CECS 285: Lab 8

OBJECTIVES:

- Use Serial Communication to control LED Brightness via PWM

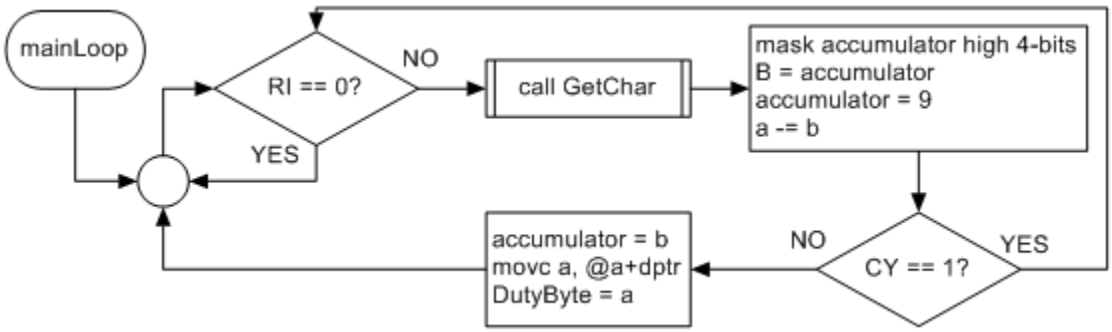
Write a program to that interfaces with the 8051 through the serial console so that keyboard strokes typed into the serial console be used to control the brightness of 8051 LEDs. Each portion of the program to be created is outlined as follows:

1. At the very beginning of your program listing, declare the following variables:

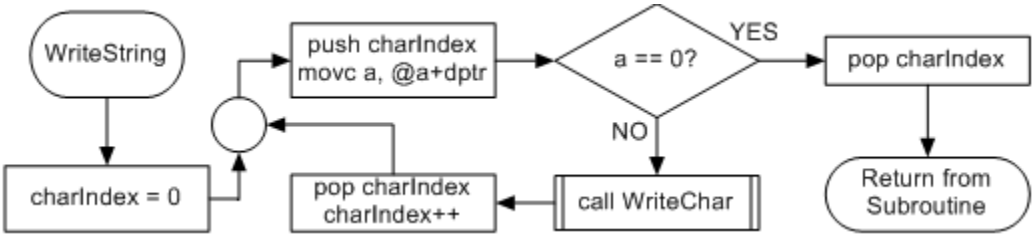
```
charIndex  EQU  0xE0 ;alias variable for Accumulator
DutyByte   EQU  0x30
DutyFlag   BIT  0x00
```

2. The Timer 0 Interrupt Service Routine from the previous lab will be used for PWM generation so be sure a jump to the Timer 0 ISR is placed at the correct interrupt vector
3. The **main** program section will begin at ROM location 0x30 where the following initialization steps must be performed:
    a. Configure Port 1 as output (*or whichever port is connected to LEDs, but not P3!*)
    b. Configure Timer 1 to mode 2, Timer 0 to mode 1
    c. Configure Timer 1 to operate at 9600 baud
    d. Configure Serial communication for 8-bit data, 1 start bit, 1 stop bit, no flow control
    e. Start Timer 1 to enable serial communication
    f. Load the address of Welcome message into the data pointer
        i. Welcome message is defined at step 9 in this procedure
    g. Call the WriteString subroutine to write the welcome message to serial console
    h. Load the address of the DutyValues lookup table into the data pointer
    i. Load Timer 0 high register with the constant value 0xFF
    j. Load Timer 0 low register with the constant value 0x60
    k. Enable Timer 0 Interrupt
    l. Enable Global Interrupt
    m. Start timer 0

4. Next create the **mainloop** section which is defined by the following flow chart:



5. After the main program loop, put the Timer 0 Interrupt service routine code from your previous lab
6. Next put the **GetChar** subroutine presented in class in your program listing
7. Next create the **WriteString** Subroutine according to the following flow chart:



8. Next put the **WriteChar** subroutine presented in class in your program listing.
9. Finally at the end of your program listing, define lookup tables for the Welcome message string that will be displayed in the serial console and the DutyValues constant array that will be used for PWM delay timing generation

```
org   0x200
Welcome:     DB     "Enter a value 0 through 9 for LED brightness control: ",0
DutyValues:  DB     0x60, 0x70, 0x80, 0x90, 0xA0, 0xB0, 0xC0, 0xD0, 0xE0, 0xF0
end
```

Testing your program:

1. Once your program has been written and builds without errors/warning, create a hex file and load it onto your 8051 hardware.
2. Next open a serial terminal application (if you've not done so already) and ensure it is configured to be connected with the COM port where your 8051 is connected (the serial port on the back of a lab computer is COM 1, otherwise if you are using a USB/Serial converter the COM port is likely the highest numbered COM port.
    a. (COM 4 or COM 5 is typical but it could be COM 10, COM 12, etc.).
3. Once the serial terminal is configured, put the 8051 hardware into RUN mode and press the Reset button and the Welcome message should appear in the serial terminal
4. Click within the serial terminal window and then your keyboard button presses will be sent over serial communication to the 8051, number key presses should predictable control LED brightness

Deliverables:

- Your commented assembly language program source code. The following commenting criteria must be met
    - Comment each configuration instruction
    - Comment blocks of code otherwise