Experiment 3: Programming Arduino in Assembly Language

Due date : 29.11.2020 23:59

Res. Asst. Kadir Ozlem
kadir.ozlem@itu.edu.tr

1 Introduction

In this experiment, you will generate various led patterns and seven-segment display output that can be created in Figure 1, depending on the state of the push-buttons using assembly language. You can look up blink led example in Assembly Language. You can access the datasheets of the components using ATmega328P, CD4511, and seven-segment display documentation links. You can attain the design via this link.

- **You can not use any C language instruction.**
- **You can only use Assembly Language for this experiment.**
- **You must write your code between BEGIN and END statements in the ASM functions.**
- **You can not use any loop statement and recursive functions except main LOOP label and delay functions.**
- **You can not use block programming provided by tinkercad.**
- **You can not use constant numbers for patterns.**
- **You must add comments for each line. Otherwise, you will get zero point.**

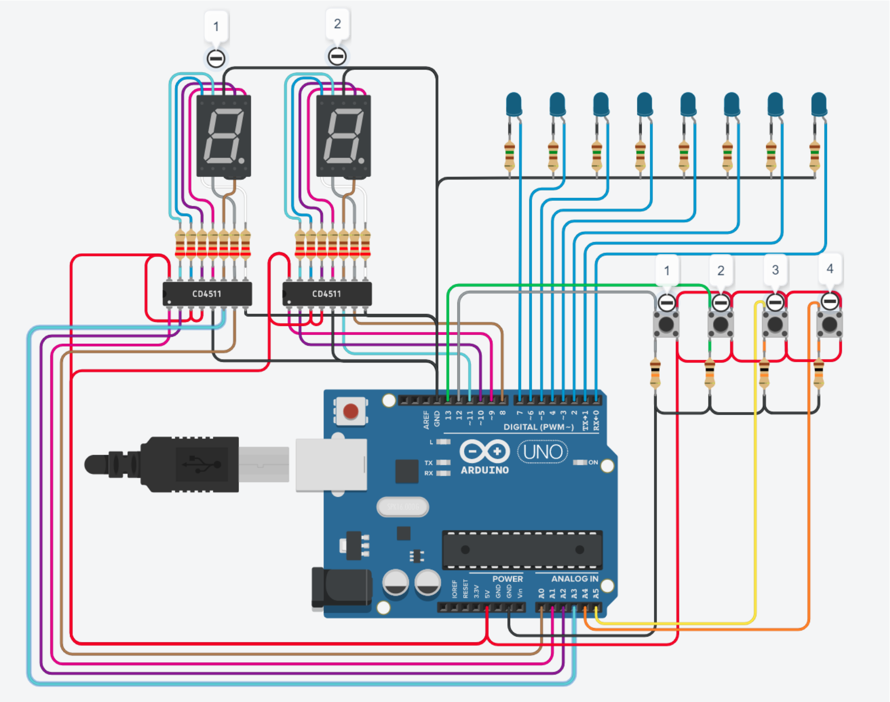

Figure 1: Circuit diagram.

2 Part 1

In this part, you will write an assembly program that generates the sequence over LEDs in Table 1.

time	led pattern
0	0000000**1**
1	000000**1**0
2	00000**1**00
3	0000**1**000
4	000**1**0000
5	00**1**00000
6	0**1**000000
7	**1**0000000
8	0000000**1**
9	000000**1**0
10	00000**1**00
11	0000**1**000
12	000**1**0000
13	00**1**00000
14	0**1**000000
15	**1**0000000
...	...

Table 1: Example sequence of LEDs with 1 meaning led is open and 0 meaning led is closed.

3 Part 2

In this part, you will code a simple two-digit decimal counter using two seven-segment displays and push-buttons. The first seven-segment display will show the tens digit and the second seven-segment display will show the ones digit. Your program will increase the counter when the first push-button is clicked and your program will decrease the counter when the second push-button is clicked. The click means to press and release the buttons. You shouldn't increase or decrease the counter regularly while pressing the button continuously.

4 Part 3

In this part, you are asked to develop an up-counter. The seven-segment displays show the actual number as a decimal and the LEDs show it as binary. The counter varies between 0 and 99. The initial value of the counter is 0. The increment number will decrease when the first push-button is clicked, the increment number will increase when the second push-button is clicked. The initial increment number is 1 and varies between 1 and 10. The counter will increase with the increment number for each period. The initial period is 500 ms. The period time will decrease 100 ms when the third push-button is clicked and it will increase 100 ms when the fourth push-button clicked. The period time varies between 100 ms to 1000 ms. Increasing and decreasing push-buttons do not work as circular. For instance, the first push-button can not decrease the increment number when the increment number is 1. Example sequences of the counter are shown in Table 2.

increment number	decimal display
1	00-01-02-03-04-05-06
2	99-01-03-05-07-09-11
3	66-69-72-75-78-81-84
...	...

Table 2: Example sequence of leds with 1 meaning led is open and 0 meaning led is closed.

Note: Assume that push-button press and release time is greater than 1 ms. You shouldn't miss the click event of the push-button because of delay function dead time and you can not check the push-button state in the delay function. You can not write different delay functions for each period time such as delay_100ms, delay_200ms, and delay_300ms. You cannot call the delay function repeatedly.
Hint: You can write your delay function to generate different period time.

5 Part 4

In this part, you will write a code that generates the sequences over LEDs in table 3 according to the pattern no. The first seven-segment display shows the pattern number. It varies between 1 and 4. The pattern number will decrease when the first push-button is clicked and it will increase when the second push-button is clicked. Your LEDs show a new sequence for each period. The initial period is 500 ms. The period time will decrease 100 ms when the third push-button is clicked and it will increase 100 ms when the fourth push-button clicked. The period varies between 100 ms and 900 ms. The second seven-segment display shows the period time between 1 and 9. It represents n x 100 ms. Increasing and decreasing push-buttons do not work as circular. For instance, the first push-button can not decrease the increment number when the increment number is 1.
Note: Assume that push-button press and release time is greater than 1 ms. You shouldn't miss the click event of the push-button because of delay function dead time and you can not check the push-button state in the delay function. You can not write different delay functions for each period time such as delay_100ms, delay_200ms, and delay_300ms. You cannot call the delay function repeatedly.

time	led pattern
0	0000000**1**
1	000000**1**0
2	00000**1**00
3	0000**1**000
4	000**1**0000
5	00**1**00000
6	0**1**000000
7	**1**0000000
8	0**1**000000
9	00**1**00000
10	000**1**0000
11	0000**1**000
12	00000**1**00
13	000000**1**0
14	0000000**1**
15	000000**1**0
16	00000**1**00
...	...

(a) Pattern No: 1

time	led pattern
0	**1**0000000
1	0**1**000000
2	00**1**00000
3	000**1**0000
4	0000**1**000
5	00000**1**00
6	000000**1**0
7	0000000**1**
8	**1**0000000
9	0**1**000000
10	00**1**00000
11	000**1**0000
12	0000**1**000
13	00000**1**00
14	000000**1**0
15	0000000**1**
16	**1**0000000
...	...

(b) Pattern No: 2

time	led pattern
0	00000000
1	10000001
2	11000011
3	11100111
4	11111111
5	11100111
6	11000011
7	10000001
8	00000000
9	10000001
10	11000011
11	11100111
12	11111111
13	11100111
14	11000011
15	10000001
16	00000000
...	...

(c) Pattern No: 3

time	led pattern
0	0000000**1**
1	000000**1**0
2	00000**1**00
3	0000**1**000
4	00000**1**00
5	000000**1**0
6	0000000**1**
7	**1**0000000
8	0**1**000000
9	00**1**00000
10	000**1**0000
11	00**1**00000
12	0**1**000000
13	**1**0000000
14	0000000**1**
15	000000**1**0
16	00000**1**00
...	...

(d) Pattern No: 4

Table 3: Example sequence of leds with 1 meaning led is open and 0 meaning led is closed.

Hint: You can write your delay function to generate different period time.

6 Submission

- You should upload your experiment codes and report on Ninova, and please, do not send your experiment files via e-mail.
- You must upload each part's code seperately to the ninova.
- Your reports must be written with Latex format. Latex report template is available on Ninova. You can use any Latex editor whichever you want. If you upload your report without Latex file, you directly get 0 as your report grade. You should upload both .pdf and .tex files of your report.
- Finally, please do not forget that late submissions are not accepted.