

# How to start customized KSDK project based on KSDK demo code

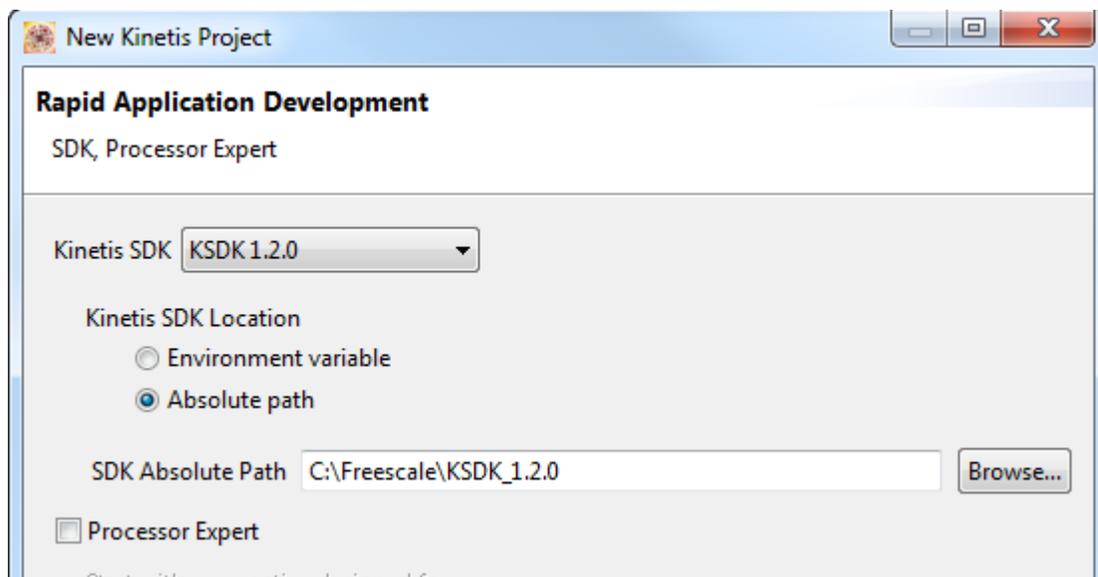
---

KDS v3.0 ( <https://community.freescale.com/docs/DOC-104880> ) and KSDK v1.2.0 ( <https://community.freescale.com/docs/DOC-104887> ) have been released for several days. If user project is expected to use the Kinetis SDK, as same as previous KDS and KSDK version, we must apply the Eclipse Update for Kinetis SDK into this tool using Help -> Install New Software. Then go to ..\KSDK\_1.2.0\tools\eclipse\_update folder to find the appropriate Eclipse update.

Recently, I had several customers query a same question about how to start their own KSDK project. I think we can do it by either using KDS project wizard or basing on a KSDK demo project under KSDK3.0 install folder.

## 1. Start customized project with wizard.

If using KDS project wizard, we just need make sure we select KSDK 1.2.0 during creating a new project.



Then follow the normal procedure to a new KSDK project.

Because this method is not the focus of this article, we will not talk this in detail.

## 2. Start customized project based on KSDK demo code

If you are beginner who is not very familiar KSDK structure, I recommend you start your own project based on KSDK demo code. There are rich amount of demo codes under KSDK 1.2.0 install. I itemize them as bellow:

- `..\KSDK_1.2.0\examples` : demos for app and common peripheral drivers
- `..\KSDK_1.2.0\middleware\filesystem\mfs\examples`: demos for file system
- `..\KSDK_1.2.0\middleware\tcpip\rtcs\examples` : demos for tcpip
- `..\KSDK_1.2.0\rtos\mqx\mqx\examples` : demos for MQX+KSDK
- `..\KSDK_1.2.0\usb\usb_core` : demos for usb device, host, otg usage

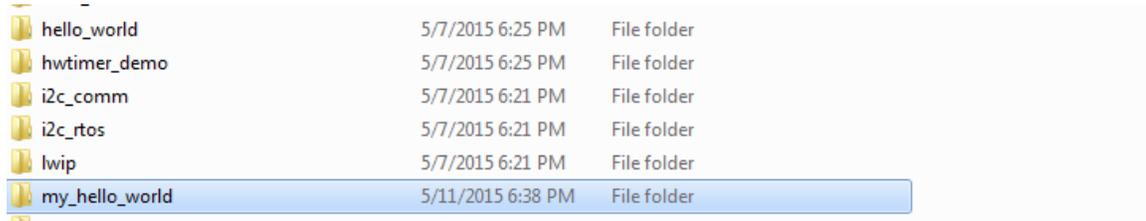
User can choose one sample project that closest to the requirement, and then modify it to customized project.

There are two cases for modification based on customized project path: set user project under the same demo code path and set user project under arbitrary user path.

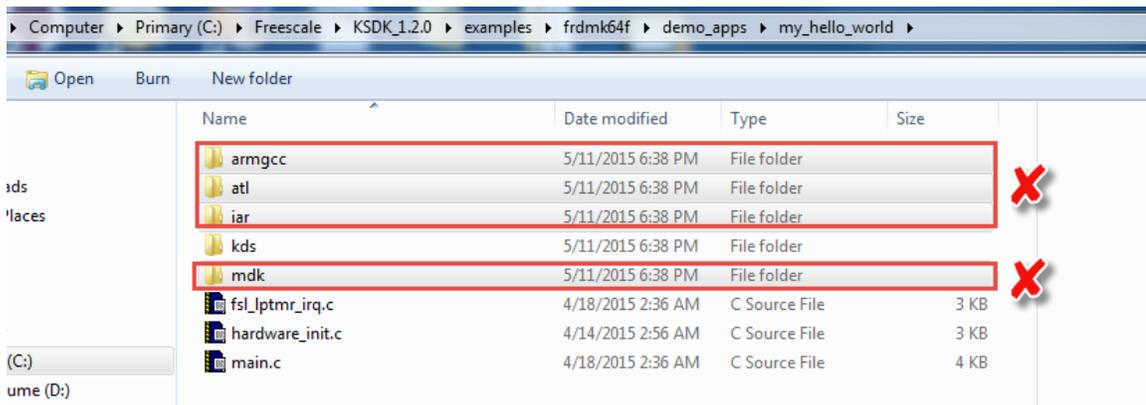
### 2.1 set user project under the same demo code path

let's discuss it with example: create a customized project based on hello\_world under ..\KSDK\_1.2.0\examples\frdmk64f\demo\_apps.

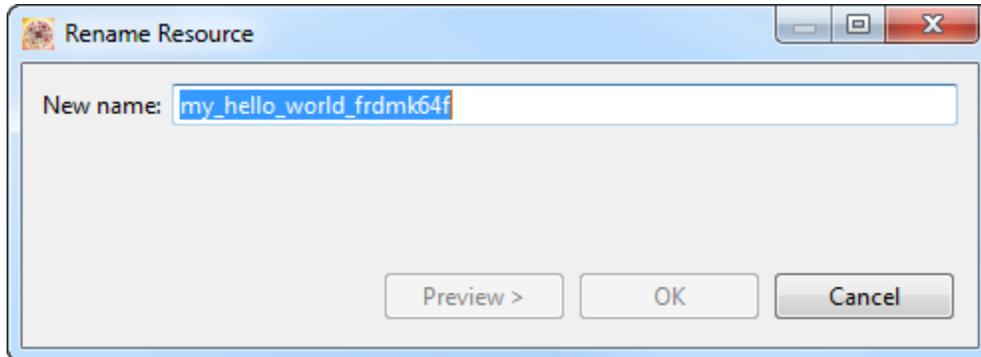
- Go to `..\KSDK_1.2.0\examples\frdmk64f\demo_apps`. Copy hello\_world folder to the same path, and rename it as my\_hello\_world



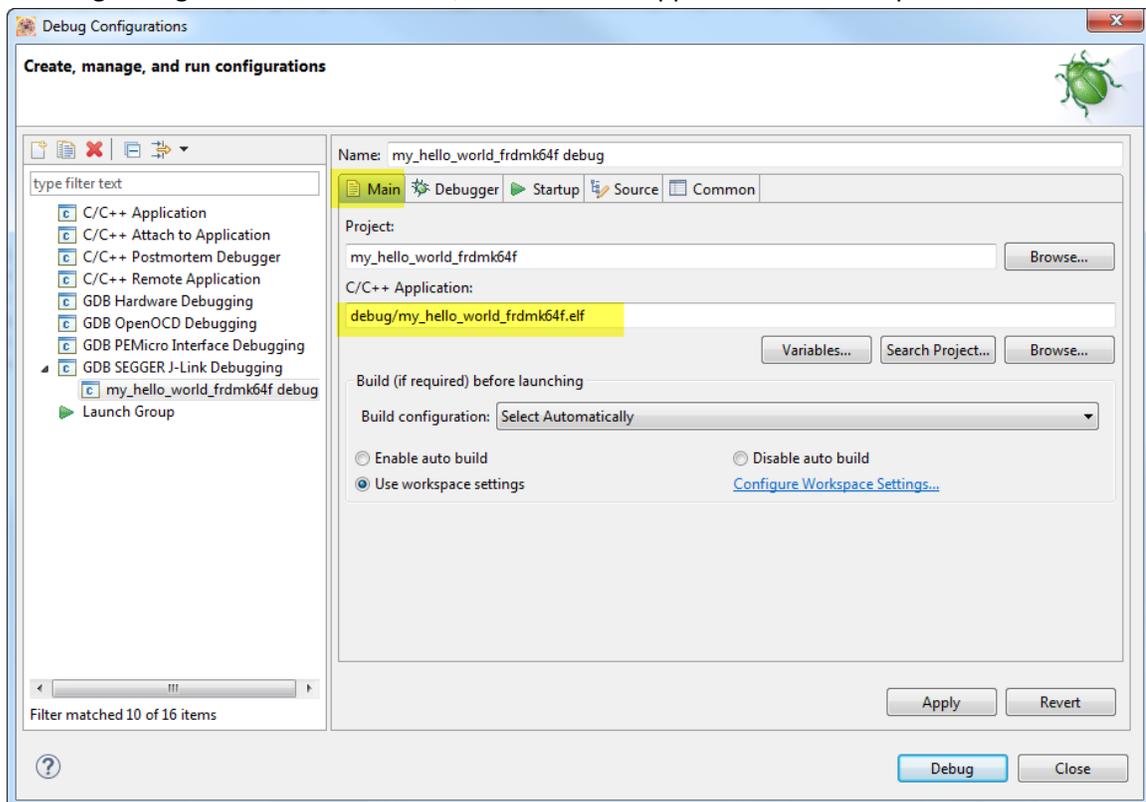
- We only work with kds project, so remove other armgcc,atl,iar,mdk folders.



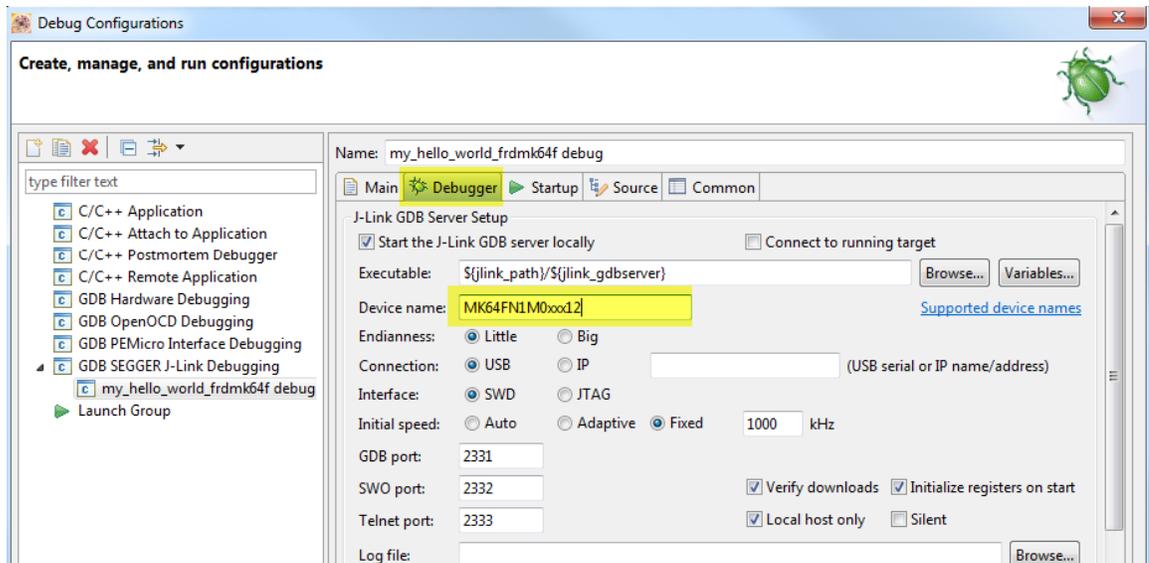
- c. Then open this project `..\KSDK_1.2.0\examples\frdmk64f\demo_apps\my_hello_world` with KDS 3.0. right click the project name in “Project Explorer” window, choose “Rename...”, change the project name as `my_hello_world_frdmk64f`



- d. Build the project
- e. In debug Configurations main window, make sure the application name is updated:



- f. In debug configuration Debugger window, make sure the correct Device name is filled:

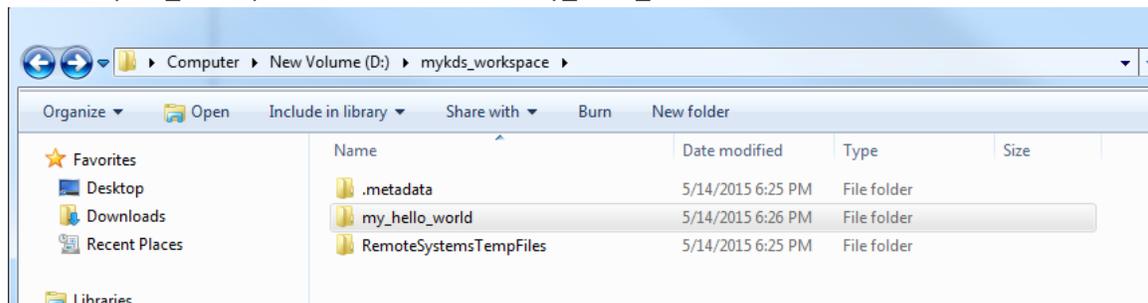


- g. Then apply all the changes, build the project, we can from on this customized project as our start.

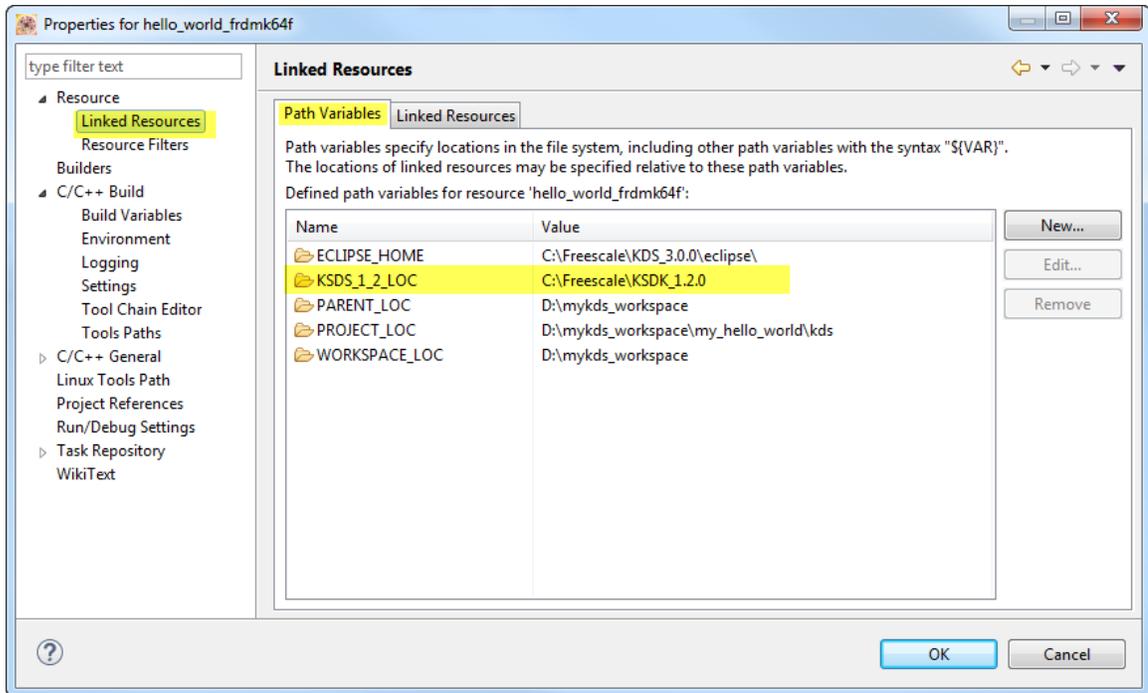
## 2.2 set user project under arbitrary user path

We also discuss it with example: create a customized project under an arbitrary user path: D:\mykds\_workspace.

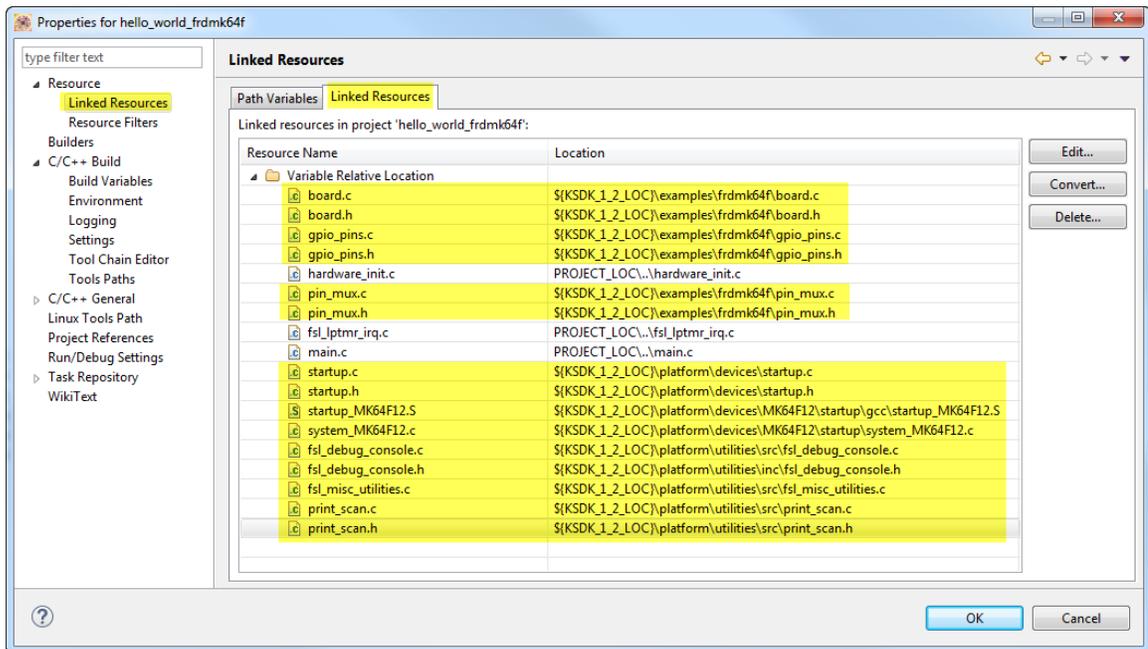
- a. Go to ..\KSDK\_1.2.0\examples\frdmk64f\demo\_apps. Copy hello\_world folder to D:\mykds\_workspace, and rename it as my\_hello\_world



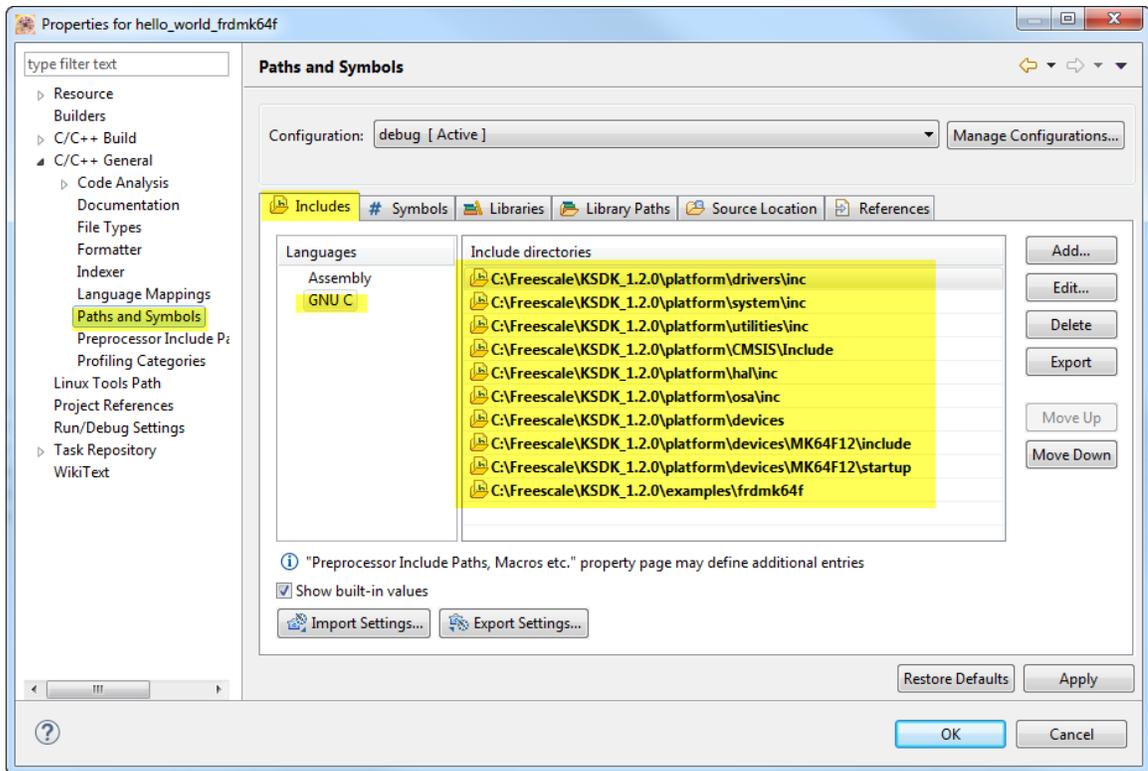
- b - f. Exact same as steps b-f in previous 2.1 set user project under the same demo code path
- g. Add KSDK 1.2.0 install path variable name as KSDK\_1\_2\_LOC



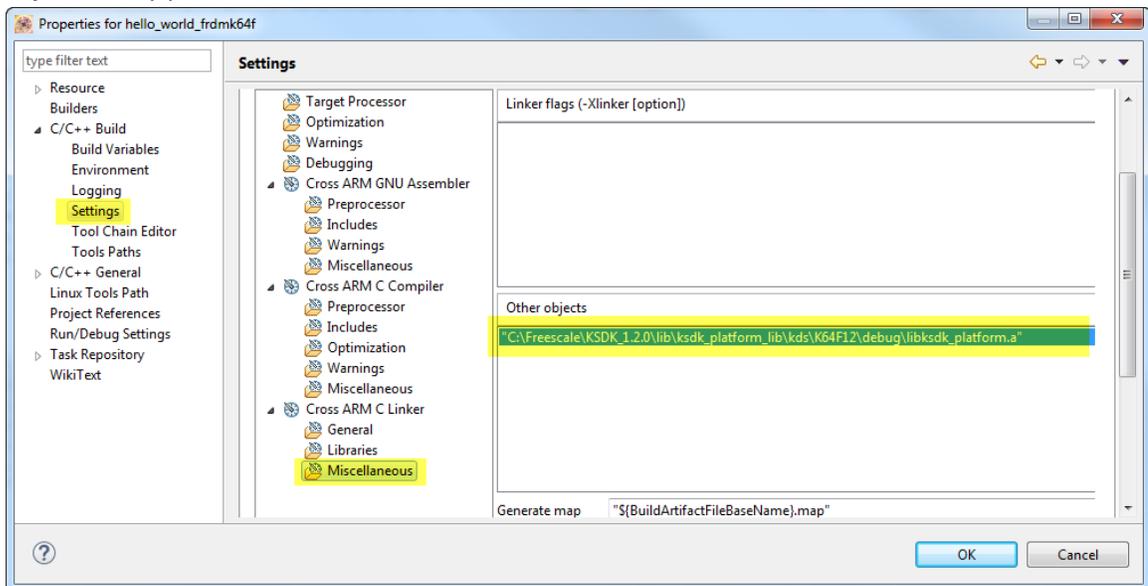
- h. The original sample code link resources paths relies on the demo code path. We need modify it to rely on KSDK install path. See is the new location setting after modification:



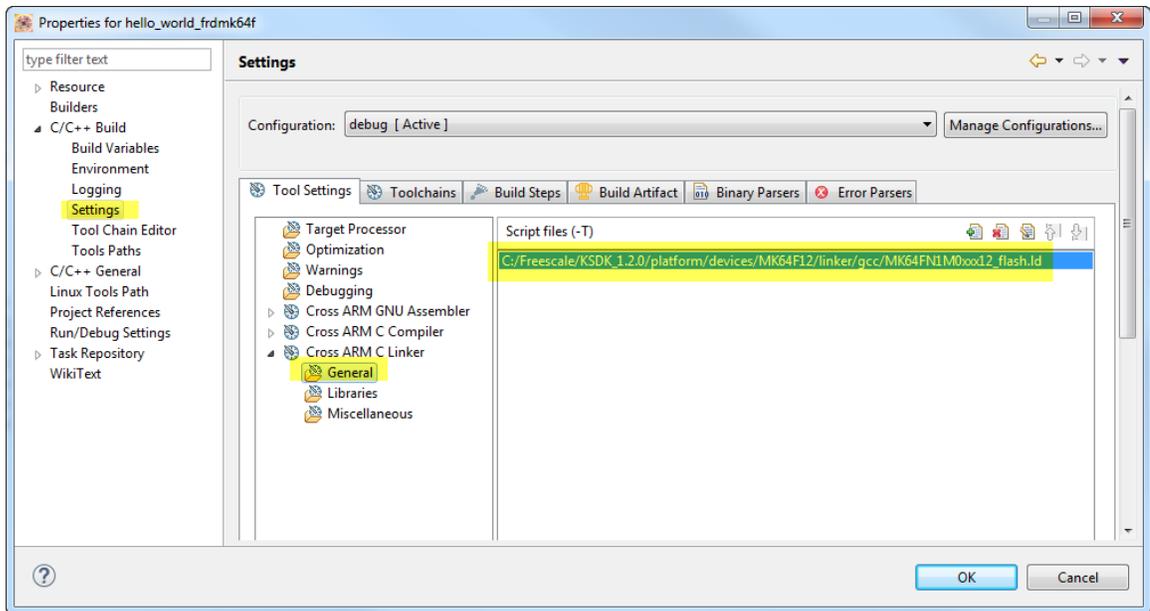
- i. Also, path and Symbols need to be refreshed to:



j. Adjust library path location:



k. Adjust linker file path location:



- I. Then apply all the changes, build the project, we can use this customized project under arbitrary user path as our start.

Enjoy KSDK programming 😊