

<b>Contents</b>	<b>Page</b>
<b>1 Introduction .....4</b>	<b>4</b>
1.1 DAVE™ Components and Features .....4	4
1.2 Acronyms and abbreviations .....5	5
<b>2 DAVE™ Installation .....6</b>	<b>6</b>
2.1 Installation through zip file .....6	6
2.2 Installation through DAVE™ set up exe .....6	6
<b>3 Getting Started .....7</b>	<b>7</b>
3.1 General Flow .....7	7
3.2 Creation of DAVE™ 3 Workspace .....8	8
3.3 Install Apps .....9	9
3.4 Creation of New DAVE™ Project .....10	10
3.5 DAVE™ App Dependency Graph View .....14	14
3.6 UI Configurations .....15	15
3.7 Signal Connection .....17	17
3.8 Manual Pin Assignment .....18	18
3.9 Report .....19	19
3.10 Code Generation .....21	21
3.11 Generated Files and Edit main.c .....22	22
3.12 Build the Project .....23	23
3.13 Target Board Setup .....24	24
3.14 Download / Debug .....24	24
<b>4 Project Management .....27</b>	<b>27</b>
4.1 DAVE™ Perspectives .....27	27
4.2 New DAVE™ Project creation .....28	28
4.3 Upgrade to DAVE™ CE project .....31	31
4.4 Active Project .....31	31
4.5 Importing/Exporting of Projects .....32	32
4.5.1 Import - Projects .....32	32
4.5.2 Export - Projects .....35	35
4.6 Views .....38	38
4.6.1 DAVE™ AppSelection View .....38	38
4.6.1.1 DAVE™ AppSelection View – Flat-list Search filter .....39	39
4.6.1.2 DAVE™ AppSelection View – Category Search filter .....40	40
4.6.2 DAVE™ AppDependency View .....40	40
4.6.2.1 S/W App Dependency Tree View .....40	40
4.6.2.2 S/W App Connectivity View .....41	41
4.6.2.3 H/W Connectivity View .....42	42
4.7 UI Editor .....42	42
4.8 Remove .....43	43
4.9 Signal Connection .....43	43
4.9.1 Signal and Connection .....43	43
4.9.2 Signal Connection - Configuration .....44	44
4.10 Signal Assignment .....45	45
4.11 Add User Label .....46	46
4.12 Manual Pin Assignment .....46	46
4.13 Report .....47	47
4.13.1 Apps .....48	48
4.13.2 Resource Mapping .....49	49
4.13.3 Pin Assignment .....50	50
4.13.4 Signal Assignment Report .....51	51
4.14 Project View .....51	51
4.14.1 Generated Files .....51	51
4.14.2 Edit MAIN.C .....52	52
4.14.3 Document generation .....53	53
4.15 Project Build .....54	54
4.16 Debugger Support .....56	56
4.16.1 Debugger Support – Using MiniWiggler .....56	56

Contents	Page
4.16.2	Debugger Support – Using JLink .....57
4.17	Project Deletion .....59
4.18	Software Updates .....60
<b>5</b>	<b>DAVE™ Apps &amp; Library Management.....62</b>
5.1	DAVE™ Apps .....62
5.2	Categories of DAVE™ Apps .....62
5.3	DAVE™ App Library .....63
5.4	Library groups .....64
5.5	DAVE™ Apps Types .....64
5.5.1	Sharable DAVE™ Apps .....64
5.5.2	Not Sharable DAVE™ Apps.....66
5.5.3	Singleton DAVE™ Apps.....67
5.6	DAVE™ App Dependency controls .....69
5.7	Install DAVE™ Apps .....71
5.8	Update & Uninstall DAVE™ Apps.....73
5.8.1	DAVE™ App update .....73
5.8.2	DAVE™ App uninstall .....73
<b>6</b>	<b>Menu .....75</b>
6.1	Install App Library .....75
6.2	DAVE™ IDE Project Wizard .....76
6.3	Upgrade to DAVE™ CE project .....77
6.4	Save Application Model.....77
6.5	Manual Pin Assignment .....78
6.6	Signal Connection .....79
6.7	Solver .....80
6.8	Reports.....80
6.9	Generate Code.....81
6.10	Generate Doc .....82
6.11	View Generate Doc .....82
6.12	Manual Resource Assignment .....83
6.13	Solver Log .....84
6.14	Generate xSPY INI File.....84
6.15	Clear Code Generator Cache .....85
6.16	Reserve Resource Group .....86
6.17	Apps Migration .....87
6.18	Device Migration .....88
<b>7</b>	<b>Configurations.....89</b>
7.1	UI Editor – Selection .....89
7.1.1	Dialogue/page display .....90
7.1.2	UI Controls .....90
7.1.2.1	Integer and their Types .....90
7.1.2.2	String .....90
7.1.2.3	Check Box .....91
7.1.2.4	Radio Box/Option Button.....91
7.1.2.5	Combo Box.....91
7.1.2.6	Enabled and Disabled .....91
7.1.2.7	UI control Dependency.....91
7.1.2.8	Tool Tip Support.....92
7.1.2.9	Range [Minimum & Max Value] supported .....93
7.1.2.10	Non-Configurable Apps .....93
7.2	DAVE™ Settings.....94
7.2.1	Preferences Settings .....94
7.2.2	About DAVE .....96
<b>8</b>	<b>DAVE™ Help System .....98</b>
8.1	Dynamic Help .....98
8.2	DAVE™ App Help .....101

Contents	Page
<b>9</b>	<b>Signal Connection.....103</b>
9.1	Signal Connection – Signal to Signal .....103
9.2	Signal Connection – Signal to Port .....103
9.3	Signal Connection – Signal to Interrupt/ NVIC .....104
<b>10</b>	<b>Manual Pin Assignment.....106</b>
10.1	Configuring GPIO Input .....106
10.2	Configuring GPIO Output .....107
10.3	Changing the Pin solution provided by Solver to different one .....109
<b>11</b>	<b>Report (Resource Binding Information) .....111</b>
11.1	Resource Binding Report .....111
11.1.1	Resource consuming SFRegister/Bitfield .....111
11.1.2	Resource consuming Slice.....112
11.1.3	Resource consuming Interrupt Resource .....113
11.1.4	Resource consuming PORT Resource .....113
11.1.5	Resource consuming Scenarios .....113
11.2	PIN Assignment Report.....114
11.3	Signal Assignment Report.....114
<b>12</b>	<b>DAVE™ App&amp; Device Migration .....116</b>
12.1	Apps Migration Feature .....116
12.2	Device Migration Feature .....119
<b>13</b>	<b>BMI Feature.....121</b>
<b>14</b>	<b>xSPY Feature .....122</b>
<b>15</b>	<b>ARM GCC compiler .....123</b>
<b>16</b>	<b>FAQ.....124</b>
16.1	Project Management .....124
16.1.1	How user will know which perspective is active? .....124
16.1.2	What is DAVE™ IDE and DAVE™ CE perspective? .....124
16.1.3	How user will know about Active Project? .....124
16.1.5	Why Solver will throw Error? .....125
16.1.6	How to check which device is configured / selected for the current project? .....125
16.2	Versions of App .....126
16.2.1	How can user know which Version of App is installed? .....126
16.2.2	How many versions of the same App installed in Library? .....126
16.3	UI Configuration .....127
16.3.1	How different conversion types will work for an integer parameter? .....127
16.4	Code, Compiler & Debugger .....128
16.4.1	Which are the generated files that user can edit?.....128
16.4.2	What are Multiplexer.c & h files?.....128
16.4.3	Which are the compilers DAVE™ Supports?.....128

# 1 Introduction

DAVE™ version 3 is a free and high productivity development platform for code generation based on predefined application use case oriented SW components: DAVE™ Apps

Today the most used methodology to initialize and drive the peripherals of microcontrollers are static configurable libraries DAVE™ significantly improves this concept by introducing DAVE™ Apps.

DAVE™ Apps are application use case oriented SW components that cover a wide range of use cases such as generation of PWM signals, sending or receiving messages over SPI, UART or CAN, converting analog signals and much more. DAVE™ Apps abstract any kind of application use case and can be regarded as a class that can be instantiated as often as necessary and flexible combined to build the required library. A resource solver supports the assignment of chip resources avoiding any HW conflict. Once the library is created the user can control the peripherals via well defined programming interfaces (APIs).

## 1.1 DAVE™ Components and Features

DAVE™ includes:

- Eclipse CDT based IDE with improved project management
- GNU C-Compiler tools
- Debugger incl. Flash loader
- Code generation plug-in (Code Engine) with graphical user interfaces to generate the library code
- A resource solvers provides automatic or constrained assignment of chip resources to the DAVE™ Apps
- Library manger to download and manage the DAVE™ Apps or Example projects.
- Data visualization tool xSPY
- DOXYGEN documentation tool

DAVE™ can be extended with 3rd party or open source plug-ins like the µVision plug-in or SVN.

The generated library code can be imported and used from other tool chains, like Altium, Atollic, IAR, Keil, or Rowley.

## 1.2 Acronyms and abbreviations

Acronym/ Abbreviation	Description
μC	Microcontroller
API	Application Programming Interface
APP	Software components that solves a specific uses case
CDT	C/C++ Development Toolkit
DAS	Device Access Server
DAVE	Digital Application Virtual Engineer Version 3
Eclipse	Eclipse is a multi-language software development environment
GUI	Graphical User Interface
HW	Hardware
IDE	Integrated development environment
JDT	Java Development Toolkit
MBS	Managed Build System
MS	Mother System
OS	Operating System
SDK	Software Development Kit
SW	Software
UML	Unified Modeling Language

**Table 1      Acronyms**

Note: All images / pictures in this document are with respect to Windows7 OS, similar behavior can be observed with other platforms.

## 2 DAVE™ Installation

### 2.1 Installation through zip file

This package contains a zip files with all required files to run DAVE™ on a PC plus the installation setup for the SEGGER J-Link drivers. The zip file can be unzipped anywhere and DAVE™ can be started from the ...\eclipse folder. The package is wrapped in an exe that users can accept the licenses conditions.

### 2.2 Installation through DAVE™ set up exe

The DAVE-3.1.8 can be installed parallel to an older version of the DAVE™ as well. Starting with DAVE-3.1.8, maintaining parallel installations of DAVE™ is supported. The existing DAVE™ installation and projects will be retained.

To use the existing projects with DAVE™ 3.1.8, Please use the import feature of DAVE. For detailed instructions, please refer to section 4.5.1

#### **Installation of DAVE™ Apps libraries and device description:**

If the installer doesn't find an existing default local Library Store:

..\Users\<user>\Infineon\D3LibraryStore\_3.1.8

It creates a new local Library Store and stores a complete set of DAVE™ App libraries and device descriptions. Hence it is not required to additionally download the DAVE™ Apps and device description from the web after the DAVE™ installation.

### 3 Getting Started

#### 3.1 General Flow

In order to use Apps getting working on target board follow the steps as mentioned below.

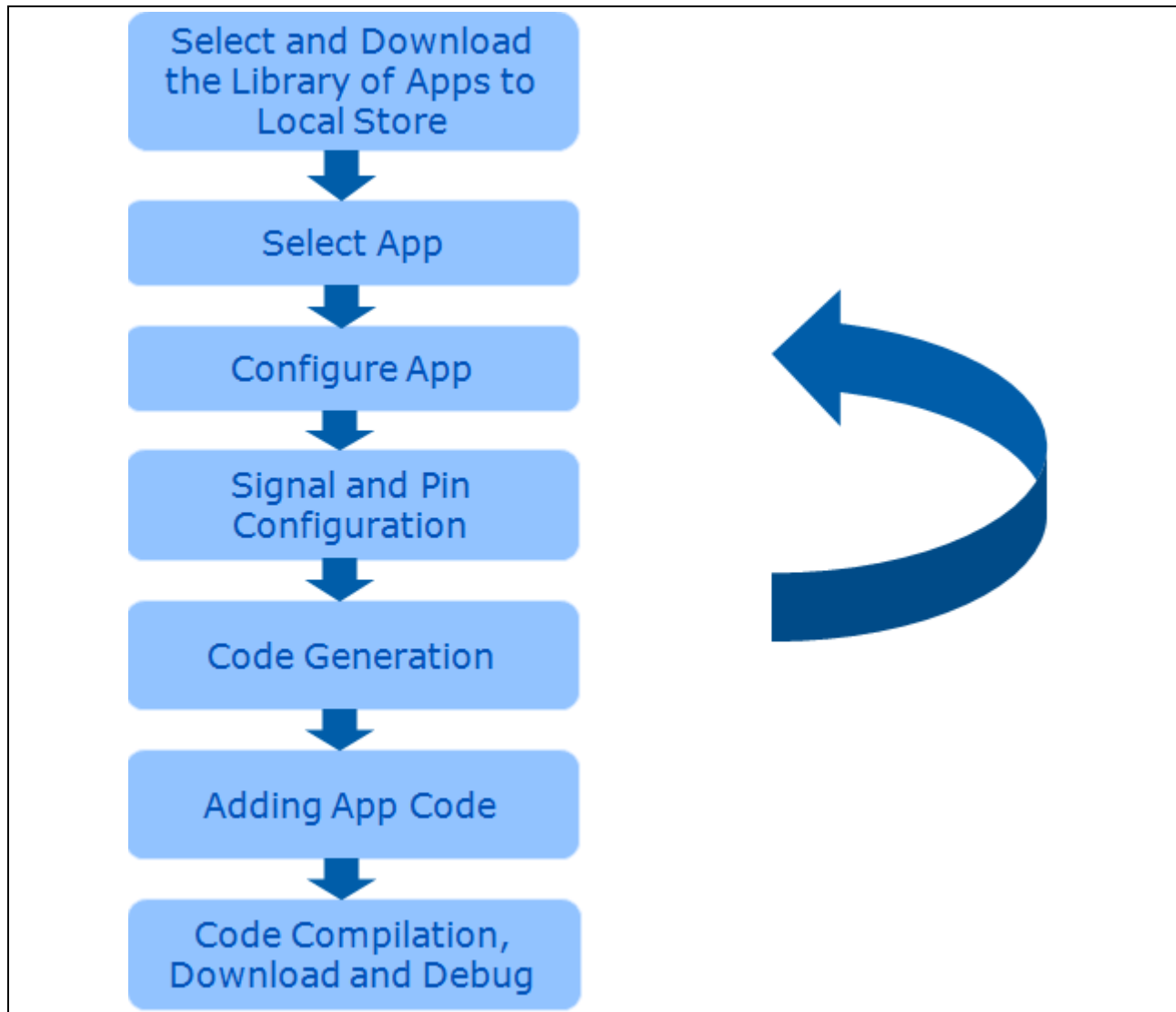


Figure 1 DAVE™ General Flow

The below sections/ steps are explained by considering example of toggling LED by using interrupt on PWM compare event.

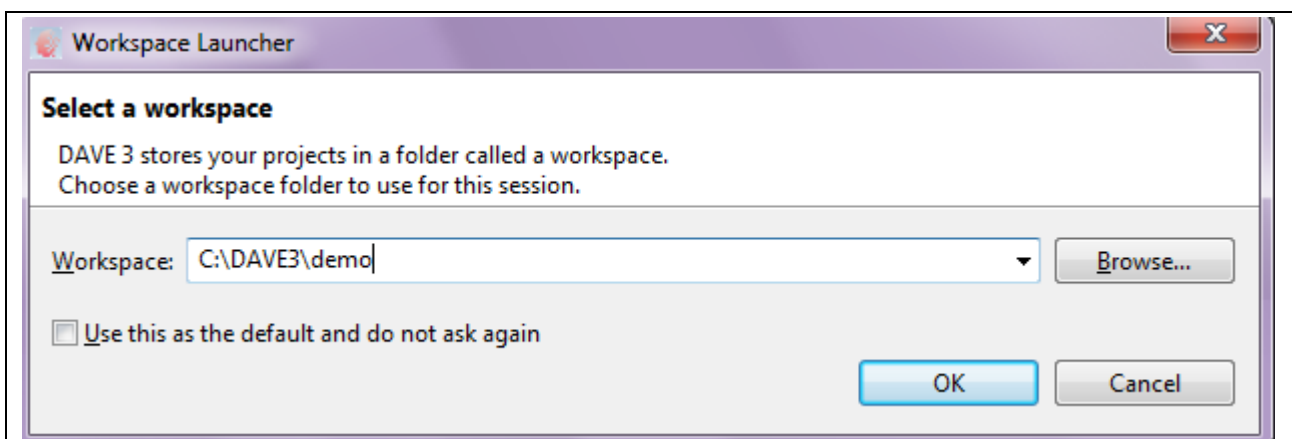
### 3.2 Creation of DAVE™ 3 Workspace

→ To open DAVE, user has to execute DAVE-3.1.8.exe or shortcut of DAVE-3.1.8.exe.



**Figure 2 DAVE™ Getting Started – Tool opens**

→ Workspace launcher window will be opened, provide/select user workspace through Browse button and click on OK button.



**Figure 3 DAVE™ Getting Started - New workspace**

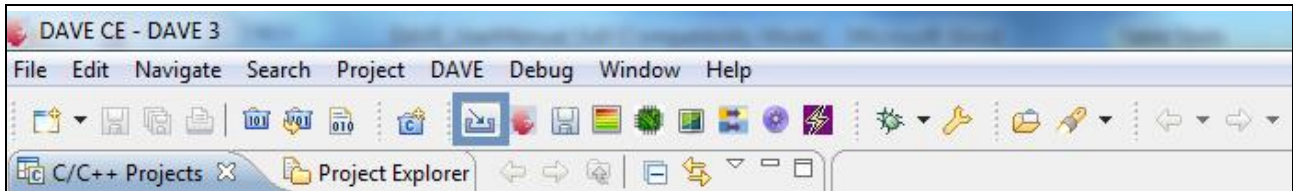
- It opens DAVE™ Workspace
- DAVE™ IDE – DAVE™ workspace contains [DAVE™ IDE perspective] (Detailed explanation is available in later sections)
  - Project
  - Editor view



### 3.3 Install Apps

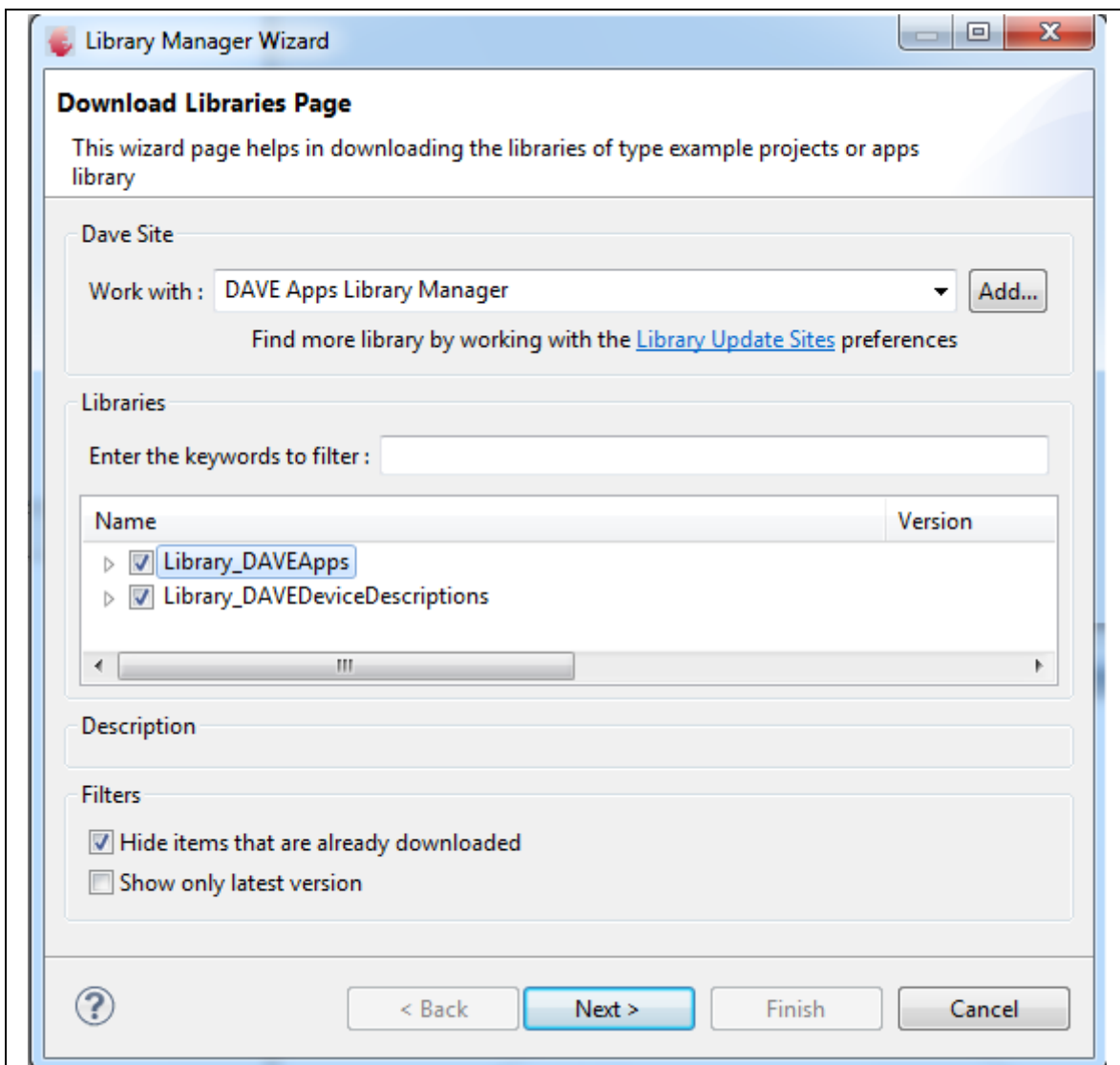
To start with demo app running on target board, first DAVE™ Applibrary has to be installed.

- Install the DAVE™ AppLibrary via Install DAVE™ AppLibrary Menu button



**Figure 4 DAVE™ Getting Started - Install Apps - via Install App Library**

- ➔ By clicking Install Apps, 'Download Library page' wizard window will open
  - Through this user can download DAVE™ Applibraries to local store. For Example C:\\user...
- ➔ Click 'Add' and provide path
- ➔ Check the Library
- ➔ Click Finish



**Figure 5 DAVE™ Getting Started - Download Libraries – Device**

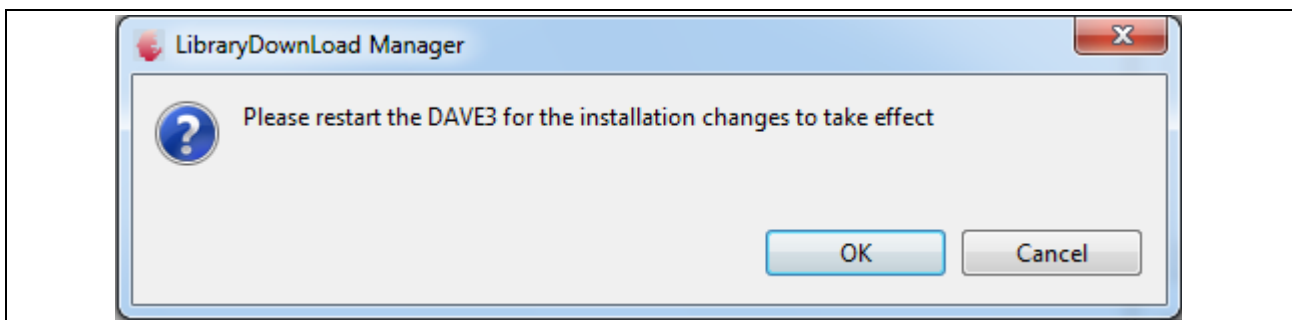
Please accept the licence agreement in order to install Apps.

NOTE: Library contains bunch of Apps. DAVE™ will take some time to Install Selected app to local library store (Depends on PC CPU speed)

- ➔ After successful installation of DAVE™ AppLibrary, DAVE™ will ask for restart
  - Click 'OK' button to restart DAVE

Note:

1. 'Restart' will only restarts the DAVE™ 3 Workbench.
2. If 'Cancel' button is selected then DAVE™ AppLibrary may not be installed correctly.



**Figure 6 DAVE™ Getting Started - Library Download Manager Restart**

- ➔ Then it will open last Workspace
  - Please select the workspace and then click OK

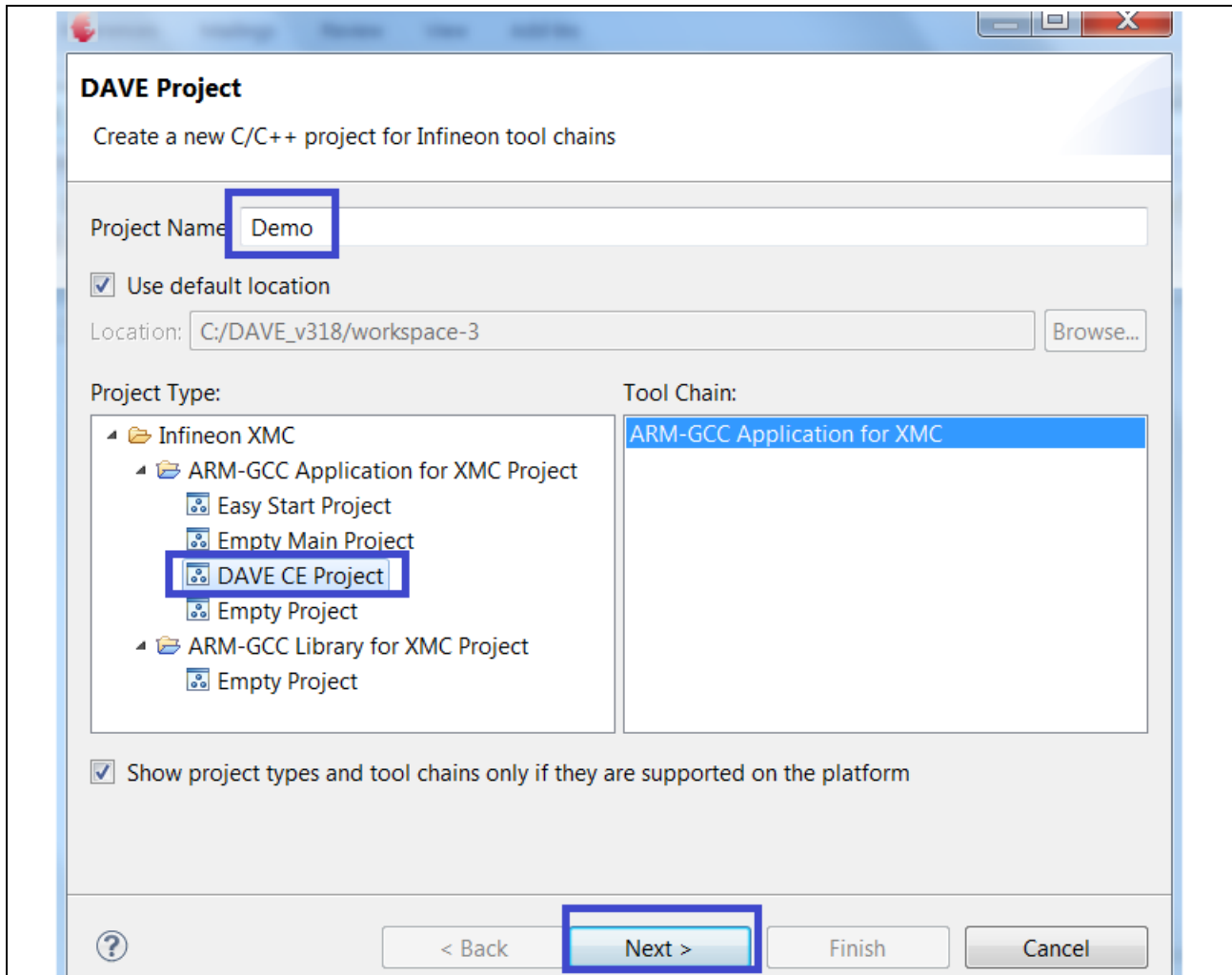
### 3.4 Creation of New DAVE™ Project

- ➔ Click DAVE™ IDE New Project Wizard menu to create the new project



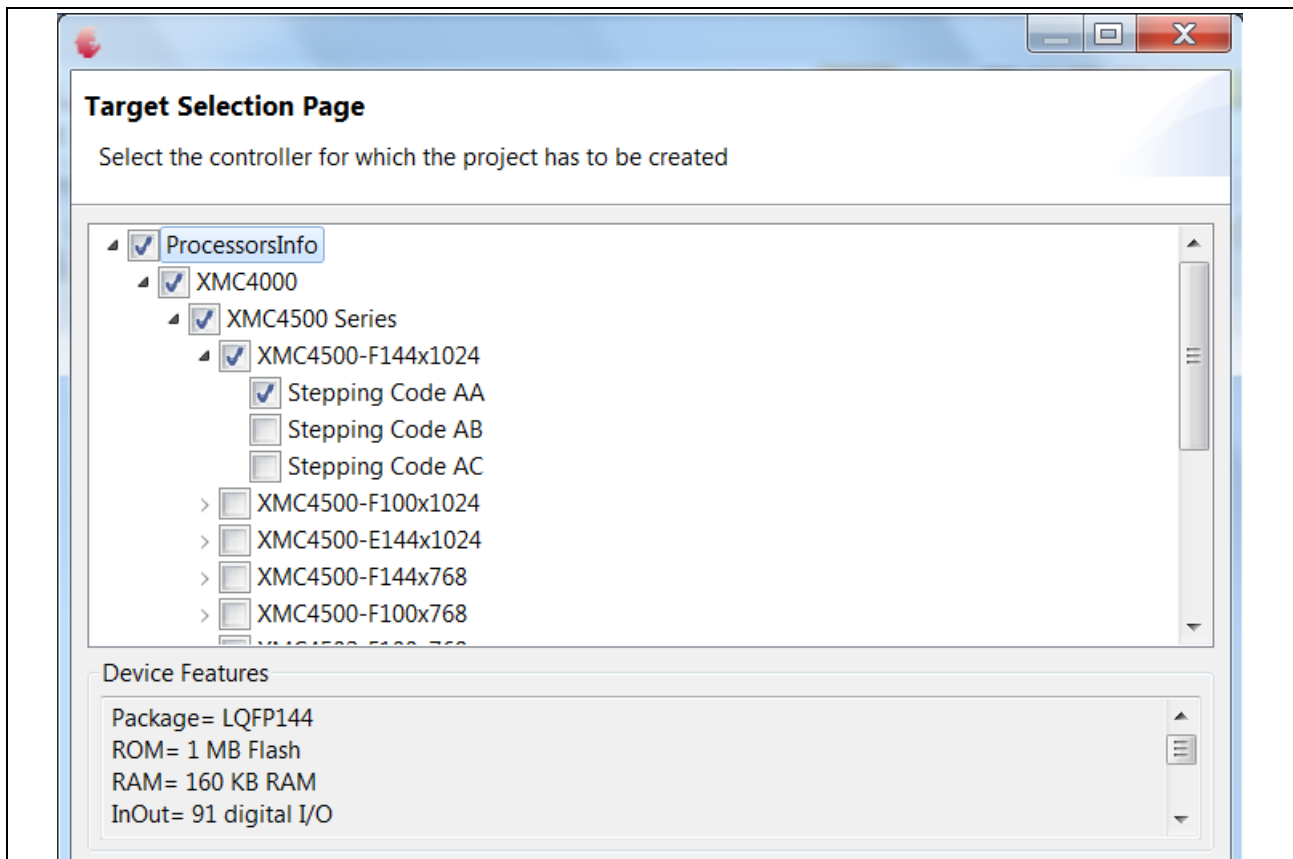
**Figure 7 DAVE™ Getting Started – New Dave Project**

- ➔ It opens Dave Project Wizard
  - Provide Project Name Ex: 'Demo'
- ➔ Select 'DAVE™ CE Project' from Infineon XMC4000, ARM-GCC Application for XMC4000 and click next`



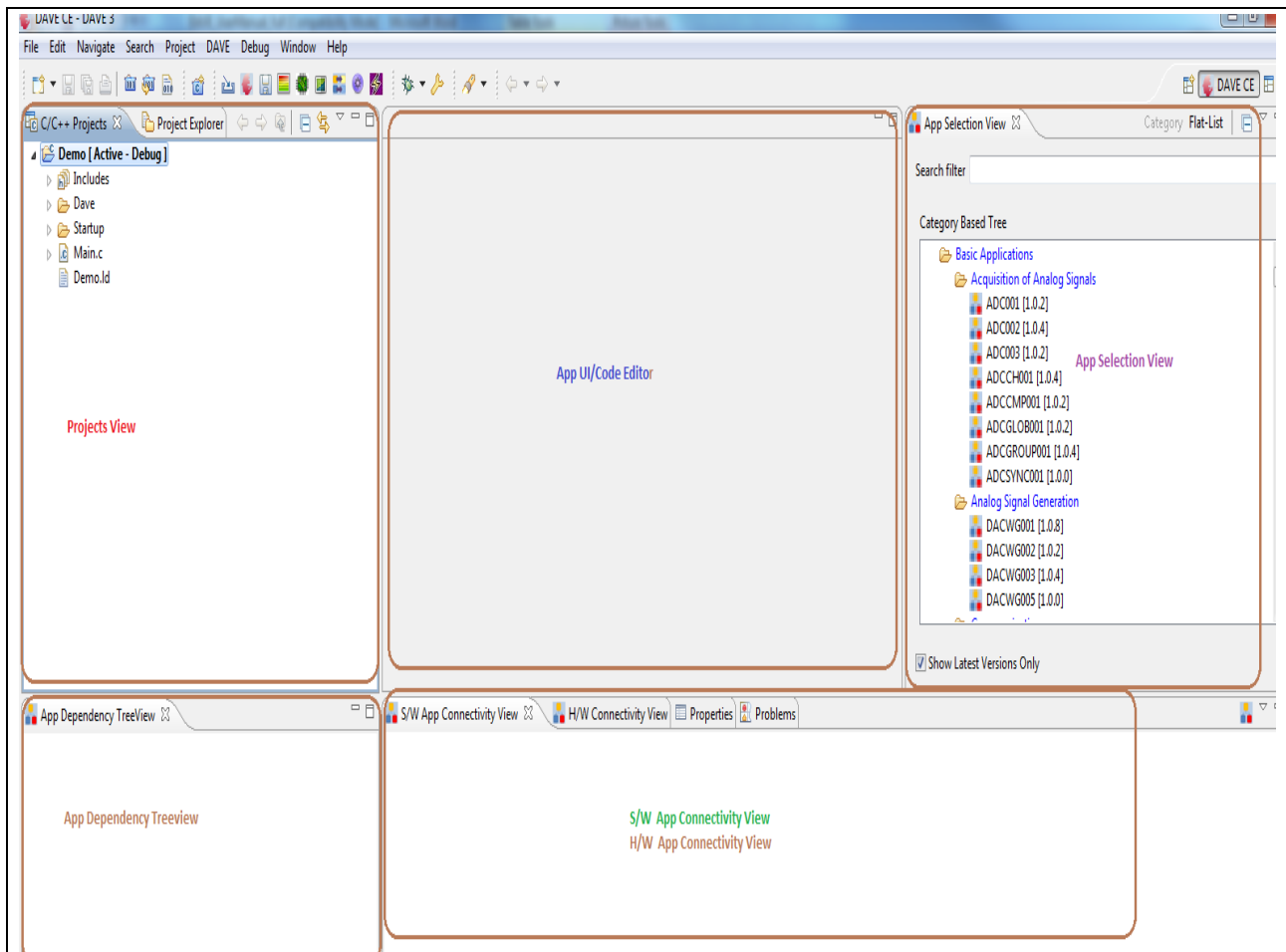
**Figure 8 DAVE™ Getting Started - Select DAVE™ CE Project and Provide Name**

- ➔ Select XMC4500-F144x1024 from Target Selection Page
- ➔ Then Click Finish.



**Figure 9 DAVE™ Getting Started - Target Selection**

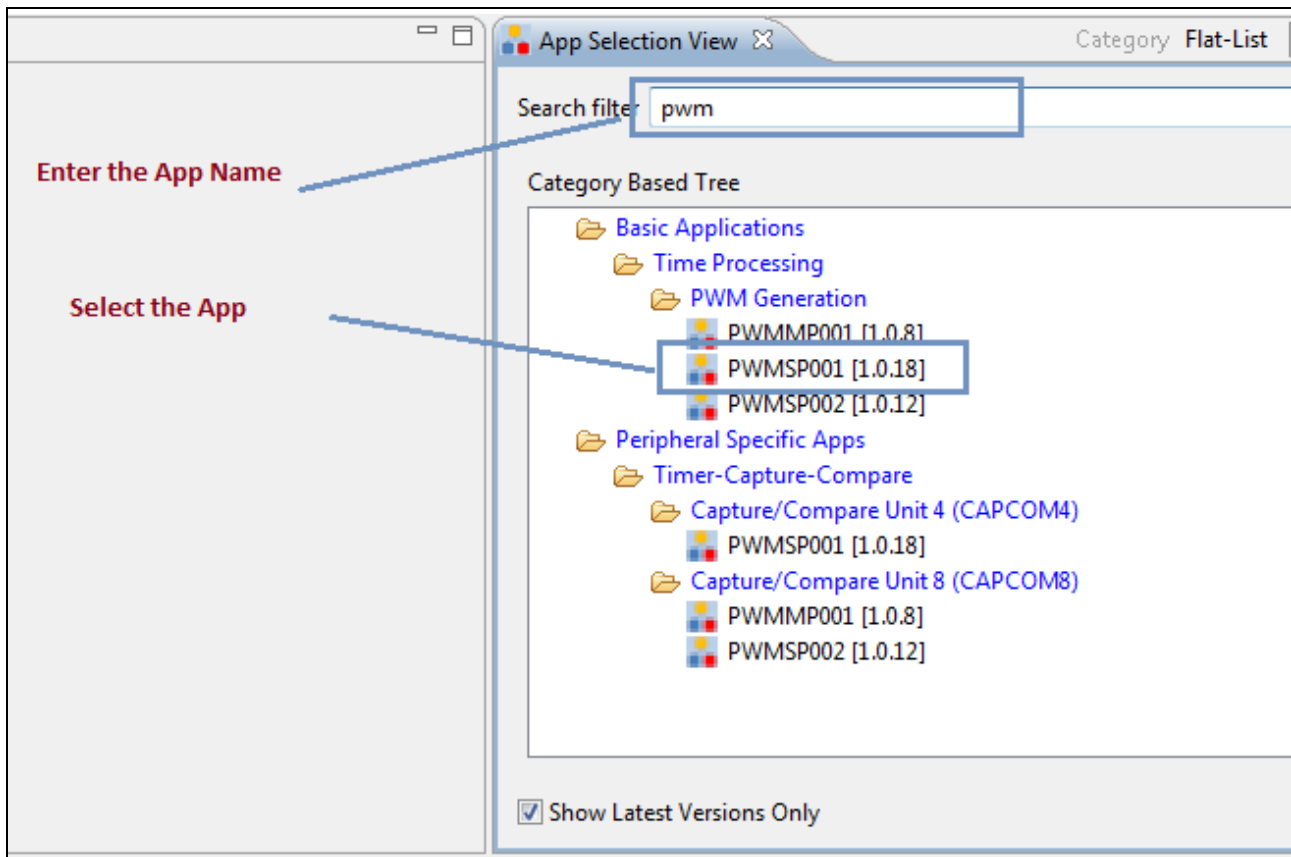
- ➔ Observe Project in Project Window and installed Apps in App selection view window
- ➔ DAVE™ CE – workspace contains the following
  - Project
  - App UI/Code Editor
  - App Selection View
  - App Dependency Tree View
  - App Dependency Graph View
  - Help/Welcome



**Figure 10 DAVE™ Getting Started – DAVE™ workspace**

All installed App are available in App Selection View window. User can select App, depending on use case

- ➔ Select PWMSP001, NVIC002 and IO004 Apps for demo example to toggle the LED using PWM compare match event
  - Below picture illustrate PWM app selection, similiary need to select other apps as well.
- ➔ Double Click on App for selection



**Figure 11 DAVE™ Getting Started - App Selection**

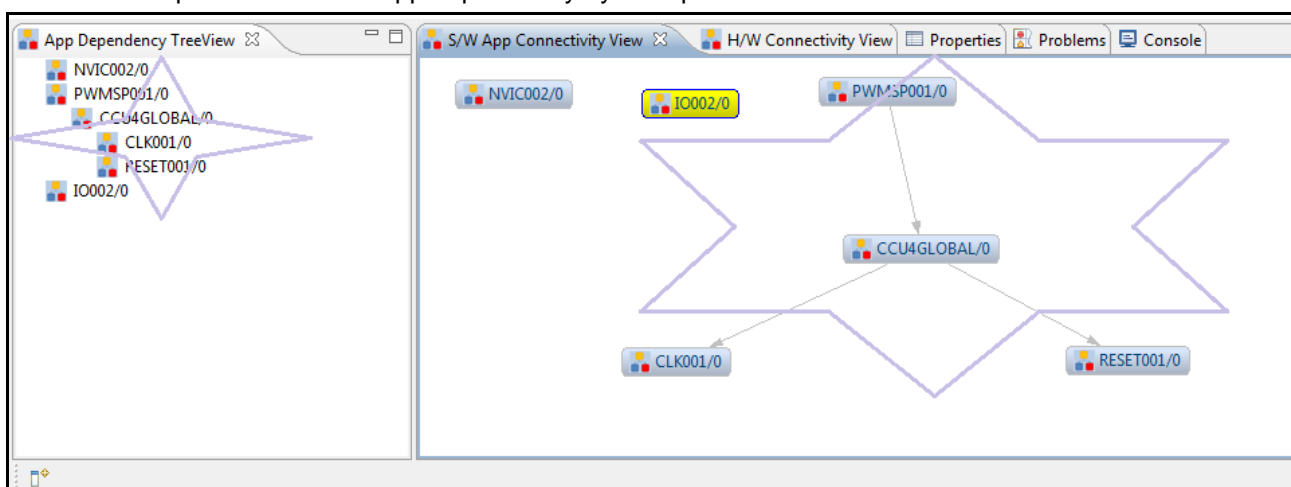
Note: Version mentioned in the figure and the installed version may differ.

### 3.5 DAVE™ App Dependency Graph View

→ The selected Apps will be displayed on App Dependency Graph View window.

This shows that the PWMSP001 App requires CCU4GLOBAL App for clock and IO002 app for output

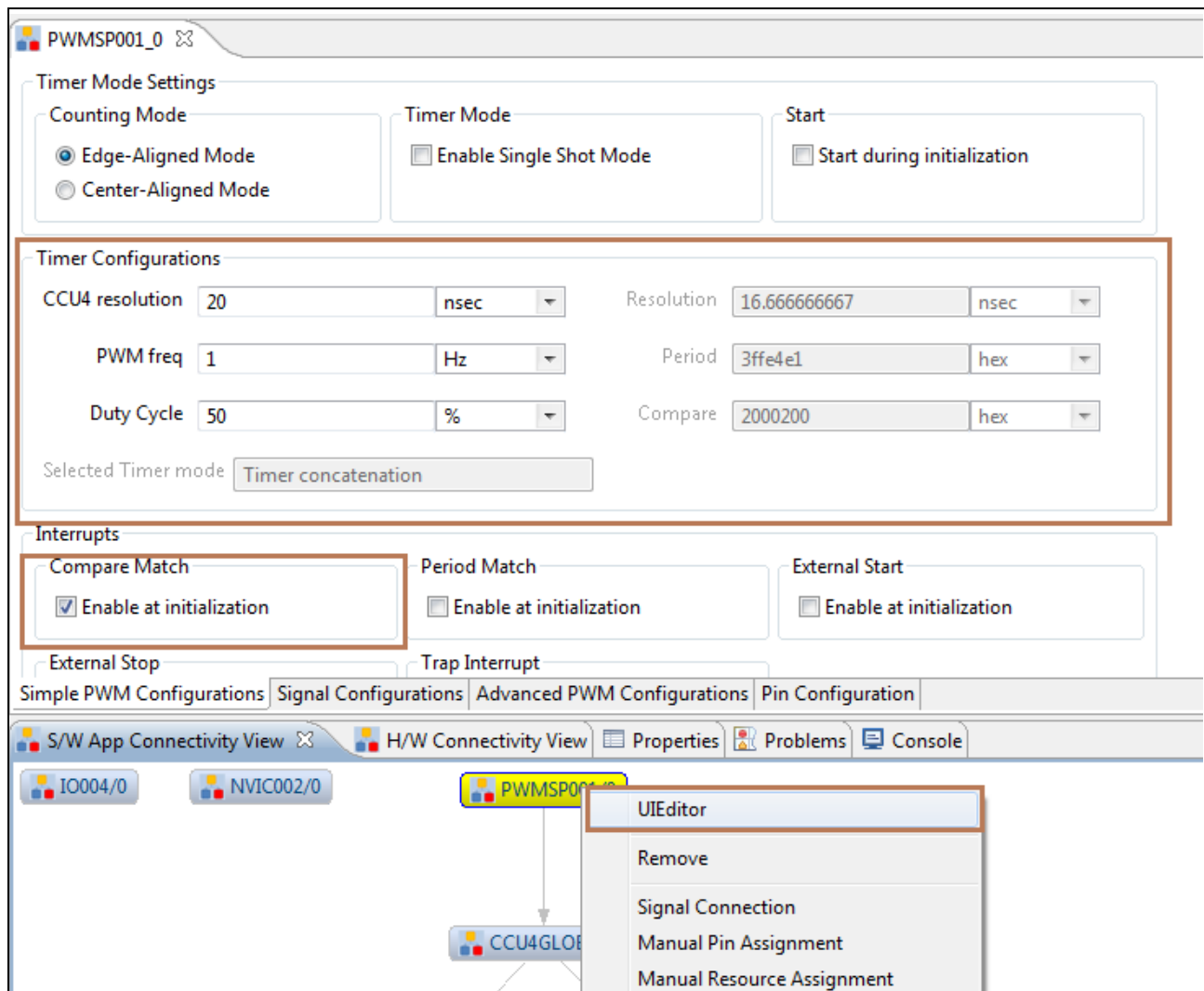
NOTE: Below picture shows all app dependency by Collapses tree & shows root node



**Figure 12 DAVE™ Getting Started - App Dependency Graph View**

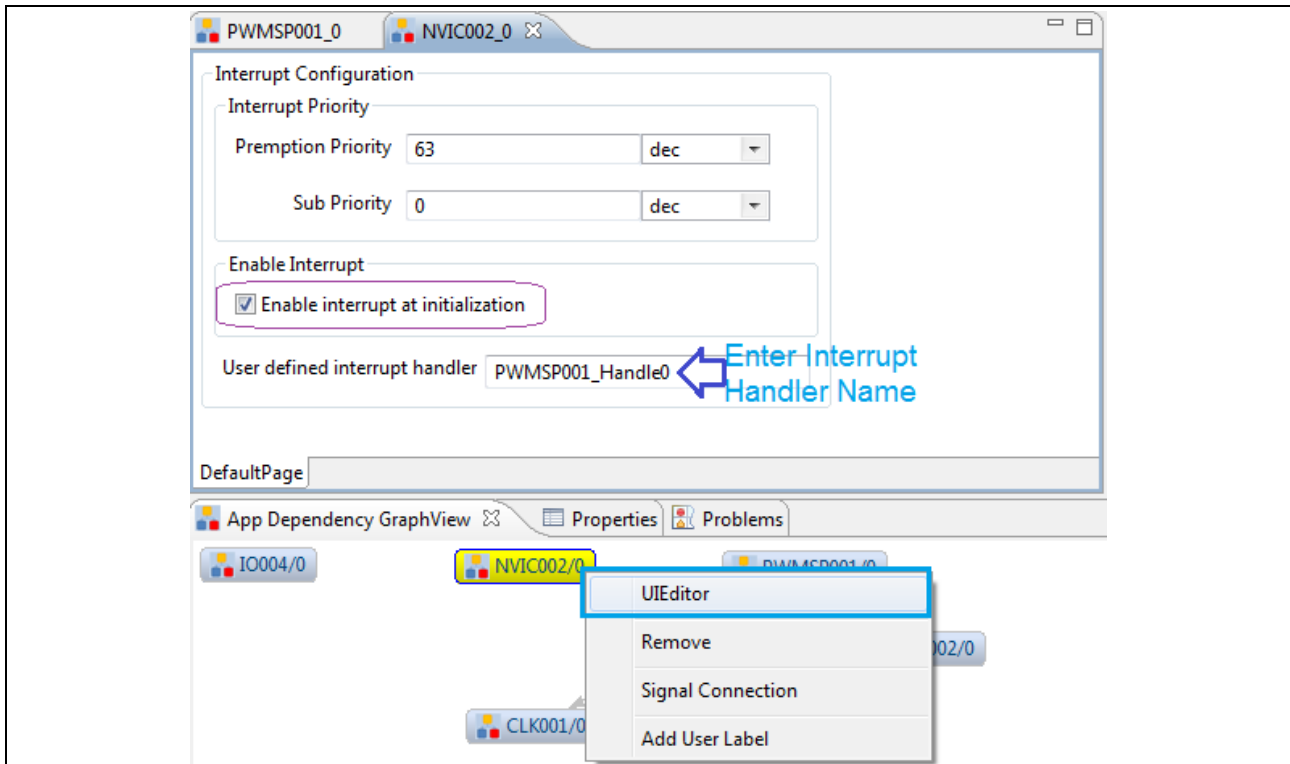
### 3.6 UI Configurations

- ➔ Select PWMSP001, NVIC002 and IO004 Apps
- ➔ Right click and select UI Editor option for respective Apps for the configuration
- ➔ UI Editor page will open
  - PWMSP001 App
    - Check that default CCU4 Resolution and Duty cycle are configured automatically.
    - Configure the PWM frequency to 1 Hz to set the PWM waveform period to 1 sec.
    - Select the Compare Match Interrupt option in the Interrupt section.



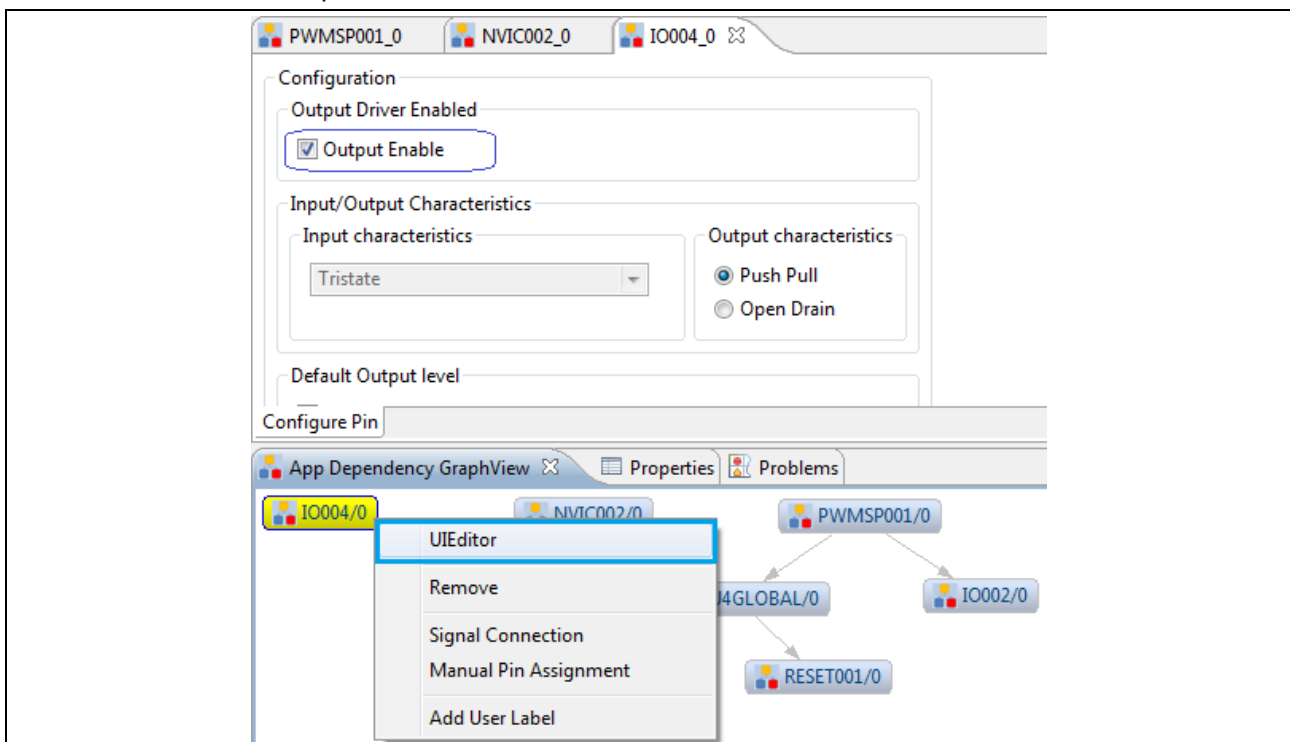
**Figure 13 DAVE™ Getting Started – UI Editor PWMSP001 App**

- NVIC002 App
  - Check that default Preemption priority is configured automatically.
  - Enable the Interrupt and provide the interrupt handler name in the space provided as shown in figure



**Figure 14 DAVE™ Getting Started – UI Editor NVIC002 App**

- IO004 App
  - Check Output Enable



**Figure 15 DAVE™ Getting Started – UI Editor IO004 App**



### 3.7 Signal Connection

- ➔ Click on Signal connection Menu button for signal connection of PWM interrupt and NVIC App
- ➔ Signal connection window will open
- ➔ Perform the signal connection by selecting the PWM Compare Match Interrupt signal from PWMSP001 App and the Destination app as NVIC002 App and signal as NVIC Interrupt.
- ➔ Click on Solve and Save button to execute the solver to validate the configurations made. (Solver execution is taken care internally by the tool).

Refer figure below for the Signal configurations.

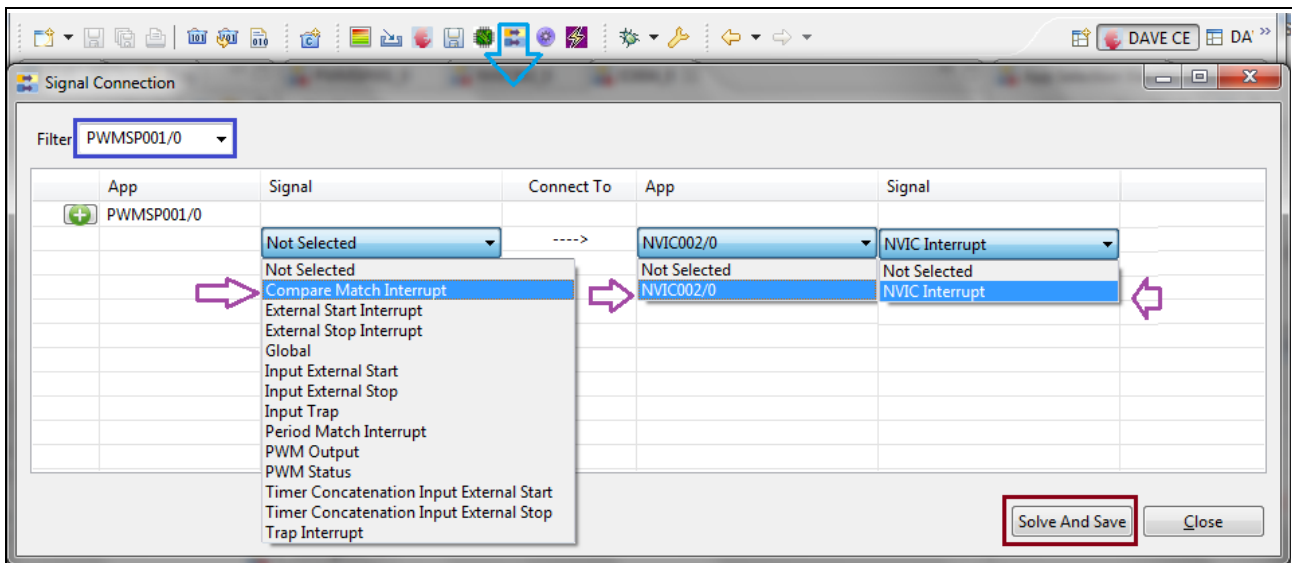


Figure 16 DAVE™ Getting Started – Signal connection

- ➔ After solve and save, for the signal configuration changes the final application of PWM will be as shown in the below figure in H/W Connectivity view.

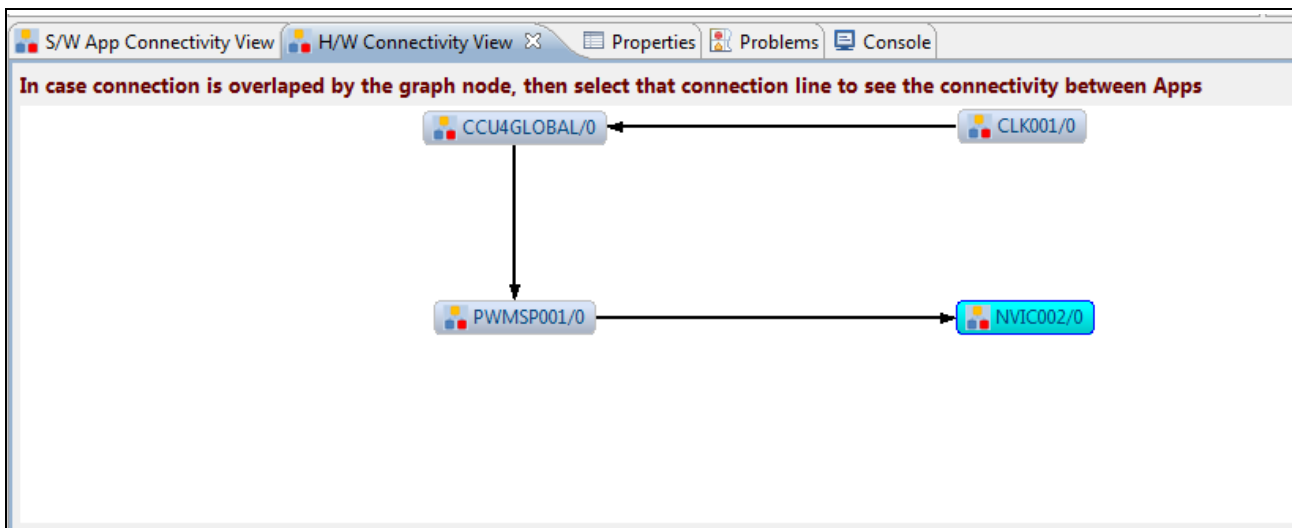


Figure 17 DAVE™ Getting Started – H/W Connectivity View after Signal connection

### 3.8 Manual Pin Assignment

- ➔ Select Manual Pin Assignment menu.
- ➔ It Opens Manual Pin Assignment Window
  - IO004/0 - Select pin from Resource Name and Pin P3.9/#12 from Port-pin/Pin Number list.
  - Then click 'Solve and Save'

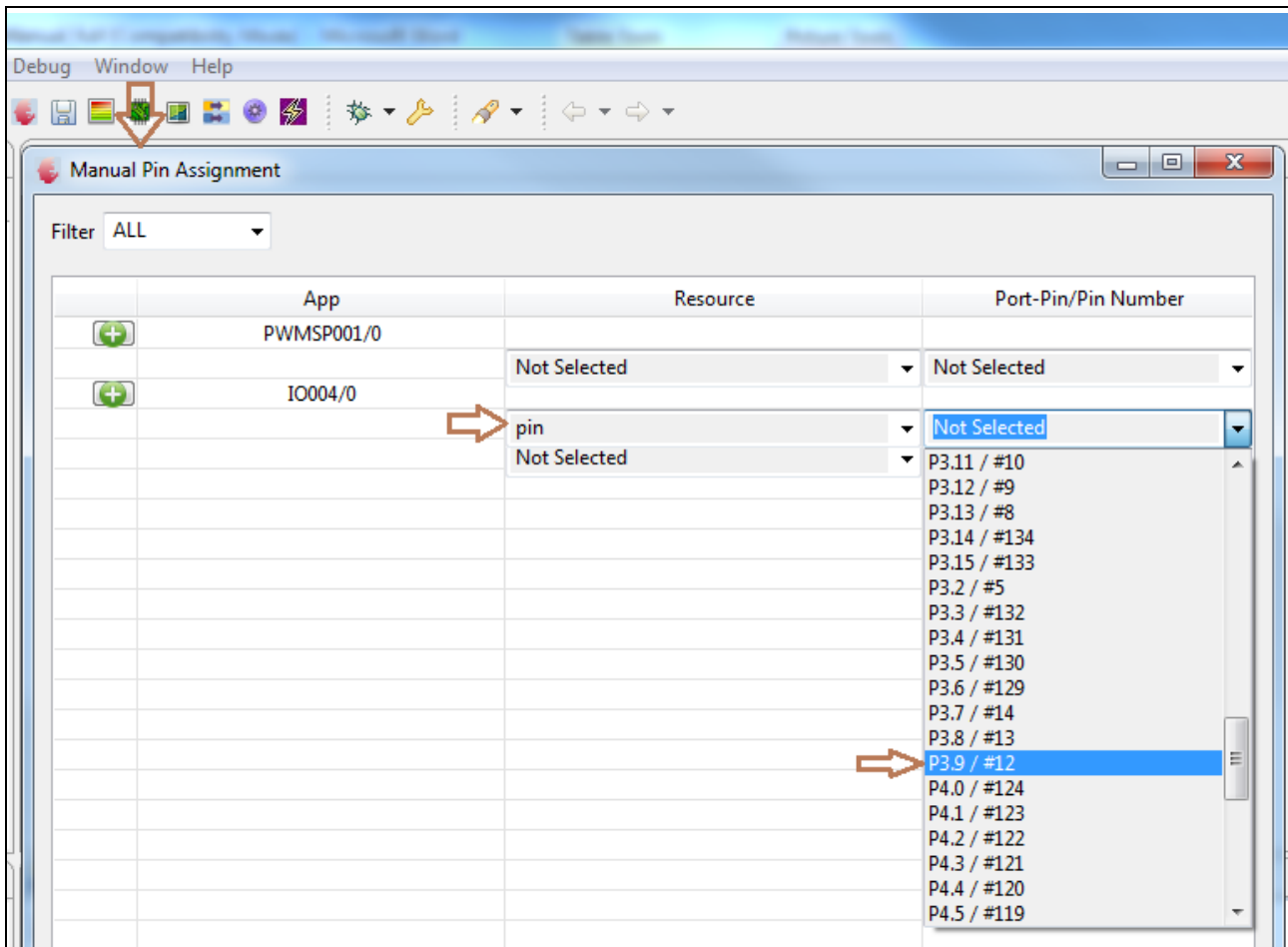


Figure 18 DAVE™ Getting Started – Manual Pin Assignment for IO004

### 3.9 Report

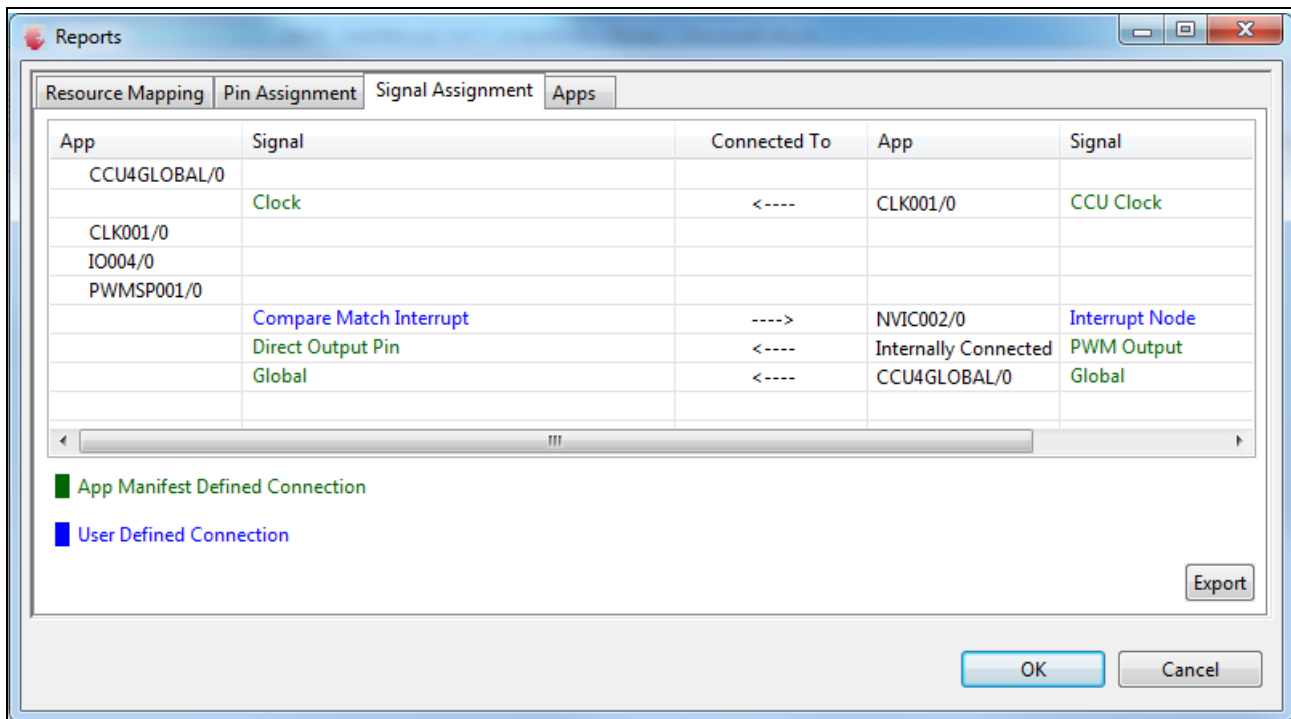
- ➔ Click on Resource Binding information menu to know about App Consumed Resource and other information
  - The Apps tab will display the used Apps with version for the use case

[illegible]

**Figure 19 DAVE™ Getting Started - Report – Apps with version**

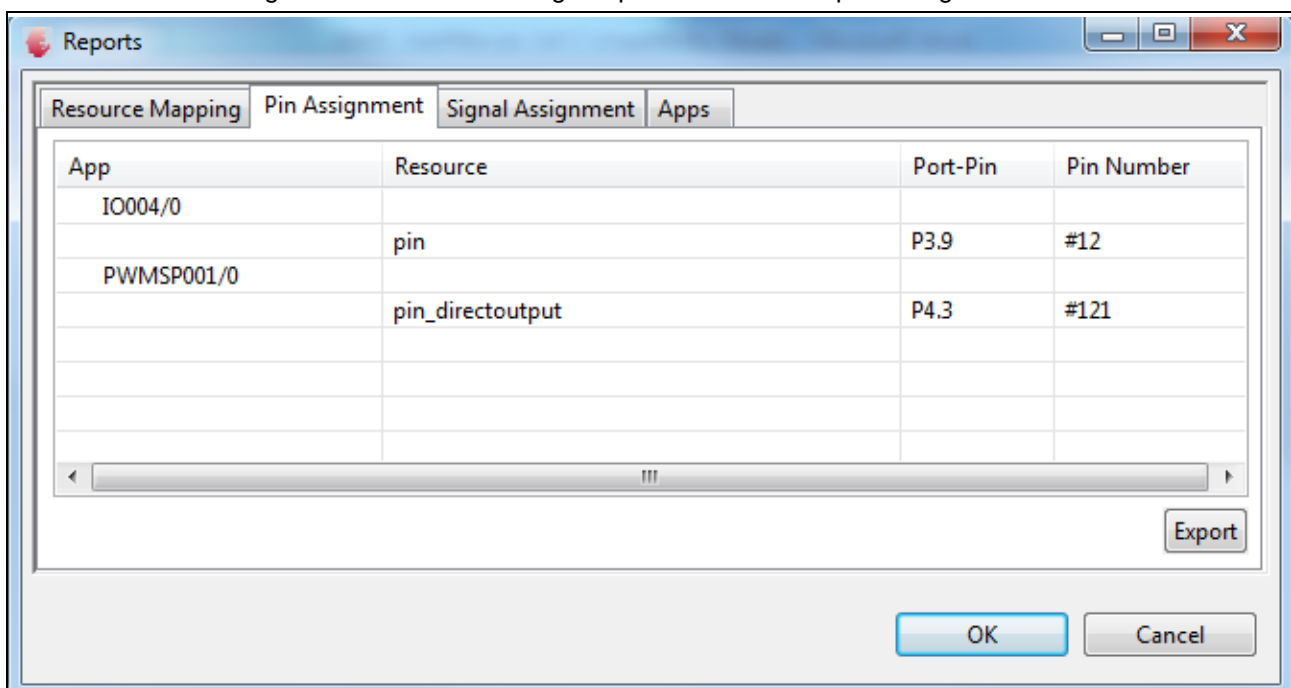
Note: App Name/\* indicates the instances of the the App e.g. PWMSP001/0, PWMSP001/1...

- ➔ Click Signal Assignment tab to know assigned signal information
- ➔ The Blue color fonts indicate that the signal connection is manually performed. Whereas Green color fonts indicate that the signals are connected internally through App Manifest.



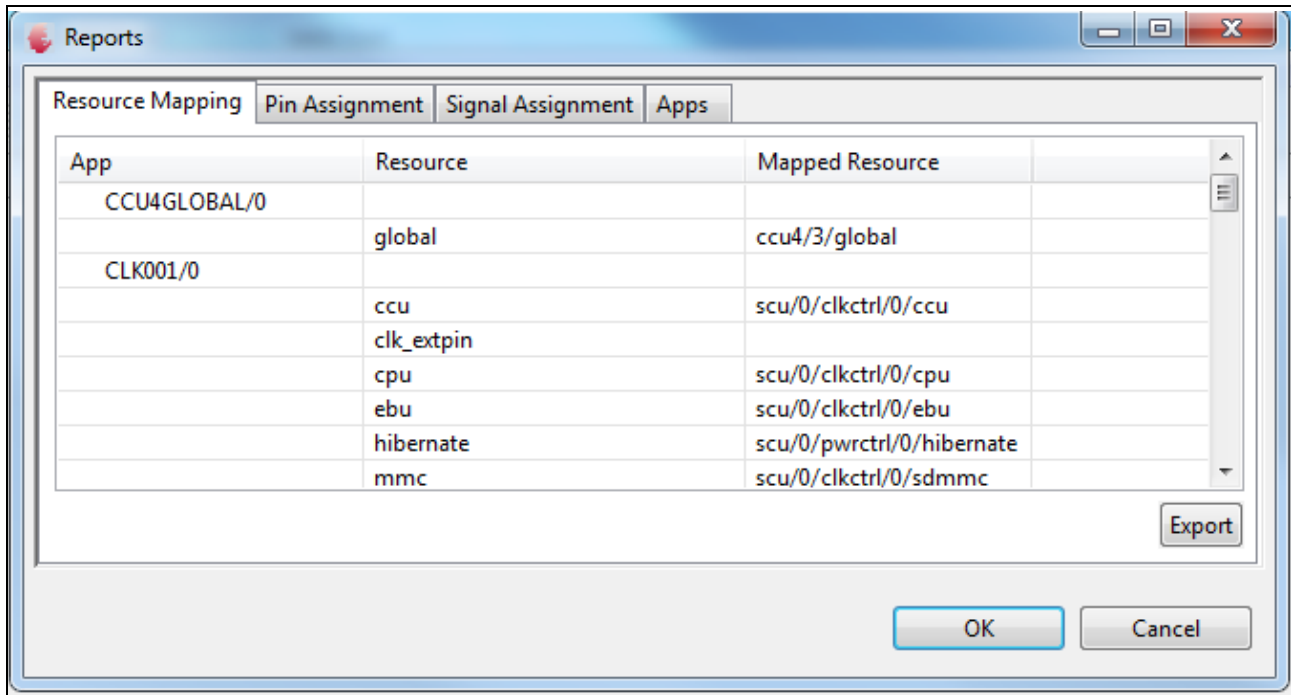
**Figure 20 DAVE™ Getting Started - Report – Signal Assignment**

→ Click Pin Assignment tab to know assigned pin information as per configurations



**Figure 21 DAVE™ Getting Started - Report – Pin Assignment**

- Click on Resource Mapping tab to know about the hardware resources used by the apps.



**Figure 22 DAVE™ Getting Started - Report – Resource Mapping**

### 3.10 Code Generation

- ➔ Click Generate Code Menu button to generate code. By using this option, solver solution is also taken care internally by the tool.
- ➔ All generated \*.h and \*.c files are added to the project
- ➔ Unfold the “Dave” folder
- ➔ Generated file contains both Source and Include files in src & inc folder respectively.

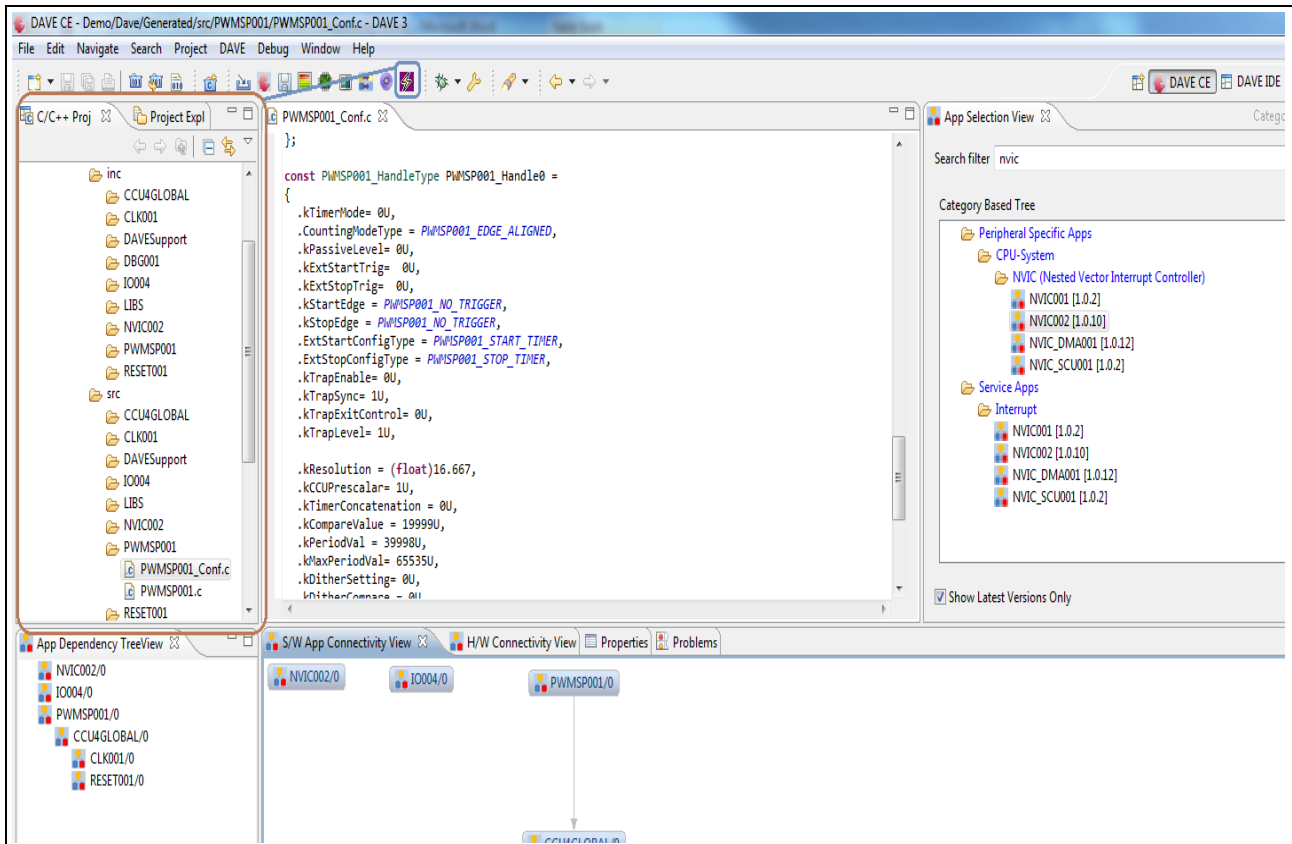


Figure 23 DAVE™ : Getting Started - Code Generation – PWMSP001\_Conf.c

### 3.11 Generated Files and Edit main.c

➔ Edit main.c [for Interrupt Handler Interface & toggling LED On-board]

#include

```

/*****
** INCLUDE FILES
*****/
#include <XMC4500.h> //SFR declarations of the selected device
#include <DAVE3.h> //Declarations from DAVE3 Code Generation

```

Add code for Pin P3.9 for toggling LED and UART send data

```

// User defined code START
uint8_t CmpMatEvtStatus;

void PWM_IRQHandler(void)
{
    //Compare Match
    PWMSP001_GetPendingEvent(&PWMSP001_Handle0, PWMSP001_COMPAREMATCHEVENT,
        &CmpMatEvtStatus);
    if(CmpMatEvtStatus == (uint8_t)SET)
    {
        // LED Toggle code
        I0004_TogglePin(I0004_Handle0);

        //Clear the event flag
    }
}

```

```

        PWMSP001_ClearPendingEvent(&PWMSP001_Handle0,
                                   PWMSP001_COMPAREMATCHEVENT);

        //reset the software flag
        CmpMatEvtStatus = (uint8_t)RESET;
    }

}

// User defined code END

int main(void)
{

    DAVE_Init(); // Initialization of DAVE™ Apps

    while(1)
    {
        // User code
    }

    return 0;
}

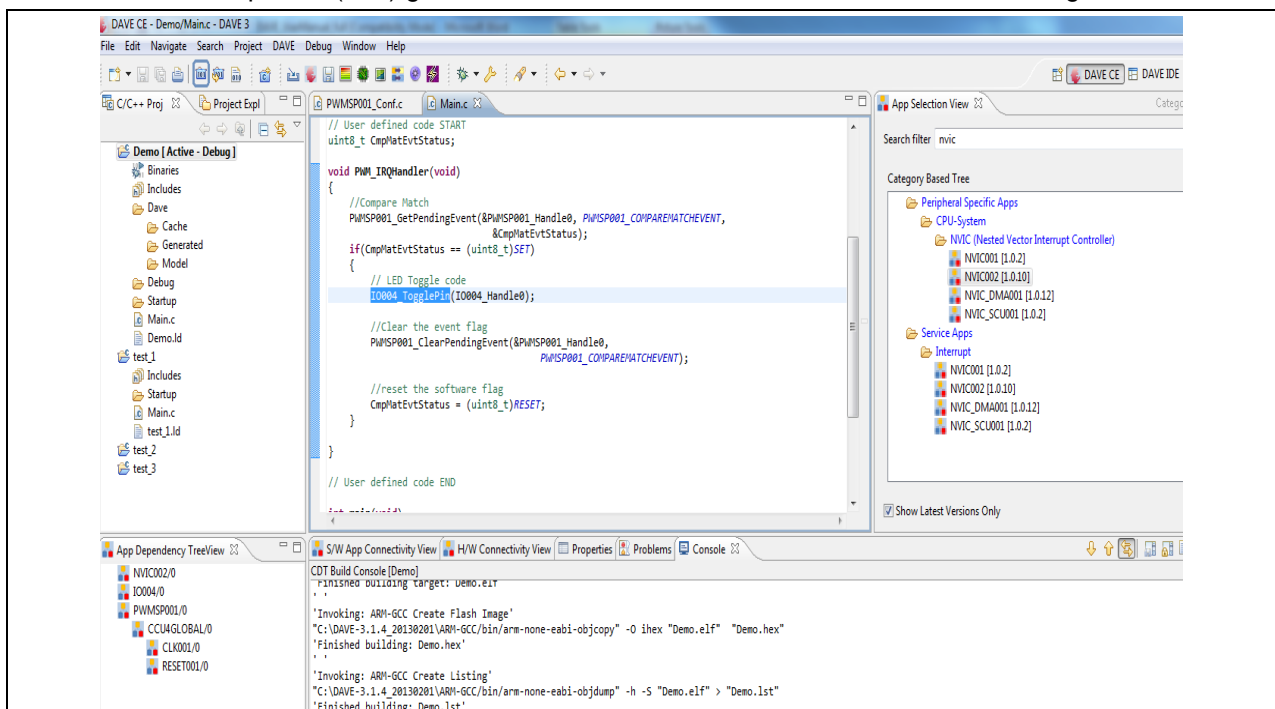
```

### Code Listing 1 C – Code for toggling LED upon PWM compare event

Note: If user edits any of the generated files in dave folders (either inc or src folders), they will be overwritten by the tool upon regeneration of the code.

## 3.12 Build the Project

- ➔ Click Build Active Project menu to compile, link and generate output file for downloading to the board.
- ➔ Observe output file (\*.elf) generated in CDT Build Console after successful building.



**Figure 24 DAVE™ Getting Started - Build Project & Generate output file.**

Note: The output file may be \*.elf, \*.hex or \*.out etc... as per the linker used for building the project.

### 3.13 Target Board Setup

- ➔ Connect the Target board to PC using USB cable.

Note: Refer Board User Manual from [www.infineon.com](http://www.infineon.com) for more information about the board details and connections.



Figure 25 DAVE™ Getting Started - Board Setup

### 3.14 Download / Debug

The download/ Debug process can be achieved by using Infineon MiniWiggler or by JLINK connection.

In this section, Download/ Debug process is explained by using JLINK.

The Process using Miniwiggler is explained in later sections.

- ➔ Click Debug menu to start download/debug the code on target board

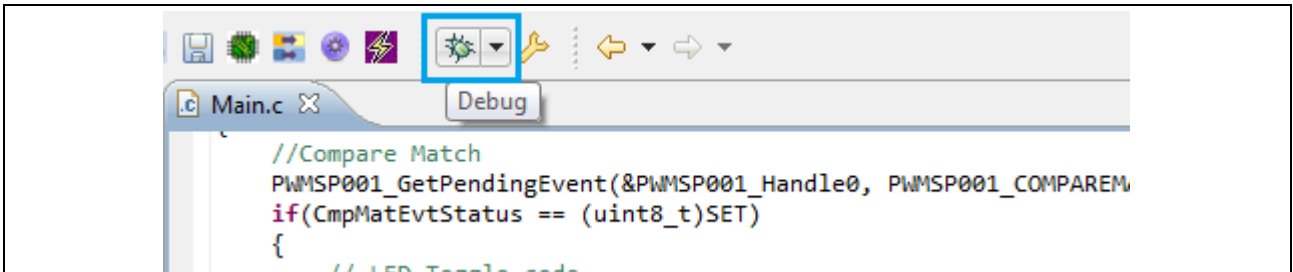
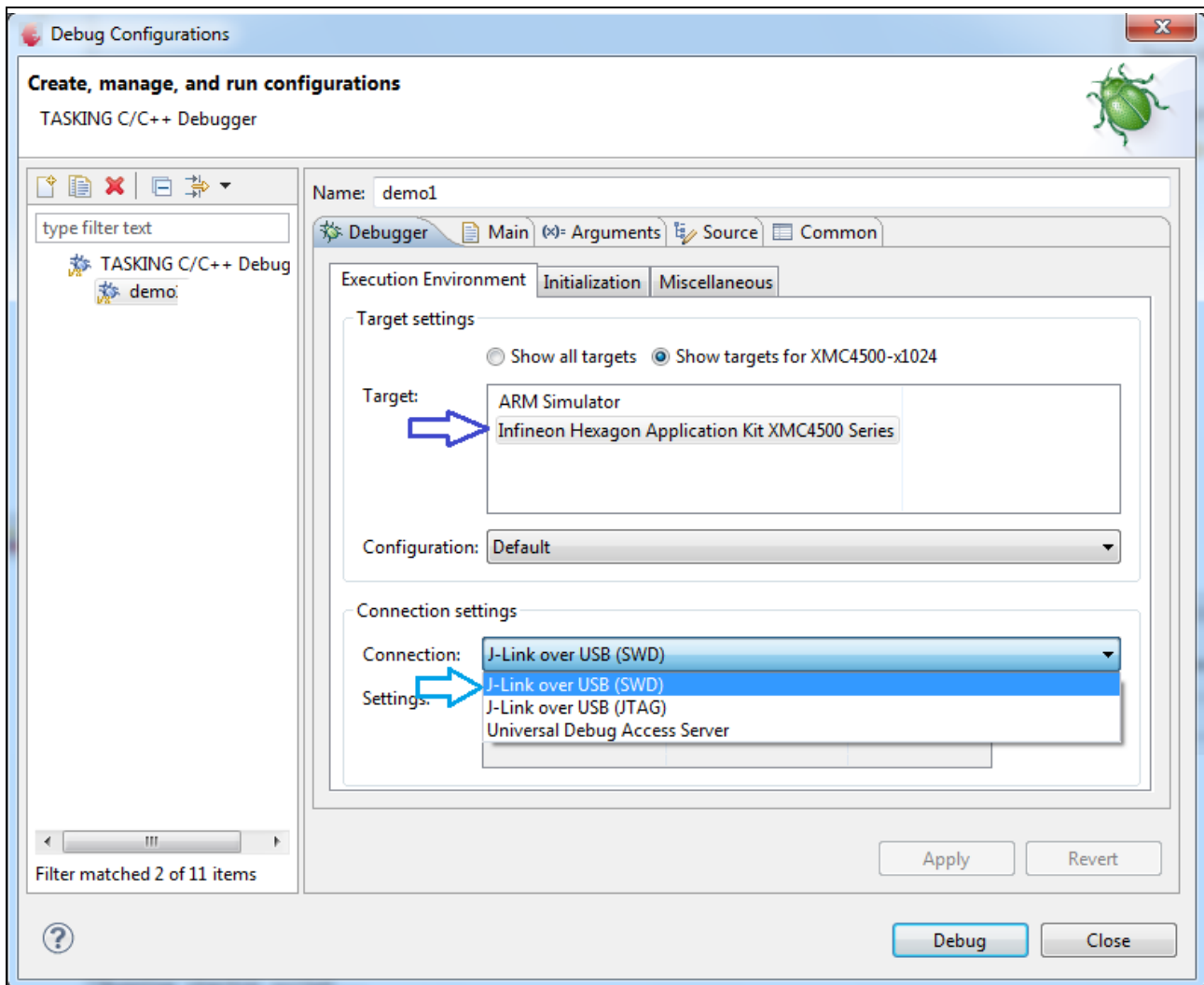


Figure 26 DAVE™ Getting Started - Debug Setup-1

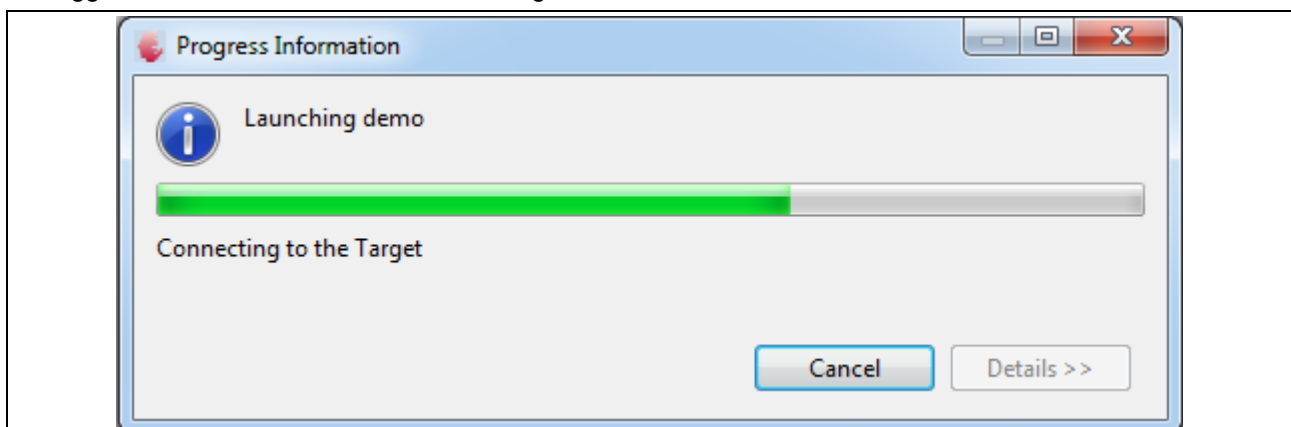
- ➔ Click TASKING C/C++ Debug to create one instance of debug
- ➔ Check Debugger tab
- ➔ Select Communication Setup
- ➔ Click Debug button for Debug





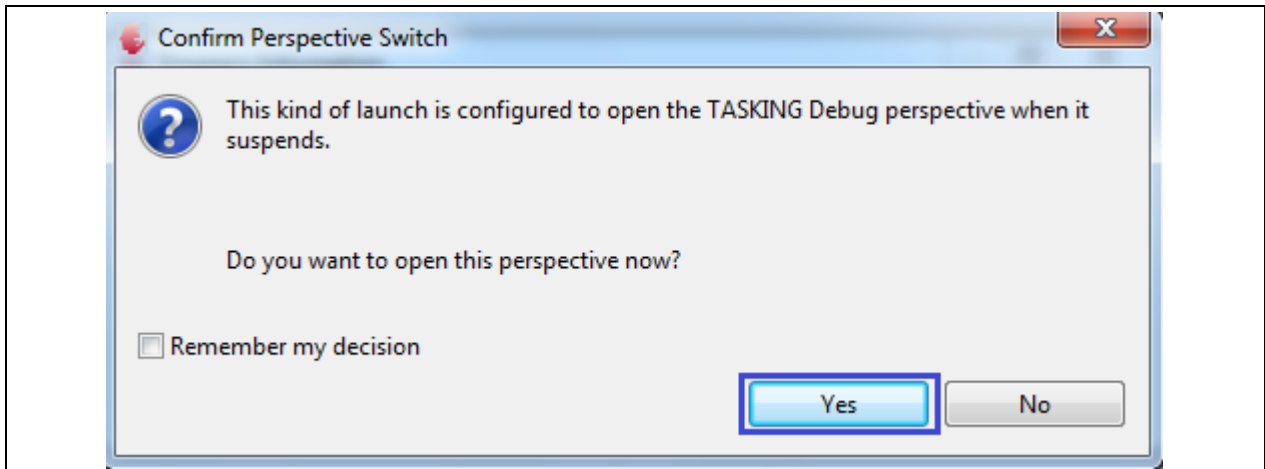
**Figure 27 DAVE™ Debugger Support – JLink JTAG/SWD options**

Debugger will start Download binaries to Target Board.



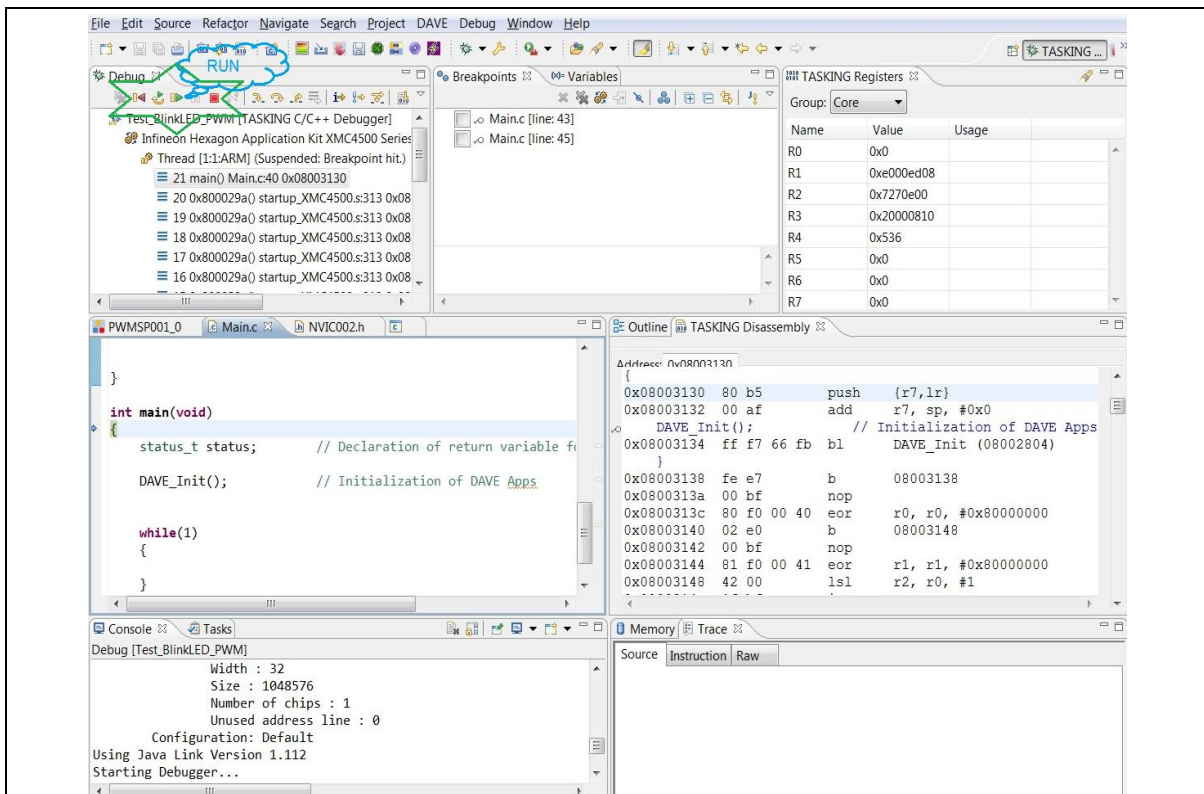
**Figure 28 DAVE™ Download to Target Board**

➔ To open Debugger Perspective Press 'Yes' button



**Figure 29 DAVE™ Download to Target Board – Confirm Perspective Switch**

- ➔ Click “Run” button from Tasking Debug perspective
- ➔ Debug features supported are
  - To Debug Variable, Breakpoint, Registers
  - C File/editor
  - Disassembly
  - Console
  - Memory/ Register



**Figure 30 DAVE™ Download to Target Board – Debug Launch**

Note: More Information of Apps and App Help are available in section 8.

## 4 Project Management

### 4.1 DAVE™ Perspectives

The Dave Workspace supports more than one perspective, which has the different views and editors.

The main perspectives which will be used by the user regularly are,

- DAVE™ IDE (default)
- DAVE™ CE
- Debug
- Tasking Debug

#### DAVE™ IDE

- On creation of new DAVE™ workspace, by default DAVE™ IDE perspective will be opened.
- This perspective is mainly used for the creation of projects, building the project etc...

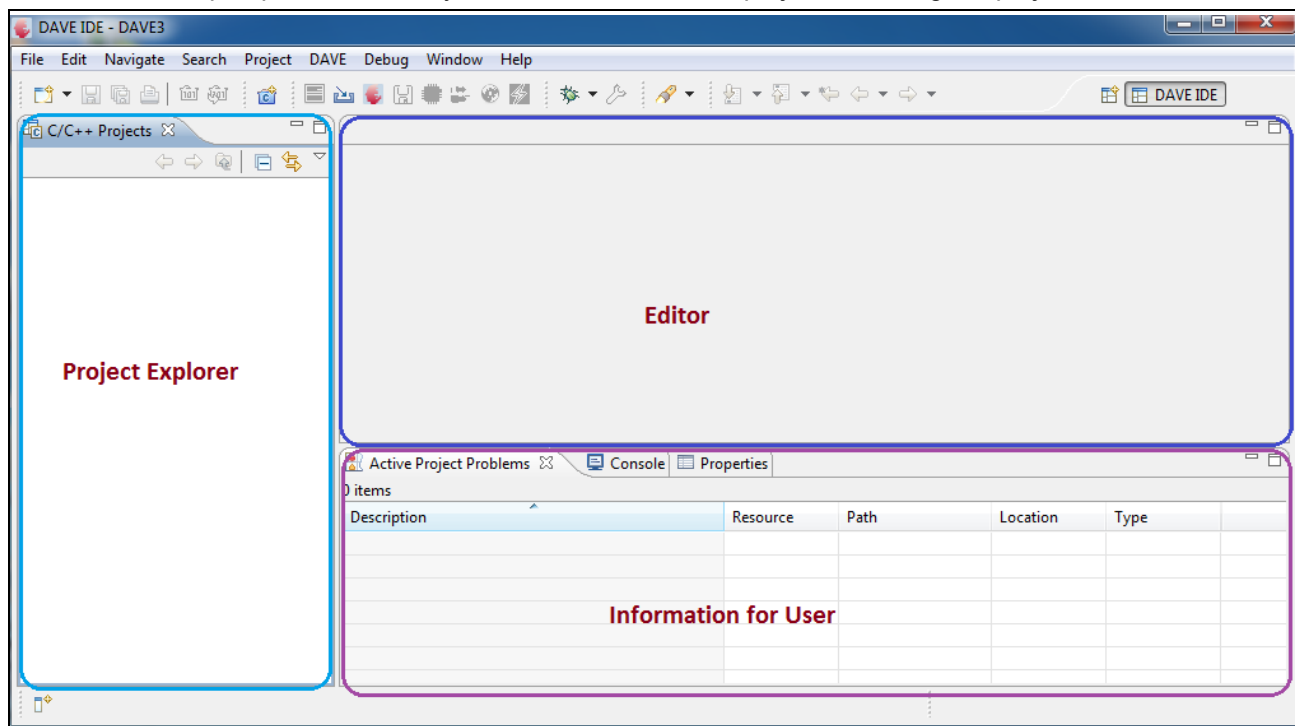
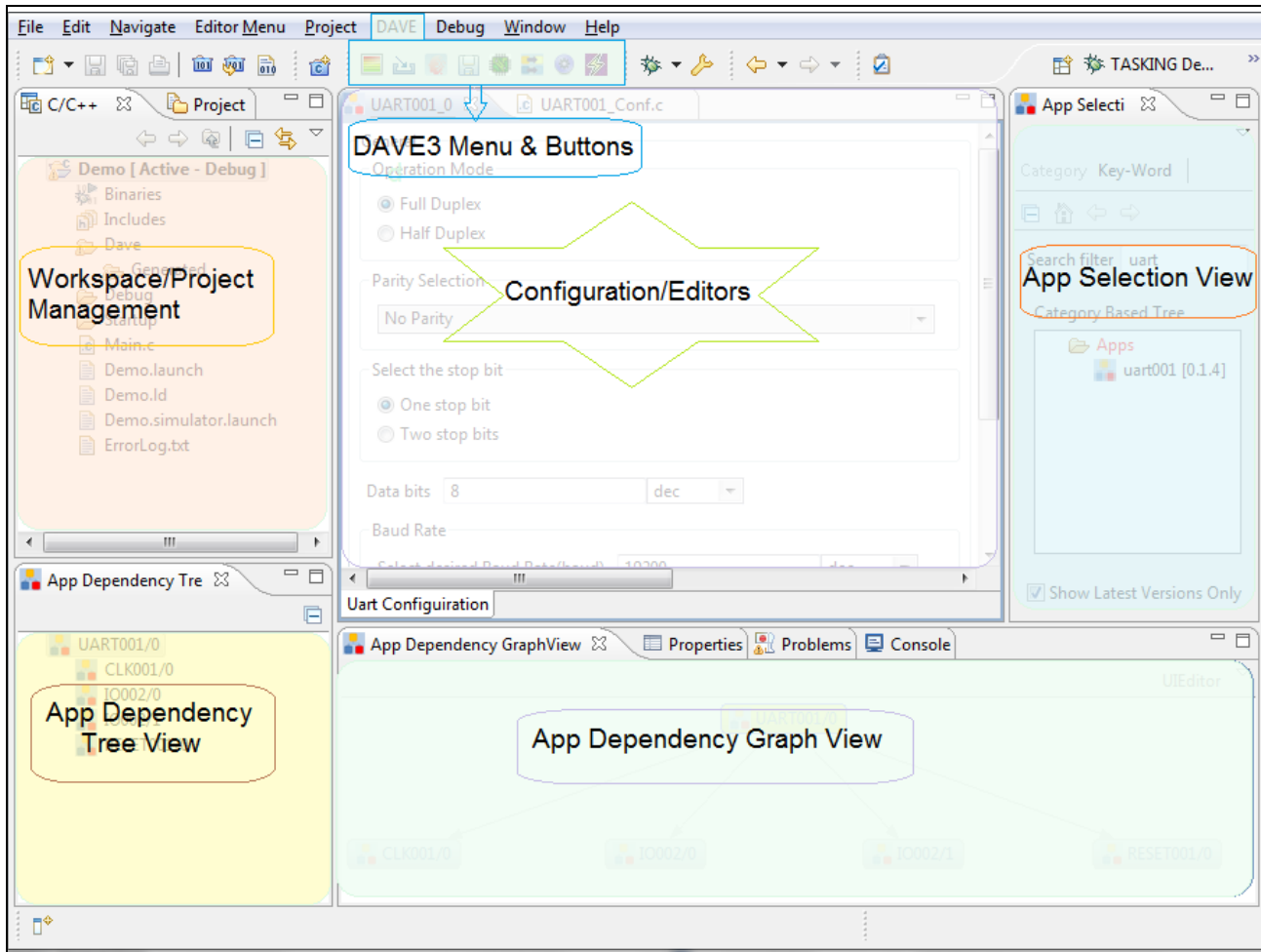


Figure 31 DAVE™ DAVE™ IDE perspective

#### DAVE™ CE

- This is the main perspective of DAVE, where major tasks will be performed.
- This perspective is used for App selection, binding, Pin configurations and other tasks



**Figure 32 DAVE™ DAVE™ CE perspective**

## 4.2 New DAVE™ Project creation

DAVE™ supports different types of projects for the user to create from the options in new project wizard.

The different type projects are,

DAVE™ Project – ARM-GCC Application for XMC4000 [DAVE™ CE Project]

- ➔ Creates an DAVE™ CE Project and ready to use without any additional task

DAVE™ Project – ARM-GCC Application for XMC4000 [Easy Start Project]

- ➔ Creates an XMC4000 Empty Project. User need to add void main (void) routine and respective codes

DAVE™ Project – ARM-GCC Application for XMC4000 [Empty Main Project]

- ➔ Creates an XMC4000 Project with void main (void) routine. User need to add respective codes

DAVE™ Project – ARM-GCC Application for XMC4000 [Empty Project]

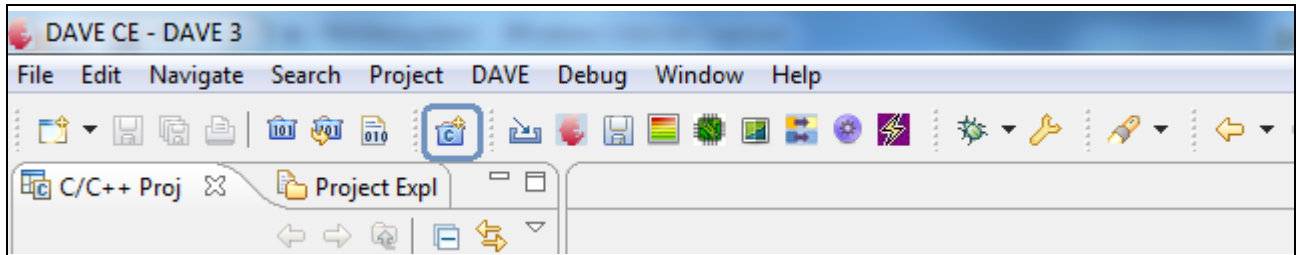
- ➔ Creates an XMC4000 Empty Project

DAVE™ Project – ARM-GCC Library for XMC4000 Project [Empty Project]

- Creates an XMC4000 Empty Project

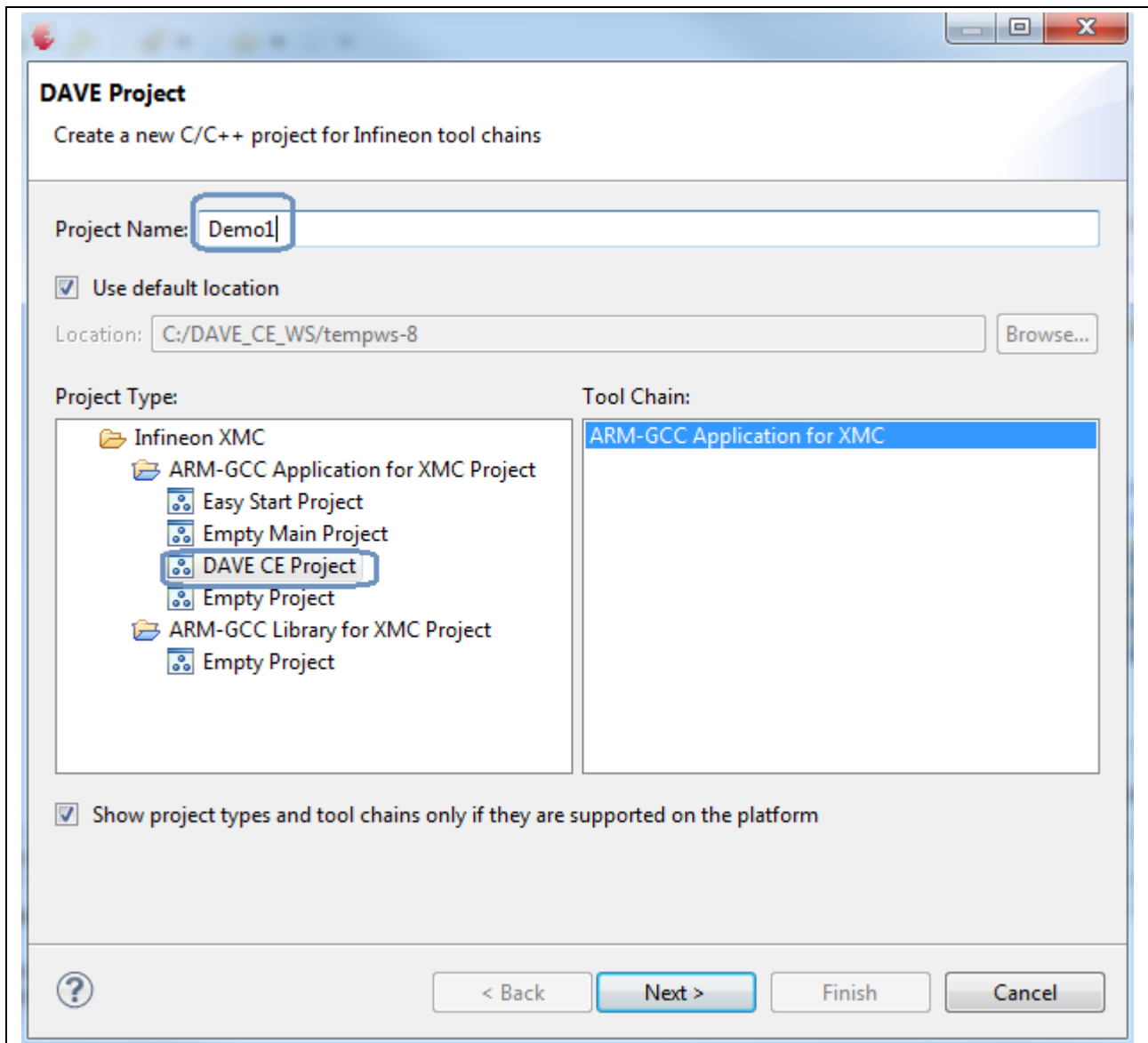
The steps to create the project is as mentioned below,

- Create new project by Selecting DAVE™ IDE New Project Wizard Menu



**Figure 33 DAVE™ Project Management – New Project Creation**

- Provide Project name e.g.: Demo
- Select Empty Main Project from ARM-GCC Application for XMC4000 from the list



**Figure 34 DAVE™ Select Empty Main Project and Provide Project Name**

- ➔ Select XMC4500-F144x1024 from Target Selection Page
- ➔ Select Start up file Preferences to Add/ Update startup files for the created project
- ➔ Then Click Finish.

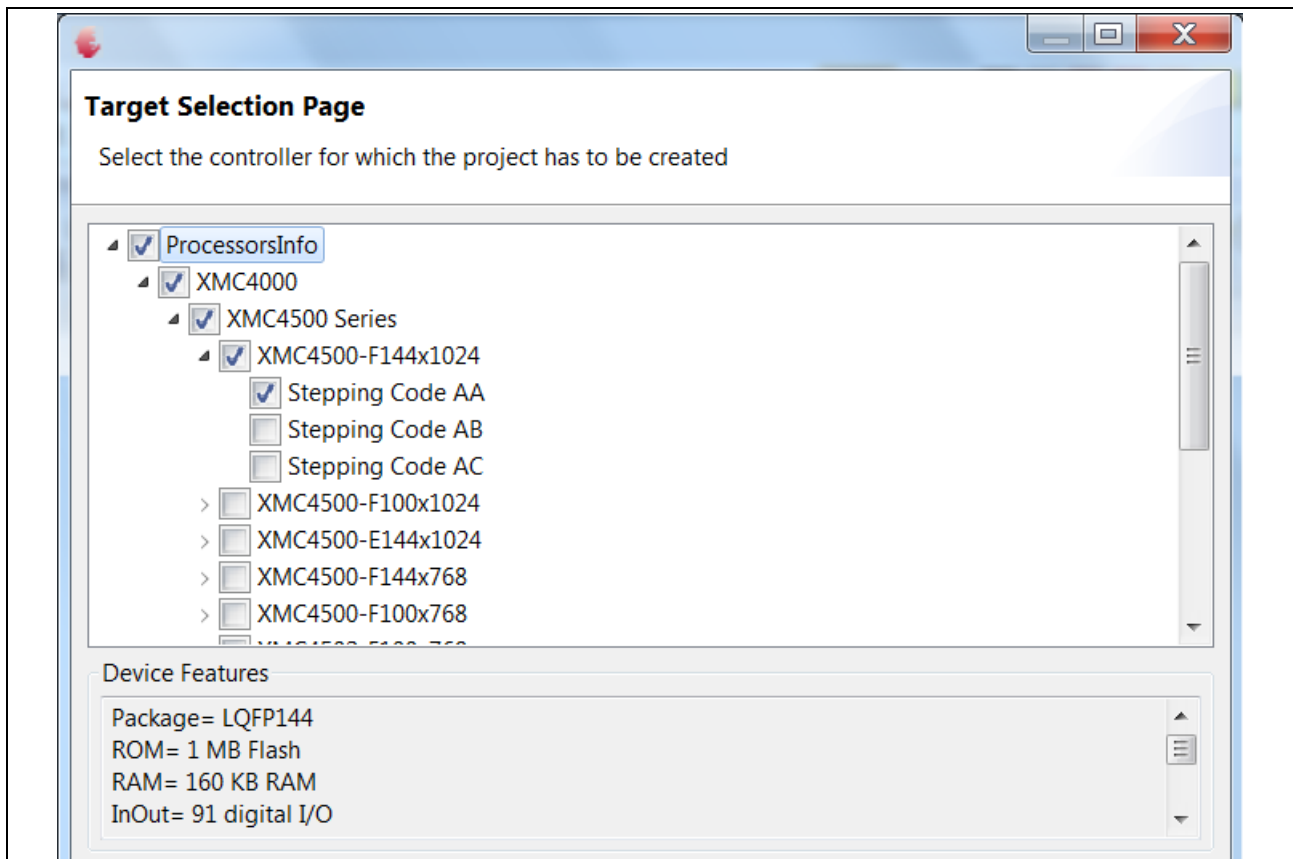


Figure 35 DAVE™ Getting Started - Target Selection Page

### 4.3 Upgrade to DAVE™ CE project

- Click on 'Upgrade to DAVE™ CE project' to convert the created project to DAVE™ CE project. By doing this all installed Apps will lists in App Selection View window

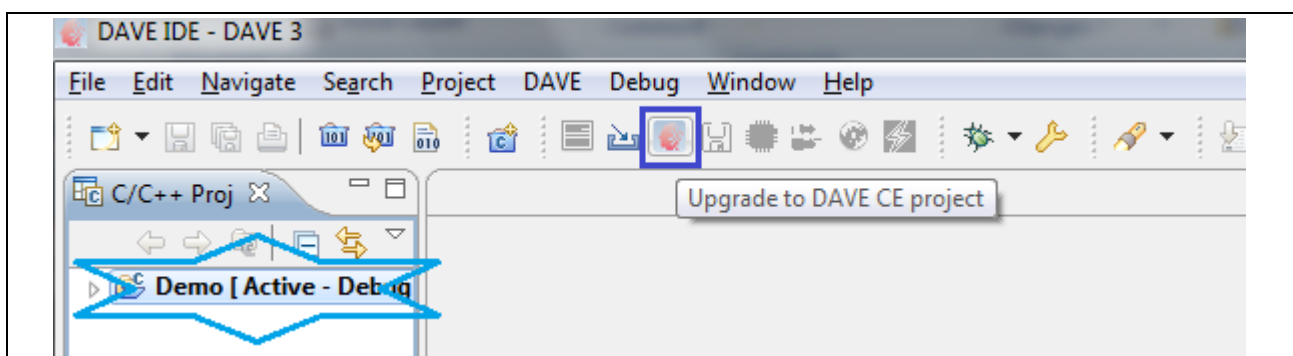


Figure 36 DAVE™ Upgrade to DAVE™ CE project

Note: Upgrade to DAVE™ CE project is not applicable, when DAVE™ CE project is created.

### 4.4 Active Project

In projects window there may be more than one project. Among them one project will be Active. By default newly created project will be Active Project.

Note: In order to change Active project for another project, select the project, right click and select 'Set Active Project'.

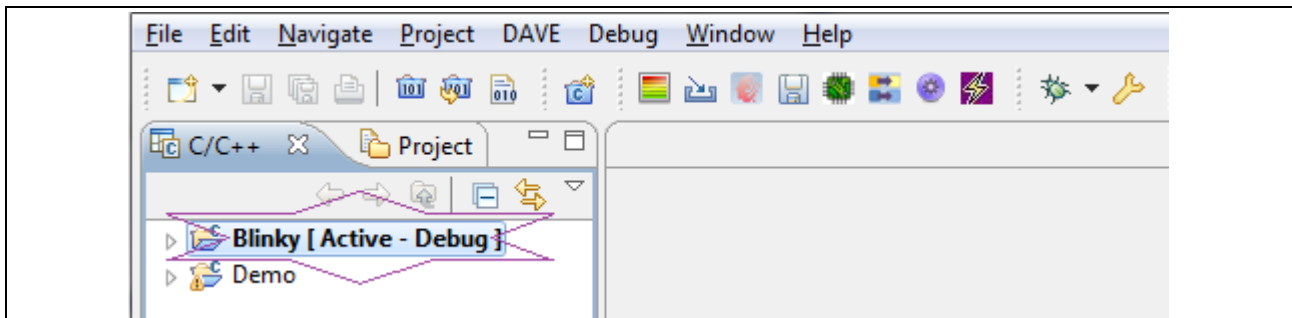


Figure 37 DAVE™ Project Selection View – Active Project

## 4.5 Importing/Exporting of Projects

Import & Export of the project helps to copy the projects from source to destination or vice versa.

Note: The Import and Export procedure is as per Eclipse IDE.

### 4.5.1 Import - Projects

Import: helps to import/getting back the projects which are available. This can be done using various methods:

- ➔ Right click on the Projects Window Select Import option

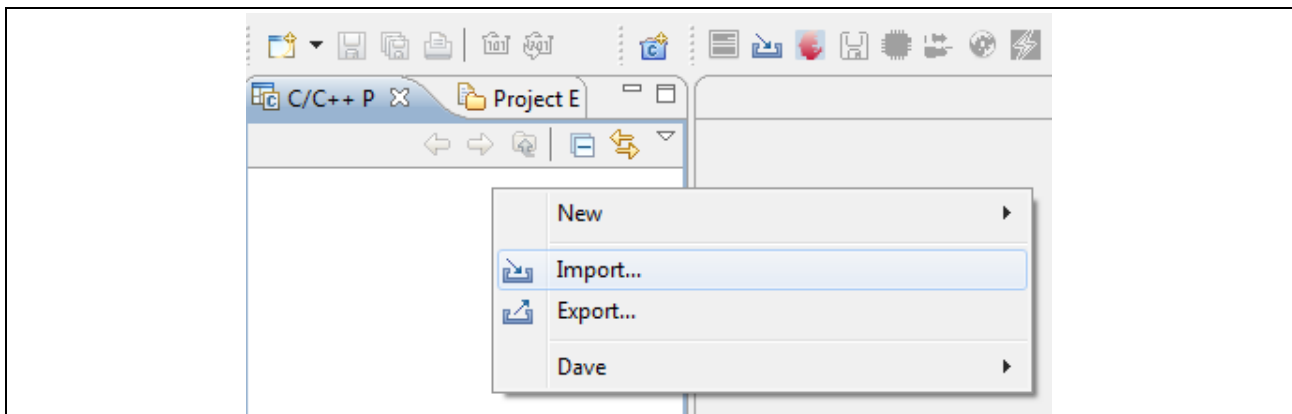
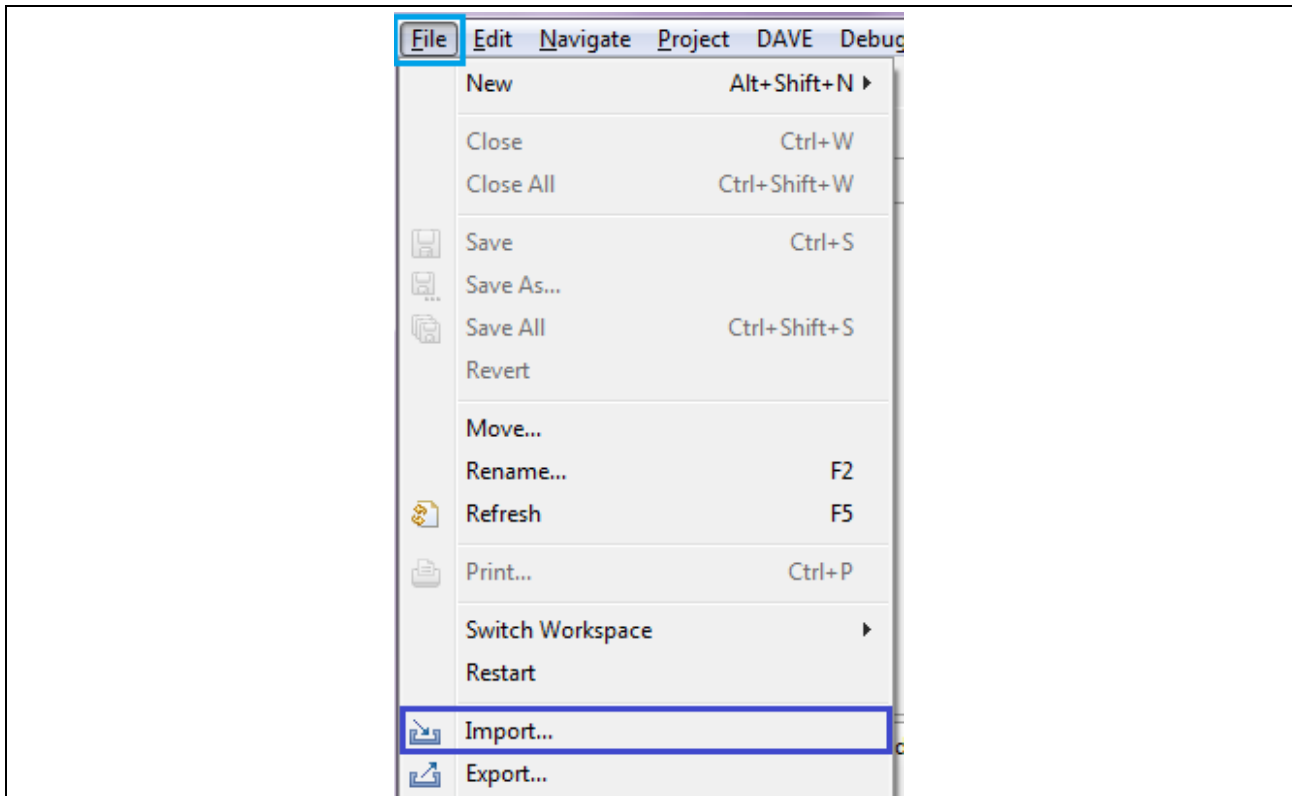


Figure 38 DAVE™ Import – Project window right click Select Import

OR

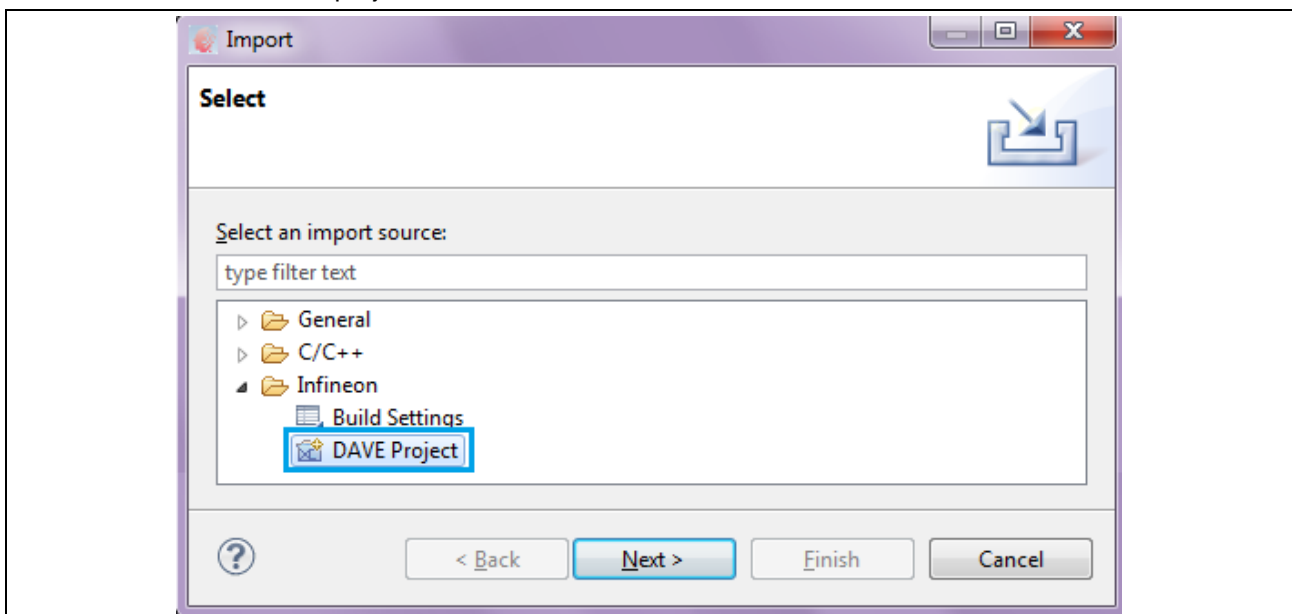
- ➔ Click on File menu and Select Import option





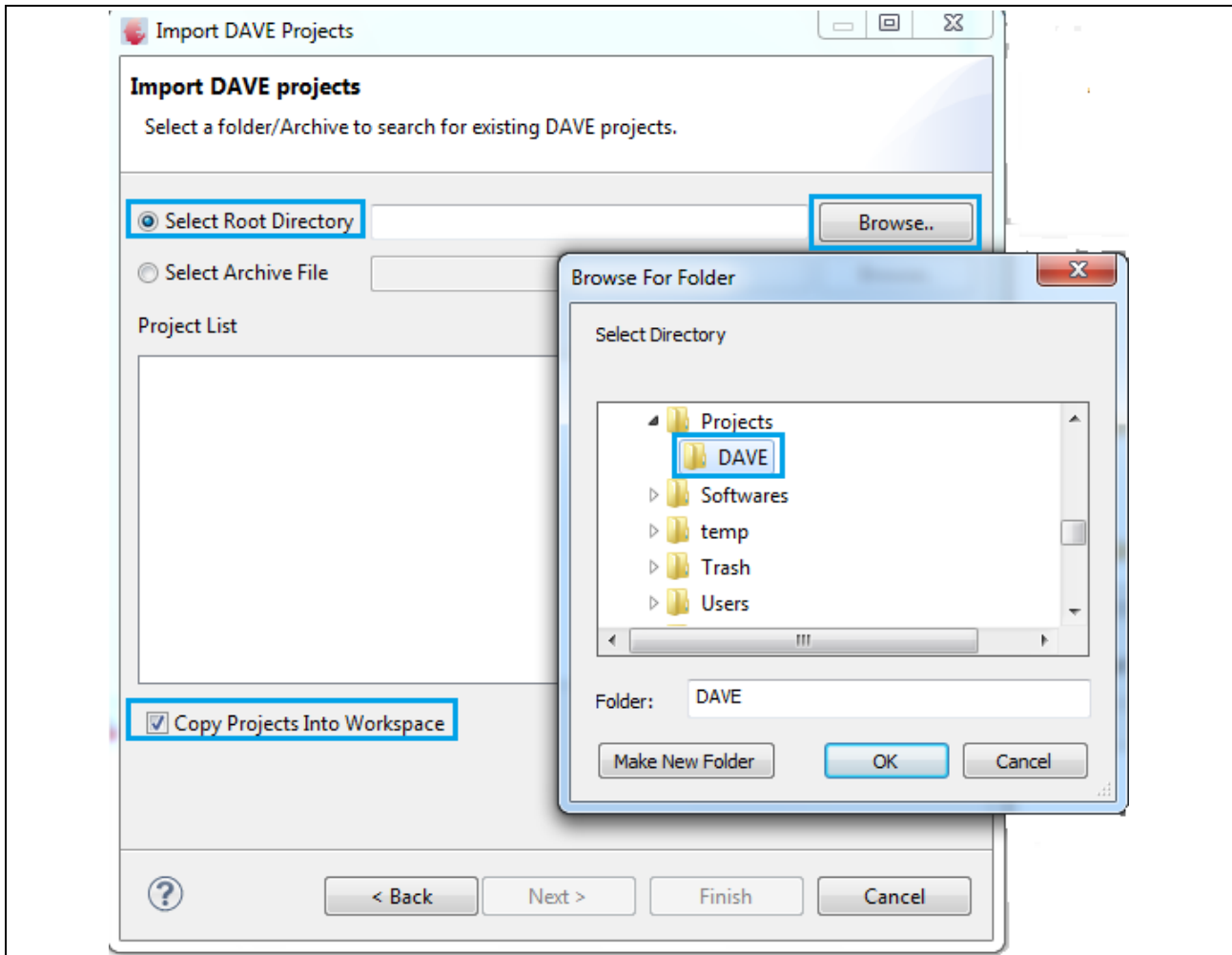
**Figure 39 DAVE™ Import – File -> Import**

→ Select the DAVE™ projects from Infineon and click on Next



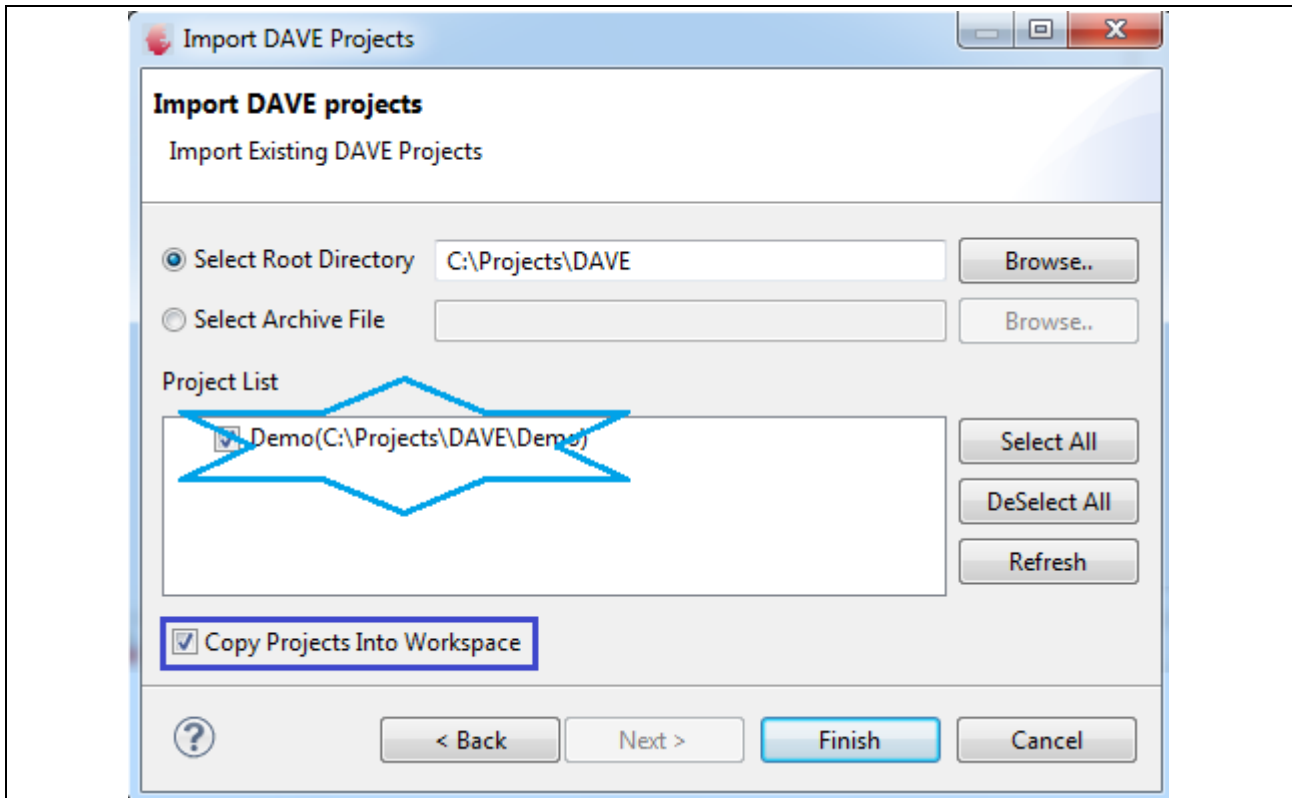
**Figure 40 DAVE™ Import – Select**

- Select Root Directory – In case the projects are in folder
- Click “Browse” and provide location



**Figure 41 DAVE™ Import – Select Root Directory**

- ➔ Select 'Copy Projects Into workspace' option to make a copy of the projects into the workspace
- ➔ Click "Finish" button



**Figure 42 DAVE™ Import – selected folder “Project”**

Note:

1. If Copy Projects into Workspace option is not selected, then the imported projects will be accessed from the path from which they are imported.
2. The same procedure has to be followed, if the projects are available in zip file format by selecting the Archive file option.

→ The imported project will be available on the Projects window

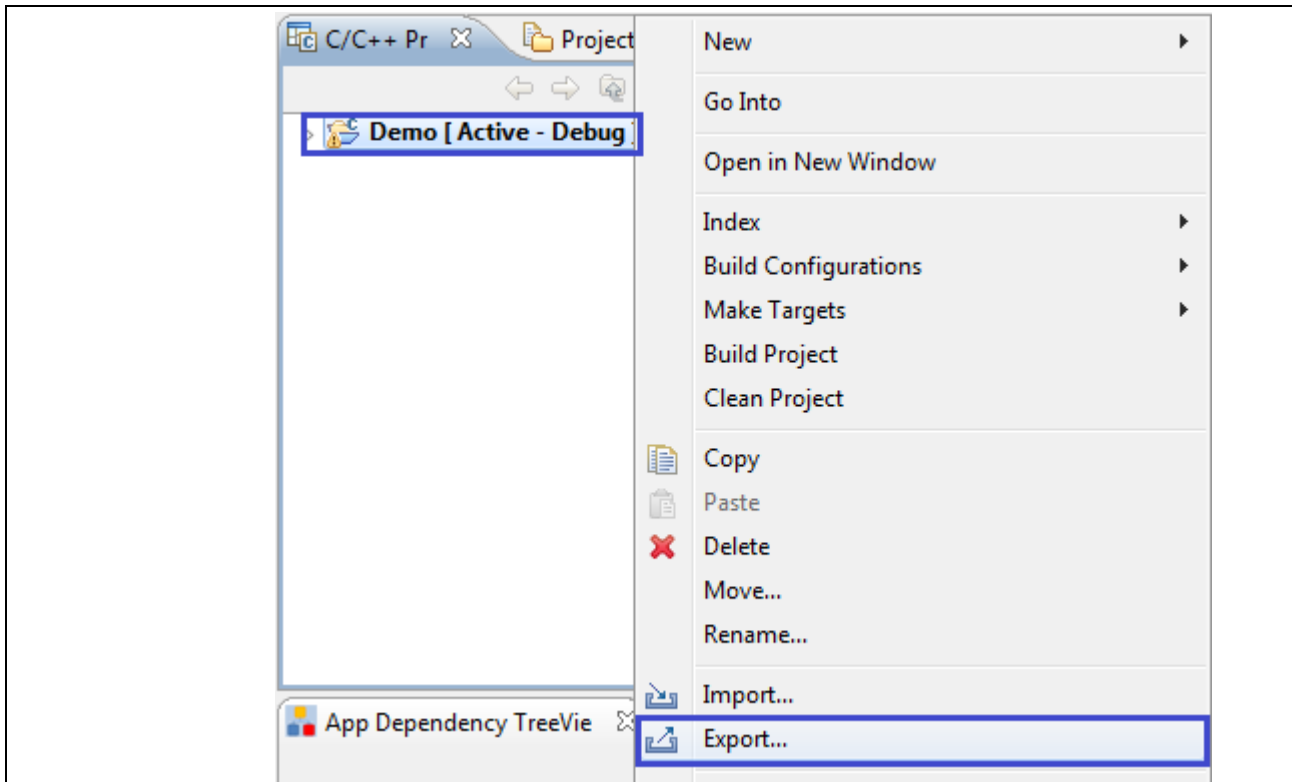


**Figure 43 DAVE™ Import – Imported project**

## 4.5.2 Export - Projects

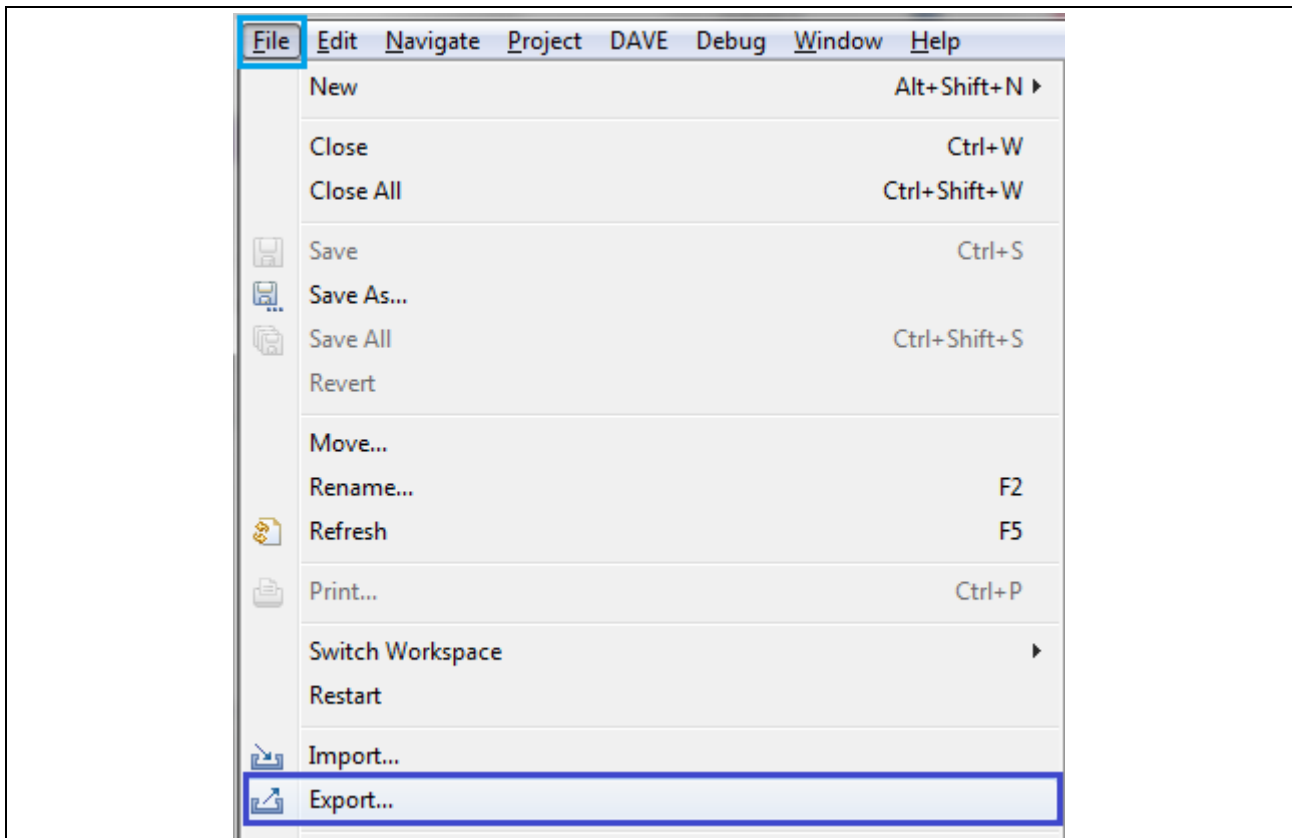
Export: helps to save the projects in to the local disc/drive. This can be done using various methods:

→ Right click on the Project Window and Select Export option      OR



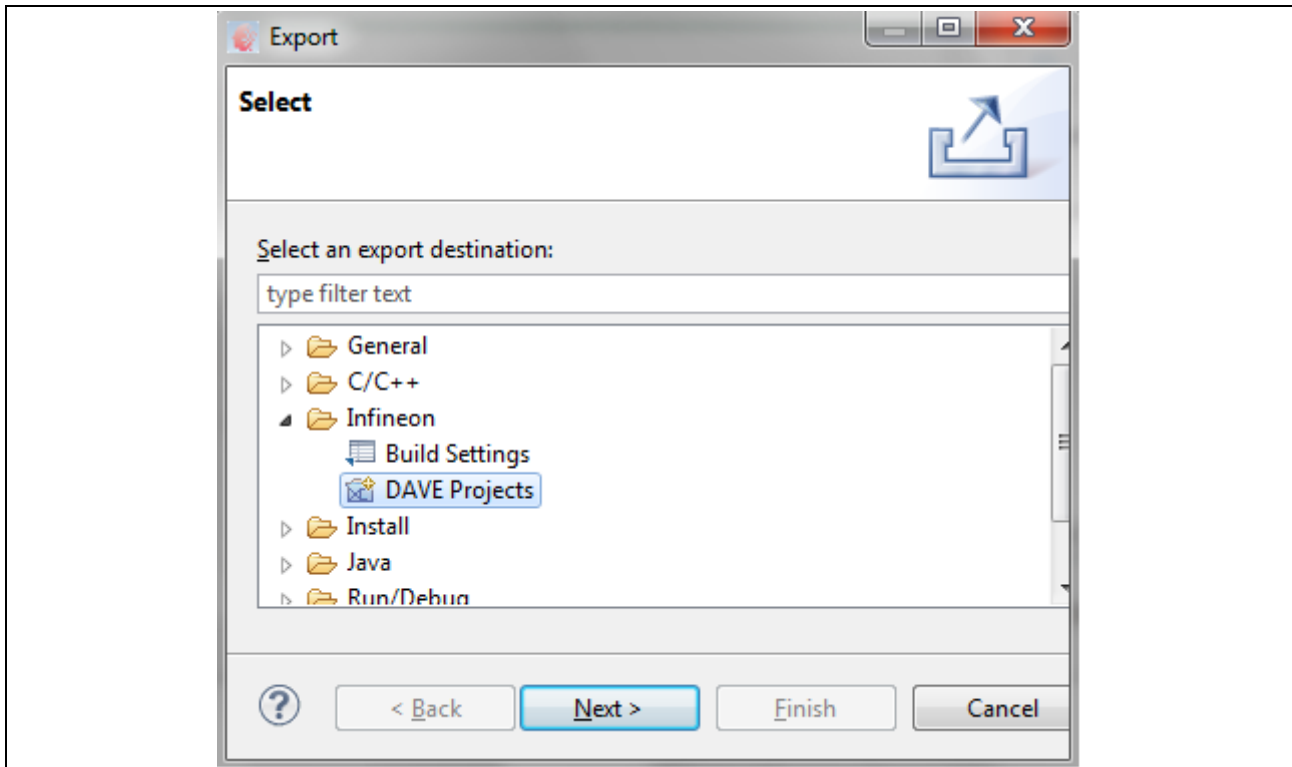
**Figure 44 DAVE™ Export - Project window right click Select Export**

→ Click on File menu and Select Export option



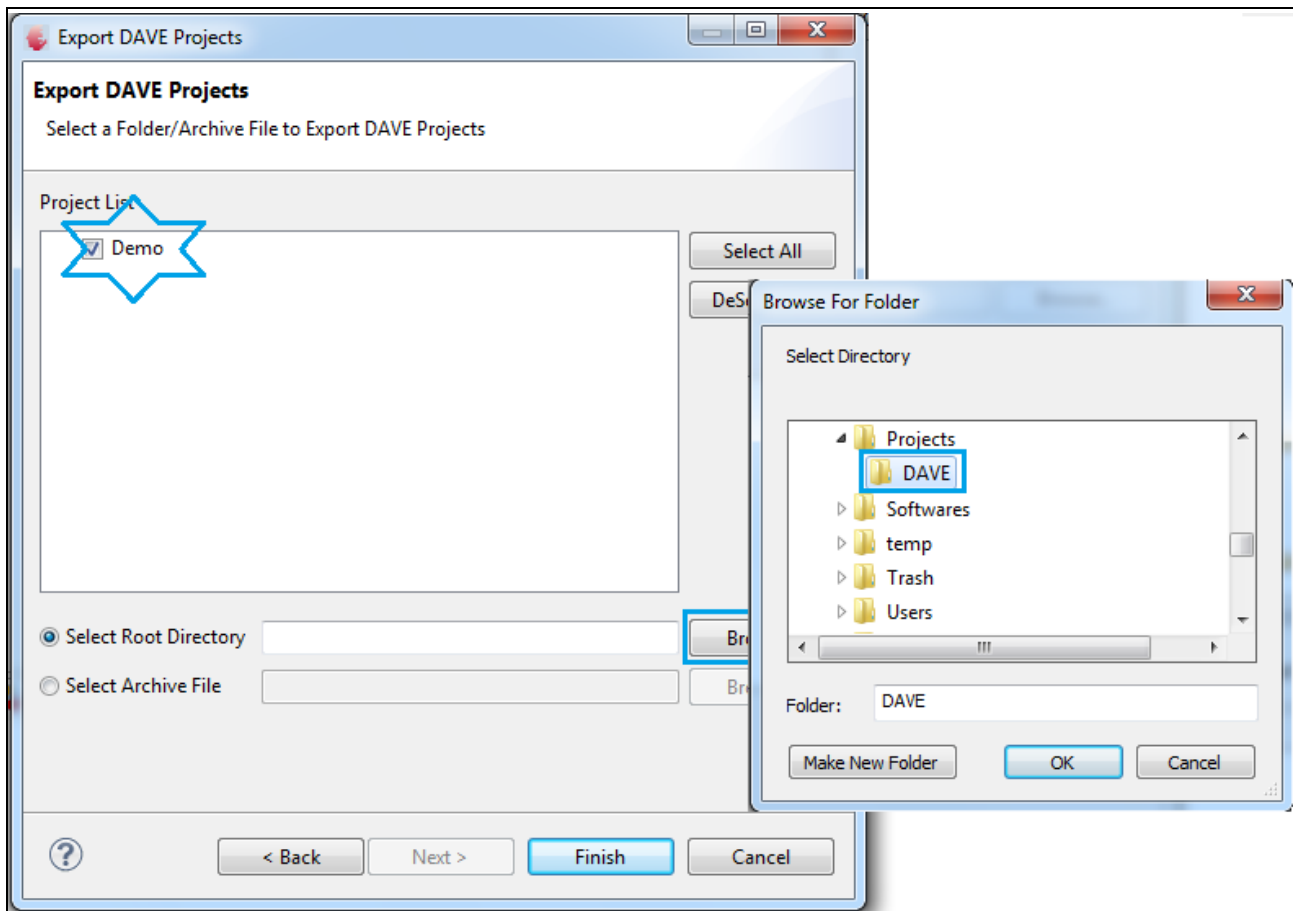
**Figure 45 DAVE™ Export – File menu Select Export**

→ Select Infineon -> DAVE™ Projects



**Figure 46 DAVE™ Export – Select**

- ➔ Selected project from Project window is automatically selected
- ➔ Select Browse – button and provide the path for export
- ➔ Then Click 'Finish' for completing Export



**Figure 47 DAVE™ Export select path**

Note: The same procedure has to be followed, if the projects are exported in zip file format by selecting the Archive file option.

## 4.6 Views

DAVE™ CE perspective supports different views for user interface.

### 4.6.1 DAVE™ AppSelection View

DAVE™ AppSelection View helps to list all the Installed Apps. The App version also listed with respect to Apps Name. To select required Apps user can get in two views Category view & Keyword view.

- Category view displays all available Apps in a top down manner – this filter Apps depends on category of Apps.
  - o For example UART001 app related to Communication category. By entering communication in filter option will list all communication related Apps, in which user can easily chose the required App.
- Key-Word view displays all available Apps for a certain peripheral – this filter Apps depends on App key word.
  - o For example UART001 app related to USIC channel. By entering USIC, it will list the app which is depends on this peripheral, user can easily choose the App.
- Both views provide a search engine for Apps, just type in the field “Search filter”

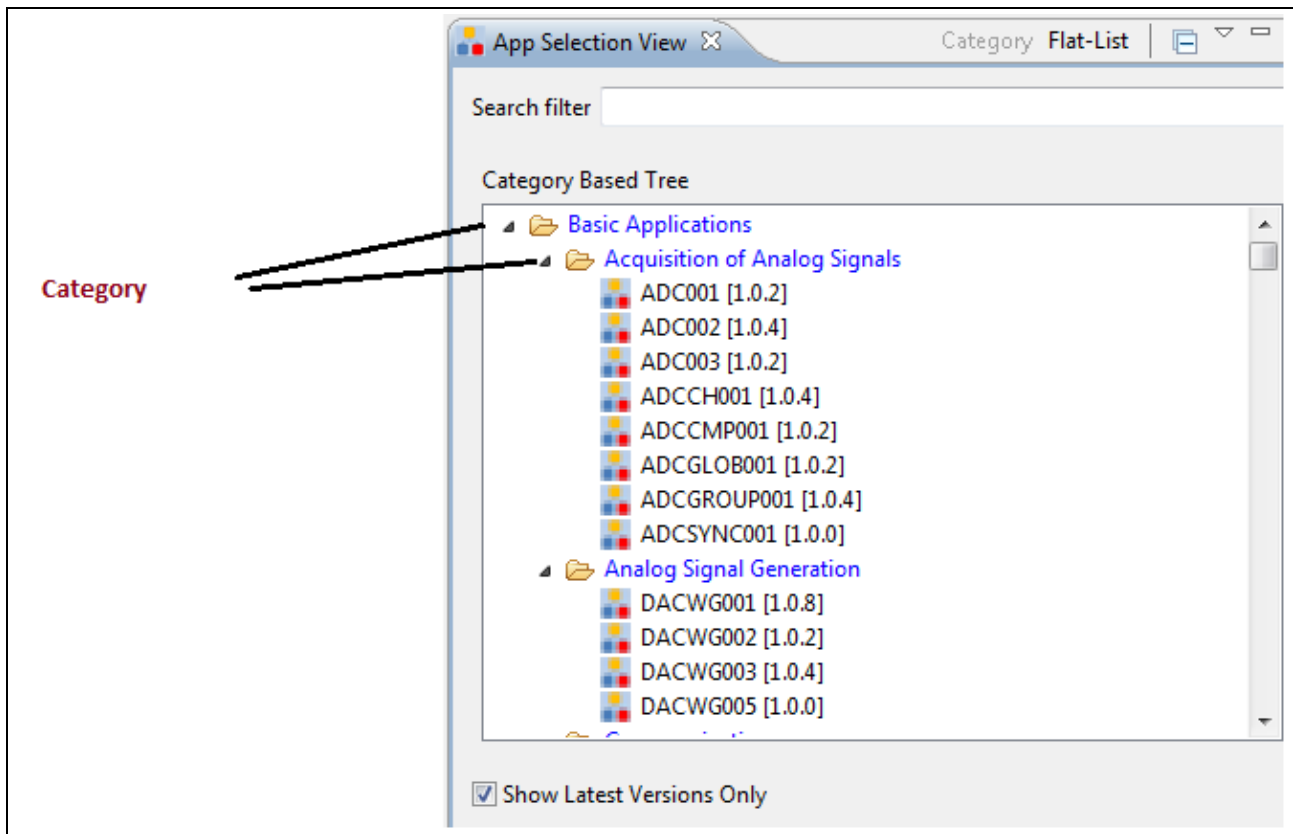


Figure 48 DAVE™ Views –App Selection View

#### 4.6.1.1 DAVE™ AppSelection View – Flat-list Search filter

To search the Apps depends on peripheral used, this feature shall be used.

For example by entering usic in search filter it will list all available Apps with respect to usic peripheral as shown below.

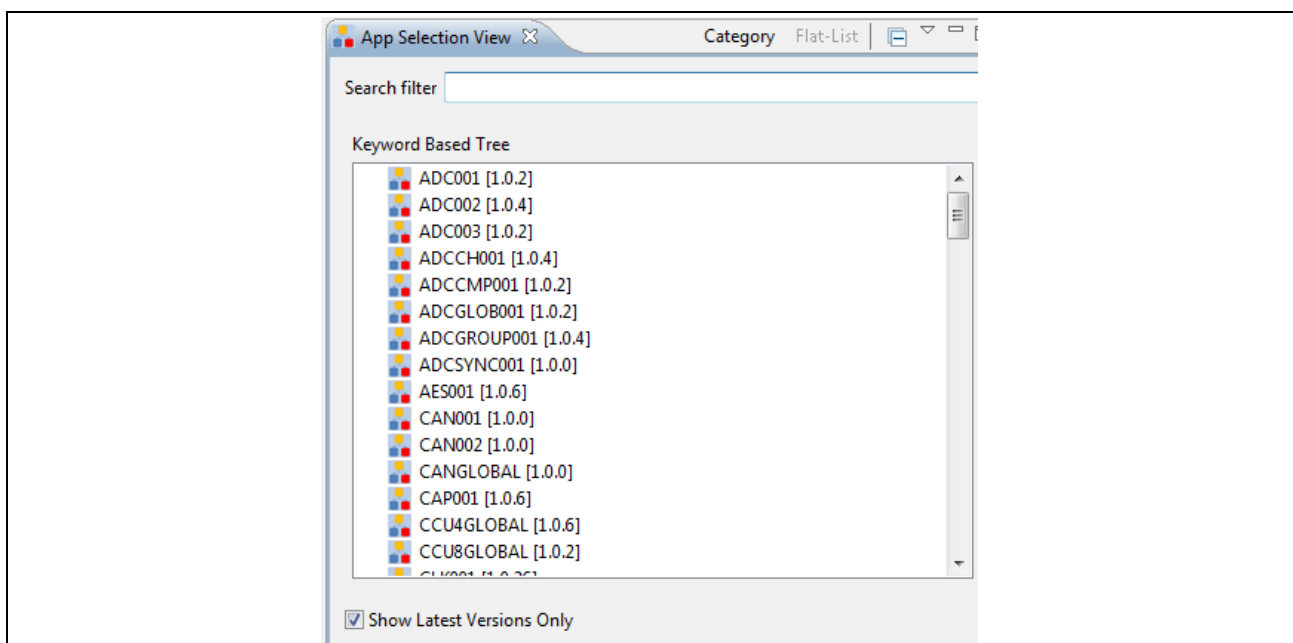
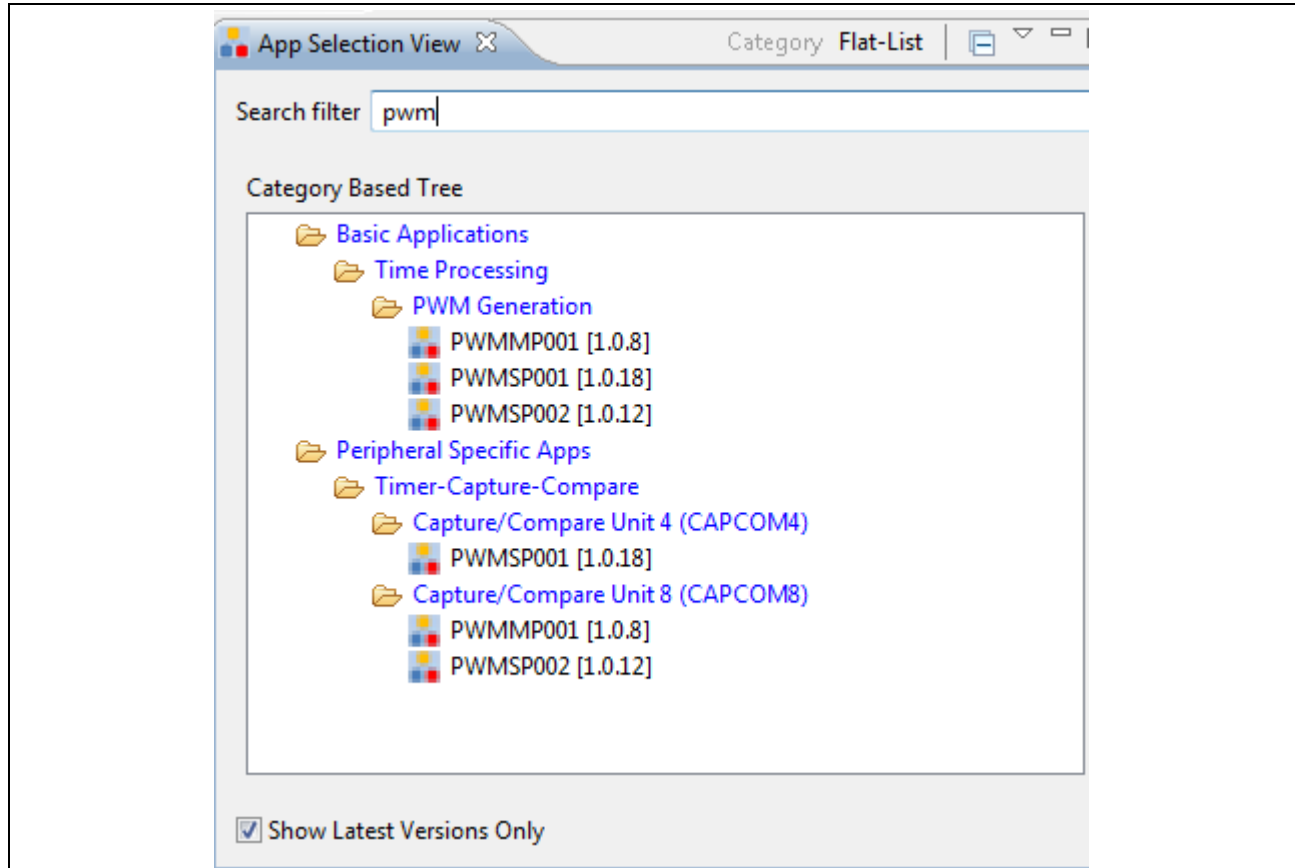


Figure 49 DAVE™ Views – App Selection Key Word Search filter

#### 4.6.1.2 DAVE™ AppSelection View – Category Search filter

To search the depending on category of App that belongs to this option shall be used. For example user can select any PWM Apps from the listed PWM related Apps.



**Figure 50 DAVE™ Views – App Selection Category Search filter**

App selection View – Show Latest Version Only

By default 'Show Latest Version Only' check box is enabled. By selecting the App selection View window will make sure that, it will display only the latest version of the installed App when multiple version of the same app is present.

By unchecking this, user gets the benefit of selecting the old version of App as well.

#### 4.6.2 DAVE™ AppDependency View

##### 4.6.2.1 S/W App Dependency Tree View

As name mentioned this view shows the dependency of Apps. This view helps to know app structure how each app is depends on others. For example UART001 app depends on CLK001 and RESET001 Apps.



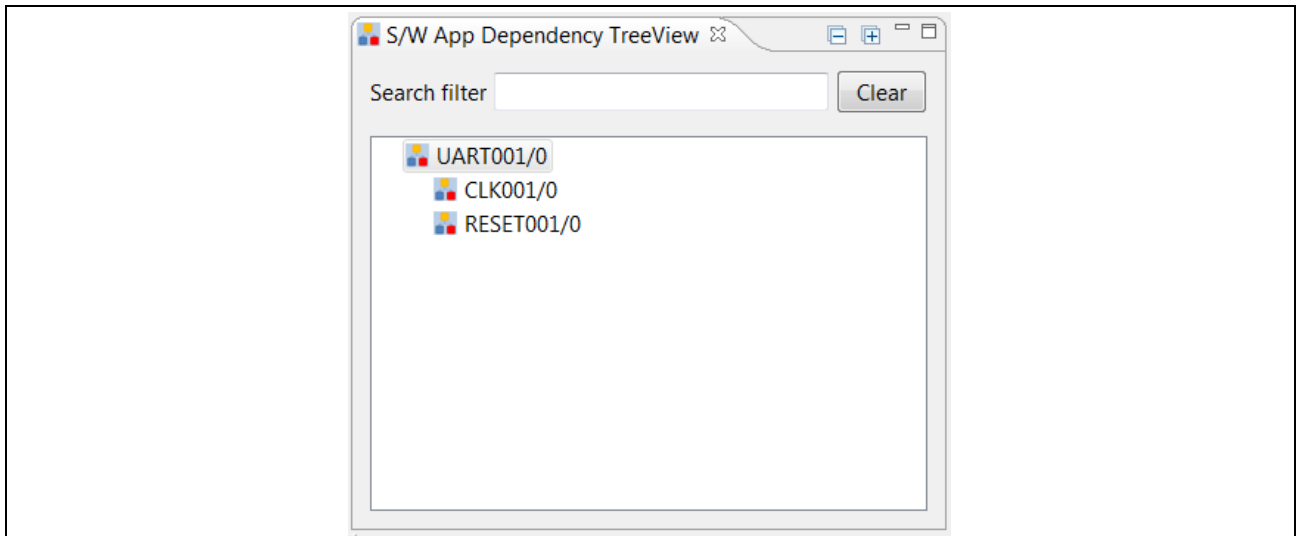


Figure 51 DAVE™ View – S/W App Dependency Tree View

#### 4.6.2.2 S/W App Connectivity View

S/W App Connectivity View will show the App Software dependency in graphical manner, how App depends on other Apps. Below figure shows the dependency of UART001 App with (Consumes) CLK001 and RESET001 Apps.

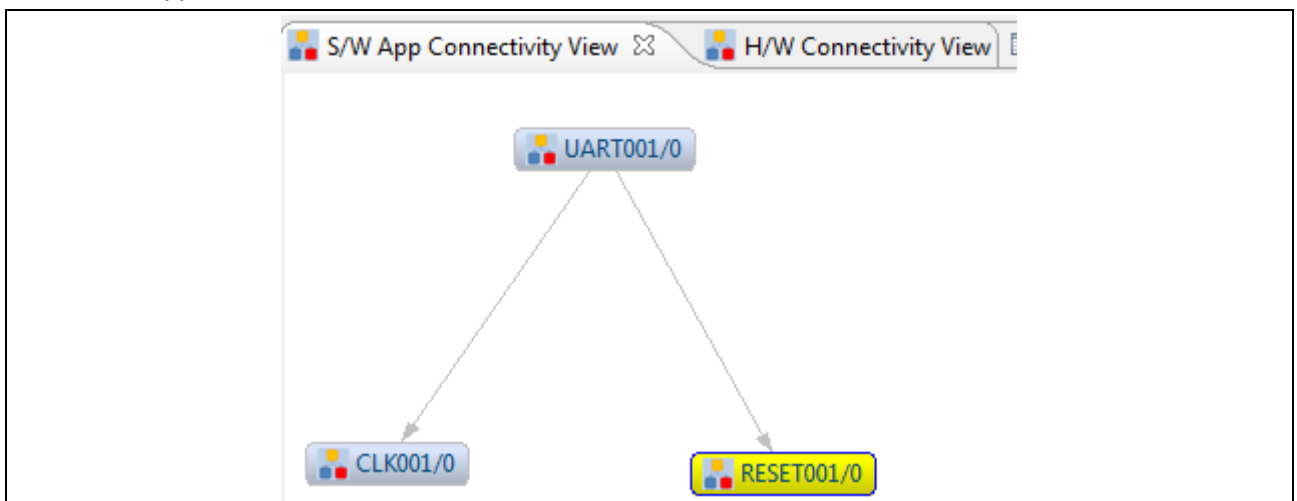


Figure 52 DAVE™ View – S/W App Connectivity View

### 4.6.2.3 H/W Connectivity View

H/W App Connectivity View will show the App/s H/W Signal Connectivity in graphical manner, Show App Signal connectivity between Apps.

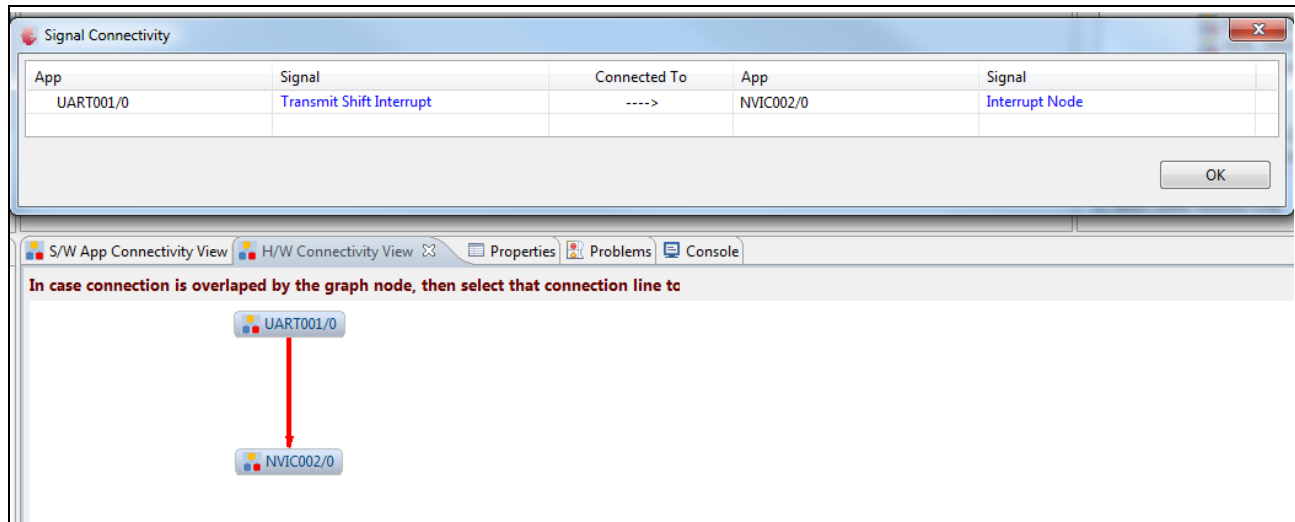


Figure 53 DAVE™ View – H/W Connectivity View

## 4.7 UI Editor

UI editor is used for the configuration of selected app. Further Information about UI Editor is available in the section: 7.1

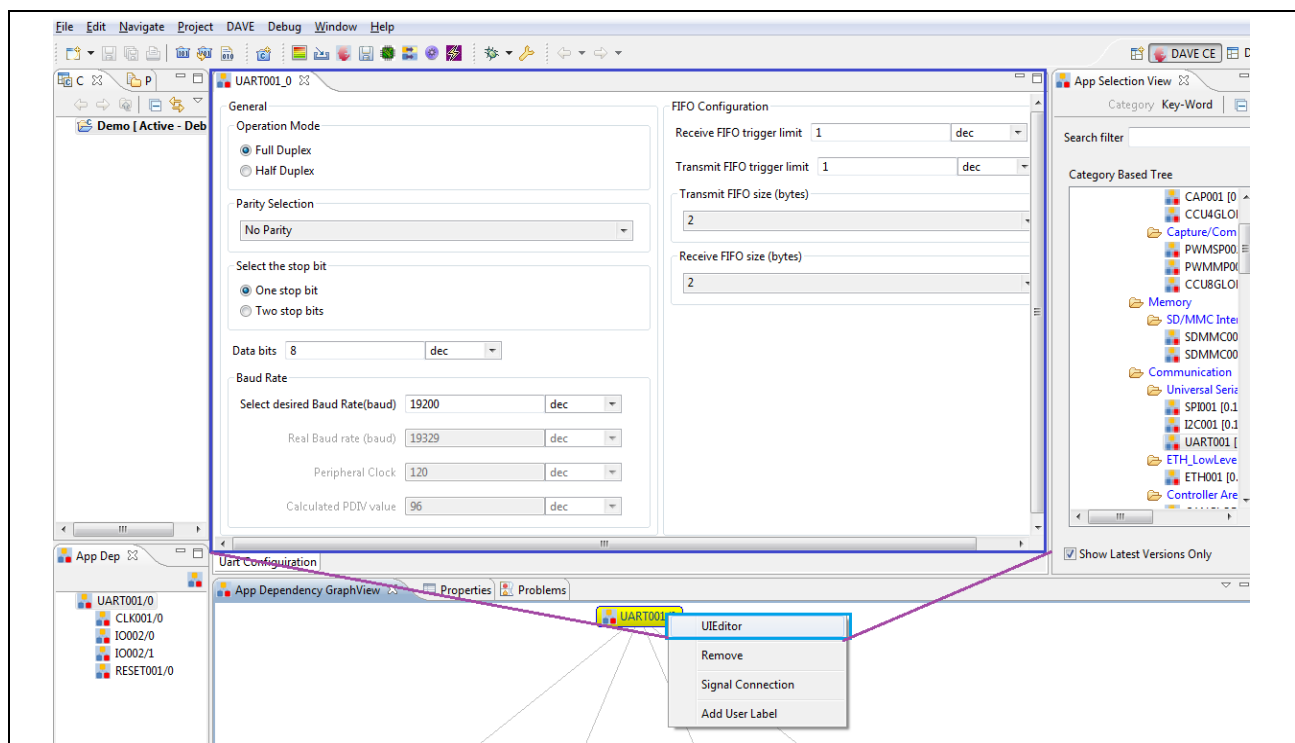
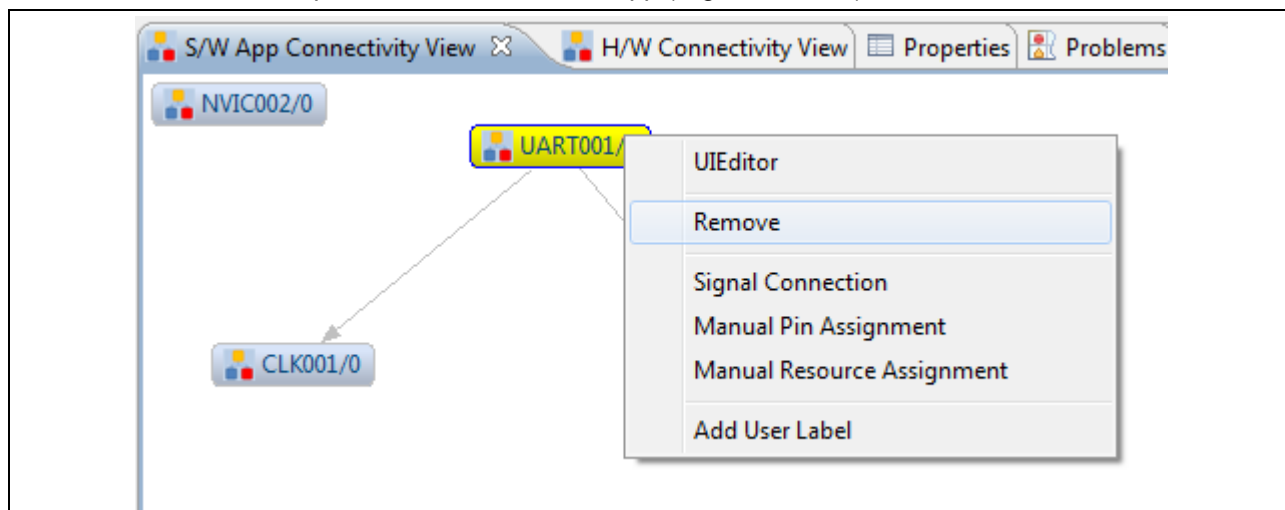


Figure 54 DAVE™ UI Editor – App Selection

## 4.8 Remove

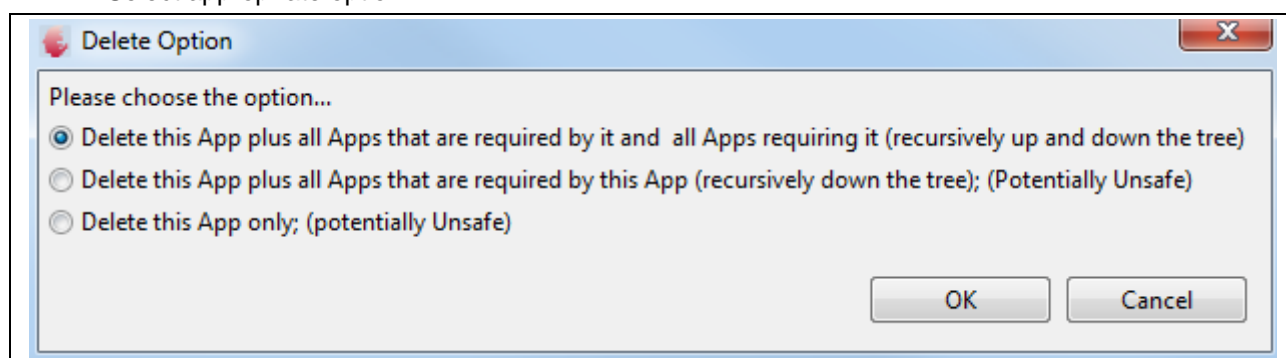
Individual Apps can be removed from App Dependency Graph view by using Remove option

- Select App and Right Click
- Select Remove Option to remove the main app (E.g.: UART001)



**Figure 55 DAVE™ S/W App Connectivity – Main App – Remove**

- Select appropriate option



**Figure 56 DAVE™ S/W App Connectivity – Dependent App – Remove**

## 4.9 Signal Connection

Hardware device has many signals connecting from one peripheral to other.

### 4.9.1 Signal and Connection

Signal mean the physical connection (e.g. signal name in the netlist (CCU8xOut)) and not the logical signal as trigger; interrupt that characterized by level or pulse width.

In the context of DAVE™ Signal connection has different meaning than the common one. It is not an electrical connection but more a logical relationship because typically mux are implemented with or\_port and not as simple switch (bi-directional)

In the GUI for each signal owned by the App it is possible to show the possible connection to the signal of the apps inside the workspace (not real hardware signals) (e.g. appPWM1\_CCU8\_Out).

At this stage, exists only "virtual signal" apps and DAVE™ doesn't know which is the final correspondence between "virtual signal" and hardware. This is valid also for registers. (E.g. The app knows that the registers of CCU8 slice are required but not which HW CCU8 slice instance.)

At every selection of DAVE™ will check that it possible to find a configuration or hardware placement of signals and registers that satisfies the connection requirements.

Types of Signal relates resources

- Virtual Signals
  - o define logical relationship between Apps
  - o have a direction (provided → required)
  - o can have additional attributes, e.g. frequency for periodic signals
- Physical Signals
  - o represent the “lines” on chip hardware
  - o for peripherals/slices: provided resources only
  - o for Apps: required resources only

Virtual Signals are statically assigned to Physical Signals inside the app Manifest as value

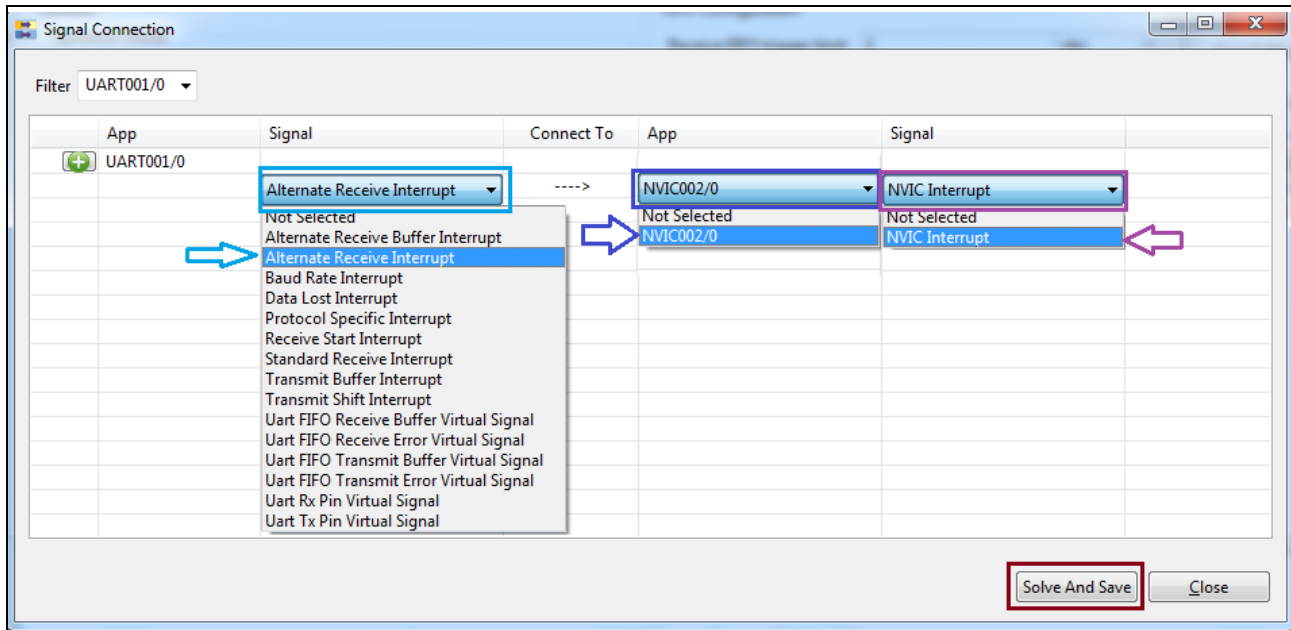
- Mapping Phase
  - o During this phase DAVE™ map virtual HW resources to defined HW resources.

## 4.9.2 Signal Connection - Configuration

Signal Connection helps to configure signals manually for virtual signal to other. The Source Signal and Destination signals should be matched with connectivity then only signals will be connected. To check signal connections are valid or not need to click ‘Save and Solve’, after signal connection is completed. Solver will connect only possible signal, else it will give error message in ‘ErrorLog.txt’.

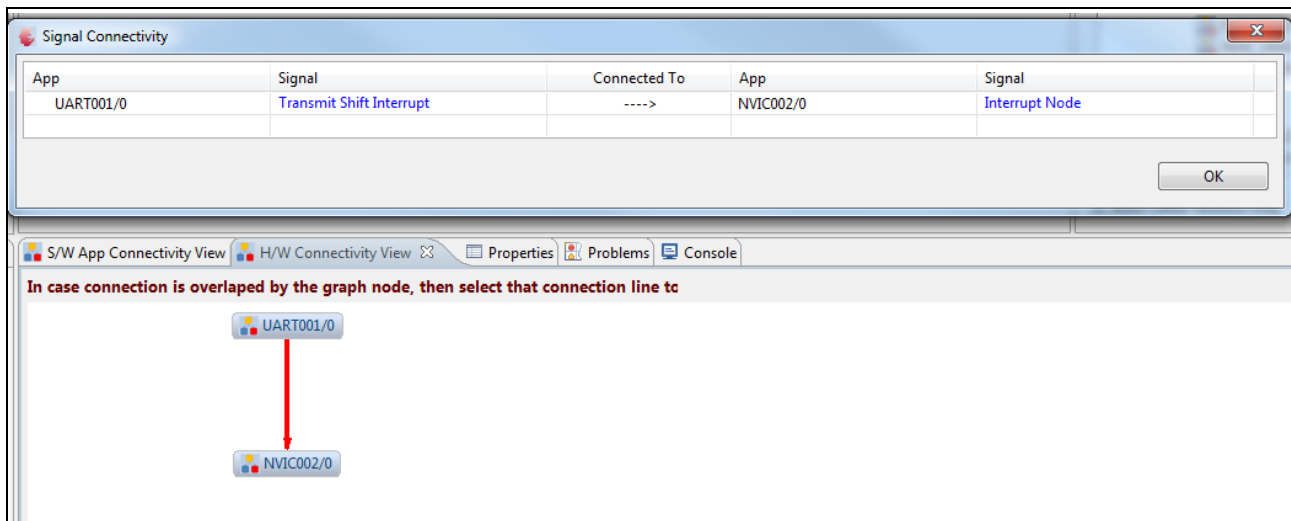
Now shall check one example by connecting UART001 Interrupt to NVIC002 App.

- Select UART and NVIC002 App from App Selection View and perform the IO Pin selections
- Select the App [UART001]– right click select Signal Connection
- Select the Signal/Interrupt from Signal List e.g. Alternate Receive Interrupt
- Select newly added NVIC002 app from App List
- Select NVIC Interrupt from Signal
- Then click ‘Save And Solve’



**Figure 57 DAVE™ Signal Connection**

During this process DAVE™ executes solver internally to check for the available resources. The final UART App after signal Connection will be as shown below in H/W Connectivity view,



**Figure 58 DAVE™ Signal Connection – UART001 final App**

Note: Detailed explanation of Signal Connection is available in section 9.

## 4.10 Signal Assignment

By verifying the Report window Signal Assignment page use will come to know about which signals are connected manually to the App.

The fonts in Blue color indicate the manual signal connection and where as green color indicates static signal connection of the App (Through App Manifest).

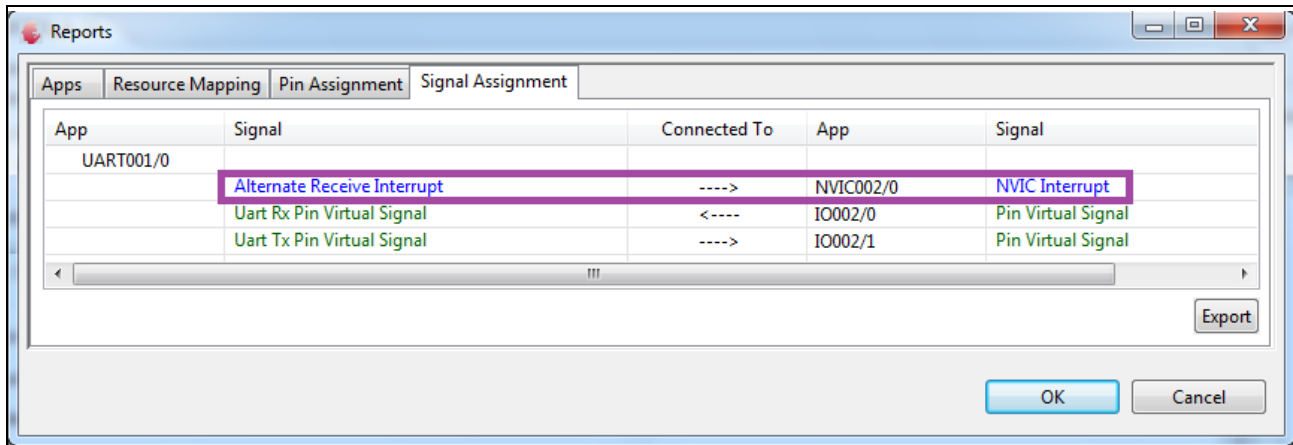


Figure 59 DAVE™ App Dependency – Signal Assignment

#### 4.11 Add User Label

User can add Label to the selected App for better understanding the use case.

- Right click on the selected App
- Select Add User Label option
  - o Provide desire Label and press OK.



Figure 60 DAVE™ App Dependency – Add User Label

Note: The User Label should be as per 'C Language' variable/macro declaration and it should be unique

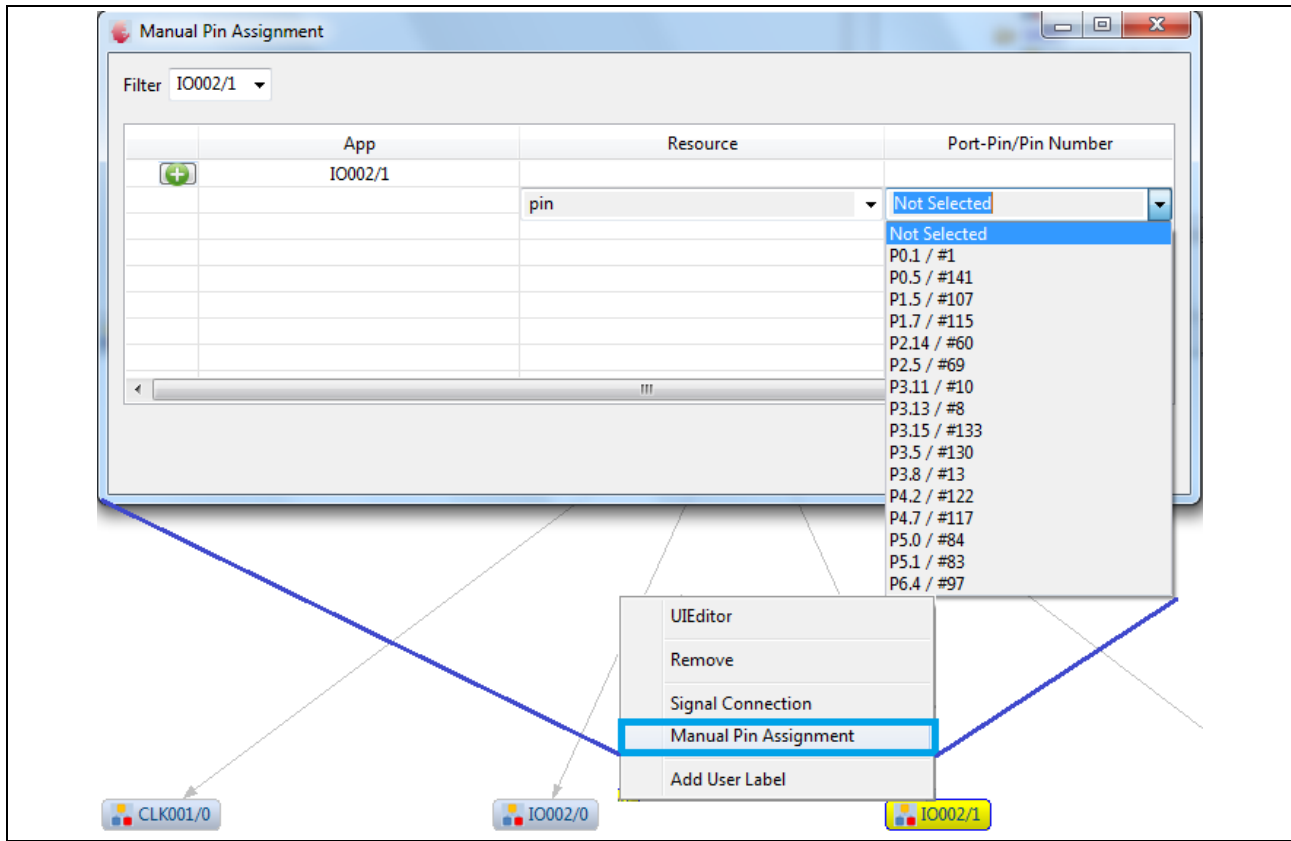
#### 4.12 Manual Pin Assignment

Manual Pin Assignment will be used to configure IO Pins.

Note: This feature is available only for IO related Apps

- Select IO App [IO002] – Right click & Select Manual Pin Assignment
  - o Opens the Manual Pin Assignment Window
- Select Resource Name and Port-Pin/Pi-Number and save the configurations
- Execute solver to solve the pin configurations made

Further information about resource is available in later sections



**Figure 61 DAVE™ Manual Pin Assignment – Configurations**

Note: Detailed explanation of Manual Pin Assignment is available in section 10.

## 4.13 Report

### Resource/Groups

Grouping of resources, logically belonging together, in particular

- ☐ the parameters of an App (App as a group),
- ☐ the peripherals inside a device (device as a group)

Note: The same peripheral may be contained multiple times in a device. There is no hierarchical grouping of the multiple instances of a single peripheral inside a device

- ☐ the registers in a slice in a peripheral (slice is a group inside the group peripheral)
- ☐ the bit fields inside a register
- ☐ the pins of a package

Note: Resource groups contain other resources; concurrently, they are themselves resources.

- Resource groups are defined by
  - Apps → App resource type
  - Peripherals → Peripheral resource type
  - Packages → Package resource type

- Devices → Device resource type
- Registers → SFR resource type
- Slice<sup>+</sup> (i.e. channels) of a peripheral or (generic) groupings → ResourceGroup resource type

Further Information about resource is available in Section: 11.2 Resource Binding Information.

NOTE: <sup>+</sup>the terms “slice” and “channel” are used interchangeably in descriptions; formally, the term “slice” shall be used.

Resource Mapping Report will provide information about resources binding.

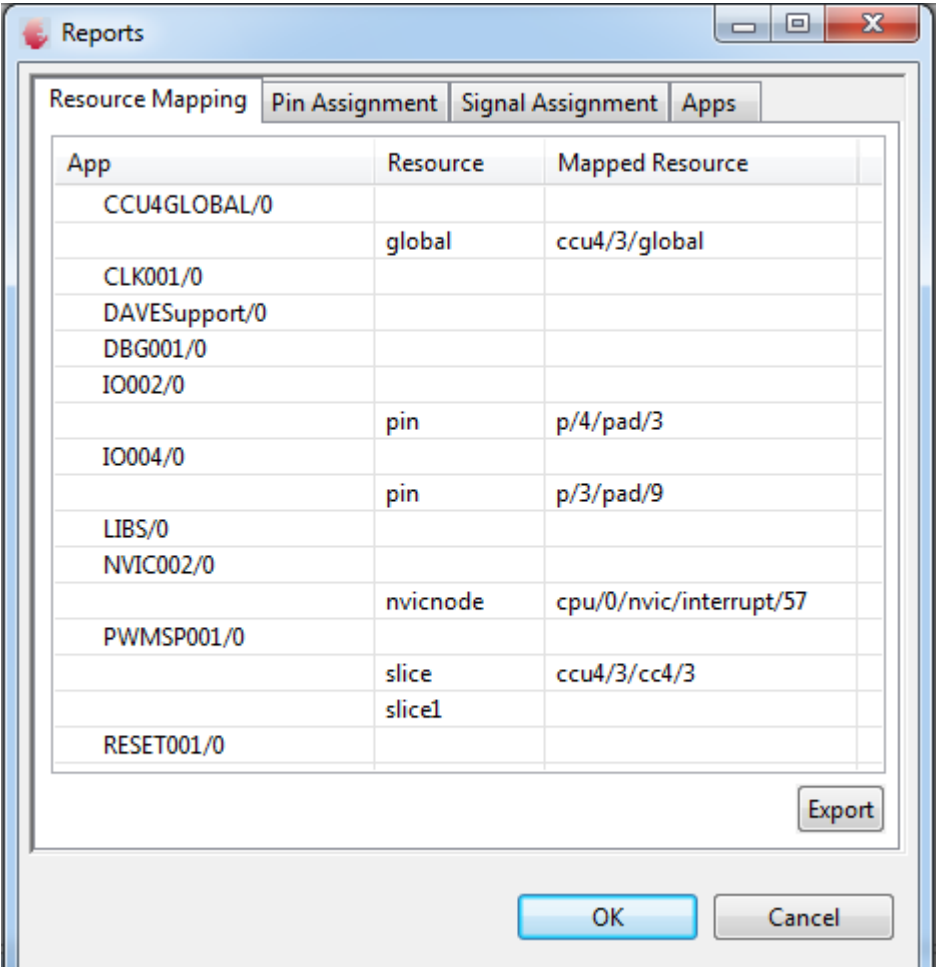
#### **4.13.1 Apps**

The Apps page will display the information about the apps used along with the version being used for the use case.





Resource mapping page will display the information about how the app resources are mapped to the available hardware resources/peripherals.

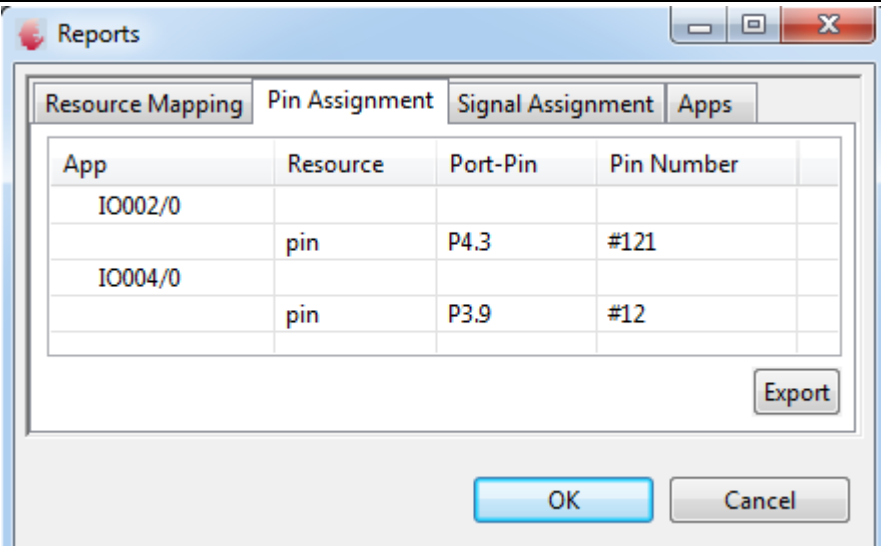


App	Resource	Mapped Resource
CCU4GLOBAL/0		
	global	ccu4/3/global
CLK001/0		
DAVESupport/0		
DBG001/0		
IO002/0		
	pin	p/4/pad/3
IO004/0		
	pin	p/3/pad/9
LIBS/0		
NVIC002/0		
	nvicnode	cpu/0/nvic/interrupt/57
PWMSP001/0		
	slice	ccu4/3/cc4/3
	slice1	
RESET001/0		

Figure 63 DAVE™ Report – Resource Mapping

### 4.13.3 Pin Assignment

Pin Assignment page will provide details of the pin configurations for the App.



App	Resource	Port-Pin	Pin Number
IO002/0			
	pin	P4.3	#121
IO004/0			
	pin	P3.9	#12

Figure 64 DAVE™ Report – Pin Assignment

#### 4.13.4 Signal Assignment Report

Signal Assignment page will provide the Signal Configuration made for the App.

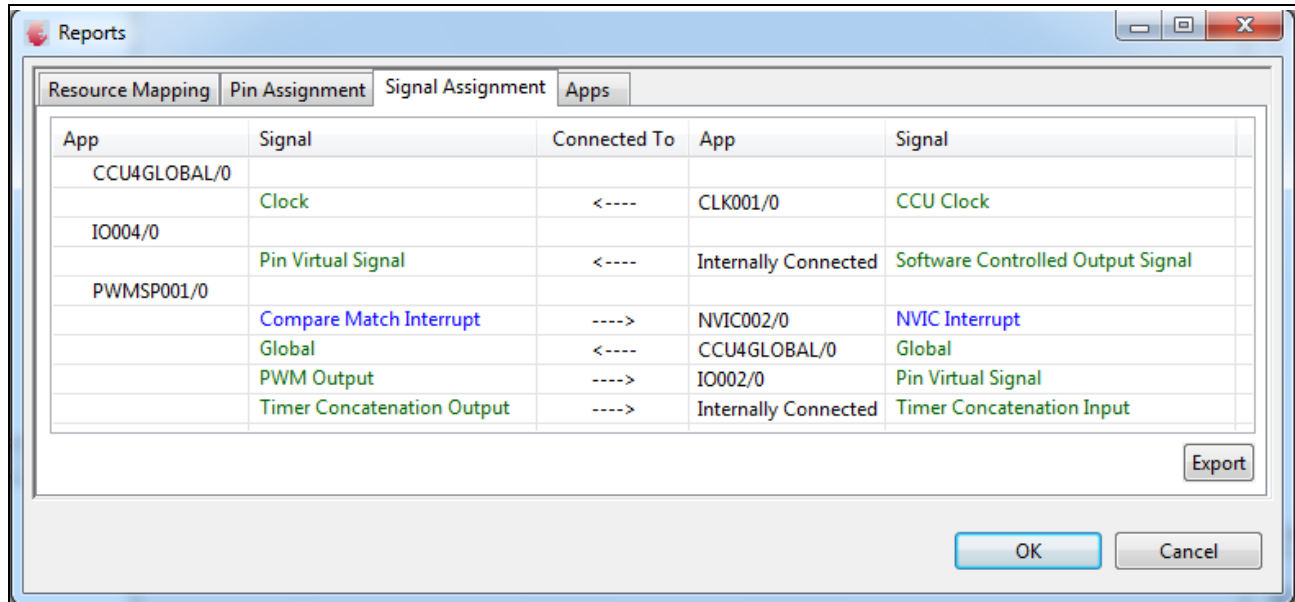


Figure 65 DAVE™ Report - Signal Assignment

NOTE:

1. Export will export the settings to \*.csv file and can be opened by using Microsoft Excel Application.
2. More detailed explanation of Report is available in section 11.

#### 4.14 Project View

Project Selection view is responsible for New Project/Active Project, Generated files, Project Deletion, Project Import/Export, etc.

##### 4.14.1 Generated Files

The source and include files can be generated by using the Generate code menu button for the Apps available in App Dependency Graph view.

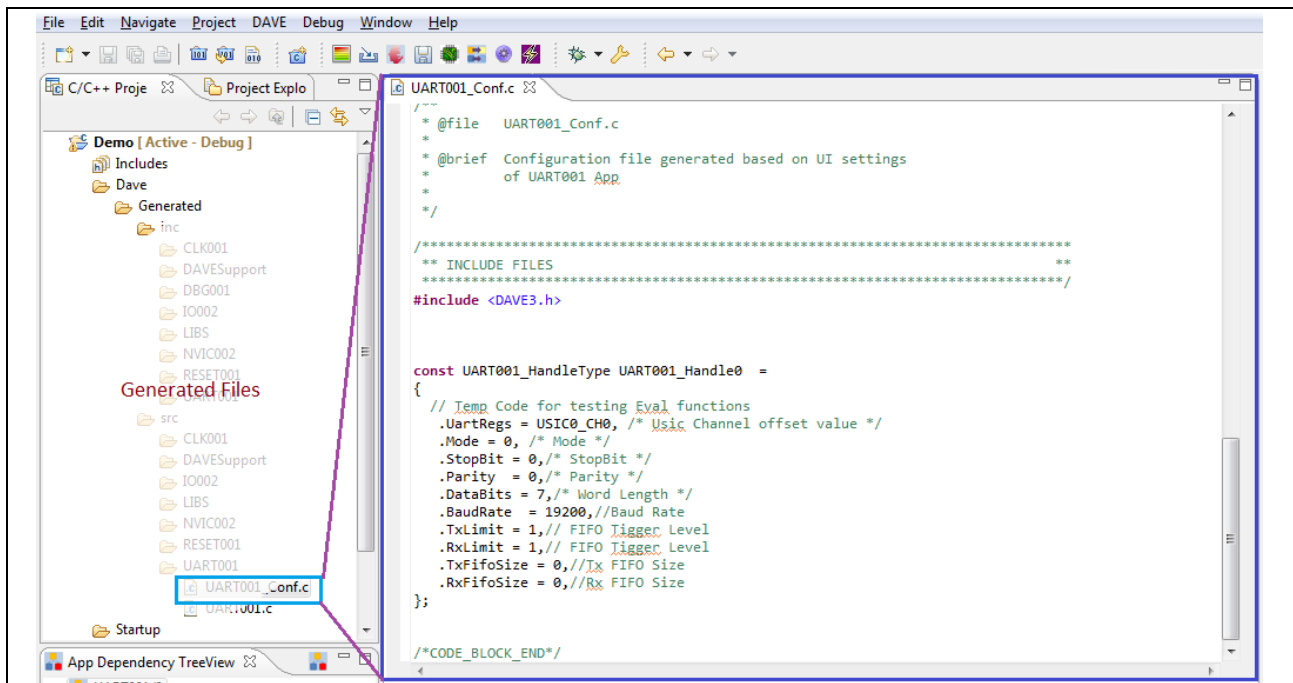


Figure 66 DAVE™ Project Selection View – Generated Files for the App

#### 4.14.2 Edit MAIN.C

The user can add code in main.c file by editing the same.

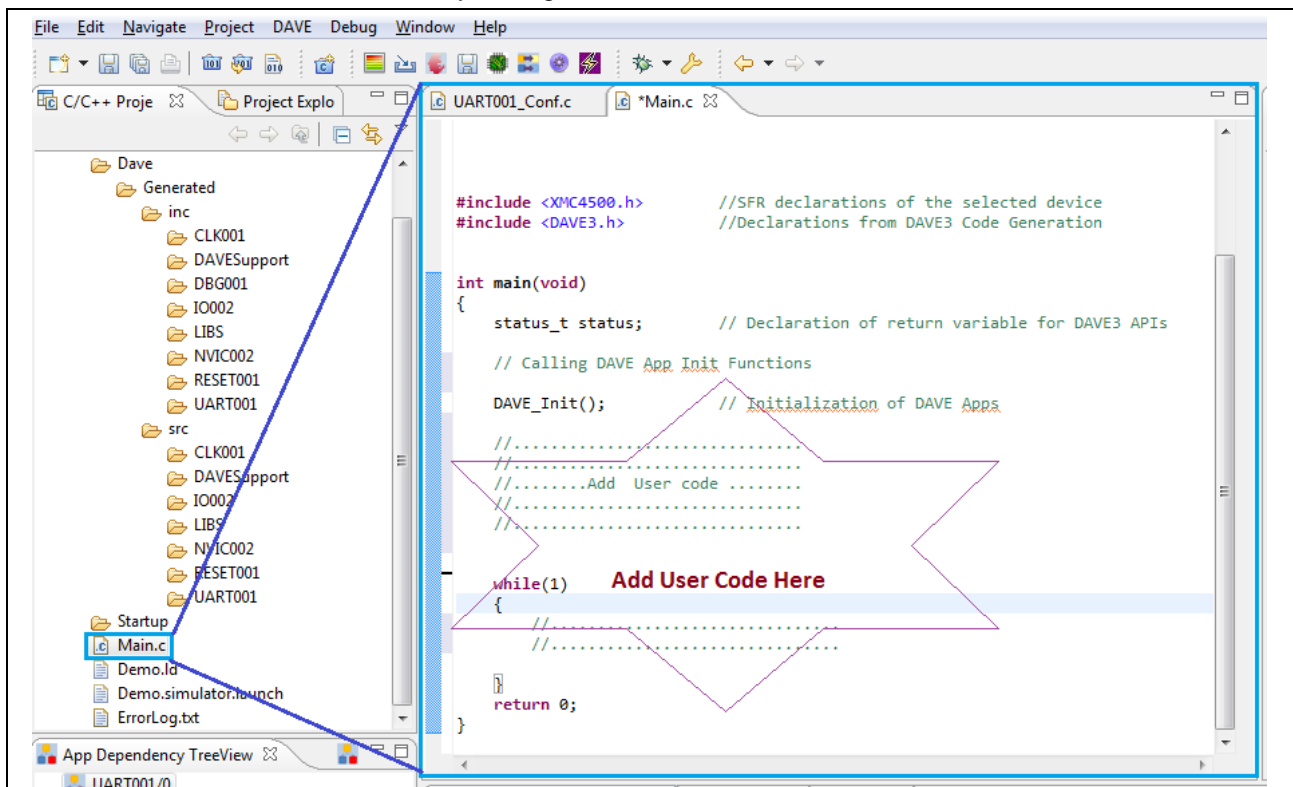


Figure 67 DAVE™ Project Selection View - Adding User codes etc...

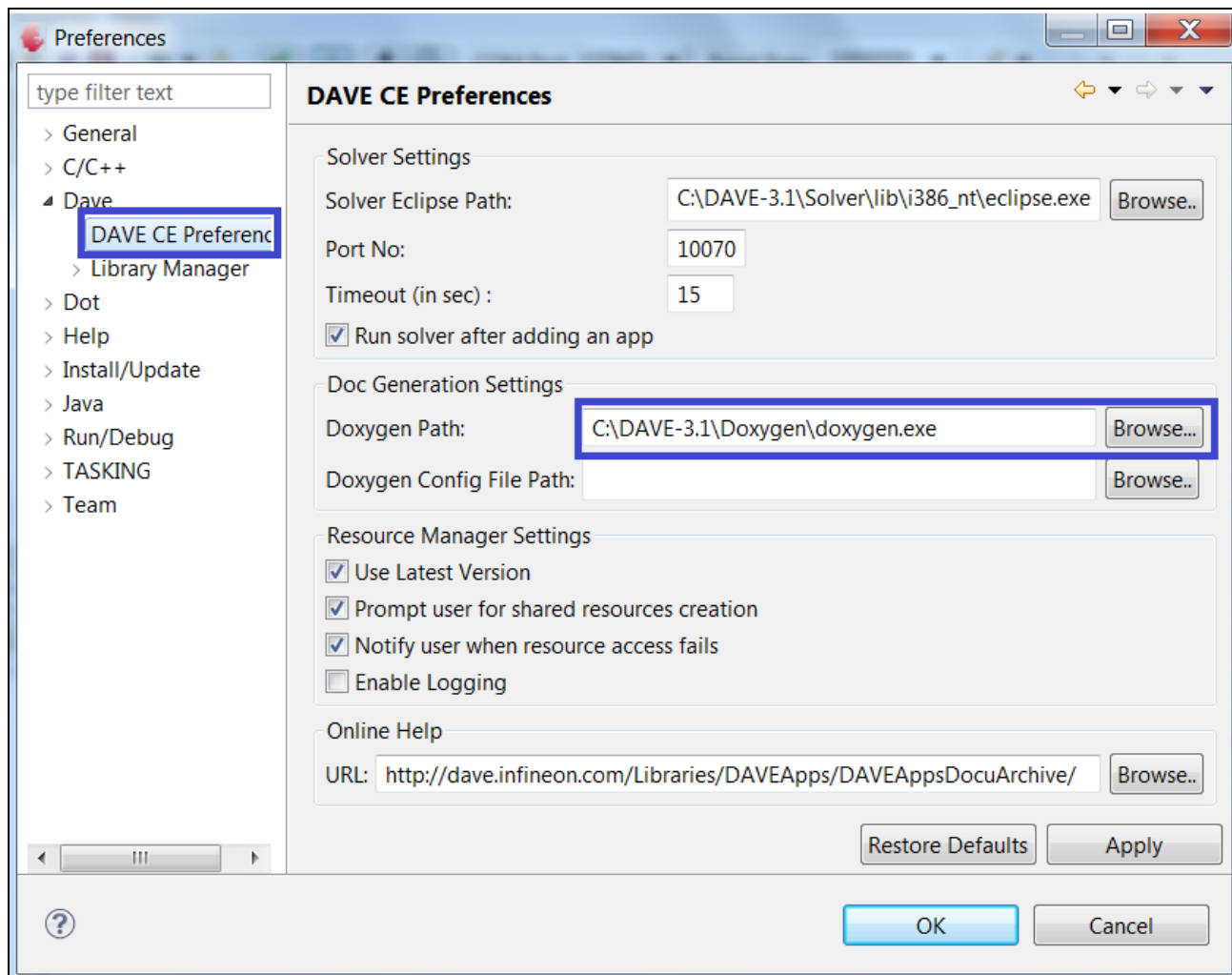
Note: Any changes/modification done in Dave Generated files other than main.c file by user it will be overwritten by the tool on re-code generation.

### 4.14.3 Document generation

The document is generated for the DAVE™ generated files/code by using the Generate Doc option from Dave menu.

To generate the document,

- Select Preferences option from Windows menu,
- Preferences window will open
- Provide the Doxygen\* Installed exe path Dave->DAVE™ CE Preferences – Doxygen Path.



**Figure 68 DAVE™ Document Generation – Preferences**

Note: Code generation should be done before generating the doc generation.

Doxygen\* - Doxygen needs to be installed separately.

- Select Generate doc from Dave Menu for generation of the documents.
- It will take some time to generate the documents

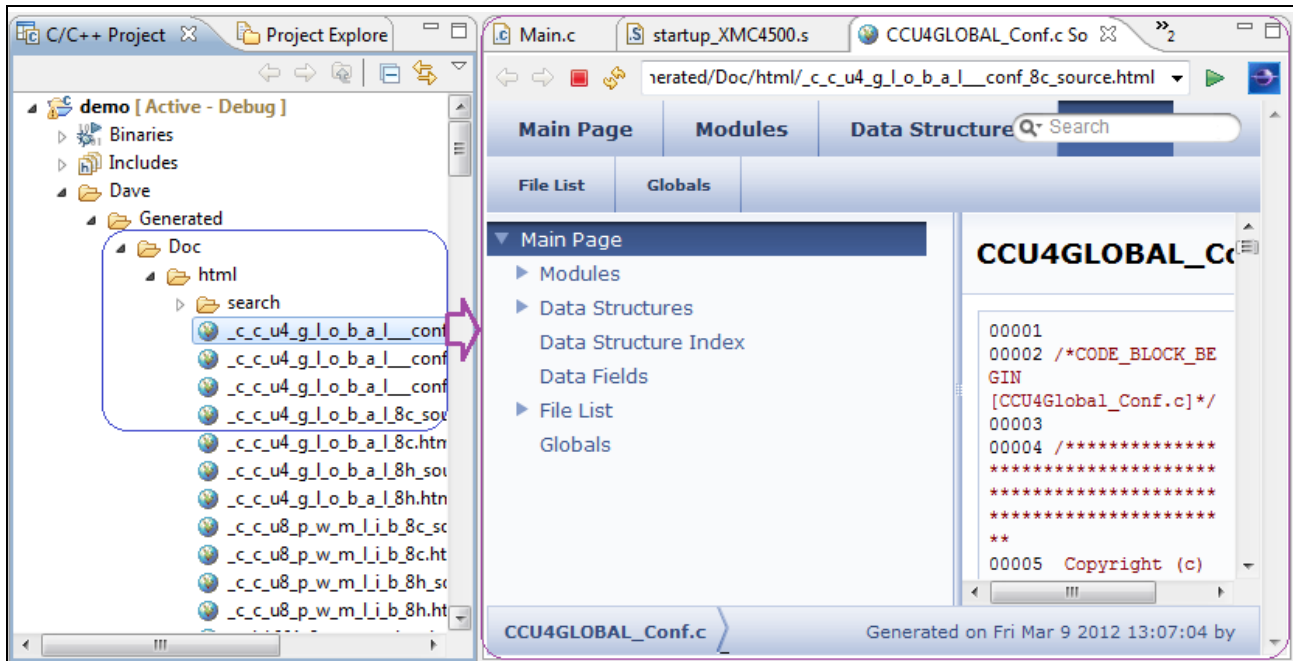


Figure 69 DAVE™ Document Generation – Generated Document

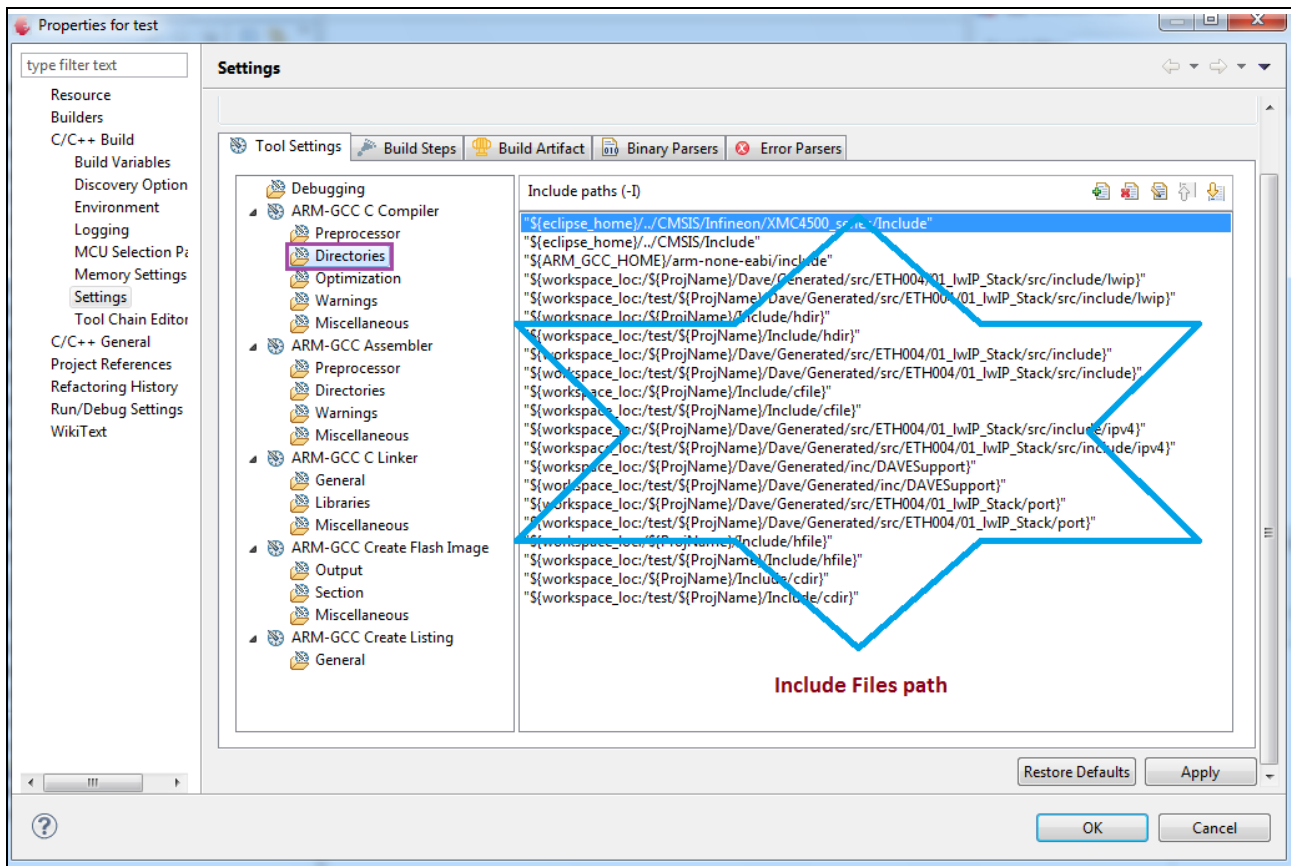
## 4.15 Project Build

Dave supports ARM GCC compiler/Linker/Assembler for building the project to generate the output file to download to the target board.

Include file path for the project can be know through project settings page.

- Right click on the project window and select properties
- Properties window will open
- Select C/C++ Build option -> Settings option on the left side
- Select Directories on ARM-GCC C Compiler option on Right side of the window

Which will display the include path of the files



**Figure 70 DAVE™ Project Build Include Path**

To know the included library path,

- Select Libraries on ARM-GCC C Linker option on Right side of the window  
Which will display the include Libraries path for the project

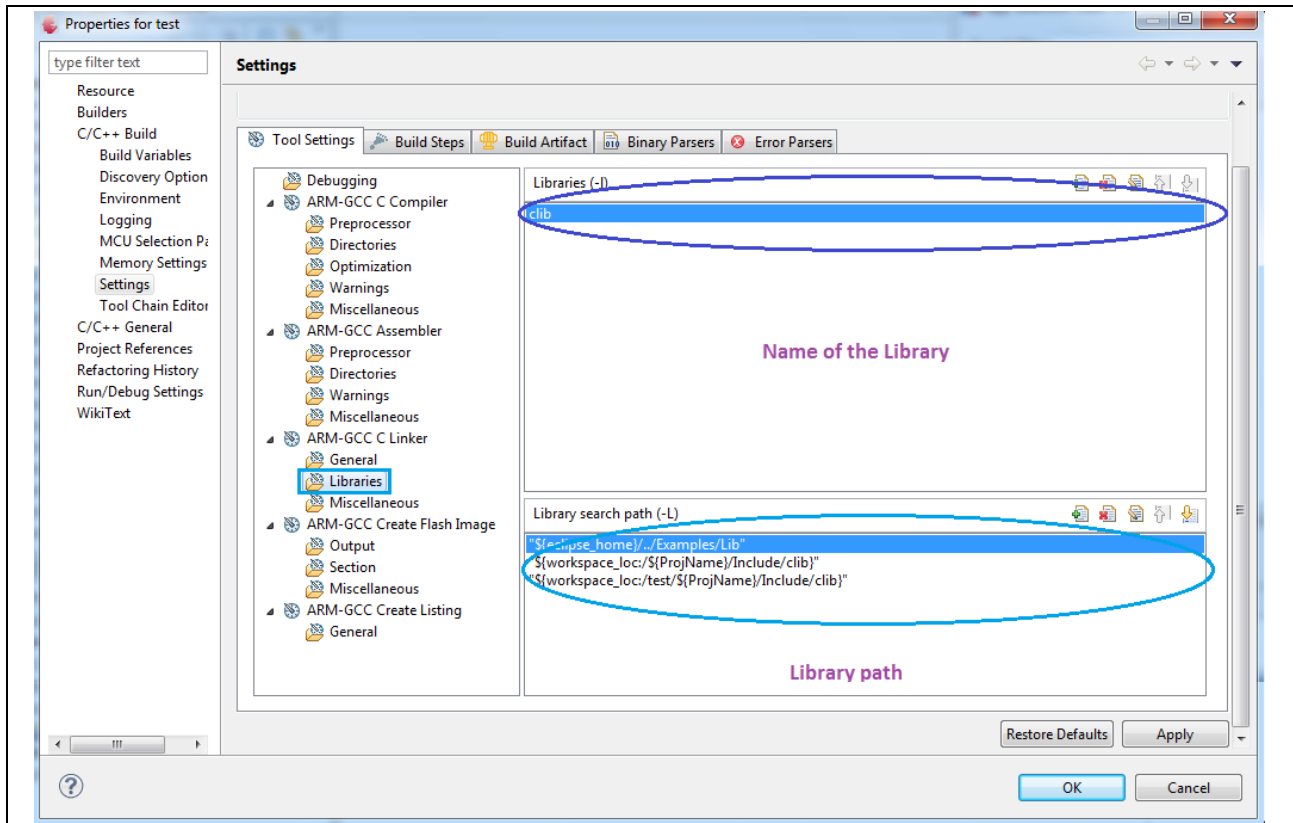


Figure 71 DAVE™ Project Build Libraries Path

## 4.16 Debugger Support

DAVE™ Integrated with TASKING C/C++ Debugger

- ➔ Click Debug menu button

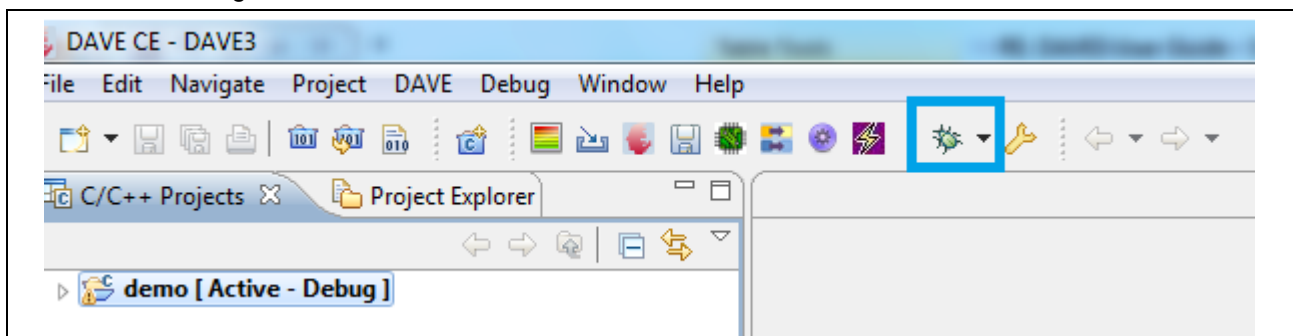


Figure 72 DAVE™ Debug Setup

- ➔ It opens TASKING C/C++ Debugger configuration window

The Debug configurations can be made by using two methods, one by using Infineon mini wiggler and other is by using JLINK (SWD/ USB).

### 4.16.1 Debugger Support – Using MiniWiggler

Infineon MiniWiggler is supported by debugger

- ➔ Click TASKING C/C++ Debug to create one instance of debug
- ➔ Check Debugger tab to know the proper device is available or not



→ Click Debug button for Debug

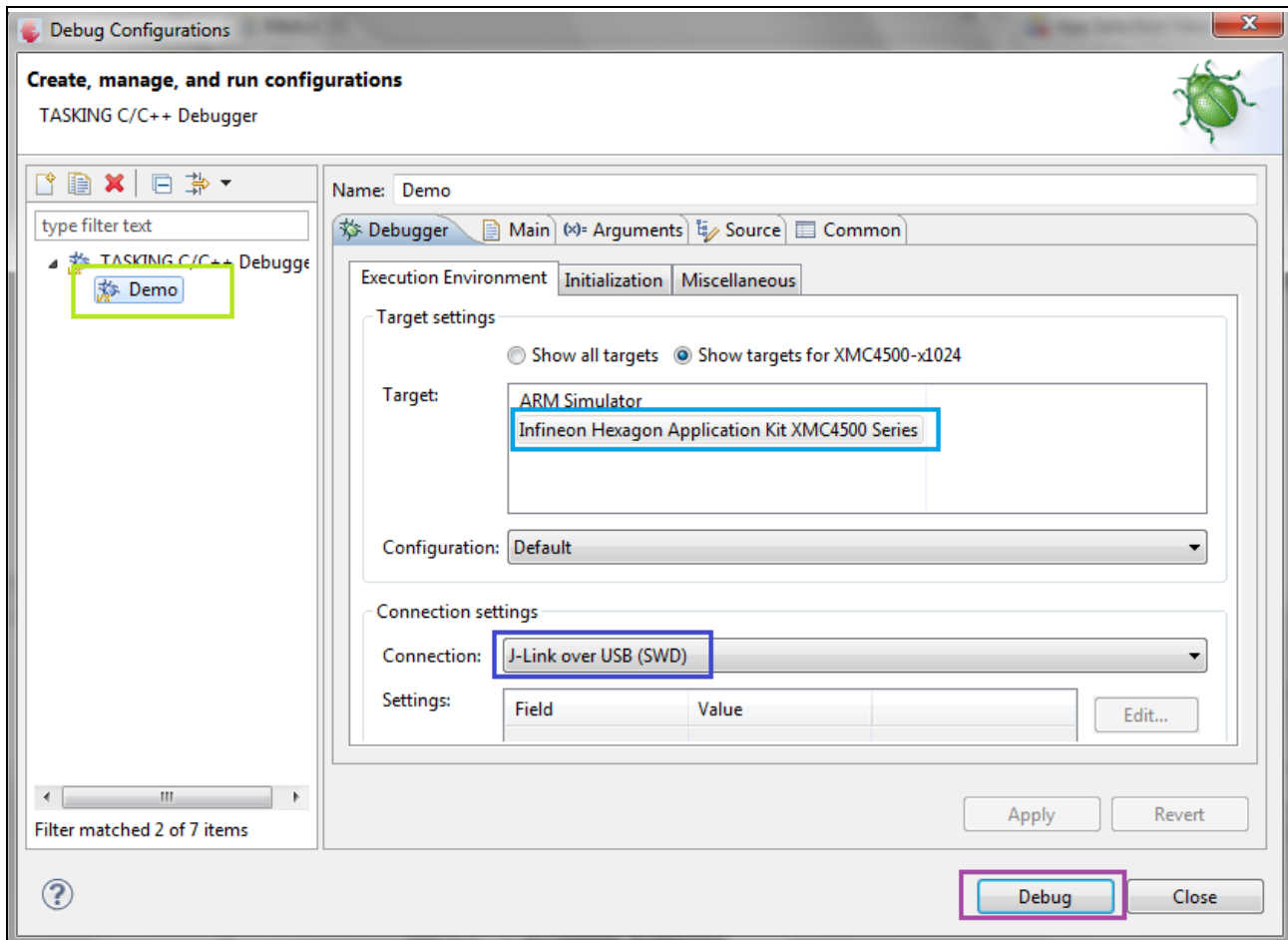
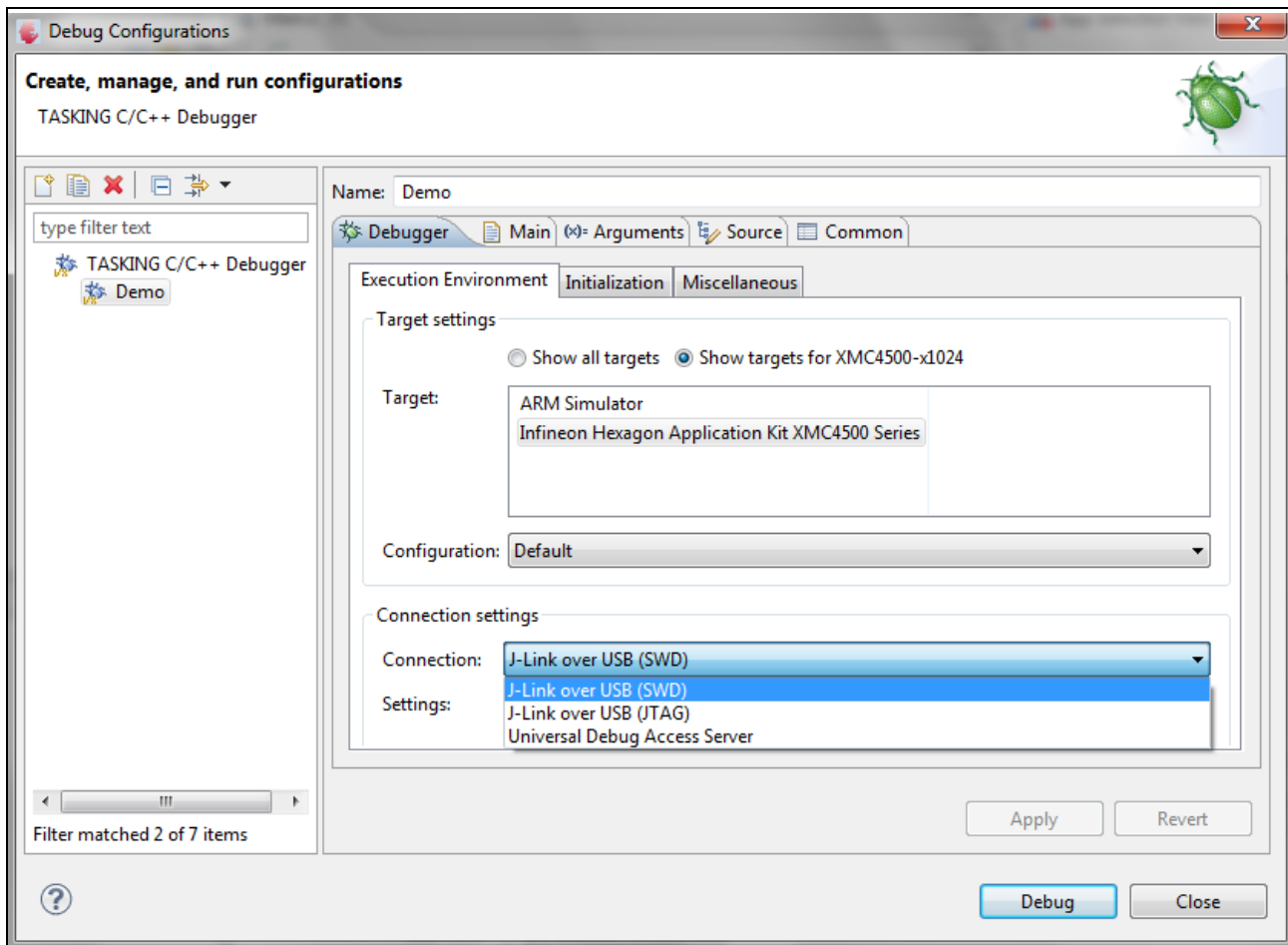


Figure 73 DAVE™ Debugger Support – MiniWiggler

## 4.16.2 Debugger Support – Using JLink

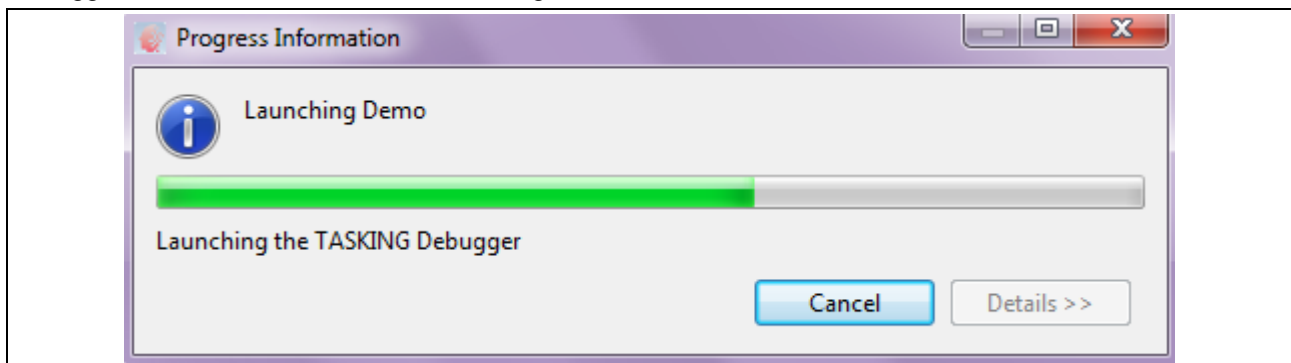
Infineon JLink is supported by debugger

- Click TASKING C/C++ Debug to create one instance of debug
- Check Debugger tab
- Select Communication Setup
- Click Debug button for Debug
  - Select J-Link over USB JTAG/SWD



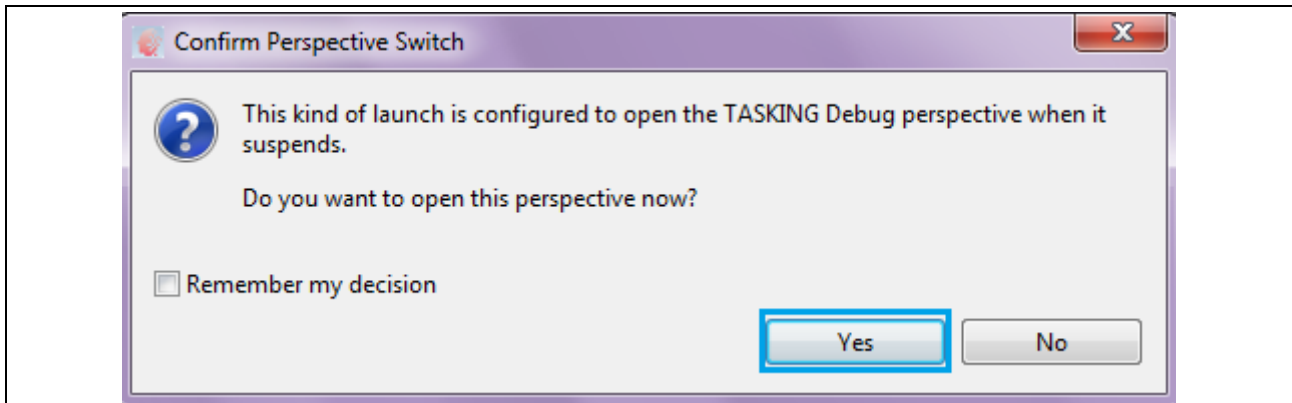
**Figure 74 DAVE™ Debugger Support – JLink JTAG/SWD options**

Debugger will start Download binaries to Target Board.



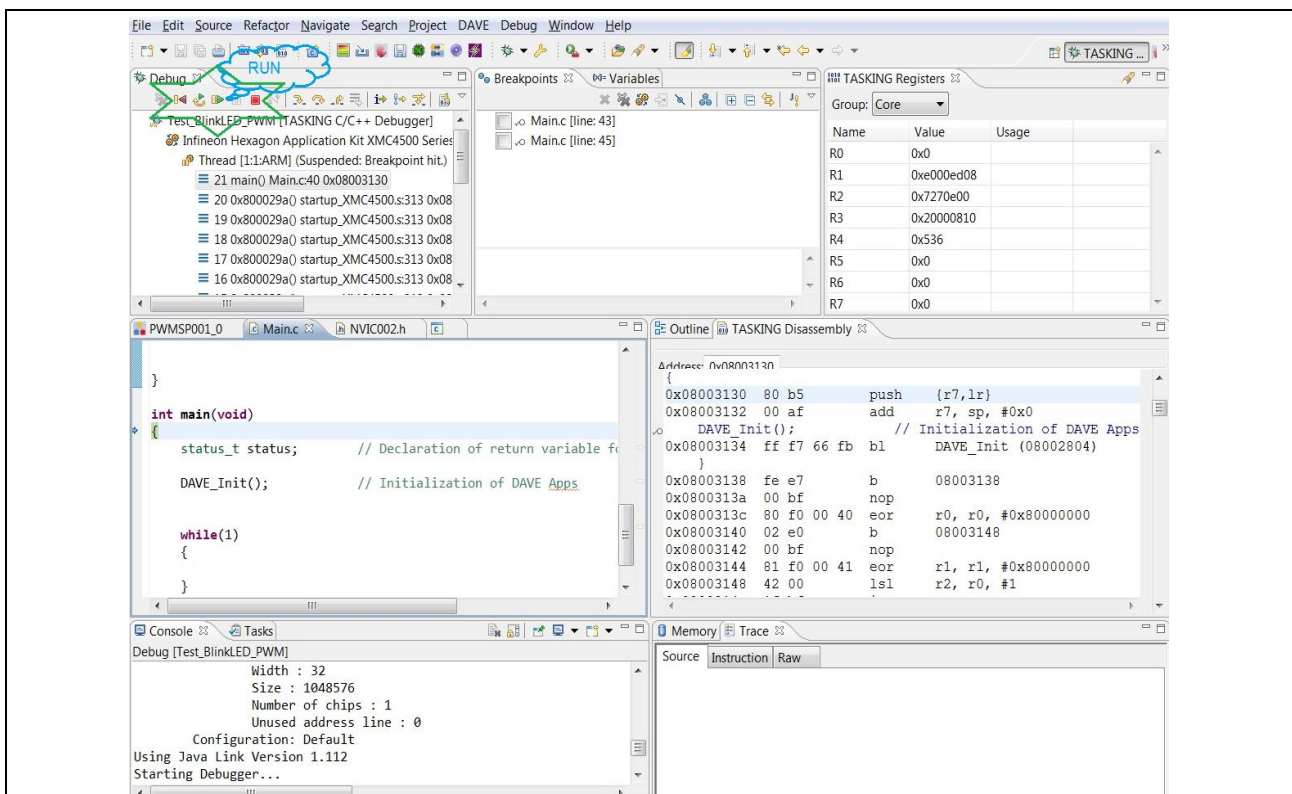
**Figure 75 DAVE™ Download to Target Board**

➔ To open Debugger Perspective Press 'Yes' button



**Figure 76 DAVE™ Download to Target Board – Confirm Perspective Switch**

- ➔ Click “Run” button from Tasking Debug perspective
- ➔ Debug features supported are
  - To Debug Variable, Breakpoint, Registers
  - C File/editor
  - Disassembly
  - Console
  - Memory/ Register

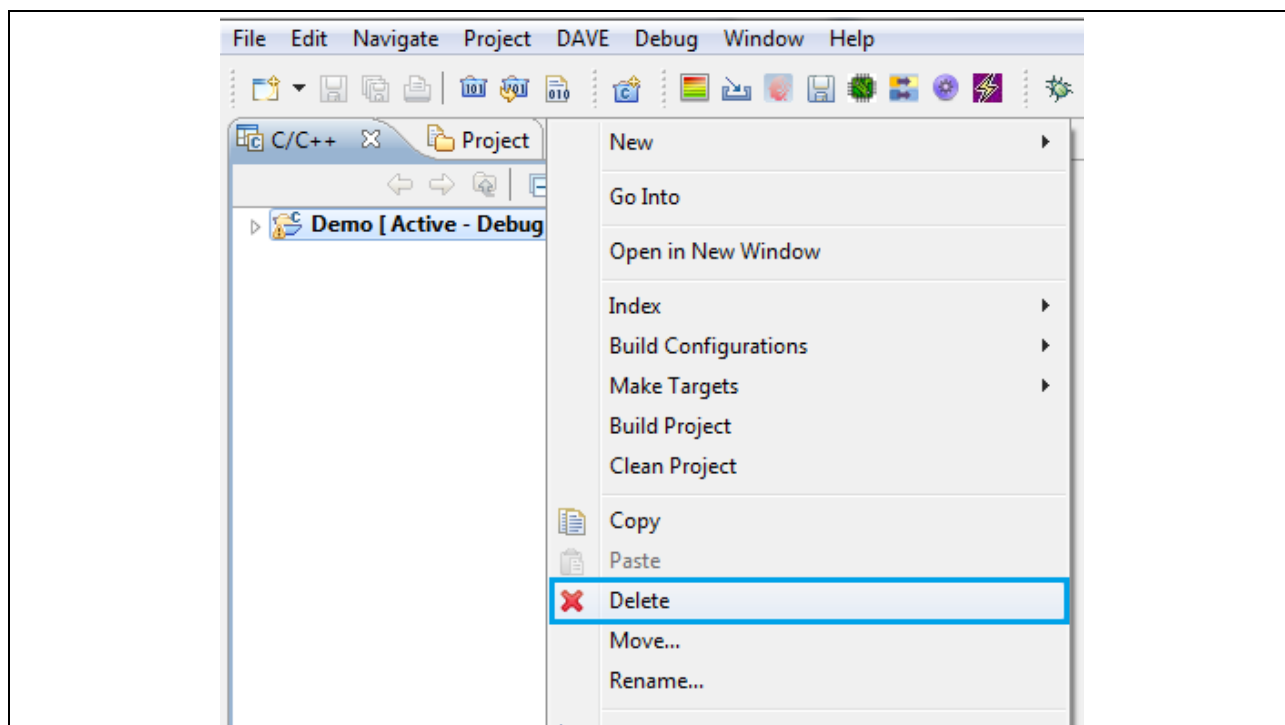


**Figure 77 DAVE™ Download to Target Board – Debug Launch**

## 4.17 Project Deletion

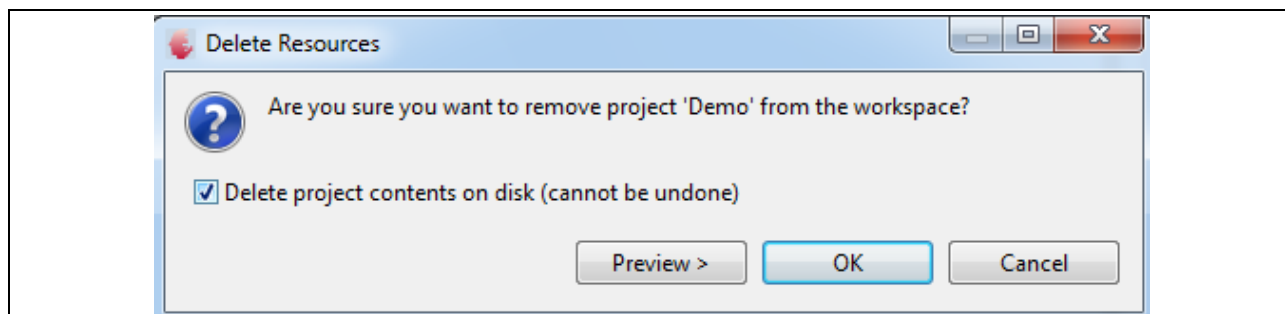
To delete existing project from Project Window/space

- ➔ Select the project, right click on the selected project and choose Delete Option.



**Figure 78 DAVE™ Project Deletion**

- ➔ By enabling the 'Delete project contents on disk' the project will be deleted from disk (local memory) and cannot be restored.

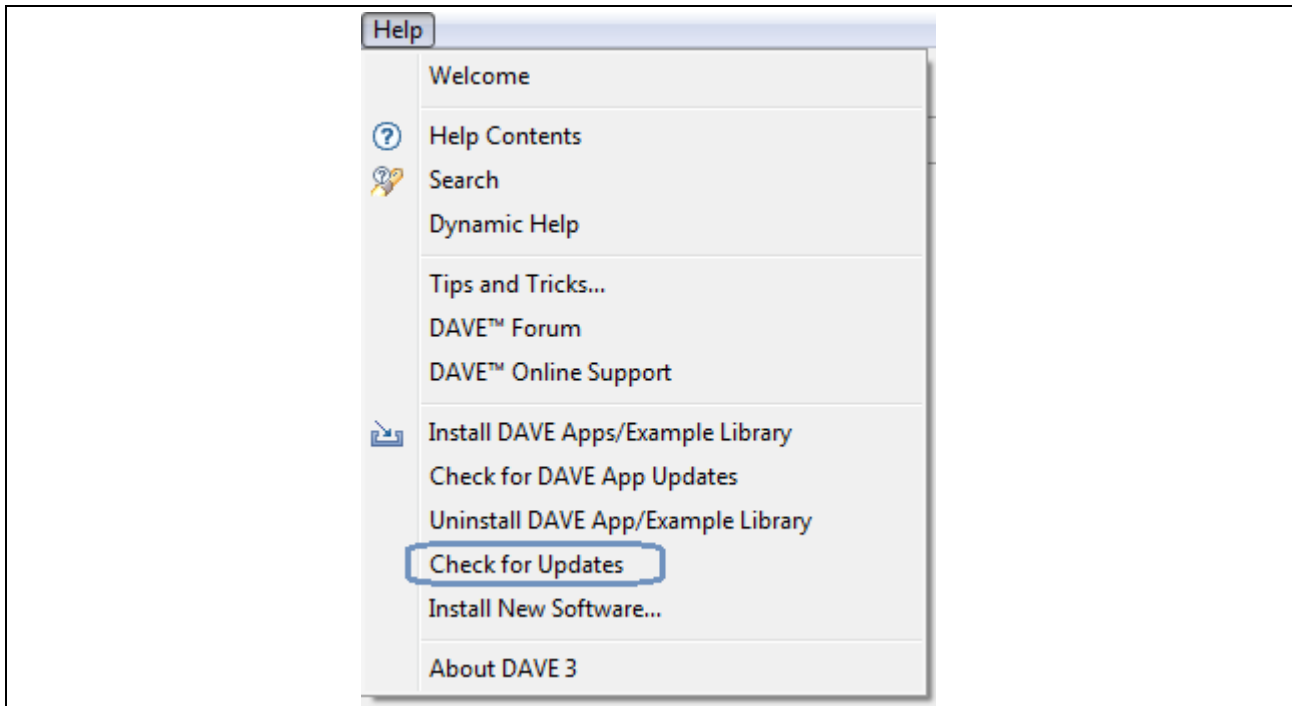


**Figure 79 DAVE™ Project Deletion – Delete Resources Message box.**

## 4.18 Software Updates

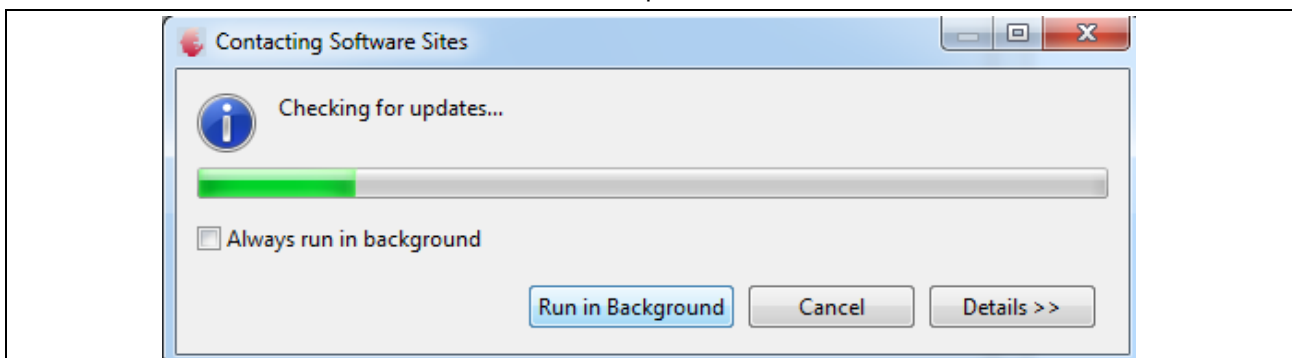
The user can check for the updates for the software by using Check for Updates option from the Help Menu.

- ➔ Click on Help menu and Select Check for Updates option.



**Figure 80 DAVE™ Check for Updates**

→ DAVE™ will take some time to check for the updates



**Figure 81 DAVE™ Checking for Updates.....**

- When there are updates on the software it will install the updates and asks to restart the workspace.
  - Restart the DAVE™ workspace
- If there are no updates found then it will display as No Updates Found.

## 5 DAVE™ Apps & Library Management

DAVE™ Apps are application use case oriented SW components that cover a wide range of use cases such as generation of PWM signals, sending or receiving messages over SPI, UART or CAN, converting analog signals and much more.

### 5.1 DAVE™ Apps

A DAVE™ App is a **self-contained, self-descriptive** implementation of an Application Use Case (configuration or dynamic behavior) in the project

#### ■ A DAVE™ App can...

- ☐ ...**generate code** into the project workspace (using Jet code generation framework)
- ☐ ...**copy files** (source or object files) into the project workspace
- ☐ ...**add include and library directories** to the project
- ☐ ...**declare resources it consumes (chip resources or other DAVE™ Apps**
- ☐ ...**provide resources** to the project (at least the parameters that determine the configuration of an App are provided resources)
- ☐ ...**provide an UI editor** to configure its parameters
- ☐ ...**use APIs provided by other Apps**

#### ■ Examples

- ☐ UART001 App – provides an UI editor for the low-level access (read/write) on the USIC configuration registers

### 5.2 Categories of DAVE™ Apps

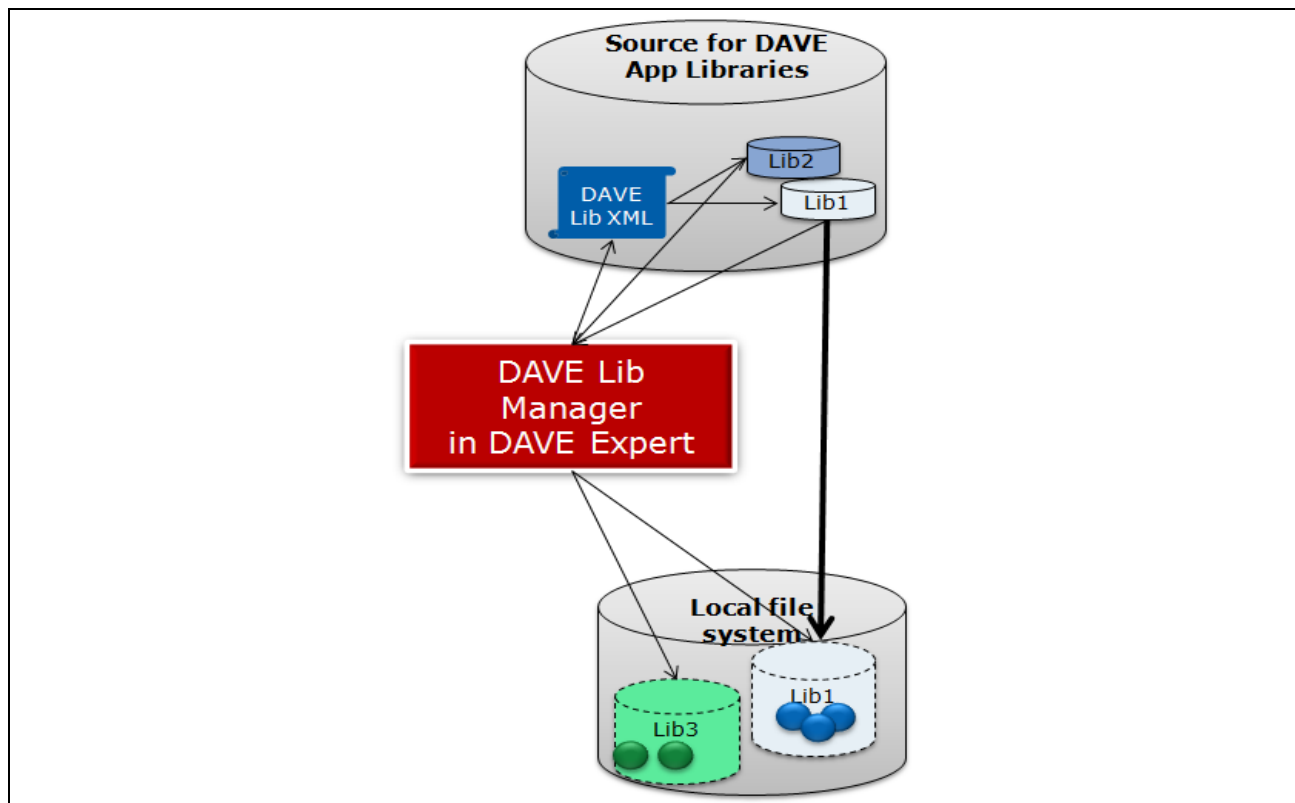
Complete Solutions	DAVE™ Apps that create a complete configurable solution, like Webserver.
Service Apps	DAVE™ Apps that provide services mainly required by other DAVE™ Apps e.g.clock, reset, debugging, interrupts, exceptions, RTOS, etc.
Middleware Apps	Apps that provide an abstraction of complex functionality e.g. communication stacks, file system or other comprehensive SW solution. Preferably these Apps use (re-use) other Apps for HW access and the RTOS App services.
Application specific Middleware Apps	DAVE™ Apps that provide high level abstractions of dedicated applications like motor control or Lighting.
Algorithm Apps	Control algorithms or mathematical solutions to use the CPU in an optimized way and to reduce programming efforts at customers.
Low Level Driver Apps or Basic Applications	HW/Peripheral related SW solution in form of APIs that can be used by the user to solve the application requirements or that can be used by other Apps to form more abstract solutions or complete applications.
Register Initialization Apps	Initialization of HW/Peripherals and simple accessing of registers for use cases those are not (yet) supported by the Low Level Driver Apps.

Table 2 DAVE™ App types

### 5.3 DAVE™ App Library

A DAVE™ Library or DAVE™ Apps Library is the container of DAVE™ Apps that should be downloaded from a source (CD, Web, local file system...) to the PC of the user in order to make these Apps available for usage with DAVE.

- For this purpose DAVE™ provides a DAVE™ Library manager that provides the functionality
  - ☐ Connection to the source
  - ☐ Open of an source specific DAVE™ App Lib XML file (update site) that contains essentially the URI of all DAVE™ App Libraries available on the respective source
  - ☐ Reading the description of all these DAVE™ App Libraries
  - ☐ Composing a selection wizard to select the appropriate DAVE™ App Lib via a hierarchy of categories or via key word selection
  - ☐ Downloading and unpack the DAVE™ App Library on the PC
  - ☐ Triggers re-initialization of the App selector



**Figure 82 DAVE™: Library Manager**

Note: The local file system is the local library store where all Apps are installed. The default location of the local library store is: C:\Users\<user\_name>\Infineon\D3LibraryStore\_xxx

## 5.4 Library groups

DAVE™ Apps are grouped in different libraries. Currently there are the following libraries:

- General standard library called DAVEApps\_Library\_001  
This DAVE™ App library contains all DAVE™ Apps that are not assigned to an application specific library
- Application specific DAVE™ Apps libraries  
These are DAVE™ Apps are dedicated to specific application segments like motor control or lighting, etc

## 5.5 DAVE™ Apps Types

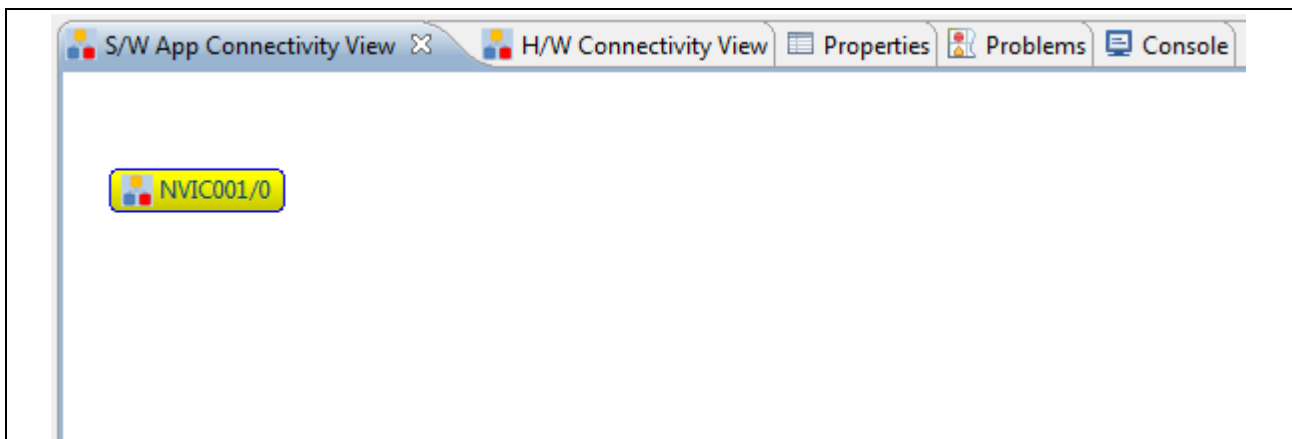
Following Types of Apps are available

- ➔ Sharable – One instance will be created first time, then onwards it will ask message to user to create new instance or Use existing one. Since these Apps are Sharable.
- ➔ Not Sharable – New Instance of App will be created each time.
- ➔ Singleton – Only one instance of App will be created per Project. New same App instance will not be allowed to create.

### 5.5.1 Sharable DAVE™ Apps

If app is Sharable each type by creating Only Once [Instance] will be Available in App Dependency View. In case if user wants one more instance it will ask for Message for creation of New Instance. By clicking OK it will create new Instance else user needs to use same. By Default One Instance will be created

For example NVIC001: creates one NVIC001/0



**Figure 83 DAVE™: Apps Type – Sharable – NVIC001/0**

By creating again it will ask for message depending on user Input It will create 'New Instance'. Select New Instance and Press OK to create new instance 'NVIC001/1'. Or select Existing Instance and press OK to use the old one.

Example 1: NVIC001 App



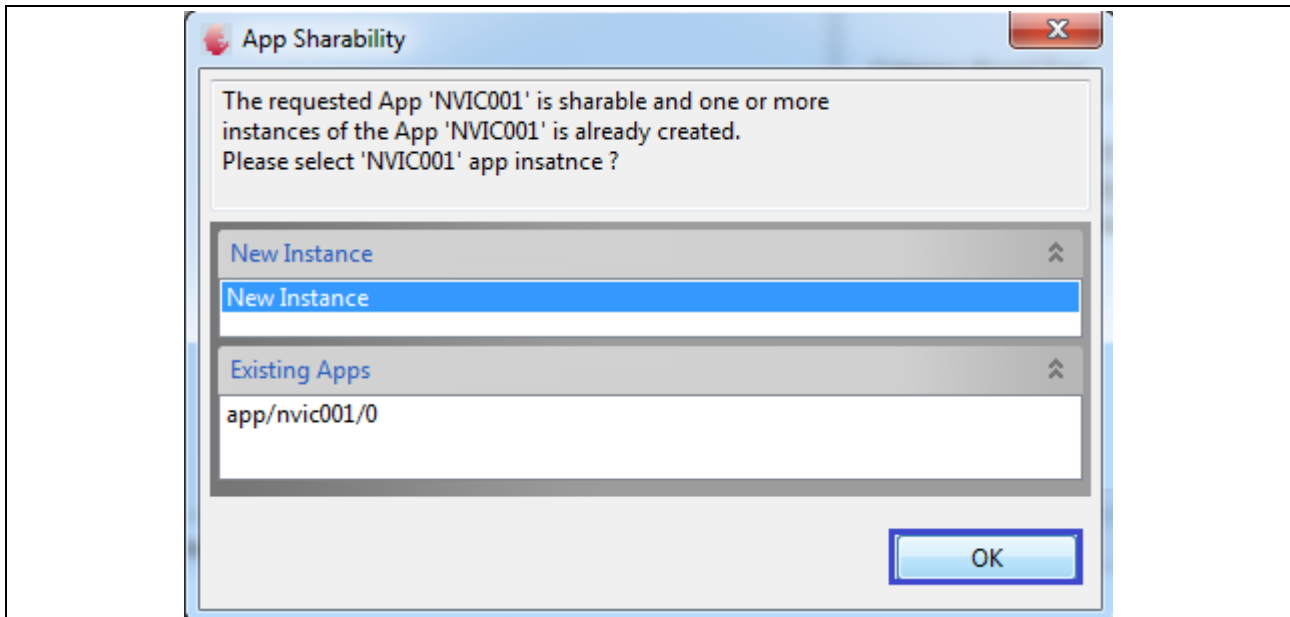


Figure 84 DAVE™: Apps Type – Sharable – NVIC001/0 – App Sharability

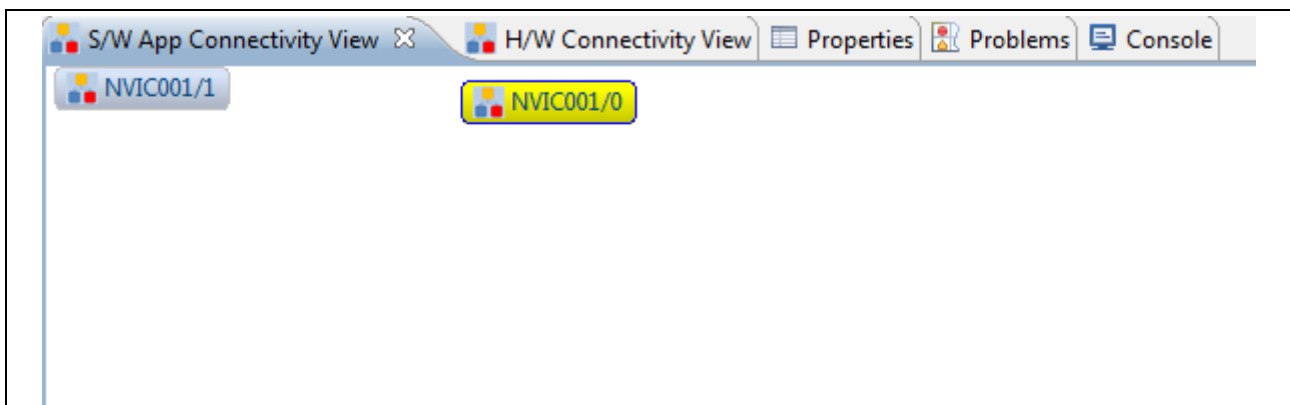


Figure 85 DAVE™: Apps Type – Sharable – NVIC001/1

Example 2: CNT001 App

Select CNT001 app, it will create App instance CNT001/0 with resource consuming CCU4GLOBAL/0

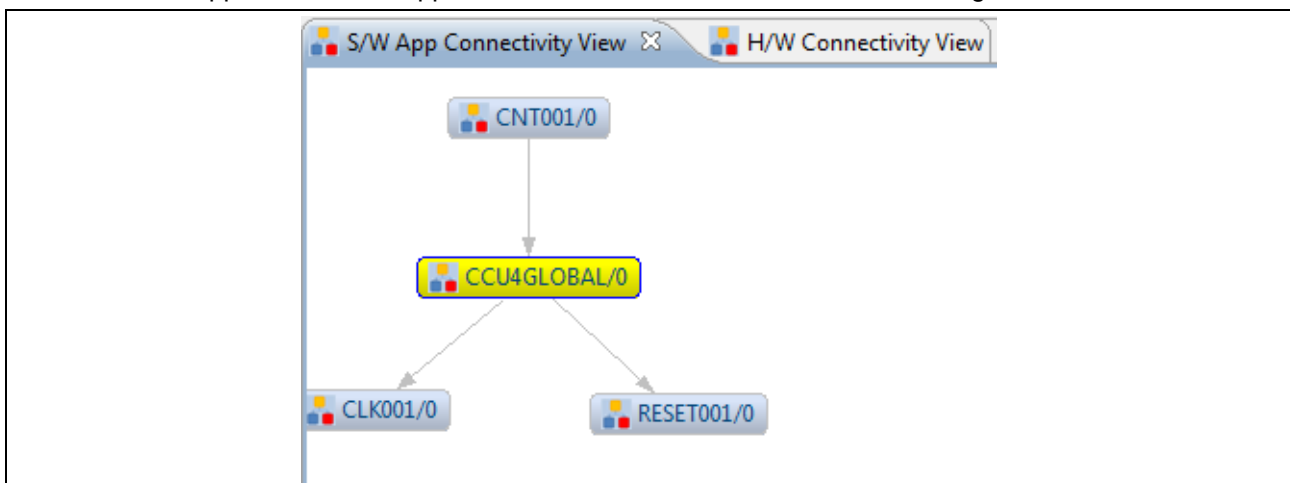
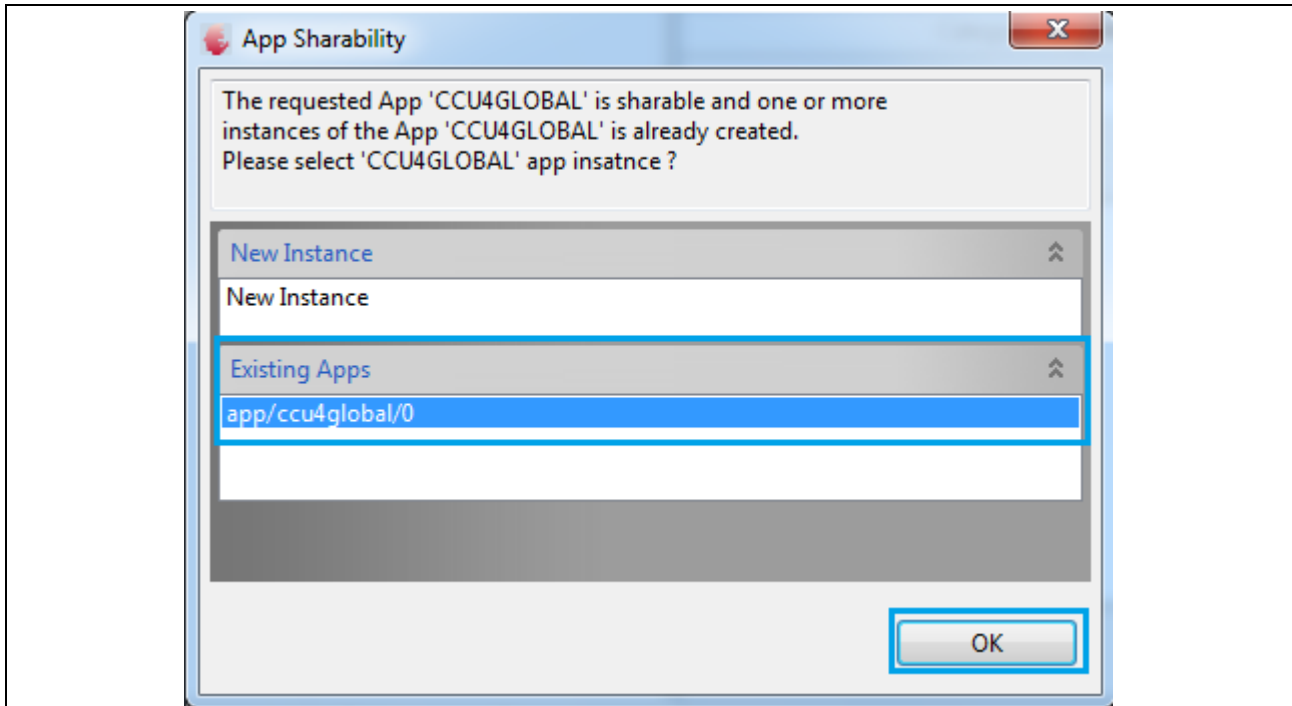


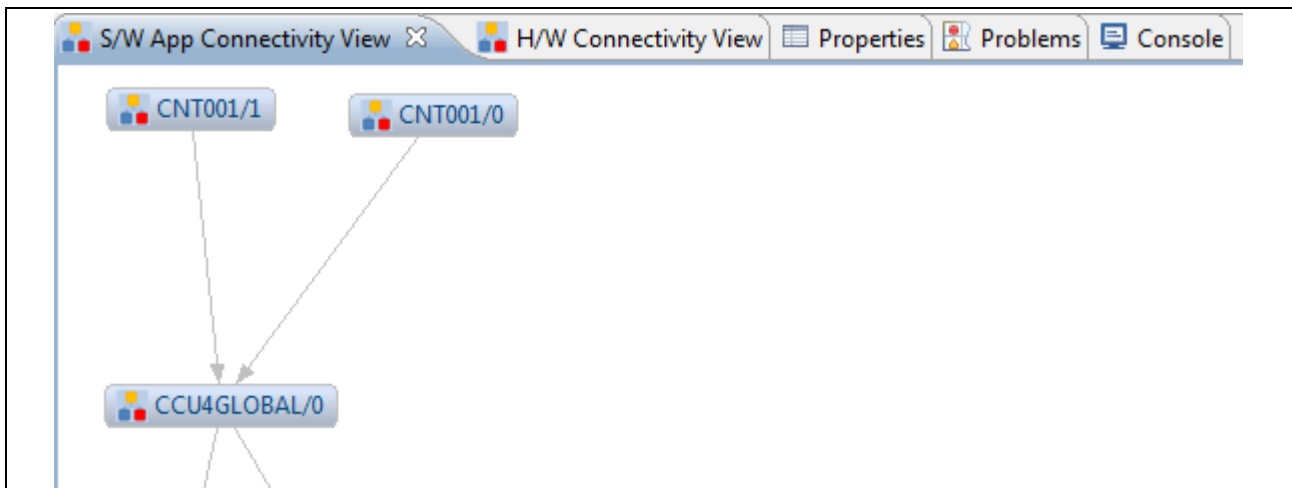
Figure 86 DAVE™: Apps Type – Sharable – CNT001/0 with CCU4GLOBAL/0

Again upon selection of CNT001 New Instance App Sharability – Upon selection of existing App, it will not create the new instance of CCU4Global App.



**Figure 87 DAVE™: Apps Type – Sharable – CNT001/0 – App Sharability – Existing App**

Upon selection of Cancel - existing app will be used.



**Figure 88 DAVE™: Apps Type – Sharable – CNT001/1 with CCU4GLOBAL/0**

## 5.5.2 Not Sharable DAVE™ Apps

- ➔ If app is Not Sharable each type by creating App new App Instance will be available in App Dependency View

E.g.: IO002 App

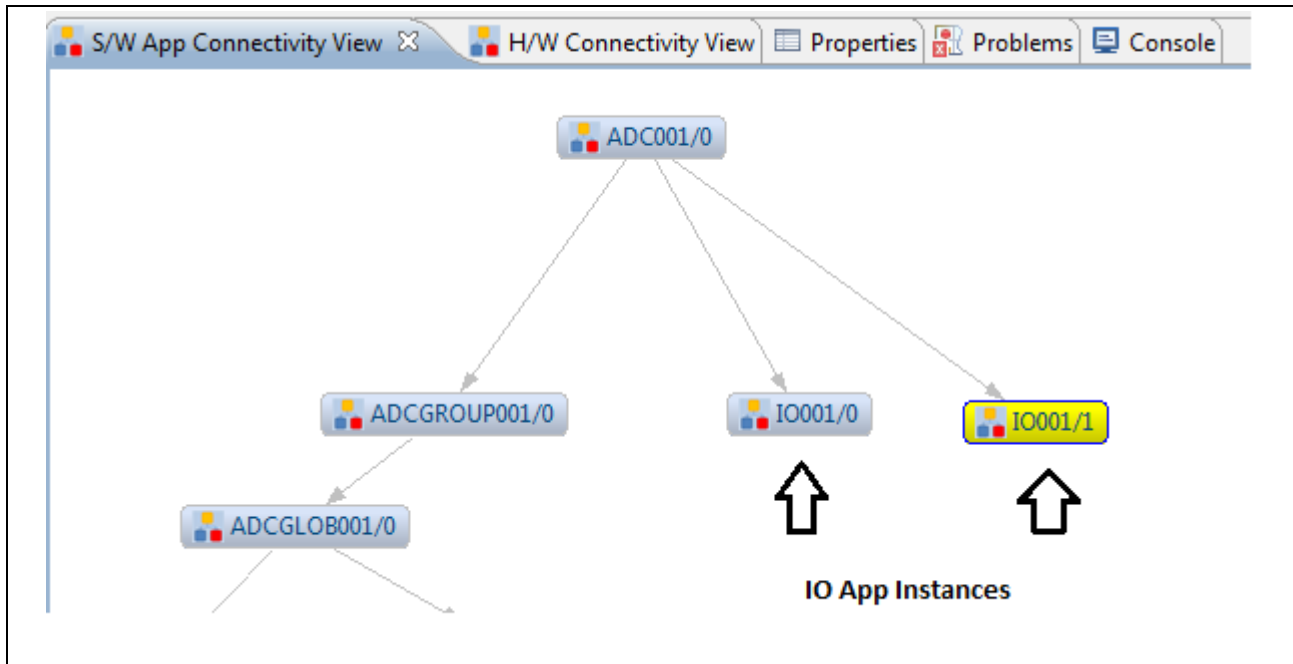
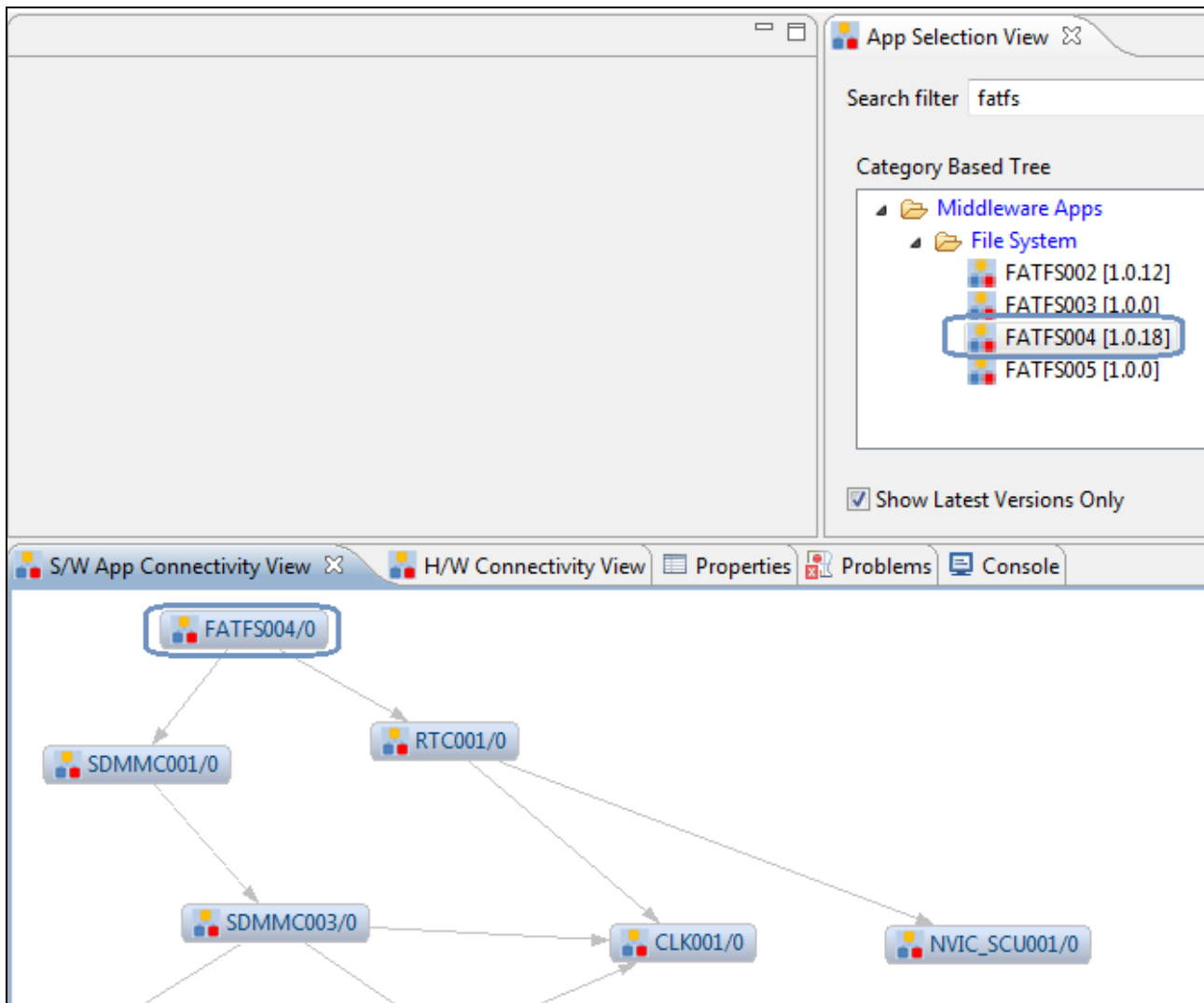


Figure 89 DAVE™: Apps Type – Not Sharable

### 5.5.3 Singleton DAVE™ Apps

- Only One instance will be created per project, and then onwards even try to create it will not allow creating new instance of same App.



**Figure 90 DAVE™: Apps Type – Singleton**

Following are category based types of Apps

➔ System Defined Apps

- Apps which will perform the system based tasks and are low level Apps. These Apps are not visible to the user in app Selection/ App Dependency views but available in the local Library store.

E.g.: Dave Support, DEBUG...

➔ Service Apps

- Apps which provide the service related functionalities like flash programming, memory management, interrupt...

E.g.: GEN002, NVIC002, LM001...

➔ Peripheral Specific Apps

- Apps which uses peripheral of the microcontroller like timers which uses CCU channels as its main base, IO ports, communication peripherals like CAN, USIC channels

E.g.: CNT001, PWMSP001, PWMSP002, I2C001, UART001, CAN001...

➔ Basic Applications

- Apps which provide the basic operation like communication, signal capture etc. And which uses internally the basic peripherals of the microcontroller

Egg: CAP001, SYSTM001, ETH004...

#### ➔ Middleware Apps

- Apps which provide the next level operation compare to the low level drivers/peripherals like file system, stack ... and consumes low level Apps

E.g.: SLTHA001, ETH007...

#### ➔ Complete solutions

- Apps which are having complete solutions with them and any additional App is not required to fulfill the App functionality requirement.

E.g.: SNMP001, WEBSERVER001...

The above mentioned App categories are used for searching the App in category based.

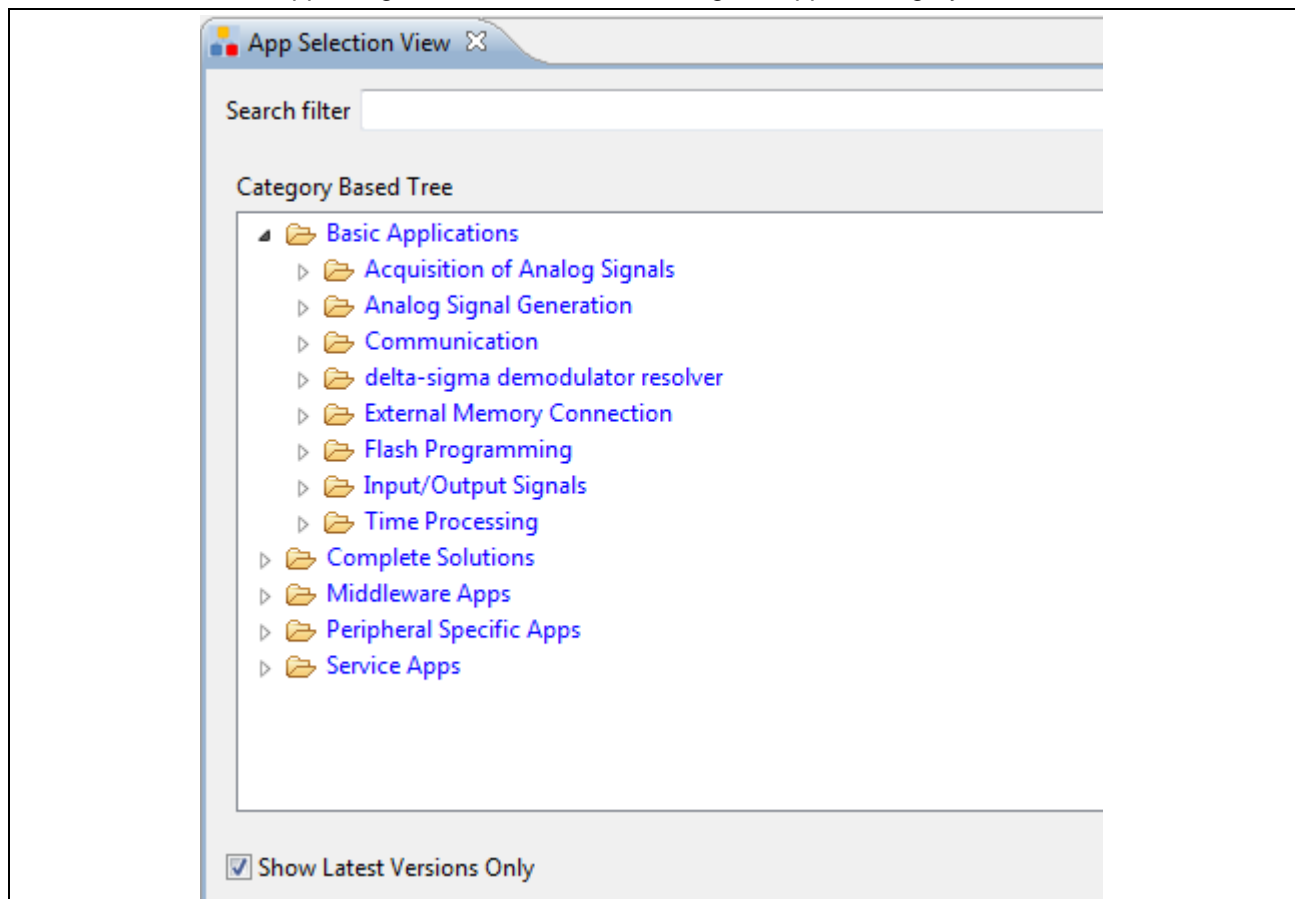
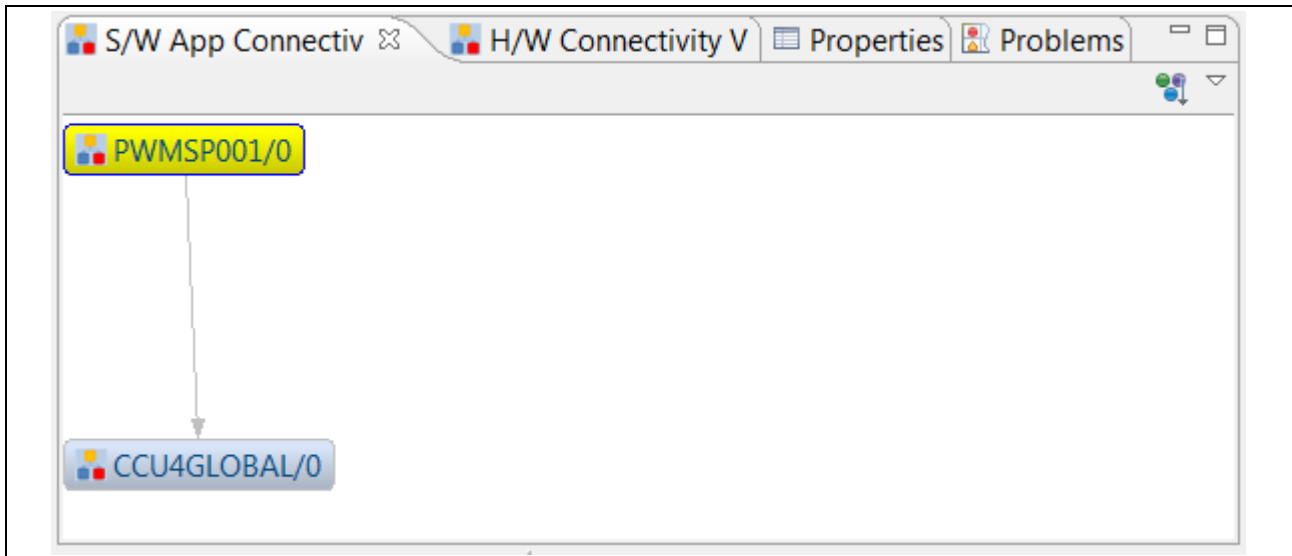


Figure 91 DAVE™: Apps Type –Category based

## 5.6 DAVE™ App Dependency controls

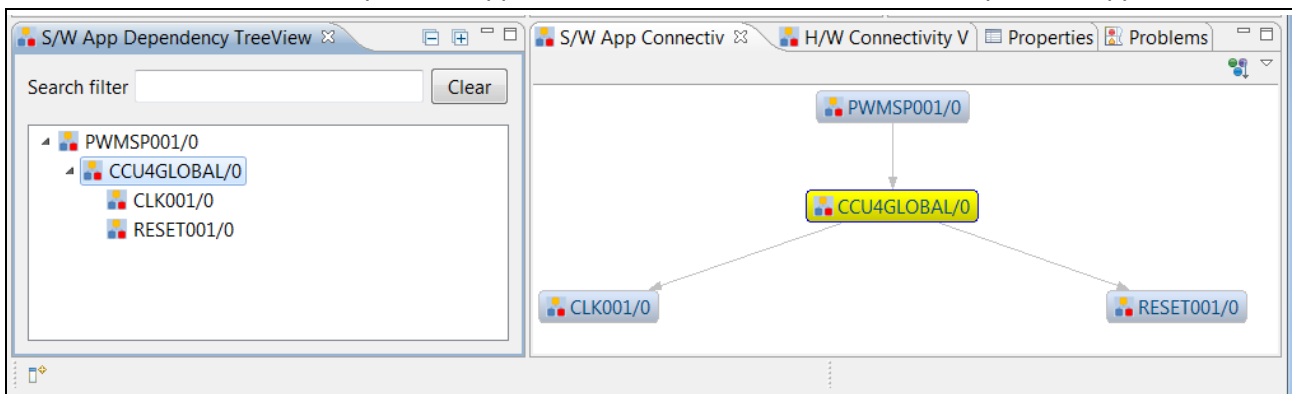
App Dependency View will show the 1<sup>st</sup> level depended apps of the main App.

- Ex: If PWMSP001 app is selected, it will be as shown below



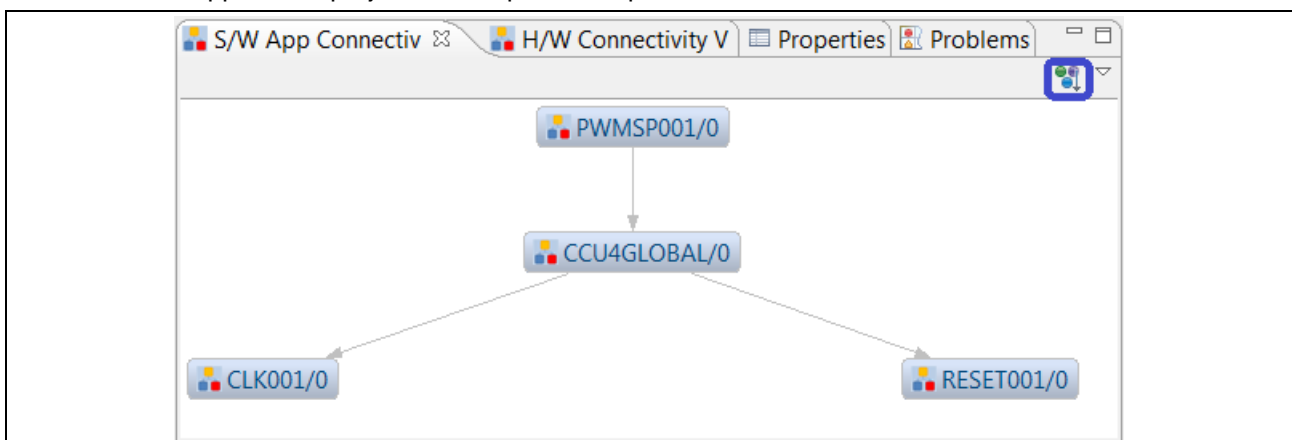
**Figure 92 DAVE™: S/W App Connectivity – Hierarchy level 1**

- If the next level dependent App is selected it will show the next level dependent Apps



**Figure 93 DAVE™: S/W App Connectivity – Hierarchy level 2**

- Click on 'Refresh & Restore both App Tree & Graph Views' option on S/W App Connectivity, all the apps in the project and respective dependencies shall be shown



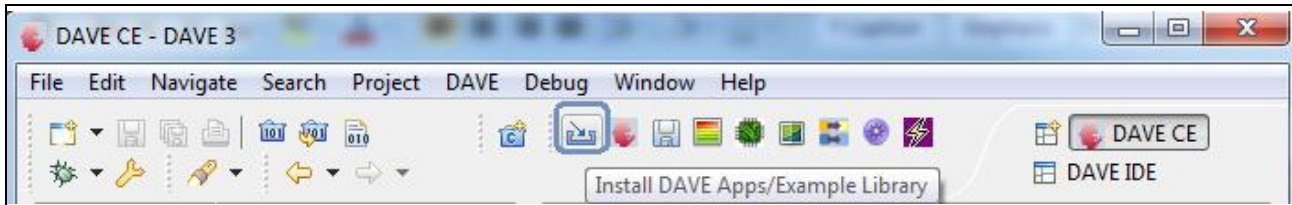
**Figure 94 DAVE™: S/W App Connectivity – Hierarchy level 3**

If a DAVE™ App is required by another App then we call this a low level App for the App that requires this App. The requiring App is called top level App. In figure 94, CCU4Global is a low level App for PWMSP001 and a top level App for CLK001 or RESET001.

## 5.7 Install DAVE™ Apps

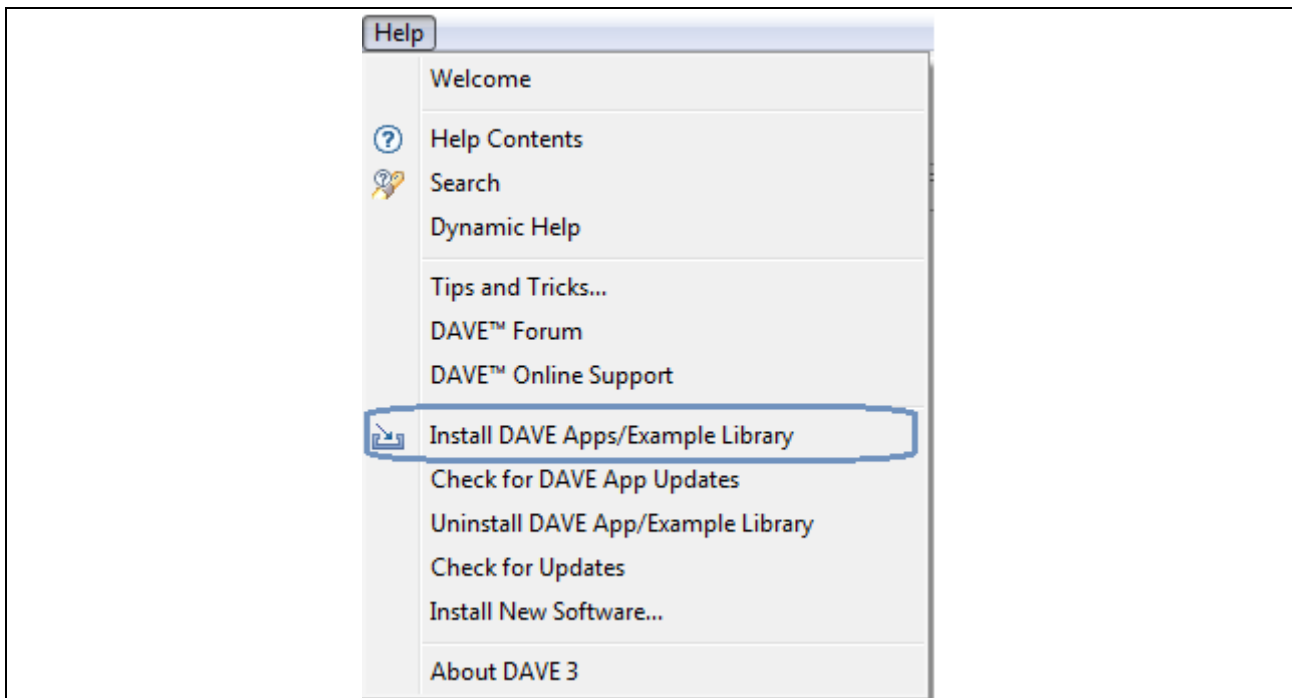
Apps from the CD/Web/... can be downloaded to DAVE™ library store via Install Apps. Apps in the CD/Web/... will be in the format of eclipse update site. By selecting the update site user can install apps to their local library store by following procedure. Installing Apps from Update site can be done in two ways

- ➔ Install via Install Apps Menu button



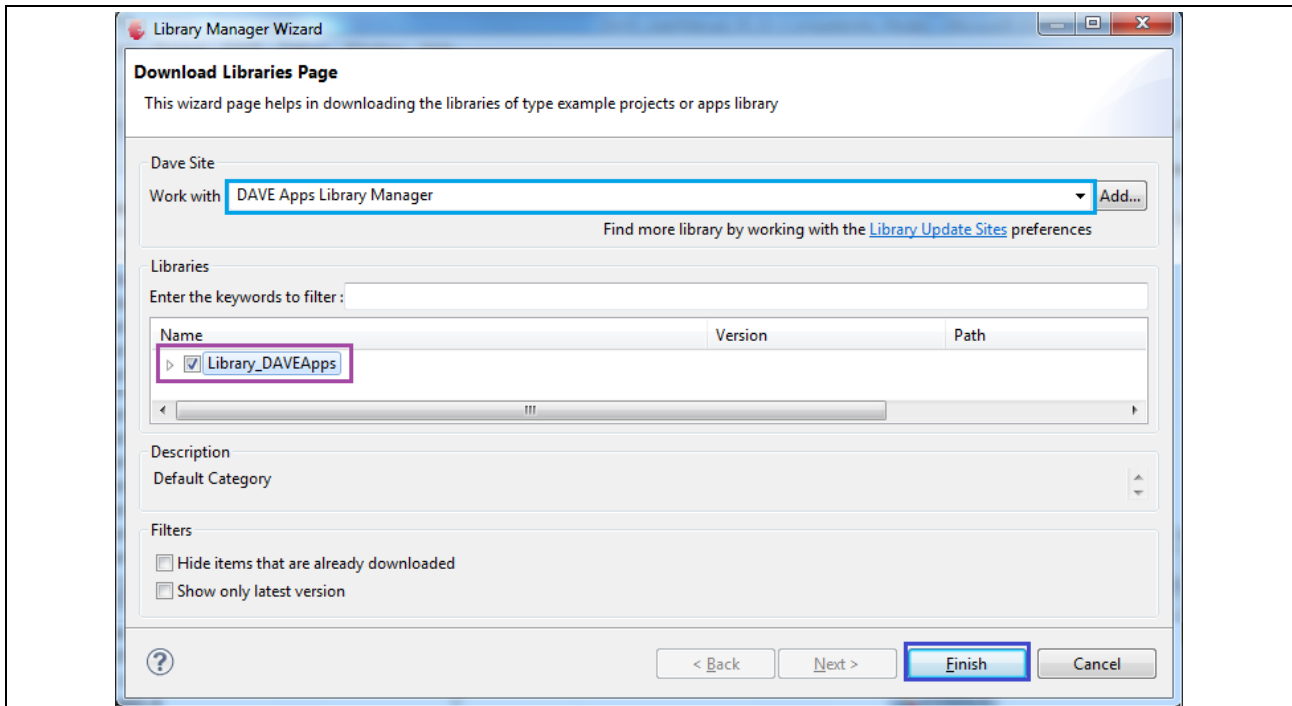
**Figure 95 DAVE™: Install Apps - via Install Apps Menu button**

- ➔ Click 'Install Apps' from Help Menu



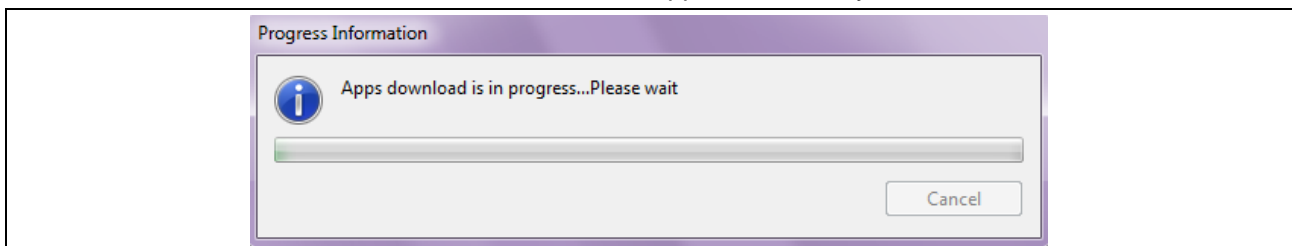
**Figure 96 DAVE™: Install Apps/Example Library - via Help Menu**

- ➔ By clicking Install DAVE™ Apps/Example Library, 'Download Library page' wizard will open
- ➔ Please Enter the name of the URL in Work with option or choose from drop down menu [DAVE™ Apps Library Manager]
  - Add the Link
  - Check the Checkbox Library\_DaveApps
  - Click Finish to Install the Apps.



**Figure 97 DAVE™: Download Libraries Wizard**

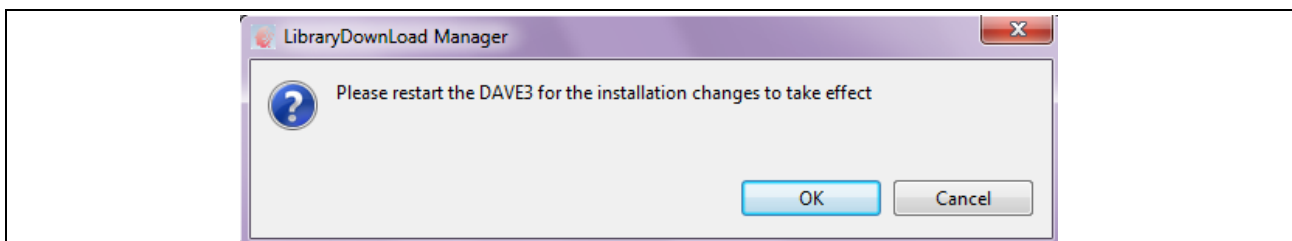
→ DAVE™ will take some time to install selected app to local library store



**Figure 98 DAVE™: Progress Information**

→ After successful installation, DAVE™ will ask for Restart

- Please click 'OK' button to for Restart the DAVE



**Figure 99 DAVE™: Library Download Manager Restart**

- If Cancel option is selected then the Library may not install correctly.
- After successful restart it will open last Workspace
  - Please select the workspace and then click OK

Note: If library is corrupted due to any factor, a fresh copy of App library and device drivers has to be installed.

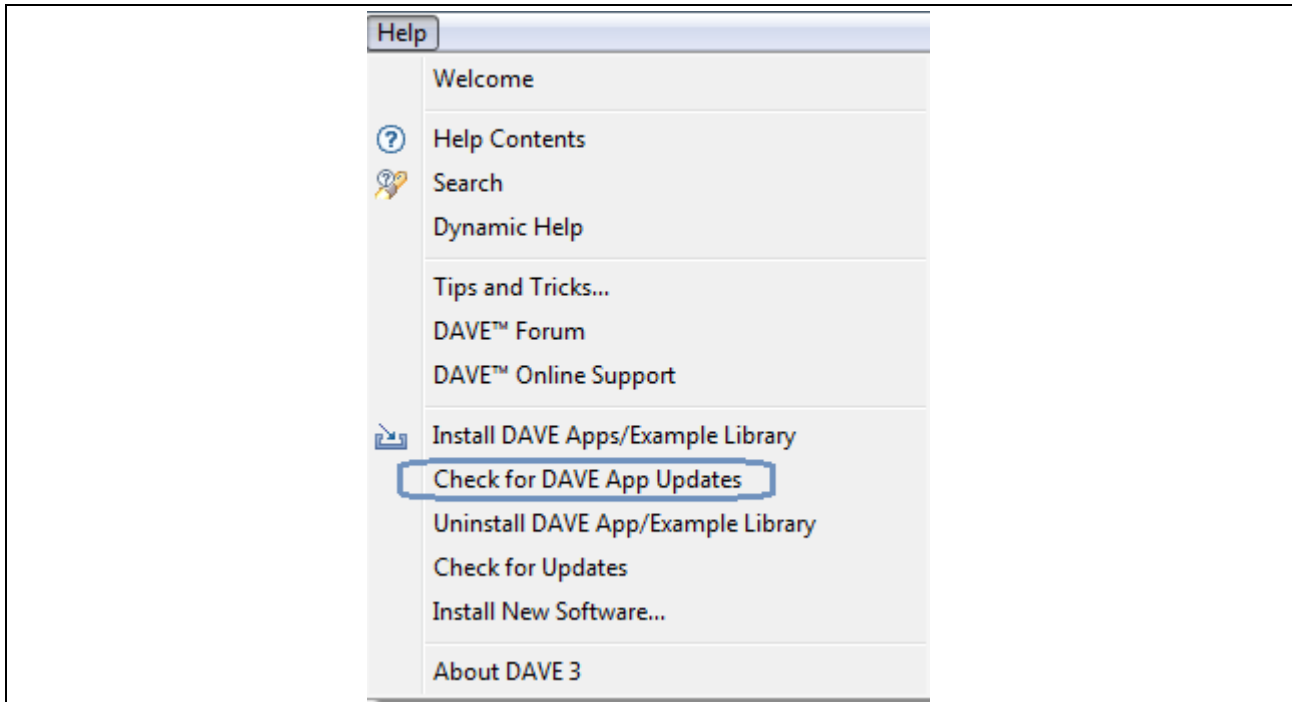


## 5.8 Update & Uninstall DAVE™ Apps

Existing App data base can be updated with new version by using update method and can be deleted from the App store using Uninstall Apps.

### 5.8.1 DAVE™ App update

- ➔ Select Check for DAVE™ App Updates from Help Menu



**Figure 100 DAVE™: Check for DAVE™ App Updates**

- ➔ Select the App Library and Click Update.

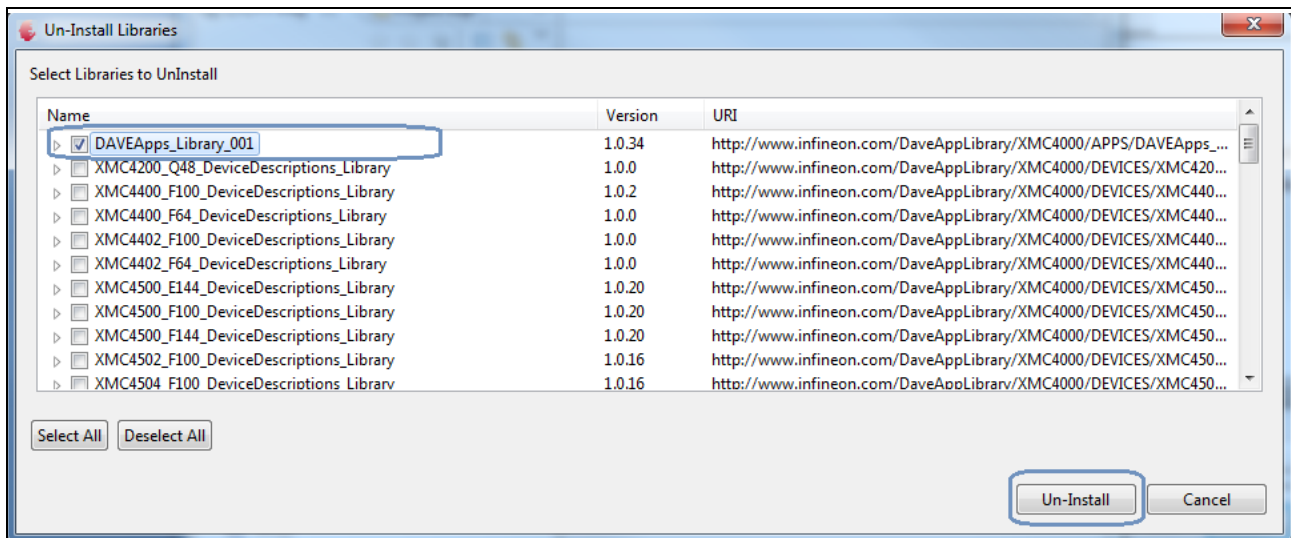
Note:

1. When there is a update on the library for selected in local library store, it will be updated
2. When there is no update on the library for selected in local library store, it will display as “No Updates Found”.

### 5.8.2 DAVE™ App uninstall

To uninstall the selected App library from store use following method,

- ➔ Select Uninstall DAVE™ App/Example Library from Help Menu
- ➔ Select the DAVE™ App/Example Library and Click Un-Install.



**Figure 101 DAVE™: Un-Install DAVE™ Apps/Example Library**

## 6 Menu

### 6.1 Install App Library

Installs DAVE™ Apps/Example Libraries

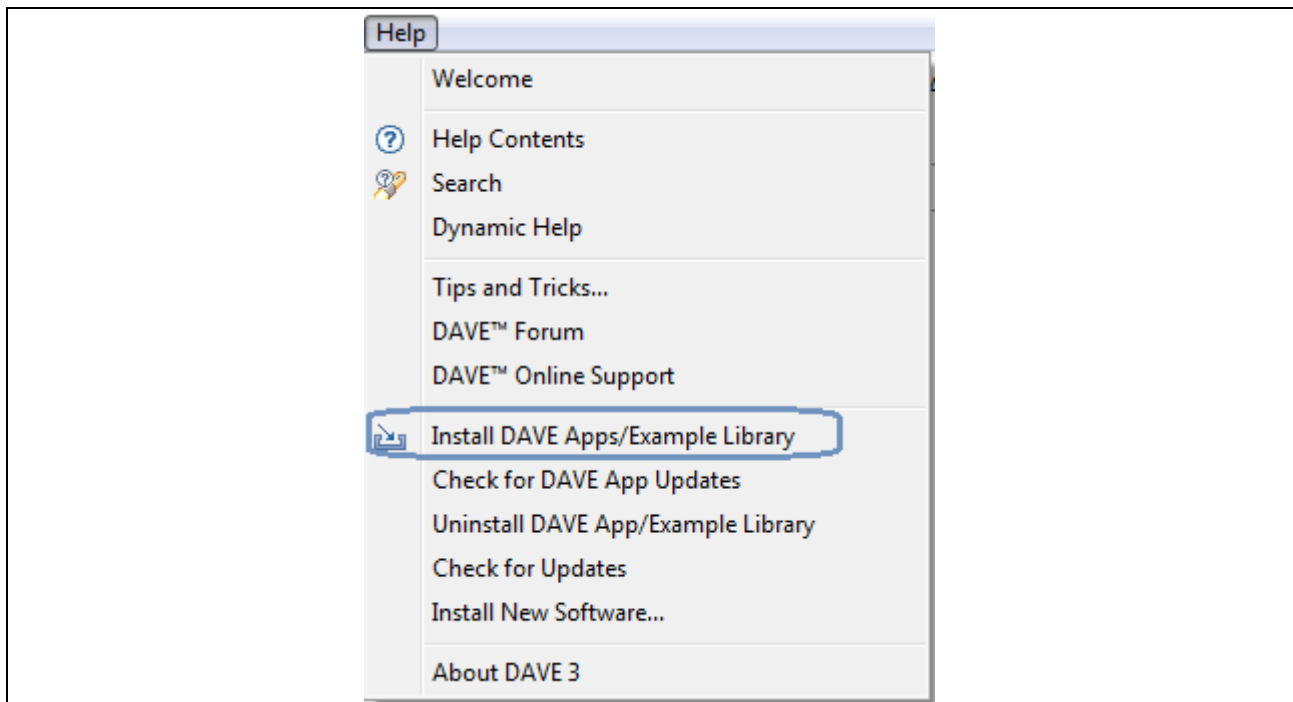
- Click on 'Install DAVE™ Apps/Example Library'



**Figure 102 DAVE™ Menu - Install DAVE™ Apps/Example Library**

Select DAVE™ Menu

- Click on 'Install DAVE™ Apps/Example Library'



**Figure 103 DAVE™ Menu – Help/Install App Library via Help Menu**

## 6.2 DAVE™ IDE Project Wizard

### → Create new DAVE™ Projects

- New Project creation Click on DAVE™ IDE New Project Wizard Menu

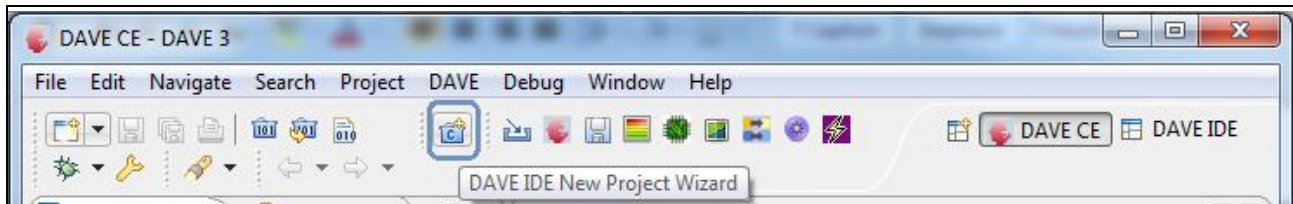


Figure 104 DAVE™ Menu – DAVE™ IDE New Project Wizard

### → Select DAVE™ Menu

- Select DAVE™ Project

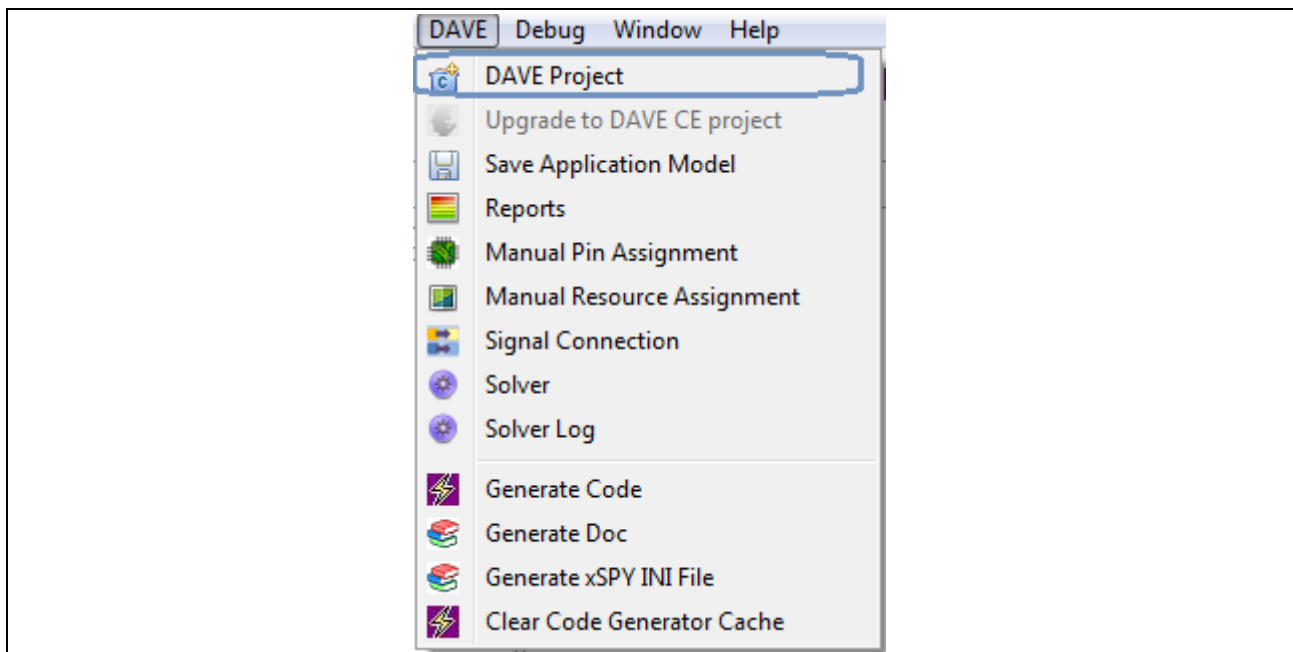


Figure 105 DAVE™ Menu – DAVE-> DAVE™ Project

### → File Menu

- Select New-> DAVE™ Project

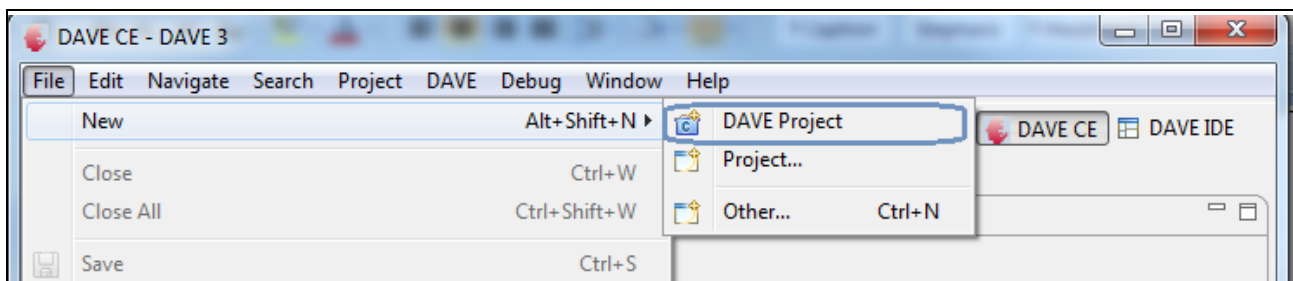


Figure 106 DAVE™ Menu – File->New->DAVE™ Project

### 6.3 Upgrade to DAVE™ CE project

This option converts DAVE™ IDE Projects (Easy Start/ Empty Main) to Code Engine Project supporting Code Generation functionality

→ Select Upgrade to DAVE™ CE project menu



Figure 107 DAVE™ Menu – Upgrade to DAVE™ CE project

→ Select DAVE™ menu and Upgrade to DAVE™ CE project option

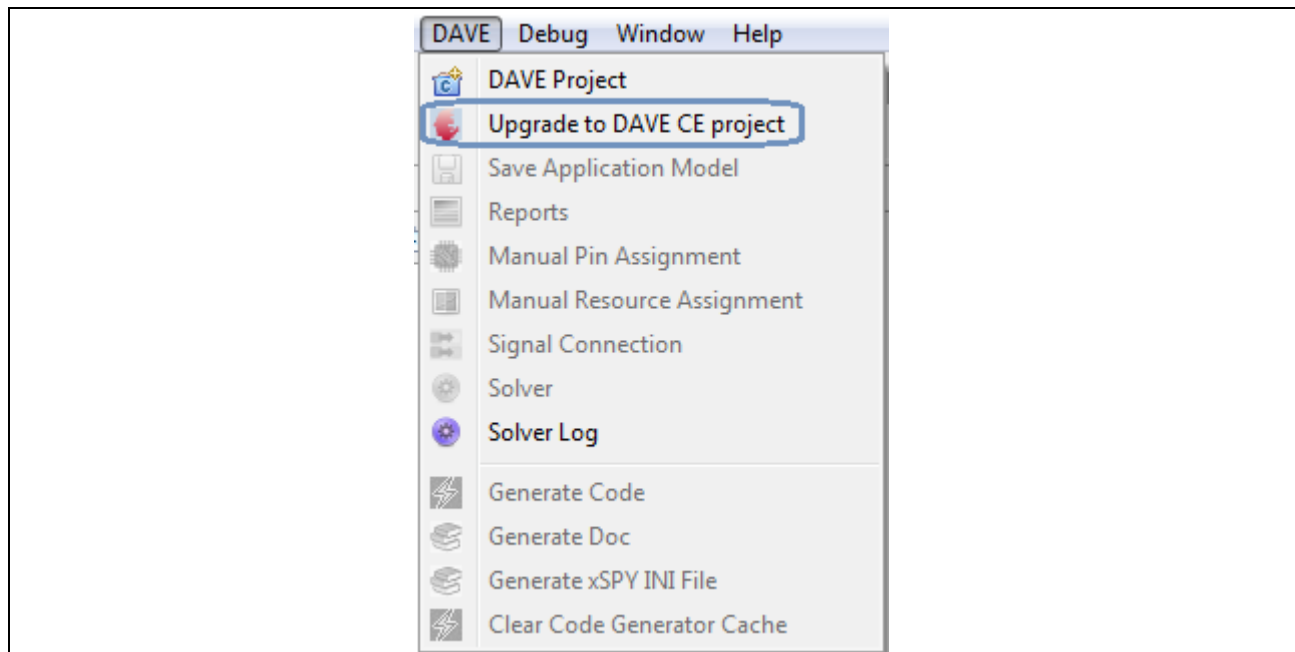


Figure 108 DAVE™ Menu – DAVE/Upgrade into DAVE™ CE project

### 6.4 Save Application Model

Application model save option below is recommended before exporting a DAVE™ Project. This option saves/persists the application model to a project file.

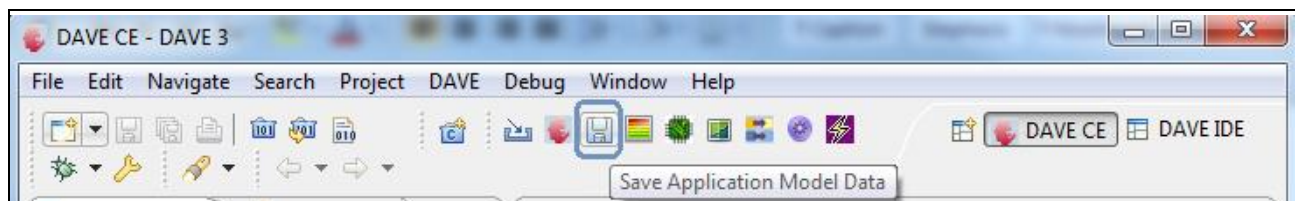


Figure 109 DAVE™ Menu - Save Application Model

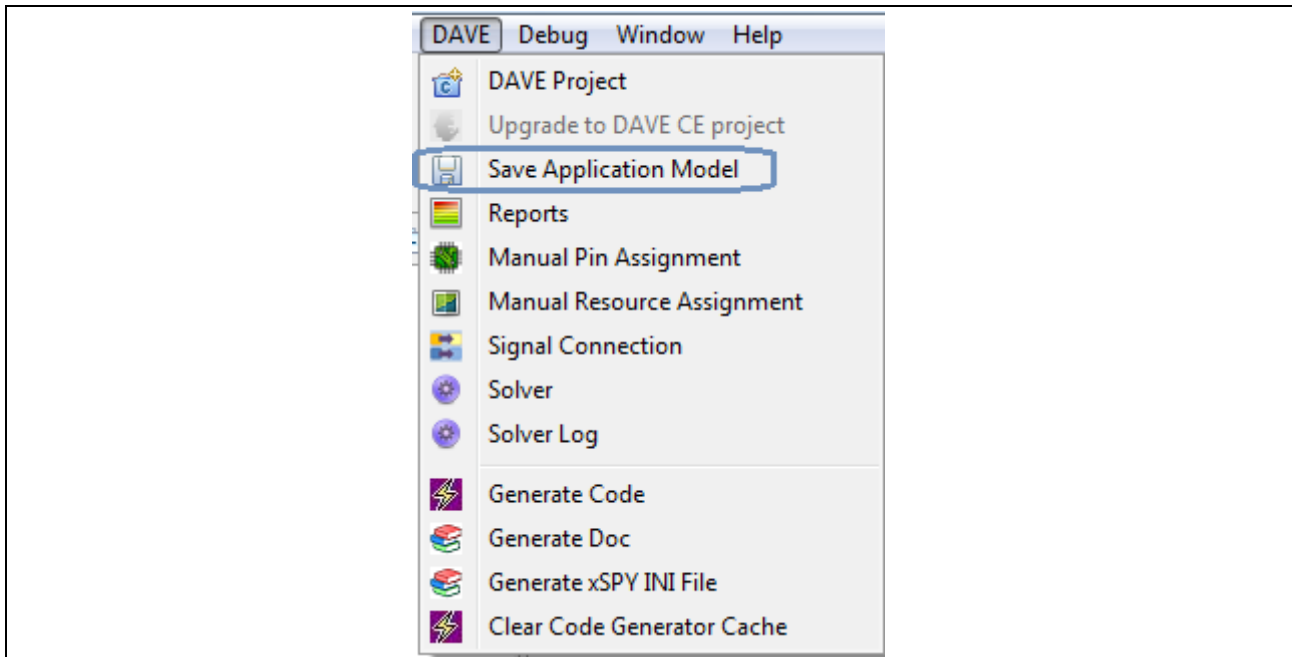


Figure 110 DAVE™ Menu – DAVE/Save Application Model Data

## 6.5 Manual Pin Assignment

This option is for Pin Configuration of selected or set of DAVE™ Apps in the project.

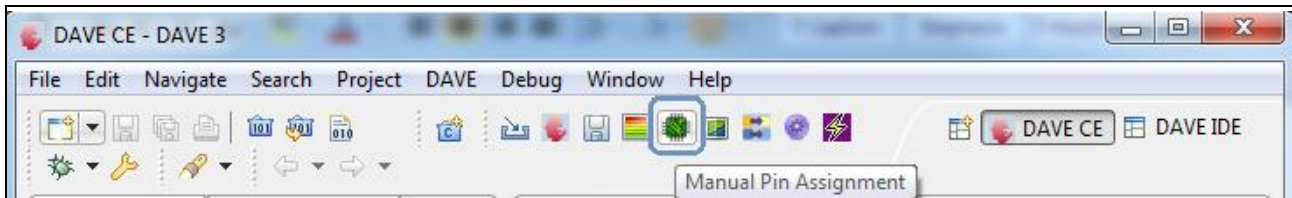


Figure 111 DAVE™ Menu – Manual Pin Assignment

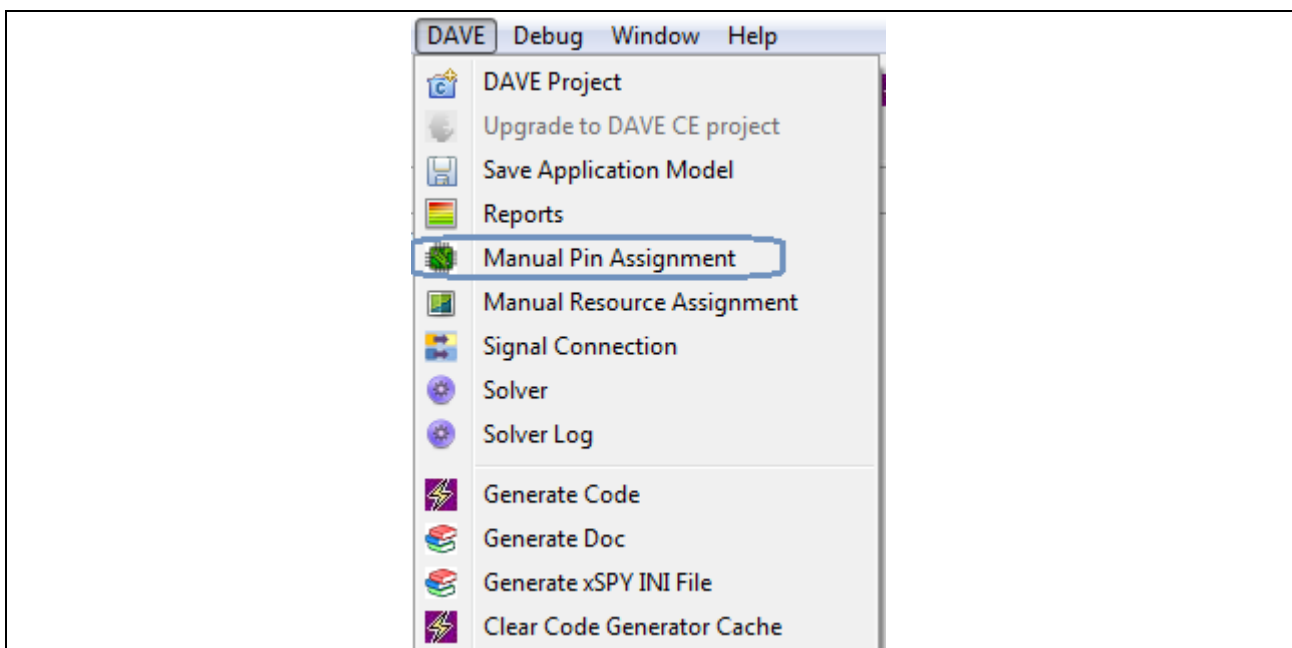


Figure 112 DAVE™ Menu – DAVE™ Manual Pin Assignment

## 6.6 Signal Connection

This option is for manual signal connectivity between DAVE™ App/s

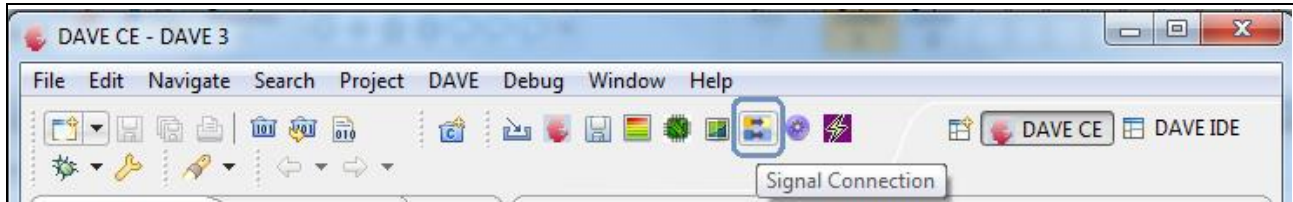


Figure 113 DAVE™ Menu – Signal Connection

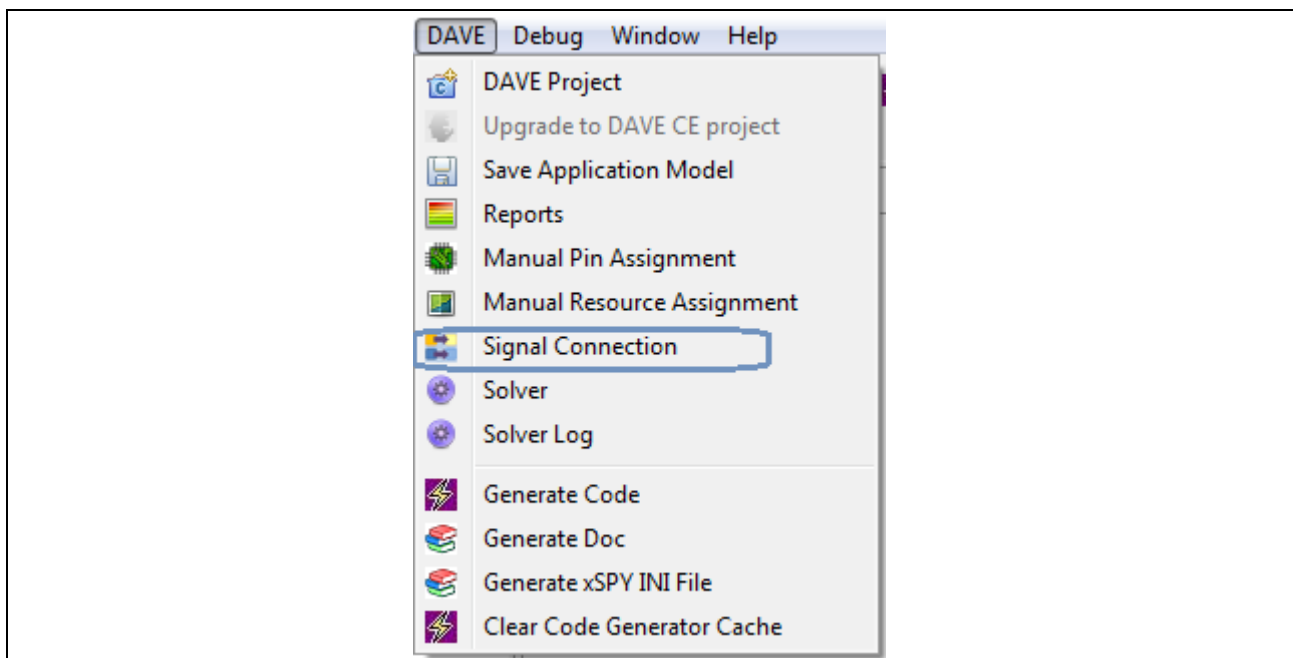


Figure 114 DAVE™ Menu – DAVE-Signal Connection



## 6.7 Solver

This option is for solve the connectivity and resource mapping of DAVE™ App/s in the project

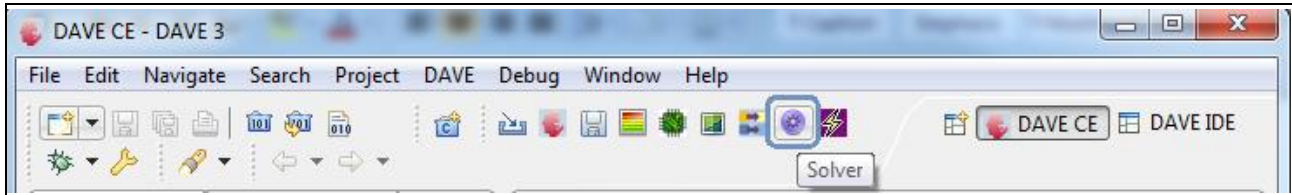


Figure 115 DAVE™ Menu – Solver

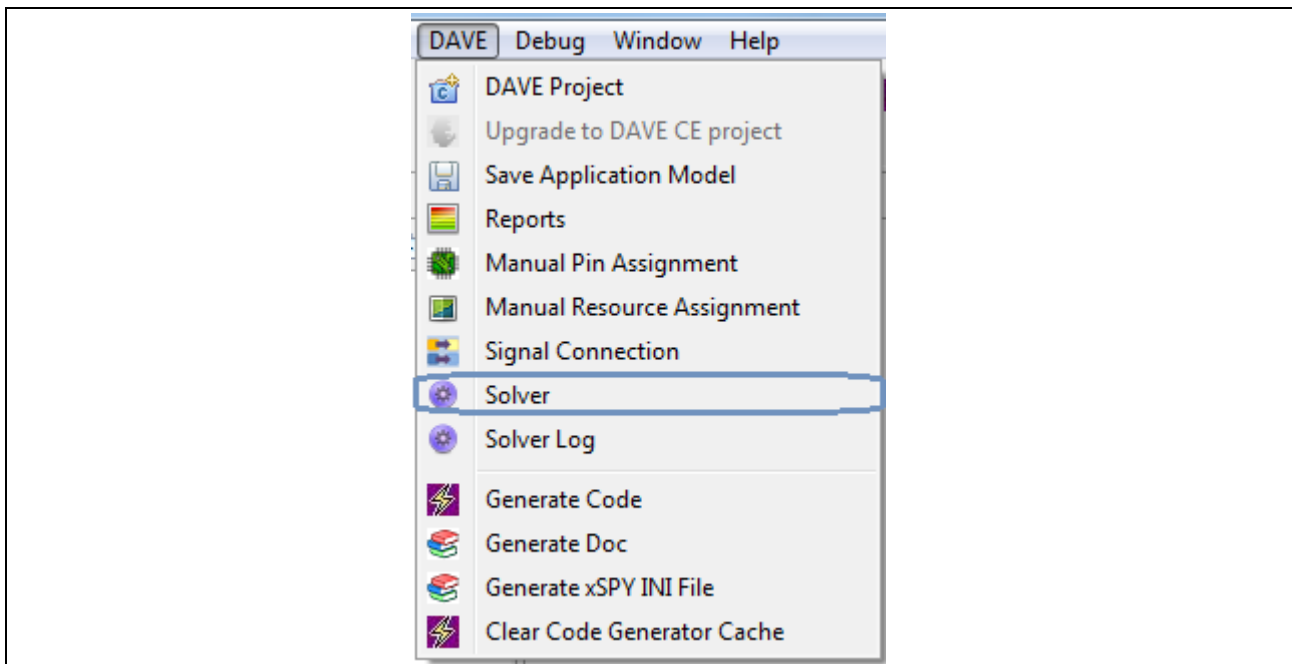


Figure 117 DAVE™ Menu – DAVE-Solver

## 6.8 Reports

Resource Mapping Information will be available in report

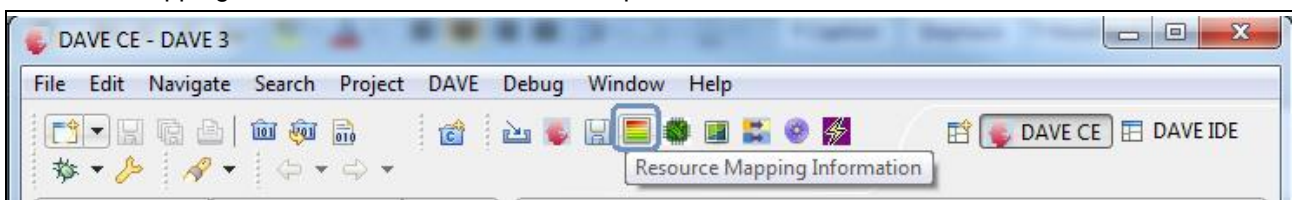


Figure 118 DAVE™ Menu – Resource Binding Information



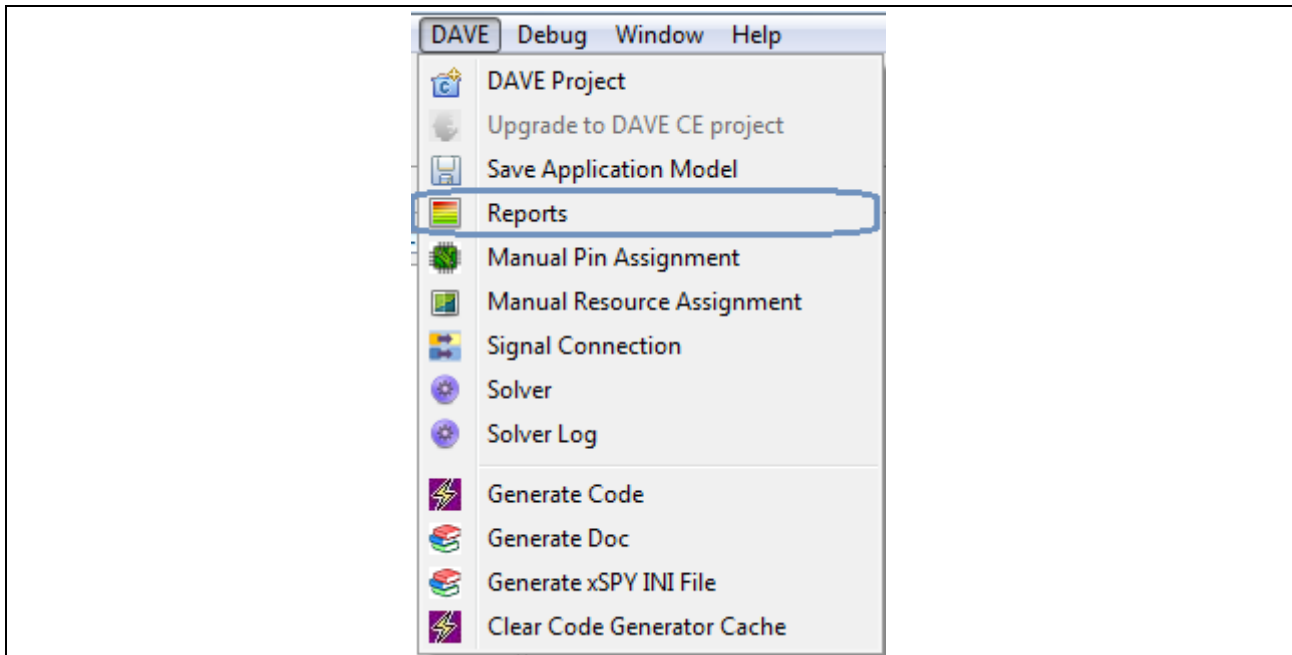


Figure 119 DAVE™ Menu – DAVE™ Reports

## 6.9 Generate Code

This option generates the code for the selected App/s in the project

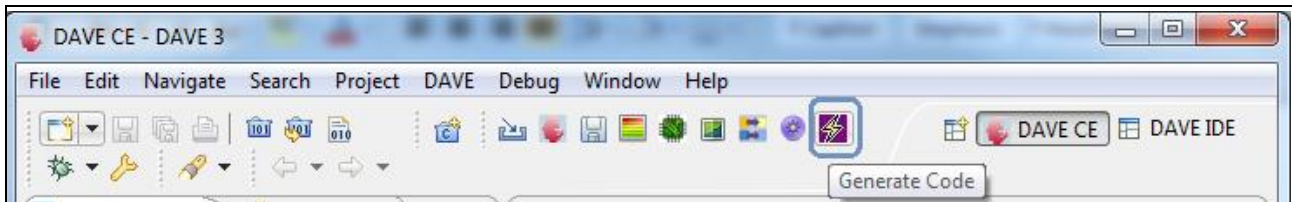


Figure 120 DAVE™ Menu – Generate Code

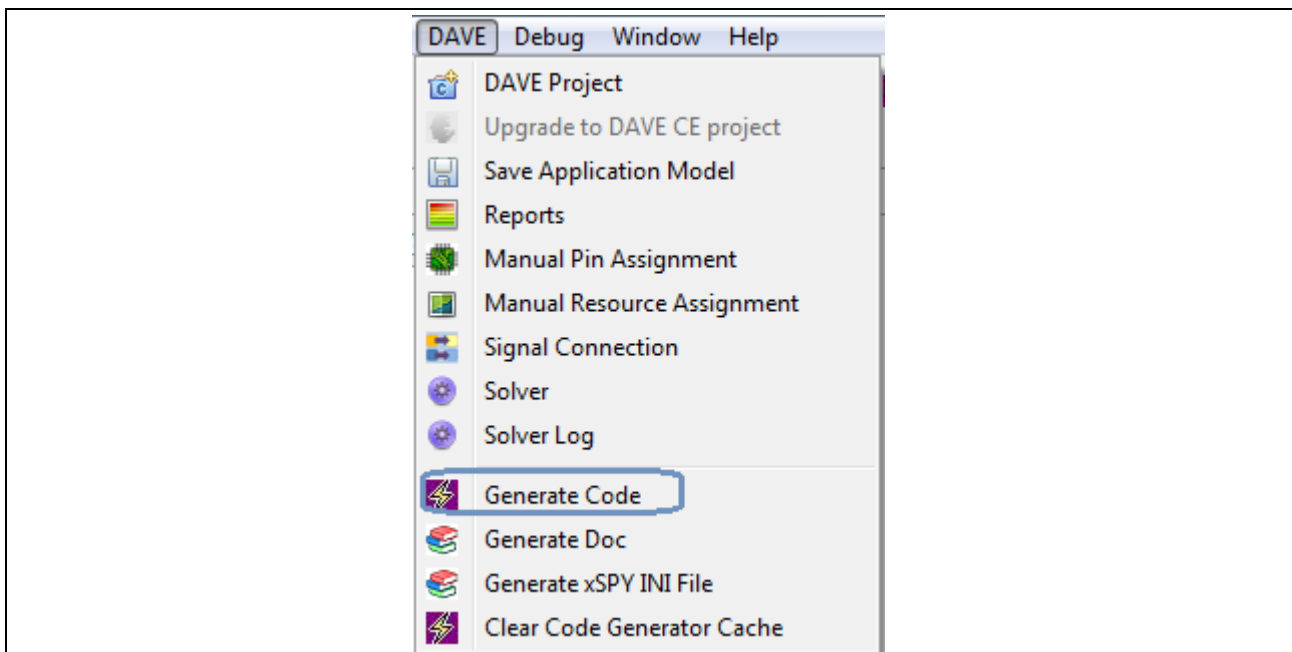


Figure 121 DAVE™ Menu – DAVE™ Generate Code

## 6.10 Generate Doc

This option generates the doxygen formatted documentation for selected DAVE™ App/s in the project

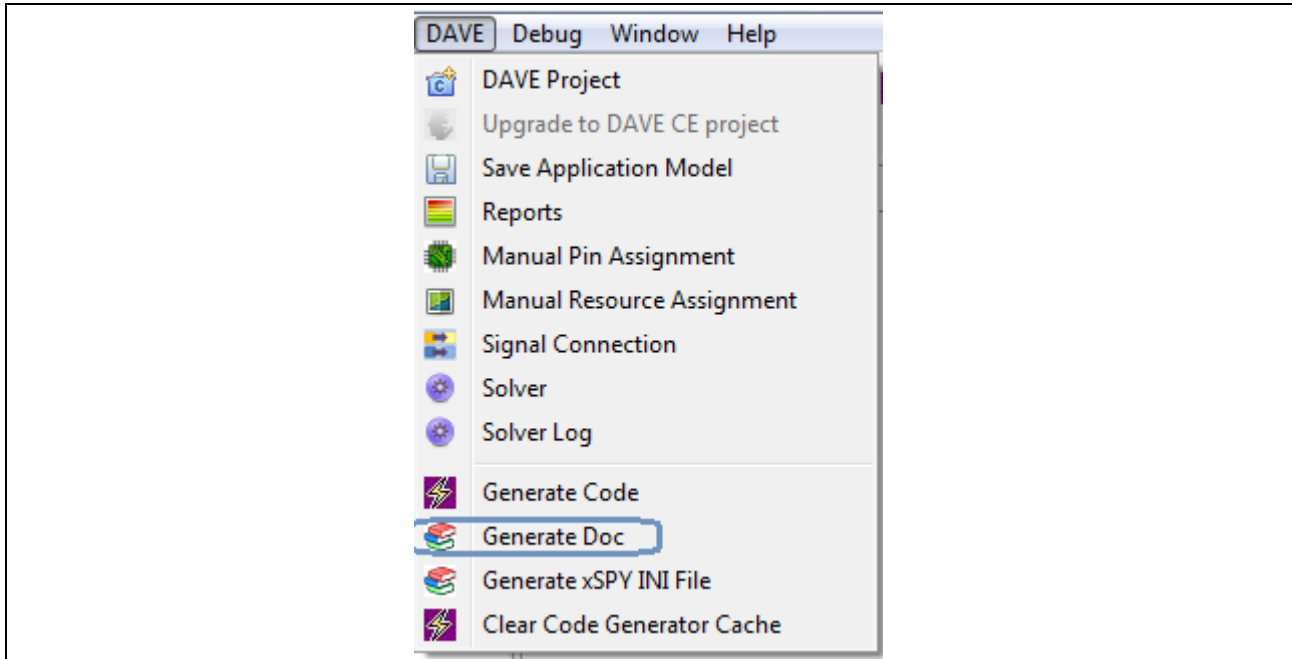


Figure 122 DAVE™ Menu – DAVE™ Generate Doc

## 6.11 View Generate Doc

This option provided facility to view generated doxygen documentation in Html format

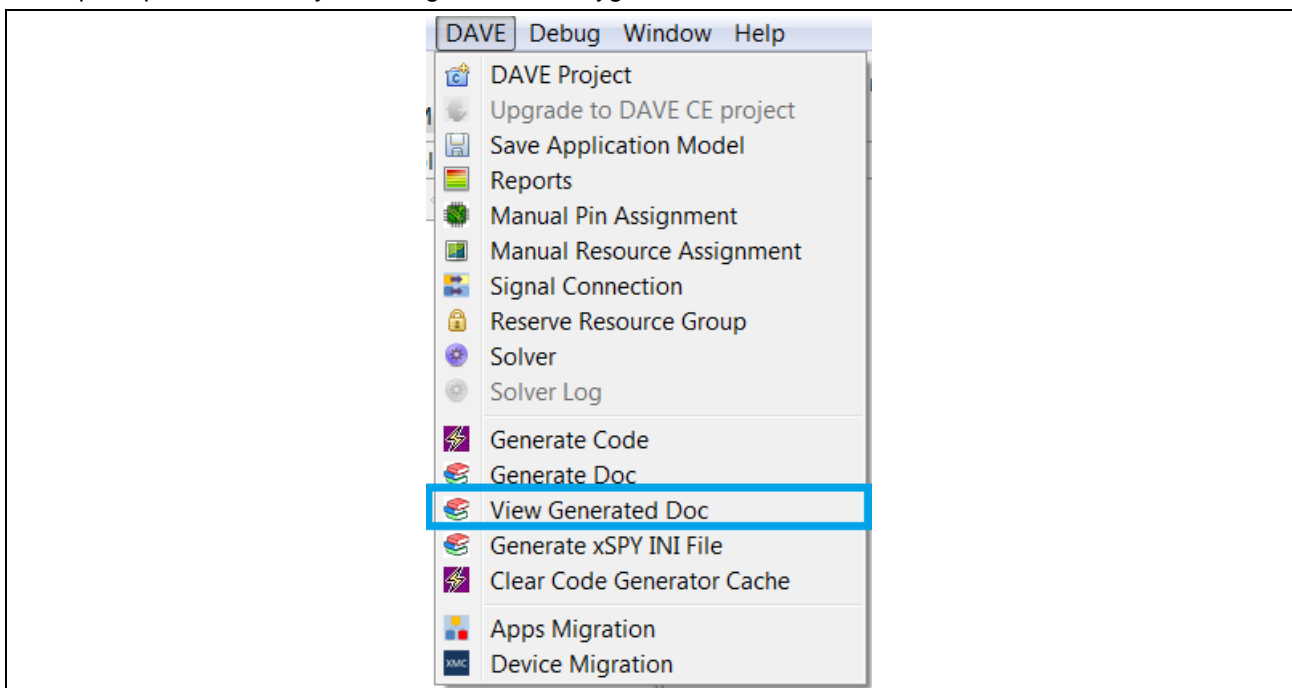


Figure 123 DAVE™ Menu – DAVE™ Generate Doc

## 6.12 Manual Resource Assignment

This option provides facility for Manual Resource Assignment of DAVE™ AppConsumed Resource/s

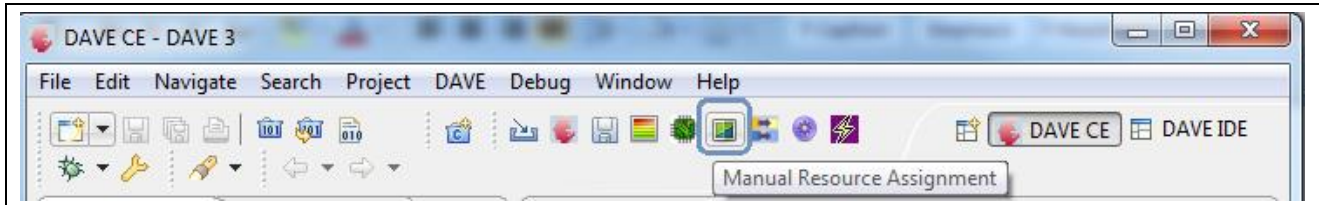


Figure 124 DAVE™ Menu – Manual Resource Assignment

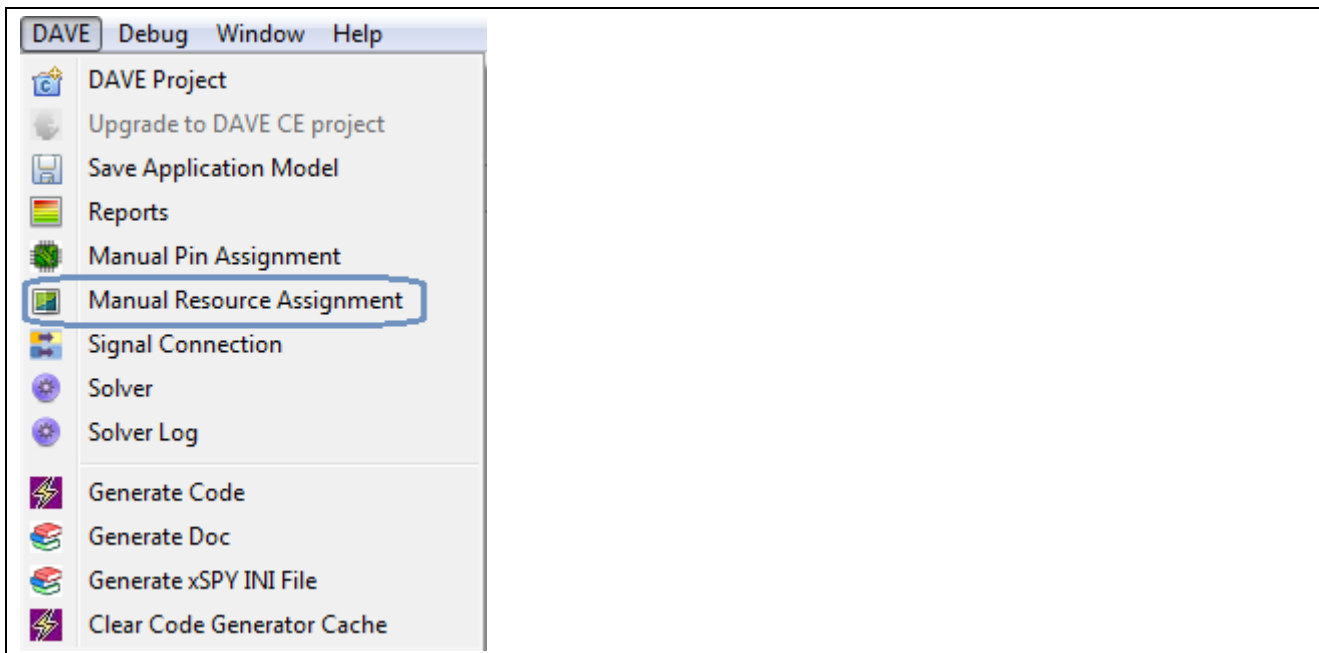


Figure 125 DAVE™ Menu – Manual Resource Assignment

### 6.13 Solver Log

This option generates the solver query log of the selected project

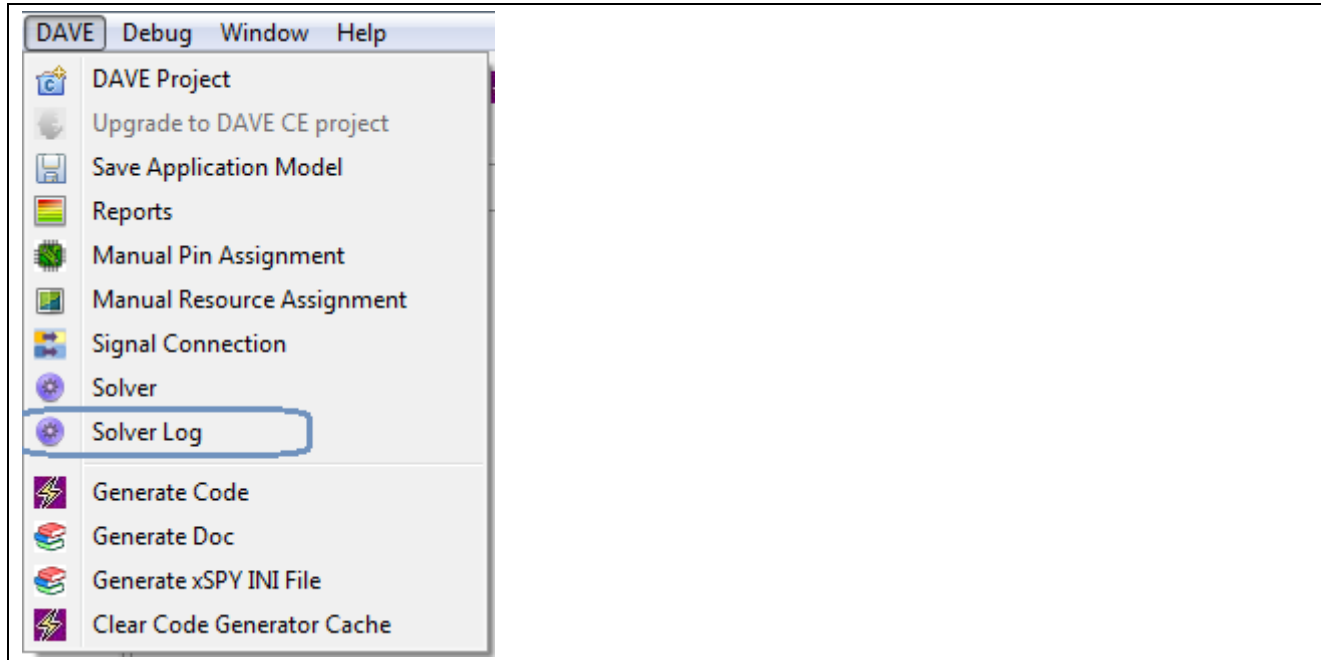


Figure 126 DAVE™ Menu – Manual Resource Assignment

### 6.14 Generate xSPY INI File

This option generates the xSPY INI file. This file is used by xSPY Plug-in for display of dynamic error/pbc messages sent by target.

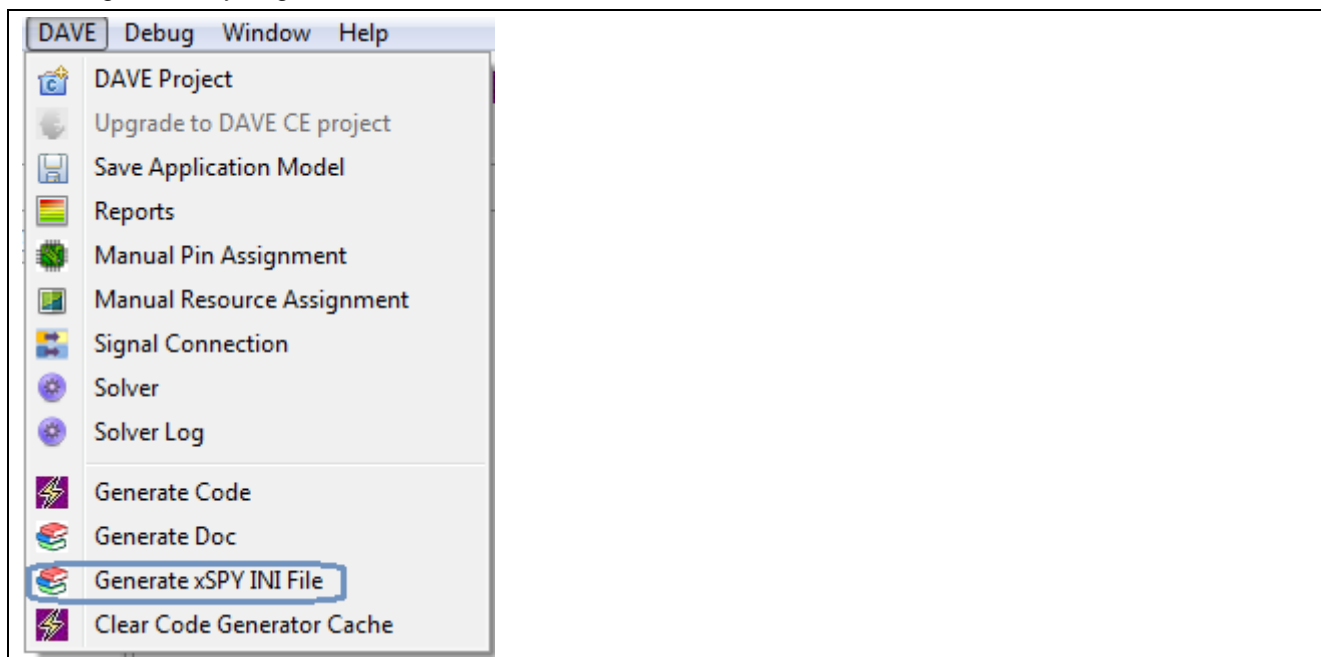


Figure 127 DAVE™ Menu – Generate xSPY INI File

## 6.15 Clear Code Generator Cache

Code Generator Engine stores the intermediate files (cache) for faster code re-generation. This option is to clear this cached information. In the subsequent code-generation cache shall be re-created.

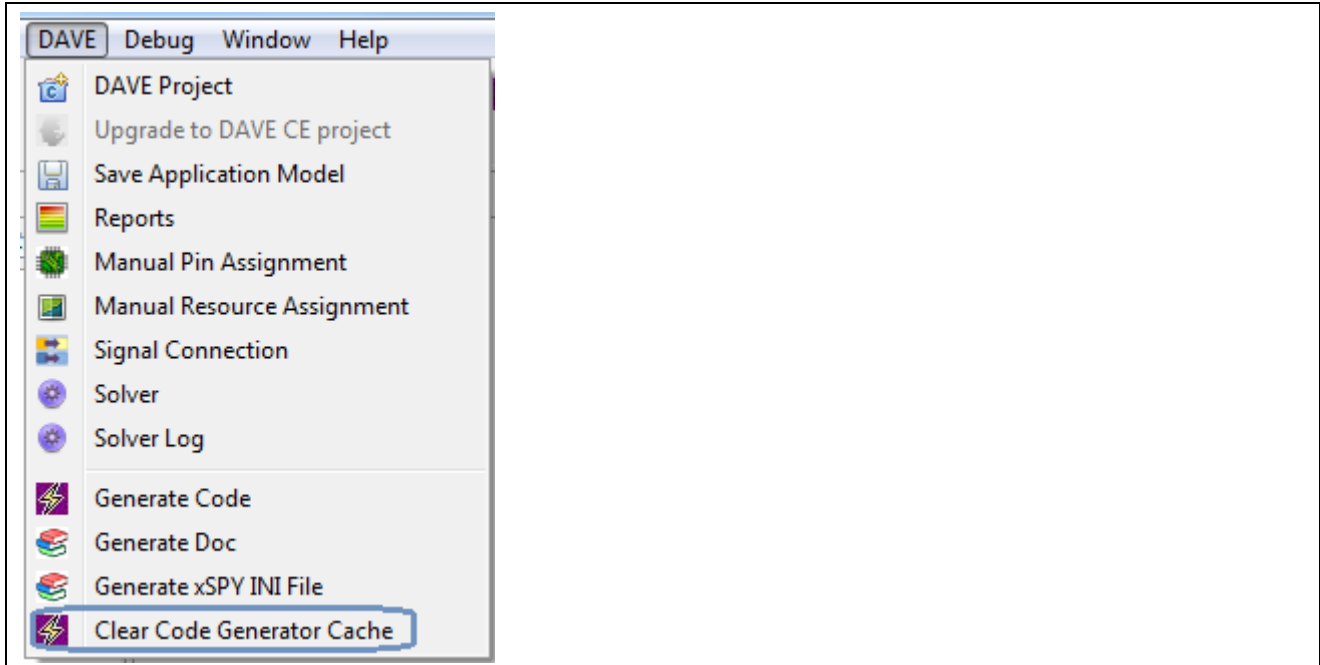
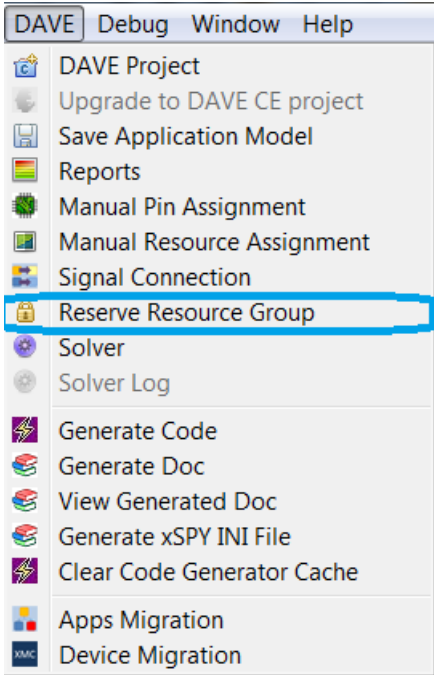


Figure 128 DAVE™ Menu – Clear Code Generator Cache

## 6.16 Reserve Resource Group

This feature allows reserving of chip resources from usage by the resource solver. Reserved chip resources will not be used by the resource solver to assign them to any DAVE™ App. This feature makes it easy to combine existing SW or SW from third party vendors with DAVE™ Apps. The chip resources used by these SW modules will be reserved and hence a resource conflict with DAVE™ Apps will not appear.

In the current implementation the chip resources will be reserved as resource groups similar as they are modeled in the Device Description.



The UI to reserve HW resource groups can be opened from the DAVE menu.

A frequently use case could be to reserve the System Timer because another third party RTOS is used in DAVE. The respective resource group would be:

```
/cpu/0/systick
```

The user has to type the resource group starting with a slash in the first free field. A completion function can be activated with:

-> CTRL + SPACE

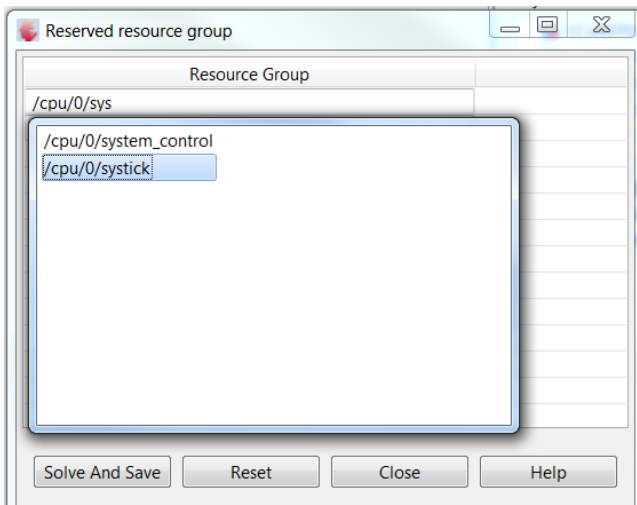
This shows all matching resource group options.

In this case /cpu/0/systick has to be selected.

By pressing "Solve and Save" all defined resource groups will be reserved.

The "Help" button provides further details.

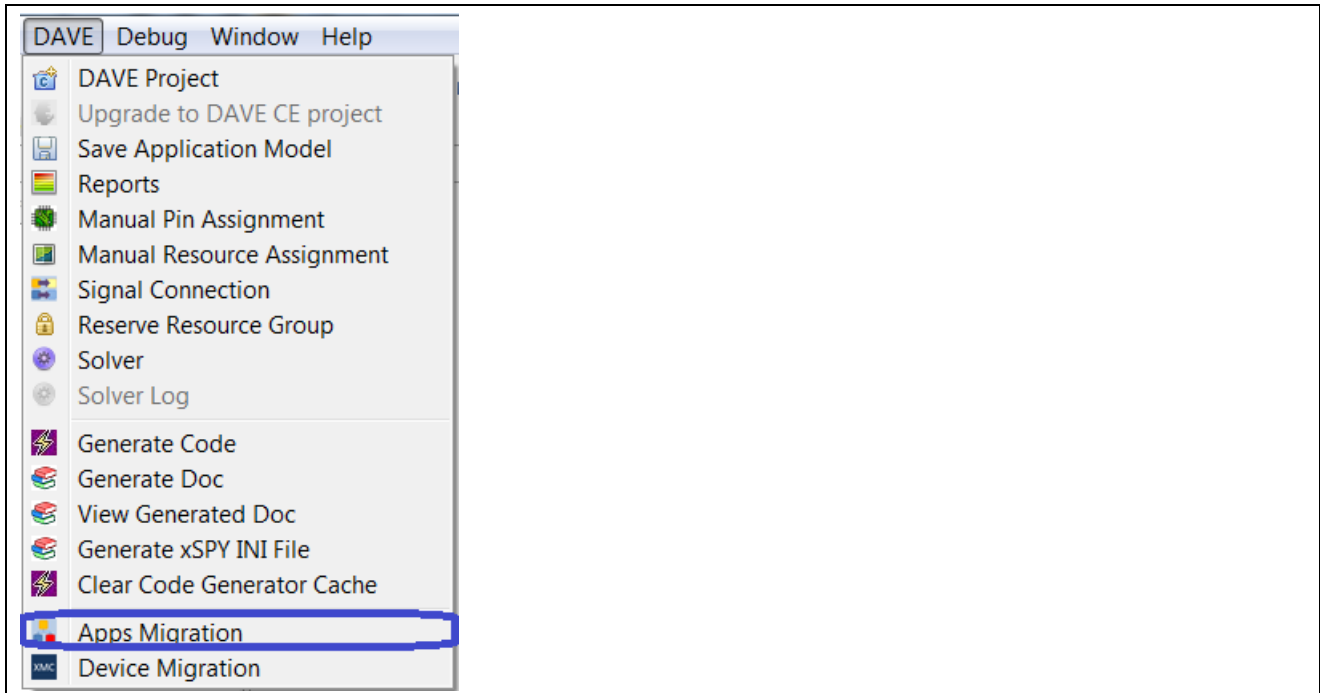
Resource group reservation based on register names will be added soon.



**Figure 129 DAVE™ Menu – Reserve Resource Group**

## 6.17 Apps Migration

This feature makes it easy to migrate an existing project with older DAVE™ App versions to newer DAVE™ App versions. If the latest DAVE™ App version in the local library store is newer than the same DAVE™ App of the active project, then the respective DAVE™ Apps of the active project will be replaced by the latest version stored in the local library store. All relevant setting from the old DAVE™ App version will be copied to the new version.

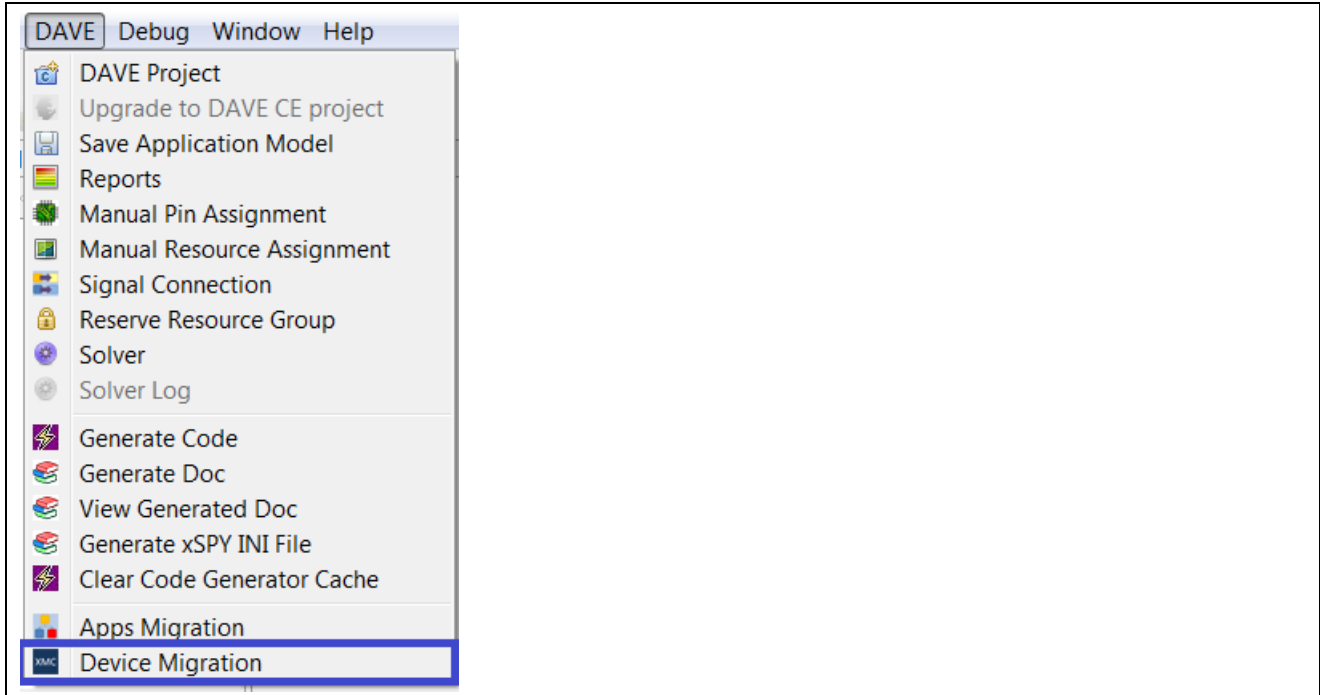


**Figure 130 DAVE™ Menu – Apps Migration**

Further details are explained in chapter 12

## 6.18 Device Migration

This feature allows changing the target MCU device of the active project to another target microcontroller within the same XMC series.



**Figure 131 DAVE™ Menu – Device Migration**

Further details are explained in chapter 12



## 7 Configurations

### 7.1 UI Editor – Selection

- The UI Editor will open by double click on the App in App Dependency Graph View **or**
- Right click on App and select UI Editor

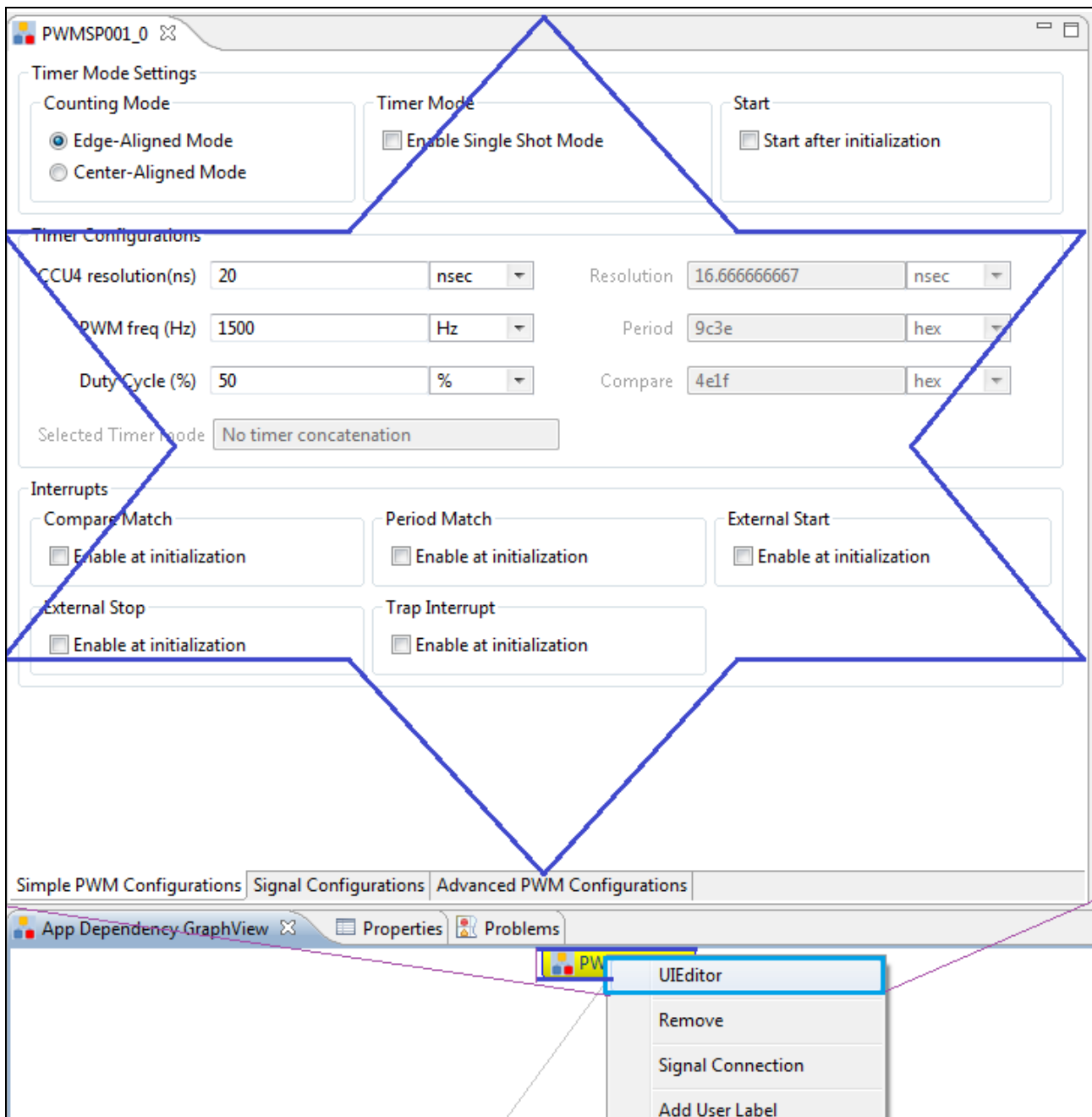


Figure 132 DAVE™ UI Editor – App dependency Graph View

### 7.1.1 Dialogue/page display

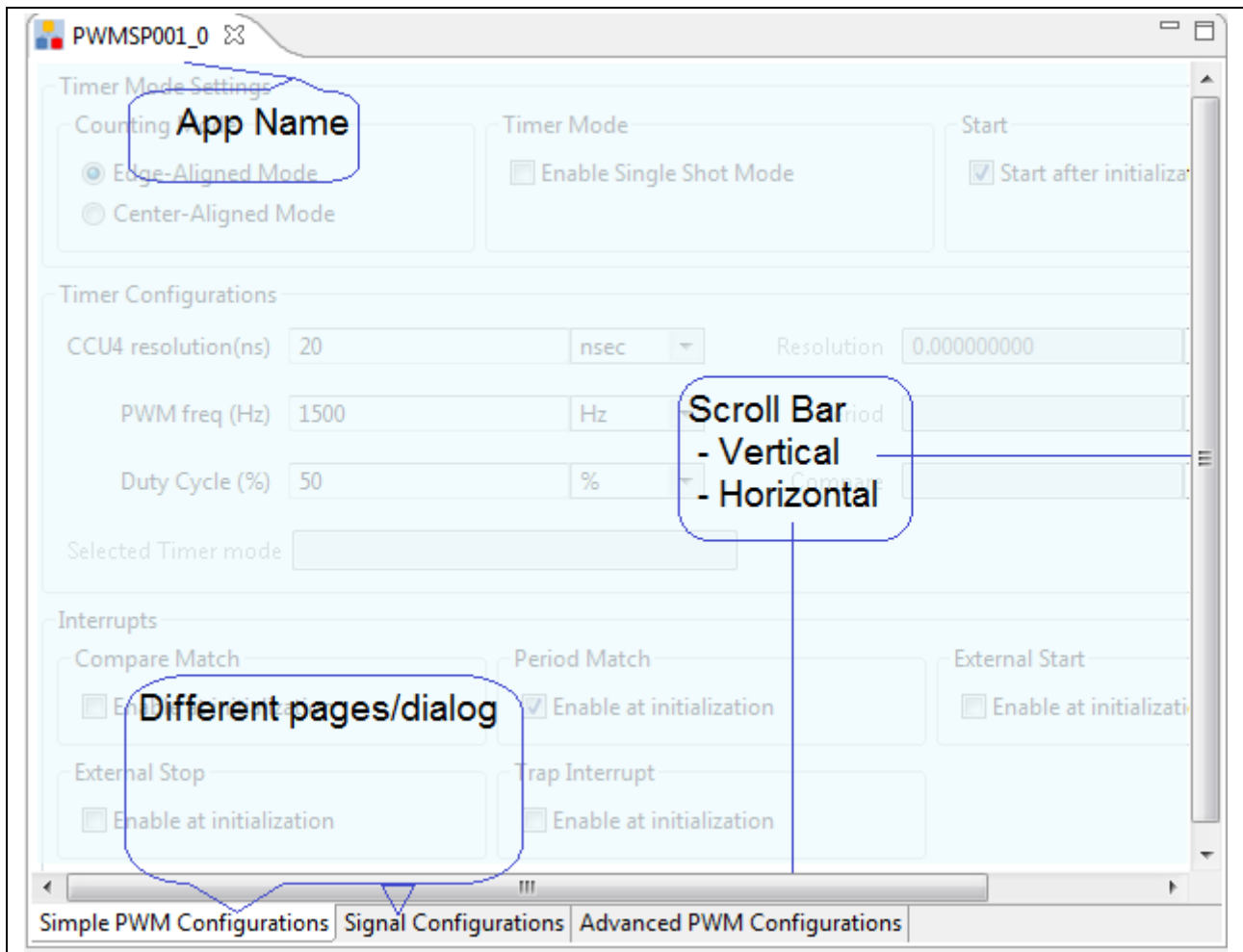


Figure 133 DAVE™ UI – page

### 7.1.2 UI Controls

#### 7.1.2.1 Integer and their Types



Figure 134 DAVE™ UI Controls – Integer

#### 7.1.2.2 String



Figure 135 DAVE™ UI Controls – String

### 7.1.2.3 Check Box

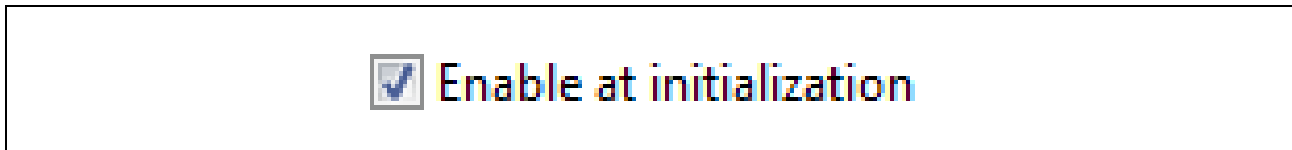


Figure 136 DAVE™ UI Controls – Check Box

### 7.1.2.4 Radio Box/Option Button

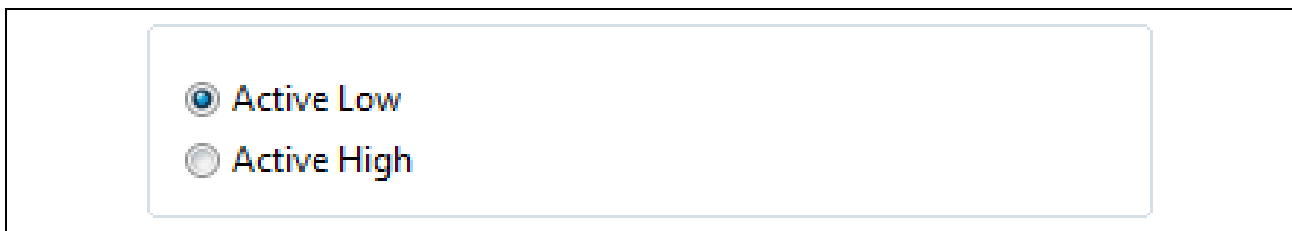


Figure 137 DAVE™ UI Controls – Radio/Option Button

### 7.1.2.5 Combo Box

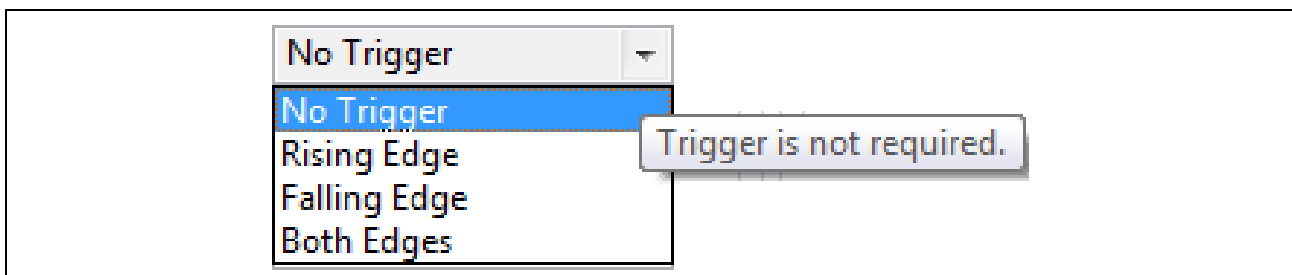


Figure 138 DAVE™ UI Controls – Combo Box

### 7.1.2.6 Enabled and Disabled

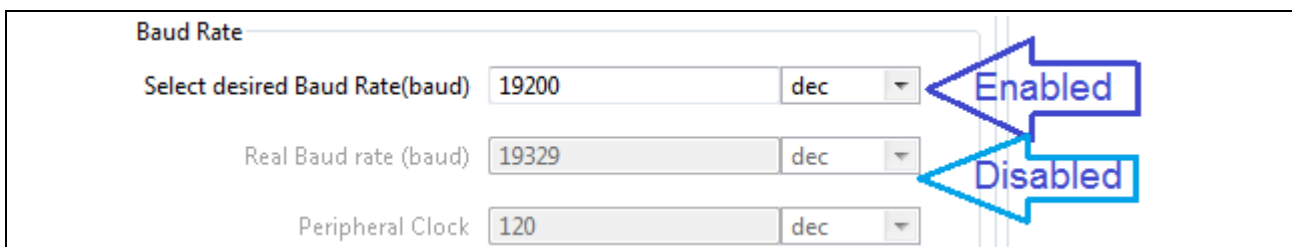


Figure 139 DAVE™ UI Controls – Enabled and Disabled

### 7.1.2.7 UI control Dependency

Some of the dependent UI controls, changes to editable when main controls are modified/changes.

Ex: PWMSP002 App signal configurations in UI editor, Start/ Stop configurations are readable/ disabled. But, these parameters can be editable/ selectable by configuring the Start/ Stop trigger configurations.

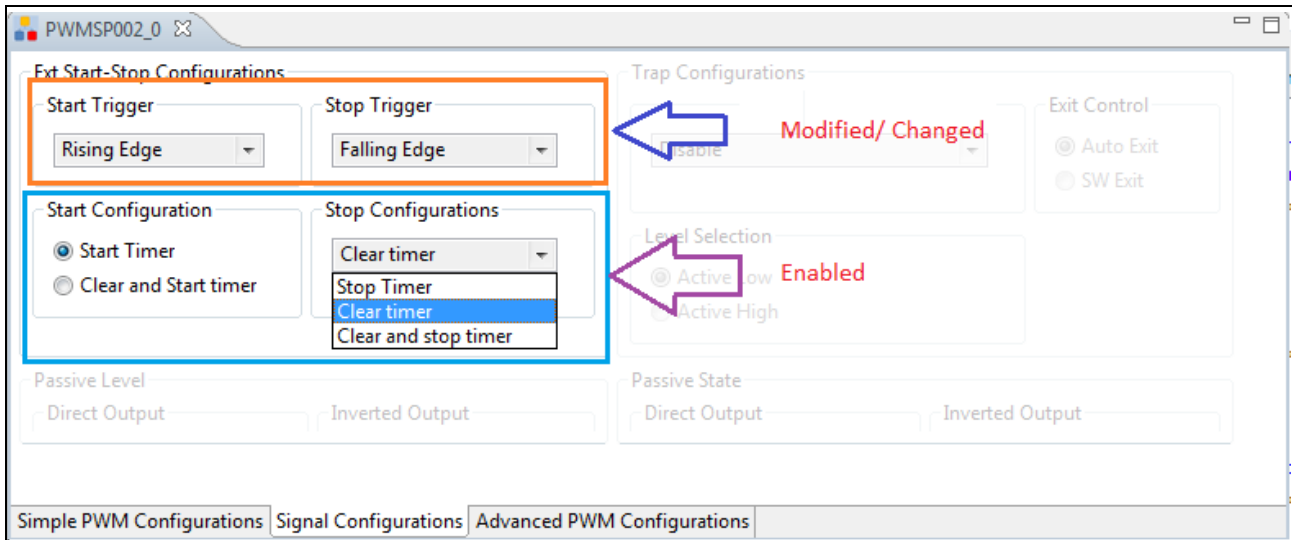


Figure 140 DAVE™ UI Controls – Editable - Disabled

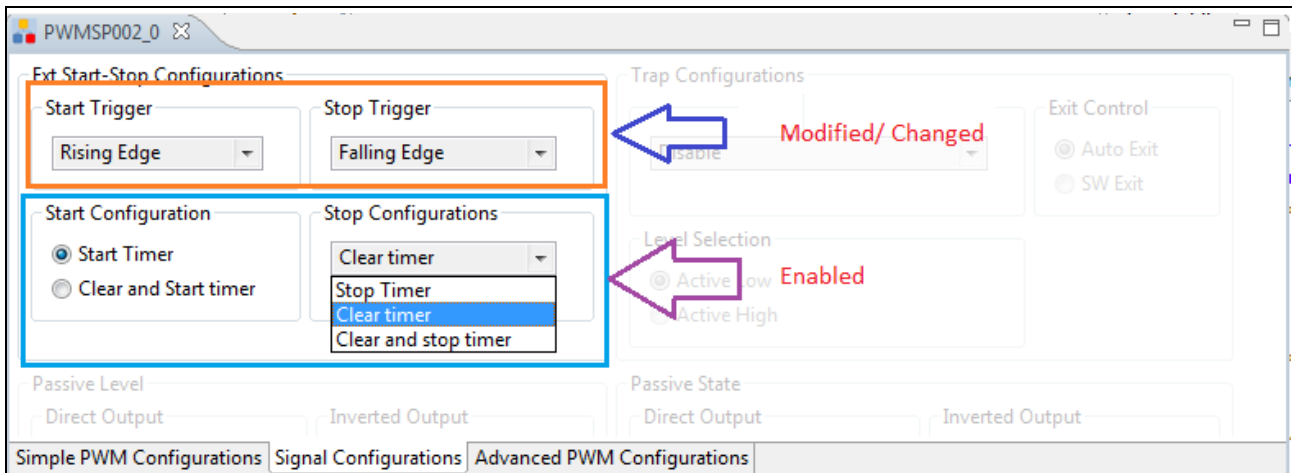


Figure 141 DAVE™ UI Controls Dependency

### 7.1.2.8 Tool Tip Support

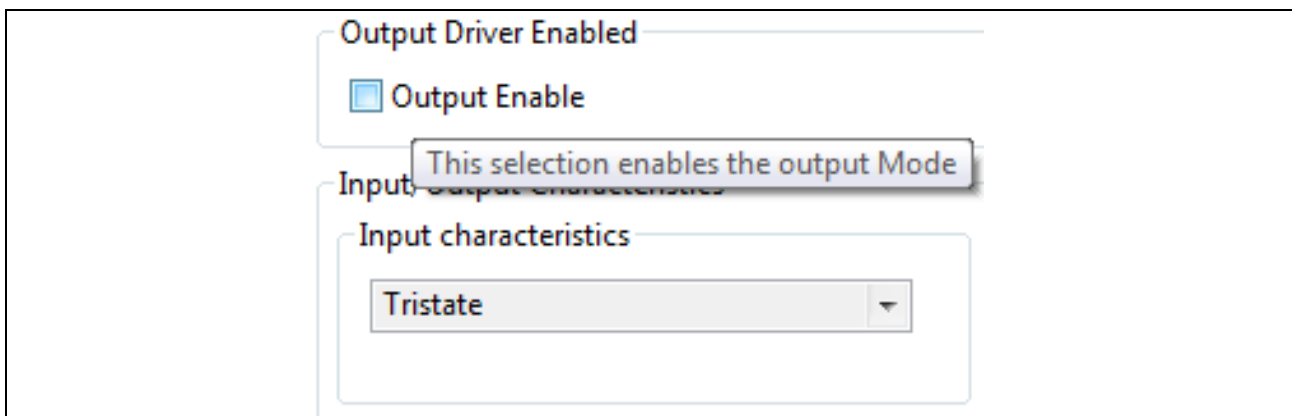


Figure 142 DAVE™ UI Controls - Tool Tip for Description of Functionality

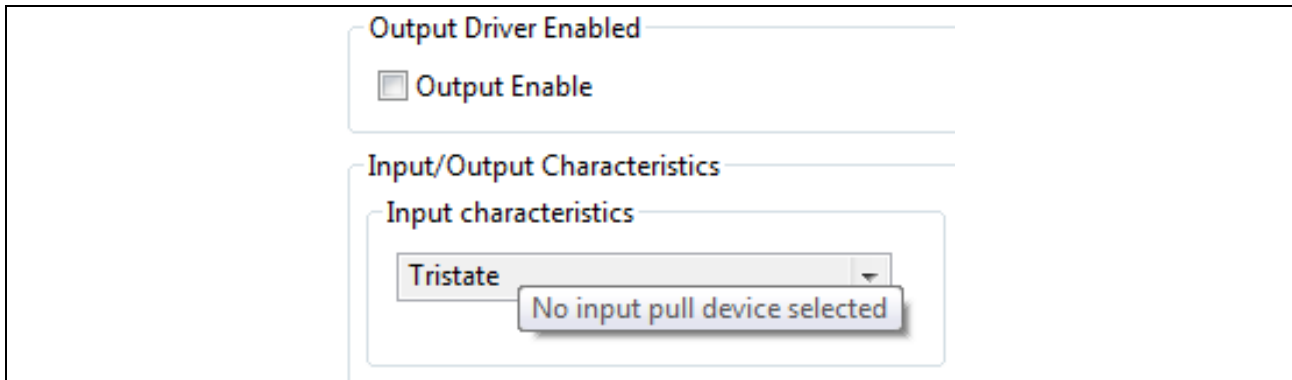


Figure 145 DAVE™ Tool Tip for Dependency description

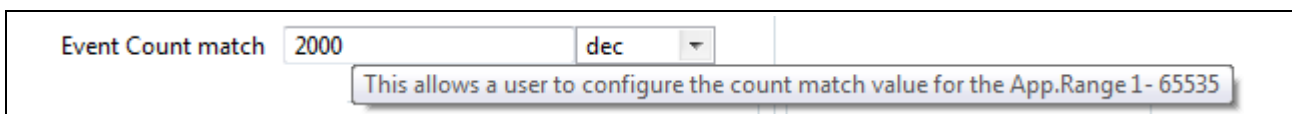


Figure 143 DAVE™ UI Controls – Tool Tip Describes Range

### 7.1.2.9 Range [Minimum & Max Value] supported

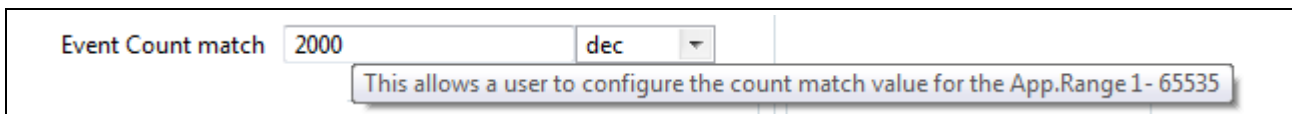


Figure 144 DAVE™ UI Controls – Range

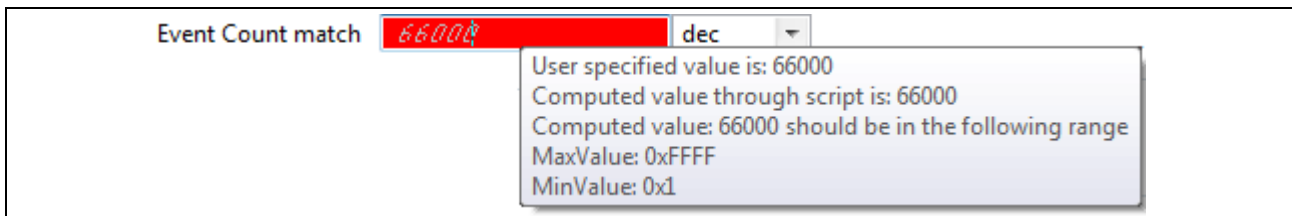


Figure 145 DAVE™ UI Controls – Max Value Entered

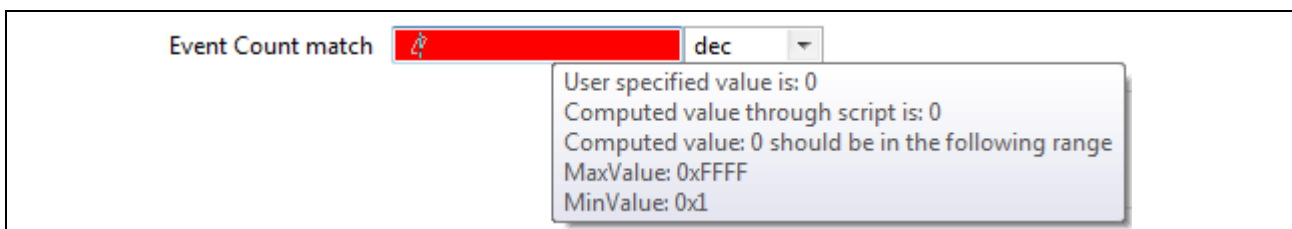


Figure 146 DAVE™ UI – Min Value Entered

### 7.1.2.10 Non-Configurable Apps

Some of the Apps in App Library are non configurable, means user configuration parameters are not present. These Apps will be providing the APIs as per the use case.

Ex: AES001

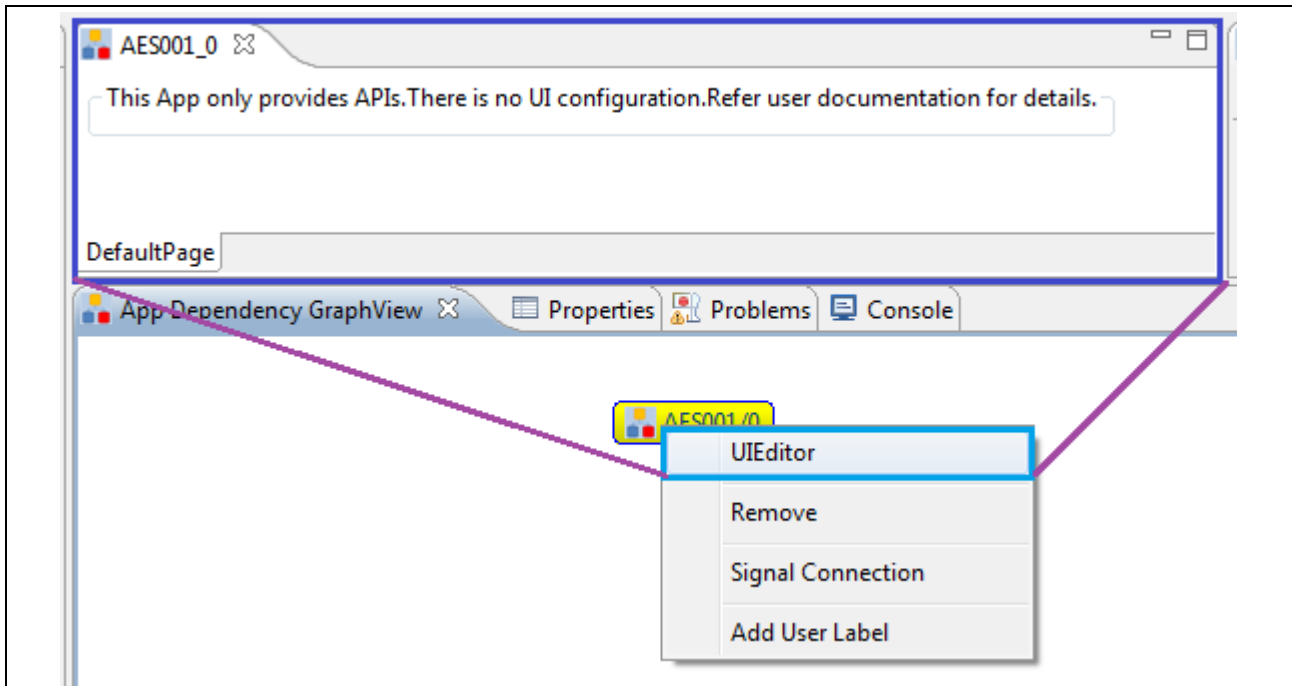


Figure 147 DAVE™ UI Controls – Non configurable Apps

## 7.2 DAVE™ Settings

### 7.2.1 Preferences Settings

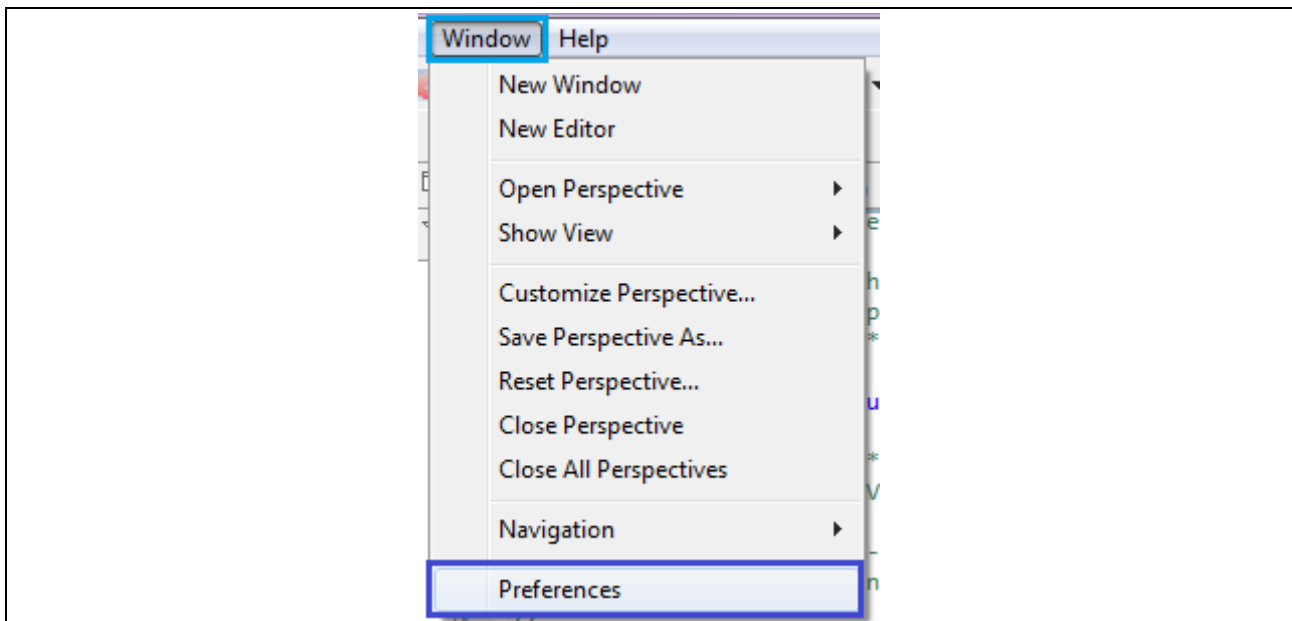
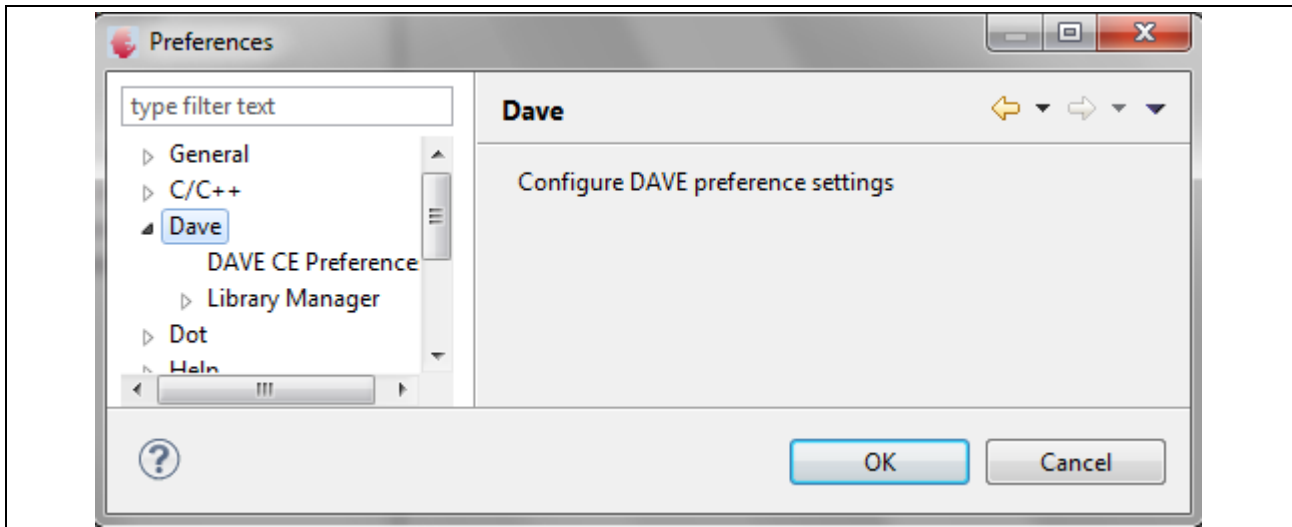
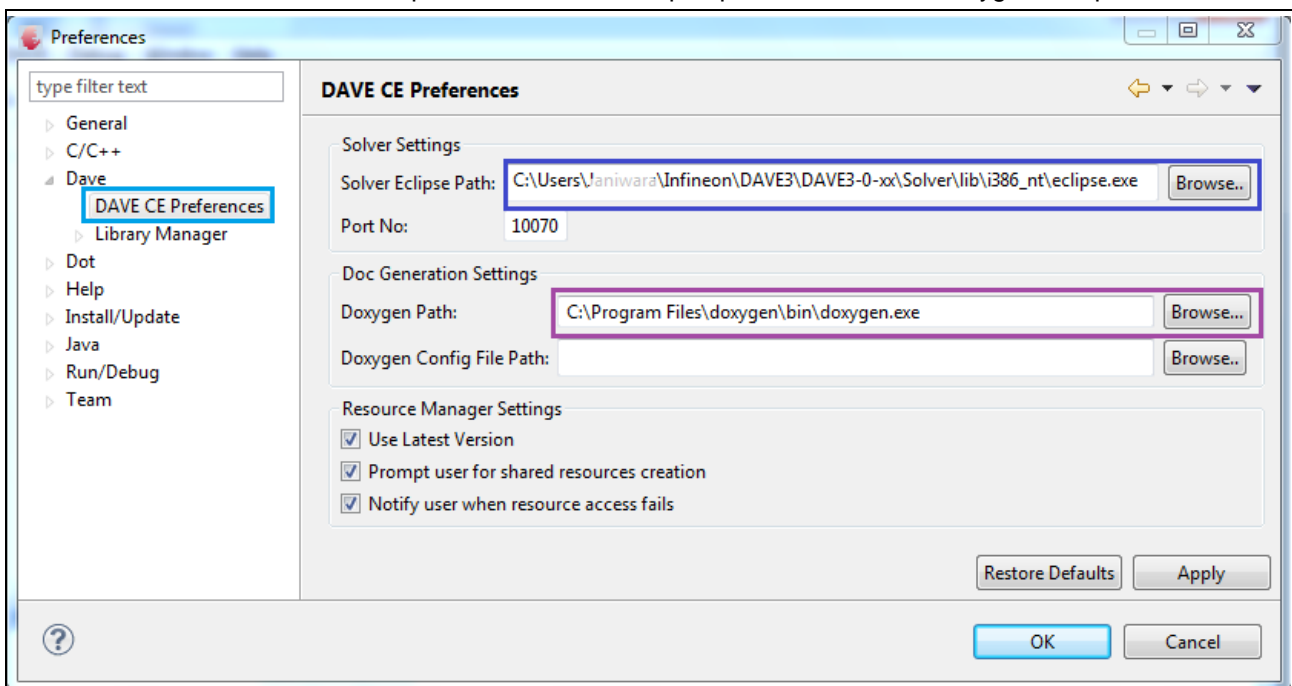


Figure 148 DAVE™ Integrated Features - Preference



**Figure 149 DAVE™ Integrated Features – Dave preferences**

→ DAVE™ CE Preferences, provide the solver eclipse path and also the Doxygen exe path.



**Figure 150 DAVE™ Integrated Features – Preferences – DAVE™ CE Preference**

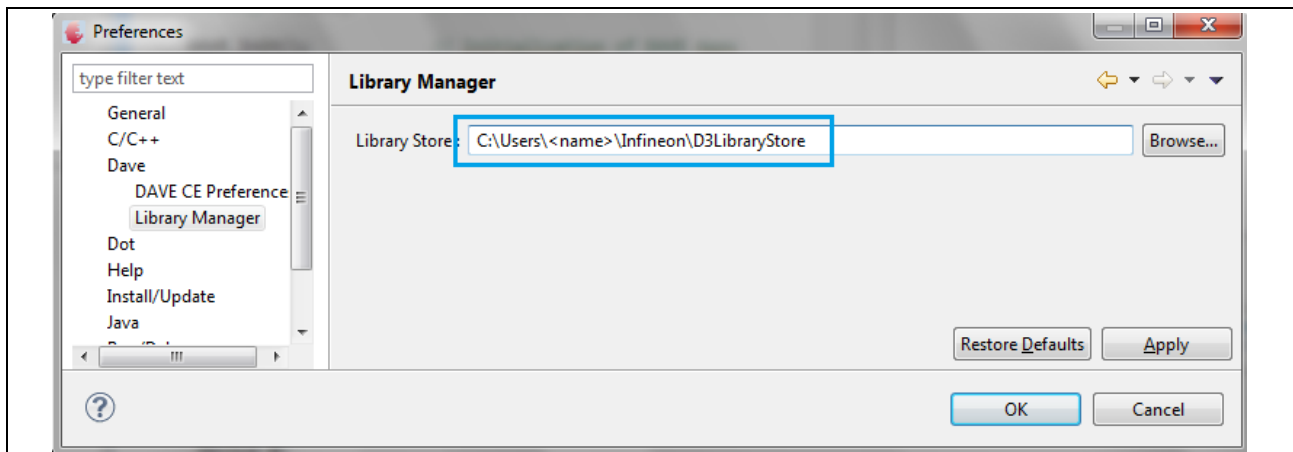
→ DAVE™ CE Preferences, Check the Library store path. The tool will be using the default path. User can change the library store path by using browse option.

Example:

For Windows 7 Operating System, the default path may be C:\Users\<User Name>\Infineon\LibraryStore

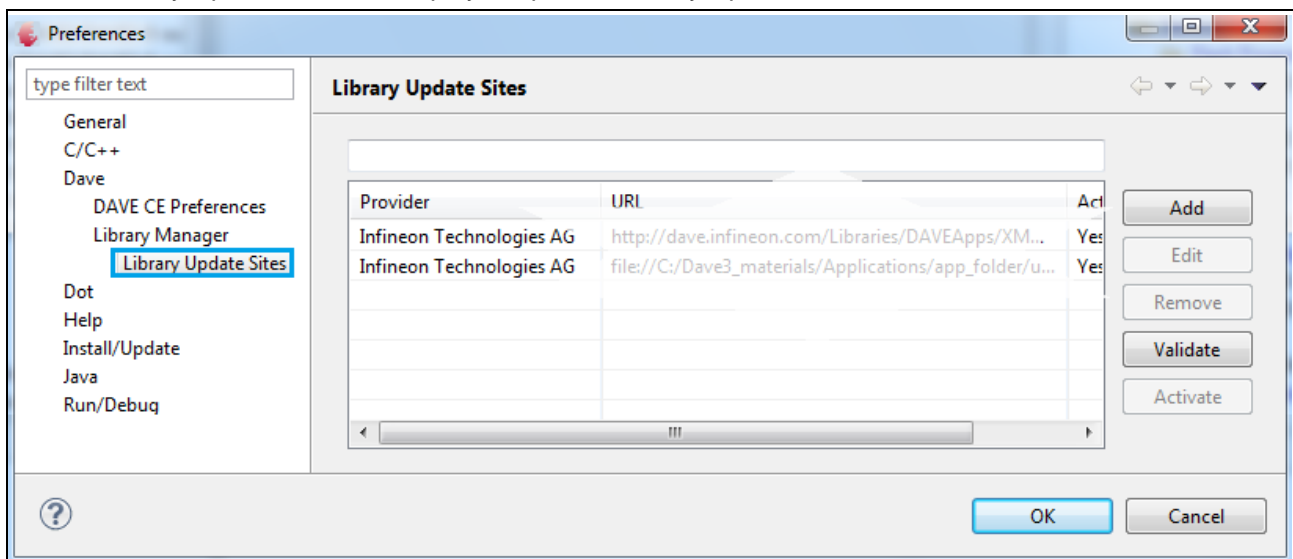
For Windows XP Operating System, the default path may be C:\Documents and Settings\<User Name>\Infineon\LibraryStore

→ After setting the path for library store, click on Apply and OK button to save the settings.



**Figure 151 DAVE™ Integrated Features – Preferences Library Manager – Default location**

→ Library Update Sites will display the path of Library Update Sites

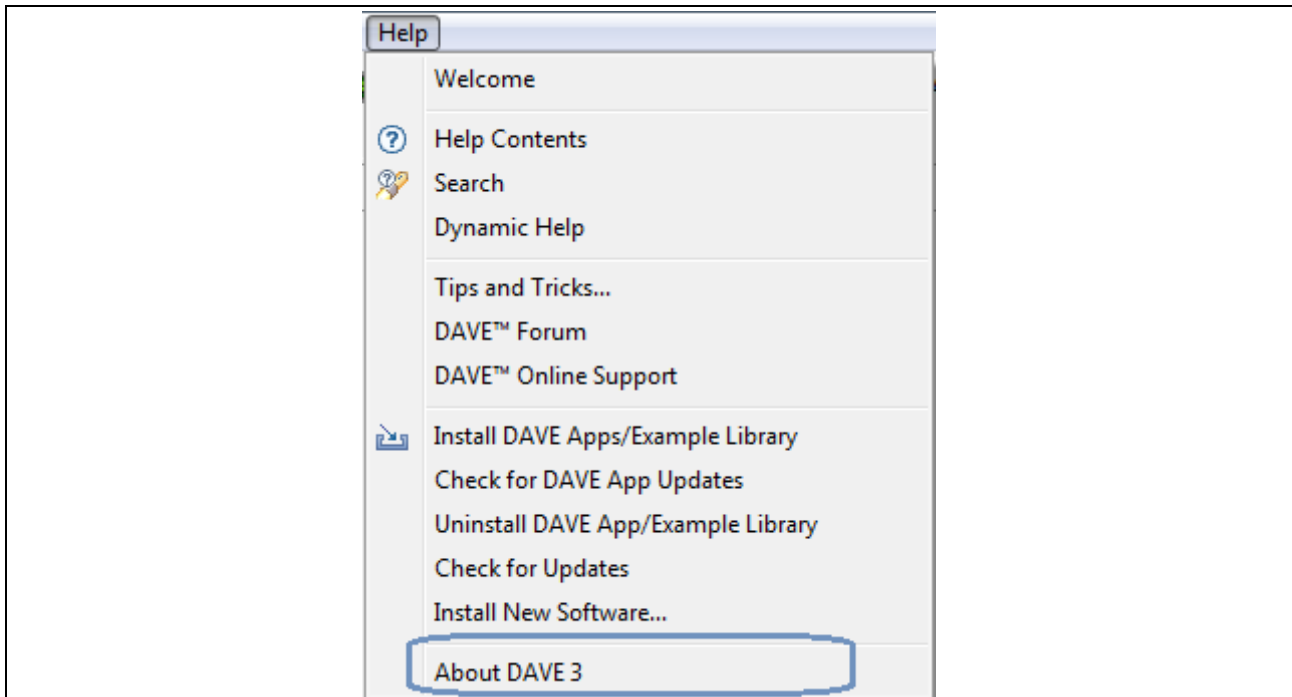


**Figure 152 DAVE™ Integrated Features – Preferences Library Manager – Update Sites**

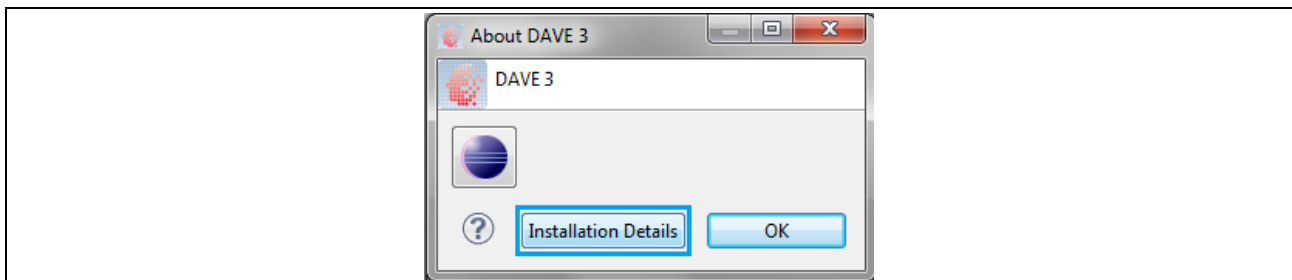
## 7.2.2 About DAVE

→ DAVE™ software information will be available by selecting Help -



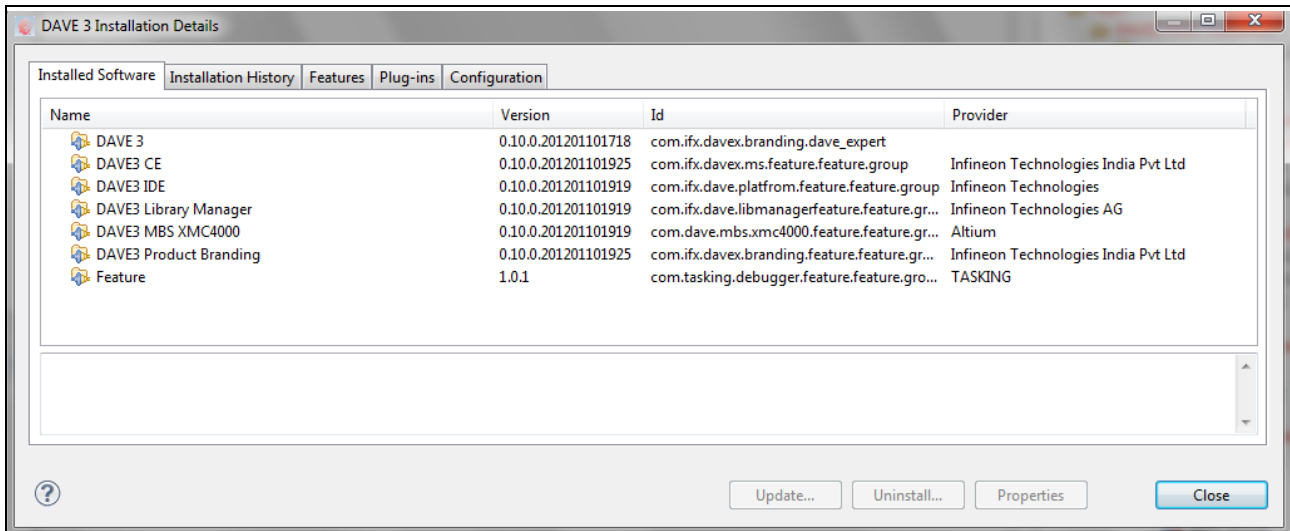


**Figure 153 DAVE™ About DAVE3**



**Figure 154 DAVE™ Installation Details**

- ➔ DAVE™ Installation Details provides following information's
- Installed Software
  - Installation History
  - Features
  - Plug-ins
  - Configuration



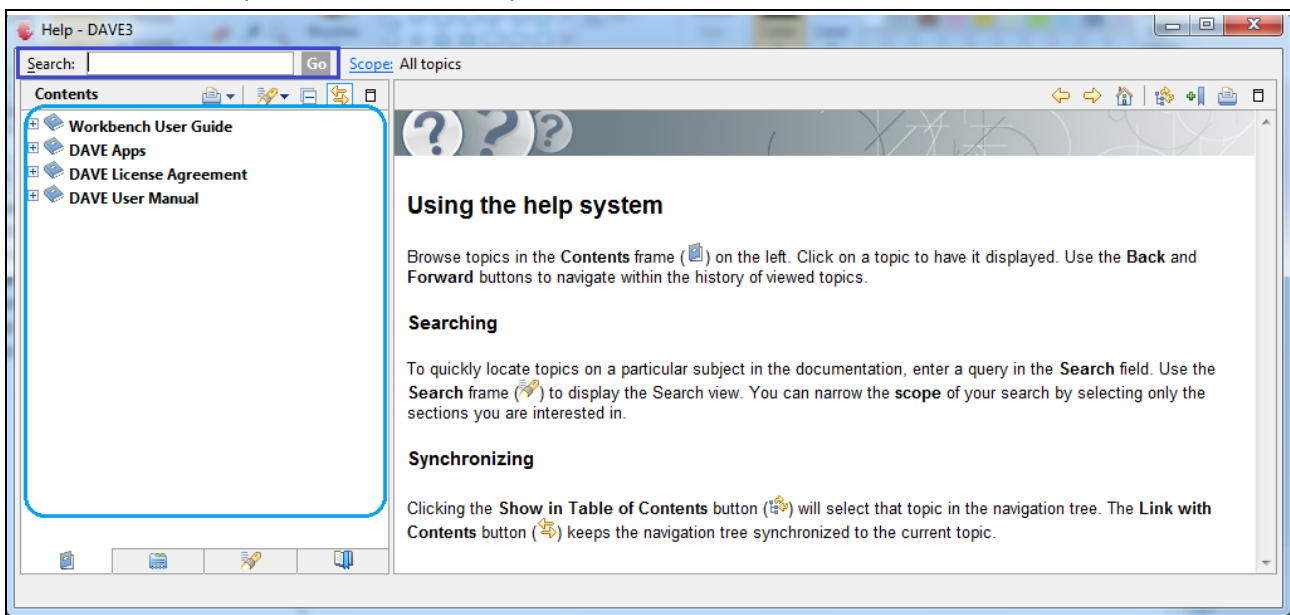
**Figure 155 DAVE™ Installed Software**

Note: The version shown in figure and the actual version installed may differ.

## 8 DAVE™ Help System

The DAVE™ help system will guide the user to understand the tool and its usage.

- Click on Help Menu and select Help contents



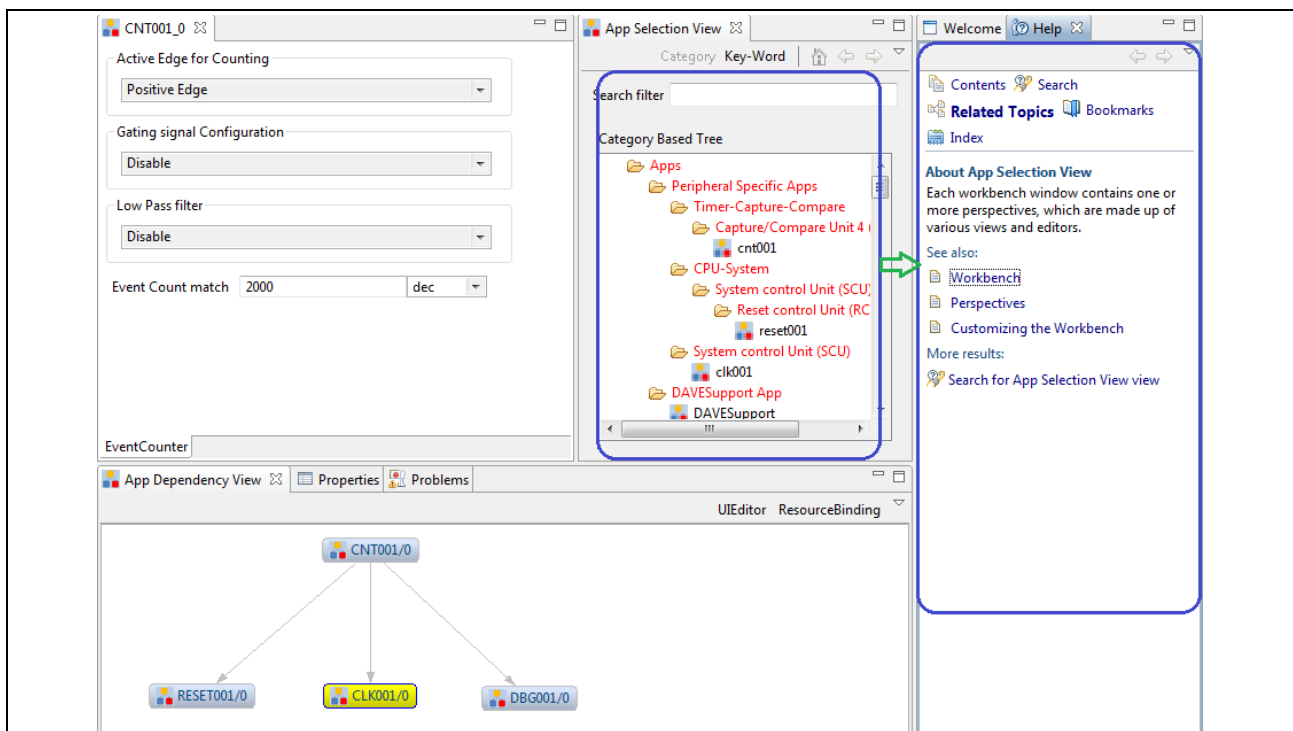
**Figure 156 DAVE™ Help**

The user can get the help on the topic which is required by entering it on the search editor

### 8.1 Dynamic Help

- Import App
- Create new DAVE™ CE Project
- Select app from the App Selection View
- Use F1 Key for App Help – information with respected to selected UI Items.

- ❖ If Mouse is selected on App Selection View and then F1 key is pressed it will show help about 'App Selection View'



**Figure 157 DAVE™ Help About 'App Selection View'**

- ❖ If Mouse is selected on UI and then press F1 it will show help about 'DAVE™ MS UI Multi Page Editor'

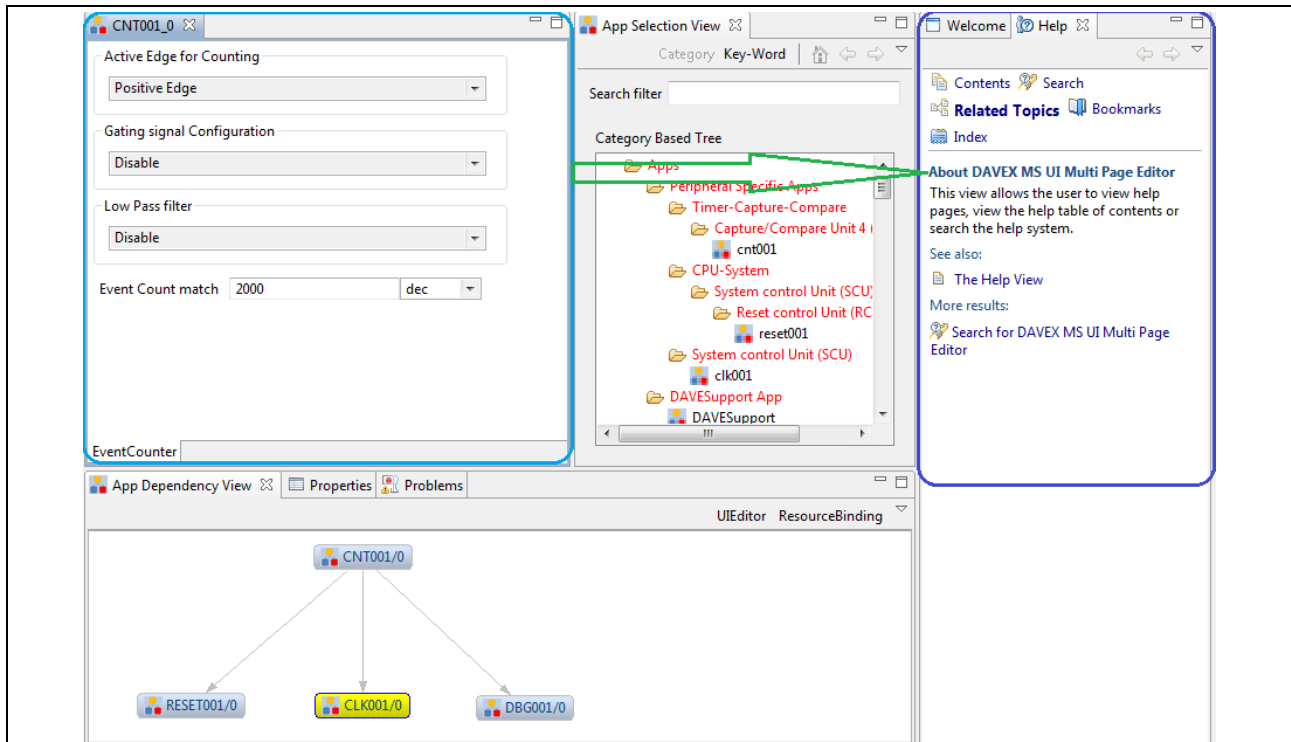


Figure 158 DAVE™ Help About DAVEX MS UI Multi Page Editor

- ❖ If Mouse is selected on UI and then press F1 it will show help about 'DAVE™ MS UI Multi Page Editor'

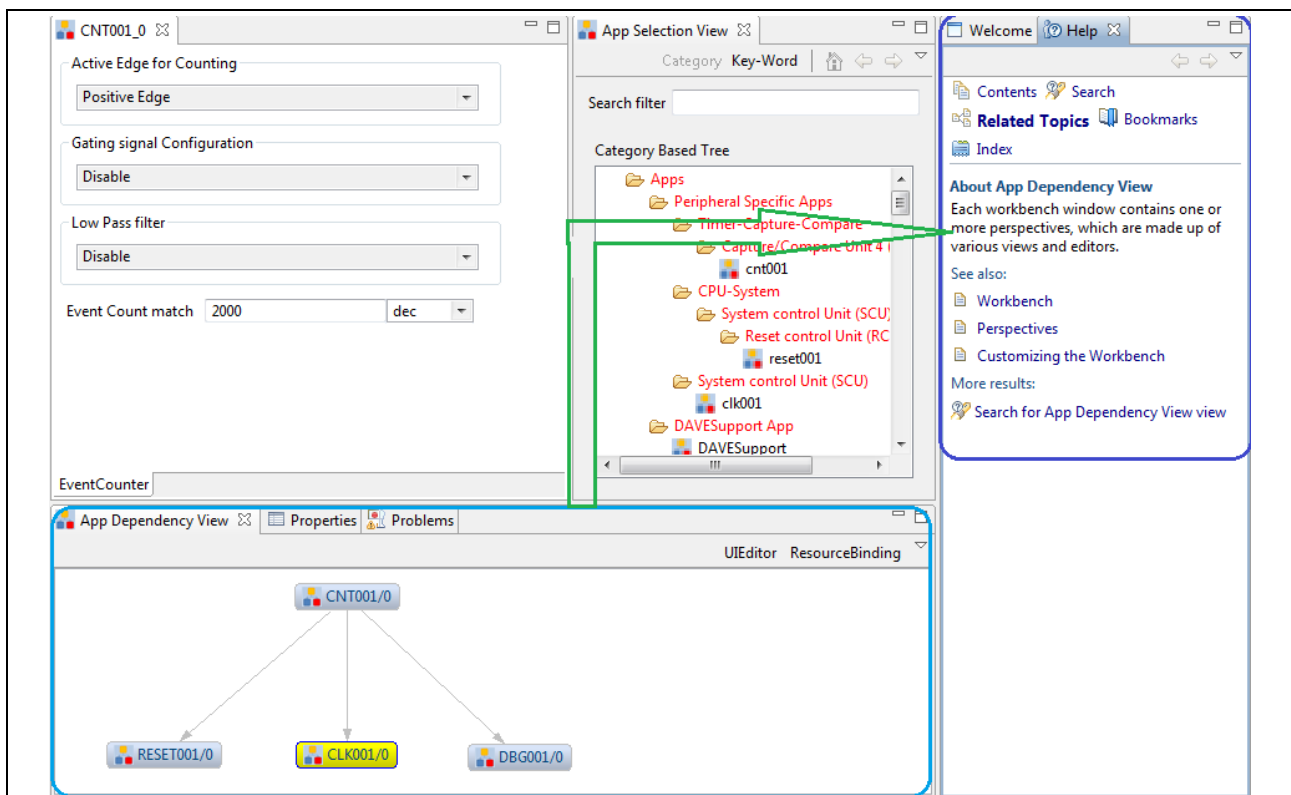


Figure 159 DAVE™ Help About App Dependency View

## 8.2 DAVE™ App Help

By Selecting particular App – user will get help information with respect to that App

- Selection through 'App Selection View'

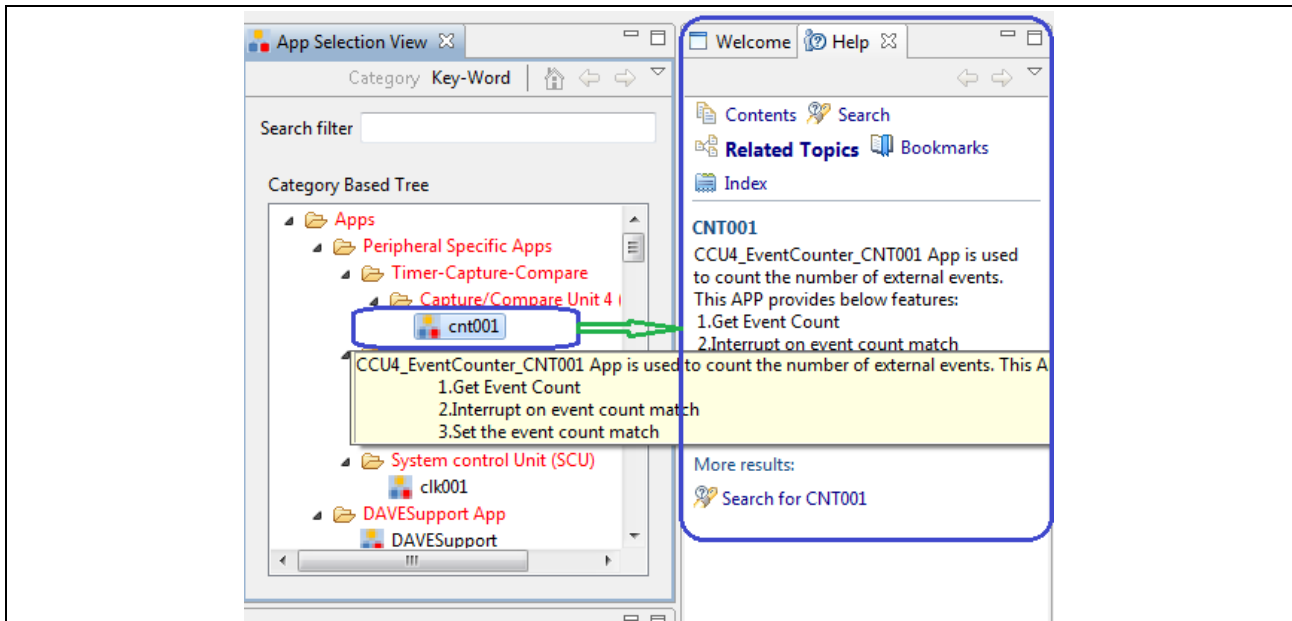


Figure 160 DAVE™ App Help – App Selection View

- Selection through 'App Dependency View'

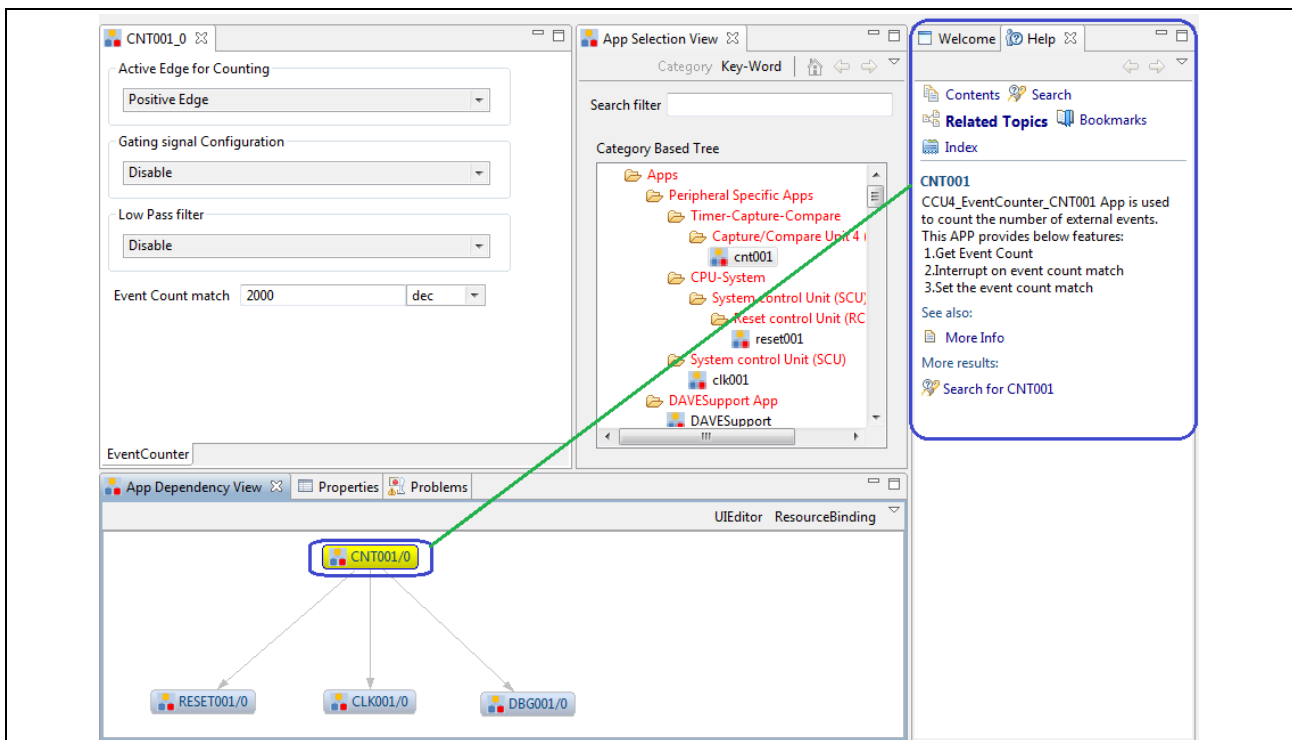


Figure 161 DAVE™ App Help – App Dependency View

More Info about App:

More information about App will be available by clicking 'More Info' Link.

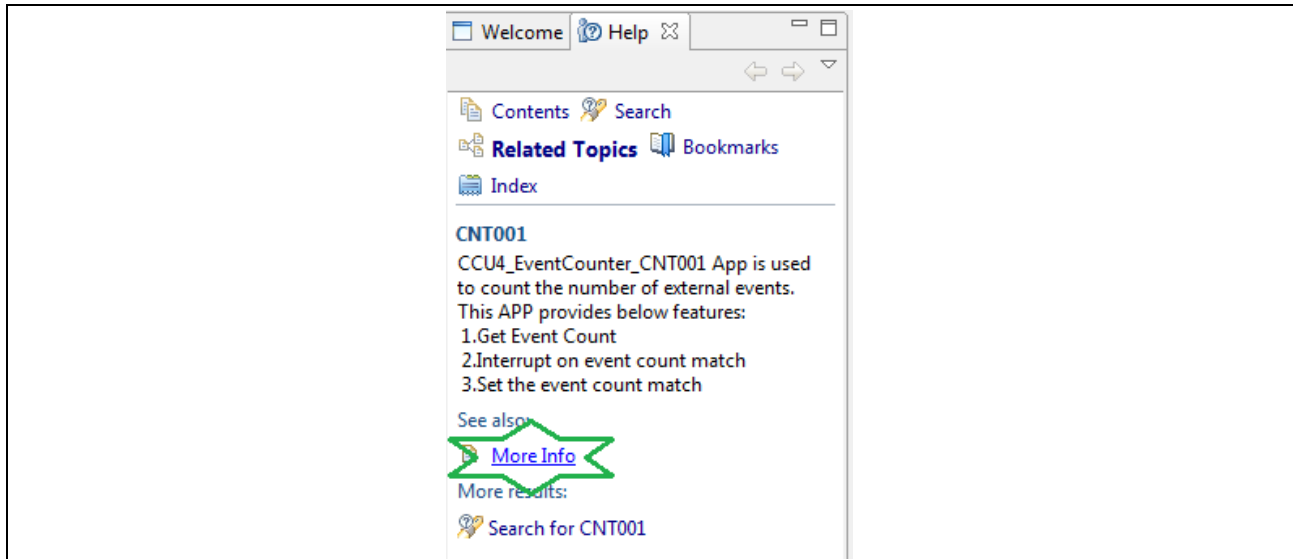


Figure 162 DAVE™ App Help – More Info

Help Via – UI Editor

- Help info are available via UI (Depends on available information provided by App)
  - By change or clicking the UI elements particular help information [html] is available
    - More Information is available by clicking 'More Info'

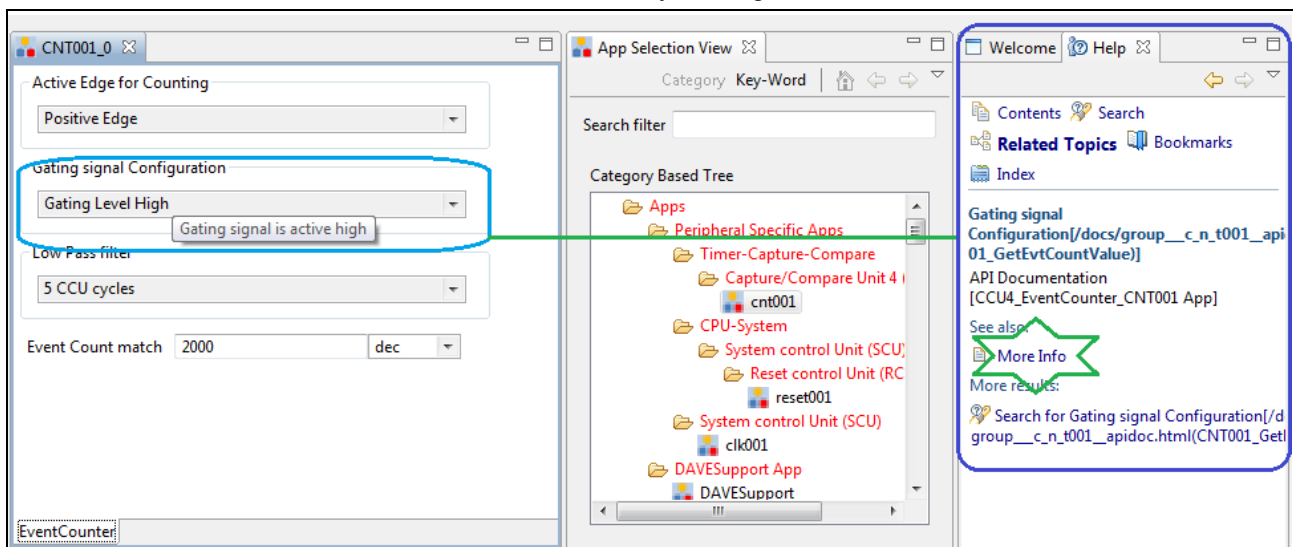


Figure 163 DAVE™ App Help – UI Change Help Information

- By change or clicking the UI elements particular help context information [html] is available

## 9 Signal Connection

### 9.1 Signal Connection – Signal to Signal

App signals can be connected other App signals as shown in below example

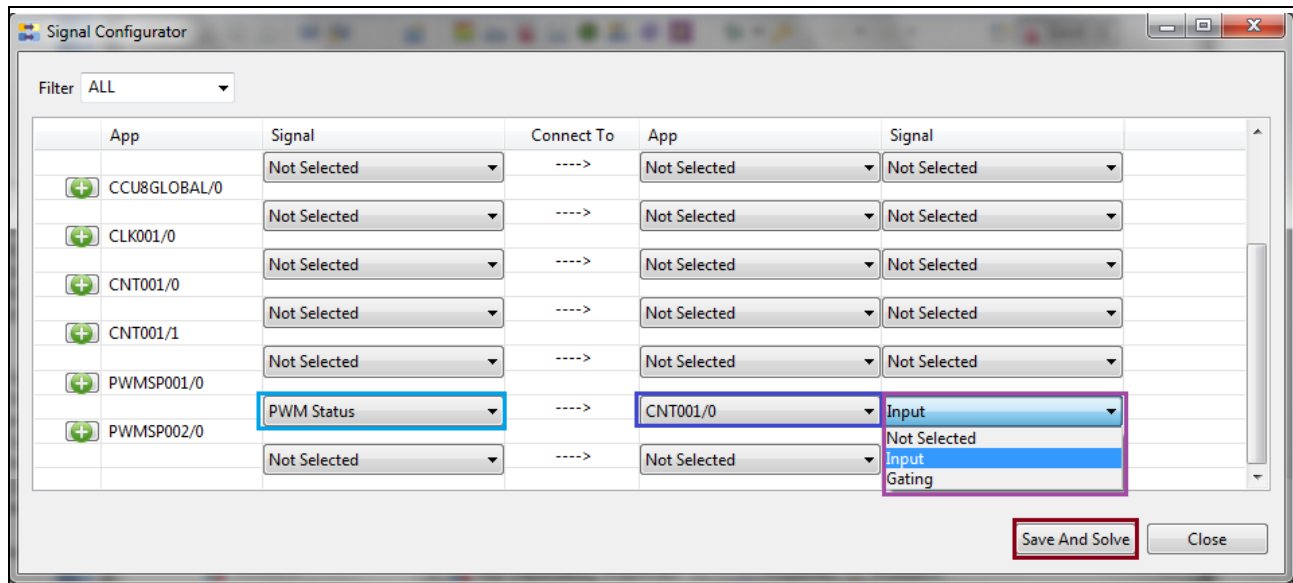


Figure 164 DAVE™ Signal Connection – Signal Input Selection

### 9.2 Signal Connection – Signal to Port

Pin Signals can be connected port pad using Manual pin assignment via Manual Pin Assignment.

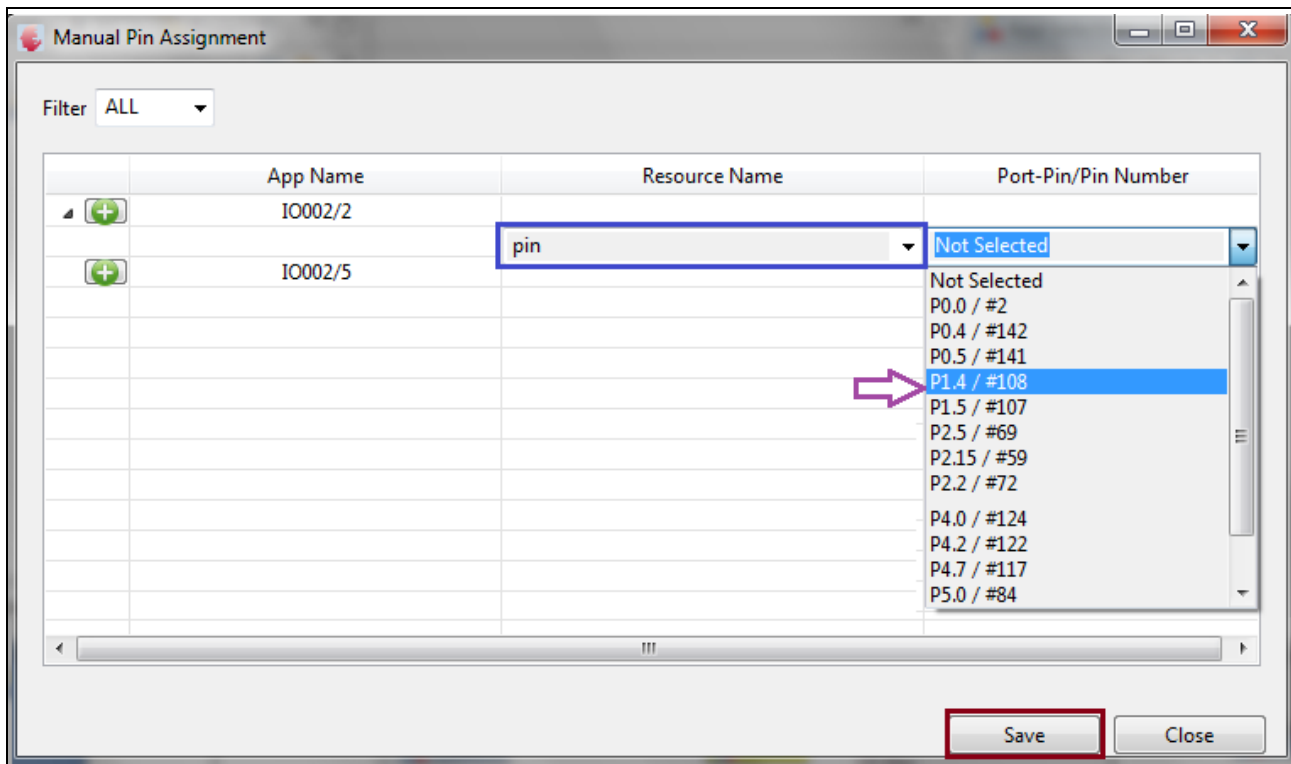


Figure 165 DAVE™ Signal Connection – Signal to Port - UART Rx

### 9.3 Signal Connection – Signal to Interrupt/ NVIC

Interrupt signals can be connected to NVIC. For example UART001 Alternate Receive Interrupt signal can be connected NVIC001 App NVIC Interrupt signal.

- Select Signal Connection
- Select Interrupt from Signal List
- Select App [NVIC001] from App List
- Select NVIC Interrupt from Signal List then click - Save and Solve



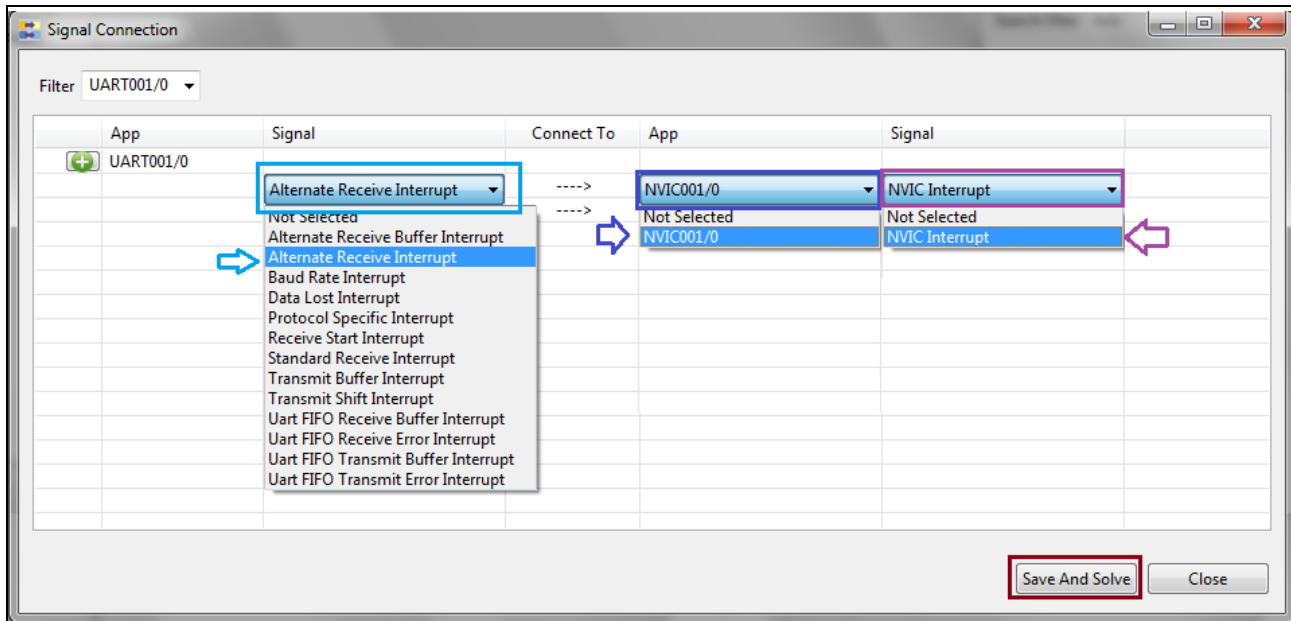
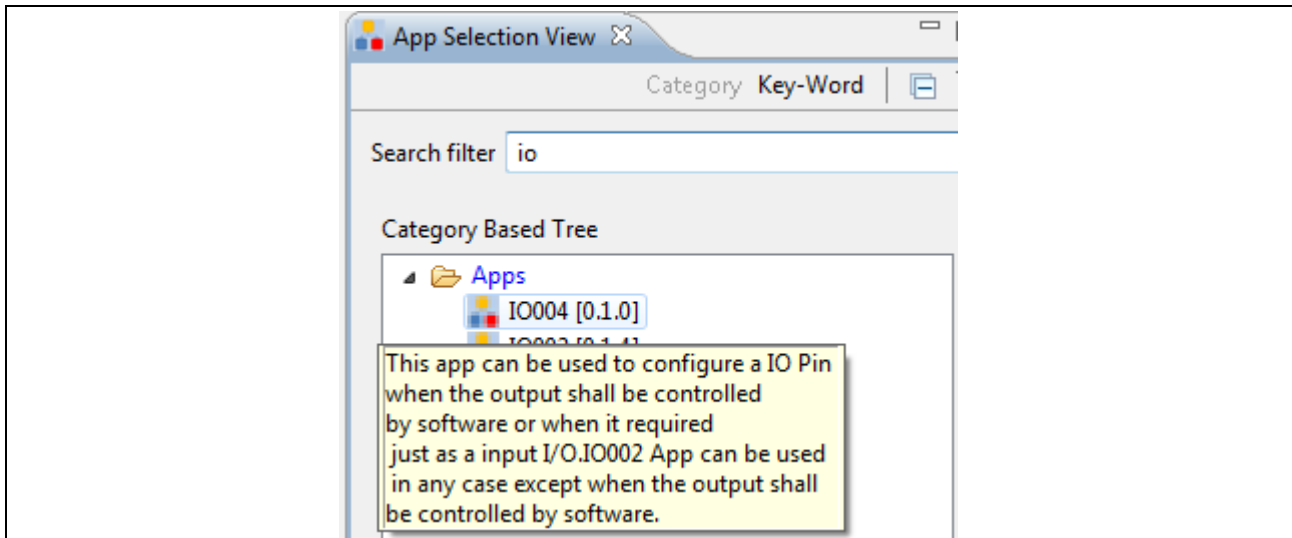


Figure 166 DAVE™ Signal Connection – Signal to Interrupt

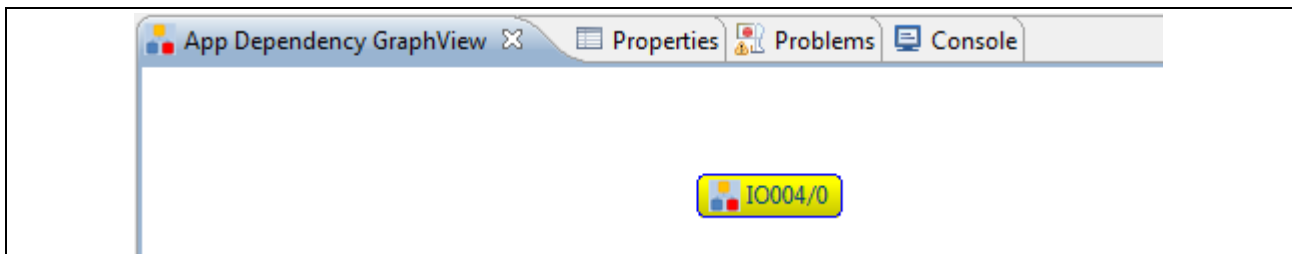
## 10 Manual Pin Assignment

### 10.1 Configuring GPIO Input



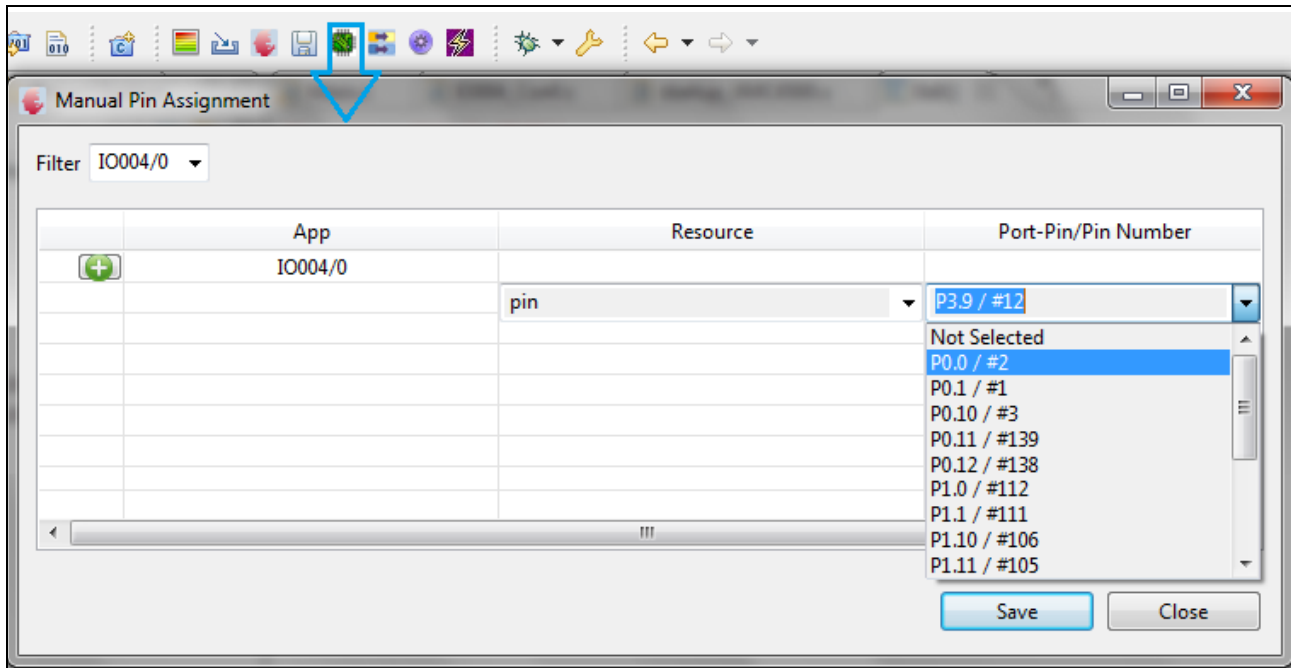
**Figure 167 DAVE™ Manual Pin Assignment – Input Pin 1**

- Select IO App e.g.: IO004



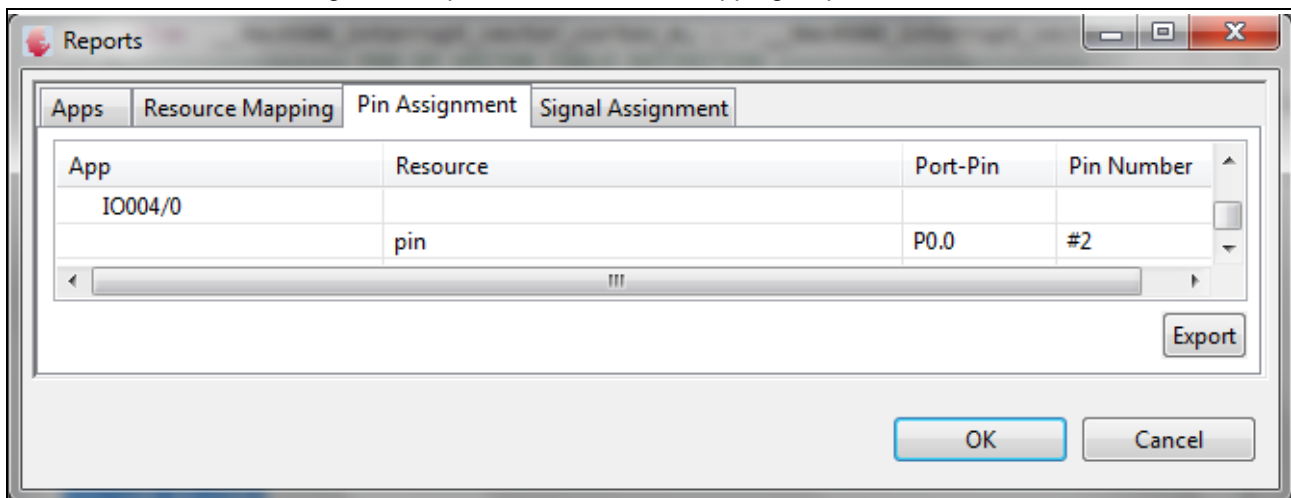
**Figure 168 DAVE™ Manual Pin Assignment – Input Pin 2**

- Select Manual Pin Assignment Menu
- Select App Instance – Select pin from Resource Name & chose pin in Port-Pin/Pin Number from list.
- Then Save & Run Solver



**Figure 169 DAVE™ Manual Pin Assignment – Input Pin – Select the Pin**

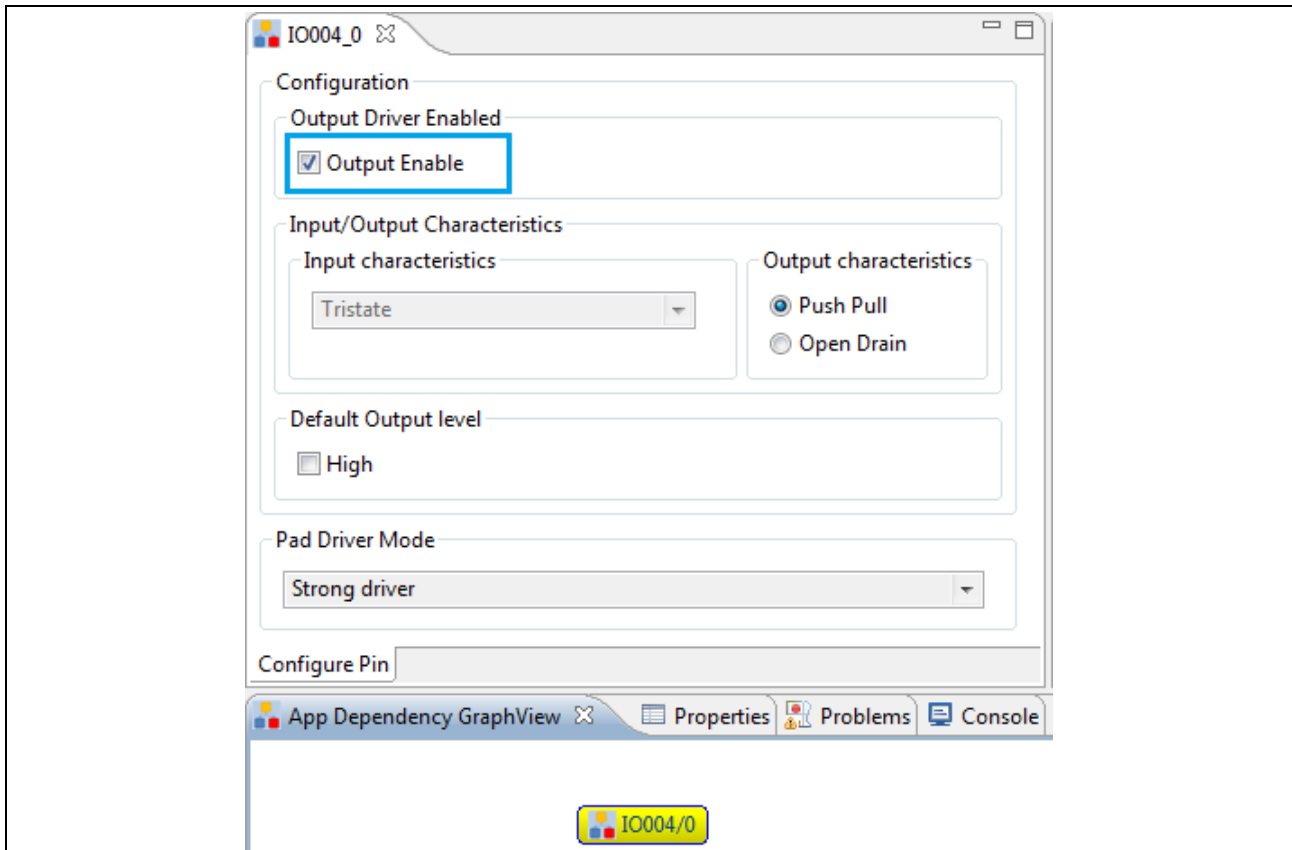
- Check the Pin Assignment report from Resource Mapping Report



**Figure 170 DAVE™ Manual Pin Assignment – Pin Assignment**

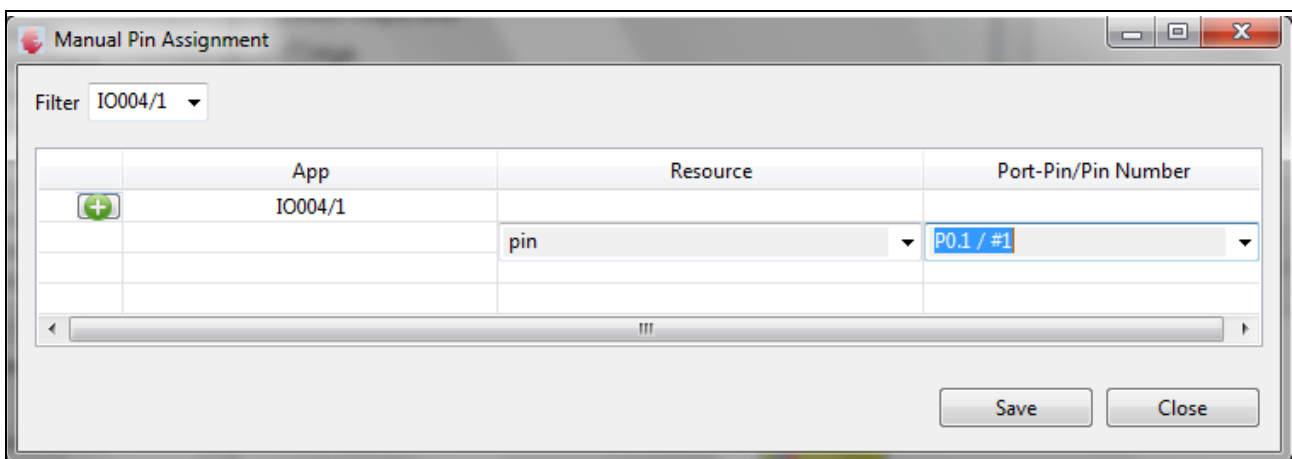
## 10.2 Configuring GPIO Output

- Select IO App
- Enable Output from UI Editor of IO004 App



**Figure 171 DAVE™ Manual Pin Assignment – Output Pin**

- Similarly configure Port-Pin/Pin Number from List from Manual Pin Assignment menu & save.
- Click 'Solver'



**Figure 172 DAVE™ Manual Pin Assignment – Output Pin – Select the Pin**

- Generate the code and check the IO004\_Config.c

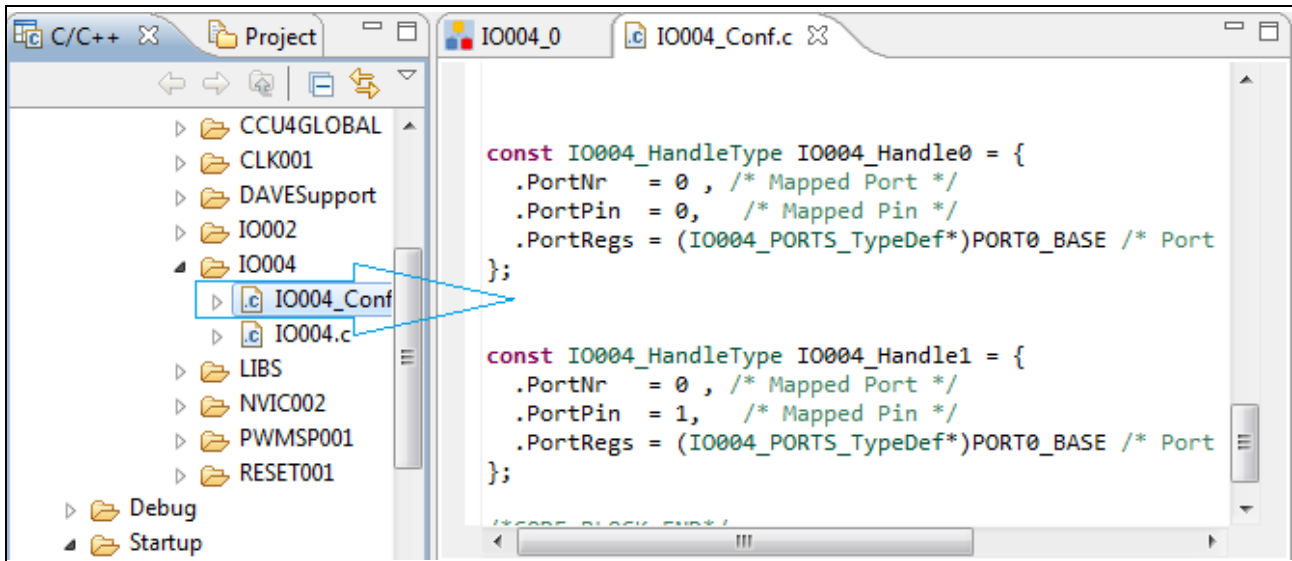


Figure 173 DAVE™ Manual Pin Assignment – Generate Code

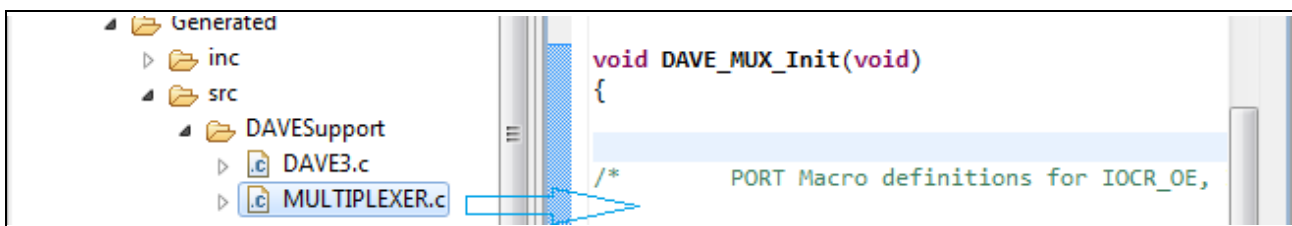


Figure 174 DAVE™ Manual Pin Assignment – Generated Code – Multiplexer.c

### 10.3 Changing the Pin solution provided by Solver to different one

- ➔ Check solver provided pin configuration by select 'Resource Binding Report'
  - Chose PIN Assignment List

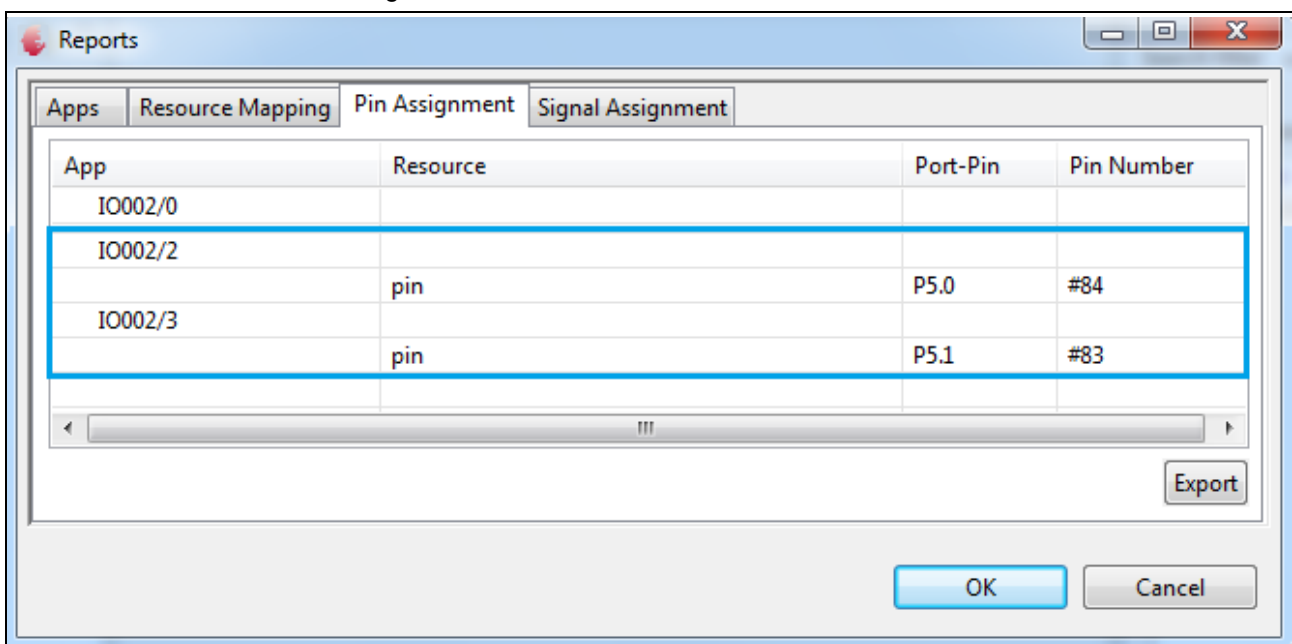
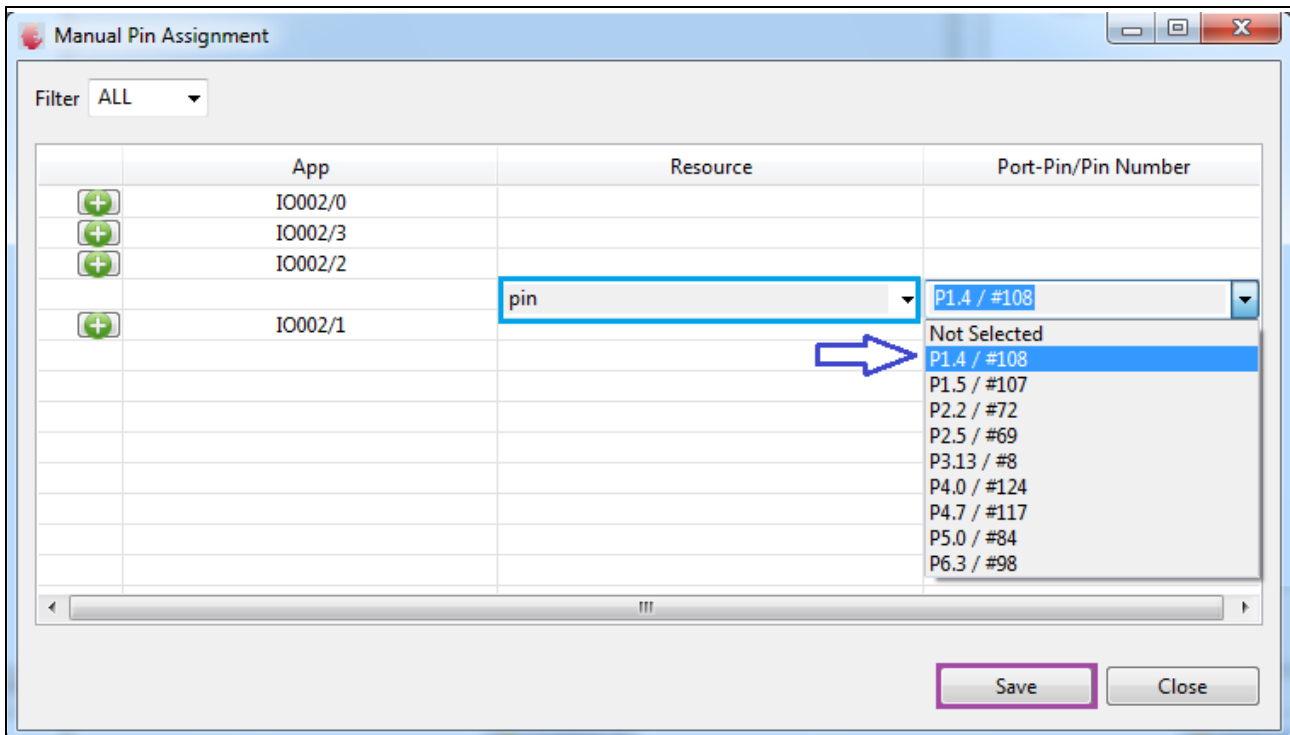


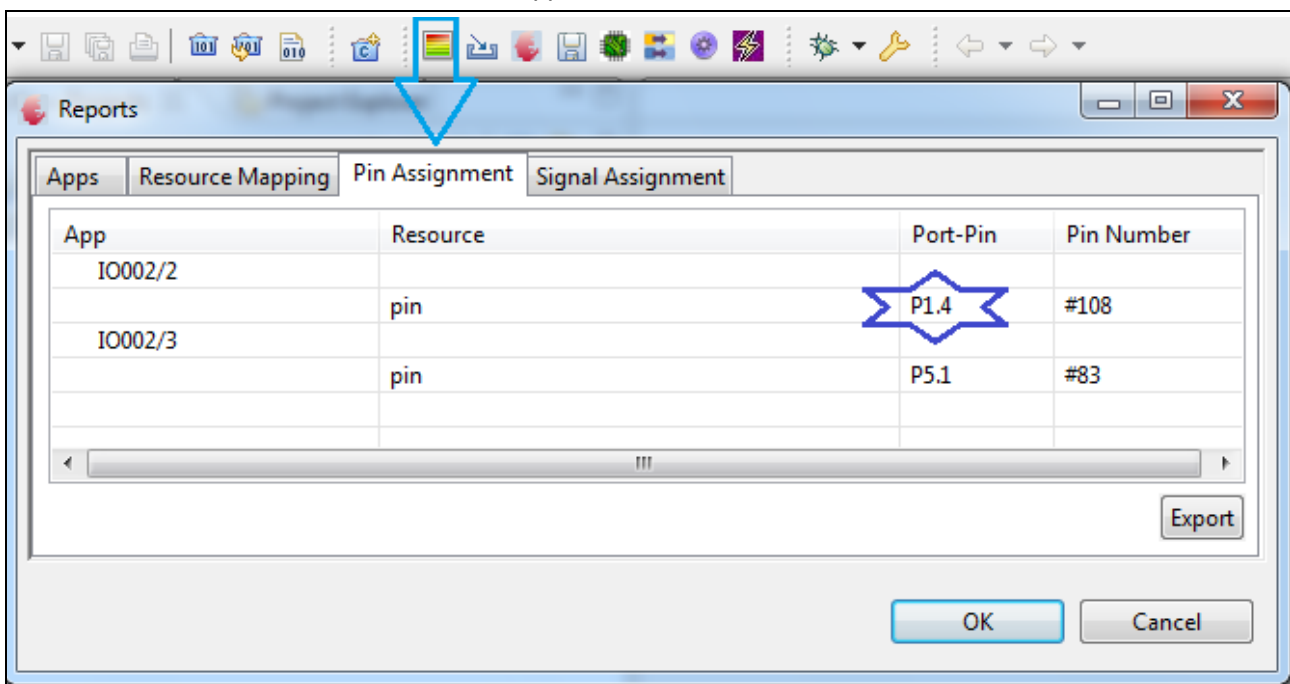
Figure 175 DAVE™ Manual Pin Assignment – Solver solution for IO002 App

- ➔ Select Pin Configuration & Select Resource Name – ‘Pin’
- ➔ Select Port-Pin/ Pin Number form list & Click – ‘Save’



**Figure 176 DAVE™ Manual Pin Assignment for IO002**

- ➔ Click Solver – is getting solutions for Pin Configurations
- ➔ By selecting Resource Mapping Report – Pin Assignment
  - Please observe IO002 is Mapped to P1.4



**Figure 177 DAVE™ Pin is changed to P1.4**

## 11 Report (Resource Binding Information)

- ➔ Resource Binding View help user to see how each resources are consumed by App. It has three kind of Information in the form of Report
  - Resource Mapping Report
  - PIN Assignment Report
  - Signal Assignment Report
- ➔ Which Slice of Resource is consumed by App
  - Example CCU4 – CCU40 / Channel 0
  - Example USIC – USIC0 / Channel 1
- ➔ To know which are the register/register bit field consumed by App
- ➔ Which are the Interrupt consumed by App

### 11.1 Resource Binding Report

Resource binding Report will contains information about App Proxy Resource to Mapped Resource

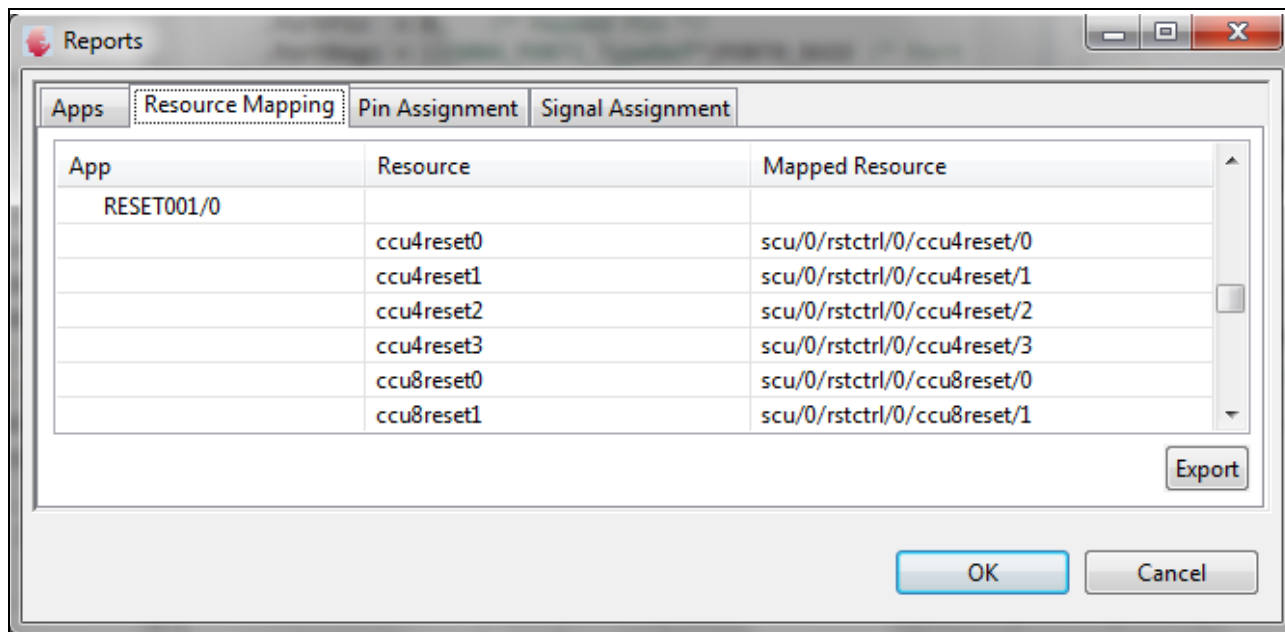
- ➔ Resource mapped to Register/ BitFields
- ➔ Resource

#### 11.1.1 Resource consuming SFRegister/Bitfield

ProxyResource	Mapped Resource	Description:
rcurst	Scu/0/rstctrl/0/rcurst	Rcrst is consuming SCU module RCURST Register
vadcreset	Scu/0/rstctrl/0/vadcreset	Same as Above – vadcreset Resource is consuming VADCRESET Register
Ccu4reset0	Scu/0/rstctrl/0/ccu4reset/0	CCU4RESET0 is consuming the Register bitfield of CCU4RESET bit 0
usicreset0	Scu/0/rstctrl/0/usicreset/0	USICRESET0 is consuming the Register bitfield of USICRESET bit 0

**Table 3 Resource consuming Register/Bitfield 1**

Below figure shows resett001/0 App



**Figure 178 DAVE™ Resource consuming SFRegister/BitField**

### 11.1.2 Resource consuming Slice

This helps user to know modules with slices will be available for example

USIC0/CH0, USIC0/CH1,...USIC1/CH1

CCU40/CH0, CCU40/CH1, CCU43/CH3

ProxyResource	Mapped Resource	Description:
Uart001_desbaudrate	NOT MAPPED	This is local variable, not mapped to any resource
channel	Usic/1/channel/0	Channel resource consuming one of available USIC channel – here it is consuming usic1 channel 0
Pwm001_temperature	NOT MAPPED	This is local variable, not mapped to any resource
slice	Ccu4/3/cc4/3	Similarly slice is consuming one of the available CCU4 channel – here it is consuming CCU43 Channel 3

**Table 4 Resource consuming Register/Bitfield 2**



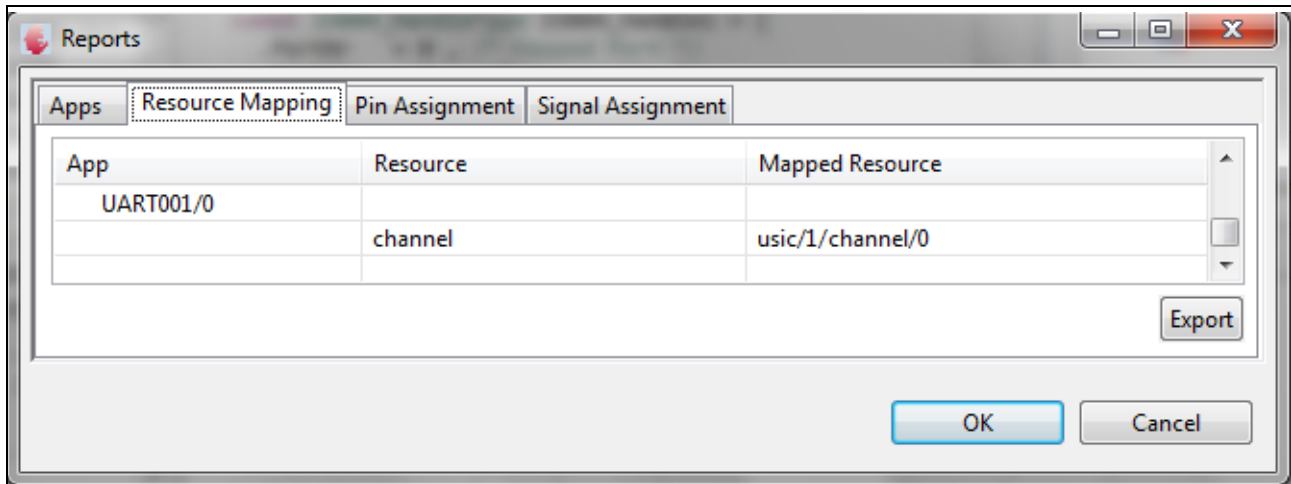


Figure 179 DAVE™ Resource consuming Channel/Slice

### 11.1.3 Resource consuming Interrupt Resource

Interrupt/NVIC app will consume the Interrupt/NVIC node. For example below example shows nvic002 app consumes nvic/interrupt/90.

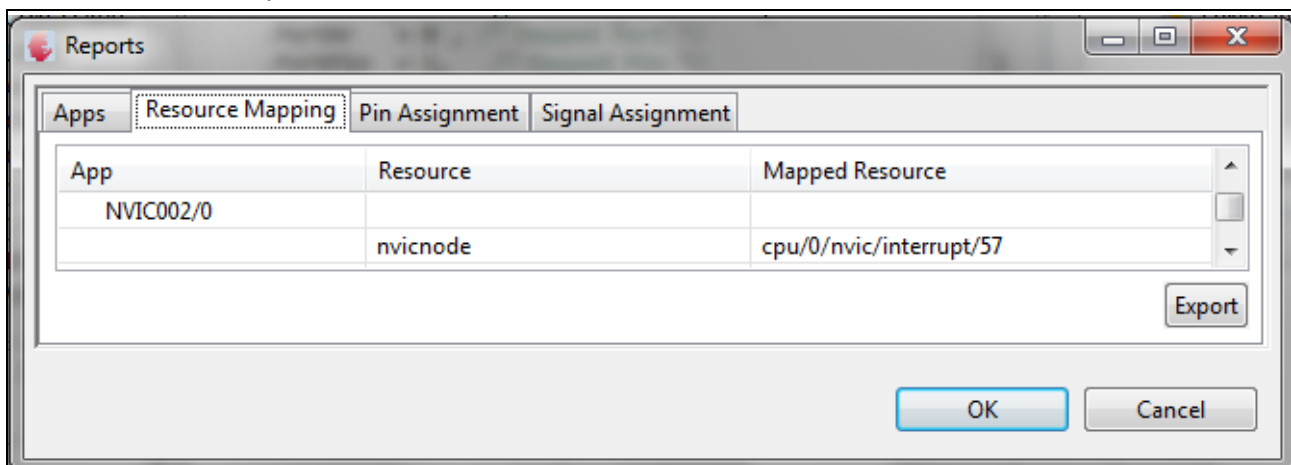


Figure 180 DAVE™ Resource consuming Interrupt Resource

### 11.1.4 Resource consuming PORT Resource

Port app will consume the port pad. For example below example shows io002 app consumes p/2/pad/14 that is P2.14.

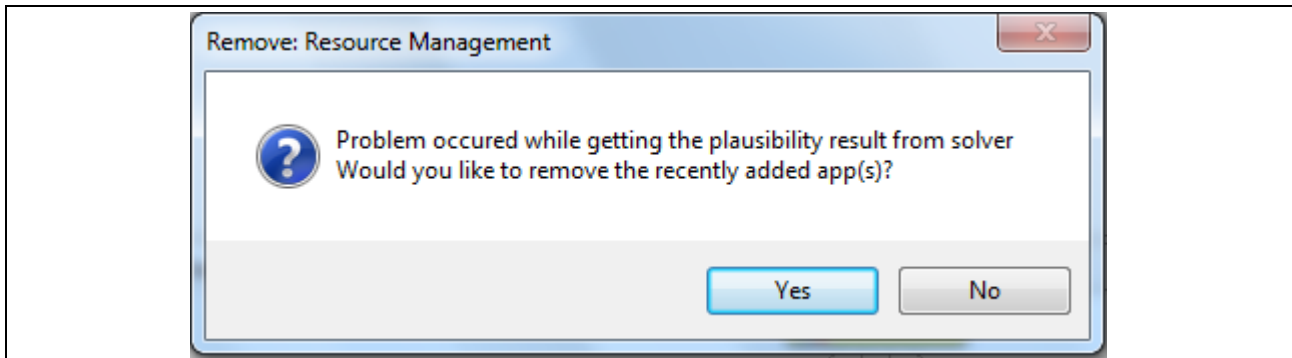
io002/2	pin	p/2/pad/14
---------	-----	------------

Figure 181 DAVE™ Resource consuming PORT Resource

### 11.1.5 Resource consuming Scenarios

Adding number of apps in app dependency view depends on the peripherals of the target/controller as per the selection. When the respective peripheral is not available to map to the selected app a message will be popped in a window.

Example, when UART/ I2C app is selected for more than one instance, depending on the non availability of USIC channels a message will be displayed.

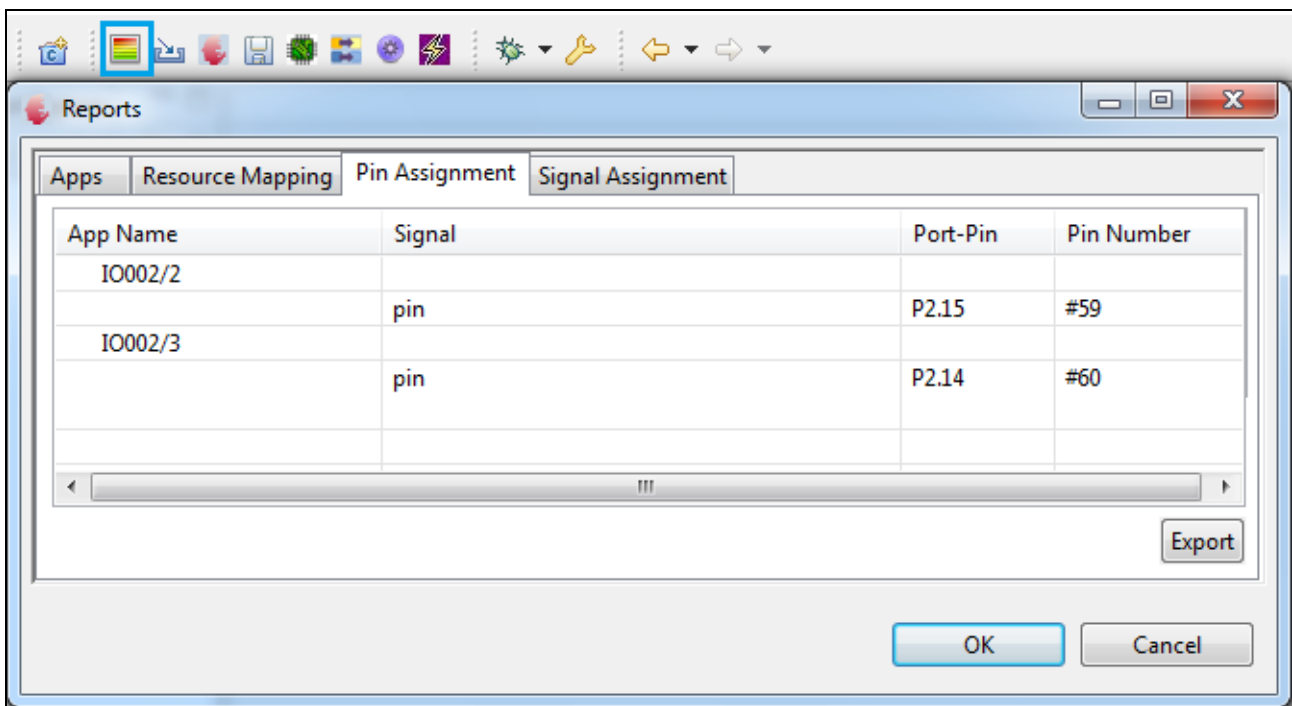


**Figure 182 DAVE™ Resource consuming error Scenarios1**

- When **yes** option is selected, the newly added App will be removed. And further the App which requires the same peripherals cannot be added.

## 11.2 PIN Assignment Report

PIN Assignment report will display list of Apps using Port-Pin and Pin Number it is using. User can export to \*.csv format.



**Figure 183 DAVE™ PIN Assignment Report**

## 11.3 Signal Assignment Report

Signal Assignment report will display list of Apps Signals Interrupt/Port using. User can export to \*.csv format.



## 12 DAVE™ App& Device Migration

### 12.1 Apps Migration Feature

With this feature it is now very easy to migrate in an existing project with older DAVE™ App versions to newer DAVE™ App versions. If the latest DAVE™ App version in the local library store is newer than the same DAVE™ App of the active project, then the respective DAVE™ Apps of the active project will be replaced by the latest version stored in the local library store. All relevant setting from the old DAVE™ App version will be copied to the new version.

Although a copy of the project will be stored in the “User\...\AppData\...” folder, we recommend to create a backup copy of the project that should be migrated to newer DAVE™ Apps versions before starting the migration process.

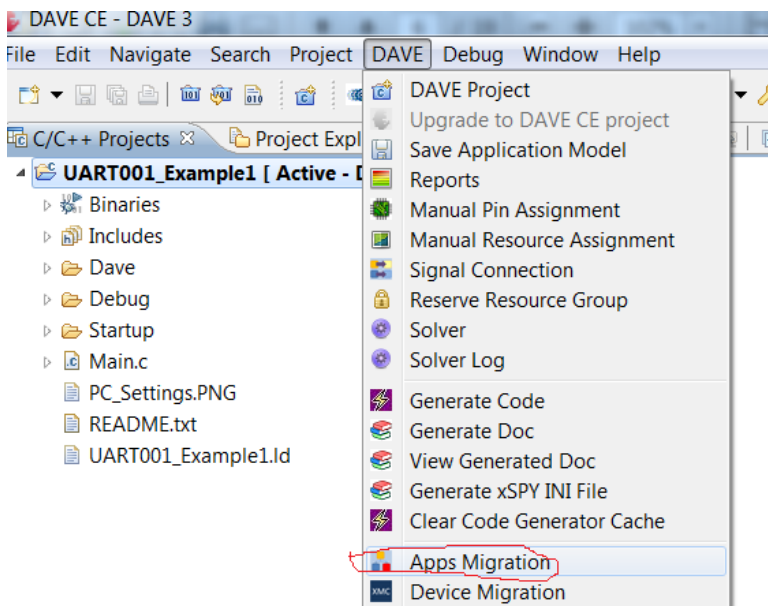


Figure 185 DAVE™ App Migration Feature

The migration process can be starting by pressing “App Migration” in the DAVE menu. Then the DAVE Apps of the active project will be migrated to the latest DAVE App version that can be found in the local library store.

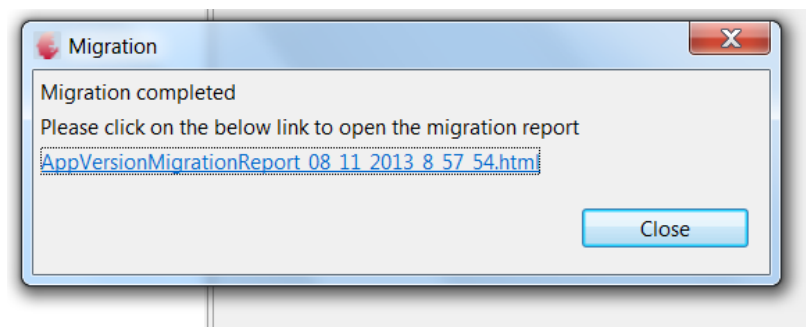


Figure 186 DAVE™ App Migration Finish Dialog

After the migration is finished a migration report has been created (html format) that can be viewed with a web browser. After pressing close, the report will be copied into the active project.

Pre Migration Report		
List of Apps		
App Name		Message
CLK001 [v1.0.32]	Errors:0 Warnings:0 Info:0	No higher version found in the local App store
DAVESUPPORT [v1.0.28]	Errors:0 Warnings:0 Info:0	Higher version 1.0.32 found in local App store
DBG001 [v1.0.10]	Errors:0 Warnings:0 Info:0	No higher version found in the local App store
LIBS [v1.0.10]	Errors:0 Warnings:0 Info:0	Higher version 1.0.18 found in local App store
MOTORLIBS [v1.0.18]	Errors:0 Warnings:0 Info:0	No higher version found in the local App store
RESET001 [v1.0.10]	Errors:0 Warnings:0 Info:0	No higher version found in the local App store
UART001 [v1.0.12]	Errors:0 Warnings:0 Info:0	Higher version 1.0.16 found in local App store

Among other information the migration report contains a list of all migrated DAVE Apps including number of errors, warnings and info that are shown in a separate table.

**Figure 187 DAVE™ App Migration report**

It can happen that a migration causes error, in particular if the active project contains very early releases of DAVE™ Apps. In this case the migration process cannot be reverted. Therefore we recommend creating a backup before starting the migration process. If the user has not done so, DAVE™ copies the project to the following location:

C:\Users\<user>\AppData\Local\Temp\UART001\_Example1\_08\_11\_2013\_8\_57\_15.zip

From there it can be re-imported to the current or any other workspace (this is also described in the migration report).

The migration problems that may occur under certain circumstance and respective correction options are described below.

#### 1. IO002 Pad Speed class inclusion in solver query

##### Problem Description:

The old version of the IO002 Apps has the pad speed class selection with A1 as default and the selection was not included for the solver query. However, the new version includes the pad speed class selection to the solver query. This may lead to solver error condition.

Example: Apps which consumes IO Apps, like CAN Examples, Motor control Examples (latest published examples)

##### How to solve the solver error with pad speed class?

User should review IO002 App to change the pad class selection to appropriate class type which is supported by the device. The Pad Speed Class selection UI control is shown below, This will solve the issue for the IO Apps which has the output mode configured.

#### 2. Library ordering for Linker

##### Problem Description:

As the Apps Migration creates the new version of the Apps, copy the values and generates the code, The Library ordering for linker need to be taken care as like new project creation with Library inclusion from the Apps.

##### How to avoid the build errors due to the library ordering change?

User should review “Active Project Settings □ Tool Settings □ ARM-GCC C Linker □ Libraries”, and then change desired order manually as explained in the Apps help document.

3. Required App changed from IO002 / IO004

Problem Description:

The version increment of App changes the required App from IO002 to IO004. The Apps Migration will report it as missing App in the new version and will instantiate the new Apps with default values as required by new version of the top level App.

How to get the working state of the project?

User should take care of configuring the new App with desired values.

4. UI data copy: UI Difference

Problem Description:

As the Apps Migration creates the new version of the Apps, copies the values and generates the code, UI value copied from old version to new version may be incorrect for some cases like parameter-to-parameter dependency.

How to correct the UI value?

User can review and change it to the required value as in the original project.

5. Apps increment with API signature change

Problem Description:

There is some Apps update with a change in provided APIs, The Apps Migration will give the error if the project used the APIs in the Application code.

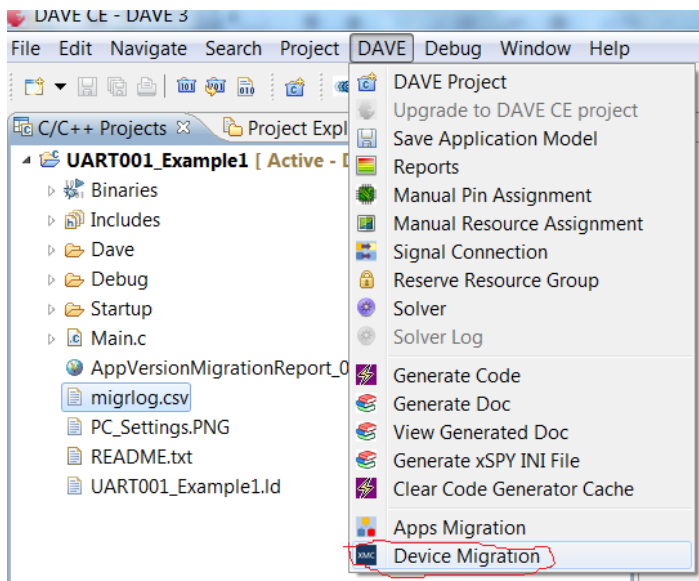
How to correct the build error?

User should review the application code and modify the API calls accordingly.

## 12.2 Device Migration Feature

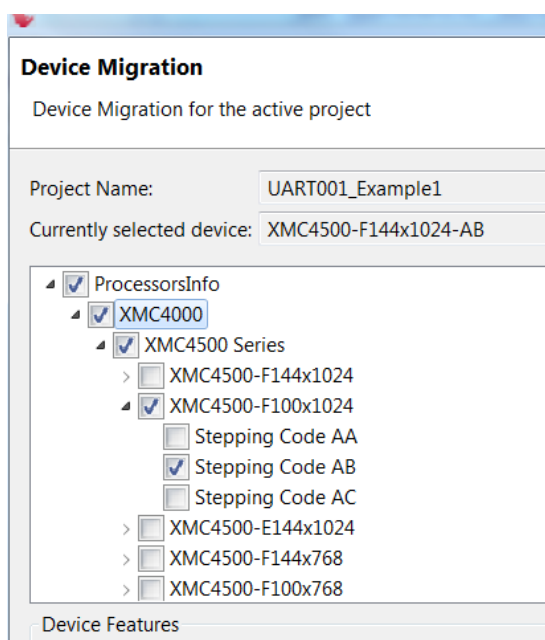
This feature allows changing the target MCU device of the active project to another target microcontroller within the same XMC series.

Although a copy of the project will be stored in the “User\...\AppData\...” folder, we recommend to create a backup copy of the project that should be assigned to a different target MCU before starting the migration process



The target device migration process can be starting by pressing “Device Migration” in the DAVE menu. Then the target MCU device of the active project will be migrated to a different target MCU device within the same series defined by the user.

Figure 188 DAVE™ Device Migration Feature



The user can define any other target device or any other device step within the same XMC series.

Figure 189 DAVE™ Device Migration Feature

After the migration has been successfully finished, code generation and build is started automatically.

Similar to the DAVE™ App migration, also here, a migration report with all migration details will be copied into the project.

If the migration fails it will be reverted to the original target device.

In any case a copy of the project has been made before migration will be started in a similar manner as described for the DAVE™ App migration functionality.

A migration problem may occur if one or more DAVE™ Apps of the active project do not support the new target MCU device. This could be because:

- a) The DAVE™ App has not yet been extended to support this new target device or new device step.
- b) The new target device does not include a required peripheral (e.g. ETH module or USB module)
- c) The new target device includes less peripheral instances than required
- d) The new target device does not include the manually assigned port pins

Case a: This should be only an intermediate situation. The respective DAVE™ App should be updated soon.

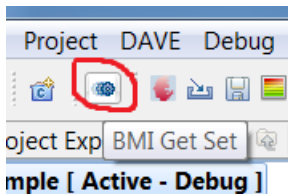
Case b and c: In this case respective DAVEApps have to be removed to fit the resources offered by the new target device

Case d: The user has to manually assign port pins that are available in the new target device, or remove the critical manual pin assignment and let the solver search for a solution.



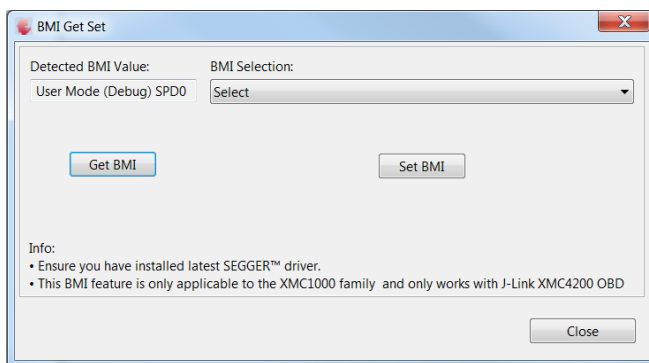
## 13 BMI Feature

The BMI Feature provides the functionality to read and change the BMI settings of the XMC1000 family.



The BMI Get SET button can be found in the tool bar.

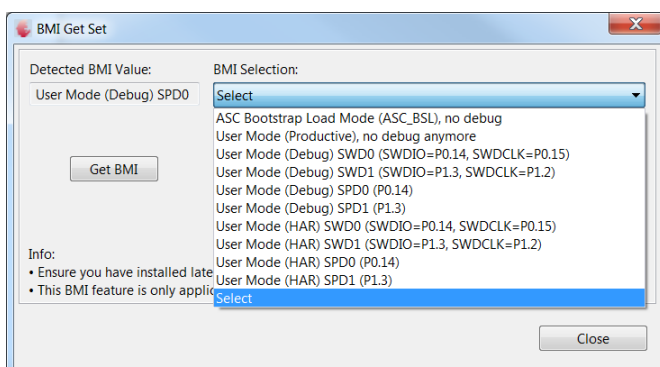
**Figure 190 DAVE™ BMI GET/SET Feature Toolbar button**



Pressing the BMI GET Sett button in the tool bar will bring up the BMI GET SET user interface.

Pressing the Get BMI button will show the current BMI setting in the field “Detected BMI Value”.

**Figure 191 DAVE™ BMI GET/SET Feature Dialog**



The “Select” menu provides all BMI setting options.

After selecting the BMI target setting and pressing the Set BMI button the XMC1000 chip will be configured with the new setting.

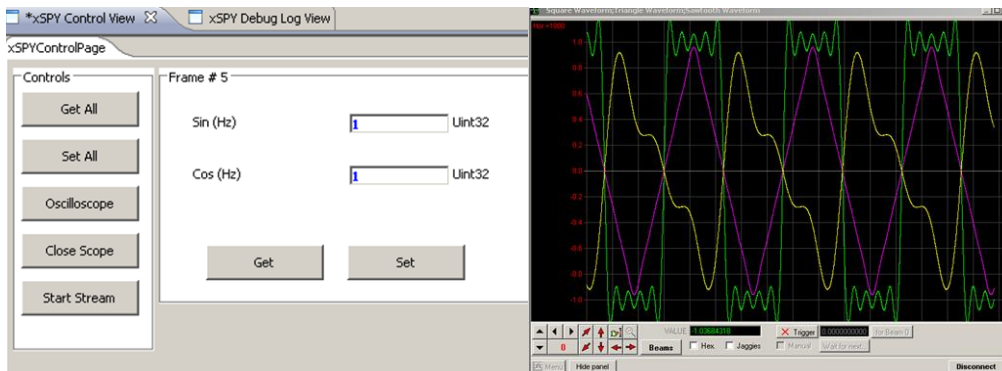
**Figure 192 DAVE™ BMI GET/SET Feature Dialog - Selections**

This BMI functionality works with the Segger J-Link OBD with the XMC4200 or with the regular commercial Segger J-Link or the J-Link EDU.

Please note, after setting the “User Mode” the chip might not be accessible any more by any tool if the programmed user code in the Flash doesn’t allow this anymore

## 14 xSPY Feature

xSPY is a free and flexible plug-in for DAVE™ to visualize data and to create an interface dashboard to control XMC target applications.



**Figure 193 DAVE™ BMI GET/SET Feature Dialog - Selections**

With xSPY and the DAVE™ App DBG002 it is possible to instrument the application software to add trace capabilities and to visualize the trace data on the PC. Vice versa, variables can be modified from a freely definable PC dashboard.

Main Features:

- Real-time streaming of data from the XMC target to the PC
- Visualization of streaming data on an oscilloscope
- Reading and modifying of variables on demand from a PC dashboard
- Communication via UART and virtual COM to the PC (Segger J-Link Lite OBD)

Please refer xSPY User Manual for more details (Help -> Help Contents -> xSPY User manual)

## 15 ARM GCC compiler

With the DAVE™ 3.1.8 installer, as well as in the zip file package we included the latest ARM GCC version (4.7 2013q2) which provides fixes of issues and performance improvements. ARM GCC is maintained by ARM® and further details are available from the below website

<https://launchpad.net/gcc-arm-embedded>

If ARM GCC v 4.6 compilers should further on be used, just save the ARM GCC folder of DAVE™ v 3.1.6 before installing DAVE™ v 3.1.8 and then replace the ARM-GCC folder in the DAVE™ v 3.1.8 installation with the ARM GCC folder from the older DAVE™ version.

## 16 FAQ

### 16.1 Project Management

#### 16.1.1 How user will know which perspective is active?

- By clicking the right most perspective tabs user can chose different perspectives.
  - o DAVE™ CE perspective
  - o DAVE™ IDE perspective

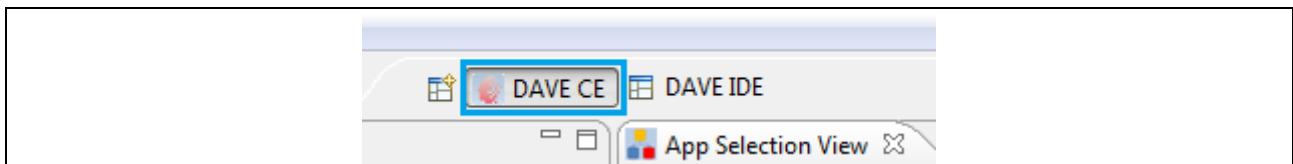


Figure 193 DAVE™ CE perspective

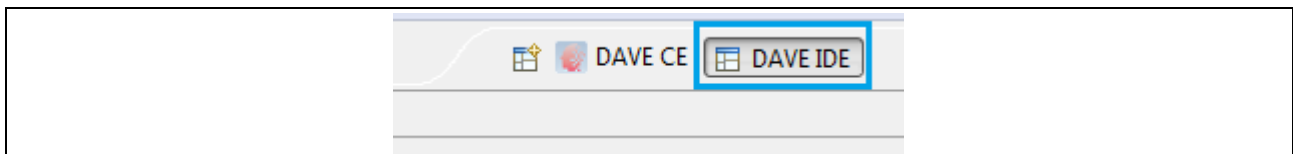


Figure 194 DAVE™ IDE perspective

#### 16.1.2 What is DAVE™ IDE and DAVE™ CE perspective?

##### DAVE™ IDE

- On creation of new DAVE™ workspace, by default DAVE™ IDE perspective will be opened.
- This perspective is mainly used for the creation of projects, building the project etc...

##### DAVE™ CE

- This is the main perspective of DAVE, where major tasks will be performed.
- This perspective is used for App selection, binding, Pin configurations and other tasks

#### 16.1.3 How user will know about Active Project?

- Active project will be highlighted in project window with [Active-Debug]

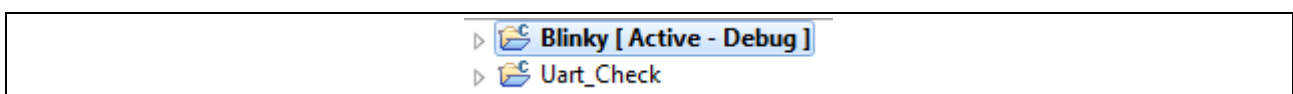


Figure 195 DAVE™ Import & Export

### 16.1.5 Why Solver will throw Error?

- Solver will throw error may be due to
  - o wrong port number,
    - user can change this port number to the other available port number

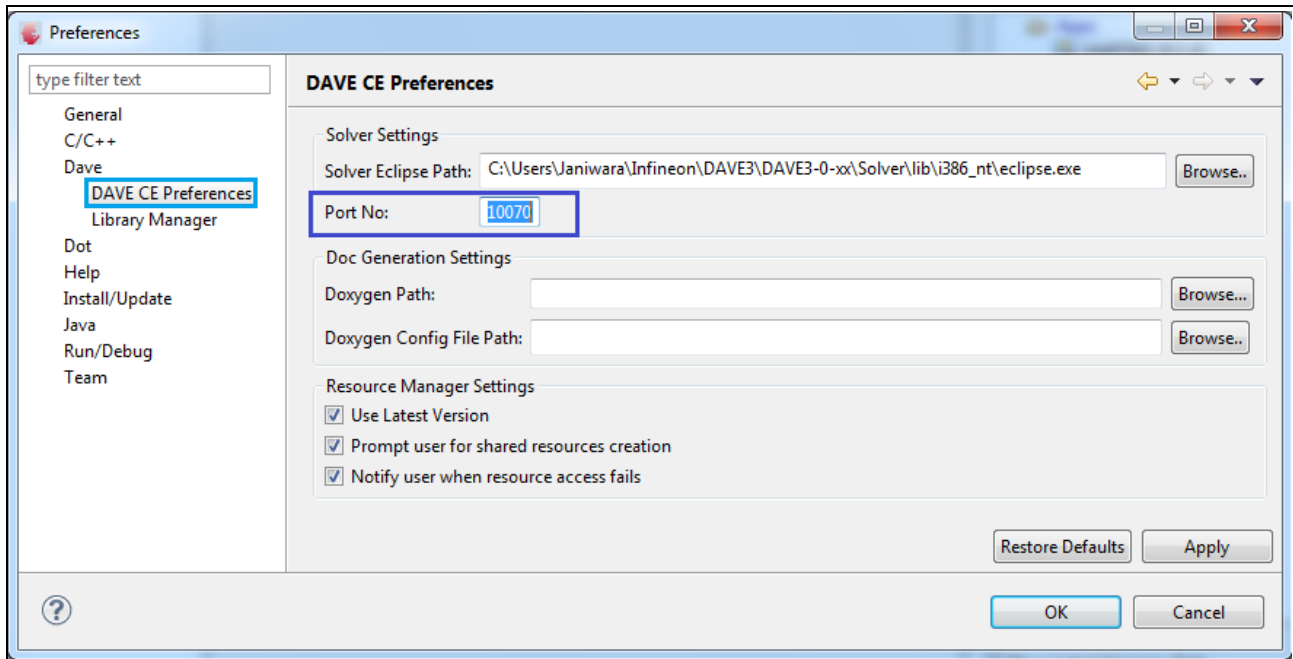


Figure 196 DAVE™ Solver Error – Port number

### 16.1.6 How to check which device is configured / selected for the current project?

- User can check the device information, by following method
- Right click on the project, select properties option
- Select C/C++ Build->MCU Selection Page to know the device selected.

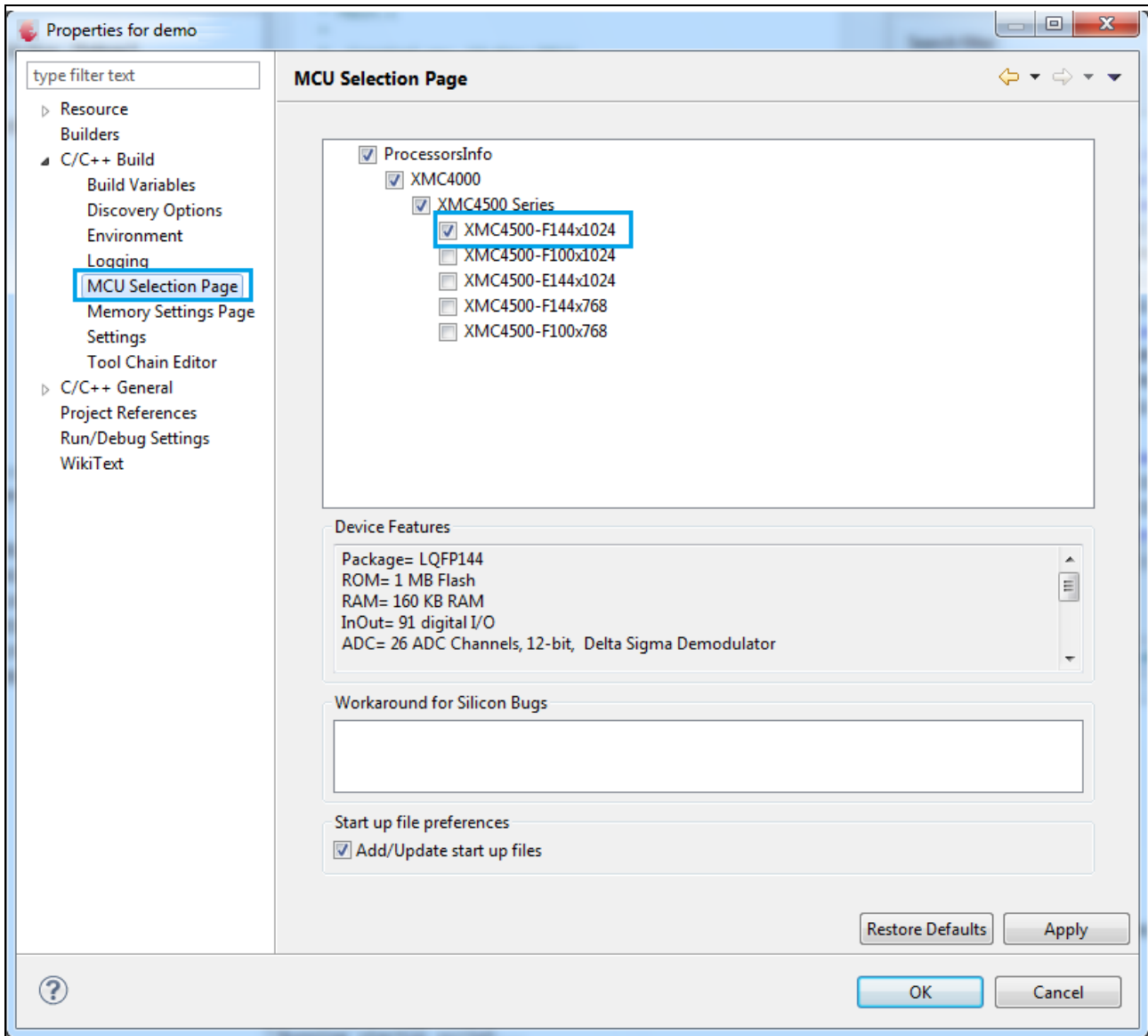


Figure 197 DAVE™ Device selection information

## 16.2 Versions of App

### 16.2.1 How can user know which Version of App is installed?

- App Version will be provided along with App Name in Square Brackets[] in app selection View.

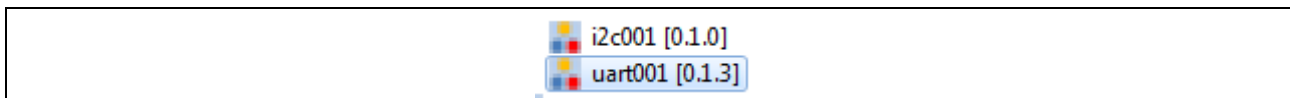


Figure 198 DAVE™ Install Apps – App Version

### 16.2.2 How many versions of the same App installed in Library?

The different versions of the Apps are visible in App Selection View as per the installation, when show latest version check box is unchecked all installed versions of the Apps are visible.

User can choose the App as per the use case, depends upon the version.

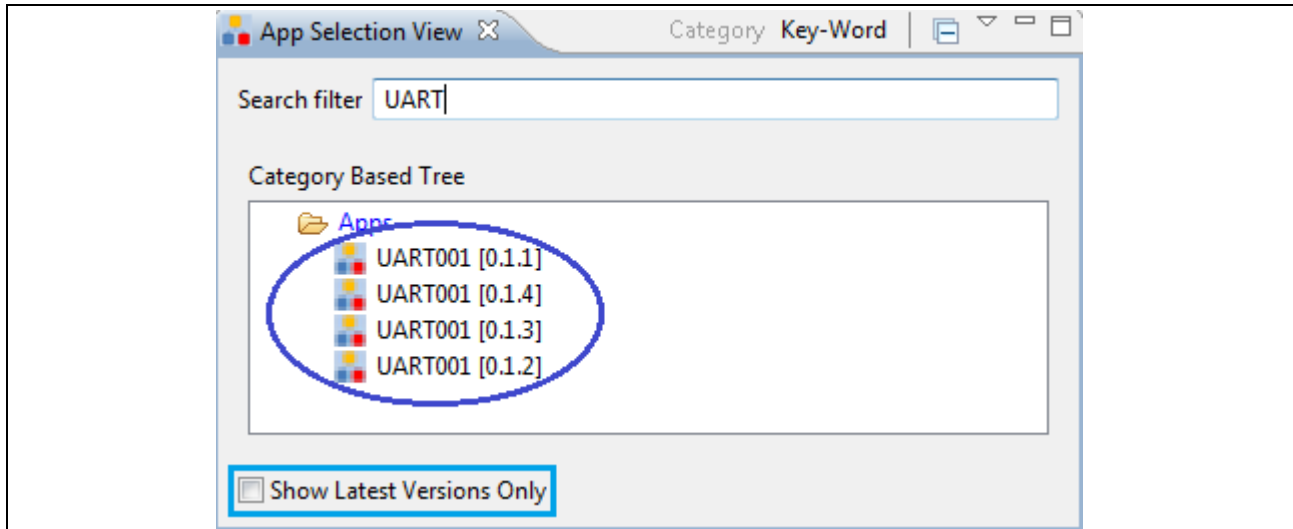


Figure 199 DAVE™ Versions of Apps

## 16.3 UI Configuration

### 16.3.1 How different conversion types will work for an integer parameter?

- Frequency related conversion types used are Hz, KHz and MHz
- Time period related conversion types used are nsec, usec and msec
- Baud rate related conversion types used are baud, Kbaud and Mbaud

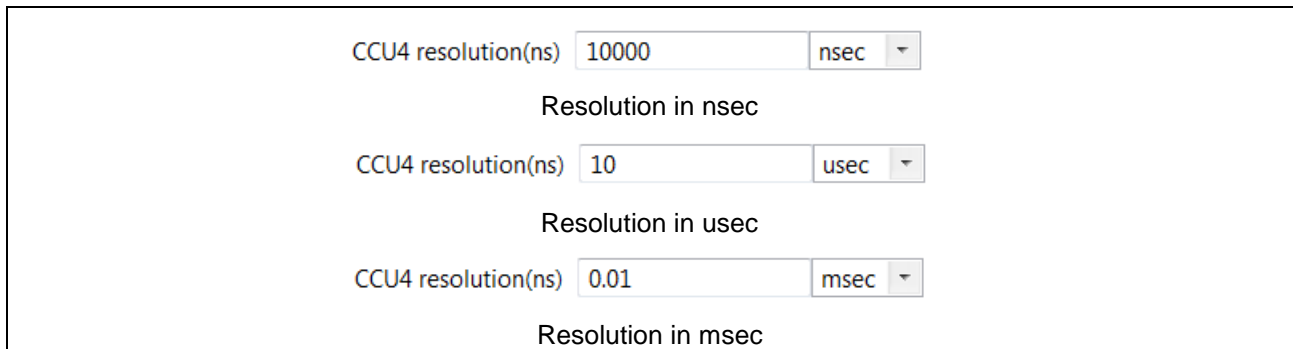


Figure 200 DAVE™ Resolution in seconds

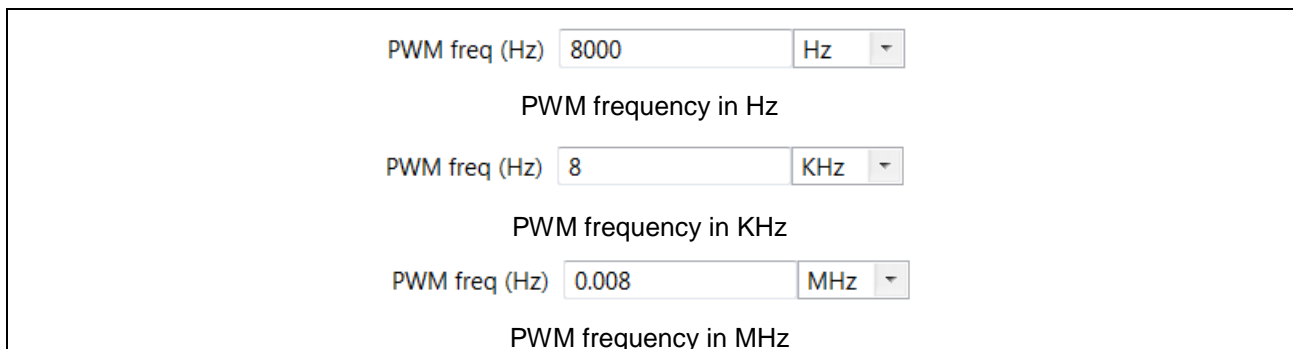


Figure 201 DAVE™ Resolution in Hertz (Hz)

Real Baud Rate  baud ▼  
 Real baud rate in buad

Real Baud Rate  Kbaud ▼  
 Real baud rate in Kbaud

Real Baud Rate  Mbaud ▼  
 Real baud rate in Mbaud

**Figure 202 DAVE™ Resolution in baud**

Count Match  dec ▼  
 Count match value in decimal

Count Match  bin ▼  
 Count match value in binary

Count Match  hex ▼  
 Count match value in hexadecimal

**Figure 203 DAVE™ Count match value**

## 16.4 Code, Compiler & Debugger

### 16.4.1 Which are the generated files that user can edit?

- User can edit only Main.c file. If other \*.c/\*.h files are edited, they will be overwritten upon re-generation of code and user will lose his edited data.

### 16.4.2 What are Multiplexer.c & h files?

- Multiplexer.c & .h files are DAVE™ generated files for configuring Multiplexed register configuration. Configured by Apps using solver.

### 16.4.3 Which are the compilers DAVE™ Supports?

- DAVE™ supports ARM GCC compilers.



<http://www.infineon.com>