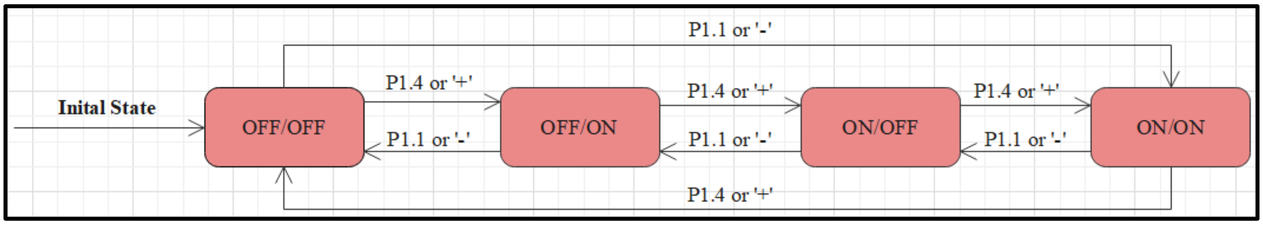
Ahmed Abdelrazik 101103048

SYSC 3310 Project

Code Description

The idea of this project is to be able to communicate with our boards through a virtual UART Port. The main purpose of this project is to send a message to the board through the UART and this message changing the state of the board.

To do that I Configured the UART Port as well as all the GPIOs needed (the two buttons and the LEDs). The Board has 4 states; OFF/OFF (both LEDs off), OFF/ON (LED 2.0 is on), ON/OFF (LED 1.0 is on), ON/ON (Both LEDs on). When Button P1.4 is pressed the State changes to the next state, when P1.1 is pressed the state changes to the previous state. The board initially starts at state 0 (OFF/OFF) and switches states according to the State Diagram below.



The Code consists of 6 functions and the main function, 2 functions that configure all the required ports (UART & GPIO); **config_UART()** & **config_GPIO()**. 2 interrupt handler functions that change the state of the board when the interrupt is flagged: **EUSCIA0_IRQHandler()**, **PORT1_IRQHandler()**. 2 functions that change the state of the board; **changeState()**, and a function that takes an action upon the current state of the board; **takesAction()**.

The states used in the code are 0x30, 0x31, 0x32, 0x33. The reason theses values are chosen is because they correspond the ASCII values of number 0, 1, 2, 3. So when we transmit the value of the variable curr_state to the UART through the transmitter register we print the number of the current state, and we can now tell our current state by looking at the output of the UART.

The way the changeState function is setup it will change the state according to the button changed until it reaches the last state where it will go back to the first state if we click forward button or we enter a ‘+’ to the UART and vice versa when we are at the first state; it will loop back to the last state if we click the backward button or we enter a ‘-‘ to the UART.