SoftAP web_hvac User Guide

v1.1 November 5, 2014

Introduction

This MQX demo re-uses the standard MQX web_hvac demo with the GT202 Wi-Fi module setup in SoftAP mode. This example shows MQX RTCS, DHCP server, and web server running in the Kinetis MCU with the Atheros drivers. The client will be able to connect to the Soft Access Point, receive an IP address, and then use a web browser to view the web_hvac web pages.

Requirements

This demo uses the following hardware and software:

HARDWARE:

- FRDM-K22F – Freescale Freedom board
- GT202 Wi-Fi Shield

SOFTWARE:

- Freescale MQX v4.1.0
- MQX Patches for GT202 and Atheros drivers v3.0.2
- IAR Embedded Workbench for ARM, tested with v7.30.1
- OpenSDA debugger loaded with Segger JLink app v2.1 (if using on-board OpenSDA debugger) www.segger.com/opensda.html

Demo Instructions

To build this demo, it is first necessary to follow the GT202 MQX instructions to install all the files needed to use GT202 with MQX on the FRDM-K22F board. Refer to the GT202 MQX PDK Quick Start Guide, and build the throughput demo to ensure this installation is complete. Then continue with the steps below.

Changes to BSP

By default, the GT202 drivers are setup to run the TCP/IP stack on the GT202 module. This SoftAP demo runs the MQX RTCS stack on the Kinetis MCU. To disable the stack offload mode, make these changes to the driver in the MQX BSP:

1. Edit a_config.h with changes below. a_config.h is located in the BSP project, found at \<MQX_PATH>\mqx\source\io\enet\atheros_wifi\custom_src\include\a_config.h. See screenshot below to find file in IAR workspace

```
#define ENABLE_STACK_OFFLOAD                    0
#define ENABLE_HTTP_SERVER                      0
```

```
#define ENABLE_HTTP_CLIENT                    0
#define ENABLE_DNS_SERVER                     0
#define ENABLE_DNS_CLIENT                     0
#define ENABLE_SNTP_CLIENT                    0
#define ENABLE_HTTPS_SERVER                   0
#define ENABLE_HTTPS_CLIENT                   0
#define ENABLE_SSL                            0
```

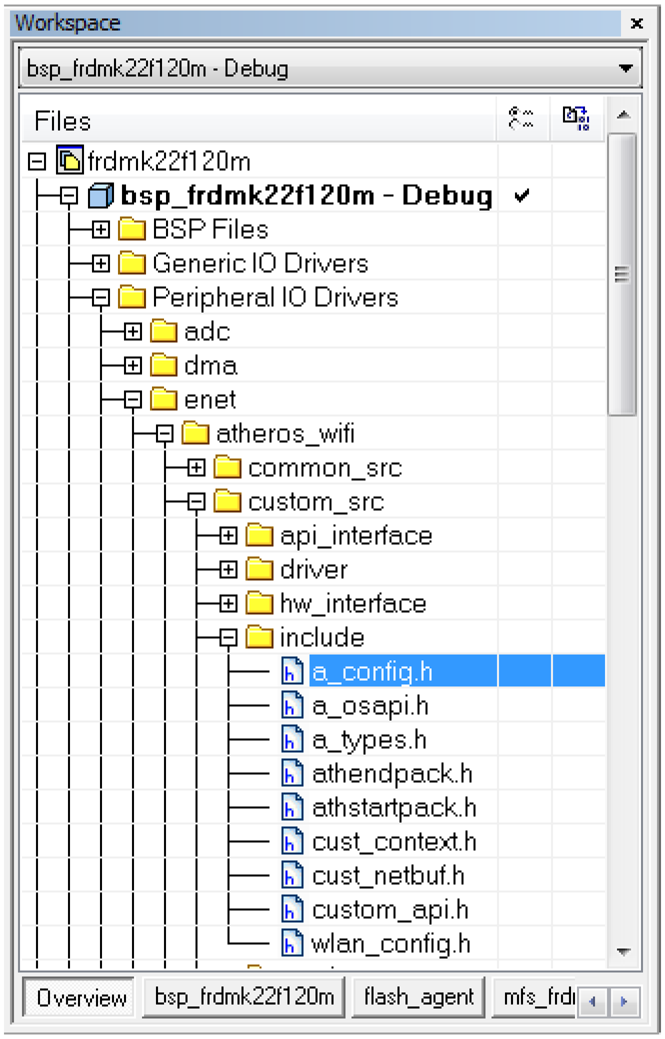


Figure 1: a_config.h location in BSP project in IAR workspace

2. OPTIONAL: the SoftAP demo will run without this change, and use the Wi-Fi network parameters in hvac.h. But to use the flashx driver and shell commands to store Wi-Fi parameters in flash, the macro MQX_ENABLE_HSRUN must be set to 0 in user_config.h. See Section Shell Commands for details on using the shell commands with flash.
3. Rebuild the BSP library

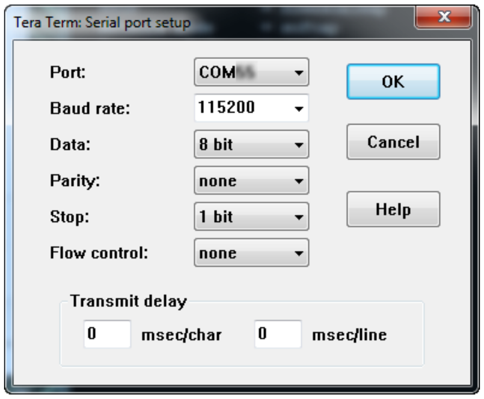
Build SoftAP Demo

1. Extract demo project to \<MQX_PATH>\demo

2. Open IAR workspace \<MQX_PATH>\demo\softap_web_hvac\build\iar\softap_web_hvac_frdmk22f120m \softap_web_hvac_frdmk22f120m.eww
3. Build softap_web_hvac project. Download to FRDM-K22F board and run.

Using SoftAP Demo

1. Connect a terminal program to the OpenSDA COM port, using these standard MQX terminal settings:



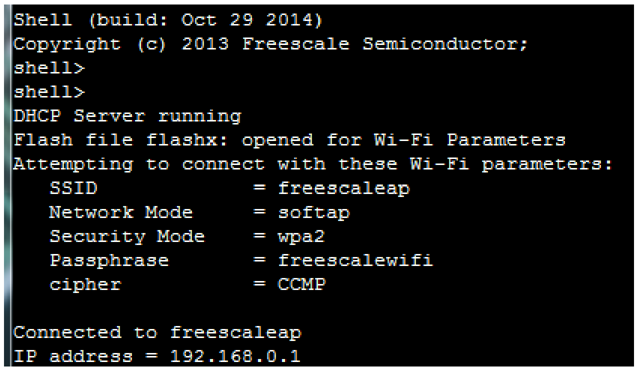
Tera Term: Serial port setup

Port:	COM[redacted]
Baud rate:	115200
Data:	8 bit
Parity:	none
Stop:	1 bit
Flow control:	none

Transmit delay: 0 msec/char 0 msec/line

Figure 2: Terminal Settings

2. Run the demo, and the output below will print to the terminal



```
Shell (build: Oct 29 2014)
Copyright (c) 2013 Freescale Semiconductor;
shell>
shell>
DHCP Server running
Flash file flashx: opened for Wi-Fi Parameters
Attempting to connect with these Wi-Fi parameters:
   SSID              = freescaleap
   Network Mode      = softap
   Security Mode     = wpa2
   Passphrase        = freescalewifi
   cipher            = CCMP

Connected to freescaleap
IP address = 192.168.0.1
```

Figure 3: SoftAP Demo Terminal Output

3. Use a Wi-Fi enabled device to connect to the freescaleap network. Use the network password freescalewifi
4. After connecting to Wi-Fi network, open a web browser and load the web server at 192.168.0.1. The MQX web_hvac web pages will be available.

Customizing SoftAP Demo

Here is a brief overview of some settings that can be easily changed in the demo

hvac.h

This header file comes from the MQX web_hvac demo, and has the main settings for the demo. The SoftAP settings can be changed using the following macros:

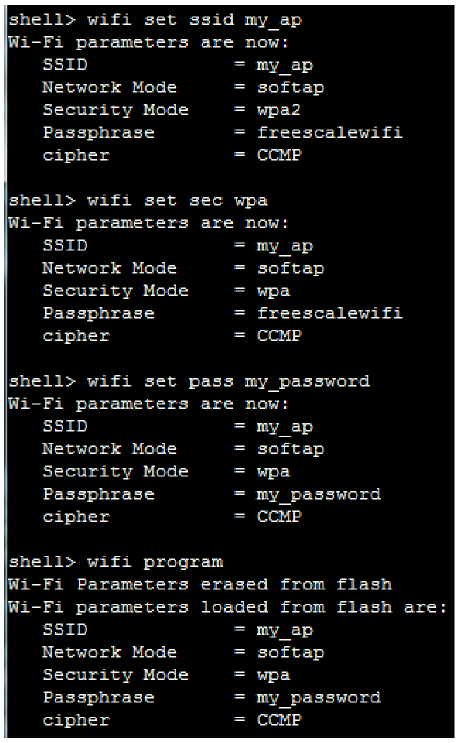
```
#define ENET_IPADDR  IPADDR(192,168,0,1)

#define ENET_IPMASK  IPADDR(255,255,255,0)
#define ENET_IPGATEWAY  IPADDR(192,168,0,1)

#define DHCP_SERVER_IP          IPADDR(192,168,0,1)
#define DHCP_BROADCAST_IP       IPADDR(192,168,0,255)
#define DHCP_MIN_IP_ASSIGNED    IPADDR(192,168,0,10)
#define DHCP_NUM_IPs_ASSIGN     10

#define DEMOCFG_SSID            "freescaleap"
#define DEMOCFG_NW_MODE         "softap"
#define DEMOCFG_SECURITY        "wpa2"
#define DEMOCFG_PASSPHRASE      "freescalewifi"
#define DEMOCFG_DEFAULT_CIPHER  CCMP     //TKIP or CCMP
```

Shell Commands

The demo also uses MQX shell to accept commands through the terminal. It includes the MQX RTCS shell commands. Also, the Atheros “wmiconfig” command is available and useful for configuring the GT202 module, refer to the Qualcomm documentation for this command.

The demo also stores the Wi-Fi network settings in flash, and shell commands can change these and re-program the flash. The macros in hvac.h are used for the default settings if there are no parameters in flash, like after the MCU is first programmed, or after an erase command through the shell. The shell command “wifi” is used for this. To see how to use it, type the command “help wifi”

This image shows an example of using the shell command to change the Wi-Fi network settings and store in flash to use after resetting the MCU:



```
shell> wifi set ssid my_ap
Wi-Fi parameters are now:
   SSID                = my_ap
   Network Mode        = softap
   Security Mode       = wpa2
   Passphrase          = freescalewifi
   cipher              = CCMP

shell> wifi set sec wpa
Wi-Fi parameters are now:
   SSID                = my_ap
   Network Mode        = softap
   Security Mode       = wpa
   Passphrase          = freescalewifi
   cipher              = CCMP

shell> wifi set pass my_password
Wi-Fi parameters are now:
   SSID                = my_ap
   Network Mode        = softap
   Security Mode       = wpa
   Passphrase          = my_password
   cipher              = CCMP

shell> wifi program
Wi-Fi Parameters erased from flash
Wi-Fi parameters loaded from flash are:
   SSID                = my_ap
   Network Mode        = softap
   Security Mode       = wpa
   Passphrase          = my_password
   cipher              = CCMP
```

Figure 4: Using shell to change Wi-Fi network parameters