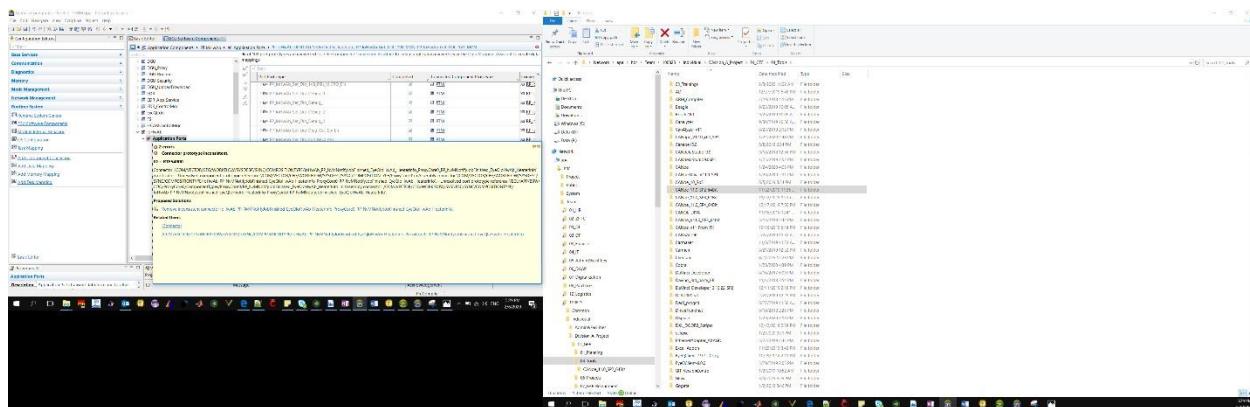
Connections to NvM:

Select Service Ports and if connections are not made do the connections. In figure connections are already made.

EyeQ CDD Integration



Remove inconsistent connectors as shown below(click on the link that is highlighted error will be removed.)



Once NvM Block references are added cross check the length of the blocks with the reference of Core software. Sizes of status blocks of DEM in NvM and admin blocks, primary blocks of dem need to be checked. If any warnings are present integrator should fix them. S37 of NVM need to adapted according to layout(Application Software).

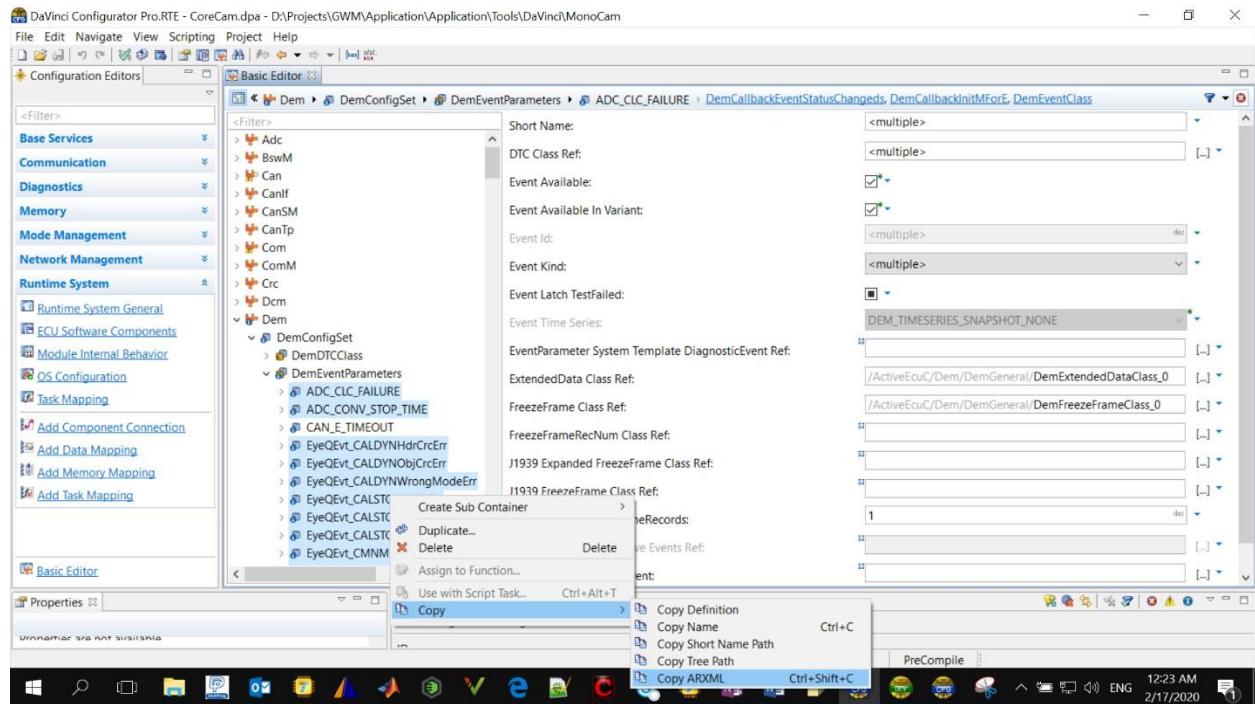
NOTE : If the arxmls(component descriptions) are over written only the effected blocks need to be configured. If component is newly added all NvM blocks need to be configured.

EyeQ CDD Integration

10.3 DEM Integration

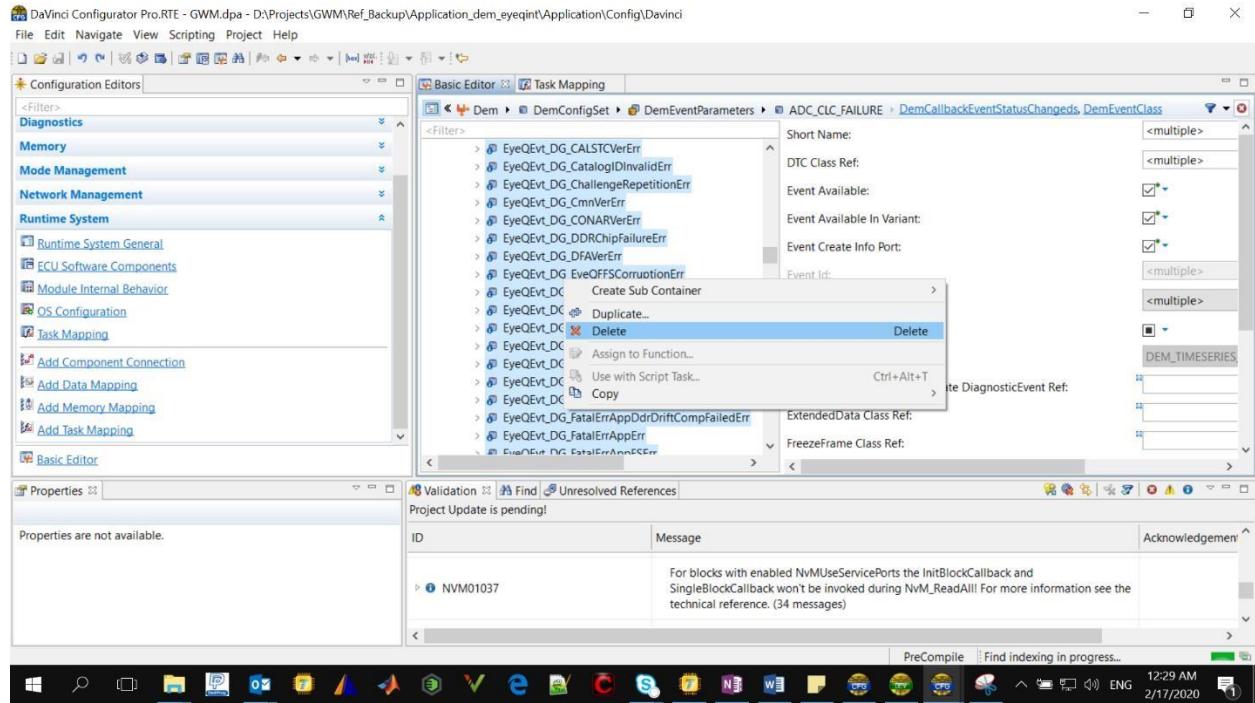
In DEM Integration all the EyeQCDD specific DEM events(Core project) need to be configured in Application project. Adapt the package paths in DEM event configurations from application project. Resolve the configuration errors if exists. Once configuration is completed connect the DEM ports(platform specific).

Copy Core DEM Events from Core project to application project

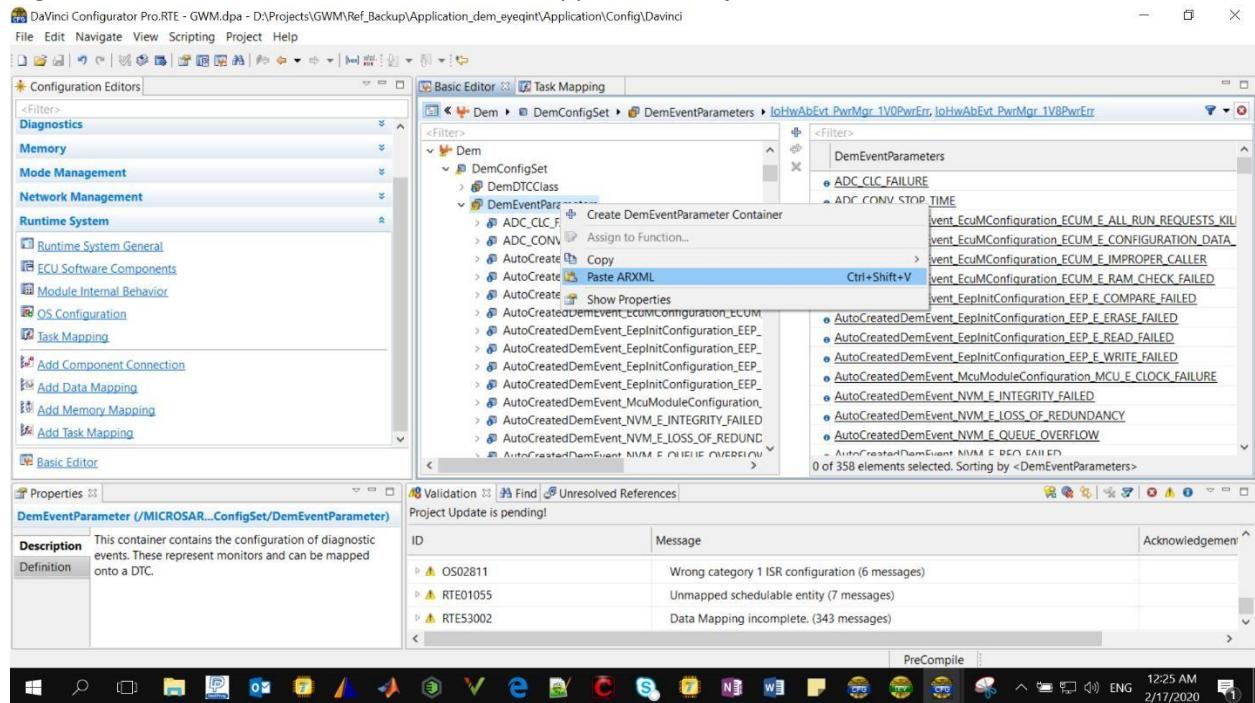


Delete existing Core DEM Events form Application project. For the first time integration this step is not applicable.

EyeQ CDD Integration



Right click on DemEventParameters Paste in Application Project



Adapt DEM configuration from Application software

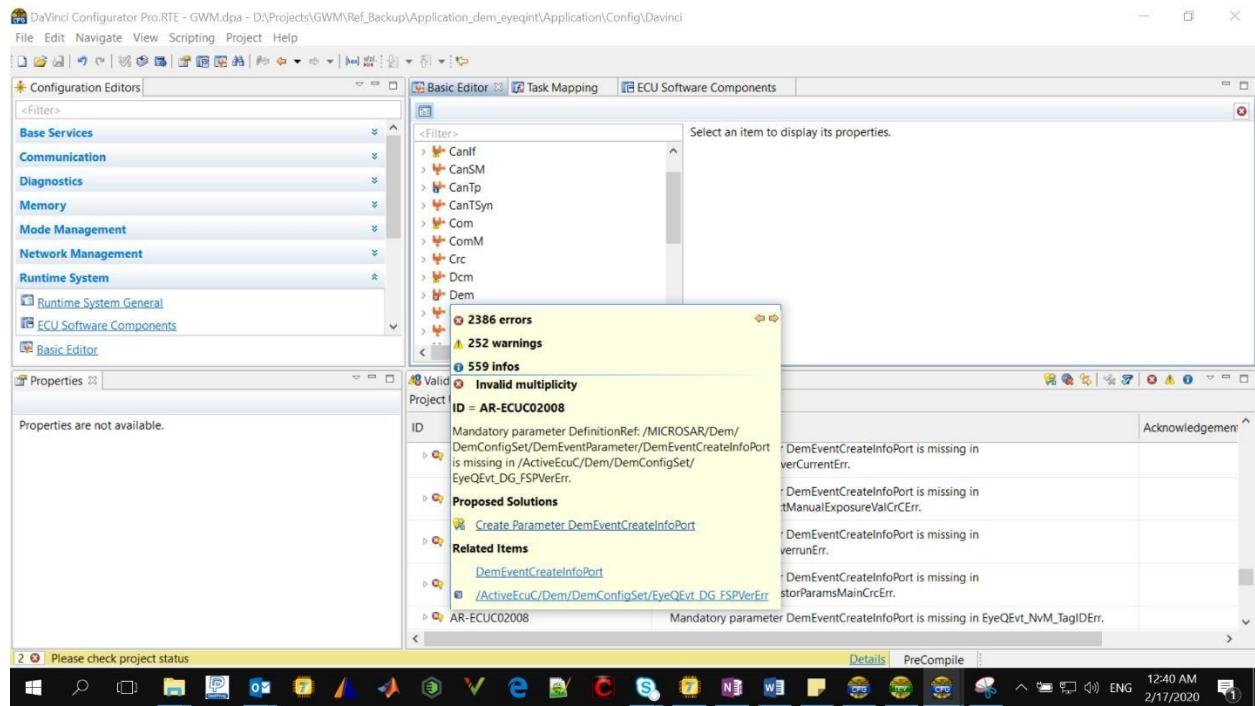
EyeQ CDD Integration

Take one reference DEM event configuration from application adapt highlighted changes into Core DEM events.

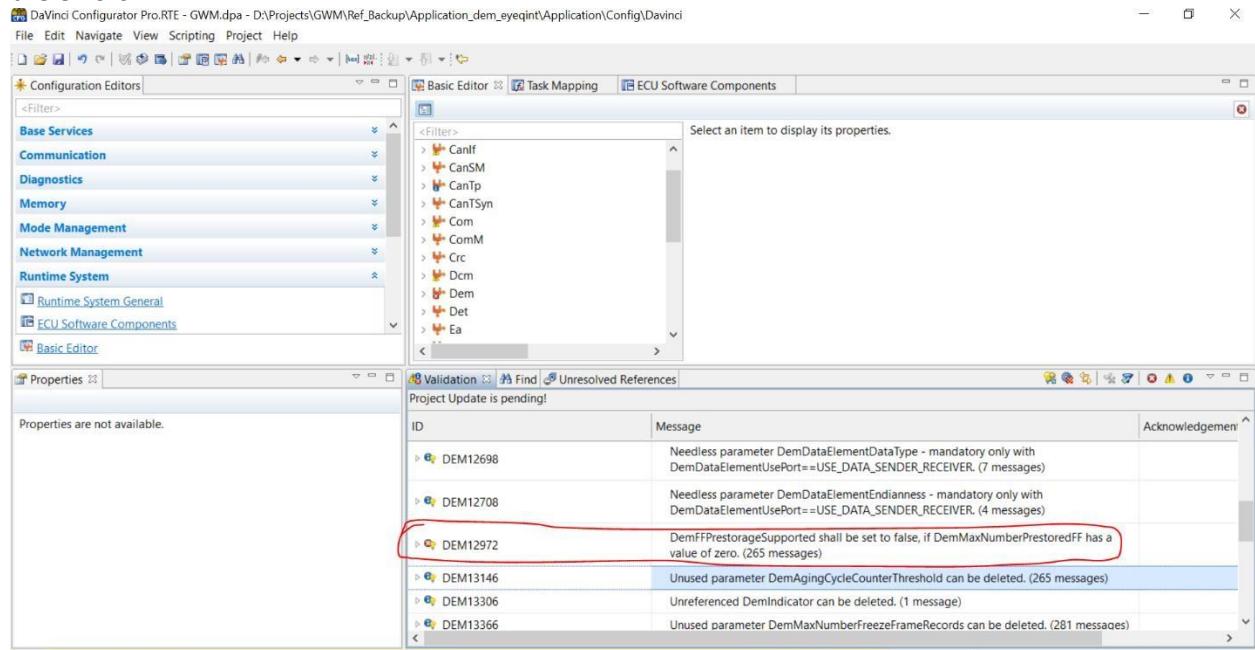
Repeat the steps for all DEM events copied from Core software.

Create a DEM port as shown. Click on Create Parameter DemEventCreateInfoPort(for all DEM Events).

EyeQ CDD Integration

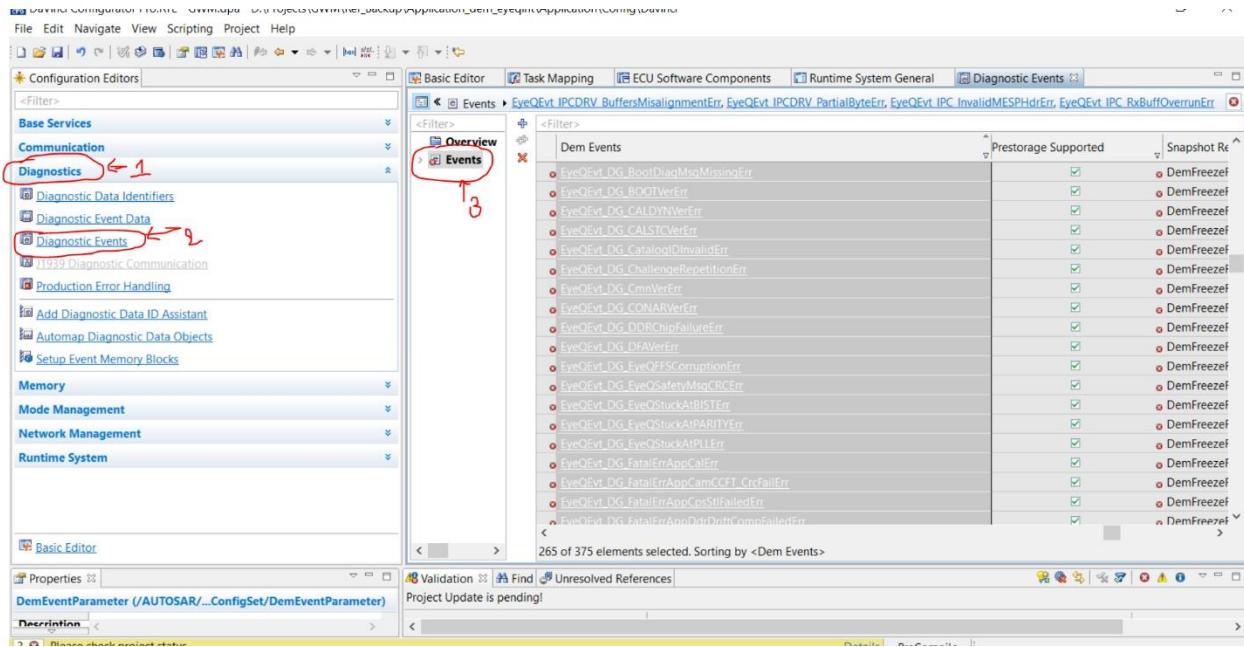


Check other errors related to DEM and resolve the errors Ex:

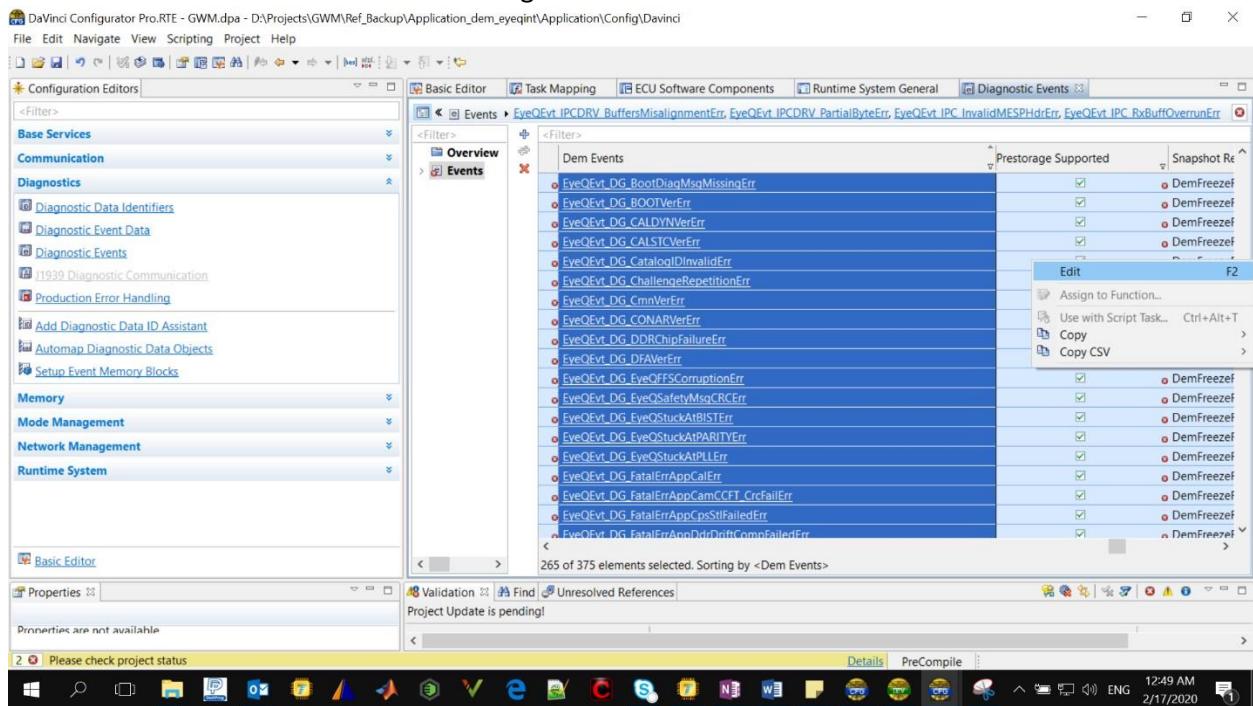


Click on Diagnostics and then select Dem Events as shown

EyeQ CDD Integration



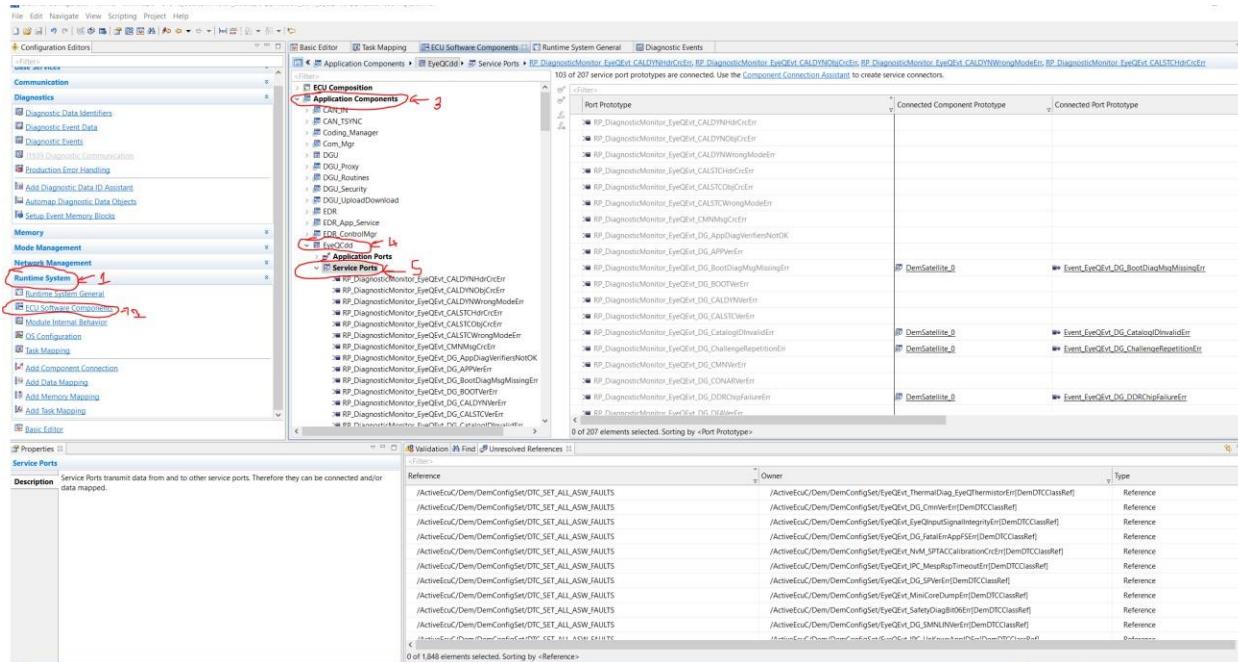
Find the Error field and select events and right click and select Edit.



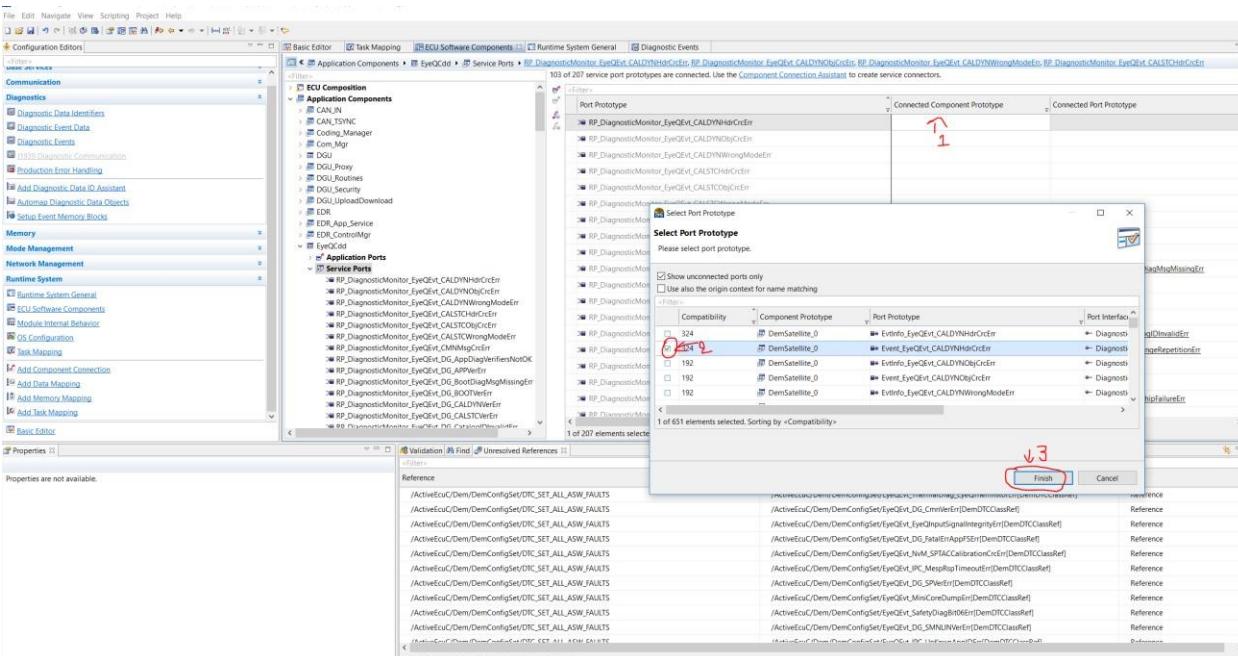
Resolve all the configuration errors.

Navigate through Runtime System->ECU Software Components->Application Components->EyeQCDD>Service Ports

EyeQ CDD Integration

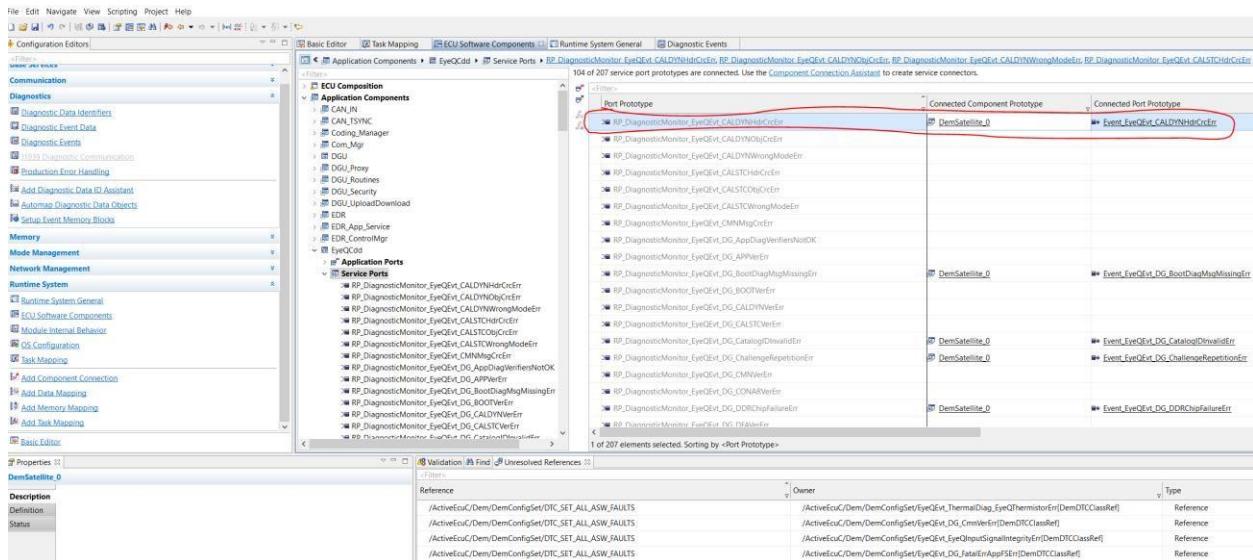


Double click on empty field in connected component prototype column and the select the destination port from opened window and then click on the check box and then click on finish.



Once port is connected connection status will be updated as shown in below figure.

EyeQ CDD Integration



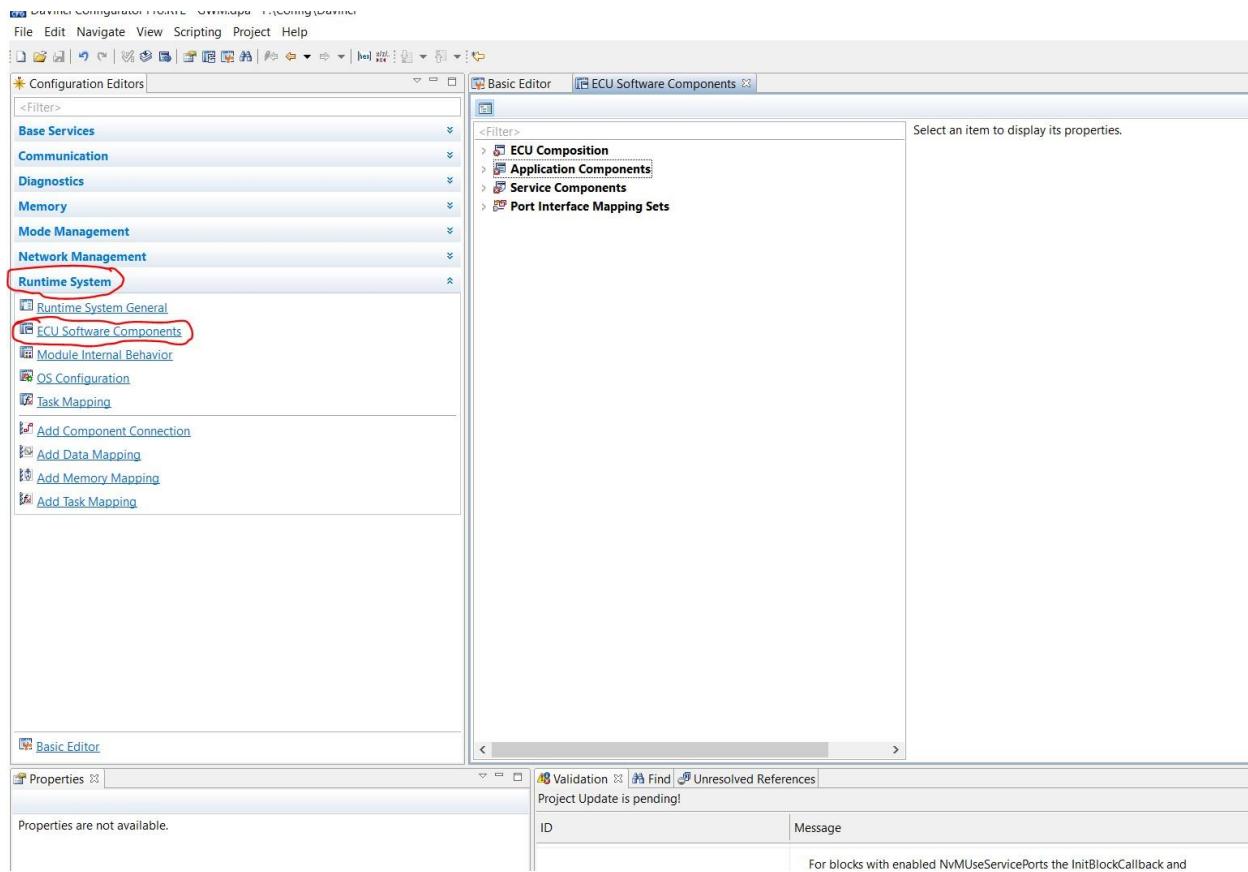
Repeat the same steps for all unconnected DEM ports.

EyeQ CDD Integration

10.4 Integration of Proxy modules

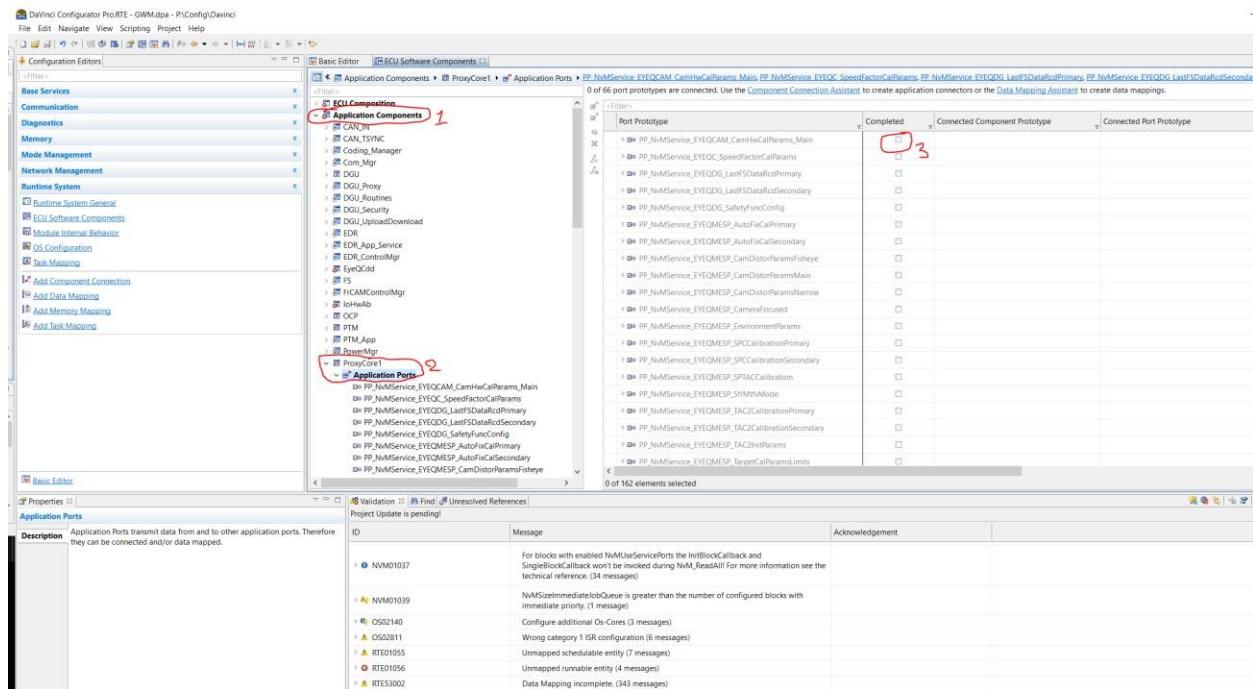
ProxyCore1(or ProxyCore0 depends on architecture) component has application port interfaces and service port interfaces. Application port interfaces need to be connected to destination ports in EyeQCDD component. Service ports need to be connected to NvM and DEM modules(Platform specific).

Click on Runtime system and then click on Ecu Software components.

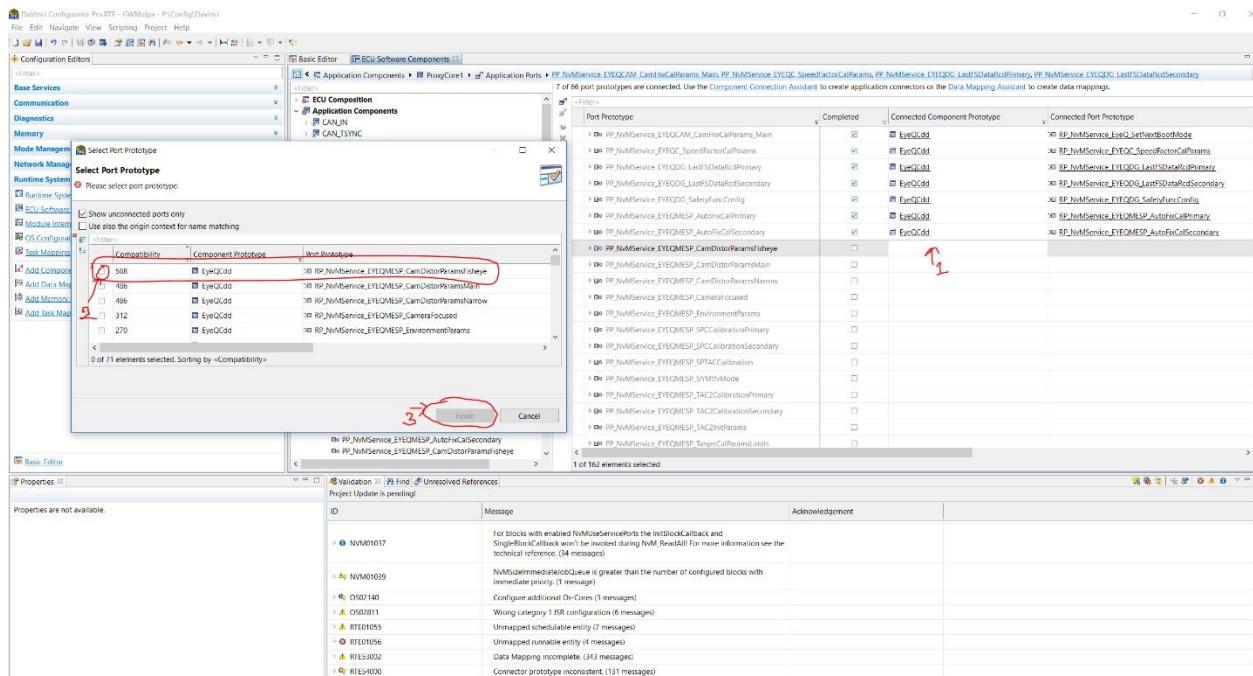


Click on Application components and then click on proxycore1 and the application ports and check for the connection status.

EyeQ CDD Integration

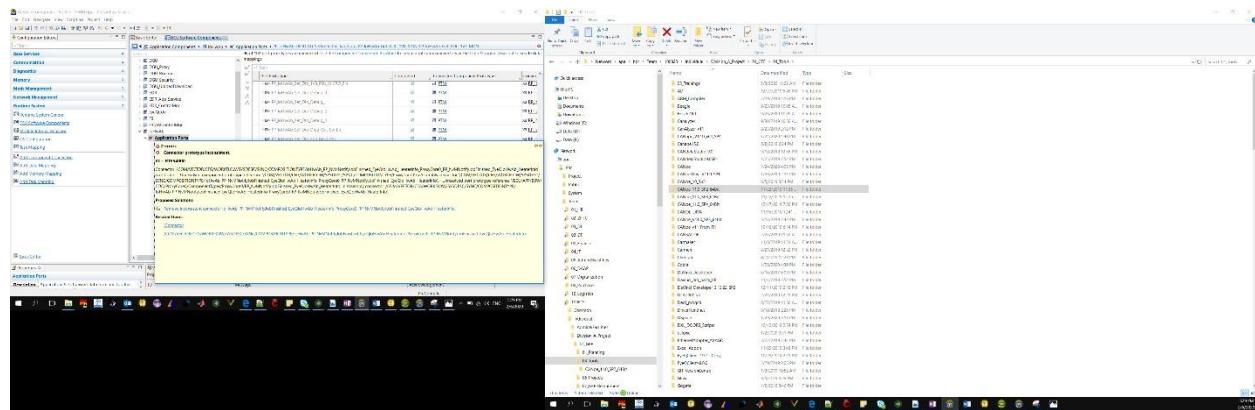


Compare with the port connections provided in core project and do the needful connections. Double click on Connected component prototype and select the port to connect and click on finish.



Remove inconsistent connectors as shown below(click on the link that is highlighted error will be removed.)

EyeQ CDD Integration

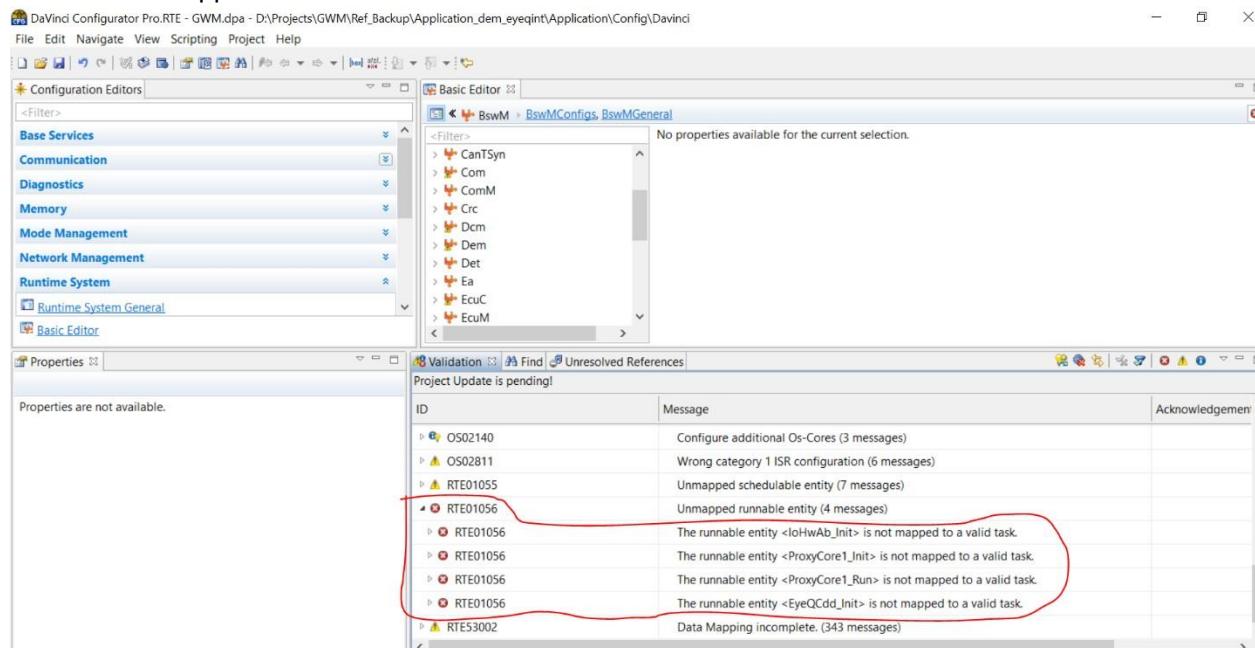


10.5 Os Configuration

In Os configuration the unmapped runnable entities should be mapped to Os tasks.

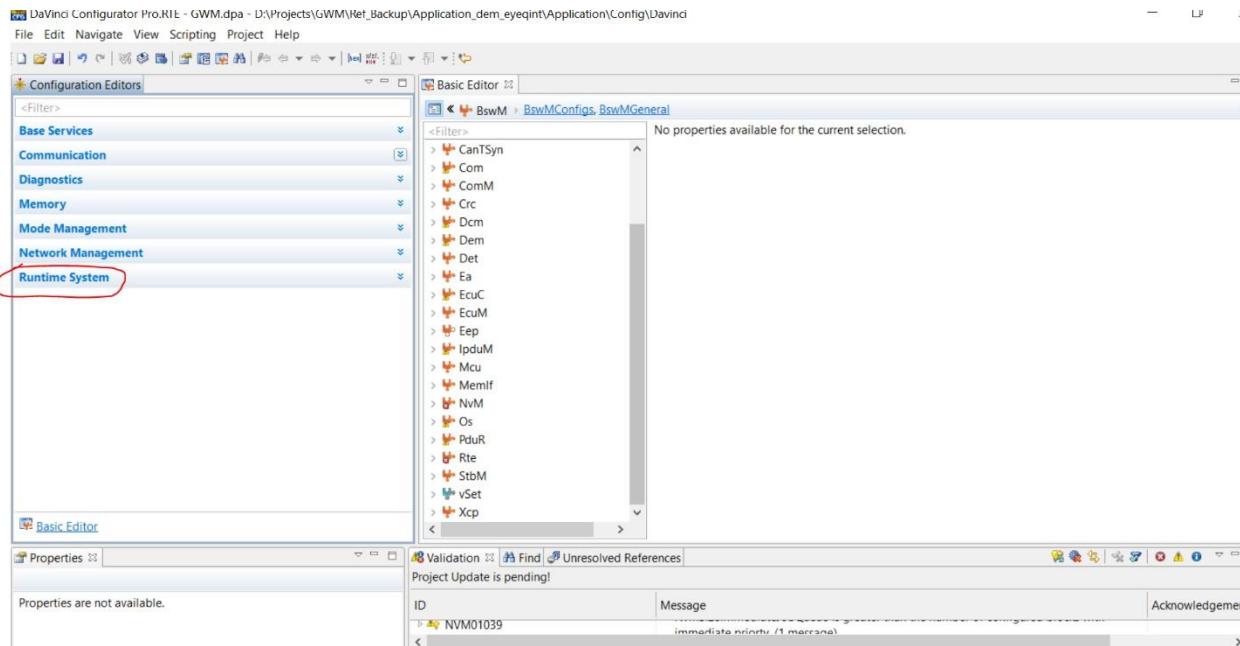
Trusted function and Irq configurations related to EyeQ CDD need to be adapted from Core project. If there are no Os tasks present in Os configuration, create Os tasks and then provide the task reference to events generated by RTE according to component descriptions. Provide an appropriate Os Application reference to the created task.

Check for unmapped runnable`e entities:

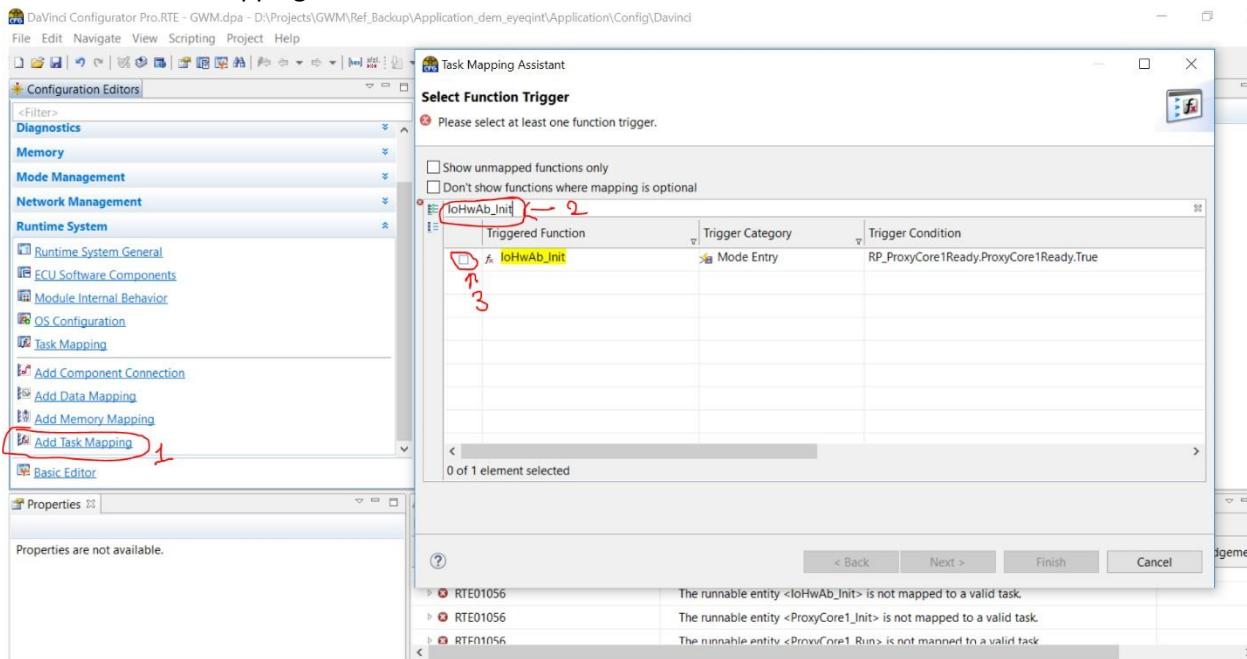


Click on runtime system

EyeQ CDD Integration

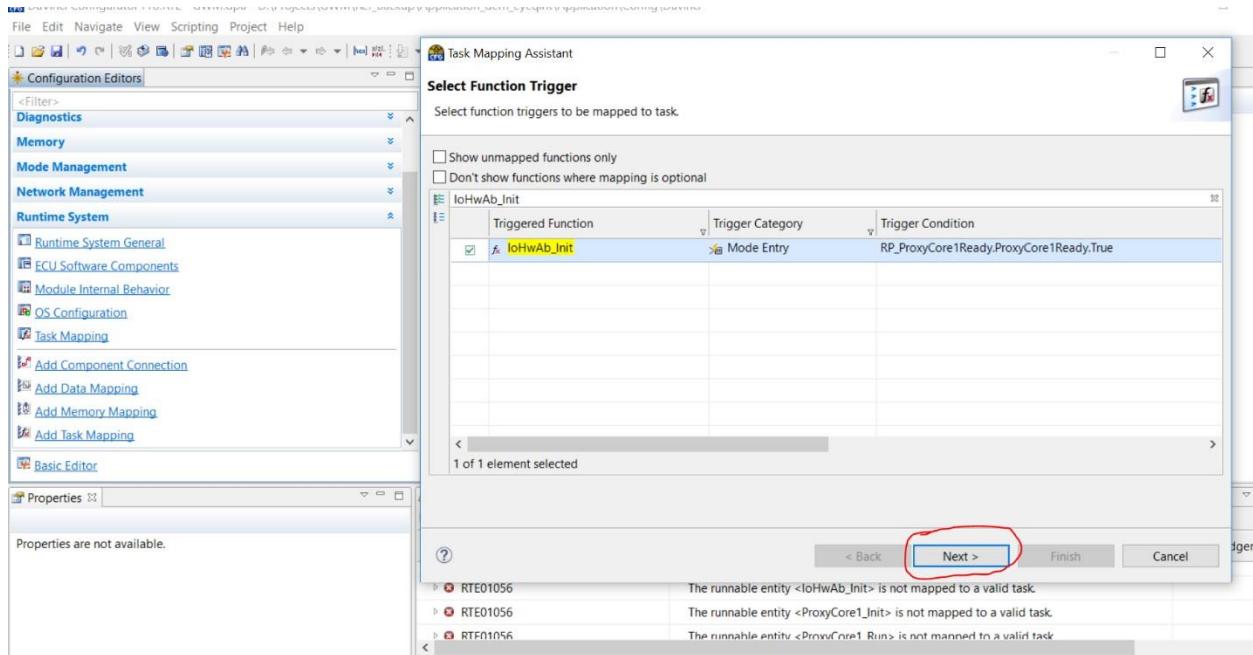


Now select Task Mapping and search for the runnable as shown and click on the check box.

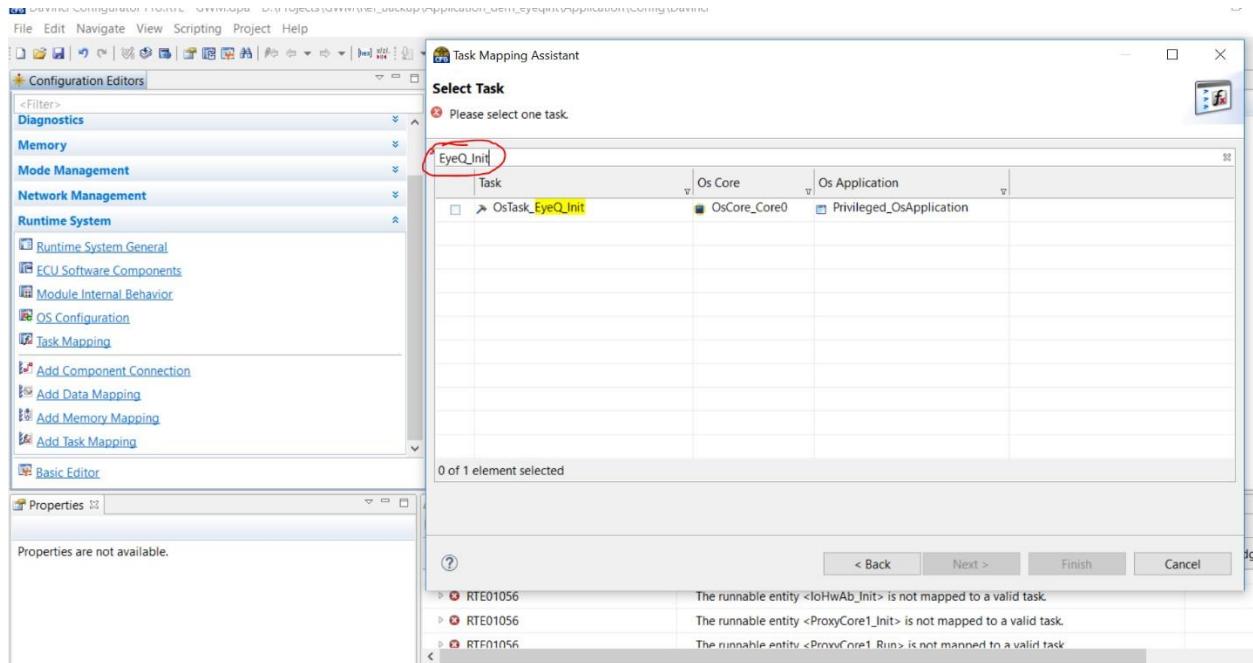


Click on Next:

EyeQ CDD Integration

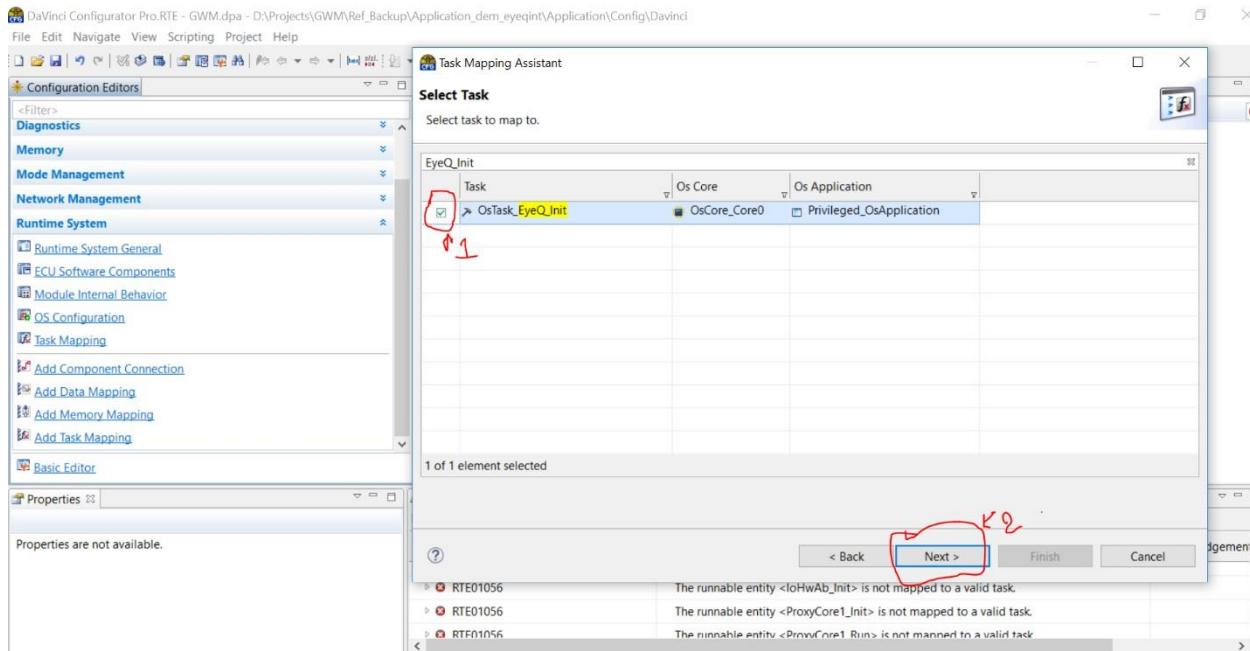


Search for the Os task to map the runnable entity

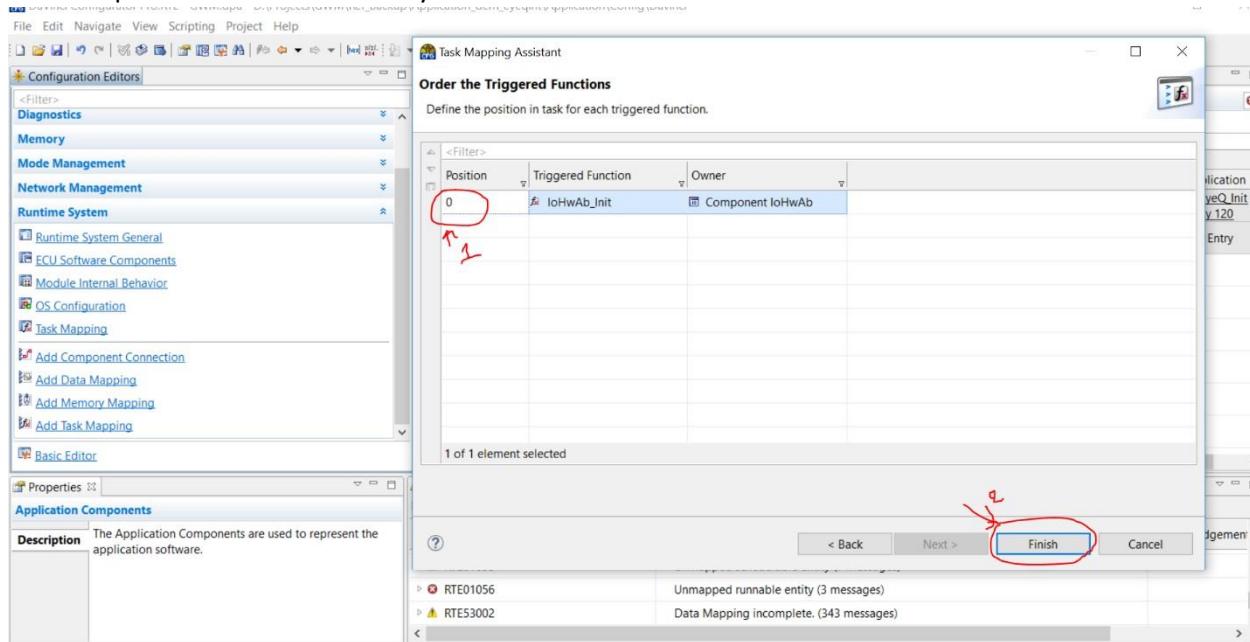


Click on the check box and click next

EyeQ CDD Integration



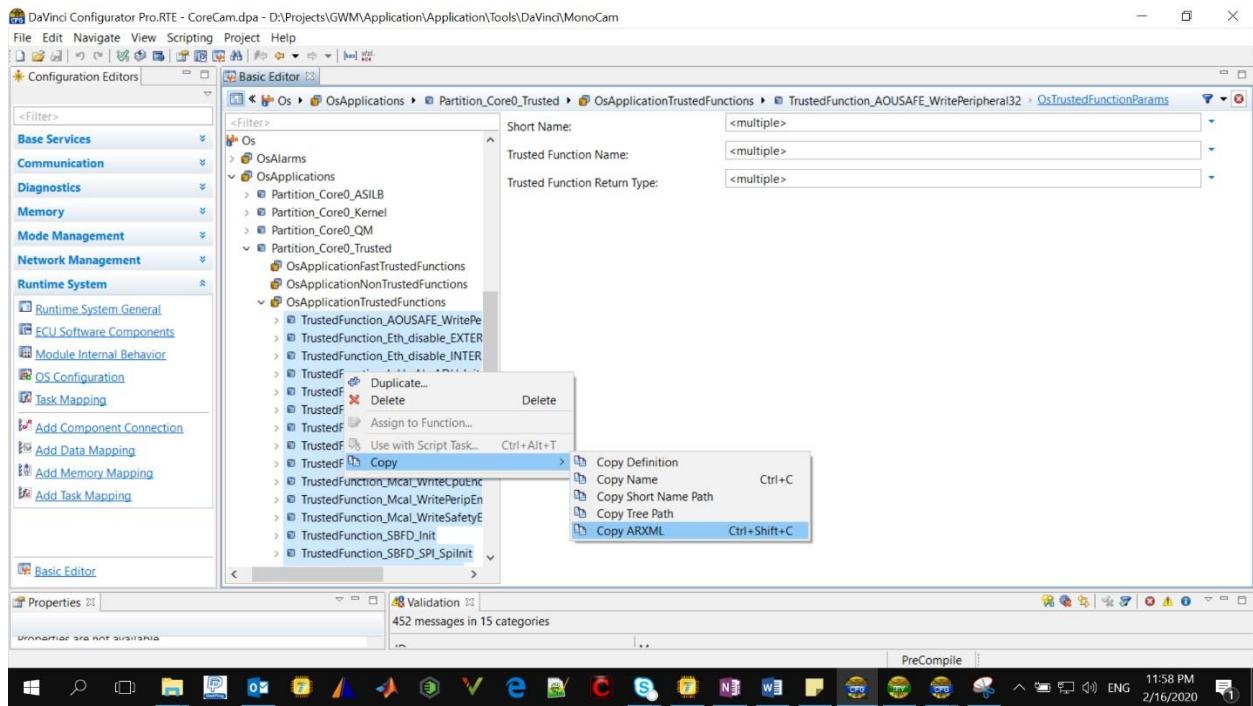
Give the position of runnable entity in the task and click on Finish



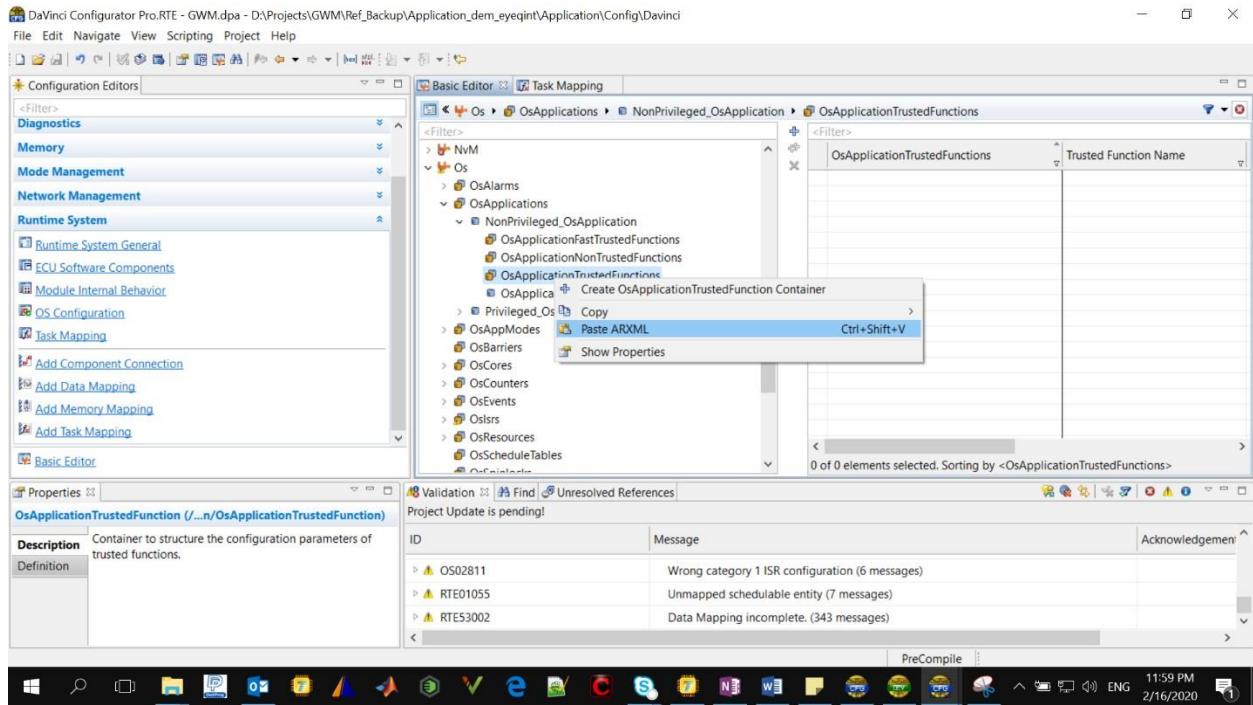
Repeat the same steps for remaining unmapped runnable entities and validate.

Adapt Os trusted function configuration from Core software. Copy the trusted function configuration from core project to application project.

EyeQ CDD Integration



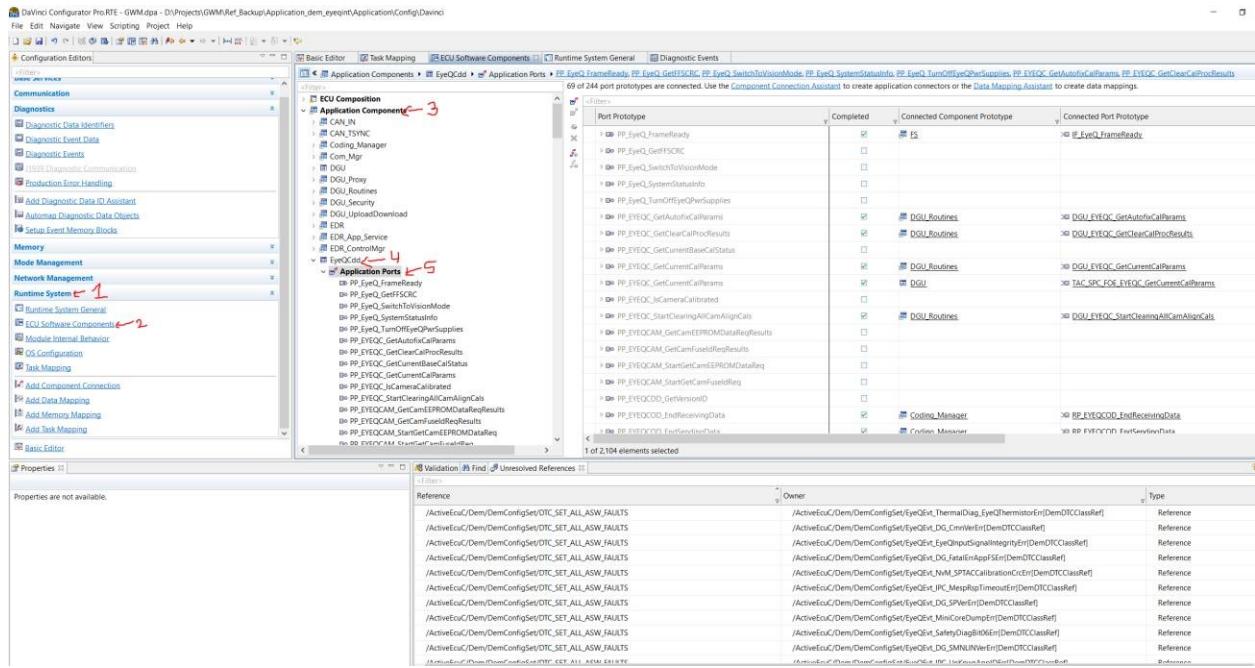
Paste the copied configuration into appropriate Os application in application project.



10.6 CDD Component Integration(Port Connections)

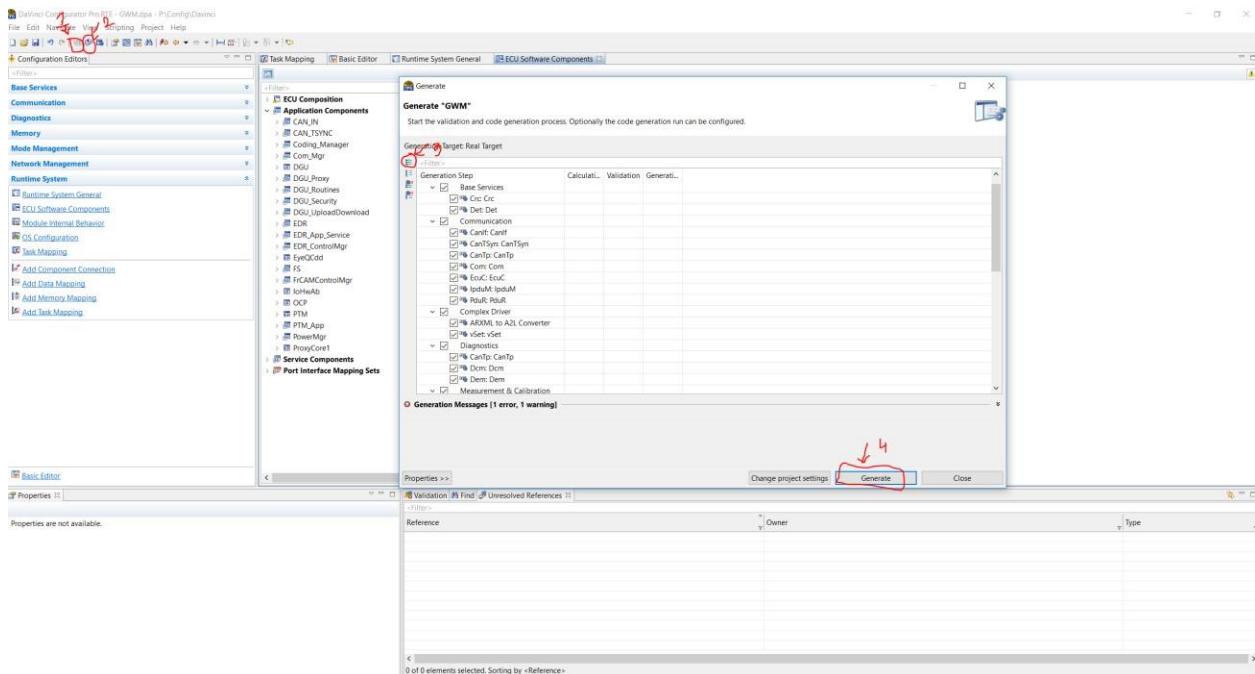
Navigate through Runtime System->ECU Software Components->Application Components->EyeQCDD->Application Ports and do the required connections(Project specific).

EyeQ CDD Integration



10.7 Source Code Generation and Compilation

After resolving all configurator errors click on validate button on the screen. If there are no error's click on Generate button, then select all modules and then generate the files.,



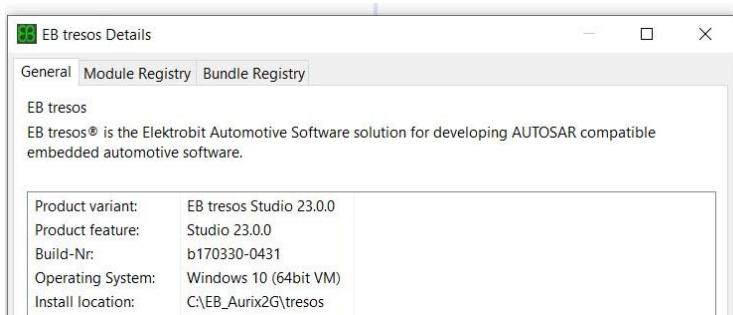
EyeQ CDD Integration

Once the files are generated copy into the destination folder in application software **Source Code**(Project Specific) folder and then go for compilation procedure(Project Specific).

11. MCAL Integration XDM changes and Dynamic code:

Prerequisites:

- Eb Tresos software should be installed below are details of version



- MCAL plugins for respective hardware should be installed ,should check plugins from core release and install respective plugins. Details can be seen in core release documents folder with name “Aurix 2G Tresos and MCAL Plugin installation guide”.

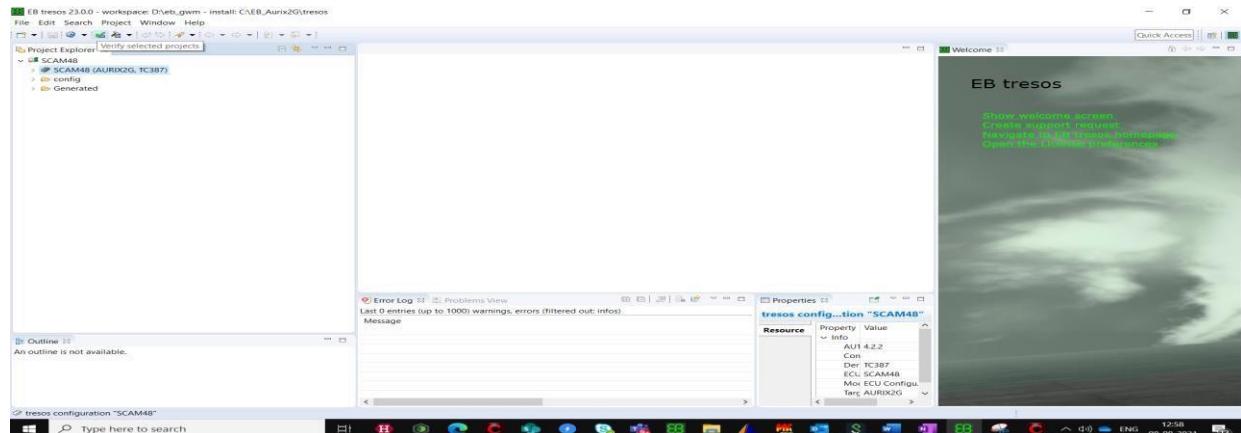
EyeQ CDD Integration

Path: Application\Documents\UserGuide\ Aurix 2G Tresos and MCAL Plugin installation guide core document path.

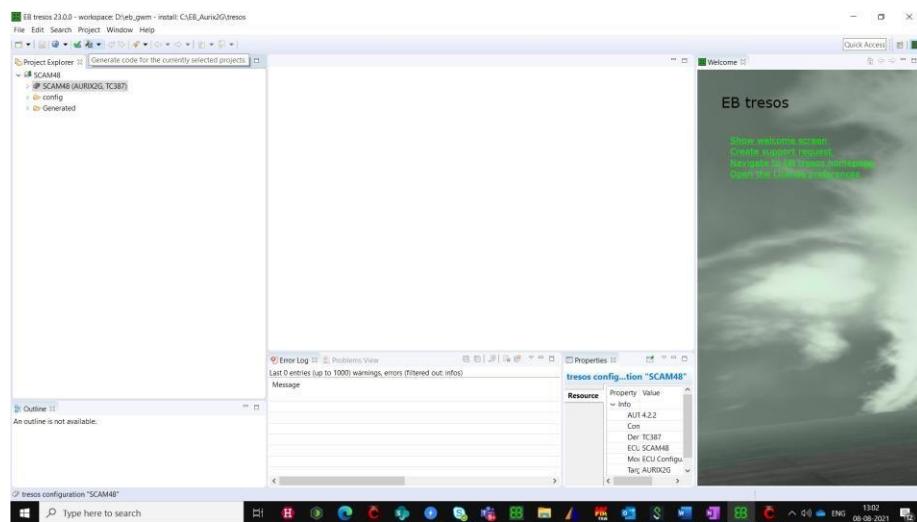
11.1 XDM integration and Dynamic code generation:

When there is new release from core we need to compare MCAL XDM's of core release from core path and application MCAL XDM's. There might be application project specific changes done in MCAL XDM's like Port,DMA,IRQ etc ,this created confusion while comparison. To avoid this we need to compare XDM's of core latest release which we are going to integrate with XDM's of core release which we integrated earlier. This clearly gives new changes from core MCAL. Merge Core XDM's which are changed in new release to application care need to be taken while merging in order not loose application MCAL changes which are project specific.

After XDM changes are done open EB tresos import project and click on verify selection project to validate.

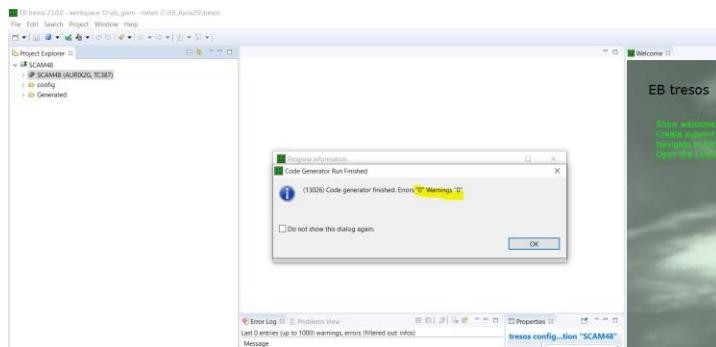


If no errors proceed for code generation as below by clicking on generate code for the current selected projects.

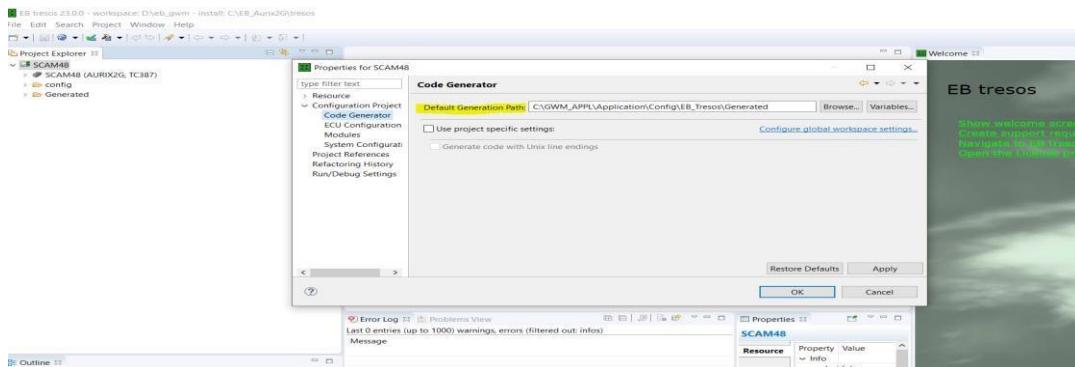


Once code is generated below window pops out.

EyeQ CDD Integration



Code is generated at path given in project options as below.



Now compare code in generated path from EB with application MCAL path "Application\SourceCode\BSW\McalGendata" and merge the code. Also we need to merge code in "Application\Config\EB_Tresos\generated"

11.2 Things to be taken care while check-in MCAL changes :

- 1)MCAL XDM path: Application\Config\EB_Tresos\config should be check-in
- 2)File in path "Application\Config\EB_Tresos\.prefs\ preferences.xdm" contains details of Eb configuration information like modules enabled version, code generation path etc should be check-in.
- 3)Generated code in:
 - a)" Application\SourceCode\BSW\McalGendata"
 - b)" Application\Config\EB_Tresos\generated" should be check-in.

EyeQ CDD Integration

4) Static MCAL changes which we discussed earlier in Static code merging section should be check-in.

12. NVM default values updates:

In GWM and variants NVM defaults values are generally given in NVM .cfg files. Core do not follow this, core use FEE they maintain default values in ROM. When there are any change in default values of NVM block parameters and parameters names this should also be updated in NVM .cfg files of application project to which we are integrating.

Below are ways to find differences in default values in core in latest release. Below are files where all NVM blocks with ROM defaults are defined.

Core_Application\SourceCode\CDD\EyeQ_CDD\EyeQ_NVM\src\EyeQ_NvM.c

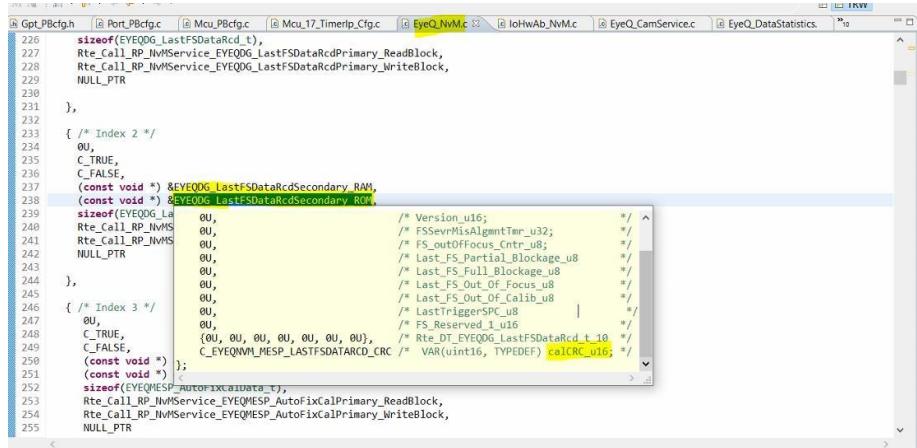
Application\SourceCode\IoHwAbs\IoHwAb_NvM\src\IoHwAb_NvM.c

There are two structures each for EyeQ blocks and one for IoHwAB blocks defined as below. In EyeQ_NvM.c and IoHwAb_NvM.c files there is structure member `/*
Rom Block location */ const void* RomBlock_p;`.

EyeQ CDD Integration

```
/* Structure to hold static block configuration data */  
typedef struct  
{  
    /* NvM TagId */  
    uint16_t TagID_u16;  
    /* TagId Error */  
    uint8_t IsSafeStateReqdForTagIDErr_u8;  
    /* CRC Error */  
    uint8_t IsSafeStateReqdForCRCErr_u8;  
    /* RamBlock location*/  
    const void* RamBlock_p;  
    /* Rom Block location */  
    const void* RomBlock_p;  
    /* Size of the RamBlock */  
    uint32_t RamBlockSize_u32;  
    /* Function to read NvData from lower layer */  
    ReadBlock_t ReadBlock_pf;  
    /* Function to write NvData to lower layer */  
    WriteBlock_t WriteBlock_pf;  
    /* Function to set the DEM event */  
    SetCRCErrorEvent_t SetCRCErrorEvent_pf;  
  
}EyeONvM_BlockConfiguration_t;  
  
/* Structure to hold static block configuration data */  
typedef struct  
{  
    /* NvM TagId */  
    uint16_t TagID_u16;  
    /* TagId Error */  
    uint8_t IsSafeStateReqdForTagIDErr_u8;  
    /* CRC Error */  
    uint8_t IsSafeStateReqdForCRCErr_u8;  
    /* RamBlock location*/  
    const void* RamBlock_p;  
    /* Rom Block location */  
    const void* RomBlock_p;  
    /* Size of the RamBlock */  
    uint32_t RamBlockSize_u32;  
    /* Function to read NvData from lower layer */  
    ReadBlock_t ReadBlock_pf;  
    /* Function to write NvData to lower layer */  
    WriteBlock_t WriteBlock_pf;  
    /* Function to set the DEM event */  
    SetCRCErrorEvent_t SetCRCErrorEvent_pf;  
  
}IoHwAbNvM_BlockConfiguration_t;
```

Example to find changes in default values: In below screen shot we can find EYEQDG_LastFSDataRcdSecondary blocks ROM member which defines default values of member of blocks ,the only way to find if any update in core is compare with earlier release of core default value.



```
226     sizeof(EYEQDG_LastFSDataRcd_t),  
227     Rte_Call_RP_NvMServc_EYEQDG_LastFSDataRcdPrimary_ReadBlock,  
228     Rte_Call_RP_NvMServc_EYEQDG_LastFSDataRcdPrimary_WriteBlock,  
229     NULL_PTR  
230 },  
231 },  
232 { /* Index 2 */  
233     0U,  
234     C_TRUE,  
235     C_FALSE,  
236     (const void *) &EYEQDG_LastFSDataRcdSecondary_ROM,  
237     (const void *) 2*EYEQDG_LastFSDataRcdSecondary_ROM,  
238     sizeof(EYEQDG_La 0U, /* Version_u16; */  
239     Rte_Call_RP_NvMS 0U, /* FSservMisAlignmTmr_u32; */  
240     Rte_Call_RP_NvMS 0U, /* FS_outOfFocus_Cntr_u8; */  
241     NULL_PTR 0U, /* Last_F5_Partial_Blockage_u8 */  
242     0U, /* Last_F5_Full_Blockage_u8 */  
243     0U, /* Last_F5_Out_Of_Focus_u8 */  
244     0U, /* Last_F5_Out_Of_Calib_u8 */  
245     0U, /* LastTriggerSPC_u8 */  
246     { /* Index 3 */  
247         0U, /* FS_Reserved_1_u16 */  
248         C_TRUE,  
249         C_FALSE,  
250         {0U, 0U, 0U, 0U, 0U, 0U}, /* Rte_DT_EYEQDG_LastFSDataRcd_t_19 */  
251         C_EYEQNV_MESP_LASTFSDATARCD_CRC /* VAR(uint16, TYPEDEF) colCRC_u16; */  
252     };  
253     sizeof(EYEQESP_AutoFixCalData_u),  
254     Rte_Call_RP_NvMServc_EYEQESP_AutoFixCalPrimary_ReadBlock,  
255     Rte_Call_RP_NvMServc_EYEQESP_AutoFixCalPrimary_WriteBlock,  
256     NULL_PTR
```

Similarly for IoHwAb blocks we need to find any changes in ROM member default values and updating NVM .cfg files in application .We can also find any changes in members of structure elements names by comparing Application\SourceCode\BSW\Generated\MonoCam\Rte_Type.h

EyeQ CDD Integration

```
④ EyeQ_NvM.c ⑤ Rte_Type.h ⑥ Rte_EyeQcdd.h ⑦ Rte_EyeQcdd_Type.h ⑧ Rte_ProxyCore1.h ⑨ Rte_ProxyCore1_Type.h ⑩ EyeQCdd_MemMap.h ⑪ Rte_Type.h
1192 #include "Rte_EyeQcdd.h"
1193 #include "Rte_ProxyCore1.h"
1194 #include "EyeQCdd_MemMap.h"
1195
1196 #define Rte_TypeDef_EYEQ0G_LastFSDataRcd_t
1197 typedef struct
1198 {
1199     uint16 TagID_u16;
1200     uint16 Version_u16;
1201     uint32 FSSevrflsAIgmtTmr_u32;
1202     uint8 F5_outOfFocus_Cntr_u8;
1203     uint8 Last_FS_Partial_Blockage_u8;
1204     uint8 Last_FS_Full_Blockage_u8;
1205     uint8 Last_FS_CalibStatus_u8;
1206     uint8 Last_FS_Out_of_Calib_u8;
1207     uint8 LastTriggerSPC_u8;
1208     uint16 FS_Reserved_1_u16;
1209     uint16 FS_Reserved_2_u16;
1210     Rte_DT_EYEQ0G_LastFSdataRcd_t_10 Reserved3_u16;
1211     uint16 CalcCRC_u16;
1212 } EYEQ0G_LastFSdataRcd_t;
1213
1214 #define Rte_TypeDef_EYEQ0G_SafetyFuncConfig_t
1215 typedef struct
1216 {
1217     uint16 TagID_u16;
1218     uint16 Version_u16;
1219     uint8 IsGeneralSafetyCritical_u8;
1220     uint8 IsAEBSafetyCritical_u8;
1221     uint8 TrigEnc_LockoutCounter_u8;
1222 } EYEQ0G_SafetyFuncConfig_t;
```

13. SConstruct and linker changes

13.1 SConstruct integration:

Gwm and variants use SConstruct as build system unlike core which use make build system. We need to identify changes like file name changes in core,new files added or files deleted and also compiler options etc changes in make of core and adapt them SConstruct of application . Care should be taken and we need to discuss with architect regarding any compiler options changes between core before changing and adapting in application complier options .

Path core Make files: Core_Application/Build/Make

Path GWM application SConstruct file: Application\Build\ SConstruct

13.2 Linker changes:

For core related linker changes we need to follow document

EyeQ CDD Integration

DAS_Core_Software_EyeQ_CDD_IntegrationGuide and compare linker files of core with previous release at core_Application\Build\Make like CoreApplicationLinker.lsl, tc3xx, trptabx etc.

Before adapting linker changes in application related linker files detail discussion with architect should be done and impact of changes if any on application should be analyzed.

For placement of core buffers and file related in particular section we need to strictly follow DAS_Core_Software_EyeQ_CDD_IntegrationGuide document in core release.

Path: Application\Documents\IntegrationGuide\CDD\
DAS_Core_Software_EyeQ_CDD_IntegrationGuide

14.General Issues, debugging steps and validating EyeQ integration.

In this chapter we will discuss about validating after EyeQ integration, general issues and debugging steps.

14.1 Validating EyeQ integration:

After EyeQ integration build is successful we need to flash respective compatible ME files using MFT, flash vision file of NVM(in GWM we will move EyeQ into vision by NVM vision file ,we can also move EyeQ into vision using PTM and debugger) .

Below are variables to be checked :

- 1) EYEQDG_CurrentMainState_u8: This variable gives EyeQ states if it is in vision or some other state. Below are some of states of EyeQ. If everything is as expected THEN EYEQDG_CurrentMainState_u8 should be 2 that is vision .

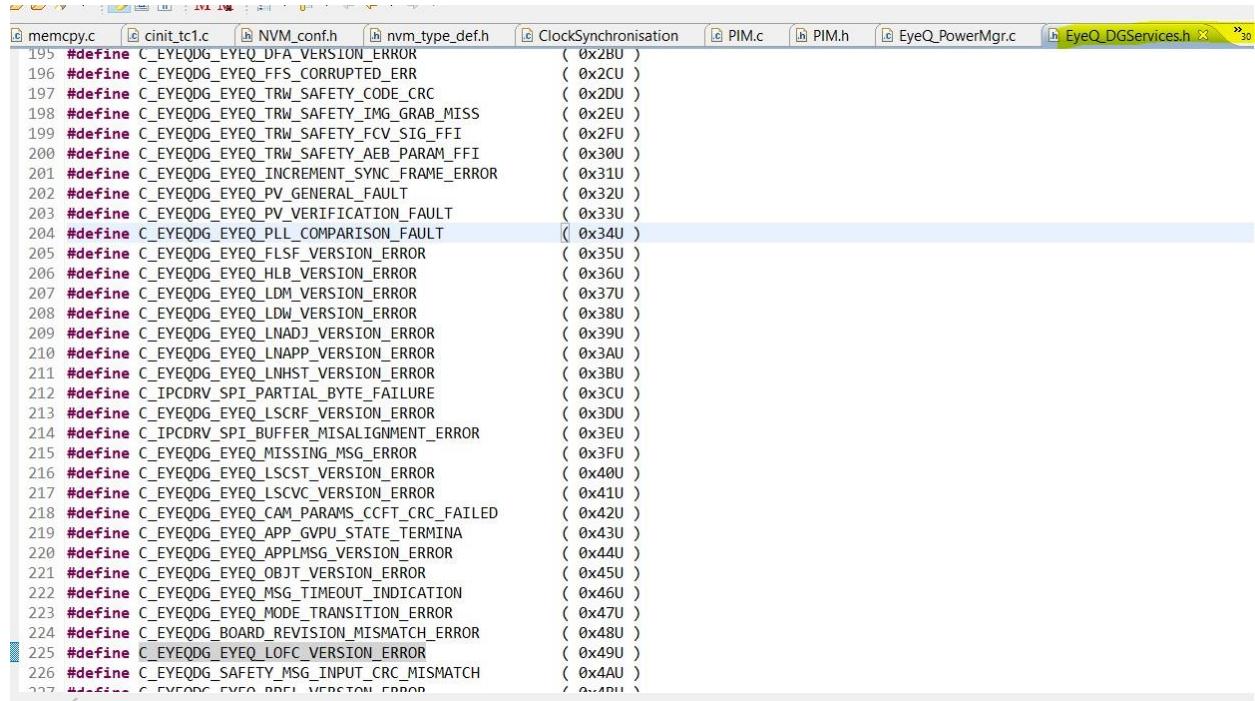
EyeQ CDD Integration

- #define C_EQBS_EQ_MAIN_STATE_UNKNOWN (0x00U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING (0x01U)
- #define C_EQBS_EQ_MAIN_STATE_BOOT (0x03U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_DV (0x85U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_APP_BURN (0xAAU)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_VISION (0x92U)
- #define C_EQBS_EQ_MAIN_STATE_VISION (0x02U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_SPC (0x82U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_SPC (0x22U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_SPTAC (0x81U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_SPTAC (0x21U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_TAC (0x80U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_TAC (0x20U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_TAC2 (0xABU)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_TAC2 (0xACU)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_SFR (0x83U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_SFR (0xB0U)
- #define C_EQBS_EQ_MAIN_STATE_PENDING_SP (0x89U)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_SP (0xAEU)
- #define C_EQBS_EQ_MAIN_STATE_RUNNING_STEREO (0xAFU) • #define C_EQBS_EQ_STATE_UNAVAILBLE (0xFFU)

- 2) EYEQP_CurrentErrorStatus_u32 :This variable value should be '0' if there are no errors reported in EyeQ. If any errors is logged to know the cause we need to refer file
EyeQ_DGServices.h at

EyeQ CDD Integration

Application\Core_Comp\CDD\EyeQ_CDD\EyeQ_DG/include/EyeQ_DGServices.h this file gives detail of error which is logged as below.



```
195 #define C_EYEQDG_EQEQ_DFA_VERSION_ERROR          ( 0x2BU )
196 #define C_EYEQDG_EQEQ_FFS_CORRUPTED_ERR          ( 0x2CU )
197 #define C_EYEQDG_EQEQ_TRW_SAFETY_CODE_CRC        ( 0x2DU )
198 #define C_EYEQDG_EQEQ_TRW_SAFETY_IMG_GRAB_MISS   ( 0x2EU )
199 #define C_EYEQDG_EQEQ_TRW_SAFETY_FCV_SIG_FFI       ( 0x2FU )
200 #define C_EYEQDG_EQEQ_TRW_SAFETY_AEB_PARAM_FFI    ( 0x30U )
201 #define C_EYEQDG_EQEQ_INCREMENT_SYNC_FRAME_ERROR ( 0x31U )
202 #define C_EYEQDG_EQEQ_PV_GENERAL_FAULT           ( 0x32U )
203 #define C_EYEQDG_EQEQ_PV_VERIFICATION_FAULT      ( 0x33U )
204 #define C_EYEQDG_EQEQ_PLL_COMPARISON_FAULT        ( 0x34U )
205 #define C_EYEQDG_EQEQ_FLSF_VERSION_ERROR          ( 0x35U )
206 #define C_EYEQDG_EQEQ_HLB_VERSION_ERROR           ( 0x36U )
207 #define C_EYEQDG_EQEQ_LDM_VERSION_ERROR           ( 0x37U )
208 #define C_EYEQDG_EQEQ_LDW_VERSION_ERROR           ( 0x38U )
209 #define C_EYEQDG_EQEQ_LNADJ_VERSION_ERROR         ( 0x39U )
210 #define C_EYEQDG_EQEQ_LNAPP_VERSION_ERROR         ( 0x3AU )
211 #define C_EYEQDG_EQEQ_LNHST_VERSION_ERROR         ( 0x3BU )
212 #define C_IPCDRV_SPI_PARTIAL_BYTE_FAILURE        ( 0x3CU )
213 #define C_EYEQDG_EQEQ_LSCRF_VERSION_ERROR         ( 0x3DU )
214 #define C_IPCDRV_SPI_BUFFER_MISALIGNMENT_ERROR    ( 0x3EU )
215 #define C_EYEQDG_EQEQ_MISSING_MSG_ERROR           ( 0x3FU )
216 #define C_EYEQDG_EQEQ_LSCST_VERSION_ERROR         ( 0x40U )
217 #define C_EYEQDG_EQEQ_LSCVC_VERSION_ERROR         ( 0x41U )
218 #define C_EYEQDG_EQEQ_CAM_PARAMS_CCFT_CRC_FAILED ( 0x42U )
219 #define C_EYEQDG_EQEQ_APP_GVPU_STATE_TERMINA     ( 0x43U )
220 #define C_EYEQDG_EQEQ_APPLMSG_VERSION_ERROR       ( 0x44U )
221 #define C_EYEQDG_EQEQ_OBJT_VERSION_ERROR          ( 0x45U )
222 #define C_EYEQDG_EQEQ_MSG_TIMEOUT_INDICATION     ( 0x46U )
223 #define C_EYEQDG_EQEQ_MODE_TRANSITION_ERROR       ( 0x47U )
224 #define C_EYEQDG_BOARD_REVISION_MISMATCH_ERROR    ( 0x48U )
225 #define C_EYEQDG_EQEQ_LOFC_VERSION_ERROR          ( 0x49U )
226 #define C_EYEQDG_EQEQ_SAFETY_MSG_INPUT_CRC_MISMATCH ( 0x4AU )
227 #define C_EYEQDG_EQEQ_EQEQ_VERSION_ERROR          ( 0x4BU )
```

3) EyeQLastResetSource_u8: This variable gives last error source which has reset EyeQ . We can refer EyeQDGServices.h to find details. Ideally value of variable should be 0xFF.

4) EyeQ_RestartAttemptCnt_u16: This variable gives number of resets attempts by EyeQ if any error is reported and if reset strategy is available for respective error. For reset strategy refer

Application\Core_Comp\CDD\EyeQ_CDD\EyeQ_MGR\src\EyeQ_PowerMgr.c which has reset table with details. Ideally the value of this variable should be 0.

5) EYEQBS_Versions_s: This structure will give information of ME flashed, boot version of ME, MEST version etc as below example.

EyeQ CDD Integration

```
EYEQBS_Versions_s = {
    .VerRet_u32 = 0x0,
    .VersLen_u32 = 0x5,
    .BootMgrVer_u32 = 0x08070200,
    .BootLdrVer_u32 = 0x08070200,
    .MESTVer_u32 = 0x140A0502,
    .ProtocolVer_u32 = 0x140A0C08,
    .EtcVer_u32 = 0x00050002)
    .EyeQ_PostStartAttemptsCnt.u16 = 0
```

6) EYEQDG_RxMsgTracker_s: This structure gives details of various messages received by EyeQ CDD using IPC communication this helps in debugging any timeout issue etc.

7) EYEQP_EyeQResetTracker_as: This structure gives details of any resets logged by EyeQ .To refer particular error we need to refer reset table(ResetTable_s) in

Application\Core_Comp\CDD\EyeQ_CDD\EyeQ_MGR\src\EyeQ_PowerMgr.c

```
284
285 static const EyeQResetTable_t ResetTable_s[C_EYEQP_NUM_OF_EYEQ_RESET_EVENTS] =
286 {
287     /*----- FaultSystemState_u8 = C_EYEQSYS_SYSTEM_SAFE_STATE -----*/
288     /*----- ErrEvent_u8 , ErrClass_u8 , ErrLimit_u16 -----*/
289     /*----- *Configure DG message based from here*-----*/
290     /* 0: C_EYEQDG_EYEQ_CONAR_VERSION_ERROR */ #if ( C_SUPPORT_OK == C_EYEQ_DG_CONAR_SUPPORT )
291         { C_EYEQDG_EYEQ_CONAR_VERSION_ERROR , C_EYEQDG_RESET_OK , C_EYEQ_5_RESTART_A
292         #else
293             { C_EYEQDG_EYEQ_CONAR_VERSION_ERROR , C_EYEQDG_RESET_NOK, C_EYEQ_0_RESTART_A
294             #endif
295     /* 1: C_EYEQDG_EYEQ_DFA_VERSION_ERROR */ #if ( C_SUPPORT_OK == C_EYEQ_DG_DFA_SUPPORT )
296         { C_EYEQDG_EYEQ_DFA_VERSION_ERROR , C_EYEQDG_RESET_OK , C_EYEQ_5_RESTART_A
297         #else
298             { C_EYEQDG_EYEQ_DFA_VERSION_ERROR , C_EYEQDG_RESET_NOK, C_EYEQ_0_RESTART_A
299             #endif
300 }
```

8)We need to verify if NVM integrity is proper and valid with help of structure NvM_BlockMngmtArea_at.NvRamErrorStatus_u8 should not be 3,5,6,7 or 8.

EyeQ CDD Integration

```
■ NVM_BlockMngmtArea_at =  
+ [0] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 0),  
+ [1] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 1, NvRamAttributes_u8 = 1),  
+ [2] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [3] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [4] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [5] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [6] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [7] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [8] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [9] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [10] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [11] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [12] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [13] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [14] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
+ [15] = (NvDataIndex_t = 0, NvRamErrorStatus_u8 = 0, NvRamAttributes_u8 = 1),  
#define NVM_REQ_OK (0U)  
#define NVM_REQ_NOT_OK (1U)  
#define NVM_REQ_PENDING (2U)  
#define NVM_REQ_INTEGRITY_FAILED (3U)  
#define NVM_REQ_BLOCK_SKIPPED (4U)  
#define NVM_REQ_NV_INVALIDATED (5U)  
#define NVM_REQ_CANCELED (6U)  
#define NVM_REQ_REDUNDANCY_FAILED (7U)  
#define NVM_REQ_RESTORED_FROM_ROM (8U)
```

9) SysTimer: System timer should be running.

10) We need to verify if diagnostic commands are working by using giving commands in canoe and also should verify CAN messages log in canoe.

11) If gliwa is enabled We need to verify any EyeQ task or other task are overloading

14.2 General Issues, debugging steps:

General issue in EyeQ integration are as follows along with debugging steps.

1)Safety reset issues .

Safety related resets we need to refer structure AoUSafe_LastMcuResetReason_a, AoUSafe_SafetyModulesTestStatus for source of smu alarm which triggered by which we can debug issue.

2)Bootloader and application shared variables not at same location. To find this issue we need to flash faulty application see if this is reported at bootloader

3) Application user configuration file not properly configures which is compatible with ME. ME supported feature are present Meio.cfg file which is discussed earlier in **ME** related chapter.

Core\Comp\ AppConfig\ Config\ EyeQCdd\ include\ EyeQ_DGUserServices.h

EyeQ CDD Integration

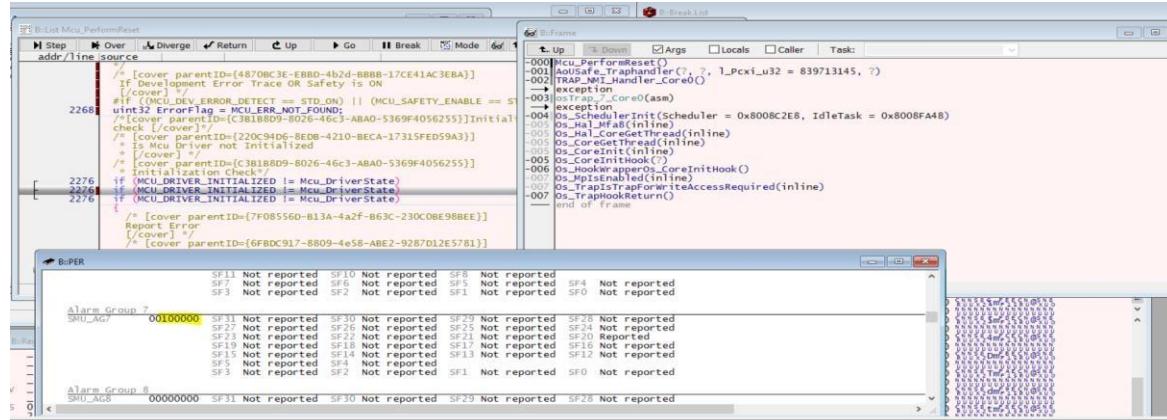
We see #define C_EYEQDG_EYEQ_MSG_TIMEOUT mismatch.

(0xD3U) error if any

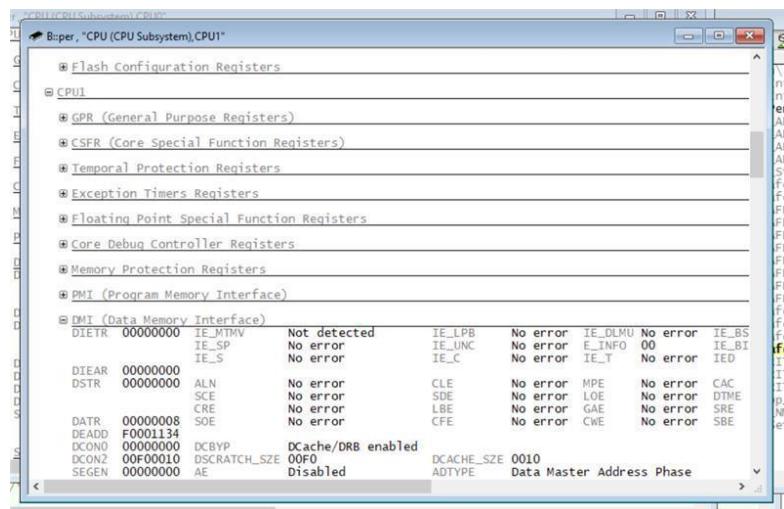
Examples of some of issues faced in G35 jupiter integration which gives details how to debug:

ISSUE 1) STM 1 Access Failure

a) Tried to find stack frame and SMU alarm's, its access related alarm



b) Identified source of reset from DEADD register its 0XF0001134 as below



c) From MCAL register address 0XF0001134 belong to STM1 as below

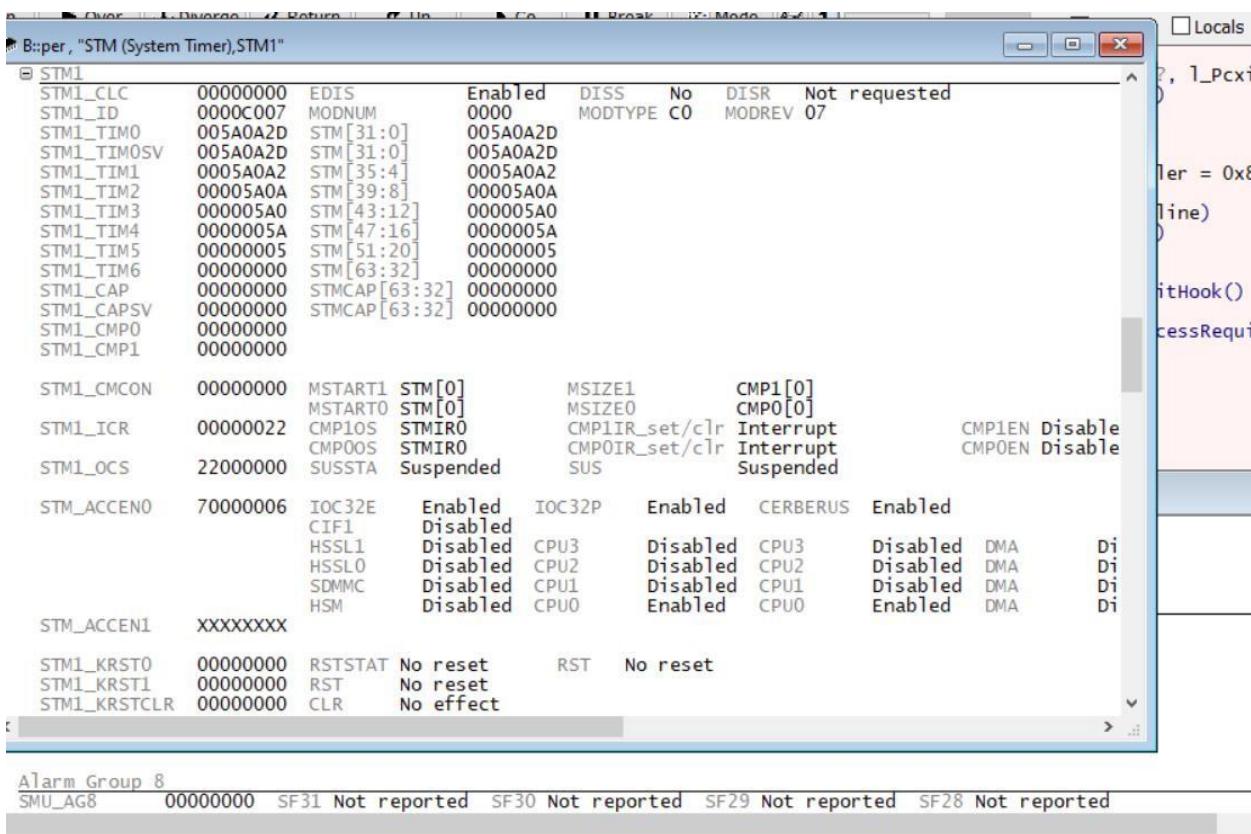
EyeQ CDD Integration

```

155 /** \brief 18, Timer Register 2 */
156 #define STM1_TIM2 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_TIM2*)0xF0001118u
157
158 /** \brief 1C, Timer Register 3 */
159 #define STM1_TIM3 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_TIM3*)0xF000111Cu
160
161 /** \brief 20, Timer Register 4 */
162 #define STM1_TIM4 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_TIM4*)0xF0001120u
163
164 /** \brief 24, Timer Register 5 */
165 #define STM1_TIM5 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_TIM5*)0xF0001124u
166
167 /** \brief 28, Timer Register 6 */
168 #define STM1_TIM6 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_TIM6*)0xF0001128u
169
170 /** \brief 2C, Timer Capture Register */
171 #define STM1_CAP /*lint --e(923, 9078)*/ (*volatile Ifx_STM_CAP*)0xF000112Cu
172
173 /** \brief 30, Compare Register 0 */
174 #define STM1_CMP0 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_CMP*)0xF0001130u
175
176 /** \brief 34, Compare Register 1 */
177 #define STM1_CMP1 /*lint --e(923, 9078)*/ (*volatile Ifx_STM_CMP*)0xF0001134u
178
179 /** \brief 38, Compare Match Control Register */
<

```

d) In below screen shot STM1_ACCENO is having access to core0 ,but expected is in core 1.



EyeQ CDD Integration

- e) Root cause as core as BSW in core1 its correct from them but G35 BSW is in core0 ,so for G35 we need to give CPU1 access by using separate MACRO, as BSW_TAGIG is 0.

```

D:\jup1.47\Application\SourceCode\CDD\AoUSafe\AoUSafe_AccProt\cfg\AoUSafe_AccProtCfg_38x.h
31-05-2021 15:41:37 79,360 bytes Read-only C/C++/C#/.Obj/Source ANSI PC
620 #endif
621
622
623 #if (AOUSAFE_CONVCTRL_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
624 #define AOUSAFE_CONVCTRL_MASTERS AOUSAFE_MASTER_CORE_TAGID
625 #endif
626
627 #if (AOUSAFE_STM0_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
628 #define AOUSAFE_STM0_MASTERS AOUSAFE_MASTER_CORE_TAGID
629 #endif
630
631 #if (AOUSAFE_STM1_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
632 #define AOUSAFE_STM1_MASTERS AOUSAFE_BSW_TAGID
633 #endif
634
635 #if (AOUSAFE_STM2_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
636 #define AOUSAFE_STM2_MASTERS AOUSAFE_CPU2_TAGID
637 #endif
638
639 #if (AOUSAFE_STM3_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
640 #define AOUSAFE_STM3_MASTERS AOUSAFE_CPU3_TAGID
641 #endif
642
643
644 #if (AOUSAFE_DMA_RP0_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
645 #define AOUSAFE_DMA_RP0_MASTERS AOUSAFE_EYEQCOD0_TAGID
646 #endif
647
648 #if (AOUSAFE_DMA_RP1_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
649 #define AOUSAFE_DMA_RP1_MASTERS AOUSAFE_EYEQCOD1_TAGID
650 #endif

```



```

D:\g35_debug\AoUSafe_AccProt\cfg\AoUSafe_AccProtCfg_38x.h
03-06-2021 16:06:45 79,511 bytes C/C++/C#/.Obj/Source ANSI PC
621 #endif
622
623
624 #if (AOUSAFE_CONVCTRL_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
625 #define AOUSAFE_CONVCTRL_MASTERS AOUSAFE_MASTER_CORE_TAGID
626 #endif
627
628 #if (AOUSAFE_STM0_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
629 #define AOUSAFE_STM0_MASTERS AOUSAFE_MASTER_CORE_TAGID
630 #endif
631
632 #if (AOUSAFE_STM1_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
633 #define AOUSAFE_STM1_MASTERS AOUSAFE_CPU1_TAGID
634 #endif
635
636 #if (AOUSAFE_STM2_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
637 #define AOUSAFE_STM2_MASTERS AOUSAFE_CPU2_TAGID
638 #endif
639
640 #if (AOUSAFE_STM3_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
641 #define AOUSAFE_STM3_MASTERS AOUSAFE_CPU3_TAGID
642 #endif
643
644
645 #if (AOUSAFE_DMA_RP0_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
646 #define AOUSAFE_DMA_RP0_MASTERS AOUSAFE_EYEQCOD0_TAGID
647 #endif
648
649 #if (AOUSAFE_DMA_RP1_ACCESSPROTECTION == AOUSAFE_ACCPROT_ENABLE)
650 #define AOUSAFE_DMA_RP1_MASTERS AOUSAFE_EYEQCOD1_TAGID
651 #endif

```

- f) Separate macro is added for CPU1 and provided STM access as above with below macro.

```

D:\jup1.47\Application\SourceCode\CDD\AoUSafe\AoUSafe_AccProt\cfg\AoUSafe_AccProtCfg_38x.h
31-05-2021 15:41:37 79,360 bytes Read-only C/C++/C#/.Obj/Source ANSI PC
440 /**
441 #define AOUSAFE_TRACE32_DEBUG (AOUSAFE_CERBERUS | AOUSAFE_IOC32P | AOUSAF
442 #define AOUSAFE_IOMHAB_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE
443 #define AOUSAFE_EYEQCOD0_TAGID (AOUSAFE_CPU1_NONSAFE | AOUSAFE_CPU1_SAFE
444 #define AOUSAFE_BSW_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
445 #define AOUSAFE_MASTER_CORE_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
446 #define AOUSAFE_CPU2_TAGID (AOUSAFE_CPU2_NONSAFE | AOUSAFE_CPU2_SAFE
447 #define AOUSAFE_CPU3_TAGID (AOUSAFE_CPU3_NONSAFE | AOUSAFE_CPU3_SAFE
448 #define AOUSAFE_ALL_CPU_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
449
450
451 #define AOUSAFE_EYEQCOD_DMA_PARTITIONS (AOUSAFE_DMA_RESOURCE_PARTITION0 | AOUSAF
452
453 #define AOUSAFE_NO_MASTER_TAGID (AOUSAFE_DISABLE_ALL_MASTERSACCESS | AOUSAF
454 #define AOUSAFE_ALL_MASTER_TAGID AOUSAFE_DEFAULT_ACCESS
455
456 /**
457 * @details The Macros below defines the Assigned Masters of Peripherals to access its register
458 *
459 * Example Configuration:
460 *
461 * 1. #define AOUSAFE_FCE_MASTERS DISABLE_ALL_MASTERSACCESS - This configuration disables t
462 */

```



```

D:\g35_debug\AoUSafe_AccProt\cfg\AoUSafe_AccProtCfg_38x.h
03-06-2021 16:06:45 79,511 bytes C/C++/C#/.Obj/Source ANSI PC
440 /**
441 #define AOUSAFE_TRACE32_DEBUG (AOUSAFE_CERBERUS | AOUSAFE_IOC32P | AOUSAF
442 #define AOUSAFE_IOMHAB_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE
443 #define AOUSAFE_EYEQCOD0_TAGID (AOUSAFE_CPU1_NONSAFE | AOUSAFE_CPU1_SAFE
444 #define AOUSAFE_BSW_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
445 #define AOUSAFE_MASTER_CORE_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
446 #define AOUSAFE_CPU1_TAGID (AOUSAFE_CPU1_NONSAFE | AOUSAFE_CPU1_SAFE
447 #define AOUSAFE_CPU2_TAGID (AOUSAFE_CPU2_NONSAFE | AOUSAFE_CPU2_SAFE
448 #define AOUSAFE_CPU3_TAGID (AOUSAFE_CPU3_NONSAFE | AOUSAFE_CPU3_SAFE
449 #define AOUSAFE_ALL_CPU_TAGID (AOUSAFE_CPU0_NONSAFE | AOUSAFE_CPU0_SAFE | AOU
450
451
452 #define AOUSAFE_EYEQCOD_DMA_PARTITIONS (AOUSAFE_DMA_RESOURCE_PARTITION0 | AOUSAF
453
454 #define AOUSAFE_NO_MASTER_TAGID (AOUSAFE_DISABLE_ALL_MASTERSACCESS | AOUSAF
455 #define AOUSAFE_ALL_MASTER_TAGID AOUSAFE_DEFAULT_ACCESS
456
457 /**
458 * @details The Macros below defines the Assigned Masters of Peripherals to access its register
459 *
460 * Example Configuration:
461 *
462 * 1. #define AOUSAFE_FCE_MASTERS DISABLE_ALL_MASTERSACCESS - This configuration disables t
463 */

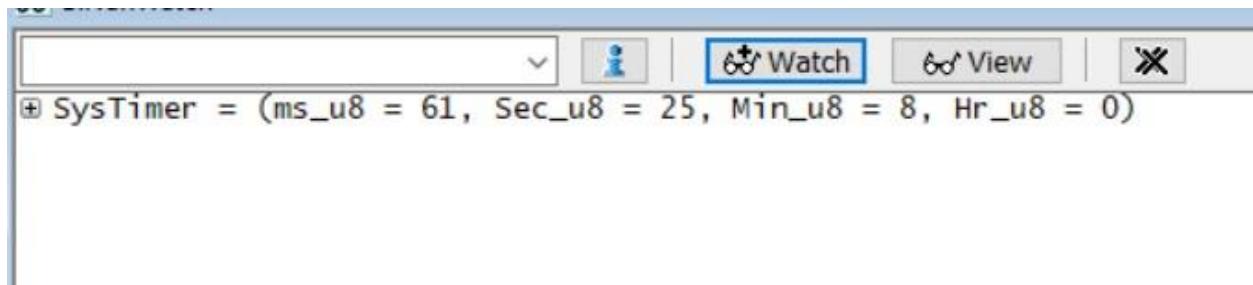
```

- g)STM1 as STM_ACCEN0 with access given to CPU1 now as below

EyeQ CDD Integration

B::per, "STM (System Timer), STM1"							
STM1	00000000	EDIS	Enabled	DISS	No	DISR	Not requested
STM1_CLC	0000C007	MODNUM	0000	MODTYPE	C0	MODREV	07
STM1_TIM0	2A2A4035	STM[31:0]	2A2A4035				
STM1_TIM0SV	2A2AD225	STM[31:0]	2A2AD225				
STM1_TIM1	02A2B760	STM[35:4]	02A2B760				
STM1_TIM2	002A2B76	STM[39:8]	002A2B76				
STM1_TIM3	0002A2B7	STM[43:12]	0002A2B7				
STM1_TIM4	00002A2B	STM[47:16]	00002A2B				
STM1_TIM5	000002A2	STM[51:20]	000002A2				
STM1_TIM6	00000000	STM[63:32]	00000000				
STM1_CAP	00000000	STMCAP[63:32]	00000000				
STM1_CAPSV	00000000	STMCAP[63:32]	00000000				
STM1_CMP0	2A38F99C						
STM1_CMP1	2A4558F2						
STM1_CMCON	001F001F	MSTART1 STM[0]		MSIZE1		CMP1[31:0]	
		MSTART0 STM[0]		MSIZE0		CMP0[31:0]	
STM1_ICR	00000073	CMP10S STMIR1		CMP1IR_set/clr	Interrupt		
		CMP00S STMIR0		CMP0IR_set/clr	Interrupt		
STM1_OCS	02000000	SUSSTA	Not suspended	SUS		Suspended	
STM_ACCENO	70000060	IOC32E	Enabled	IOC32P	Enabled	CERBERUS	Enabled
		CIF1	Disabled				
		HSSL1	Disabled	CPU3	Disabled	CPU3	Disabled
		HSSL0	Disabled	CPU2	Disabled	CPU2	Disabled
		SDMMC	Disabled	CPU1	Enabled	CPU1	Enabled
		HSM	Disabled	CPU0	Disabled	CPU0	Disabled
STM_ACCEN1	XXXXXXXX						
STM1_KRST0	00000000	RSTSTAT	No reset	RST	No reset		
STM1_KRST1	00000000	RST	No reset				
STM1_KRSTCLR	00000000	CLR	No effect				

h) Issue is resolved as software is running.



ISSUE 2) DMA related additional access problem as CDD is not integrated as per project Decision.

- a) Could see safety alarm set from SBFD_SPI_TX_RX

EyeQ CDD Integration

Browser.Function

AFE_Error_Handler Type: Functions Source

B::List 'Smu_SetAlarmStatus'

Step Over Diverge Return Up Go

addr/source

** Return value : E_OK - Operation successful
** E_NOT_OK - Operation failed
** parameters.

2481 Std_ReturnType Smu_SetAlarmStatus
{
 const Smu_AlarmGroupId AlarmGroup,
 const Smu_AlarmIdType AlarmPos

 Std_ReturnTypeRetVal = E_NOT_OK;
 uint32 Timeout = 0U;
 uint32 AlarmStatusReadback = 0xFFFFFFFFU;

 /*Check if the DET or Safety checks are
 * [cover parentID={FA987376-A7E7-4508-9500-000000000000}]
 * DET OR Safety Checks ON
 * [/cover] */
 #if (SMU_DEV_ERROR_DETECT == STD_ON) ||

 uint32 AlarmRes = 0U;
 uint32 CoreAlarmIdReservedType[SMU_CORE_TOTAL_ALARM_GROUPS] =
 {
 SMU_GROUP0_POS,

SF5 Not reported SF4 Not reported
SF3 Not reported SF2 Not reported SF1 Not reported SF0 Not reported

6 B::Frame

Up Down Args Locals Caller Task:

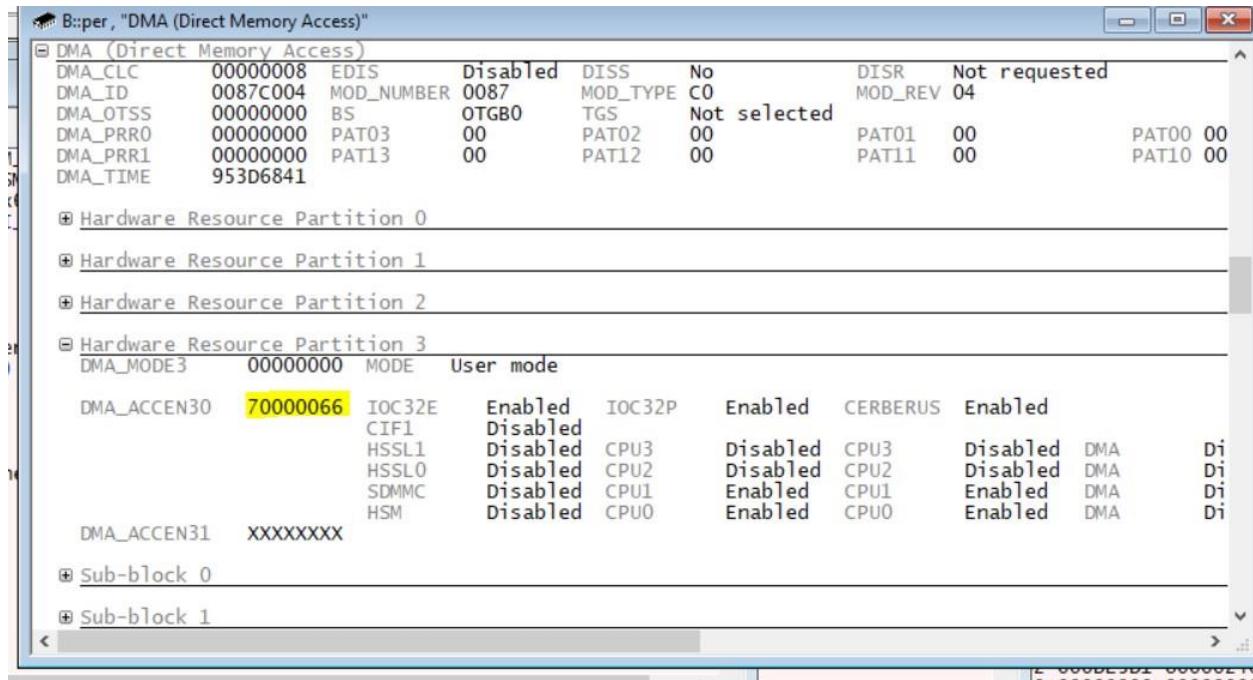
-000 Smu_SetAlarmStatus(AlarmGroup = SMU_ALARM_GROUP10, AlarmPos = SMU_ALARM_1)
-001 AOSAFE_Error_Handler(f_Test_Status_e = SMU_TRIGGER_RESET)
-002 TRUSTED_SBFD_SPI_TxRx(dataBufOut_pub = 0x6002F400, dataBufIn_pu8 = 0x6002EC00, ?, num_bytes_u16 = 9)
-003 TRUSTED_Os_ServiceCallee_TRUSTED_SBFD_SPI_TxRx(?, FunctionParams = 0x70014F00)
-004 Os_Hal_IntIsLevelSupported(inline)
-004 Os_InterruptLockOnLevelIsNeeded(inline)
-004 Os_IntSuspend(inline)
-004 Os_ServiceTfCall(inline)
-004 Os_Api_CallTrustedFunction(?, ?)
-005 CallTrustedFunction(FunctionIndex = Os_ServiceCallee_TRUSTED_SBFD_SPI_TxRx, FunctionParams = 0x70014F00)
-006 Os_Call_TRUSTED_SBFD_SPI_TxRx(?, ?, ?, ?)
-007 SBFD_SendReadDataCommand(?, ?)
-008 EYEQ_SBS_Verify()
-009 Os_Task_EyeQ_IPC_2ms_OsTask_1()
-010 Os_MpIsEnabled(inline)
-010 Os_TrapIsTrapForWriteAccessRequired(inline)
-010 Os_TrapTaskMissingTerminateTask()
end of frame

b) IN SDFD_SPI_TX_RX source of safety issue is DMA_TX_REGCHECK function

```
CoreAlarmIdReservedType = (32768, 32768, 0, 0, 0, 0, 0, 32768, 0, 0, 2147830800, 2149597630, 21485995 ^  
AOUSAFE_Error_Handler()  
    .f_Test_Status_e = SMU_TRIGGER_RESET)  
.l_IsSafetyTestFailed_bo = 1  
AoUSafe_SafetyModulesTestStatus = (ModuleTestElements = (LBIST_Test_Status_b = 0, MONBIST_Test_Status_b = 0,  
    {  
        /* Perform the SMU RESET action */  
        (void)Smu_SetAlarmStatus(SMU_ALARM_GROUP10, SMU_ALARM_1); /*QAC 3200*/  
TRUSTED_SBFD_SPI_TxRx()  
    #dataBufOut_pu8 = 0x6002F400,  
    #dataBufIn_pu8 = 0x6002EC00,  
?,  
    .num_bytes_u16 = 9)  
.ss_code_u32 = 9  
.IPCDRV_FinalBacon_u32 = 2476817561  
.IPCDRV_BeginBacon_u32 = 2476817560  
  
    /* Register check Test */  
    SBFD_DmaTx_ReqCheck();  
TRUSTED_Os_ServiceCallee_TRUSTED_SBFD_SPI_TxRx()  
?,  
#FunctionParams = 0x70014F00)
```

c) Could see DMA belong to partition 3 with access to both CPU0,CPU1.

EyeQ CDD Integration



d)root cause , access in below file with access to CPU1,CPU0.

```
#\g35_debug\{Application\Core_Comp\AppConfig\Config_AoUSafe\include\AoUSafe_Application_Config.h
03-06-2021 17:41:26 17,708 bytes C,C++,C#,ObjC Source ▾ ANSI ▾ PC
207 */
208 #define AOUSAFAE_CLOCK_PLAUSIBILITY_SUPPORT (AOUSAFAE_SUPPORT_OK)
209 /* @details Enable/disable safety test
210 * AOUSAFAE_SUPPORT_OK - Enables FEE
211 * AOUSAFAE_SUPPORT_NOK - Disables FEE
212 */
213
214 #define AOUSAFAE_SUPPORT_FEE (AOUSAFAE_SUPPORT_NOK)
215
216 #if (ECUM_NUMBER_OF_CORES > 1U)
217 /**
218 * @Details Number of lock step cores in the 38x variant
219 */
220 #define AOUSAFAE_NUMBER_OF_LOCKSTEP_CORES_USED 2U
221
222 #elif (1U == ECUM_NUMBER_OF_CORES)
223 /**
224 * @Details Number of lock step cores for single core
225 */
226 #define AOUSAFAE_NUMBER_OF_LOCKSTEP_CORES_USED 1U
227
228#endif
229
230
231 /*Access protection configuration macros*/
232 #define AOUSAFAE_AP_DMA_RP3_APP_MASTERS (AOUSAFAE_CPU0_NONSAFE | AOUSAFAE_CPU0_SAFE | AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
233
234 #define AOUSAFAE_AP_FCE_APP_MASTERS (AOUSAFAE_CPU0_NONSAFE | AOUSAFAE_CPU0_SAFE)
235
236 #define AOUSAFAE_AP_GETH_APP_MASTERS (AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
237 #define AOUSAFAE_AP_GETH_D0_APP_MASTERS (AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
238 #define AOUSAFAE_AP_GETH_D1_APP_MASTERS (AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
239 #define AOUSAFAE_AP_GETH_D2_APP_MASTERS (AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
240 #define AOUSAFAE_AP_GETH_D3_APP_MASTERS (AOUSAFAE_CPU1_NONSAFE | AOUSAFAE_CPU1_SAFE)
```

e)After going through DMA reg check function there its giving access to only 1 core need to remove extra access as below. Issue introduced as we did not integrate CDD as per project decision. In latest code of CDD this issue will not be there as REG check of DMA is present .

EyeQ CDD Integration

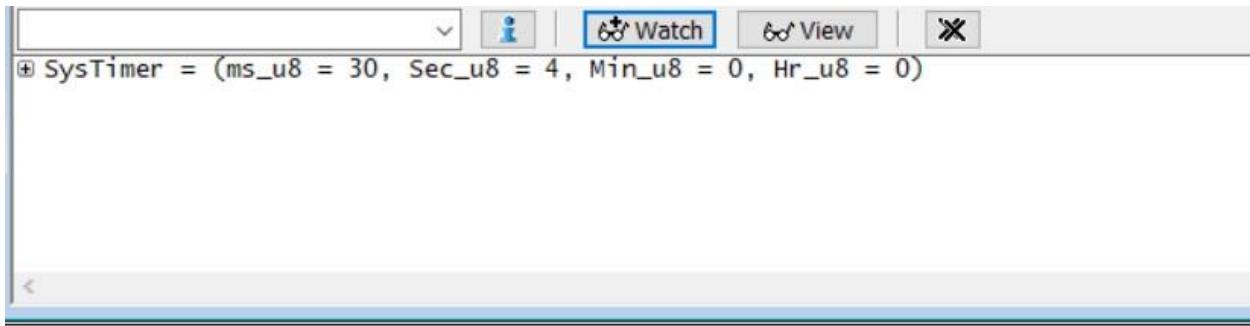
```
j35_debug\Application\Core_Comp\ AppConfig\Config_AoUSafe\include\AoUSafe_Application_Config.h
06-2021 17:58:41 17,667 bytes C,C++,C#,ObjC Source ▾ ANSI ▾ PC
228 #endif
229
230
231 /*Access protection configuration macros*/
232 #define AOUSAPE_AP_DMA_RP3_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE)
233
234 #define AOUSAPE_AP_FCE_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE)
235
236 #define AOUSAPE_AP_GETH_APP_MASTERS (AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
237 #define AOUSAPE_AP_GETH_D0_APP_MASTERS (AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
238 #define AOUSAPE_AP_GETH_D1_APP_MASTERS (AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
239 #define AOUSAPE_AP_GETH_D2_APP_MASTERS (AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
240 #define AOUSAPE_AP_GETH_D3_APP_MASTERS (AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
241
242 #define AOUSAPE_AP_CAN0_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
243 #define AOUSAPE_AP_CAN0_CTR_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
244
245
246 #define AOUSAPE_AP_CAN1_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
247 #define AOUSAPE_AP_CAN1_CTR_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
248
249 #define AOUSAPE_AP_CAN2_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
250 #define AOUSAPE_AP_CAN2_CTR_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
251
252 #define AOUSAPE_AP_DMU_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_CPU1_NONSAFE | AOUSAPE_CPU1_SAFE)
253
254 #define AOUSAPE_AP_QSPI3_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE|AOUSAPE_DMA_RESOURCE_PARTITION3)
255 #define AOUSAPE_AP_QSPI4_APP_MASTERS (AOUSAPE_CPU0_NONSAFE | AOUSAPE_CPU0_SAFE)
256
257
258
259 ****
260 /*
```

f) After modification we could see ACCESS is given to only CPU0 as below

B::per , "DMA (Direct Memory Access)"								
DMA (Direct Memory Access)								
DMA_CLC	00000008	EDIS	Disabled	DISS	No	DISR	Not requested	
DMA_ID	0087C004	MOD_NUMBER	0087	MOD_TYPE	C0	MOD_REV	04	
DMA_OTSS	00000000	BS	OTGBO	TGS	Not selected			
DMA_PRR0	00000000	PAT03	00	PAT02	00	PAT01	00	PAT00 00
DMA_PRR1	00000000	PAT13	00	PAT12	00	PAT11	00	PAT10 00
DMA_TIME	29356868							
⊕ Hardware Resource Partition 0								
⊕ Hardware Resource Partition 1								
⊕ Hardware Resource Partition 2								
⊕ Hardware Resource Partition 3								
DMA_MODE3	00000000	MODE	User mode					
DMA_ACCEN30	70000006	IOC32E CIF1 HSSL1 HSSL0 SDMMC HSM	Enabled Disabled Disabled Disabled Disabled Disabled	IOC32P CPU3 CPU2 CPU1 CPU0	Enabled Disabled Disabled Disabled Enabled	CERBERUS CPU3 CPU2 CPU1 CPU0	Enabled Disabled Disabled Disabled Enabled	DMA Di Di Di DMA Di
DMA_ACCEN31	XXXXXXXXXX							
⊕ Sub-block 0								
⊕ Sub-block 1								

g) Above change fixed issue , software is running as expected.

EyeQ CDD Integration



Issue 3) Shared variables between application and bootloader location should be identical.

These shared variables helps in knowing any resets of SMU, MCU. We have modified address in application to match bootloader otherwise these variables interpretation will be wrong we will not be able to log issues in bootloader .

```
3
4 eserved "APPLICATION_CHECKSUM" (size = SIZE_APPL_CRC);
5
6 #####/#
7 RAM_NO_CLEAR
8
9 group RAM_NO_CLEAR_SHARED (contiguous, ordered, attributes=rws, run_addr=[0xF0240100..0xF0240140]
0
1 select ".bss.AoUSafe_ErrorHandler.#g_SafetyModulesTestStatus_u32";
2 select ".bss.IoHwAb_PowerMgr.#Cypress_Flashcorruption_Status";
3
4
5 ***** Variable used by Application in this region will persist across reset cycles must not be
6 reserved "RAM_NO_CLEAR_APPL" (contiguous, ordered, attributes=rws, run_addr=[0xF0240140..0xF0240400];
7
8
9 select ".bss.AoUSafe_ErrorHandler.#_LastSmuResetReason_u32";
0
1
2
3 group CORE1_NOINIT_INIT(ordered,run_addr = mem:dspr1) ///*mem:dspr1*/
4
5 group osFarCore1(ordered, contiguous, fill, align=8)
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
607
608
609
609
610
611
612
613
614
614
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534

```