

Installation and Quick Start of iSYSTEM's winIDEA Open in DAVE™

Tutorial

Version 1.0, May, 2014



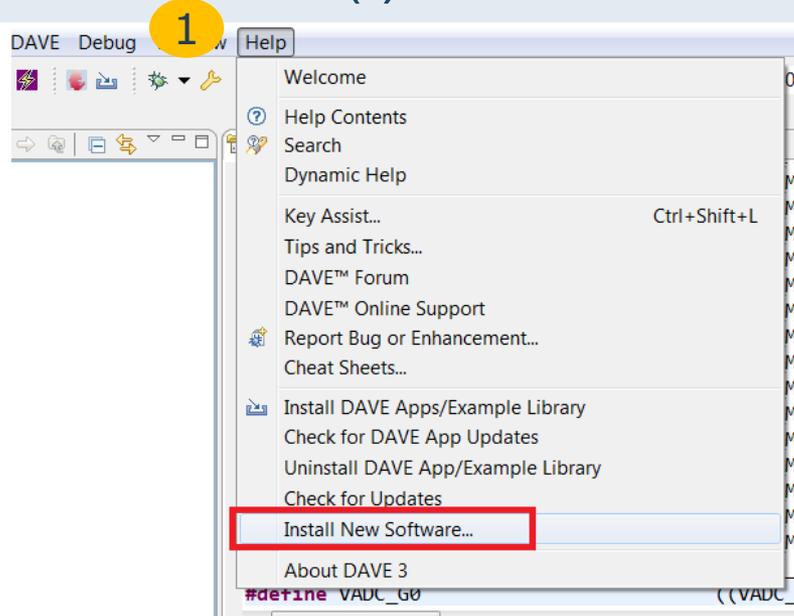
About winIDEA Open

- iSYSTEM provides a free version of its debugger IDE called winIDEA Open; it can use the Segger J-Link SW and J-Link HW to connect to the MCU target
 - [Debugger in a proprietary IDE plus the ARM GCC](#)
 - [Eclipse plug-in to use the CDT debug perspective or the proprietary IDE](#)
- The eclipse plug-in can be installed in DAVE to extend the debug capabilities of DAVE
- The winIDEA Open Debug IDE provides extended functionalities beyond the default DAVE debugger:
 - Real-time period update and graphical display of variables while the target MCU is running
 - RTOS Kernel aware debug e.g. FreeRTOS
 - Integrated unit test tool (without code instrumentation)
 - Process automation and integration, scripting (isystem.connect)

Content of this Tutorial

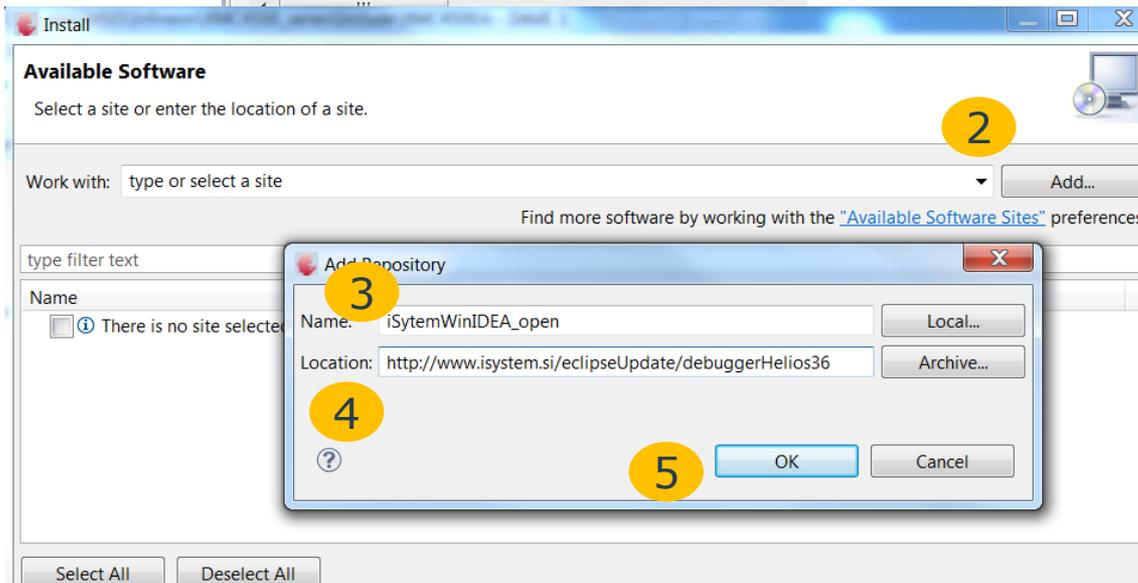
- This tutorial provides a step by step description to
 - Install the winIDEA Open eclipse plug-in into DAVE
 - Create a very simple project for the XMC2 Go kit
 - Generation of a low frequency PWM signal using DAVE Apps to toggle a LED
 - Set up the debug configuration in DAVE to use winIDEA Open with the CDT debug perspective in DAVE
 - Set up the debug configuration in DAVE to use the winIDEA Open in its proprietary IDE started from DAVE
- Prerequisites
 - DAVE v3.1.10 or later should be installed
 - Including the latest version of the DAVE Apps libraries
 - Segger J-Link SW v4.80f or higher should be installed
 - XMC2 Go kit (or any other XMC kit with J-Link OBD)
 - If a different XMC kit is used the device selection (page 8) and the selected output pin for the PWM signal (page 1) may be different
 - If a XMC kit without J-Link OBD is used a separate J-Link debugger e.g. J-Link Lite is required

Installation of the winIDEA Open eclipse plug-in into DAVE (I)



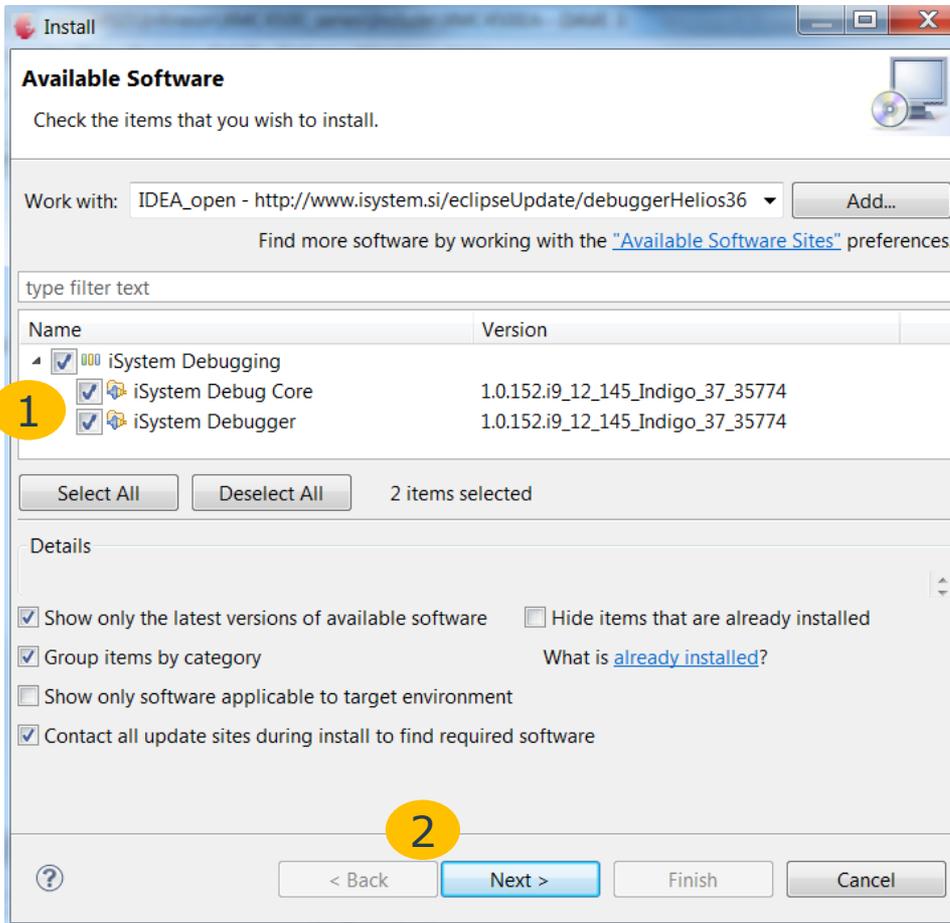
- 1 Press in DAVE:
->Help ->Install New Software.
- 2 Press ->Add in "Work with".
- 3 Enter any name, here,
"iSystemWinIDEAopen" is used.
- 4 Copy in "Location" the below
shown http link.

<http://www.isystem.si/eclipseUpdate/debuggerHelios36>



- 5 Press ->OK

Installation of the winIDEA Open eclipse plug-in into DAVE (II)



1 Select iSystem plug-in.

2 Press ->Next.

Then follow the further instructions, accept the terms of use and restart DAVE after successful installation of the plug-in.

<http://www.isystem.si/eclipseUpdate/debuggerHelios36>

Installation of the winIDEA Open debug engine and IDE

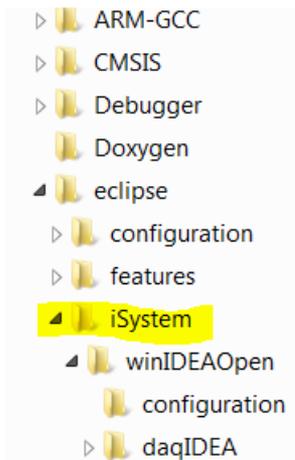
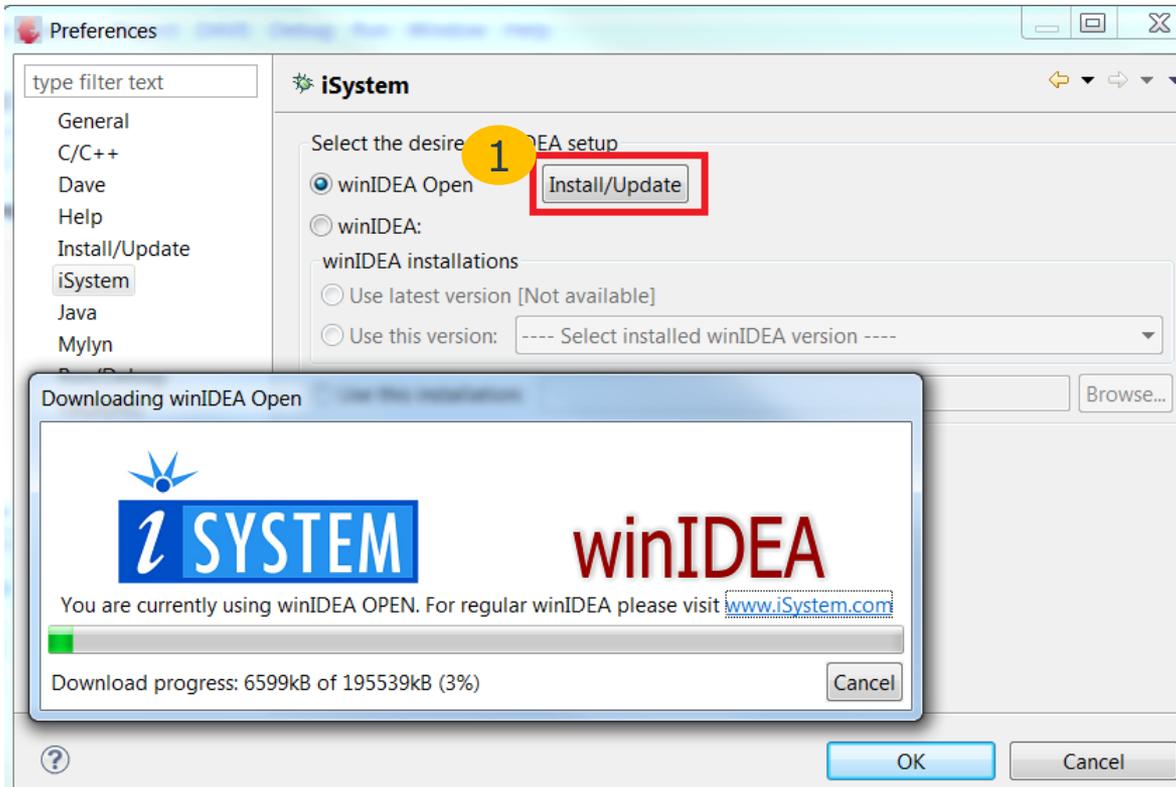
Press in DAVE ->Window ->Preferences ->iSystem
Then the below user interface will appear.

1 Press ->Install/Update.

Then the installation control of winIDEA open will appear.

Follow/ accept subsequent installation prompts.

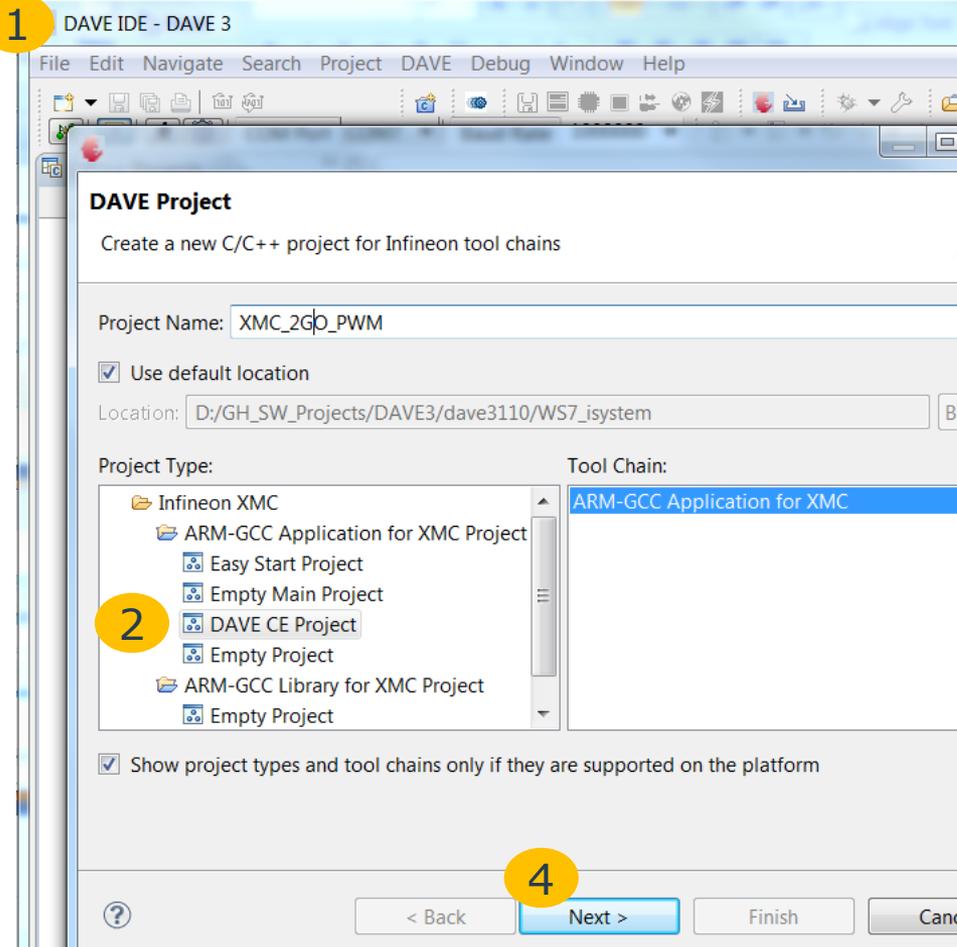
The winIDEA open debugger engine and IDE is then installed in the eclipse folder of the the DAVE installation.



Create a very simple project for the XMC2 Go kit

(I)

We will now generate as simple project using a PWM App to blink one of the LEDs of the XMC 2Go kit with the frequency of the PWM signal.



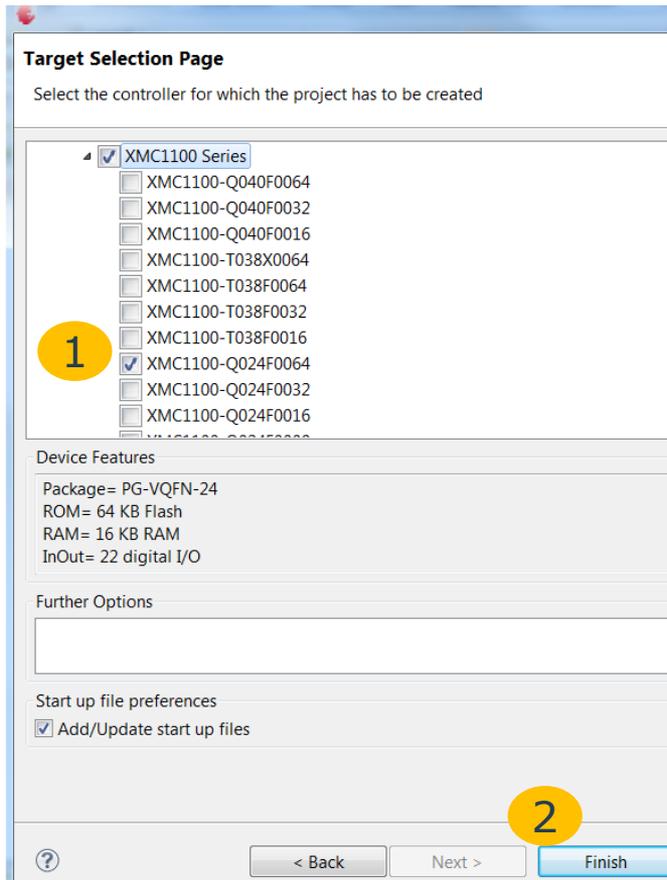
1 Press in DAVE:
-> File ->New ->DAVE Project.

2 Select "DAVE CE Project".

3 Enter a project name, here:
XMC_2GO_PWM

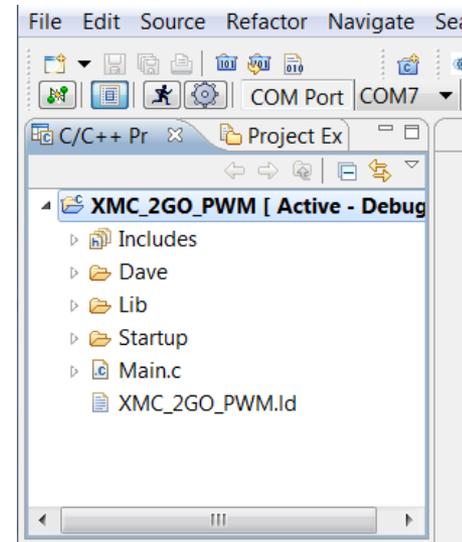
4 Press -> Next.

Create a very simple project for the XMC2 Go kit (II)



1 Select the XMC1100 Q024F0064
If a different board is used select the appropriate target MCU

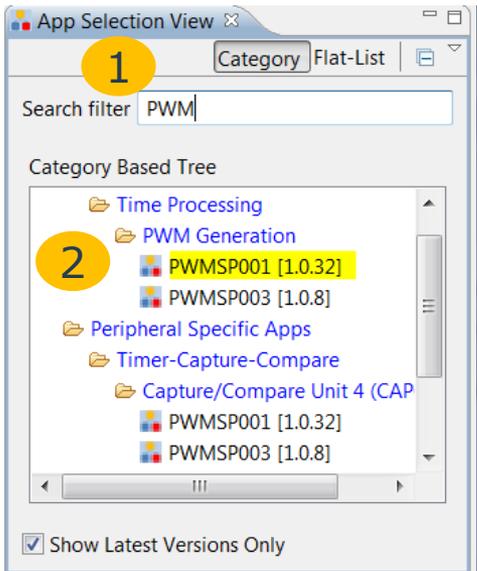
2 Press -> Finish



Finally a new project has been added into the workspace.

The project includes a very simple main.c that calls the DAVE init function, the target specific startup files and linker script file and the stubs for the c library.

Adding and configuring a PWM App to generate the required PWM signal

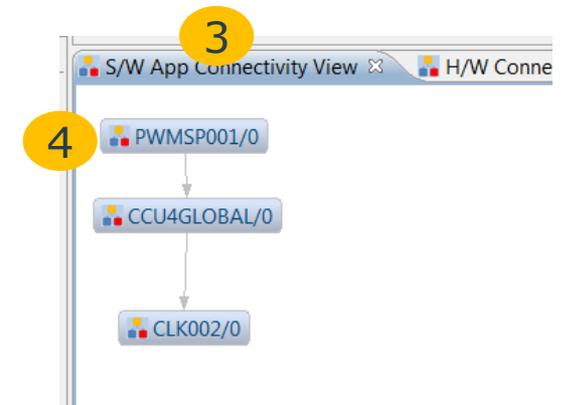
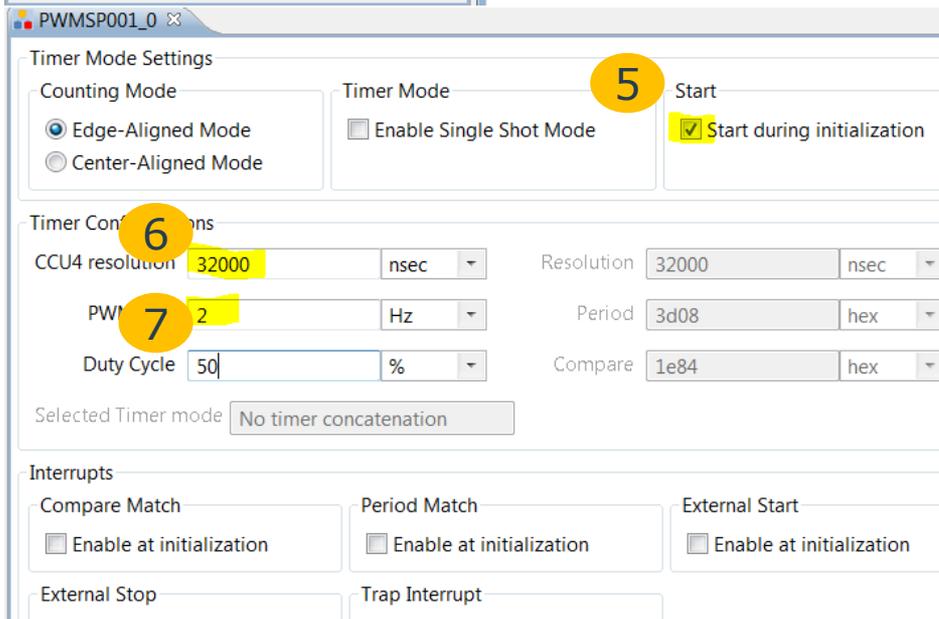


1 Enter in App Selection View of DAVE "PWM".

2 Double click on PWMSP001 to add this DAVE App to the project. Then this App plus additional required Apps can be seen in the S/W App Connectivity View. 3

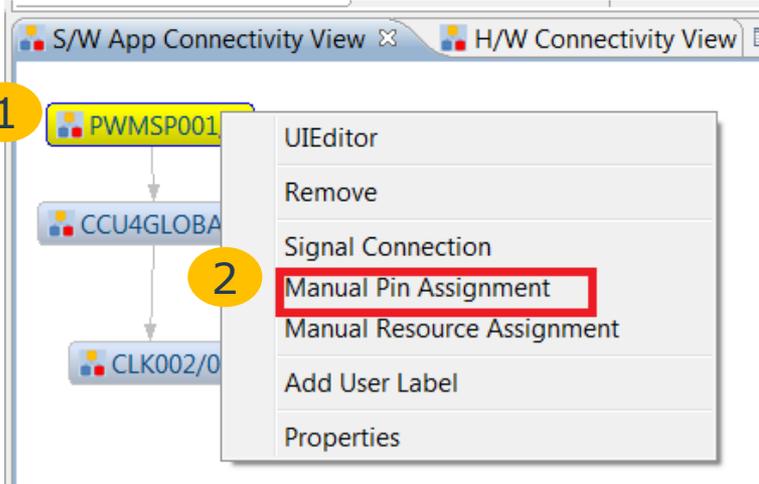
4 Double Click on the PWMSP001 App to open the configuration GUI.

5 6 7 Change the marked settings in the configuration GUI.

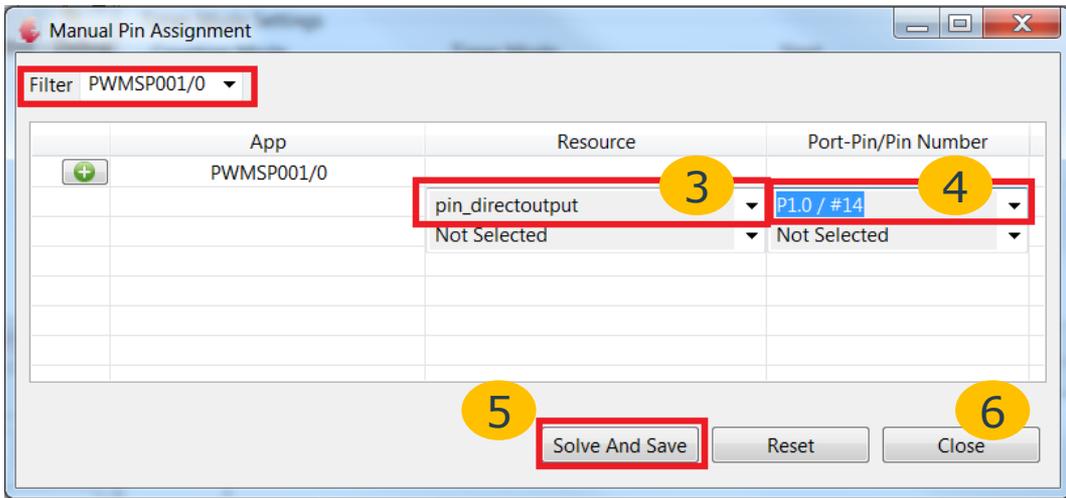


Assign the PWM signal to the required pin

DAVE provides a resource solver to map the appropriate chip resources to the DAVE Apps. To make sure that the PWM signal will be generated at the pin where the LED is connected we need to assign the required pin manually as constrain for the solver.



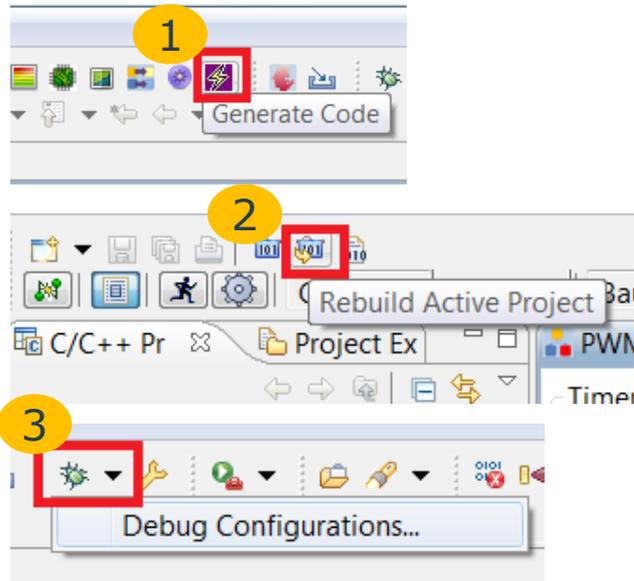
- 1 Right mouse click on the PWMSP001 App.
- 2 Select "Manual Pin Assignment".
- 3 Select "pin_directoutput".
- 4 Select "P1.0/#14".
- 5 Press ->Solve And Save.



If a different board than the XMC2 Go kit is used a different pin has to be assigned.

- 6 Press ->Close.

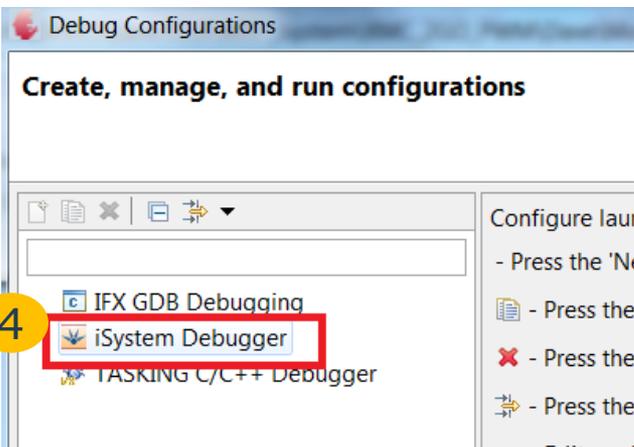
Generation of the PWM library, compilation, and creating a new debug configuration



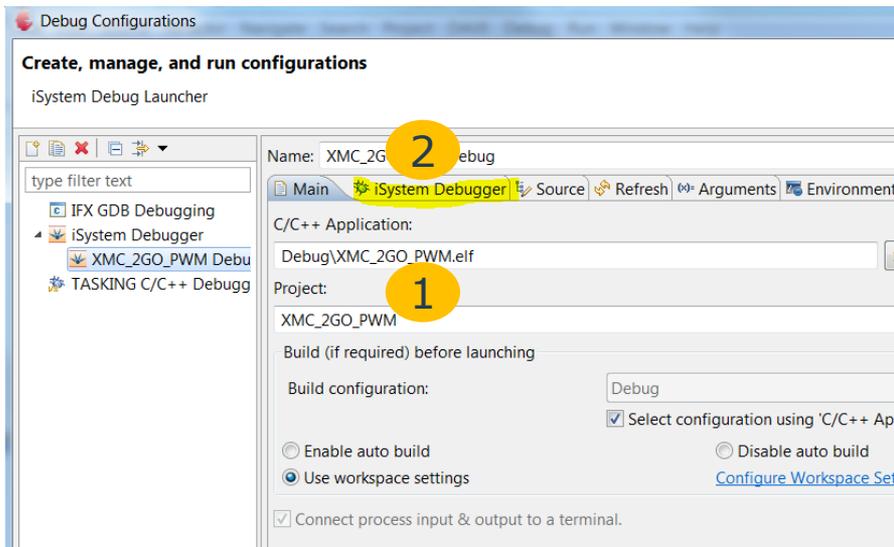
- 1 Press "Generate Code".
- 2 Press "Rebuild Active Project"
Click on the project name in the workspace.
- 3 Press the combo box of "Debug Configuration".
- 4 Double Click on "iSystem Debugger" to create a new debug configuration.

Notes:

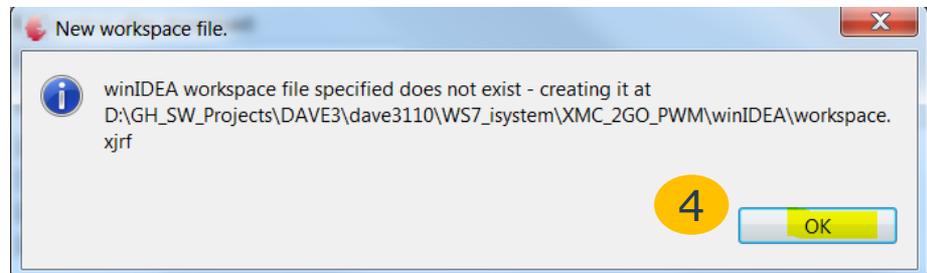
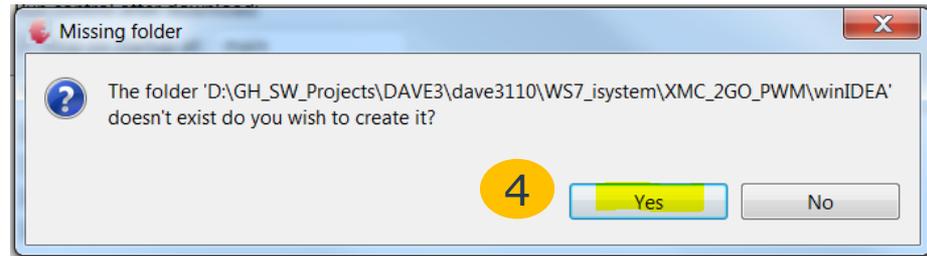
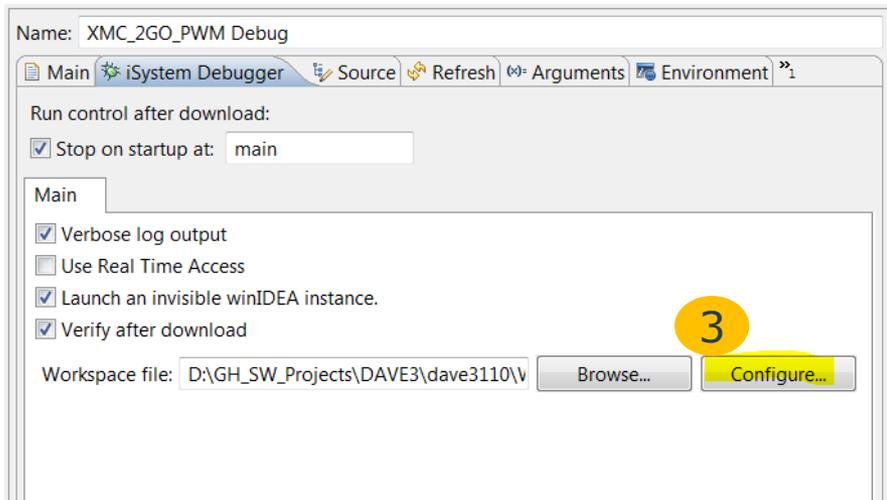
- We don't need to add any user code to call any PWM library function because main.c contains already a call of the DAVE init function which start the PWM signal with 2 Hz according to our configuration (page 9).
- Before we execute step3 it is important to select the project (click on the project name), otherwise the debug configuration will not be created with the correct project information (winIDEA does not use the active project functionality to determine the project).



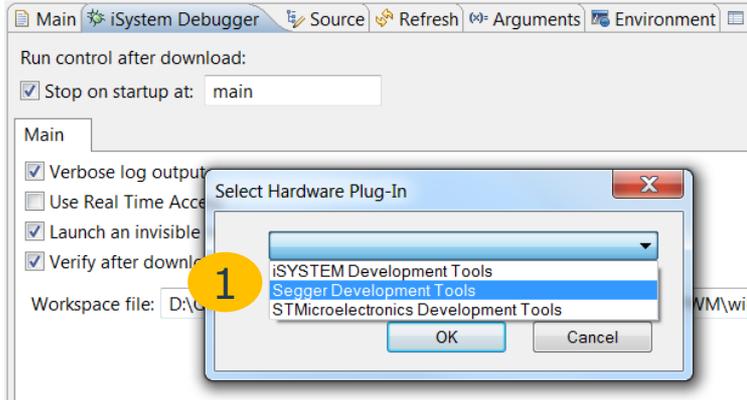
Correct setup of the debug configuration



- 1 The debug configuration should contain the correct project name and elf file.
- 2 Switch to the "iSystem Debugger" tab.
- 3 Press ->Configure.
- 4 Confirm creation a win IDEA workspace folder and workspace file in the DAVE project.

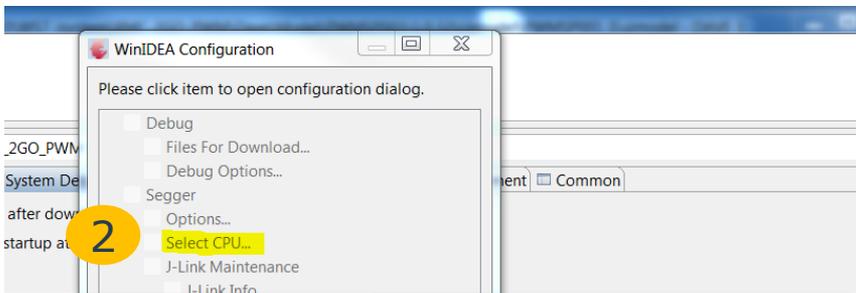


Selecting the MCU target



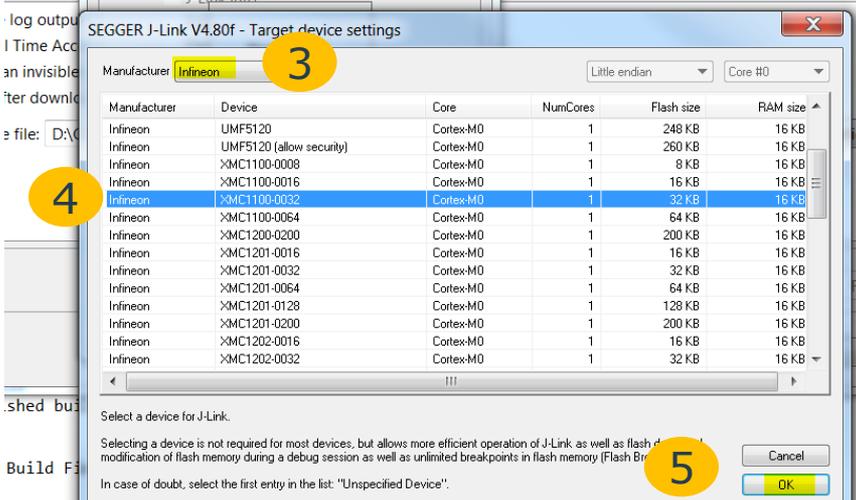
winIDEA Open doesn't take the selected device from the DAVE project, therefore the MCU target has to be assigned once more.

After the winIDEA Open workspace file has been created the control to "Select Hardware Plug-In" will appear.



1 Select "Segger Development Tools".

2 Press ->Select CPU...

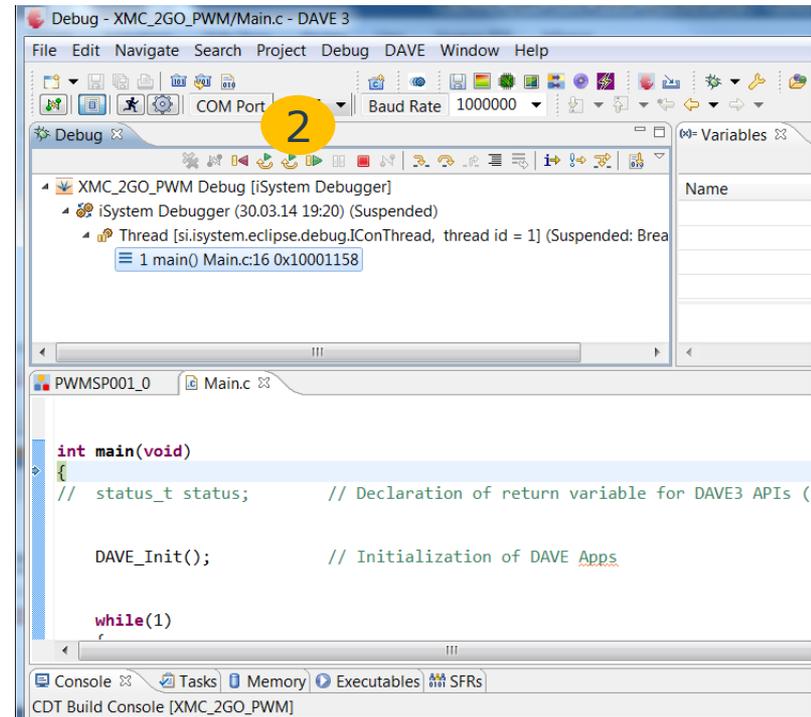
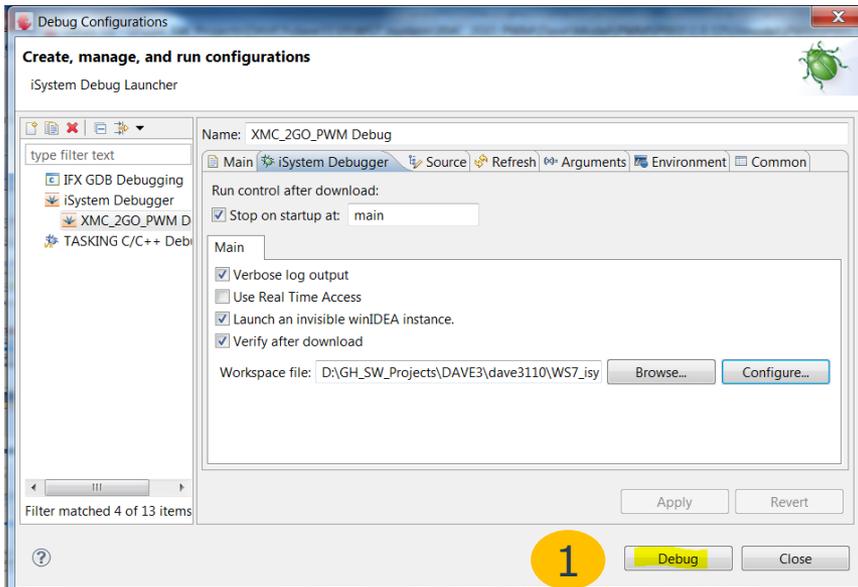


3 Select "Infineon" in Manufacturer combo box.

4 Select "XMC1100-0064".

5 Press ->OK.

Start the debug session



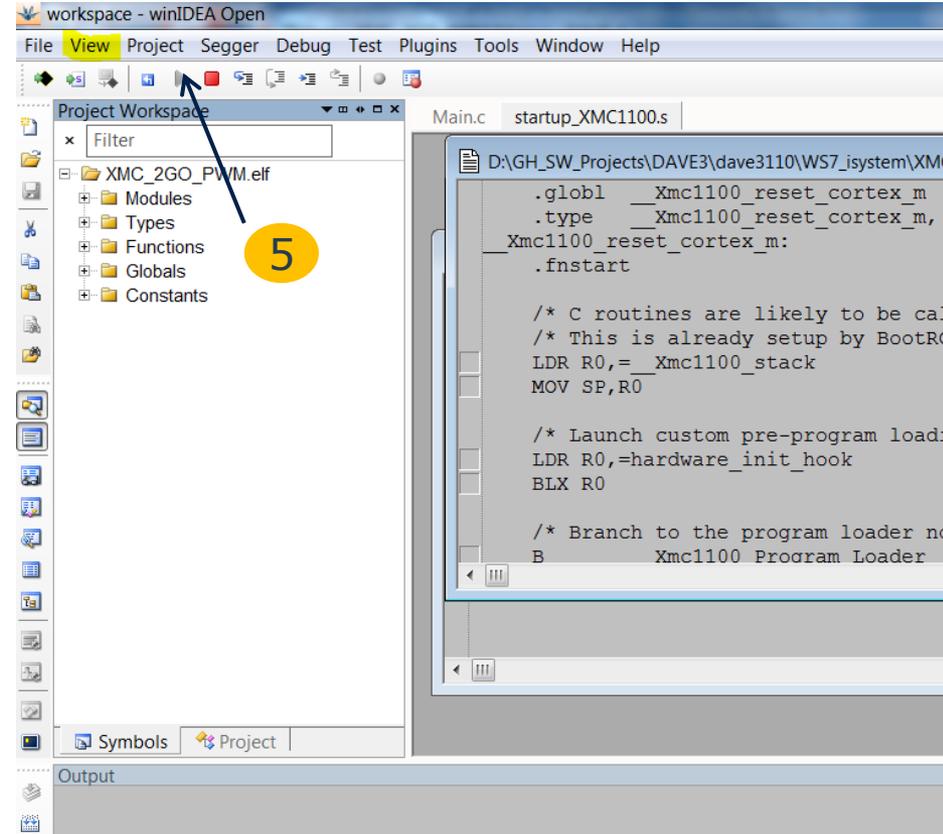
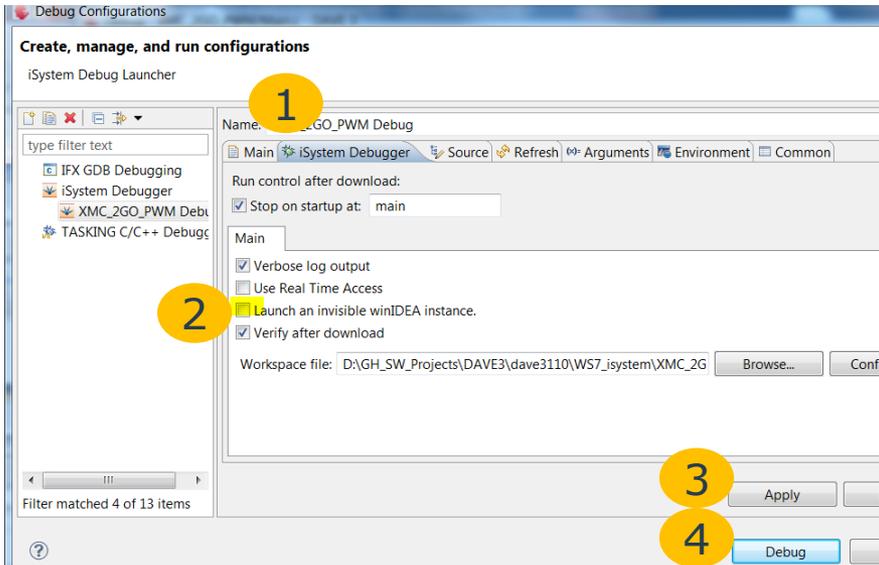
1 Press ->Debug in the fully configured debug configuration to start the debug session.

2 Press ->Resume in the debug perspective to run the program.

Now a build will be started, the image will be downloaded to the target and a message to change to the debug perspective will be prompted. After confirmation, the debug perspective will open.

Using winIDEA open IDE as debugger instead of the eclipse debug perspective

The winIDEA Open IDE provides more comprehensive debug features than implemented in the eclipse CDT debug perspective. Therefore it might make sense to use this debug user interface instead of the eclipse perspective.

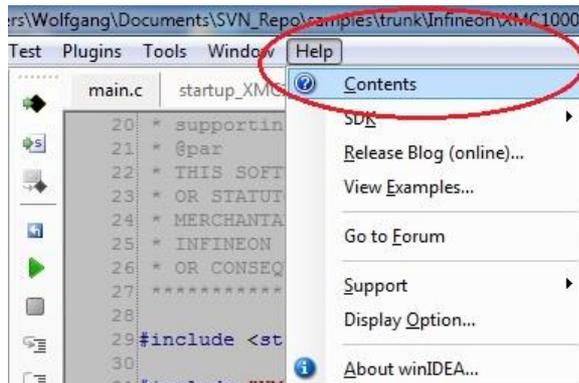


- 1 Open in DAVE the debug configuration in the iSYSTEM Debugger tab.
- 2 Uncheck "Launch an invisible winIDEA instance".
- 3 Press ->Apply.
- 4 Press ->Debug to start a debug session.

- 5 User interface of winIDEA Open as separate IDE. Press the "Run" button to start the program. If there is no run button the debug controls have to be enabled: ->View ->Toolbars ->Debug.

Further information

- iSystem-EclipseDebugPlugin-UsersGuide.pdf
 - <http://www.isystem.com/download/eclipse>
- iSystem-winIDEA Open extensive Help Menu
 - Accessible in the IDE of winIDEA Open





ENERGY EFFICIENCY MOBILITY SECURITY

Innovative semiconductor solutions for energy efficiency, mobility and security.

