

Using MCUXpresso Config Tools with Custom Hardware

Generating a Board Support Configuration (.mex)

This section covers using the MCUXpresso Config Tools to create a custom configuration for a new board.

1. Create a new LPC845 based project and enable MCUXpresso Config Tools

1.1. Open MCUXpresso IDE, shortcut located on desktop.

1.2. Use the Quickstart Panel to create and “New project...” and select the LPC845 device.


- Expand “LPC84x” from the list of “SDK MCUs” Targets
- Select LPC845 from the menu, but do not select a photo of a board under “Available boards”.
- Select **Next**

1.3. Under “Device Packages” select the **LPC845M301JBD48** package.

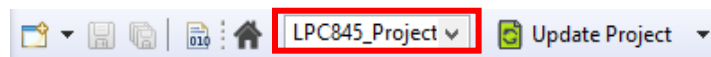
1.4. Under “driver” check the boxes next to:

clock, common, gpio, power, reset, swm, and usart.

1.5. Select **Finish**


1.6. Use the Project Explorer menu bar to launch Config Tools and Open Pins 

1.7. Ensure the correct project is from the drop-down menu in the tool bar



U1 LPC845_QFP48	
PIO0_0/ACMP_11/TDO	36 PIO0_00
PIO0_1/ACMP_12/CLKIN/TDI	24 PIO0_01
SWDIO/PIO0_2/TMS	10 PIO0_02
SWCLK/PIO0_3/TCK	8 PIO0_03
PIO0_4/ADC_11/TRSTN/WAKEUP	6 PIO0_04
RESET/PIO0_5	5 PIO0_05
	14 PIO0_06

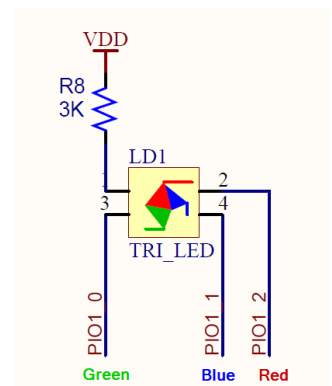
2. Use MCUXpresso Config Tools to mux pin signals

2.1. Ensure the Pins Tool is open using the perspective icons  located in the upper right:



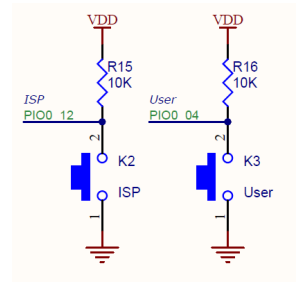
2.2. Add GPIOs Muxing for LEDs

- To enable the GREEN LED as GPIO, click on the pin name that matches the schematic (PIO1_0) under the “GPIO” column in the Pins window.
- Enter GREEN as the identifier for the pin.
- In the Routed Pins window, specify the Direction as Output (*optional*).
- Repeat for BLUE (PIO1_1) and RED (PIO1_2) LEDs.



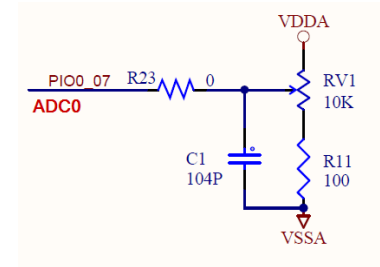
2.3. Add PINTs Muxing for Pushbuttons

- To enable to the USER button, click on the pin name that matches the schematic (PIO0_4) under the “PINT” column in the Pins window.
- Select (PINT, 0) in the resulting selection dialog
- Enter USER as the identifier for the pin.
- Repeat for ISP (PIO0_12) -> (PINT, 1) button.



2.4. Add ADC for onboard potentiometer

- To enable ADC, click on the pin name that matches the schematic (ADC_0) under the “ADC” column in the Pins window.
- Enter ADC as the identifier for the pin.



2.5. Optional pin muxing steps:

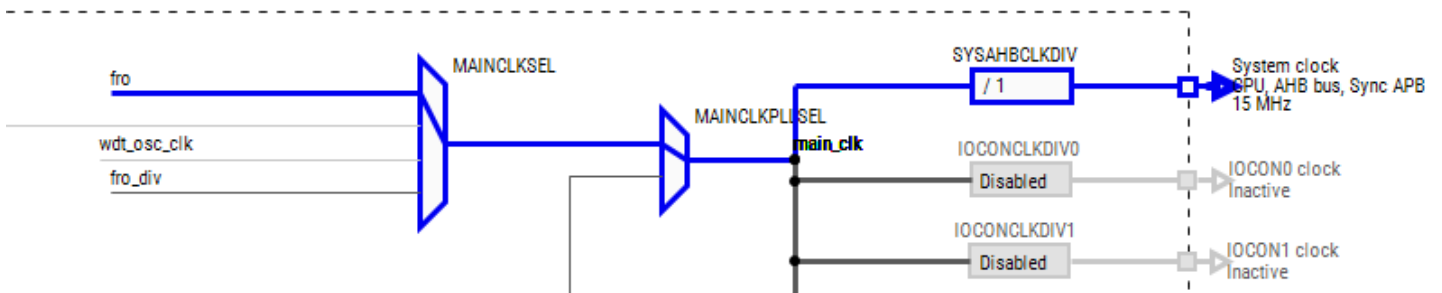
- Configure Pin 42 as ACMP_I5 (Label: CAPY)
- Configure Pin 9 as CAPT_X0 (Label: CAPX)

3. Use MCUXpresso Config Tools to change the clock configuration

3.1. Switch to the Clock Tools using the perspective icon in the upper right:



3.2. Select the “Clock Diagram” view and select the “System clock” to highlight the clock tree path



*note: you will need to scroll down to see the origin of the “fro” input signal

3.3. With the “System clock” highlighted, review the “Details” panel for specific clock configurations related to this clock path.

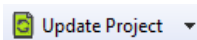
3.4. Change the **fro** setting to “**direct from FRO oscillator**”, this provides a 30MHz System Clock, but introduces an error with the PLL, which has a max frequency of 25MHz (see error message and tooltip).

3.5. Resolve the PLL error by changing the **SYSPLLCLKSEL** input source from “**fro**” to “**fro_div**”

3.6. Optionally change the **PLL Multiplier** to ***2** to set the SYSPLL clock output to the 30MHz maximum.


4. Update MCUXpresso IDE project with new Config Tools setting

4.1. Use the **Update Project** to save the config settings (.mex) and generate the pin and clock source files.



4.2. Review the generated source files.

5. Save the .mex file to an easy to access location for further use in the workshop.

- 5.1. Find the .mex file in the Project Explorer and save a copy by copying it to the Desktop (select the file and press ctrl + C, then paste on Desktop).  LPC845_Project.mex

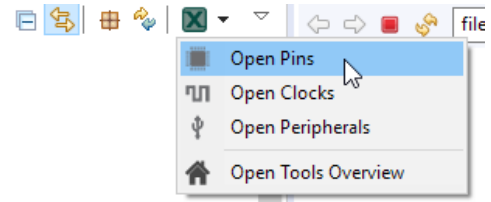
Apply Board Support Configuration to a New Project

This section covers applying a board support configuration to a blank new project.

6. Create another “New Project...” for the LPC845 device

- 6.1. Repeat steps 1.2 – 1.6 to create an additional project, but provide a unique **suffix** to the project name

- 6.2. Use the Project Explorer menu bar to enable the Config Tools and Open Pins



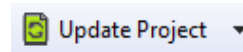
7. Apply the previously create configuration to this new project

- 7.1. Import .mex file created in Step 1 to the new project

- **File → Import.**
- Under MCUXpresso Config Tools select “Import Configuration (*.mex)” and click **Next**.
- Select the .mex file created previously as the Existing configuration, and the name of the current project as the Target Project. Click **Finish**.

- 7.2. Observe the pins that are activated and notice that the configuration is the same as in the previous case. Open the Clock Tool to see that the System clock has also been updated to 30 MHz.

- 7.3. Save and **Update Project** with the Config Tools settings.



- 7.4. Examine resulting project files.

8. Build and Debug application on LPC Breakout Board

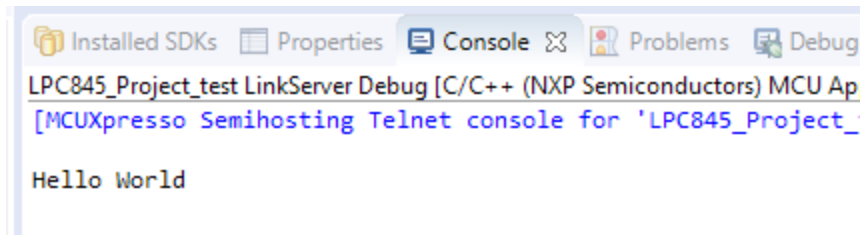
- 8.1. From the Quickstart Panel select “Build”

- 8.2. From the Quickstart Panel select “Debug” and follow resulting dialog windows to setup the correct debug interface.

- 8.3. Use the Debug control buttons to “Resume” the application from the initial main() breakpoint.



- 8.4. “Hello World” should output on the console.



Apply Board Support Configuration to an Existing Example Project

This section covers applying a customized board support configuration to an existing SDK example project.

9. Open / Import an existing MCUXpresso SDK Example project for an NXP development board

9.1. Use the Quickstart Panel to Import SDK Example Import SDK example(s)...



- Select “LPC84x” from the list of “SDK MCUs” targets
- Locate and select the picture of the **LPCXpresso845max** board. Select **Next**.
- Under “**demo_apps**” select the example called “**led_blinky**”.
- Select **Finish**.

10. Convert project to correct device package (impacts to IDE, SDK, and Config Tools)

10.1. Convert IDE project to correct Package (Note, the IDE doesn't really care, this just helps avoid other conflicts)

- Expand **lpcxpresso845max_led_blinky** -> **Project Settings** -> **MCU**, note chip and package values
- Right-click **MCU**, select **Change package** from the menu, then select **LPC845M301JBD48**

10.2. Enable Pins Tools

- Use the Project Explorer menu bar to launch  Config Tools and Open Pins.
- Use the Switch Package button toolbar in the top right corner of the Package window to select the “LQFP 48 package”. 
- Optionally, you can right-click the error message in the Config Tools problem window to switch the package type directly.

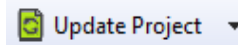
11. Apply the previously create configuration to this new project

11.1. Import .mex file created in Step 1 to the new project

- **File → Import.**
- Under MCUXpresso Config Tools select “Import Configuration (*.mex)” and click **Next**.
- Select the .mex file created previously as the Existing configuration, and the name of the current project as the Target Project. Click **Finish**.

11.2. Observe the pins that are activated and notice that the configuration is the same as in the previous case. Open the Clock Tool to see that the System clock has also been updated to 30 MHz.

11.3. Save and **Update Project** with the Config Tools settings.



11.4. Examine resulting project files.

12. Edit source code to “connect” new LED signal

12.1. Open the pinmux.h file under the board folder in the example project.

12.2. Scroll down to find the definitions for each of the LEDs enabled earlier. Find:

```
#define BOARD_INITPINS_GREEN_PORT 1U
```

```
#define BOARD_INITPINS_GREEN_PIN 1U
```

12.3. In the led_blinky.c file, find the definitions:

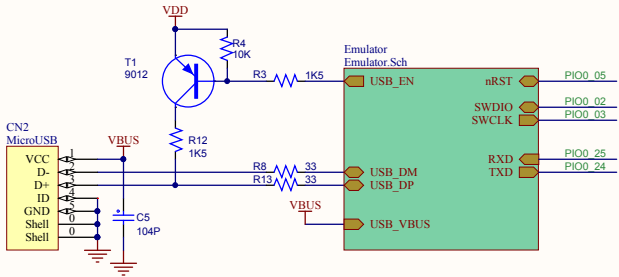
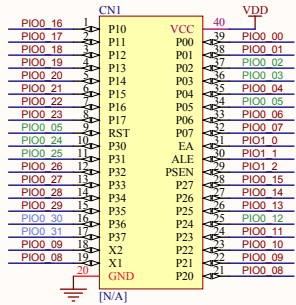
```
#define BOARD_LED_PORT 0U
```

```
#define BOARD_LED_PIN 12U
```

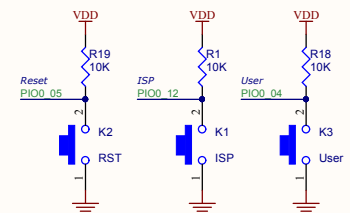
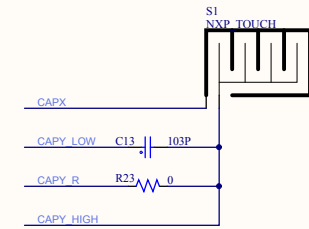
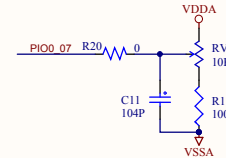
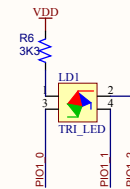
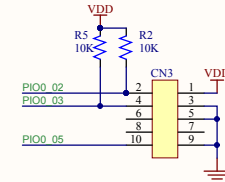
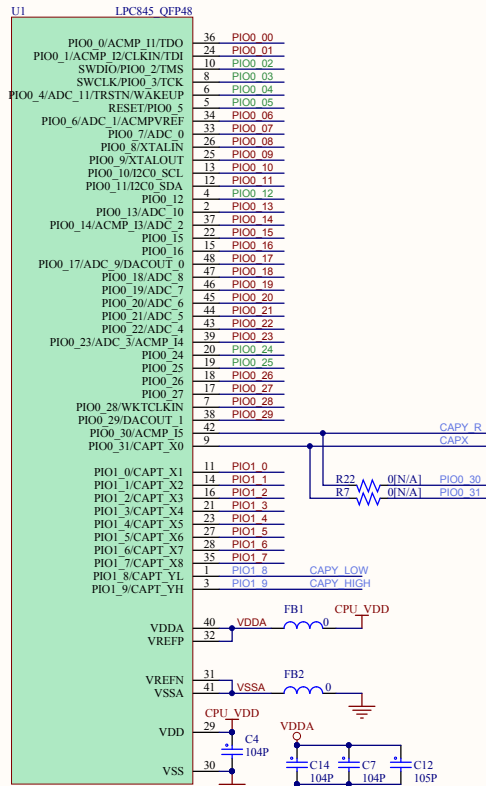
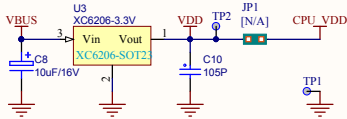
12.4. Copy the variable names **BOARD_INITPINS_GREEN_PORT** and **BOARD_INITPINS_GREEN_PIN** from the pinmux.h file to replace the hard coded values of the definitions in led_blinky.c (0U and 12U respectively).

12.5. Build, Debug, and Run the application to see the green LED flashing.

12.6. End the debug session using the Terminate (red stop) button.



POWER & Current measurement



Emulator

