**Introduction**

This document explains briefly how to create an interface through Ethernet.

**Resources**

PC running Windows 7 with the following software:

• Kinetis Design Studio (KDS) v2.0.0

• MQX for KSDK v1.1.0

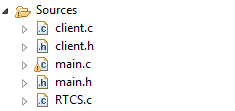
Hardware:

• FRDM-K64F

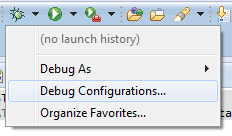
1. **Implementation**
   1. By now a ‘*New MQX RTOS for KSDK Project Wizard’* does not exists. For this Lab use **‘MQX for KSDK Lab 4 - Base Project’**.

**Note:** If you want to know how to create a base project using RTCS see the **‘Appendix’** in this document.

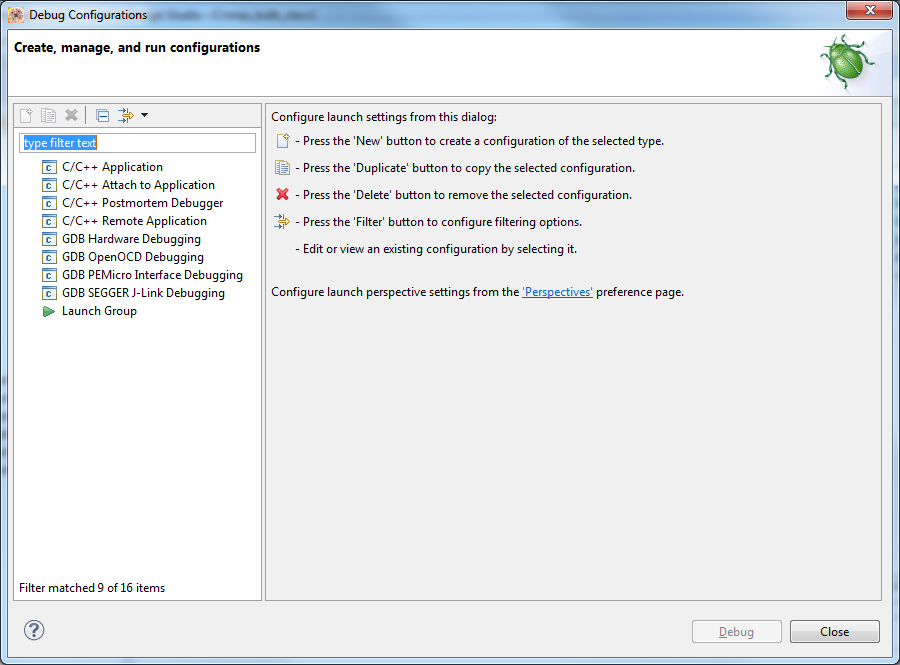
* 1. Copy all the files from **‘MQX RTOS for KSDK TCP Client’** folder to the **‘Sources’** folder in your project. After copying the files, **'Sources'** folder should look like this:



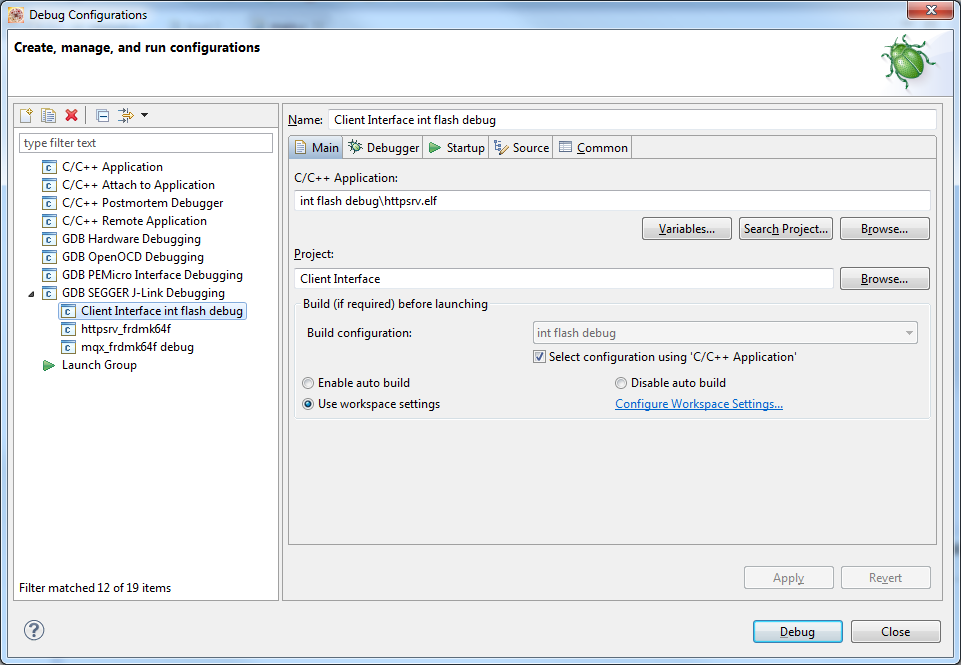
* 1. Build your application, go to **menu Project > Build Project.** Alternately click the hammer button.
  2. Debug your application, go to ***Debug Configurations.***



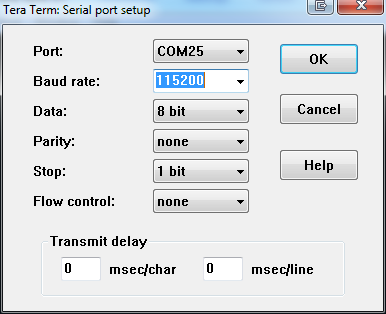
* 1. Double click on GDB Segger J-Link Debugging.



* 1. Click on **Debug**.



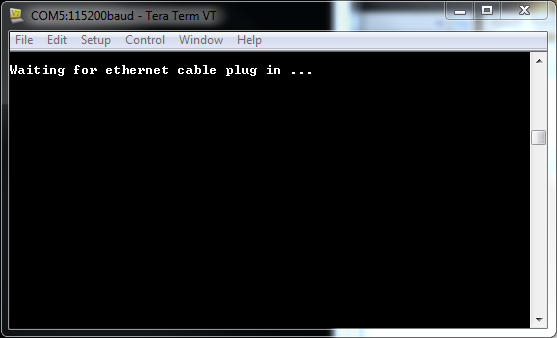
* 1. Open a terminal, select the appropriate port and set baud rate to.



* 1. Click **Resume**.Alternately press F8.

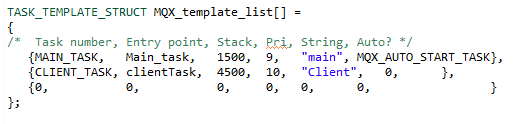


* 1. After running your project and configuring the terminal, you should see the following message on the terminal (Without the Ethernet cable connected).



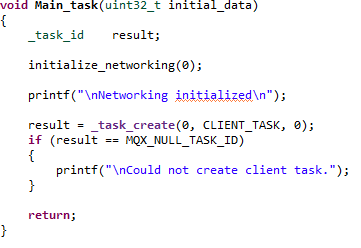
1. **How does it work?**

#### 2.1. Open **‘main.c’** file of your project, as you can see there are 2 tasks “main” and “Client”. Main task is "Auto Start", this means that it will be executed automatically when the project starts.



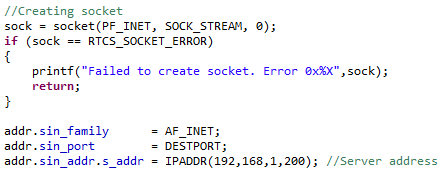
#### 2.2. Main\_task

This task initializes the RTCS and all the networking settings. If the input parameter is 0 a static IP is assigned to the Kinetis device and if it’s 1, it assigns a dynamic IP using DHCP. Then it creates a "Client" task which is not executed until "Main Task" finishes as Client task has a lower priority.

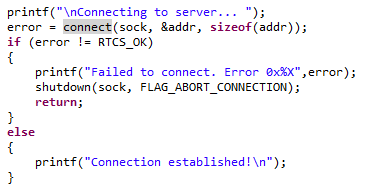


#### 2.3. clientTask

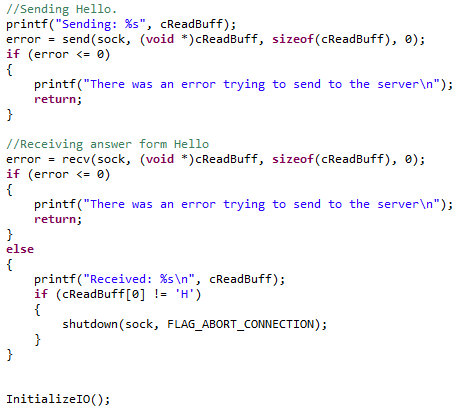
When the “Client” task is created, it first creates a socket for communication between the Kinetis devices, and it sets the socket parameters; type, port number and server address which in this case is: 192, 168, 1, 200.



The connect function initiates a connection from the socket with file descriptor socket to the socket whose address is specified by the addr and length arguments. (This socket is typically on another machine, and it must be already set up as a server.)

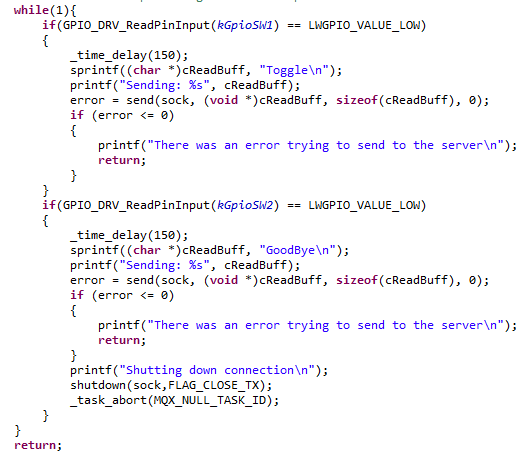


Once the connection is established a "Hello" message is sent to the server, then it waits form an answer from the server. After the communication is tested function "InitializeIO" is called to initialize the onboard LEDs and buttons.



Finally the application enters an endless loop.

* If button 1 is pressed the message "Toggle" is sent to the server.
* If button 2 is pressed the message "GoodBye" is sent.



1. **Appendix**

To create a new project you will have to copy/paste the httpsrv\_frdmk64f located in

***C:\Freescale\KSDK\_1.1.0\tcpip\rtcs\examples\httpsrv***

into your workspace and rename it. When you copy/paste a project into your workspace, your sources folder is linked to the original one, so if you change anything on your new project it will be changed in the original one too. To avoid this problem you will have to erase the sources folder of your new project and create a new one with the same name (this one will not be linked to the previous project).

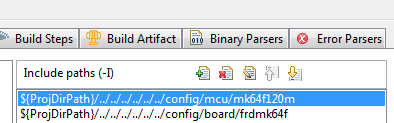
* 1. If your workspace is not the same that the hello\_frdmk64f project you will have to modify the paths of the project. To do this you will have to go to **Project->Properties->C/C++ Build ->Settings**
  2. You will have to change the every single path of the program, to do this you will have to look for the right path the program is referring. For example, in the

**Cross ARM GNU Assembler -> Includes** the first path will be **${ProjDirPath}/../../../../../../config/mcu/mk64f120m**

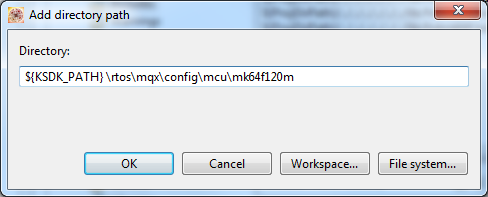
You will have to look into the installation folder **(C:\Freescale\KSDK\_1.1.0)** the last directories of this path. In this case you will find it at

**C:\Freescale\KSDK\_1.1.0\rtos\mqx\config\mcu\mk64f120m**.

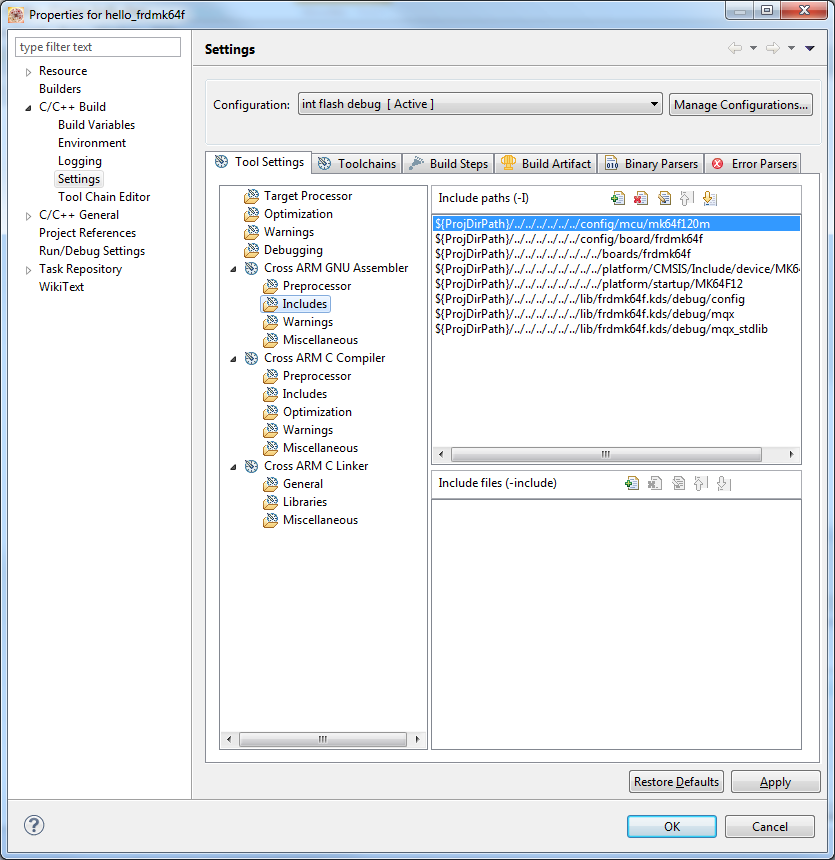
* 1. Once you get the correct path you will have to click the add option.



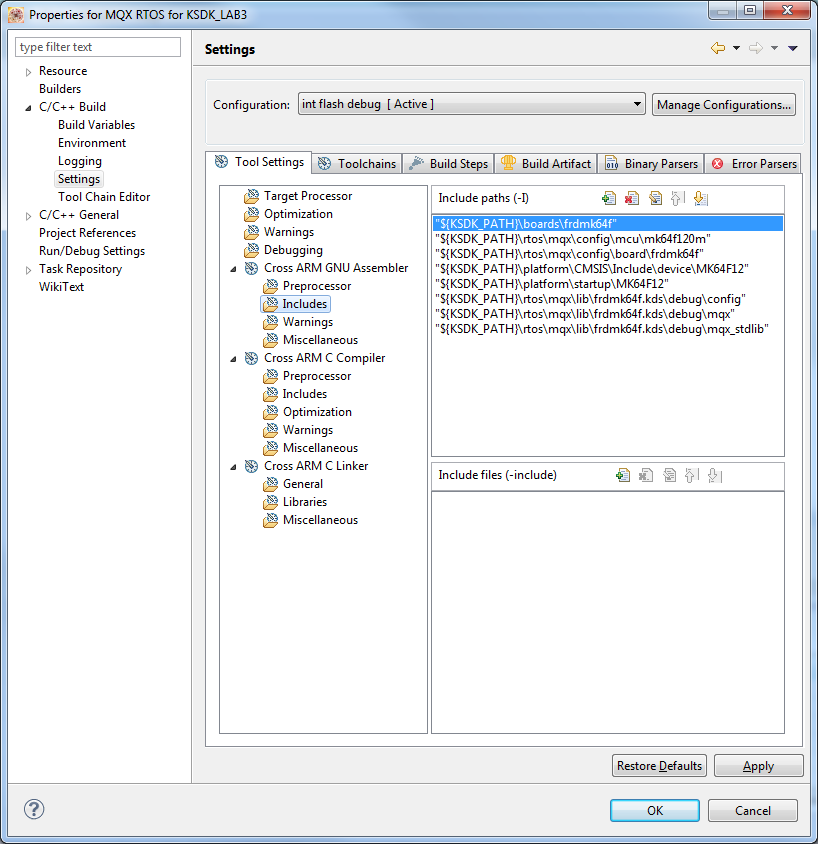
* 1. This window will appear, here you will have to add the correct path and substitute the **C:\Freescale\KSDK\_1.1.0** with **${KSDK\_PATH}** and then click ok.(Note: is important to have defined KSDK\_PATH as an environment variable)



* 1. Once you have added this new path you can erase the other one. You will have to repeat this steps in the next folders.



* 1. Your new paths will have to see as follows.



* 1. Once you change every single path you will be able to compile and debug your new project.
  2. Is suggested erase your actual source folder and create a new one with the same name. If you don’t do this your source folder will be linked to the original one and all your modifications will be made in that folder too.

All this process is necessary because the MQX RTOS for KDS wizard is still in development.