3. Refer to Chapters 10 & 13 in the FRDM-K64F User Manual to find out what pins the three built-in LEDs and one built-in push button are connected to. The pushbutton we will be working with is SW2 and is connected to PTC6. The LEDs are referenced as Blue, Red and Green and are accessible through PTB21, PTB22, and PTE26. The K64 Documentation can be found in MyCourses under Reference → FRDMK64UG.

4. The pin naming convention is as follows: "PTA0 = Port A, Pin 0". Copy down the pin numbers for the LEDS and push button (also make sure these are documented in your code):

Table 1.1: Table for Recording Ports & Pins for LEDs

	Port	Pin	Name
Red LED	A	22	PTA 22
Green LED	E	26	PTE 26
Blue LED	A	21	PTA 21
Switch 2	C	6	PTC6

5. Enable the LED and button control signals by setting the Pin Control Register (PORTx_PCRn) for each LED pin to GPIO Pin Mux Control.

   (a) Assign the LED pins found above to the value PORT_PCR_MUX(1), which enables GPIO mode.

   (b) For example, "PORTA_PCR11 = PORT_PCR_MUX(1);" wakes up the 11th bit of PORT A to work in GPIO mode.

6. Set the bits of each Port to input/output mode according to the requirements of the lab. If set to output, the pin can then be set, cleared, or toggled. If set to input, the pin can be read. (Chapter 55 of K64 reference manual).

   (a) Configure GPIO pins for Input or Output using the GPIO Data Direction Register (PDDR).

   (b) Make sure to use the correct Port for each pin.

   (c) To load data into the direction register, use a simple shift left operation for each of the desired port pins. [e.g. GPIOA_PDDR = (1 ≪ n), where n is the LED pin from PORTA that we want to set to output. You can OR as many of these together in one line as GPIOA_PDDR = (1 ≪ 11) — (1 ≪ 12) to set pins 11 and 12 as output.]

7. Write code to turn on the LEDs and to control them using the pushbutton. (Chapter 55 of K64 reference manual).

   (a) Drive the output to the LEDs high by writing a '1' to the GPIOx_PDOR register for each LED, initializing them to the 'off' state. Keep in mind that the LEDs are active low.

   (b) Note, when controlling the GPIO, we will be using five commands:

      (i) GPIOx_PDOR = <value>; this sets the entire register to <value>

      (ii) GPIOx_PSOR = <value>; this sets all bits in the register for which <value> has a '1' bit (in binary)

      (iii) GPIOx_PCOR = <value>; this clears all bits in the register for which <value> has a '1' bit (in binary)