

Code Explanation
Code-file Name: Meow.c
Done By: Laith Abdelrazeq 101110866

The architecture of my code consists of four main functions. The first function is **"stateChange"**. **stateChange** is a function that is responsible for toggling the current state. **stateChange** takes one parameter (**value**). If the inputted parameter == 1, then the **stateChange** function toggles the state to the next state. If the inputted parameter == 2, then the **stateChange** function toggles the state to the previous state. **stateChange** works by having the states in a loop. For example, toggling to the next state 4 times starting at state **0** would result in the following:

0 -> 1 -> 2 -> 3 -> 0

This also applies to previous but in the opposite direction. For example, toggling to the previous state 4 times starting at state **3** would result in the following:

3 -> 2 -> 1 -> 0 -> 3

This logic is done using 4 (if/else if) statements that check for the inputted parameter and current state to determine the direction of toggling. The second two functions are **the interrupt service routine functions**. The first interrupt service routine checks for interrupts from the **UART** for specific keys on the pressed on the **keyboard**. For example, using putty to view the state of the UART, if the user presses the (**greater than**) button, this calls the **stateChange** function to toggle next. If the user presses the (**less than**) button, the **stateChange** function is called to toggle previous. The second interrupt service routine checks for interrupts that occur on the **board** from **port1**, pins **P1.1** and **P1.4**. These two interrupt service routine functions call the **stateChange** function with an input of (**1**) for toggling to "next" state and an input of (**2**) for toggling back to the "previous" state. The last function in my code is the **main**. In main, I configure the UART and its pins. This section relied on the "echo" example given to us. Next, I configure the GPIO ports and pins. Of course, I start by disabling the watchdog timer. Then setup ports 1&2 as GPIO and then I initialize the pins that I decided to use. Then, I configure my interrupts and enable them globally.

Running the code should result in the following state diagram:

State	"0" Both LED's Are OFF	"1" One LED is ON	"2" One LED is ON	"3" Both LED's Are ON
Toggle Next ">"	➔ 1	➔ 2	➔ 3	➔ 0
Toggle Previous "<"	➔ 3	➔ 0	➔ 1	➔ 2