



ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

242-302 ADV COM ENG LAB II

Lab 3HB05-06 AVR Microcontroller

เสนอ

อาจารย์ พัชร เทพนมิตร

จัดทำโดย

นาย ยุทธนา บ่อทรัพย์ 5635512070

Section: 01

ภาคการศึกษาที่ 2 ปีการศึกษา 2558

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

Lab 3HB05-06 AVR Microcontrolle

วัตถุประสงค์

- เพื่อให้ให้นักศึกษาได้เรียนรู้วิธีการเขียนโปรแกรมประยุกต์ใช้งานไมโครคอนโทรลเลอร์ AVR ด้วยภาษา Assembly และ C
- เพื่อให้ให้นักศึกษาได้เรียนรู้เทคนิคการดีบั๊กโปรแกรมภาษา Assembly และ C ก่อนที่จะนำโปรแกรมลงทดสอบบนฮาร์ดแวร์จริง
- เพื่อให้ให้นักศึกษาได้เรียนรู้เทคนิคการออกแบบฮาร์ดแวร์และซอฟต์แวร์ร่วมกัน เพื่อให้ง่ายในการทดสอบความถูกต้อง
- เพื่อให้ให้นักศึกษาได้ฝึกการเขียนซอฟต์แวร์สำหรับการทดสอบความถูกต้องของฮาร์ดแวร์

เป้าหมาย

- นักศึกษาสามารถใช้โปรแกรม AVRStudio ในการดีบั๊กโปรแกรมภาษา Assembly และ C ของชิพ AVR ได้อย่างมีประสิทธิภาพ
- นักศึกษาสามารถใช้โปรแกรม Proteus ในการทดสอบการทำงานร่วมกันของฮาร์ดแวร์และซอฟต์แวร์ได้อย่างมีประสิทธิภาพ ก่อนที่จะนำซอฟต์แวร์ไปทดสอบการทำงานบนฮาร์ดแวร์จริง

การทดลองที่ 1 เขียนโปรแกรมเพื่ออ่านค่าจากสวิตช์และแสดงผลทาง 7-segment LED

CODE C++

FILE .hex = 631 byte

CODE Assambly

FILE .hex = 157 byte

CODE การทดลองที่ 1 ด้วย C++

```
Program1 | Arduino 1.6.4
File Edit Sketch Tools Help

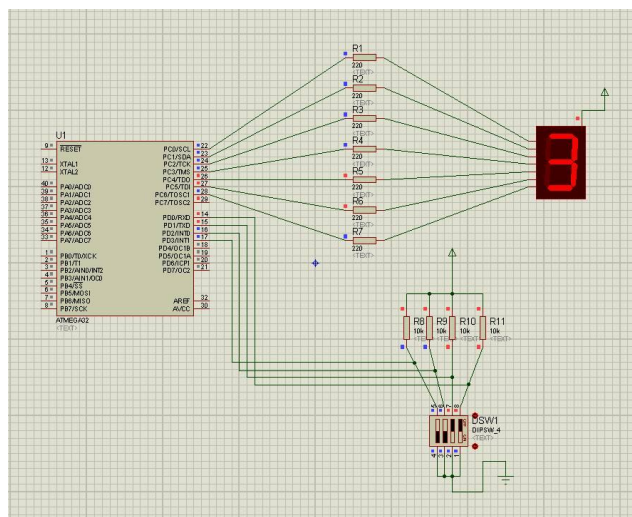
Program1

#include <avr/io.h>
int main (void)
{
    unsigned char LOOKUPTB[] = { 0b00111111,
                                0b00000110,
                                0b01011011,
                                0b01001111,
                                0b01100110,
                                0b01101101,
                                0b01111101,
                                0b00000111,
                                0b01111111,
                                0b01101111,
                                0b01110111,
                                0b01111100,
                                0b00111001,
                                0b01011110,
                                0b01111001,
                                0b01110001 };

    unsigned char DISPLY, SWITCH_V;

    DDRC = 0xFF;
    DDRD = 0x00;
    while(1)
    {
        SWITCH_V = PIND;
        SWITCH_V &= 0xF0;
        SWITCH_V = SWITCH_V >> 4;
        DISPLY = LOOKUPTB[SWITCH_V];
        PORTC = DISPLY;
    }
}
```

ผลการรันและวงจร การทดลองที่ 1 ด้วย C++



การทดลองที่ 2 เขียนโปรแกรม เพื่อนับลอจิกต่ำจากดิปสวิทช์

โปรแกรมจะทำการอ่านค่าจากดิปสวิทช์จำนวน 8 ตัว ที่ต่อกับพอร์ท D จากนั้น ทำการตรวจสอบบิตที่มา
ค่าลอจิกต่ำ และแสดงผลออก 7-segment ที่ต่อที่พอร์ท C

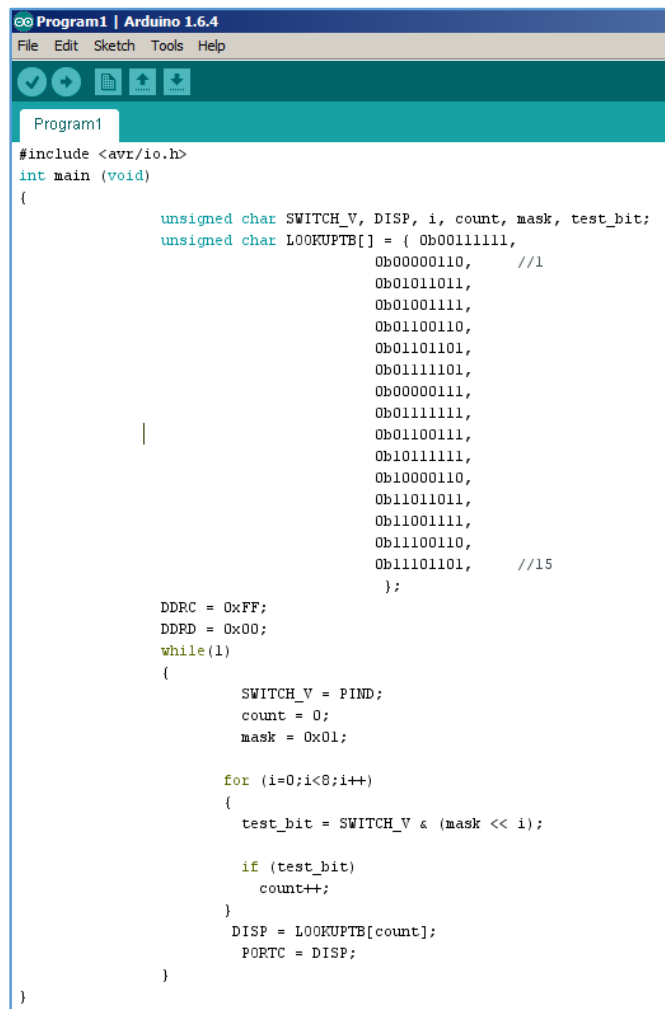
CODE C++

FILE .hex = 848 byte

CODE Assambly

FILE .hex = 206 byte

CODE การทดลองที่ 2 ด้วย C++

A screenshot of the Arduino IDE interface. The title bar reads "Program1 | Arduino 1.6.4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, running, and other functions. The main text area shows a C++ program. The code includes a header file, defines variables for a switch, display, and a lookup table, and contains a loop that reads the switch state and updates the display.

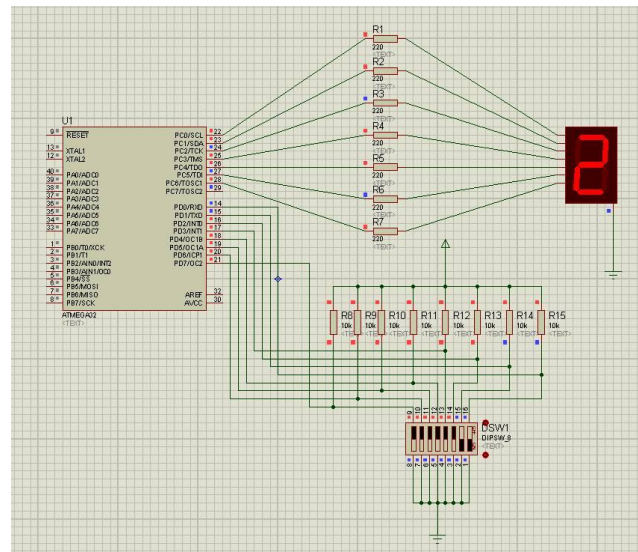
```
Program1
#include <avr/io.h>
int main (void)
{
    unsigned char SWITCH_V, DISP, i, count, mask, test_bit;
    unsigned char LOOKUPTB[] = { 0b00111111, //1
                                0b00000110,
                                0b01011011,
                                0b01001111,
                                0b01100110,
                                0b01101101,
                                0b01111101,
                                0b00000111,
                                0b01111111,
                                0b01100111,
                                0b10111111,
                                0b10000110,
                                0b11011011,
                                0b11001111,
                                0b11100110,
                                0b11101101, //15
                                };

    DDRC = 0xFF;
    DDRD = 0x00;
    while(1)
    {
        SWITCH_V = PIND;
        count = 0;
        mask = 0x01;

        for (i=0;i<8;i++)
        {
            test_bit = SWITCH_V & (mask << i);

            if (test_bit)
                count++;
        }
        DISP = LOOKUPTB[count];
        PORTC = DISP;
    }
}
```

ผลการรันและวงจร การทดลองที่ 2 ด้วย C++



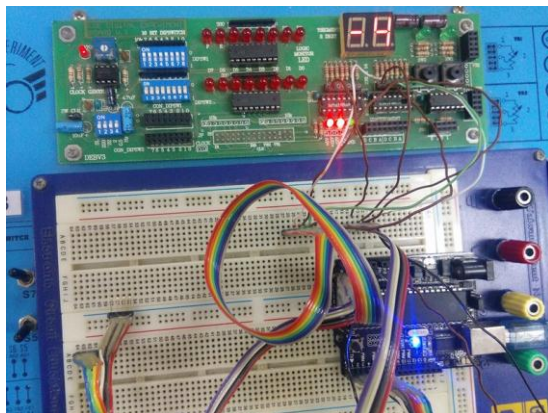
การทดลองที่ 3 เขียนโปรแกรม เพื่ออ่านค่าจากสวิตช์ โดยข้อมูลเป็นเลข signed number

จงเขียนโปรแกรมเพื่อแปลงค่าจากสวิตช์เป็นเลขฐานสิบแบบมีเครื่องหมาย ซึ่งมีค่าอยู่ระหว่าง -8...+7

ออกแสดงผลทางแอลอีดี แบบ 7 เซกเมนต์ 2 ตัว

CODE การทดลองที่ 3 ด้วย C++ ----->

ผลการรันและวงจร การทดลองที่ 3 ด้วย C++



```

Program1 | Arduino 1.6.4
File Edit Sketch Tools Help

Program1
#include <avr/io.h>
int main (void)
{
    unsigned char LOOKUPTB[] = { 0b00111111,
                                0b00000110,
                                0b01011011,
                                0b01001111,
                                0b01100110,
                                0b01101101,
                                0b01111101,
                                0b00000111,
                                0b11111111,
                                0b10000111,
                                0b11111101,
                                0b11101101,
                                0b11100110,
                                0b11001111,
                                0b11011011,
                                0b10000110,
                                };

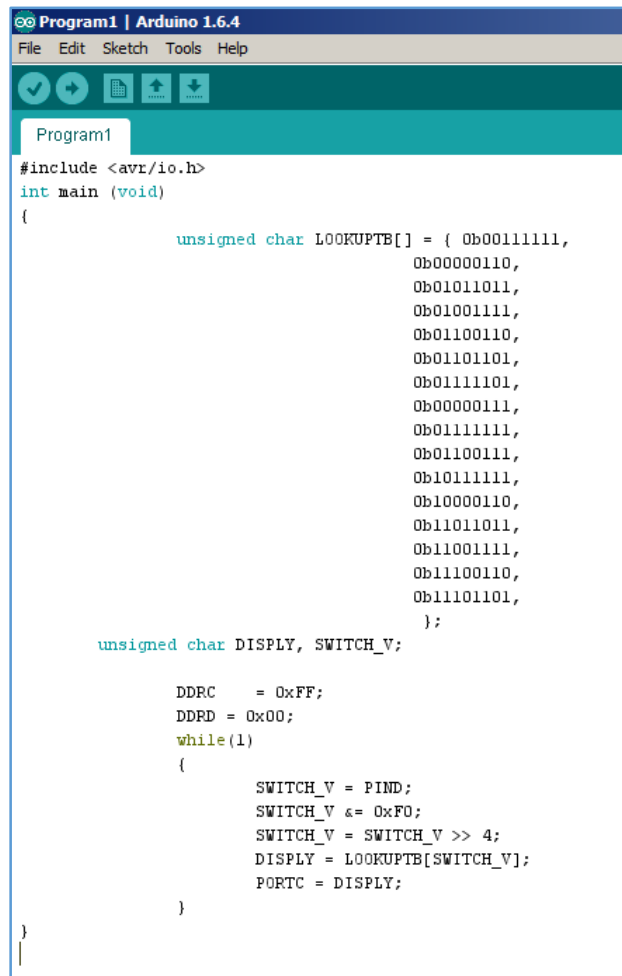
    unsigned char DISPLY, SWITCH_V;

    DDRC = 0xFF;
    DDRD = 0x00;
    while(1)
    {
        SWITCH_V = PIND;
        SWITCH_V &= 0xF0;
        SWITCH_V = SWITCH_V >> 4;
        DISPLY = LOOKUPTB[SWITCH_V];
        PORTC = DISPLY;
    }
}
    
```

การทดลองที่ 4 เขียนโปรแกรม เพื่อแสดงค่าจากสวิตช์ โดยข้อมูลเป็นเลขฐานสิบ

จงเขียนโปรแกรมเพื่อแปลงค่าจากสวิตช์เป็นเลขฐานสิบแบบไม่มีเครื่องหมาย ซึ่งมีค่าอยู่ระหว่าง 0-15
ออกแสดงผลทางแอลอีดี แบบ 7 เซกเมนต์ 2 ตัว

CODE การทดลองที่ 4 ด้วย C++

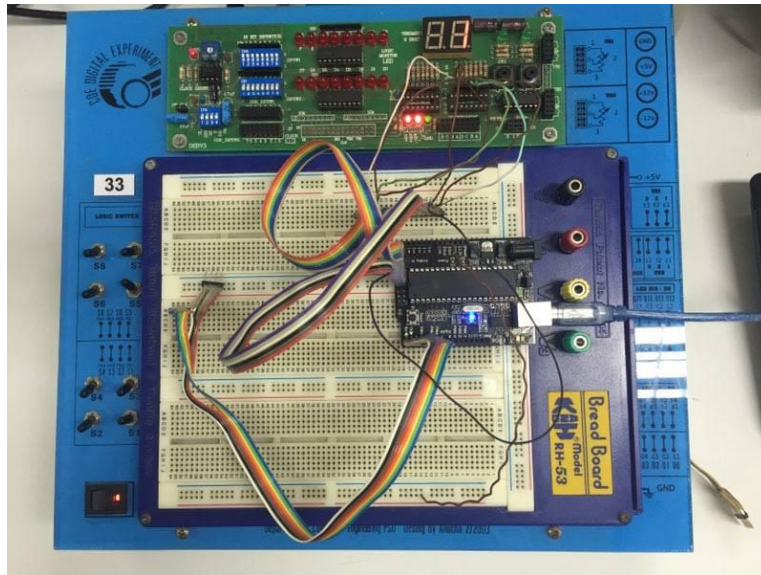
The image shows a screenshot of the Arduino IDE interface. The title bar reads "Program1 | Arduino 1.6.4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, and uploading. The main text area shows the following C++ code:

```
Program1
#include <avr/io.h>
int main (void)
{
    unsigned char LOOKUPTB[] = { 0b00111111,
                                0b00000110,
                                0b01011011,
                                0b01001111,
                                0b01100110,
                                0b01101101,
                                0b01111101,
                                0b00000111,
                                0b01111111,
                                0b01100111,
                                0b10111111,
                                0b10000110,
                                0b11011011,
                                0b11001111,
                                0b11100110,
                                0b11101101,
                                };

    unsigned char DISPLY, SWITCH_V;

    DDRC  = 0xFF;
    DDRD = 0x00;
    while(1)
    {
        SWITCH_V = PIND;
        SWITCH_V &= 0xF0;
        SWITCH_V = SWITCH_V >> 4;
        DISPLY = LOOKUPTB[SWITCH_V];
        PORTC = DISPLY;
    }
}
```

ผลการรันและวงจร การทดลองที่ 4 ด้วย C++



การทดลองที่ 5 ออกแบบและทดสอบระบบสวิตช์สัมผัสสำหรับเปิดปิดเครื่องใช้ไฟฟ้า

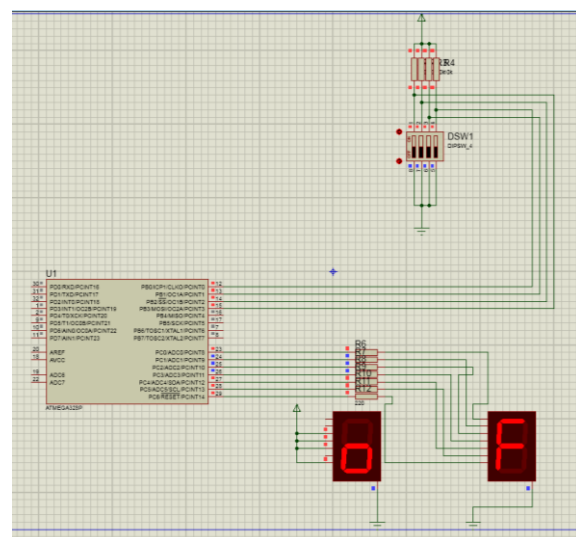
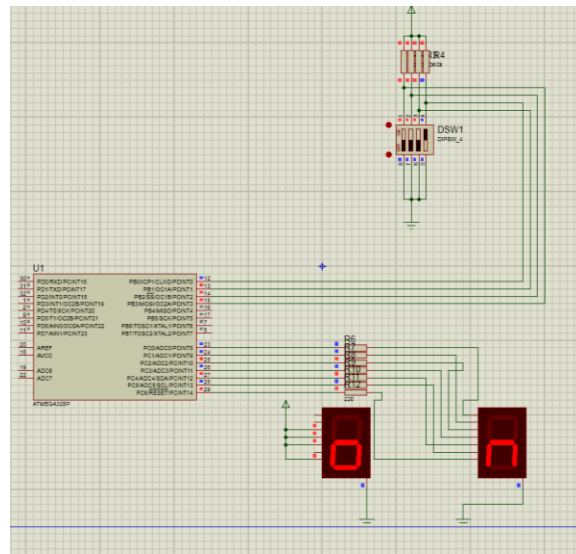
การทดลองที่ 5.1 ออกแบบและทดสอบระบบสวิตช์สัมผัสสำหรับเปิดปิดเครื่องใช้ไฟฟ้า

- เมื่อเริ่มจ่ายไฟให้กับโปรแกรม หลอด L1 จะดับอยู่ และ 7-segment แสดงคำว่า off
- เมื่อกดสวิตช์ หลอด L1 จะติด และ 7-segment แสดงคำว่า on
- เมื่อกดสวิตช์อีกครั้ง หลอด L1 จะดับ และ 7-segment แสดงคำว่า off

CODE การทดลองที่ 5.1 ด้วย C++

```
1  #include <avr/io.h>
2
3  unsigned char TB7SEG[] = {
4      0b01110001,
5      0b01010100
6  };
7  unsigned char mode,tmp,c;
8  int main()
9  {
10     // Write your code here
11     DDRB = 0x00;
12     DDRC = 0xFF;
13     mode = 0x00;
14     PORTC = TB7SEG[0];
15     tmp = 0x00;
16     c = 0x00;
17     while (1){
18         c = tmp;
19         tmp = PINB;
20         tmp &= 0x0F;
21         if(tmp != c){
22             mode = mode ^ 0x01;
23             PORTC = TB7SEG[mode];
24         }
25     }
26     return 0;
27 }
```

ผลการรันและวงจร การทดลองที่ 5.1 ด้วย C++



การทดลองที่ 5.2 เขียนโปรแกรมตั้งเวลา

- เมื่อเริ่มจ่ายไฟให้กับโปรแกรม 7-segment แสดงเลข 20 เมื่อกดสวิตช์ 7-segment จะเริ่มนับถอยหลังจากทุกๆ 1 วินาที จะมีค่าลดลงทีละหนึ่ง จนเหลือ 00
- ถ้ากดสวิตช์อีกครั้ง 7-segment จะรีเซ็ตจากค่า 00 มาเป็น 20
- ถ้ากดสวิตช์อีกครั้ง 7-segment จะนับถอยหลังไปที่เลข 00 การทำงานจะเป็นสเตทสลับไปมา

CODE การทดลองที่ 5.2 ด้วย C++

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#define VALUE_T1 34286
unsigned char LOOKUPTB[] = {0b00111111,0b00000110,
                           0b01011011,0b01001111,
                           0b01100110,0b01101101,
                           0b01111101,0b00000111,
                           0b01111111,0b01101111,
                           0b01110111,0b01111100,
                           0b00111001,0b01011110,
                           0b0111001,0b01110001 };

unsigned char count;
int main()
{
    DDRB = 0xFF; // out signal at oscilloscope
    DDRC = 0xFF; // output at 7-segment (unit)
    DDRD = 0xFF; // output at 7-segment (ten)

    count = 20; // max value that decrement count
    PORTC = LOOKUPTB[count%10]; // display 20 in first time
    PORTD = LOOKUPTB[count/10];
```

```
    TCCR1A = 0x00; // set timer (normal mode)
    TCCR1B = 0x04; // set prescalar = 256
    TIMSK1 = 0x01; // set Overflow interrupt enable

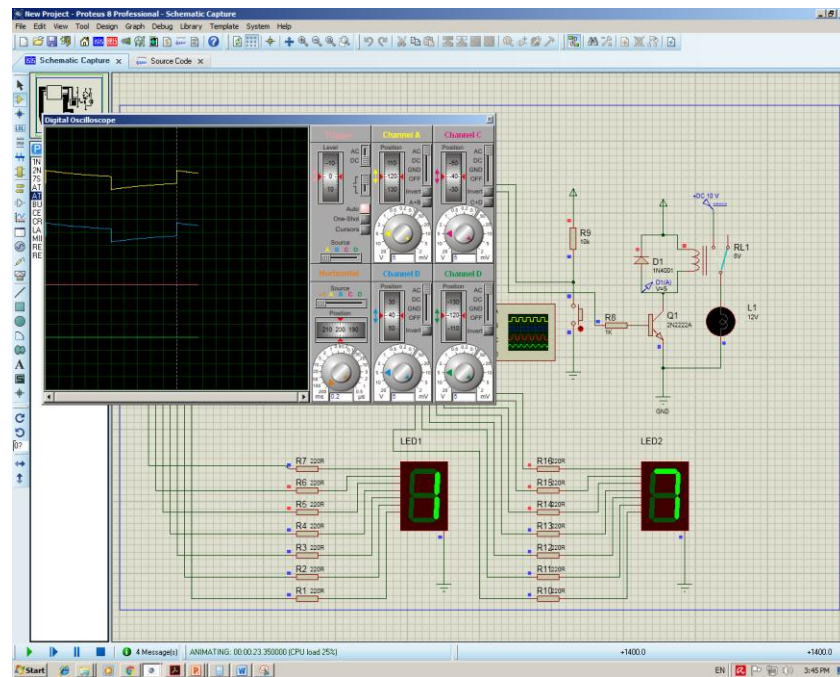
    cli(); // disable all interrupt
    TCNT1 = VALUE_T1; // initial TCNT1
    sei(); // enable all interrupt

    while (1)
    ;
    return 0;
}
```




```
ISR (TIMER1_OVF_vect)
{
    TCNT1 = VALUE_T1;
    PORTB = PORTB ^ 0x0C; // 0000 1100 output at oscilloscope

    count--;
    if(count == 0){
        count = 20;
    }
    PORTC = LOOKUPTB[count%10]; // unit
    PORTD = LOOKUPTB[count/10]; // ten
}
```

ผลการรันและวงจร การทดลองที่ 5.2 ด้วย C++



การทดลองที่ 5.4 รวมโปรแกรม

- เมื่อเริ่มจ่ายไฟให้กับเครื่อง เครื่องใช้ไฟฟ้าจะดับอยู่ และแอลอีดีแสดงสถานะ 
- เมื่อกดสวิตช์ 1 ครั้งจะเป็นการกดเปิดเครื่องใช้ไฟฟ้าโดยมีแอลอีดี 7 เซกเมนต์ 2 ตัวแสดงสถานะของวงจรเป็น 
- เมื่อกดสวิตช์อีก 1 ครั้งจะเป็นการสั่งปิดเครื่องใช้ไฟฟ้า แต่เครื่องใช้ไฟฟ้าจะไม่ถูกตัดไฟในทันที แต่จะมีการหน่วงเวลาออกไป 20 วินาที พร้อมทั้งแสดงการนับลงที่แอลอีดีทั้งสอง เมื่อครบ 20 วินาทีจึงดับเครื่องใช้ไฟฟ้าแล้วแสดงสถานะที่แอลอีดีเป็น 

CODE การทดลองที่ 5.2 ด้วย C++

```
1  #include <inttypes.h>
2  #include <avr/io.h>
3  #include <avr/interrupt.h>
4  #include <avr/sleep.h>
5  #define VALUE 34286
6  unsigned char count = 10;
7  unsigned char zero = 0;
8  unsigned char zero1 = 0b01010100;
9  unsigned char zero2 = 0b01011100;
10 unsigned char zero3 = 0b01110001;
11 unsigned char zero4 = 0b01011100;
12 unsigned char LOOKUPTB[] = { 0b00111111,0b00000110,0b01011011,0b01001111,0b01100110,
13                               0b01101101,0b01111101,0b00000111,0b01111111,0b01101111,
14                               0b01110111,0b01111100,0b00111001,0b01011110,0b01111001,
15                               0b01110001 };
16
17 int main()
18 {
19     //set output
20     DDRB = 0xFE; // last pin (bit 0) is input 1111 1110
21     DDRC = 0xFF;
22     DDRD = 0xFF;
23
24     PORTB = PORTB & 0xF3; // 1111 0011
25
26     PORTC = LOOKUPTB[count%10]; // unit
27     PORTD = LOOKUPTB[count/10]; // ten
28
29     TCCR1A = 0x00;
30     TCCR1B = 0x04;
31     TIMSK1 = 0x01;
32     PCICR = 0x01; //enable check port B [14:8] Pin change interrupt
33     PCMSK0 = 0b00000001; //use port B pin 0123
34     //Start OFF
35     PORTC = zero3;
36     PORTD = zero4;
37     TIMSK1 = 0x00;
38
39     cli();
40     TCNT1 = VALUE;
41     sei();
42     while (1);
43     return 0;
44 }
```

```
44
45 ISR(TIMER1_OVF_vect)
46 {
47     TCNT1 = VALUE;
48     PORTB = PORTB ^ 0x0E; // 0000 1110 back bit
49     if(count == 0)
50     {
51         PORTC = zero3; // unit
52         PORTD = zero4; // ten
53         TIMSK1 = 0x00;
54         PORTB = 0x00; //LED OFF
55     }
56     else if(count == 1)
57     {
58         PORTB = 0x02;
59         count--;
60         PORTB = PORTB ^ 0x0C; // 0000 1100 back bit
61         PORTC = LOOKUPTB[count%10]; // unit
62         PORTD = LOOKUPTB[count/10]; // ten
63     }
64     else
65     {
66         PORTB = 0x02;
67         count--;
68         PORTB = PORTB ^ 0x0C; // 0000 1100 back bit
69         PORTC = LOOKUPTB[count%10]; // unit
70         PORTD = LOOKUPTB[count/10]; // ten
71     }
72 }
73
74 ISR (PCINT0_vect){
75     TIMSK1 = 0x01;
76     PORTC = zero1; // unit
77     PORTD = zero2; // ten
78     count = 21;
79 }
```

ผลการรันและวงจร การทดลองที่ 5.2 ด้วย C++

