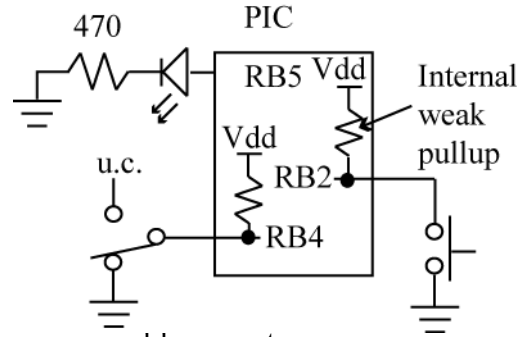


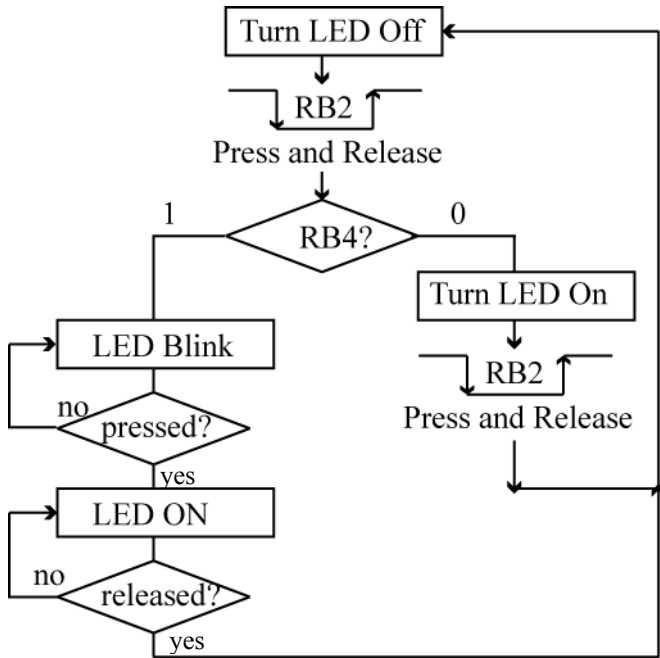
You may use only the provided reference materials.

Part I: (60 pts)

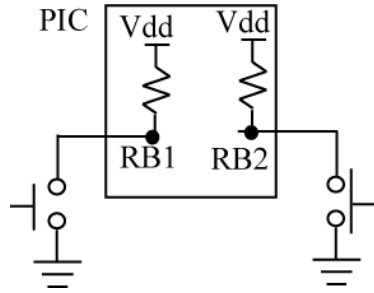
- a. (5 pts) Write *C* code that configures PORTB for the IO as shown below. The internal weak pullup must be enabled.



- b. (15 pts) Assuming the IO configuration of the previous problem, write a *while(1){}* loop that implements the LED/Switch IO state machine shown below. Either use a *switch()* statement approach or a *if-then-else* approach. Assume you have available the *DelayMs()* function.



- c. (10 pts) For the switch configuration below, assume both inputs have been configured to generate rising edge triggered inputs (high priority). Write an ISR that increments a variable named *edge_count* by 2 if the INT1 interrupt occurs, or increments *edge_count* by 1 if the INT2 interrupt occurs. After an interrupt occurs, disable that interrupt using its enable flag, clear the interrupt flag bit, and set the semaphore variable *switch_pressed* to a '1'.



- d. (10 pts) Assuming the ISR of (c), write a *main()* function that enables INT1, INT2 interrupts to be rising edge triggered, priorities disabled. Configure both pins as inputs. Then enter an infinite *while(1){}* loop that waits for the *switch_pressed* variable to become a '1' value; after *switch_pressed* is set by the ISR print a message that says 'switch pressed', clear the *switch_pressed* variable to a '0', and re-enable both interrupts via their interrupt enable flags.

- e. (5 pts) Write C code that implements the *putch(char c)* function (sends one character to the serial port). No interrupts are enabled.
- f. (5 pts) Write a C code fragment to go in the beginning of *main()* to detect a software reset condition and prints the message 'Software Reset'. Set or clear the appropriate status bit so that a software reset is not falsely detected on the next non-software-reset.
- g. (10 pts) Write a C code function *char getdata()* that waits for data to become available in a circular buffer name *buf*, takes data out of the buffer and returns it. Assume the buffer has a maximum of 16 characters, and pointers named *tail* and *head*. The *head* pointer is used for placing data into the buffer. Assume an ISR is placing data into the buffer.

Part II: (40 pts) Answer 13 out of the next 15 questions. Cross out the 2 questions that you do not want graded. Each question is worth 3 pts.

1. Give the value of one bit time in microseconds for a baud rate of 4800.
2. Given a voltage of 5 V, a clock freq of 40 MHz, and a current consumption of 20 mA, what is the new current consumption predicted by theory if the voltage is reduced to 3 V and the clock frequency to 10 MHz?
3. Draw an open-drain output..
4. Given a data format of 7-bit + odd parity, what is the value of the parity bit if the data is 0x2A?
5. What is the SBBRG value for a baud rate of 38,400 assuming an FOSC of 20 MHz and high speed mode?

14. What mechanism was used in the square root program of experiment 8 that allowed it to deal with bursts of input characters. Be specific.

15. Given an advantage of serial IO. Give a disadvantage of serial IO. Your advantage cannot be equal to the NOT(disadvantage).