Contents

Pre-requisites

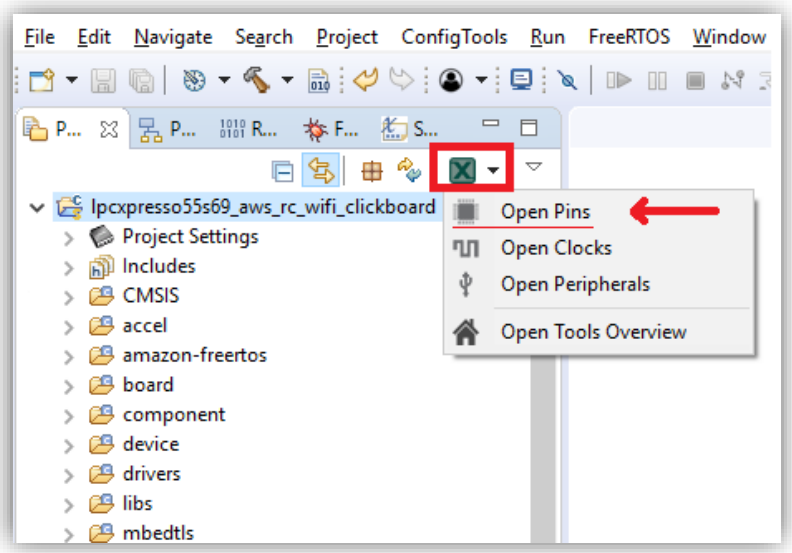
- Download the latest MCUXpresso IDE for your platform (https://mcuxpresso.nxp.com)
- Download the latest SDK for your platform (https://mcuxpresso.nxp.com)
  Note: Don't forget to enable AWS component in the SDK.
- Import "AWS Remote Control" demo

Objectives

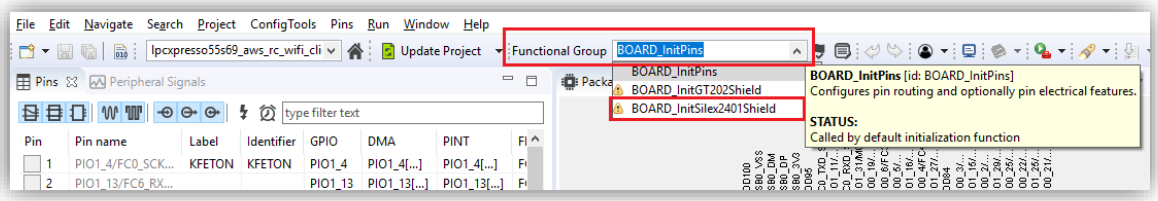
Modify existing AWS_Remote_Control project to enable Silex Clickboard (mikroBus headers) instead of Silex shield (Arduino headers) using the Pin Tools configuration tool from MCUXpresso.

Open Pin Tools

1. Select the project and click on the "Open MCUXpresso Configuration" button to open the Pin configurations as shown below.



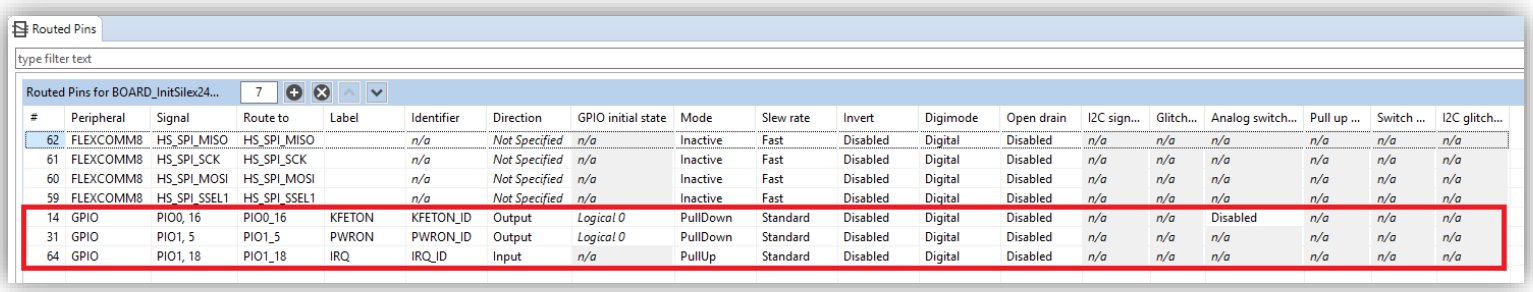
2. The MCUXpresso Configuration tool will open and display the current port configuration.
3. Click in the "Functional Group" and select "BOARD_InitSilex2401Shield" as shown below.
   Note: The "Functional Group" displays the different modules being configured in the project through the pin_mux.c/h files. Ie: BOARD_InitPins.



4. Look at the "Routed Pins" tab below, it displays the configured pins for that function. **Modify the GPIO ports** at the end of the BOARD_InitSilex2401Shield table to match the Clickboard pins:

Silex 2401 **Shield**		Silex 2401 **Clickboard**		Peripheral	Label	Direction	Initial state
Signal (Port)	Route to (Pin)	Signal (Port)	Route to (Pin)				
PIO1_4	01	PIO0_16	14	GPIO	KFETON	Output	Logical 0
PIO1_7	09	PIO1_5	31	GPIO	PWRON	Output	Logical 0
PIO0_15	22	PIO1_18	64	GPIO	IRQ	Input	n/a

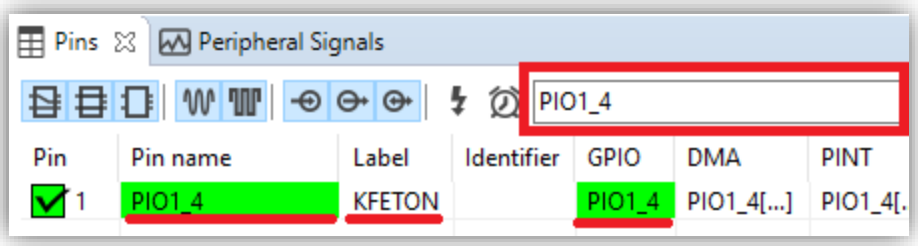
It should look like this:



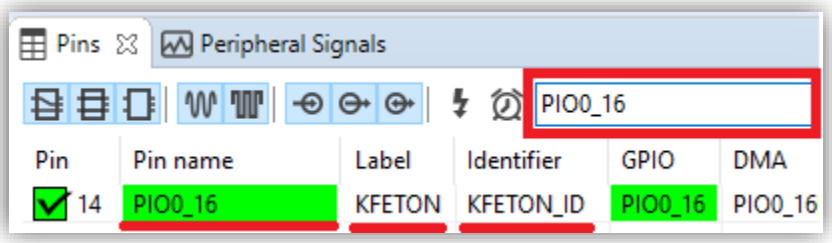
Modify the "Routed Pins" for BOARD_InitSilex2401Shield() function

You can re-configure the parameters (signal, label, identifier, etc.) directly in the routed pins table from the BOARD_InitSilex2401Shield section, or you can use the filter to find the current signals used by the Shield to disable them and then add the new signals for the Clickboard. The following steps describe the last one:

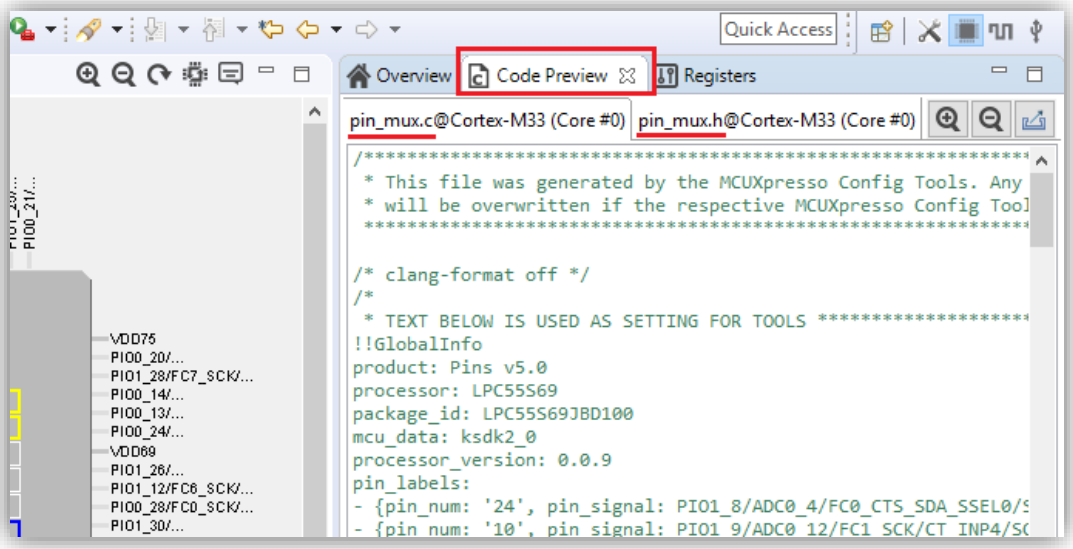
1. In the text filter field type PIO1_4 (currently used for KFETON in shield). Deselect the GPIO, cut the Label to store in the clipboard.



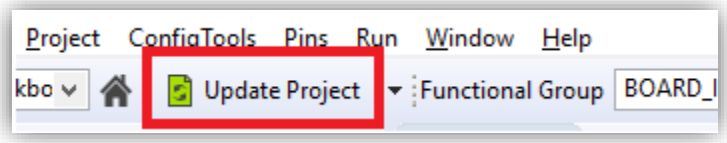
2. Clear the filter and type PIO0_16 (new signal to be used for KFETON in clickboard), select the GPIO function and paste the Label in the Label field.
3. Paste the same value with the "_ID" suffix into the Identifier field to clear the warning flags of the pins tool.



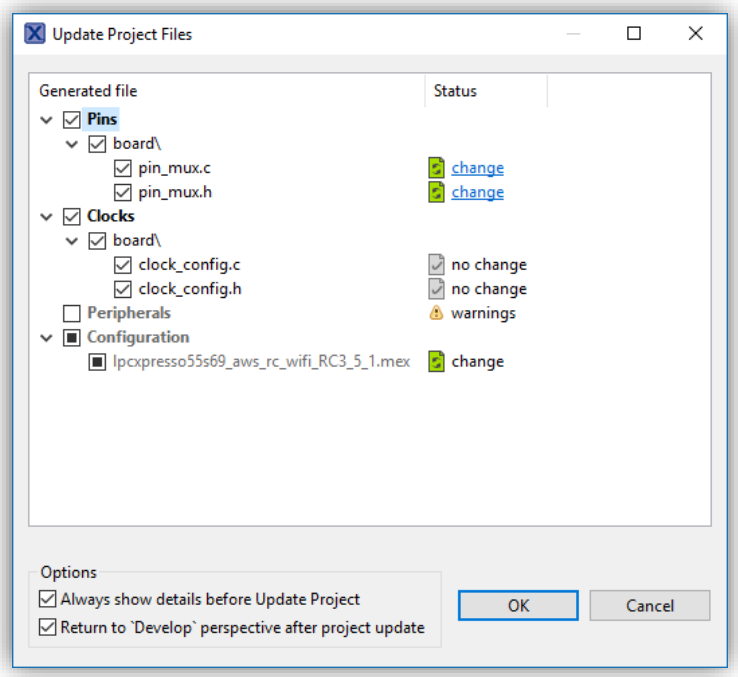
4. Repeat procedure with the PWRON and IRQ signals. Make sure your Routed Pins table looks the same as the image shown in previous section.
5. The previous step auto-generated the code used in pin_mux.c and pin_mux.h, you can confirm it by looking at the "Code preview" section as shown below.



6. Apply these changes to your project by pressing the “Update Project” button



7. A window will appear displaying the files that will be affected. You can analyze all the changes by clicking on the “change” link from the “Status” column. Click “OK” to apply the changes.



That’s it! You just re-routed a few pins to match another platform using the MCUXpresso Pin Configuration tool.
Note: You can navigate between the Developer view and the Pin configurator view by using the buttons from the upper right corner.

Additional modifications required for Silex Clickboard

After the pin configuration is updated, we just need to perform a couple of things:

1. Adjust **PWRON pin** used for the Clickboard:
   Open '\board**board.h**' and modify BOARD_INITSILEX2401SHIELD_PWRON_PIN to use the GPIO1_5 as shown below:

   ```
   #define BOARD_INITSILEX2401SHIELD_PWRON_PIN 5U
   ```

2. Adjust the interrupt pin used for the Clickboard:
   Open '\wifi_qca\port\shields\silex2401**wifi_shield_silex2401.h**' and modify WIFISHIELD_WLAN_PINT_CONNECT to use the GPIO1_18 as shown below:

   ```
   #define WIFISHIELD_WLAN_PINT_CONNECT (kINPUTMUX_GpioPort1Pin18ToPintsel)
   ```

3. Save your project.

Build and download

1. Click the "Build" button.
2. Download your application to your LPCXpresso55S69.