

# GPIO Wi-Fi Demo Quick Start Guide

---

## IN THIS DOCUMENT

- ▶ sw\_gpio\_examples GPIO and Wi-Fi demo : Quick Start Guide
- 

## 1 sw\_gpio\_examples GPIO and Wi-Fi demo : Quick Start Guide

This example demonstrates the use of two Slice Cards, XA-SK-GPIO and XA-SK-WIFI slice together with the xSOFTip components for Wi-Fi, SPI, I2C and WebServer to provide access to the GPIO slice features via a simple embedded webserver.

- ▶ Turn GPIO Slice Card LEDS on and off
- ▶ Read the room temperature via the on-board ADC and display on the web page
- ▶ Display GPIO Slice Card button press status

### 1.1 Hardware Setup

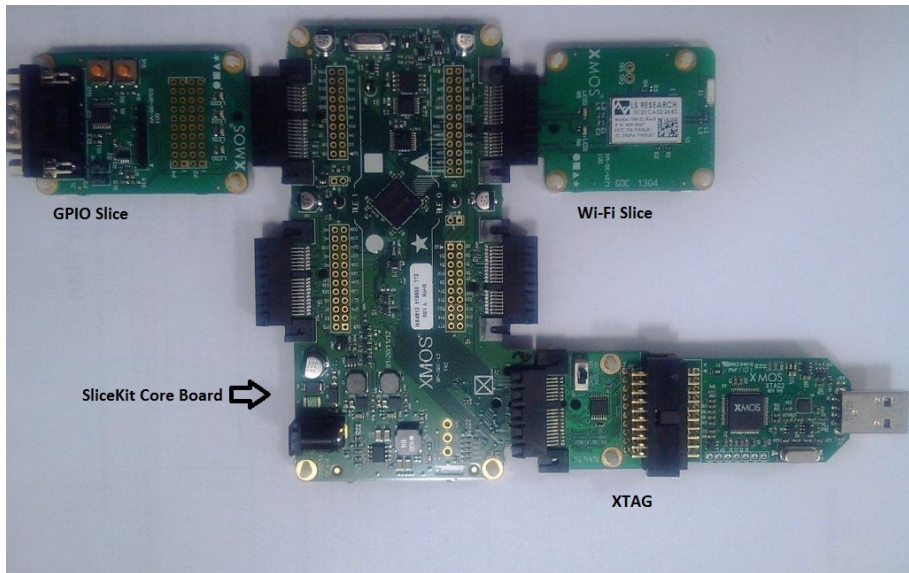
The XP-SKC-L2 Slicekit Core board has four slots with edge connectors: SQUARE, CIRCLE, TRIANGLE and STAR.

To setup up the system refer to the figure and instructions below

1. Connect the XA-SK-WIFI Slice Card to the XP-SKC-L2 Slicekit Core board using the connector marked with the TRIANGLE.
2. Connect the XA-SK-GPIO Slice Card to the XP-SKC-L2 Slicekit Core board using the connector marked with the SQUARE.
3. Connect the XTAG Adapter to Slicekit Core board, and connect XTAG-2 to the adapter.
4. Connect the XTAG-2 to the host PC.
5. Make sure your wireless access point is switched on and configured correctly.
6. Switch on the power supply to the Slicekit Core board.

### 1.2 Import and Build the Application

1. Open xTIMEcomposer and then open the edit perspective (Window->Open Perspective->XMOS Edit).



**Figure 1:**  
Hardware  
Setup for the  
GPIO and  
Wi-Fi  
Application  
Demo

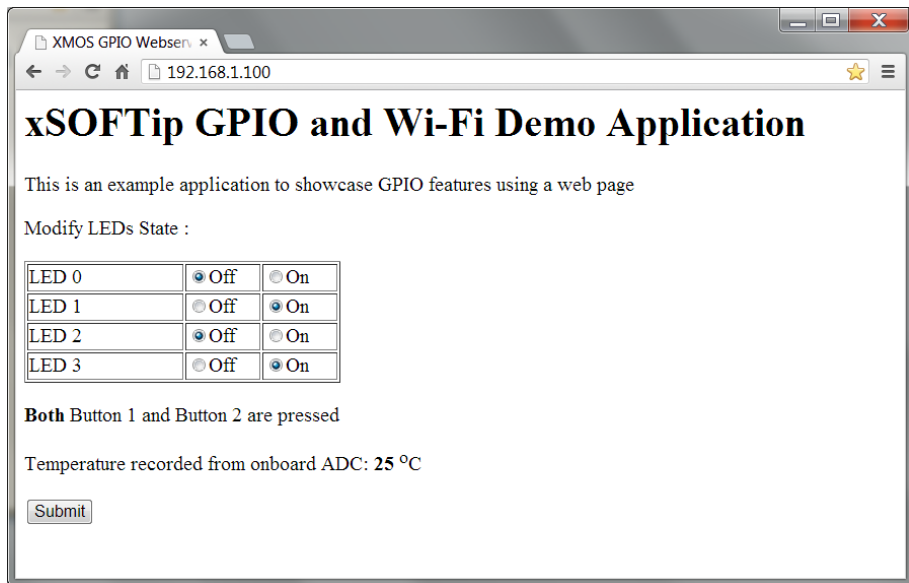
2. Locate 'Slicekit GPIO Wi-Fi Demo' item in the xSOFTip Browser window and drag it into the Project Explorer window in xTIMEcomposer. This will also cause the modules on which this application depends (in this case, module\_wifi\_tiwisl, module\_spi\_master, module\_i2c\_master, module\_webserver) to be imported as well.
3. Click on the Slicekit GPIO Wi-Fi Demo item in the Explorer pane then click on the build icon (hammer) in xTIMEcomposer. Check the console window to verify that the application has built successfully.

For help in using xTIMEcomposer, try the xTIMEcomposer tutorials, which you can find by selecting Help->Tutorials from the xTIMEcomposer menu.

### 1.3 Running the Demo

1. Enable offline Xscope printing in the run configurations. Run Configuration -> XScope: Select Offline [Xscope] mode.
2. Click on the Run icon (the white arrow in the green circle). A dialog will appear asking which device to connect to. Select XMOS XTAG2.
3. xTIMEcomposer console displays the IP address obtained by the DHCP client
4. Open a web browser on the host PC and type the IP address displayed on the xTIMEcomposer console into the browser's address bar

5. On hitting the return key, a web page should get loaded and displayed in the browser as shown in the figure below.



**Figure 2:**  
Screenshot of  
GPIO web  
page

1. Switch on all the LEDs by turning them all to 'ON' in the browser then clicking Submit. The LEDs should light, the ADC temperature display should be updated, and the webpage will report no buttons have been pressed.
2. Switch off two of the LEDs by turning two to 'OFF' in the browser then clicking Submit. Two LEDs should go out and the ADC temperature is reported again. This time also webpage will report no buttons have been pressed.
3. Press SW1 button on the GPIO slice card and then press submit. The web page should now report that Button 1 is pressed.
4. Press SW2 button on the GPIO slice card and then press submit. The web page should now report that Button 2 is pressed.
5. Just hitting Submit now displays both the buttons are not pressed. LEDs states remain unchanged unless they are explicitly changed on the web page.

## 1.4 Next Steps

### 1.4.1 Look at the Code

1. Examine the application code. In xTIMEcomposer navigate to the `src` directory under `app_sk_gpio_wifi_tiwisl_combo_demo` and double click on the `main.xc` file within it. The file will open in the central editor window.

2. The channel `c_gpio` is used between web page handler and application handler to send web page requests to the application and to collect GPIO status from the application.
3. In the `app_handler.xc` file, API `set_gpio_state` is used by the web page in order to apply web page LED settings and similarly API `get_gpio_state` is used by web page to collect the current GPIO status containing LEDs, button presses and ADC temperature values.
4. GPIO button scan logic monitors for value changes on the configured 4-bit button port (`XS1_PORT_4C`) in the application handler routine as defined in the `app_handler.xc` file. Whenever this port value changes, GPIO button states are updated accordingly.
5. You can also observe that the ADC value is read whenever there is a web page request. This value is interpolated to get a proper temperature value and is updated in the GPIO state structure before sending it to the web page.

#### 1.4.2 Building web pages for your applications

This application parses Wi-Fi data to interpret web page commands. Refer to Programming Guide section within the `SliceKit GPIO Example Applications` documentation linked from the front page documentation for this demo for more information on how to utilize the `Embedded Webservice Function Library` component in building your own custom web server applications.



Copyright © 2013, All Rights Reserved.

---

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.