STM32F4-DISCOVERY_OS3-TCPIP-HTTPc-No-Source

STM32F4-DISCOVERY Example Project Read-Me

The provided example project for which this Read-Me was made utilizes the ST STM32F4-DISCOVERY (STM32F407) evaluation board from the STM32 Family. The MCU found on this development board conforms with the ARM_Cortex_M4 architecture.

Project Download

Download Link	Micrium_STM32F4-DISCOVERY_OS3-TCPIP-HTTPc-No-Source.zip

Toolchain IDE Versions

IDE/Toolchain	Version
IAR EW for ARM	7.30

Micriμm Product Versions

Product	Version
µC/CPU	1.30.01
µC/LIB	1.38.01
µC/OS-III	3.04.04
µC/Common	1.00.00
µC/TCP-IP	3.02.00
µC/HTTPc	1.00.00
µC/DHCPc	2.10.00
µC/DNSc	2.00.00
µC/Trace	1.00.00

Hardware Setup

1. Have the board connected via **USB** into the board debugging input (**CN1**).
2. Power this board through USB connection.

Loading & Running The Project on the Board

Make sure to open the example project workspace using the mentioned IDE(s) version or newer.

IAR Embedded Workbench™

1. Click on *File–>Open–>Workspace...*
2. Navigate to the directory where the workspace is located: *$\Micrium\Examples\ST\STM32F4-DISCOVERY\OS3-TCPIP-HTTPc\IAR-No-Source\OS3-TCPIP-HTTPc-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. Make sure your hardware setup (as previously described) is correct.
7. Download the code to the board by clicking on *Project–>Download and Debug*.
8. Run the project by clicking on *Debug–>Go*. To stop the project from running, click on *Debug–>Stop Debugging*.

µC/OS-III

```
void  main (void)
{
    ...
    OSInit(&os_err);                                   /* Initialize uC/OS-III
*/        (1)

 ...
    OSTaskCreate(&AppTaskStartTCB,                     /* Create the start task
*/        (2)
                 "App Task Start",
                  AppTaskStart,
                  0,
                  APP_CFG_TASK_START_PRIO,
                 &AppTaskStartStk[0],
                  APP_CFG_TASK_START_STK_SIZE / 10u,
                  APP_CFG_TASK_START_STK_SIZE,
                  0u,
                  0u,
                  0,
                 (OS_OPT_TASK_STK_CHK | OS_OPT_TASK_STK_CLR),
                 &os_err);
    OSStart(&os_err);                                  /* Start multitasking
*/        (3)
}

static  void  AppTaskStart (void *p_arg)
(4)
{
    ....


    while (DEF_TRUE) {                                 /* Task body, always as an
infinite loop.      */        (5)
     ...
(6)

        OSTimeDlyHMSM( 0u, 0u, 0u, 500u,
(7)
                      OS_OPT_TIME_HMSM_STRICT,
                     &os_err);
    }
}
```

Listing - app.c

(1)
OSInit() initializes uC/OS-III and must be called prior to calling OSStart(), which actually starts multitasking.

(2)
OSTaskCreate() creates a task to be managed by uC/OS-III. Tasks can be created either prior to the start of multitasking or by a running task. In this case, the task "AppStartTask" gets created.

(3)
OSStart() starts multitasking under uC/OS-III. This function is typically called from the startup code but <u>after</u> calling OSInit().

(4)
AppTaskStart is the startup task created in
(2)
.

(5)
A task must be written as an infinite loop and must not return.

(6)
In most examples, there is hardware dependent code such as LED blink, etc.

(7)
OSTimeDlyHMSM() allows AppTaskStart to delay itself for a user-specified amount of time (500ms in this case). Rescheduling always occurs when at least one of the parameters is nonzero. Placing a break-point here can ensure that uC/OS-III is running, it should get hit periodically every 500 milliseconds.

For more information please refer to **uC/OS-III Users' Guide**.

µC/TCP-IP

Ping Demo

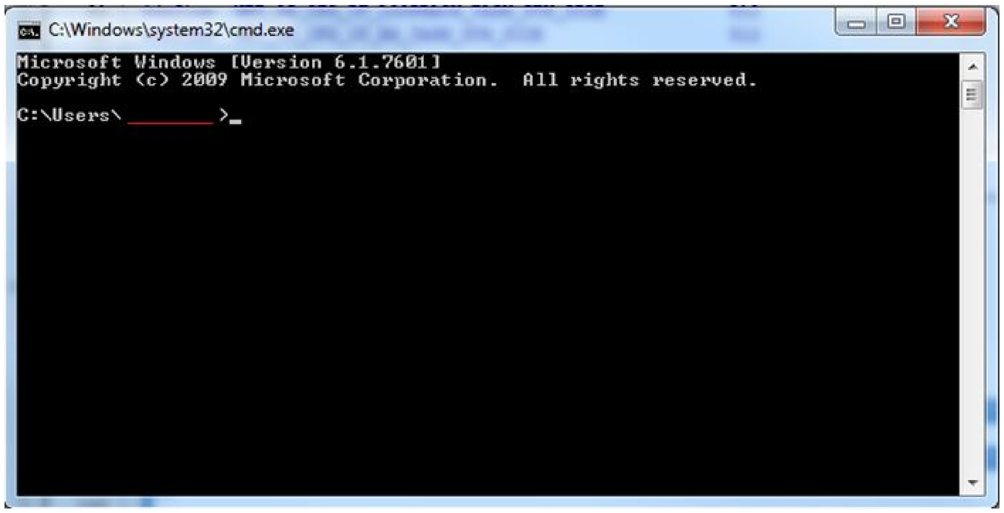
This feature will test µC/TCP-IP capabilities by performing a simple "ping" test on the STM32F4-DISCOVERY. The configuration for this demo is based on the IP address, default gateway, and subnet mask of not only the host (computer/device) but also the settings of the client.

Host 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 subnet mask correctly for the "ping" demo, the use of the Command Prompt is required to locate these settings. The following steps show how to obtain this information:

> The following steps are based on Microsoft's Windows 7™ PC as the host system. These steps may vary based on the operating system and settings.

a. Type *cmd* in the "*Search Programs and Files*" bar that appears after clicking on WindowsTM Start button. Select **cmd.exe** a pplication. A new window will appear similar to the one shown below:

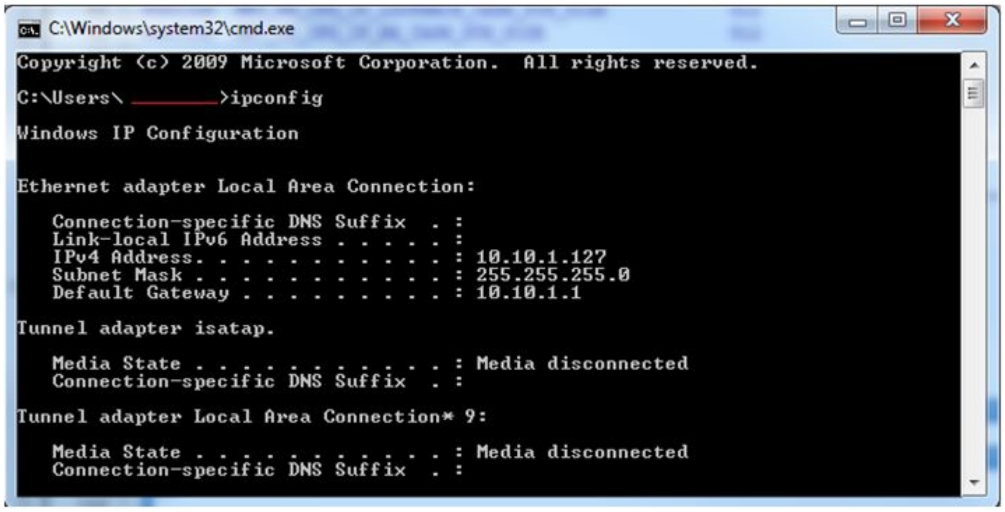


C:\Windows\system32\cmd.exe

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

C:\Users\________>_
```

b. Type *ipconfig* into the window, press **ENTER**, and the information shown in the following picture should appear. This will provide the IP address, default gateway, and subnet masks of the host system. Keep in mind that this is only a reference, the information will vary depending on the network connection and 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  . :
```

Target Configuration

- The default gateway and subnet mask will become the default gateway and subnet mask necessary for the STM32F4-DISCOVERY's configuration. These settings are usually found in **app_net.c** as the following #defines:

```
#define  APP_CFG_IP_ADDR_STR                    "10.10.1.100"          (1)
#define  APP_CFG_NET_MASK_STR                   "255.255.255.0"        (2)
#define  APP_CFG_DFLT_GATEWAY_STR               "10.10.1.1"            (3)
```

Listing - Target Configuration
(1)
The first 3 bytes (from left to right) of the IP address should be the same as the IP address from the host sytem. The 4th byte, however, must be different and also not in use by any other system on the network.

(2)
The subnet mask must be the same as the one used by the host system.

(3)
The default gateway must be the same as the one used by the host system.

- The #defines are then used to configure the interface under IPv4:

```
...
                                                                /* Set user
defined values for network                        */
ip      = NetASCII_Str_to_IPv4((CPU_CHAR *)APP_CFG_IP_ADDR_STR,      p_err);
msk     = NetASCII_Str_to_IPv4((CPU_CHAR *)APP_CFG_NET_MASK_STR,     p_err);
gateway = NetASCII_Str_to_IPv4((CPU_CHAR *)APP_CFG_DFLT_GATEWAY_STR, p_err);

(void)NetIPv4_CfgAddrAdd(if_nbr,
                         ip,
                         msk,
                         gateway,
                         p_err);
...
```

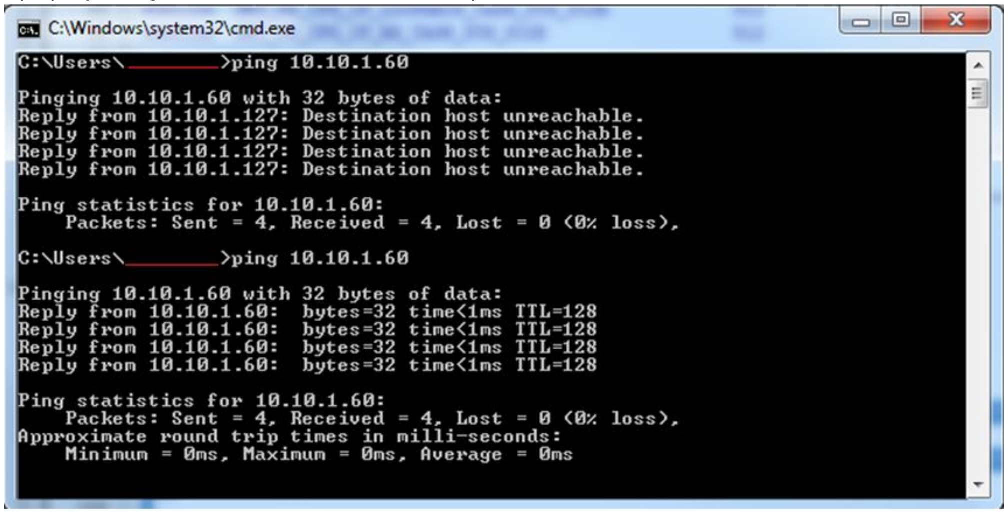
Listing - Interface Configuration

⚠ To configure the interface under IPv6, please refer to the **IP Address Configuration** section of the **µC/TCP-IP User Manual**.

Pinging the Target

Once the IP address, subnet mask, and default gateway has been configured, it is time to run the project.

1. Connect an Ethernet cable to the STM32F4-DISCOVERY's Ethernet port.
2. Compile the project as described earlier.
3. Download the code to the target and then run the project as described earlier.
4. Once the project is running, **ping** the target.
   a. To **ping** an IP address, open the "Command Prompt" and type **ping** <**IP address**>. (Replace <**IP address**> with the IP address of the target).
5. If properly configured and connected, an echo response should come back similar to the one shown in the following image:



```
C:\Users\________>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\________>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
```

If two or more targets are being used for a Client/Server application, each one must have an individual MAC address assigned.

To change the MAC address of a target, modify the last element of the device configuration structure found in **net_dev_cfg.c**.

µC/DHCPc

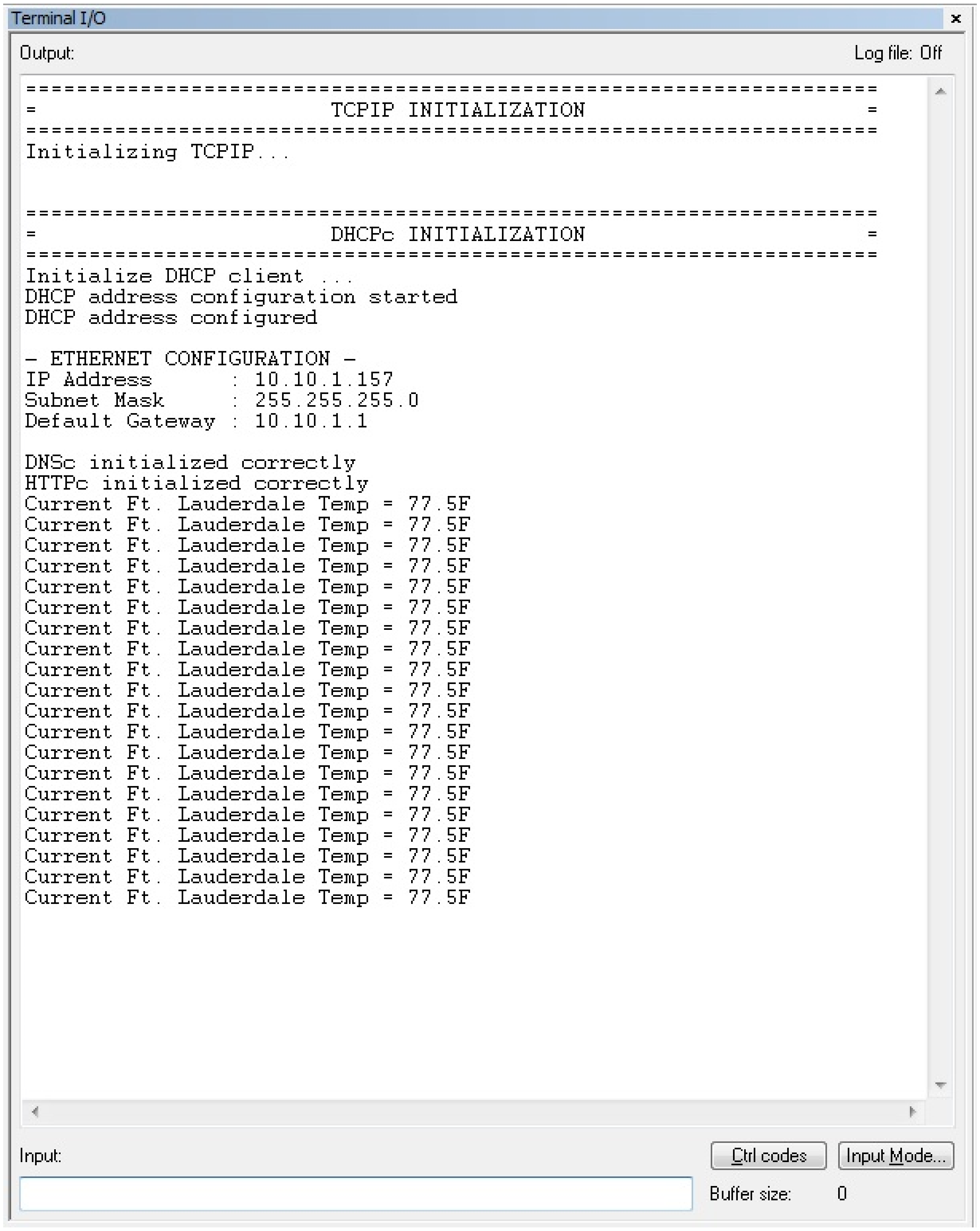
Dynamic Host Configuration Protocol (DHCP) client Demo

- If µC/DHCPc is in use with µC/TCP-IP, an IP address will be automatically assigned to the STM32F4-DISCOVERY by the DHCP client.
- Once the kernel is running, the assigned IP address is displayed on the built in I/O Terminal tab in IAR. After this, the STM32F4-DISCOVERY can be pinged.
- If the DHCP client fails to assign an IP address to the STM32F4-DISCOVERY, it defaults to a static address.

- Please note that if the STM32F4-DISCOVERY 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 STM32F4-DISCOVERY to a DHCP network or install a DHCP server on the local host system.
- If the settings of the local host system are changed to the same default gateway and subnet mask as the DHCP client settings, then the STM32F4-DISCOVERY will ping correctly if no DHCP server is present.

µC/HTTPc

The application code contained in the project uses µC/HTTPc to connect to a Web server that provides weather data, http://api.openweathermap.org. By default, the code's start task, **AppTaskStart**, repeatedly asks the weather server for the current conditions in Fort Lauderdale, FL, via HTTP GET requests. (The city is specified using the #define **HTTPc_APP_CFG_QUERY_STR_CITY** in the file *http-c_app.h*). Each time weather information is received from the server, the application displays the current temperature by making a simple call to **printf**. The output should resemble that shown in the below screenshot.



```
====================================================================
=                        TCPIP INITIALIZATION                      =
====================================================================
Initializing TCPIP...


====================================================================
=                        DHCPc INITIALIZATION                      =
====================================================================
Initialize DHCP client ...
DHCP address configuration started
DHCP address configured

- ETHERNET CONFIGURATION -
IP Address      : 10.10.1.157
Subnet Mask     : 255.255.255.0
Default Gateway : 10.10.1.1

DNSc initialized correctly
HTTPc initialized correctly
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
Current Ft. Lauderdale Temp = 77.5F
```