FRDM-K64F_OS3-TCPIP-HTTPs-DHCPc-KSDK-LIB

Download Link	Micrium_FRDM-K64F_OS3-TCPIP-HTTPs-DHCPc-KSDK-LIB.zip

This LIBRARY Project uses MOST if not ALL of the Available RAM on this board. Please note that joining this project with another project , enabling more than one class at a time, or using this example project to fuel an application could result in an error when debugging. Licensing is required when using any Micrium software, regardless of the state of the software (Library or Source form). This project is only meant for example purposes. For the full SOURCE code project, please contact Micrium.

FRDM-K64F Example Project

MCU			
Manufacturer	**Family**	**Part Name**	**Architecture**
Freescale	Kinetis K	MK64FN1M0VLL12	ARM_Cortex_M4

PROJECT INSTRUCTIONS

PRODUCTS AND VERSION REFERENCE

TOOLCHAIN IDEs	
IDE Name	**Version**
IAR EW for ARM	7.20.2
Keil MDK-ARM	5.11.1.0
Kinetis SDK	1.0.0
MICRIUM	
Micrium Product	**Version**
uC/CPU	1.30.01
uC/LIB	1.38.00
uC/OS-III	3.04.03
uC/TCPIP	3.01.00
uC/HTTPs	2.10.01
uC/DHCPc	2.10.00
uC/Common	1.00.00

LOADING & RUNNING THE PROJECT ON THE BOARD

> **[WARNING]:** Make sure to open the project using the mentioned IDE(s) version or later. Moreover, sometimes the tools or compiler will complain or throw errors when it tries to compile a file with a very long path; therefore, it is recommended to extract the zip file in a location, such as C:\, E:\ or any other directory that will have the shortest path to compile. In addition, the versions of uC-CPU, uC-LIB and uCOS-III provided in this example are modified versions, which comes included with Freescale Kinetis SDK

Getting Started with OpenSDAv2

1. Download and install the mbed OpenSDAv2 USB drivers found at http://mbed.org/handbook/Windows-serial-configuration. For convenience, the "**mbedWinSerial_16466.exe**" is provided with this project.
2. Plug in a USB cable from a USB host to the OpenSDAv2 micro-B USB connector. This USB connection will provide power to the board.
3. The board comes with the mass-storage device (MSD) Flash Programmer OpenSDAv2 Application preinstalled. It will appear as a removable storage drive with a volume label of **MBED**. Moreover, the MSD Flash Programmer also includes a USB virtual serial port, which requires an **.INF** file for proper installation in windows. The necessary **.INF** file is available in the mbed OpenSDAv2 USB drivers mentioned in Step 1.
4. Once steps 1 through 3 are completed, we are ready to program the OpenSDAv2 to behave as a J-Link. the remaining steps will explain the process to enter into OpenSDAv2 Bootloader Mode in order to provide such behavior.
5. Unplug the USB cable if Attached.
6. Press and hold the Reset button(SW1).
7. Plug in a USB cable between a USB host and the OpenSDAv2 USB connector (Labeled "SDAUSB" on board).
8. Release the Reset button.
9. A removable driver should now be visible in the host file system with a volume label of **BOOTLOADER**. You are now in OpenSDAv2 bootloader mode.
10. Download the latest JLink OpenSDAv2 firmware from segger's website https://segger.com/opensda.html and drag and drop the *.bin into the volume mentioned in step 9. For convenience, the "**JLink_OpenSDA_V2.bin**" firmware is provided with this project.
11. When step 10 is completed, then your computer should recogize the OpenSDAv2 as a "JLink OB CDC"

IAR Embedded Workbench™

1. Click on **File-->Open-->Workspace...**
2. Navigate to the directory where the workspace is located: *$\Micrium\Examples\*Freescale*\FRDM-K64F\OS3-TCPIP-HTTPs-DHCPc-KSDK\IAR\OS3-TCPIP-HTTPs-DHCPc-KSDK-No-Source.eww*
3. Click **Open**.
4. For Safety, clean the project by clicking on **Project-->Clean**. (If Available)
5. Compile the Project by clicking on **Project-->Make**.
6. Have the board connected via OpenSDAv2 into the board input (J26) **before** downloading the project to the board.
    a. Power will be provided by the OpenSDAv2 Micro USB port
7. Download the project to the board by clicking on **Project-->Download and Debug.**
8. Run the project by clicking **Debug-->Go**. To stop the project from running click **Debug-->Stop Debugging**.

Keil uVision5™

1. Click on **Project-->Open Project...**
2. Navigate to the directory where the workspace is located: *$Micrium\Examples\Freescale\FRDM-K64F\OS3-TCPIP-HTTPs-DHCPc-KSDK\KeilMDK\OS3-TCPIP-HTTPs-DHCPc-KSDK-No-Source.uvproj*
3. Click **Open**.
4. For Safety, clean the project by clicking on **Project-->Clean Target**. (If Available)
5. Compile the Project by clicking on **Project-->Build Target**.
6. Have the board connected via OpenSDAv2 into the board input (J26) **before** downloading the project to the board.
    a. Power will be provided by the OpenSDAv2 Micro USB port
7. Download the project to the board by clicking on **Debug-->Start/Stop Debug Session**.
8. Run the project by clicking **Debug-->Run**. To stop the project from running click **Debug-->Start/Stop Debug Session** again.

µC/TCP-IP

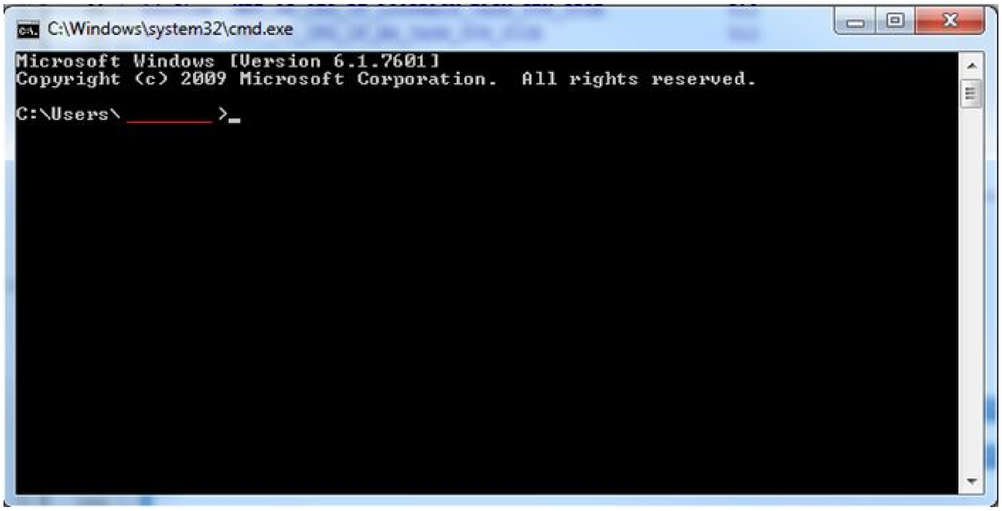
Ping Demo

This feature will test µC/TCPIP capabilities by performing a simple "ping" test on the FRDM-K64F. The configuration for this demo is based on the IP Address, Default Gateway, and Subnet Mask of not only the Host Computer/Device but also the settings of the FRDM-K64F.

- **Host Computer / Device Configuration**
    - If the Default Gateway and Subnet Mask are already known the following steps could be ignored.
    - To setup the IP Address, Default Gateway, and the Subnet Mask correctly for the "ping" demo, the use of the Command Prompt is required to locate these settings. Follow these steps to obtain this information.

> ⚠ The following steps are based on using a Windows 7™ PC as the host system. These steps may differ based on the system's operating system and settings.

- Type "cmd" in the "*Search Programs and Files*" bar of the Windows™ Start button, and select the **cmd.exe** application. A new window will appear as shown below

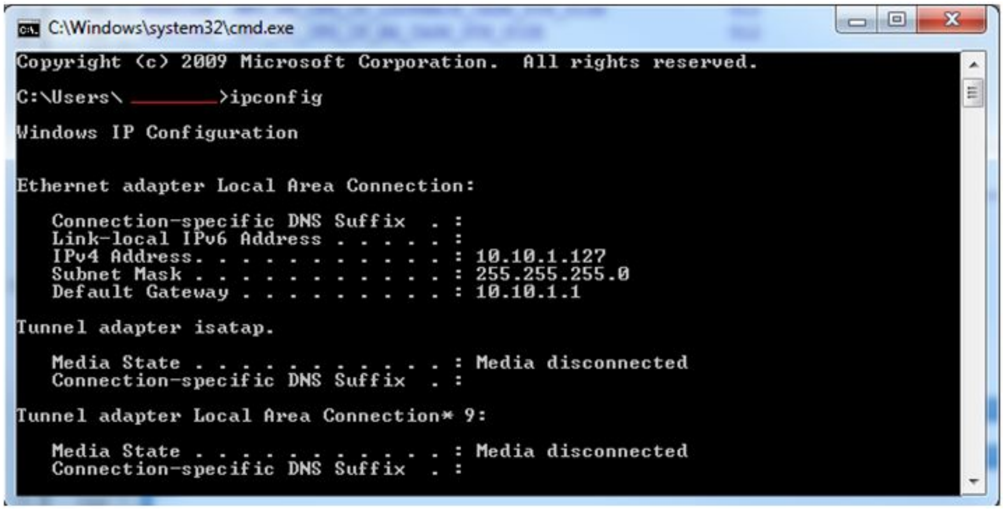


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\[redacted]>_
```

> ⚠ The "Red Line" signifies the user of the system, but for safety reasons all user names will be blacked out and replaced with a red line.

- Type in "ipconfig" into the window, press **enter**, and the following image should appear. This will provide the IP Address, Default Gateway, and Subnet Mask of the host system.



```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\[redacted]>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . :
   IPv4 Address. . . . . . . . . . . : 10.10.1.127
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.1.1

Tunnel adapter isatap.

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

> ⚠ This varies with network connection and host system.

- **Target Board Configuration & Target Board Ping**
    - The Default Gateway & Subnet Mask will become the Default Gateway and Subnet Mask necessary for the FRDM-K64F's configuration. These settings can be usually found in `app_cfg.h` file, which is located at *$/Micrium/Examples/Freescale/FRDM-K64F/OS3-TCPIP-HTTPs-DHCPc-KSDK*. If there is no #define for these, then they will be found in `app_tcpip.c` file, which is found in the same location as `app_cfg.h` file.
        - In `app_tcpip.c` file these settings are found in **AppInit_TCPIP** function.
            - An example of this configuration is shown below:

```
/* ---- IF #defines CREATED IN APP_CFG.H ---- */
ip      = NetASCII_Str_to_IPxx((CPU_CHAR *)APP_CFG_IP_ADDRESS_STR   , perr);
msk     = NetASCII_Str_to_IPxx((CPU_CHAR *)APP_CFG_IP_MASK_STR      , perr);
gateway = NetASCII_Str_to_IPxx((CPU_CHAR *)APP_CFG_IP_GATEWAY_STR   , perr);

        /* -- WHERE (in APP_CFG.H) -- */
#define  APP_CFG_IP_ADDRESS_STR                     "10.10.1.60"
#define  APP_CFG_IP_MASK_STR                        "255.255.255.0"
#define  APP_CFG_IP_GATEWAY_STR                     "10.10.1.1"

/* ---- ELSE ---- */
ip      = NetASCII_Str_to_IPxx((CPU_CHAR *)"10.10.1.60"       , perr);
msk     = NetASCII_Str_to_IPxx((CPU_CHAR *)"255.255.255.0"    , perr);
gateway = NetASCII_Str_to_IPxx((CPU_CHAR *)"10.10.1.1"        , perr);
```

`xx` - This is only relevant when using TCPIP v3.00.00 or greater. This refers to the version of Internet Protocol that you are using, either IPv4 or IPv6, for your project.

If using IPv4, then:

- `NetASCII_Str_to_IPxx becomes NetASCII_Str_to_IPv4`

If using IPv6, then:

- `NetASCII_Str_to_IPxx becomes NetASCII_Str_to_IPv6`

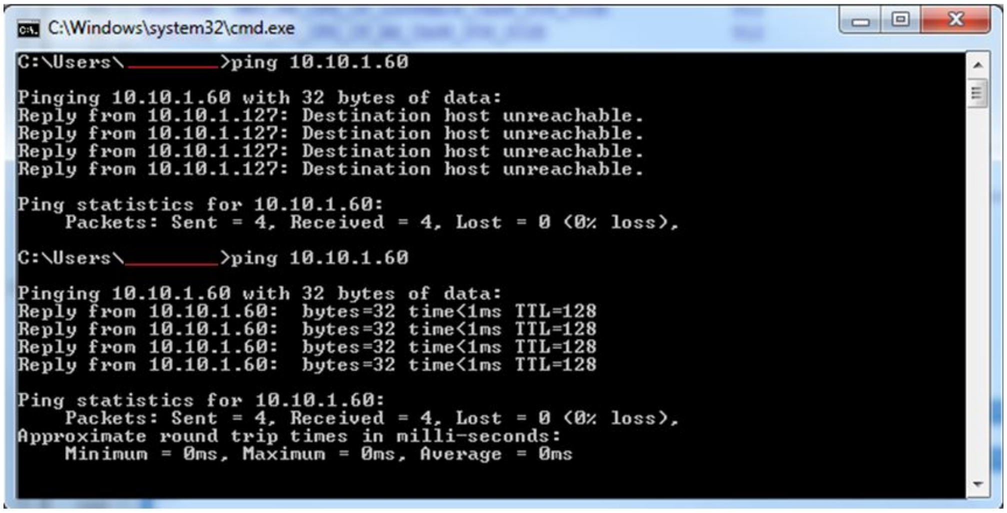
If using uC/TCPIP v2 then the `xx` is not necessary, so:

- `NetASCII_Str_to_IPxx becomes NetASCII_Str_to_IP`

    - The first 3 bytes (from left to right) to the IP Address should be the same as the inputs found from the host system.
    - The 4th byte, however, needs to be checked to make sure that that address is not being used by any other system on the host's network (if on a network).
        - To do this, open the command prompt once more, and "ping" the IP Address desired for the Target Board (FRDM-K64F).
            - If the address comes back with "*Destination Host Unreachable*" then the address is available to be used.
            - If the address comes back responsive then the IP Address is being used and a different one must be chosen.
- Once the IP Address has been chosen, and the project has been modified as necessary, it is time to run the project.
    - Connect the Ethernet Cable to the FRDM-K64F's Ethernet port.
    - Compile the Project
    - Download and Run the project.

- Once the project is running, "ping" the Target board.

    - To "ping" an IP Address, open the Command Prompt once more and type "**ping <IP Address>**" into the command prompt.
        - If properly connected, the ping should come back with the following response:



```
C:\Users\[redacted]>ping 10.10.1.60

Pinging 10.10.1.60 with 32 bytes of data:
Reply from 10.10.1.127: Destination host unreachable.
Reply from 10.10.1.127: Destination host unreachable.
Reply from 10.10.1.127: Destination host unreachable.
Reply from 10.10.1.127: Destination host unreachable.

Ping statistics for 10.10.1.60:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\[redacted]>ping 10.10.1.60

Pinging 10.10.1.60 with 32 bytes of data:
Reply from 10.10.1.60:  bytes=32 time<1ms TTL=128
Reply from 10.10.1.60:  bytes=32 time<1ms TTL=128
Reply from 10.10.1.60:  bytes=32 time<1ms TTL=128
Reply from 10.10.1.60:  bytes=32 time<1ms TTL=128

Ping statistics for 10.10.1.60:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Multiple Target Boards for Server/Client Application
If two boards are being used for a Server / Client Application, each board must have an individual MAC Address.

To change the MAC Address of a board follow these steps:

- In `net_dev_cfg.c` scroll down until **NET_DEV_CFG_ETHER** configuration is found.
- Near the end (if not the final entry) of this configuration is a character string that signifies the MAC Address.
- Change the MAC Address of one or both of the Target Boards to make sure they don't have the same MAC Address.
  - One single entry can be changed, there is no need to go to extremes and change every entry.

Here is an example of a MAC Address:

```
"00:50:C2:25:60:02",              /* Desired device hardware address; may be
NULL address or string ...                */
                                    /* ... if  device hardware address
configured or set at run-time.                     */
```

µC/DHCPc

Dynamic Host Configuration Protocol (DHCP) client Demo

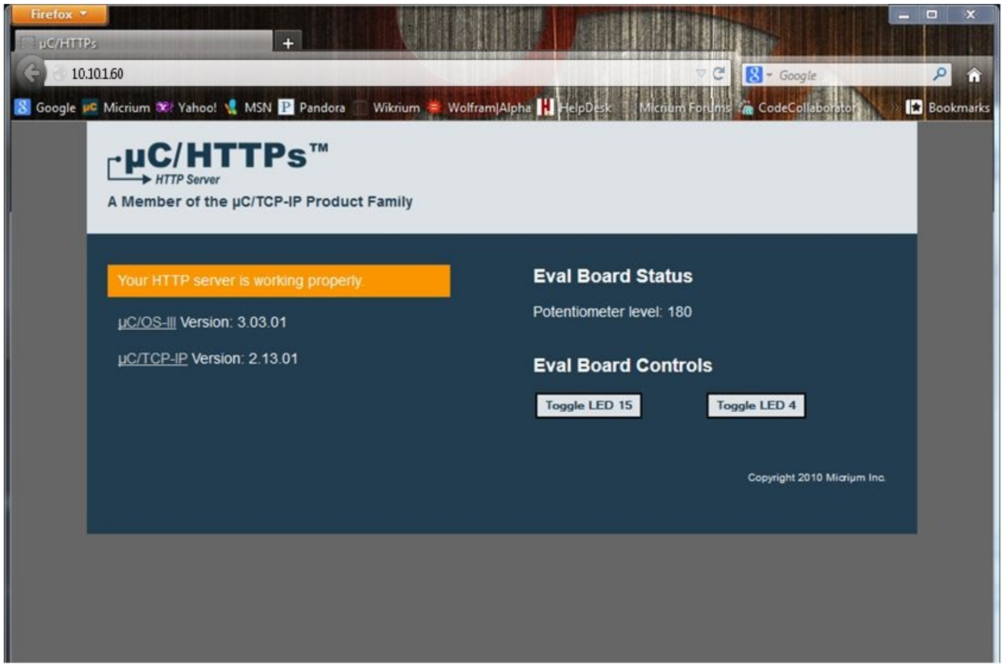
- If µC/DHCPc is in use with µC/TCP-IP, an IP address will be automatically assigned to the target board by the DHCP client.
- Once the LEDs start blinking, the assigned IP address will display in either a serial terminal or the board's LCD screen. After this, the target board can be pinged.
- If the DHCP client fails to assign an IP address to the target board, it will then default to the appropriate settings that can be configured in **App_DHCPc_Init()** function in `app.c` file.

- Please note that if the board is connected using a local network, then the DHCP settings will return a "local-link connection" and therefore the IP Address, Default Gateway, and Subnet Mask might be configured differently than what is presented in the local network.
- A DHCP server must be running in order for the DHCP client to return correct network settings. Either connect the FRDM-K64F to a DHCP network or install a DHCP server on the local Host Computer.
- If the settings of the Local Host Computer are changed to the same Default Gateway & Subnet Mask as the DHCP client settings, then the board will ping properly if no DHCP server is present.

µC/HTTPs

Webserver Demo

- Once the project is running on the target board, open a web browser of your choice (Google Chrome, Mozilla Firefox, Internet Explorer, etc.).
- In the web browser's address bar, type the target board's IP address.
- If properly connected, the web browser should display the following:
    - This is an example of how the webserver should look like.



µC/HTTPs™
HTTP Server
A Member of the µC/TCP-IP Product Family

Your HTTP server is working properly.

µC/OS-III Version: 3.03.01

µC/TCP-IP Version: 2.13.01

Eval Board Status

Potentiometer level: 180

Eval Board Controls

Copyright 2010 Micrium Inc.

- The example image shows that the HTTP server has been correctly configured and is working properly.
- By pressing the "Toggle LED 15" or "Toggle LED4" button(s) the appropriate LEDs on the board should toggle ON and OFF.