



University of Colorado
Boulder

Principles of Embedded Software

Debug Guide

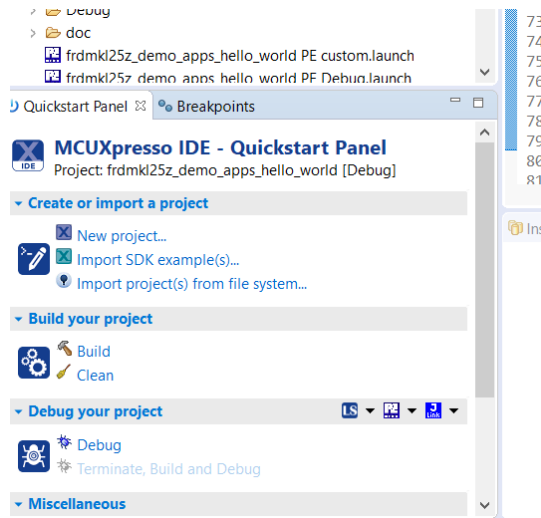
Created by:

Rakesh Kumar

Date: 09/15/2020

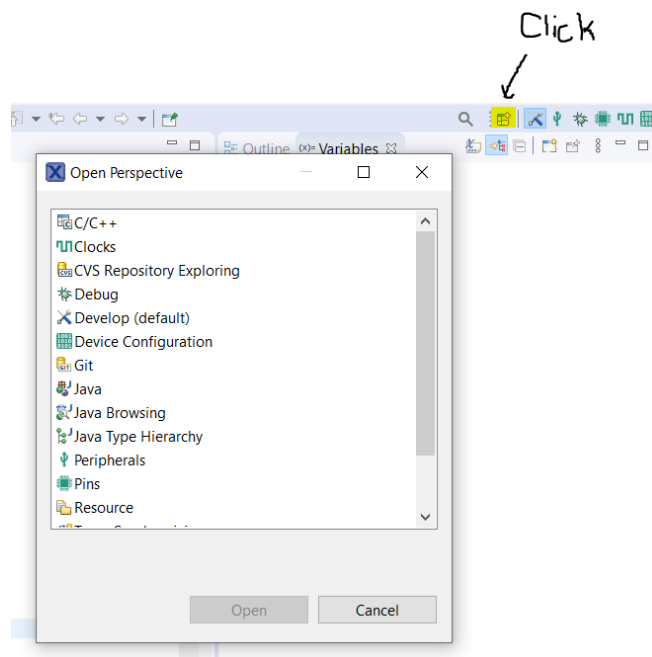
Pointer's on IDE usage

1. **QuickStart Panel** is your go-to guide whenever you feel lost be it for a setting, importing an SDK sample, creating a new project, etc.

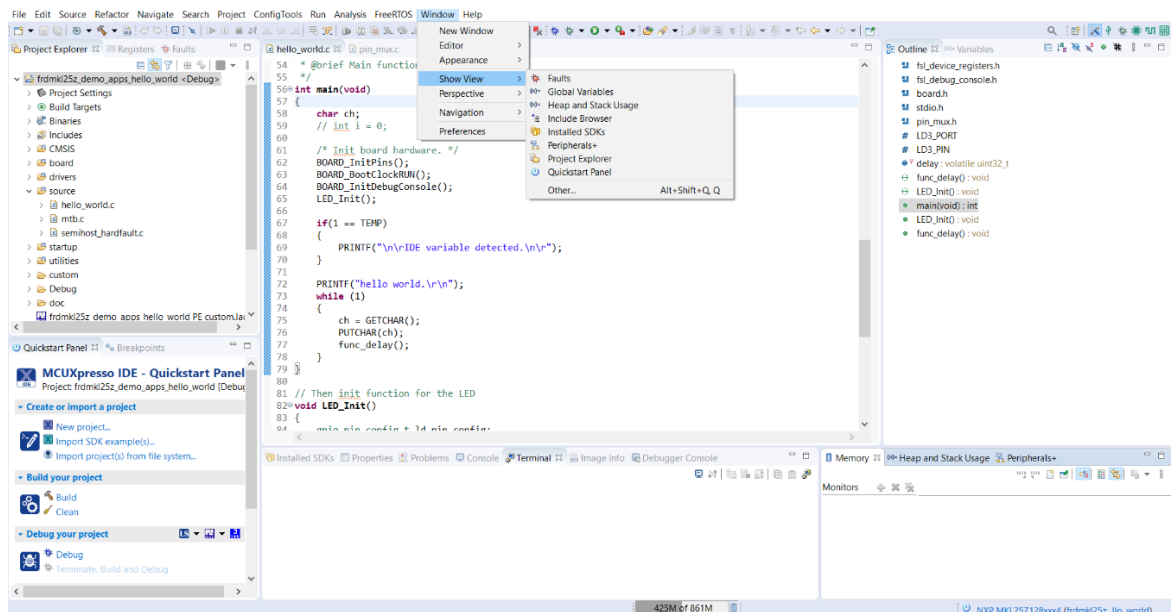


This window is present typically in the bottom left. If not already present, navigate through the Window tab on the top of the IDE, to the "Show View" and locate the QuickStart panel.

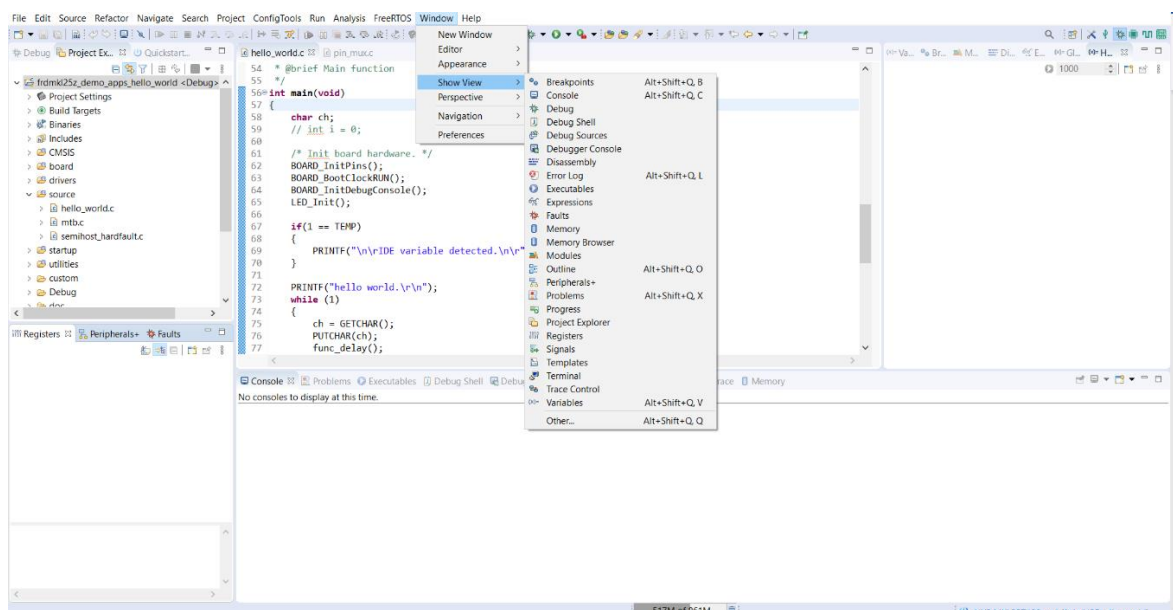
2. **Perspective Windows** are literally what their name suggests. The most common perspectives you would be venturing into are Debug, Develop, Pins and Peripherals.



- There are certain windows/views that show up only in certain perspectives. The below snaps will convey this much more clearly:



The above is pertaining to the Debug window



And the above is the options available when you are in the Debug perspective.

4. Breakpoints and related buttons



Starting from the left-most end we have the **blue** button, which resembles a striked-out breakpoint. It is pretty much what it symbolically means. This button would disable all the breakpoints that currently present in your code.

The **green** button is the 'resume' button and would resume the execution of the running code, the **pause** button besides it would pause the execution the running code and can be resumed by the resume button mentioned as mentioned earlier.

Next, we have the 'Terminate' button, the **red** square which just abruptly terminates the execution of the application.

The red zig-zag icon placed just besides the terminate icon is the 'Debug Disconnect'. This button disconnects the debugger from your board and ends the running-debugger mode

The next are the debug **stepping icons** namely 'step into', 'step over' and 'step return'. They basically aid you in efficiently debugging your code. Each option has its own significance and you will get more comfortable with it as and when during debugging you feel the need for each scenario.

5. Another important tab useful for development and debugging is the peripheral and the registers tabs

The screenshot shows the STM32CubeIDE interface. The top panel has tabs for Outline, Variables, and Registers. The Registers tab is active, displaying a table of registers. The bottom panel has tabs for Stack Usage and Peripherals+. The Peripherals+ tab is active, showing a tree view of peripherals for the MKL25Z128xxx4 device, with the GPIO register set expanded.

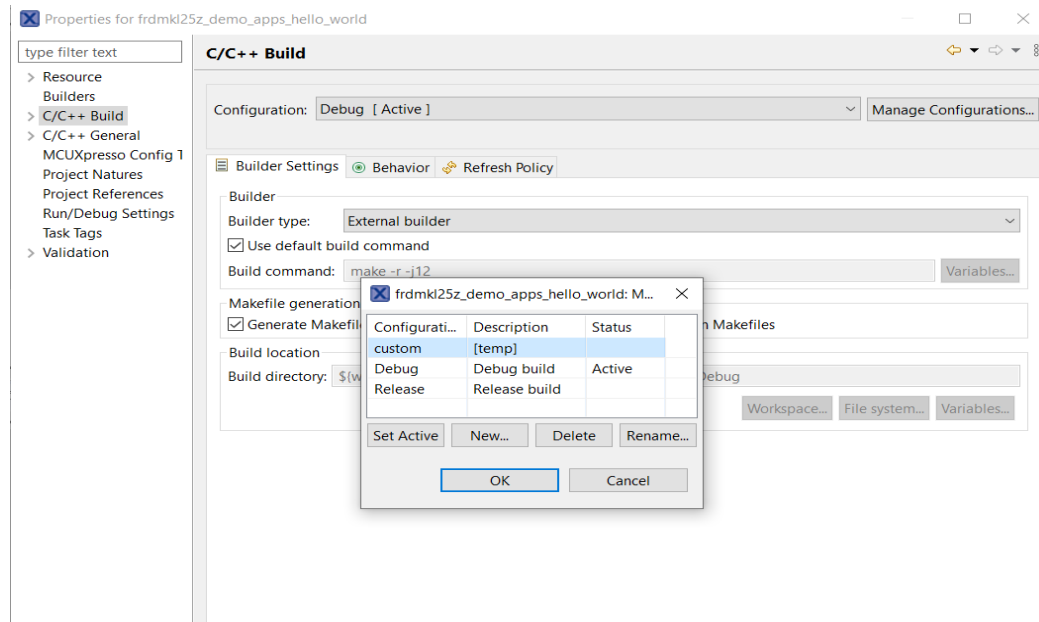
Name	Value	Description
r0	0x1fff170	Argument/Scratch...
r1	0x3	Argument/Scratch...
r2	0xd4	Argument/Scratch...
r3	0x1	Argument/Scratch...
r4	0xd4	Variable Register 1
r5	0x40020000	Variable Register 2
r6	0xffff	Variable Register 3
r7	0x20002fe0	Variable Register 4
r8	0xbfdfff7b	Variable Register 5
r9	0xffdffffe	Variable Register 6
r10	0xffdffffe	Variable Register 7
r11	0xfebfbff	Variable Register 8
r12	0x2	Intra-Procedure-C...
sp	0x20002fe0	Stack Pointer (r13)
lr	0x2467 <_main+...	Link Register (r14)

Register	Address	Value
GPIO	0x400ff0c0	
GPIO-PDOR	0x400ff0c0	0x0
GPIO-PSOR	0x400ff0c4	<writeonly>
GPIO-PCOR	0x400ff0c8	<writeonly>
GPIO-PTOR	0x400ff0cc	<writeonly>
GPIO-PDIR	0x400ff0d0	0x0
GPIO-PDDR	0x400ff0d4	0x0

Note that there is legible colour change that is done by the IDE to indicate the changes in the same.

6. Adding build targets

This can be done by navigating as below from the project properties tab:



Note you can clone the settings from any other existing configurations and then later go ahead on customizing it as desired.

7. Bonus hint for running **objdump**:

On my Windows system the following command successfully displays the dump when run on PowerShell:

.\arm-none-eabi-objdump.exe -S

C:\Users\rakma\Documents\MCUXpressoIDE_11.2.0_4120\workspace\frdmkl25z_demo_apps_hello_world\Debug\frdmkl25z_demo_apps_hello_world.axf

You can very well dump this information to a file by appending “> 'filename'” to the above command.

Do explore into various options of the objdump.

In the same manner **arm-none-eabi-gcc** and **arm-none-eabi-gdb** could be run.

So, that is all to it currently for this document, reach out to us for any concerns and doubts.