ECE 2534 Lab2 Report

The R2D2 Terminal

Section 1

In this Lab, each student is asked to develop a application that can sent and receive characters from labtop. Specifically, there is a terminal application called MobaXterm, and it is connected to the UART on the MSP432P4. Every time when I type a character on the keyboard, it will be transmitted to the UART through the MobaXter, and the UART will send back the character and display on MobaXterm. My design could work properly as described above. In addition, my design could also meet several specific requirements. The first is that the characters on LCD screen is written line by line and top from bottom, and after the full screen is written, the next character overwrite the character at top left position and continue. The second is that there are two inline commands #fn and #bn, to control the foreground and background color, where n is a number from 0 to 7 representing a color. However, the inline commands does not show on either the LCD or MobaXterm. The third specific requirement is that there are two buttons that could accomplish two things. The top button could display the current baud rate of the UART, the foreground and background color information and the total count of characters that has been received. The bottom button simply change the baud rate of the UART. My design also meet the bonus requirement. Specifically, it could display different color of the LED when corresponding types of character is received.

Section 2

There are two FSM in my design. The first one is the debounce FSM, and FSM transition graph is shown in Figure 1. This state machine describes the action of pushing and release button with a debounce timer. The state starts with stable0 and if the button is pressed, the timer will start and the state change to trans0. However, if the button is released during this process, it will return to stable0. If the button is continuously been pressed and the timer expired, the state will change to stable1, which means the button is pressed totally. Then if the button is released, then the state will change to trans1 and the timer starts. If the button is continuously been released, and timer expired, the state will change to stable0.

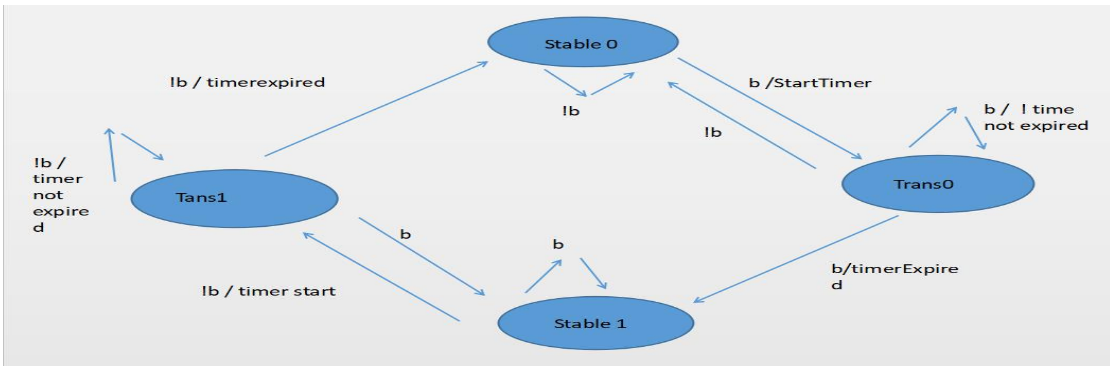
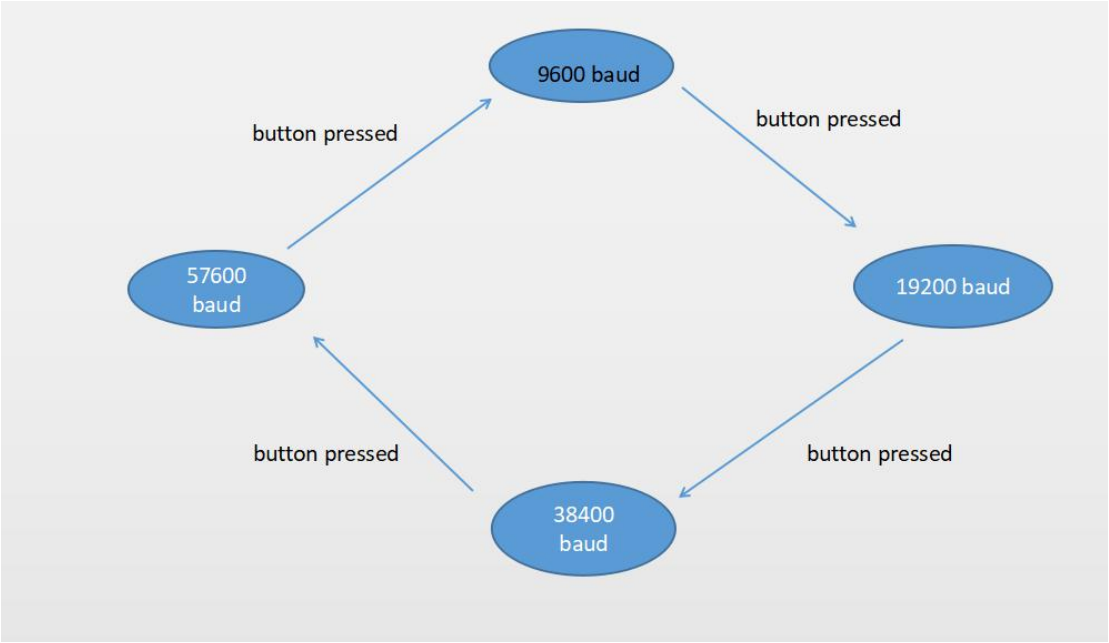


Figure 1

The second State machine I used is for the changing baud rate, the state diagram is shown below in Figure 2. There are four state for four different baud rate. Every time, if the button is pressed, the state will go with order “9600 -> 19200 -> 38400 -> 57600 -> 9600”



Section 3

1. void changeUART19200() : Change the baud rate to 19200
2. void changeUART38400(): Change the baud rate to 38400
3. void changeUART57600(): Change the baud rate to 57600
4. void ColorLEDOFF(): Turn off the LED
5. char changeColor(char c, char ColorArray[2]) :Change the color of background and foreground
6. Count countAndPrint(Count count,char c): count the character and print character
7. Count SkipPrint(Count count, char c): count the character if there is # follow with a character that is not f or b
8. Count printStatus(int countChar): print the status message
9. UARTBaudRate_t ChangeBaud(): change the baud rate
10. void ExtraCredit(char c): change the LED color if the required character is typed.