

Freescale Cloud-Connected ZigBee Thermostat Demo

Getting Started Guide

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
<http://www.freescale.com/support>

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.
© 1994-2008 ARC™ International. All rights reserved.
© 2008-2014 Freescale Semiconductor, Inc. All rights reserved.

Table of Contents

1 Introduction	1
2 Using the Demo	2
2.1 Connect Tower to Terminal	3
2.2 Connecting to Wi-Fi Access Point	3
2.3 Demo GUI	4
2.4 Cloud Connection.....	7
3 Demo Hardware	10
3.1 Preparing Hardware	11
3.2 Assemble Tower.....	18
4 Demo Software	19
4.1 Firmware Images.....	19
4.2 Demo Projects.....	19
5 Preparing the Demo	21
5.1 Loading PEG Resource File	21
5.2 Pairing ZigBee Nodes	22
6 Demo Shell	23
7 Customizing Demo	23
7.1 Wired Ethernet	23
7.2 Older GT202 Shield Board Revision.....	24
8 Troubleshooting	24

1 Introduction

This reference design is a demo combining several solutions from Freescale and our partners. The demo is a thermostat application, using the Kinetis family as a communication gateway between a ZigBee network and connecting to the cloud. The demo runs on the Freescale MQX Real-Time Operating System (RTOS). It also uses the Freescale PEG graphics library for the user interface displayed on an LCD. The ZigBee communication uses Freescale's BeeStack ZigBee stack, and connects with a Freescale wireless development board programmed as a remote temperature sensor. The demo will also connect with an off-the-shelf ZigBee light bulb, and wirelessly controls it. The demo network connection is setup for Wi-Fi, using a Wi-Fi module from Qualcomm. The cloud connection allows the thermostat to be monitored and controlled remotely with mobile devices, and uses a solution provided by deviceCloud.io.

Follow this link to see a video demonstration at the 2014 Freescale Technology Forum:
<http://youtu.be/noU7R3t3Ruo>



Figure 1: Thermostat Demo

2 Using the Demo

This section assumes the demo has already been prepared, and is ready to use. It covers how to connect to a Wi-Fi access point, and how to use the cloud connection with the demo. If the demo hardware has not yet been assembled, programmed with the firmware, or prepared, refer to the following sections.

The full demo consists of the following hardware:

- Thermostat Tower System, including the following boards:
 - TWR-SHIELD is on top, with the Qualcomm Wi-Fi GT202 carrier board seated in the TWR-SHIELD headers
 - TWR-KW24D512 ZigBee Coordinator
 - TWR-K70F120M main thermostat MCU
 - TWR-SER on the bottom for connectivity
 - TWR-LCD-RGB on the side of the Tower.
- ZigBee wireless temperature sensor – using 1323X-MRB-ADC and 1323X-REM
- ZigBee light bulb

The demo GUI will display the temperature read from the wireless temperature sensor hardware. The demo will also change the color of the ZigBee light bulb to reflect the current HVAC state. If the furnace is on, the bulb is red; when the air conditioner is on the bulb is violet; when both are off the bulb is white.

2.1 Connect Tower to Terminal

When setting up the demo, it is helpful to use the terminal output from the K70. The terminal can be accessed through the 9-pin serial connector on the TWR-SER or using the OSJTAG USB virtual COM port using the USB connector on the TWR-K70F120M board. The demo uses the MQX shell library to accept commands from the user, and setup the demo. In the terminal, type “help” for the available shell commands. See Section [6 Demo Shell](#) for more details on the shell commands. The terminal settings are 115,200 baud rate, 8 data bits, no parity, 1 stop bit, no flow control.

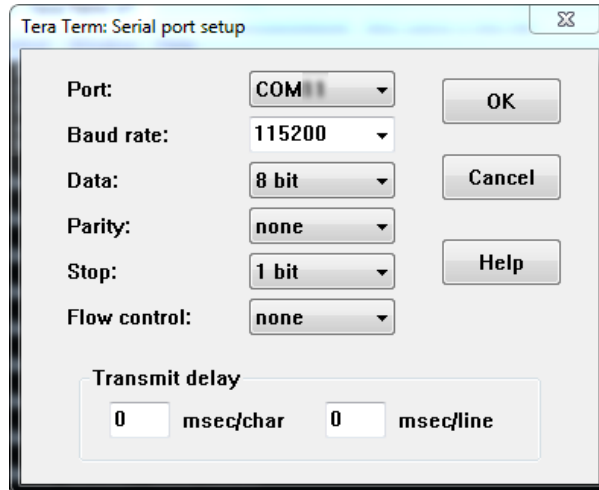


Figure 2: Thermostat Terminal Settings

2.2 Connecting to Wi-Fi Access Point

The Wi-Fi parameters can be setup using the shell, and stored in flash. Type “wifi” in the shell to see the current parameters. To change a parameter, type “wifi set <parameter> <value>”. This screenshot shows changing the SSID parameter to freescale

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

Figure 3: Changing and Viewing Wi-Fi Parameters with shell

When changing the parameter settings, the <parameter> options and values are the following:

- ssid – string for the Access Point SSID
- mode – Network Mode for the Access Point. In this demo, only “managed” value is used.
- sec – Security type used by Access Point. Options are “none”, “wep”, “wpa”, and “wpa2”
- pass – Password used if the Access Point is secured. If the security type is WEP, this is the WEP key. The value is a string

- cipher – Encryption cipher type if the security type is WPA or WPA2. The options are “CCMP” or “TKIP”. Note, if the demo attempts a WPA or WPA2 connection and fails, it will attempt with the other cipher option as well.
- key – The WEP key index if WEP is used. The only option for the demo is 1

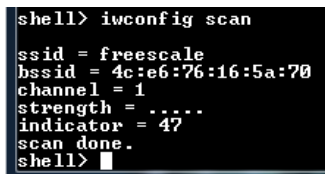
When the Wi-Fi parameters are correctly configured, the connection can be tested using the command “wifi connect”. If the settings work, they can be programmed to flash, and the demo will use them automatically after each reset. To program to flash, use the command “wifi program”. The settings in flash can also be erased to return to the default settings using the command “wifi erase”. The Qualcomm driver also supports the “wmiconfig” shell command. Refer to the Qualcomm documentation for using this command.

Default settings are also programmed in the firmware image. If any of the Wi-Fi parameter flash locations are erased, those parameters will default to the macros in the firmware. These macros are located in \Projects\HVAC_MCU_TWR-

K70F120M\examples\pro\480x272_32\web_hvac_PEG\hvac.h

```
#define DEMOCFG_SSID "freescala"
//Possible Values managed or adhoc
#define DEMOCFG_NW_MODE "managed"
//Possible vales
// 1. "wep"
// 2. "wpa"
// 3. "wpa2"
// 4. "none"
#define DEMOCFG_SECURITY "wpa2"
#define DEMOCFG_PASSPHRASE "freescalwifi"
//Possible values 1,2,3,4
#define DEMOCFG_WEP_KEY_INDEX 1
#define DEMOCFG_DEFAULT_DEVICE BSP_DEFAULT_ENET_DEVICE
#define DEMOCFG_DEFAULT_CIPHER CCMP //TKIP or CCMP
```

The demo can also scan for available Wi-Fi networks. The shell command for this is “iwconfig scan”.



```
shell> iwconfig scan
ssid = freescala
bssid = 4c:e6:76:16:5a:70
channel = 1
strength = .....
indicator = 47
scan done.
shell>
```

Figure 4: Using shell to scan for Wi-Fi networks

2.3 Demo GUI

This is the Main HVAC screen

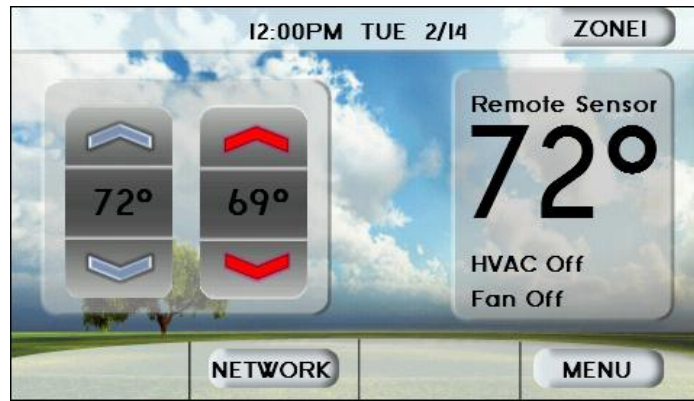


Figure 5: Main HVAC Screen

2.3.1 Displayed Temperature

The large temperature shown on the right of the screen represents the ambient temperature measured by the HVAC demo. This temperature will work with and without the ZigBee temperature sensor. When the K70 first boots, it uses a virtual temperature. The demo doesn't use an actual temperature measurement, but adjusts the virtual temperature based on the HVAC state. When the HVAC is in Heat Mode and the Heat set point is greater than virtual temperature, the temperature will increase slowly. It increments 1 degree every 5 seconds. Similarly, when the HVAC is in Cool mode, the virtual temperature will decrement slowly.

If the K70 receives a temperature report from the ZigBee remote temperature sensor via the KW24, it will stop using the virtual temperature sensor, and will use the measured temperature from the MC13237. The screen is updated with the string "Remote Sensor" above the displayed temperature when the first temperature report is received.

When the MC13237 is first paired with the KW24, it can take up to 30s before transmitting the first temperature report. After that, it will transmit regularly every few seconds.

2.3.2 Main HVAC Screen

The buttons and temperatures on the left side of the screen are for the Heat and Cool set point temperatures. These can be changed using the touchscreen and tapping the up and down arrows. The set point temperatures will also be updated if they are changed using another method, like through the cloud or shell.

The time and date at the top of the screen is displaying MQX time. Initially, MQX will start with its default initial time. But if the demo has a network connection, it will use SNTP to get the current time from an SNTP server, and update the MQX time. The screen will also update every time the MQX minute increments. The DCIO cloud agent also uses SNTP and updates MQX time.

The demo adjusts for time zone differences after receiving the time from the SNTP server. If using the DCIO agent code, the time zone is adjusted using the macros below in

`VFSL_Thermostat_Demo\Projects\HVAC_MCU_TWR-K70F120M\examples\pro\480x272_32\web_hvac_PEG\dcio-agent\dcio_system_impl.c`. If the DCIO agent is not used, the macro `SNTP_TIMEZONE` controls this

defined in `VFSL_Thermostat_Demo\Projects\HVAC_MCU_TWR-K70F120M\examples\pro\480x272_32\web_hvac_PEG\hvac.h`

```
#define DCIO_SNTP_TIMEZONE_HR    -4
#define DCIO_SNTP_TIMEZONE_MIN  -0
#define DCIO_SNTP_TIMEZONE_SEC   0
```

The HVAC and Fan state are displayed under the large temperature. These will be updated as the states change. The buttons Network and Menu change the screen for more information. The Zone1 button is for future expansion, and is not currently used.

2.3.3 Network Screen



Figure 6: Network Screen

This screen shows the network status. The IP Address, Subnet mask, and Gateway IP address values are displayed when using Wi-Fi or wired Ethernet, and are updated after the DHCP client receives them with a new connection.

If Wi-Fi is used, the SSID and security option will be displayed, and updated if they are changed through the shell. Also, the status of the Wi-Fi connection is shown. When the Wi-Fi has not yet connected, or loses connection, the string "Wi-Fi Not Connected" is shown. Once the connection is established, the string is updated with "Wi-Fi Connected". The cloud connection status is also displayed with a similar string "Cloud Connected"

When the demo is connected to the Cloud server, the button "dcio_Auth" can be pressed to request authentication with the server. If successful, this will print a 4-digit token to the top of the screen. The token can be used with the deviceCloud.io web portal to register the device with a DCIO account.

2.3.4 HVAC Menu Screen

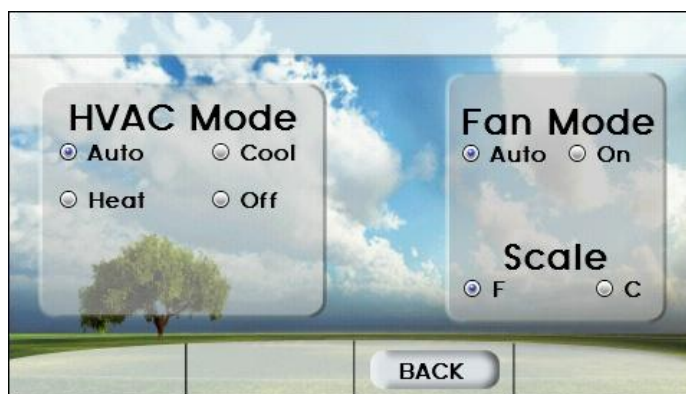


Figure 7: HVAC Menu Screen

This screen allows the user to change the HVAC and Fan mode, and the temperature scale. It will also update when these modes are changed. The Back button returns to the Main HVAC screen.

2.4 Cloud Connection

The demo uses a cloud solution from deviceCloud.io (DCIO) to provide connectivity with other devices. Once connected, the demo can be monitored and controlled with DCIO's web site or their mobile application.

2.4.1 Creating deviceCloud.io account

First, an account needs to be created to register the demo device with the cloud account. To create an account, visit the link <http://app.devicecloud.io/reg/reg.html>. The registration screen will ask for your details:

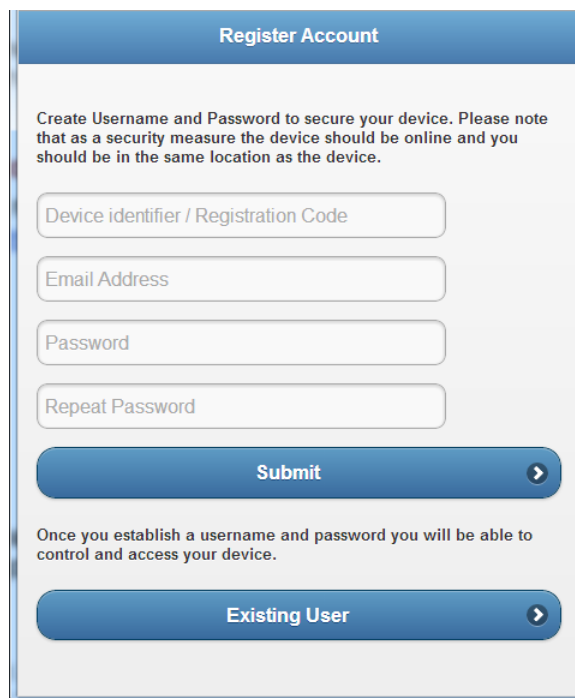


Figure 8: DCIO New User Registration

The registration code is the DCIO authentication token given by the demo device. This code can be retrieved from the GUI in the Network screen by clicking the `dcio_Auth` button. The token is displayed at the top of the screen. Or it can be retrieved in the shell using the command “`authtoken`”.



Figure 9: DCIO Token from Network Screen

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

Enter the token number in the Registration Code field with your email address and password, and click Submit button to create a new user account.

Figure 10: DCIO New User Information

The mobile application web page will then be displayed. This web link can be bookmarked, and used later to access the mobile app.

2.4.2 Using DCIO Mobile App

When the user first accesses the mobile app, the status page will be displayed. This page shows a summary of the parameters that are monitored by the cloud agent.

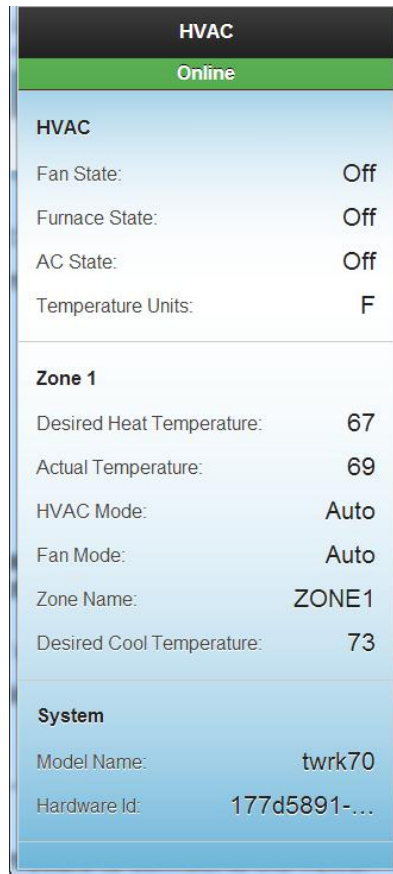


Figure 11: DCIO Mobile App Status page

Clicking on the status page will open a page to change the settings of the demo. The demo parameters can be changed, and when the “Save Changes” button is pressed, the cloud server will send the updated parameters to the thermostat device.

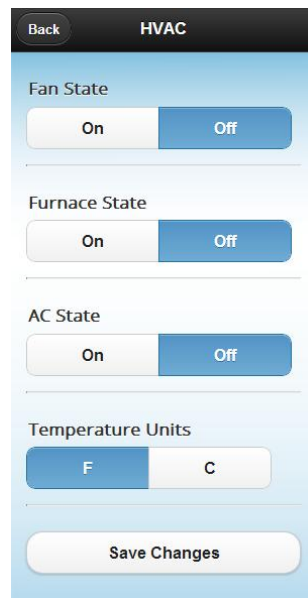


Figure 12: HVAC System Settings page

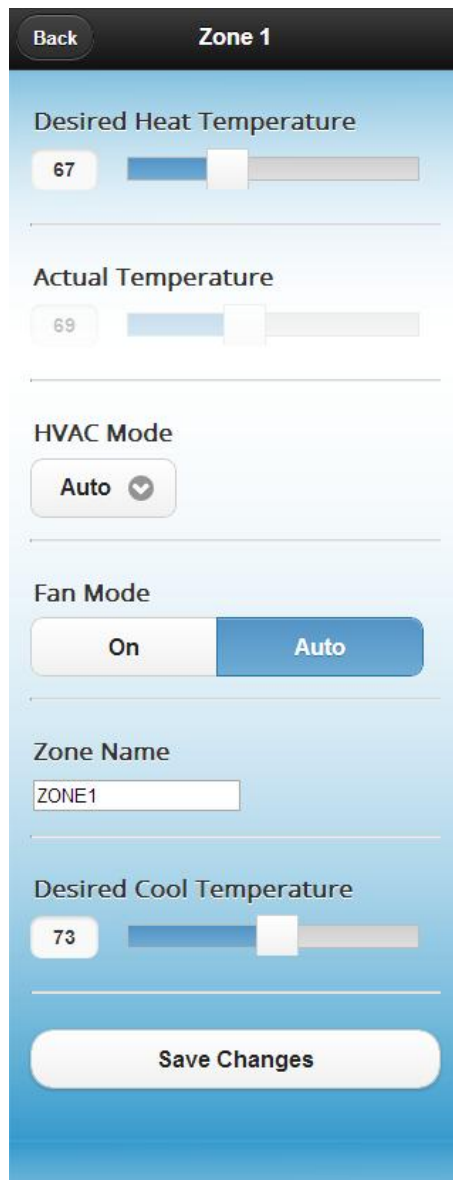


Figure 13: Thermostat Settings page

3 Demo Hardware

The demo is setup to use the following hardware. If not all of this hardware is available, refer to Section [7 Customizing Demo](#) for other options.

- TWR-K70F120M – main MCU for application. Drives LCD and runs HVAC application
- TWR-KW24D512 – ZigBee coordinator for remote temperature sensor and light bulb
- TWR-LCD-RGB – LCD for GUI
- Qualcomm GT202 carrier board – Qualcomm eval board with GT202 module, uses QCA4002 Wi-Fi SoC.
- TWR-SHIELD – allows Qualcomm board to connect to the Tower

- TWR-SER – Used for USB host to program PEG resource files into K70 NAND flash
- TWR-ELEV – both primary and secondary boards
- USB Adapter Standard-A to mini-ab – used to plug standard USB flash drive into TWR-SER board to copy PEG resource file.
- 1323X-MRB-ADC and 1323X-REM – Remote temperature sensor hardware
- Philips Hue ZigBee light bulb and standard lamp to power it – available off-the-shelf
- USB power supply – to power Tower system. Demo requires more power than many computer USB ports provide. Best to power from stand-alone supply, like a phone charger
- Access to Wi-Fi access point

3.1 Preparing Hardware

The jumper and hardware changes required are detailed below for each board.

3.1.1 TWR-SHIELD Setup

The TWR-SHIELD board comes without the headers populated, and the default connects are shorted with traces on the board. The following modifications are required by cutting traces and populating the headers, then adding jumpers.

- Cut Trace J21 between 1-2, (only needed on RevA and earlier. Newer boards do not have J21)
- Cut Trace J68 between 1-2, J68 used for Wi-Fi Power Down signal
- Remove resistor R85, next to J3, removes conflict between K70 potentiometer and GT202 module
- Need Jumper J70 on 1-2, Wi-Fi SPI Chip Select
- Leave Default - Jumper J65 on 1-2, Wi-Fi SPI Clock
- Leave Default - Jumper J73 on 1-2, Wi-Fi SPI MOSI
- Leave Default - Jumper J81 on 1-2, Wi-Fi SPI MISO
- Leave Default – Jumper J82 on 1-2, Wi-Fi Power Down signal
- Leave Default - Jumper J83 on 1-2, Wi-Fi SPI Data Ready signal
- For older GT202 shield board revision 1.0, need Jumper J68 on 2-3, for Wi-Fi Power Down signal. Revision 1.3 boards and newer use J82 with default connection for this signal.



Figure 14: TWR-SHIELD Jumper Settings

After modifications are complete, plug the Qualcomm GT202 carrier board into the TWR-SHIELD headers.



Figure 15: TWR-SHIELD with GT202 Board

3.1.2 TWR-KW24D512 Setup

- Attach the SMA Antenna
- J1 - Set jumper to 5-6 to power card from Elevator
- J17:
 - Remove Jumper on 1-2-3: Pin 3 used for WIFI_FET_ON to K70

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

- Need Jumper on 5-6: connects KW24 DataAvailablePin to K70
- Remove Jumper on 7-8-9: unused connection
- Remove Jumper 10-11-12: conflicts with WIFI_SPI_MISO
- J18 - Remove all Jumpers
 - Disconnects UART1 from Tower, jumpers 1-2-3 and 4-5-6

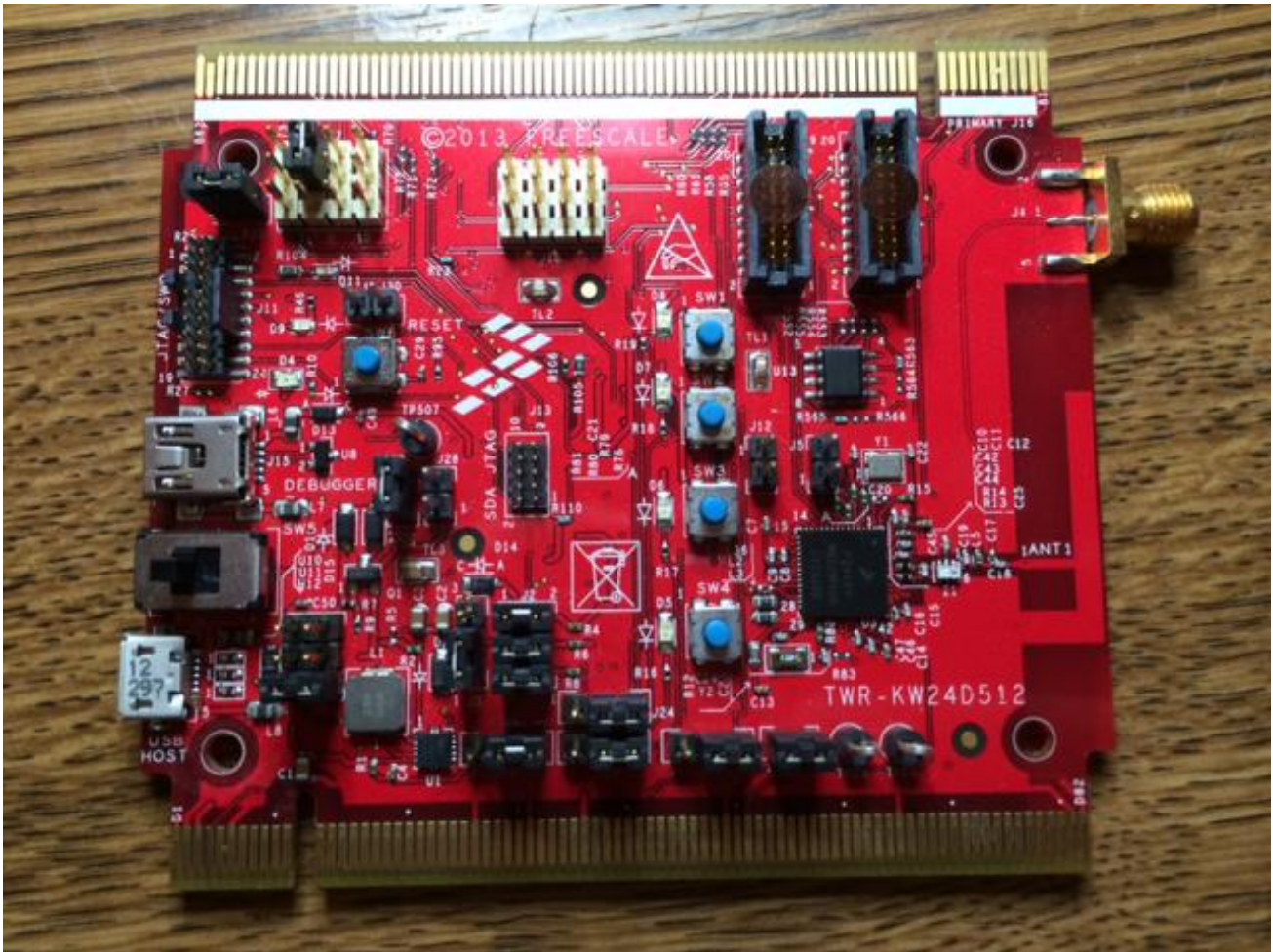


Figure 16: TWR-KW24D512 Jumper Settings

3.1.3 TWR-SER Setup

The demo is setup for Wi-Fi, and default jumper settings will work:

- J16 needs to be on pins 1-2, which is default – this is for USB Host for flash drive with LCD images.

If wired Ethernet is desired, the following changes are required. The settings for wired Ethernet will also work with Wi-Fi:

- J16 needs to be on pins 1-2, which is default – this is for USB Host for flash drive with LCD images.
- J2 CLK_SEL needs to be 3-4 – (processor clock is taken from PHY)

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

- J3 CLKIN-SEL needs to be 2-3 - (processor clock is taken from PHY)
- J12 ETH-CONFIG needs 9-10 - to select RMIII communication mode

This picture shows the jumper settings above that support both wired Ethernet and Wi-Fi



Figure 17: TWR-SER Jumper Settings

3.1.4 TWR-K70F120M

The demo is setup for Wi-Fi, and the default jumper settings will work for Wi-Fi on this board. But if wired Ethernet is desired, the following changes are required. The settings for wired Ethernet will also work with Wi-Fi:

- J18 on – disables oscillator Y1, and processor clock taken from the TWR-SER board

This picture shows the jumper settings above that work with both Wi-Fi and wired Ethernet



Figure 18: TWR-K70F120M Jumper Settings

3.1.5 TWR-LCD-RGB

No changes are required to this board; the default jumper settings will work



Figure 19: TWR-LCD-RGB Jumper Settings

3.1.6 1323X-REM and 1323X-MRB-ADC

No changes are required to this board; the default jumper settings will work

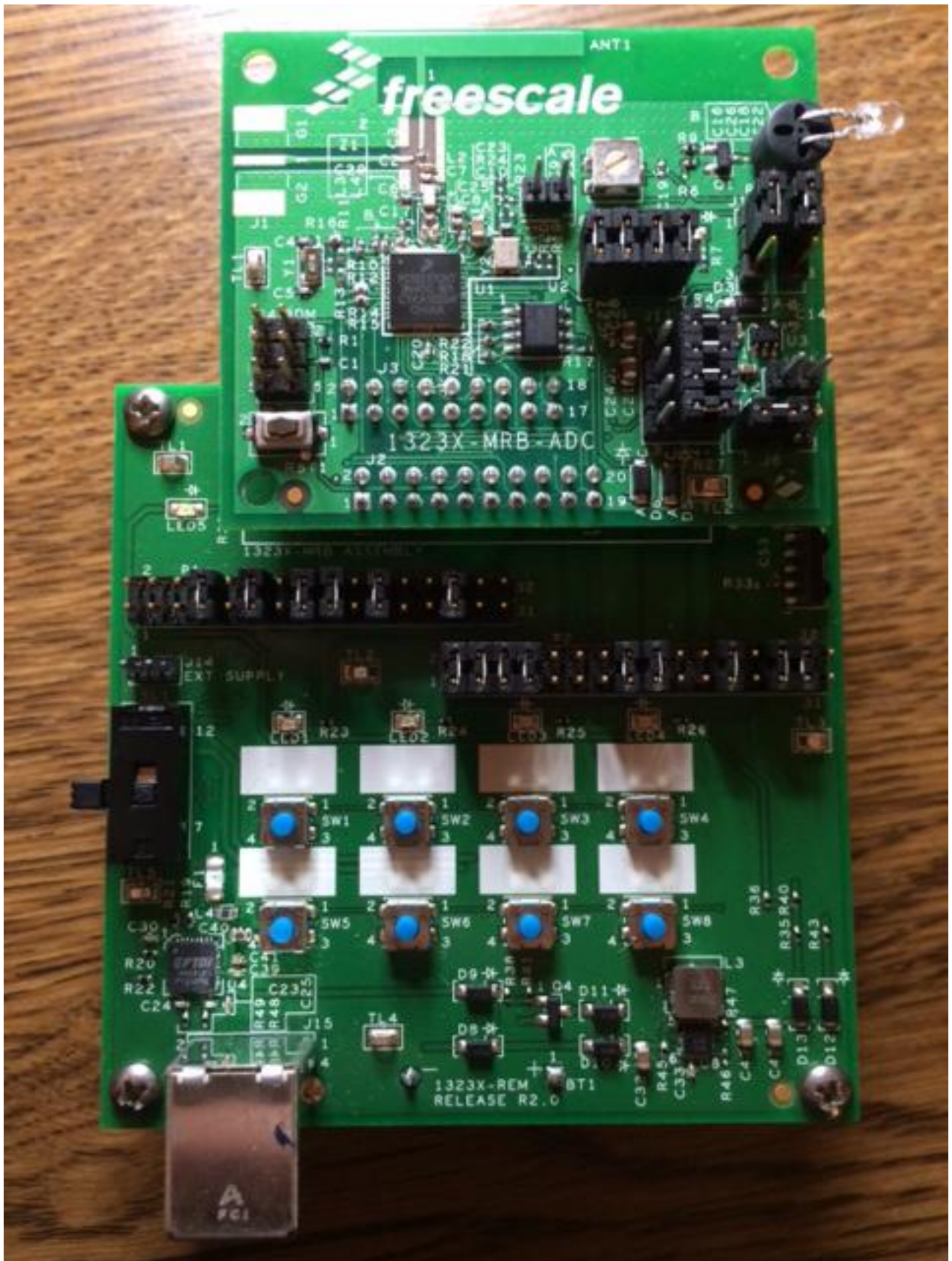


Figure 20: 1323X-REM and 1323X-MRB-ADC Jumper Settings

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

3.2 Assemble Tower

The ordering of the cards in the Tower slots is not functionally important, but this is the recommended order for using the demo:

- TWR-SHIELD is on top, with the Qualcomm GT202 carrier board seated in the TWR-SHIELD headers
- TWR-KW24D512
- TWR-K70F120M
- TWR-SER on the bottom
- TWR-LCD-RGB on the secondary side of the Tower. It will not work on the primary side. Secondary side has the black edge connectors (instead of white)

After assembling, power the Tower. With the Tower fully loaded, the demo requires more power than many computer USB ports provide. It is recommended to power from stand-alone supply, like a phone charger. Plug the USB Mini plug into the USB jack on the primary elevator card (with white edge connectors). There is also a power switch on the other side of the primary elevator card you may need to toggle to turn on the Tower.



Figure 21: Assembled Tower System

4 Demo Software

The demo uses the 3 MCUs below:

- Kinetis K70 MK70FN1M0VMJ12 – main MCU, runs HVAC application and drives LCD
- Kinetis KW24 MKW24D512VHA5 – ZigBee coordinator
- MC13237CHT – ZigBee remote temperature sensor 2.4 GHz transceiver and MCU SiP

The software release for this demo includes the projects, source code and libraries needed to build the applications. Firmware image files are also included that can be programmed into the MCUs without needing to build the projects.

4.1 Firmware Images

Firmware image files for each MCU are provided in the locations below. These files can be programmed into the MCUs using a debugger or programming tool, like IAR and CodeWarrior. Also, the BeeKit Test Tool can be used to program the KW24 and MC13237. S-Record files with .srec or .s19 file extensions are provided for each, and .out files are provided for the two Kinetis MCUs to use with IAR debugger.

- \Firmware_Images\HVAC_MCU_TWR-K70F120M
- \Firmware_Images\ZigBee_Coordinator_TWR-KW24D512
- \Firmware_Images\Remote_Temperature_Sensor_1323X-MRB-ADC

4.2 Demo Projects

The K70 and KW24 projects were tested with IAR Embedded Workbench for ARM v7.10.1. The project for the 8-bit S08 MC13237 was tested with CodeWarrior for MCUs v10.6. The following sections give instructions for building these projects.

4.2.1 K70 Project

The K70 application is built on Freescale MQX RTOS v4.0.2, and uses the Freescale PEG graphics library. The MQX source code is included for this demo, and the MQX libraries need to be built before building the application. The PEG source code is not included, but a pre-built library is included for the project, and the screen source files are included.

To build the project, use IAR to load the workspace file \Projects\HVAC_MCU_TWR-K70F120M\os\MQX\config\twrk70f120m\iar\build_libs.eww. The MQX libraries need to be built first. Batch Build them all using the **IAR menu Project->Batch Build**. Select the **Debug** batch, and hit the Make button.

After the MQX libraries build, ensure the application project is set as active. In the workspace window, right-click the project **web_HVAC_PEG**, and select “Set as Active”. Once active, that project will be bold in the workspace. Make and then Download the web_HVAC_PEG project to the TWR-K70F120M board.

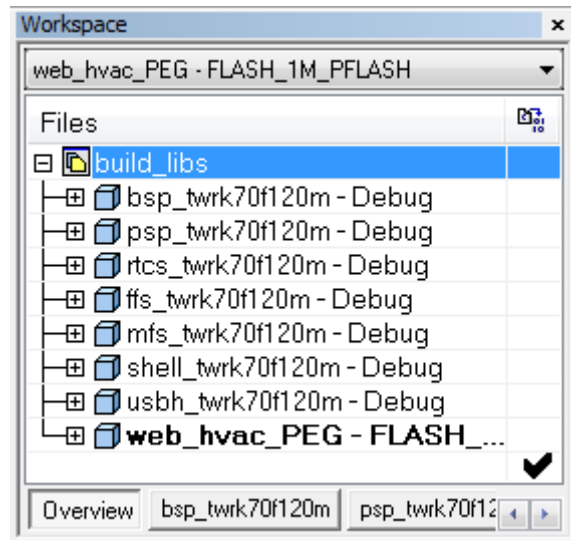


Figure 22: K70 IAR Workspace

4.2.2 KW24 Project

The KW24 application is built on Freescale MQX-Lite RTOS, and uses the Freescale connectivity stack libraries. The MQX-Lite source code is included for this demo, and the MQX-Lite library needs to be built before building the application.

To build the project, use IAR to load the workspace file \Projects\ZigBee_Coordinator_TWR-KW24D512\ConnSw_Demos\make\DP_OnOffSwitch_NwkIp\DP_OnOffSwitch_NwkIp.eww. The MQX-Lite library needs to be built first. Right-click the **MQXLite project** in the workspace, and select **Make**.

After the MQX-Lite library builds, ensure the application project is set as active. In the workspace window, right-click the project **DP_OnOffSwitch_NwkIp**, and select "Set as Active". Once active, that project will be bold in the workspace. Make and then Download the DP_OnOffSwitch_NwkIp project to the TWR-KW24D512 board.

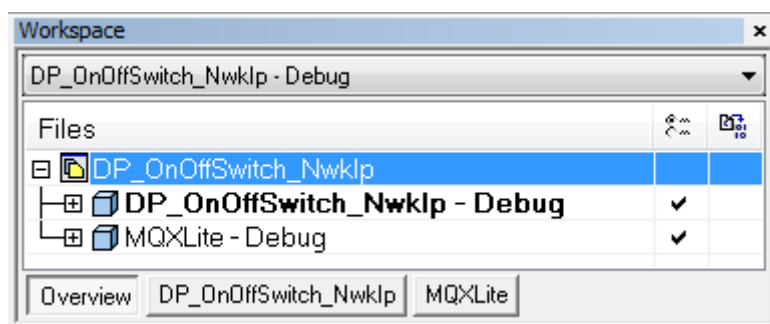


Figure 23: KW24 IAR Workspace

4.2.3 MC13237 Project

To build the project, use CodeWarrior to import the project \Projects\Remote_Temperature_Sensor_1323X-MRB-ADC. Use CodeWarrior to Build and then Debug the project, which will program the 1323X-MRB-ADC board.

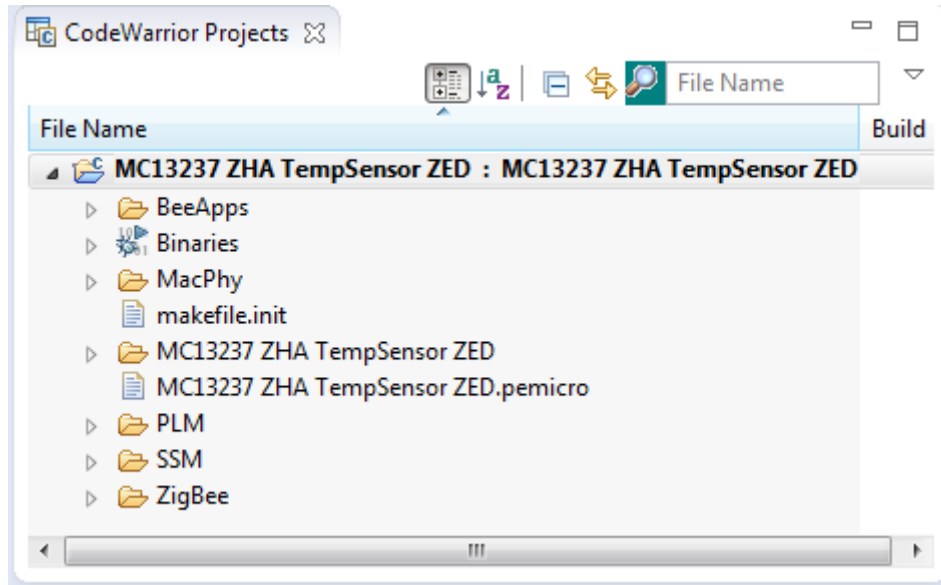


Figure 24: MC13237 CodeWarrior Workspace

5 Preparing the Demo

After programming the firmware and assembling the Tower system, there are additional steps to setup the demo.

5.1 Loading PEG Resource File

The PEG application uses a resource file for the graphics, fonts, strings, etc. This file is stored in file system mounted by the MQX application at runtime. There are two options for the location of the file. It can be stored in the NAND flash on the TWR-K70F120M board, or stored in a USB flash drive. Using the USB flash drive is useful when developing the GUI, as the file can be replaced on the flash drive and tested quickly. Once the GUI is complete, the MQX shell can be used to copy the resource file from the USB drive to the NAND drive. If the file is present in both NAND and USB, the application will use the USB file. Here are the steps for using the resource file:

Resource file on USB flash drive:

1. Use a USB Flash drive formatted as FAT (FAT32)
2. Locate the resource file in the project, found at \Projects\HVAC_MCU_TWR-K70F120M\examples\pro\480x272_32\web_hvac_PEG\UI\resources.pbr
3. Copy this file to the USB flash drive

4. Ensure the TWR-SER jumper setting is setup for USB host. The 6-pin header J16 (near the USB connector) should have a jumper on 1-2, which is default.
5. Plug the USB flash drive into the TWR-SER USB connector. An adapter is required to mate the Standard USB A connector of the USB drive to the mini-ab connector on the TWR-SER board
6. Power the Tower, and run the application. After a few seconds, MQX will mount the USB drive, and PEG will load the resources from the drive, and display the GUI on the LCD. The terminal output will also show status on mounting the USB drive and reading the file.

Resource file on NAND drive:

1. Remove the TWR-KW24D512 board from the Tower before programming the NAND flash.
2. Follow the USB steps above to mount the USB flash drive in the MQX application
3. Use the shell commands through the terminal to copy the resource file from the USB drive to the NAND flash
4. Close the file system to erase the NAND flash:
fsclose
5. The first time the NAND flash is used on the K70 board, the NAND flash should be erased. Type the command below in the terminal:
nanderasechip
6. Type the command below to open the NAND file system
fsopen
7. The first time the NAND file system is used, it needs to be formatted. Type this command:
format a:
8. Copy the file from the USB flash drive (c:) to the NAND flash (a:) with this command
copy resources.pbr a:\resources.pbr
9. Close the file system to save the file to NAND:
fsclose
10. Remove the USB flash drive and reset the K70. The application will now mount the NAND file system, and load the resource file.
11. The next time the file is copied from the USB drive to NAND, use the commands fsopen, copy and fsclose (the erase and format steps can be skipped). Then, the USB drive can be removed, and the resource file will be loaded from NAND flash after reset. Also, the MQX file system supports other DOS commands for the file system. Type help in the terminal for the available shell commands
12. The TWR-KW24D512 can now be inserted back in the Tower, and the resource file should load successfully from the NAND flash.

5.2 Pairing ZigBee Nodes

After the flash in the MCU is updated, the ZigBee nodes need to be paired. The KW24 ZigBee coordinator pairs with the temperature sensor first, and then with the ZigBee light bulb.

To pair with the temperature sensor board, press the SW4 button on the TWR-KW24D512 board. An LED will start flashing on this board. Next, power the 1323X-REM board. LED1 will start flashing indicating it needs to be paired. Press SW1 on the 1323X-REM board, and the LEDs will

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

cycle as the end device pairs with the coordinator. Be sure the 1323X boards are close to the Tower system for the pairing. Once completed, the 1323X-REM LEDs will stop cycling and LED1 and LED3 will remain lit. Also, the LED on TWR-KW24D512 will stop blinking. The pairing information for both devices is now stored in flash. If the 1323X-REM is power-cycled, it will power up with LED1 and LED3 lit indicating it is already paired. If the pairing process needs to be repeated, press and hold SW3 on the 1323X-REM board, which will erase the pairing information from flash. After pairing, the temperature sensor will start sending reports with temperature data to the coordinator. This takes about 30 seconds for the first report. The GUI will update the display with “Remote Sensor” displayed above the temperature reading after the first report is received.

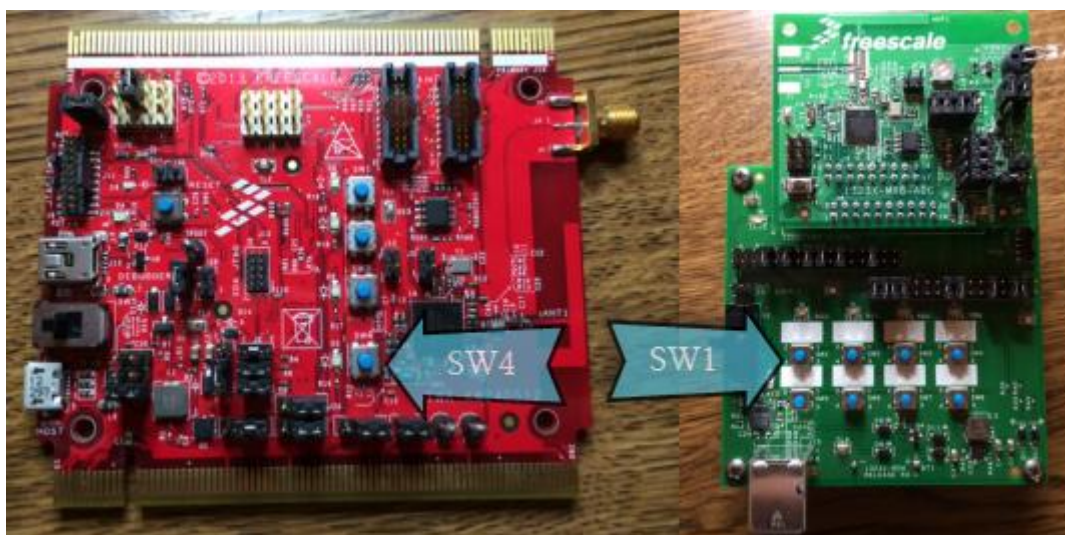


Figure 25: Buttons for ZigBee Pairing

If the ZigBee light bulb is used with the demo, it can be paired next with the ZigBee coordinator. Power the light bulb, and place it very close to the Tower system. Press SW4 again on the TWR-KW24D512 board. The LED will start blinking again as the coordinator looks for the light bulb. When the pairing is complete, the LED will stop blinking. This is also stored in flash if the Tower is power cycled. The communication with the light bulb can be tested by pressing SW3 on the TWR-KW24D512 board, which will toggle the light bulb.

6 Demo Shell

7 Customizing Demo

7.1 Wired Ethernet

Set the macros below to 0 in \FSL_Thermostat_Demo\Projects\HVAC_MCU_TWR-K70F120M\os\MQX\config\trw70f120m\user_config.h, then rebuild the MQX BSP project.

```
// For Atheros Wi-Fi driver
#define BSPCFG_DB142_IOT_MODE_SUPPORT    0
#define BSPCFG_ENABLE_ATHEROS_WIFI      0
```

Then, set the macro below to 0 in \FSL_Thermostat_Demo\Projects\HVAC_MCU_TWR-K70F120M\examples\pro480x272_32\web_hvac_PEG\hvac.h, and rebuild the demo application.

```
#define DEMOCFG_USE_WIFI                  0
```

Freescale Cloud-Connected ZigBee Thermostat Getting Started Guide Rev. 0.4

7.2 Older GT202 Shield Board Revision

With Rev1.3 of the GT202 shield board, Qualcomm changed the pin for the CHIP_PWD signal to power down the board. From Revision 1.3 and newer, this signal is on the GT202 shield pin J2-2 which is PTE27 of the K70. On the older Rev 1.0 boards, this signal was on GT202 shield pin J1-8, connected to K70 PTE0.

The MQX BSP defaults to the newer pinout using PTE27 for this signal. If the older Rev1.0 GT202 shield board is used, uncomment the macro below in user_config.h, rebuild the BSP, and then rebuild the application:

```
#define BSPCFG_GT202_REV1_0 1
```

8 Troubleshooting

Here is an example how the terminal looks after booting the demo, and successfully connecting to the Wi-Fi Access Point and cloud server.

```
shell> Resetting HAN
Shell (build: Apr 16 2014)
Copyright (c) 2013 Freescale Semiconductor;
shell>
shell>
I2C Read HANA
Flash file flashx: opened for Wi-Fi ParametersAttempting to connect with these W
i-Fi parameters:
  SSID          = freescale
  Network Mode  = managed
  Security Mode = wpa2
  Passphrase    = freescalewifi
  cipher        = CCMP
NAND flash installed to a:
PEG Opening resources.ubr from NAND FFS...Success
Connected to freescale
Polling DHCP...
DHCP bind successful.
IP address = 192.168.1.72
SNTP Server time.slackadelic.com IP Address : 216.23.247.62
IDGIO_CLIENT1: Dev->Cloud connection status : RECONNECTING
IDGIO_CLIENT1: Launching DcioClient...
IDGIO_CLIENT1: Random local port 4509
IDGIO_CLIENT1: Connected to Cloud server, IP = 54.186.2.228:80 :: 4509
IDGIO_CLIENT1: Registration success :deviceCloud Connected
IDGIO_CLIENT1: Dev->Cloud connection status : CONNECTED
IDGIO_CLIENT1: Received mt5: getallobjects
```

Figure 26: Terminal screen after boot and connecting to Wi-Fi and cloud server