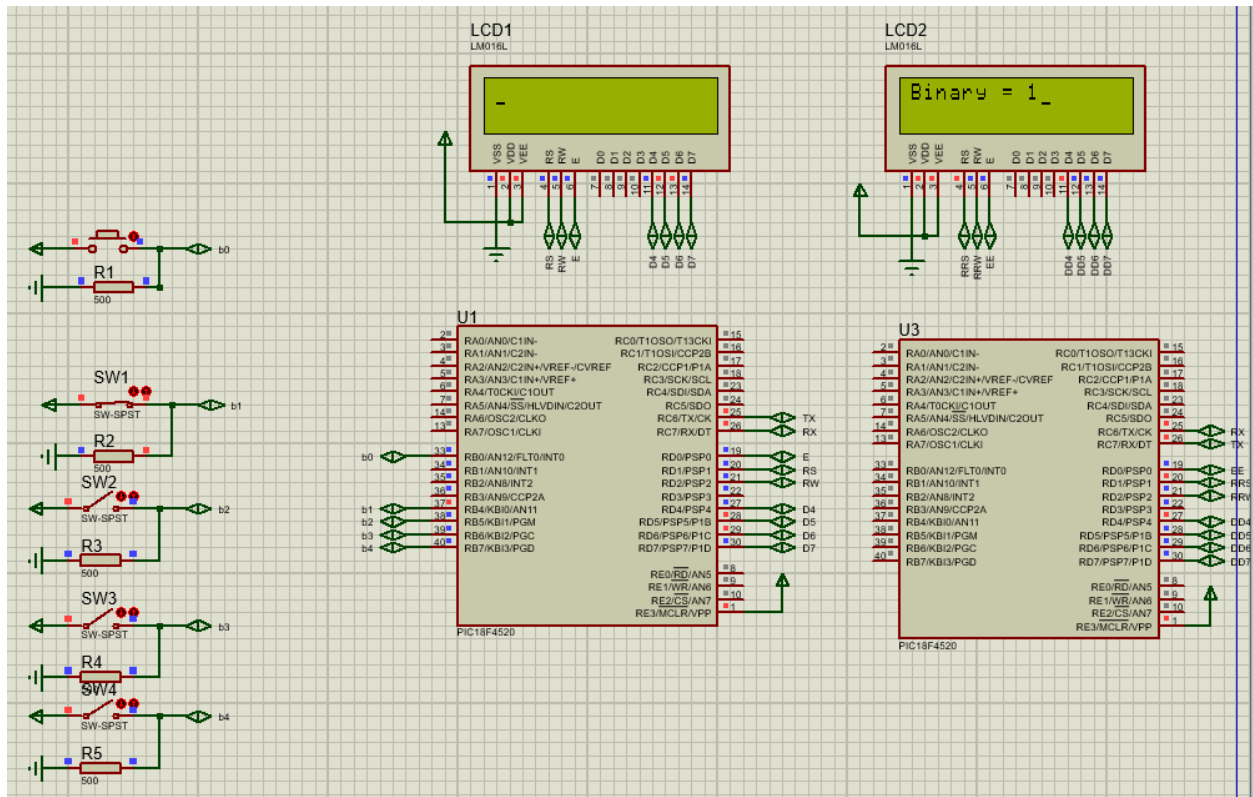
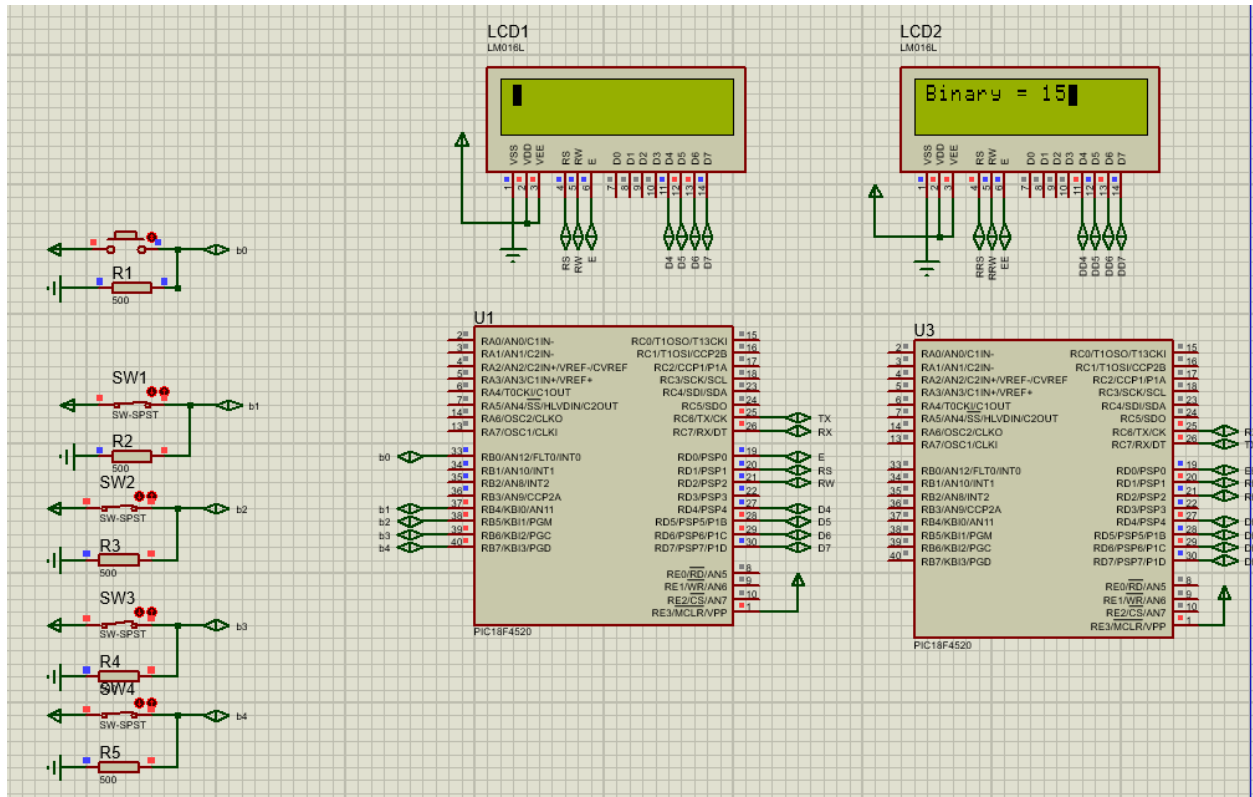


# Final Bench Exam – Switches to Binary USART – James Martin



MASTER:

```
1  #include <18f4520.h>
2  #use delay( clock = 20000000 ) // Set clock to 20MHz
3  #fuses HS, NOWDT, NOLVP
4  #include "../Library/myLibrary.h"
5  #include "../Library/modifiedlcd.h"
6
7
8  #INT_EXT
9  void int_ext_isr() {
10
11     if( input( PIN_B4 ) == 1 && input(PIN_B5) == 0 && input(PIN_B6) == 0
12         && input(PIN_B7) == 0) {
13         *TXREG = 1;
14     }
15     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 1 && input(PIN_B6) == 0
16         && input(PIN_B7) == 0) {
17         *TXREG = 2;
18     }
19     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 1 && input(PIN_B6) == 0
20         && input(PIN_B7) == 0) {
21         *TXREG = 3;
22     }
23     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 0 && input(PIN_B6) == 1
24         && input(PIN_B7) == 0) {
25         *TXREG = 4;
26     }
27     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 0 && input(PIN_B6) == 1
28         && input(PIN_B7) == 0) {
29         *TXREG = 5;
30     }
31     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 1 && input(PIN_B6) == 1
32         && input(PIN_B7) == 0) {
33         *TXREG = 6;
34     }
35     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 1 && input(PIN_B6) == 1
36         && input(PIN_B7) == 0) {
37         *TXREG = 7;
38     }
39     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 0 && input(PIN_B6) == 0
40         && input(PIN_B7) == 1) {
41         *TXREG = 8;
42     }
43     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 0 && input(PIN_B6) == 0
44         && input(PIN_B7) == 1) {
45         *TXREG = 9;
46     }
```

```

46     }
47     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 1 && input(PIN_B6) == 0
48         && input(PIN_B7) == 1) {
49         *TXREG = 10;
50     }
51     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 1 && input(PIN_B6) == 0
52         && input(PIN_B7) == 1) {
53         *TXREG = 11;
54     }
55     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 0 && input(PIN_B6) == 1
56         && input(PIN_B7) == 1) {
57         *TXREG = 12;
58     }
59     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 0 && input(PIN_B6) == 1
60         && input(PIN_B7) == 1) {
61         *TXREG = 13;
62     }
63     else if( input( PIN_B4 ) == 0 && input(PIN_B5) == 1 && input(PIN_B6) == 1
64         && input(PIN_B7) == 1){
65         *TXREG = 14;
66     }
67     else if( input( PIN_B4 ) == 1 && input(PIN_B5) == 1 && input(PIN_B6) == 1
68         && input(PIN_B7) == 1) {
69         *TXREG = 15;
70     }
71 }
72
73 main() {
74     // Initialize LCD
75     lcd_init();
76
77     // Setup TX/RX
78     *TRISC |= 0x80;           // C7 = RX, C6 = TX
79     TXSTA -> TXEN = 1;       // Transmit enable
80     TXSTA -> SYNC = 0;       // Asynch mode
81     RCSTA -> SPEN = 1;       // Serial enable
82     RCSTA -> CREN = 1;       // Receiver enable
83     TXSTA -> BRGH = 0;
84     BAUDCON -> BRG16 = 0;
85     *SPBRG = 31;
86     PIE1 -> RCIE = 1;        // Receiver interrupt enable
87
88

```

```
88
89 // Global / peripheral enable
90 INTCON -> GIE = 1; // Global
91 INTCON -> PEIE = 1; // Peripheral
92
93 ADCON1 -> PCFGx = 15; // ALL analog
94 *TRISB = 0x01;
95 INTCON2 -> INTEDG0 = 1; // Trigger on raising edge
96 INTCON -> INT0IE = 1;
97 INTCON -> PEIE = 1;
98 INTCON -> GIE = 1;
99
100 while( 1 ) {
101 }
102 }
```

SLAVE:

```
1  #include <18f4520.h>
2  #use delay (clock = 20000000)
3  #fuses HS, NOWDT, NOLVP
4  #include "../Library/myLibrary.h"
5  #include "../Library/modifiedlcd.h"
6
7  // USART Receive
8  #INT_RDA
9  void int_rda_isr() {
10     printf(lcd_putc, "\fBinary = %d", *RCREG);
11 }
12
13 main() {
14
15     // Initialize LCD
16     lcd_init();
17
18     // Setup TX/RX
19     *TRISC |= 0x80;           // C7 = RX, C6 = TX
20     TXSTA -> TXEN = 1;       // Transmit enable
21     TXSTA -> SYNC = 0;       // Asynch mode
22     RCSTA -> SPEN = 1;       // Serial enable
23     RCSTA -> CREN = 1;       // Receiver enable
24     TXSTA -> BRGH = 0;
25     BAUDCON -> BRG16 = 0;
26     *SPBRG = 31;
27     PIE1 -> RCIE = 1;        // Receiver interrupt enable
28
29     // Global / peripheral enable
30     INTCON -> GIE = 1;       // Global
31     INTCON -> PEIE = 1;      // Peripheral
32
33     while( 1 ) {
34
35     }
36
37 }
```