

EES 1031 Electronics and Embedded Systems Lab

Academic Year 2024/2025

Practical 03 Interfacing and Driving Two-Digit Seven Segment LED Displays

Name: GLNN MANJUSRI
Index No: AS20240927

PART A

Activity A-01

Identify the resistor values of the given resistors. Identify the configuration (common cathode or common anode) of the given SSDs. Interface a single SSD with an Atmega328p microcontroller. (Here you have to be careful to manage the given space in the breadboard to use 2 SSDs, resistors, 2 transistors, and 2 push buttons) Resistors were $1K\Omega$ and 330Ω . Displays were in common cathode configuration.

Activity A-02

Program the microcontroller to turn on each segment in the SSD with a 500 ms delay and then turn off each segment following the same order.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 int main()
5 {
6     for (;;)
7     {
8         DDRD = 0xFF;
9         PORTD = 0;
10
11         for (uint8_t i = 0; i < 7 ; ++i) {
12             PORTD |= (1 << i);
13             _delay_ms(500);
14         }
15
16         for (uint8_t i = 0; i < 7 ; ++i) {
17             PORTD &= ~(1 << i);
18             _delay_ms(500);
19         }
20     }
21 }
```

Activity A-03

Define a 2D array (int digit [10][7]) to represent the 0-9 digits in the SSD.
Program the microcontroller to display 0-9 digits.

Activity A-04

Identify the Base, Emitter, and Collector terminals of the provided NPN transistor BC547.

Interface the transistor with the SSD in order to drive the SSD via the transistor.

Program the microcontroller to display the 0-9 digits.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 static const char codes[] = {
5     0b1111110,
6     0b0110000,
7     0b1101101,
8     0b1111001,
9     0b0110011,
10    0b1011011,
11    0b1011111,
12    0b1110000,
13    0b1111111,
14    0b1111011
15 };
16
17 int main()
18 {
19     DDRD = 0xFF;
20     for (;;) for (int i = 0; i < 10 ; ++i) {
21         PORTD = codes[i];
22         _delay_ms(500);
23     }
24 }
```

PART B

Activity B-01

Interface the second SSD to the microcontroller which must be driven through a transistor.

Activity B-02

Program the microcontroller to display the two-digit (tens and one's position) numbers 0-99. Use for loops, arrays and user define functions as well. Hint: may useful division ($7/3 = 2$) and modulo ($7\%3 = 1$) operators.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 static const char codes[] = {
5     0b1111110,
6     0b0110000,
7     0b1101101,
8     0b1111001,
9     0b0110011,
10    0b1011011,
11    0b1011111,
12    0b1110000,
13    0b1111111,
14    0b1111011
15 };
16
17 void show_number(uint8_t n) {
18     if (n > 99) return;
19     for (char j = 0; j < 99; ++j) {
20         _delay_ms(10);
21         PORTB $= 1 << (j & 1);
22         PORTD = codes[j & 1 ? n / 10 : n % 10];
23     }
24 }
25
26 int main()
27 {
28     DDRD = 0xFF;
29     DDRB = 0b11;
30
31     for (;;) for (int i = 0; i < 100 ; ++i) {
32         show_number(i);
33         _delay_ms(10);
34     }
35 }
```

Activity B-03

Modify the above code where turn on one SSD at a time when displaying two-digit number.

Hint: Use transistor driven SSD to turn on/off each segment with a time delay.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 static const char codes[] = {
5     0b1111110,
6     0b0110000,
7     0b1101101,
8     0b1111001,
9     0b0110011,
10    0b1011011,
11    0b1011111,
12    0b1110000,
13    0b1111111,
14    0b1111011
15 };
16
17 void show_number(uint8_t n) {
18     if (n > 99) return;
19     for (char j = 0; j < 99; ++j) {
20         _delay_ms(10);
21         PORTB &= 1 << (j & 1);
22         PORTD = codes[j & 1 ? n / 10 : n % 10];
23     }
24 }
25
26 int main()
27 {
28     DDRD = 0xFF;
29     DDRB = 0b11;
30
31     for (;;) for (int i = 0; i < 100 ; ++i) {
32         show_number(i);
33         _delay_ms(10);
34     }
35 }
```

Activity B-04

Build a up/down counter using two pushbuttons.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 static const char codes[] = {
5     0b1111110,
6     0b0110000,
7     0b1101101,
8     0b1111001,
9     0b0110011,
10    0b1011011,
11    0b1011111,
12    0b1110000,
13    0b1111111,
14    0b1111011
15 };
16
17 void show_number(uint8_t n) {
18     if (n > 99) return;
19     for (char j = 0; j < 99; ++j) {
20         _delay_ms(10);
21         PORTB &= 1 << (j & 1);
22         PORTD = codes[j & 1 ? n / 10 : n % 10];
23     }
24 }
25
26 int main()
27 {
28     DDRD = 0xFF;
29     DDRB = 0b11;
30
31     for (;;) for (int i = 0; i < 100 ; ++i) {
32         show_number(i);
33         _delay_ms(10);
34     }
35 }
```