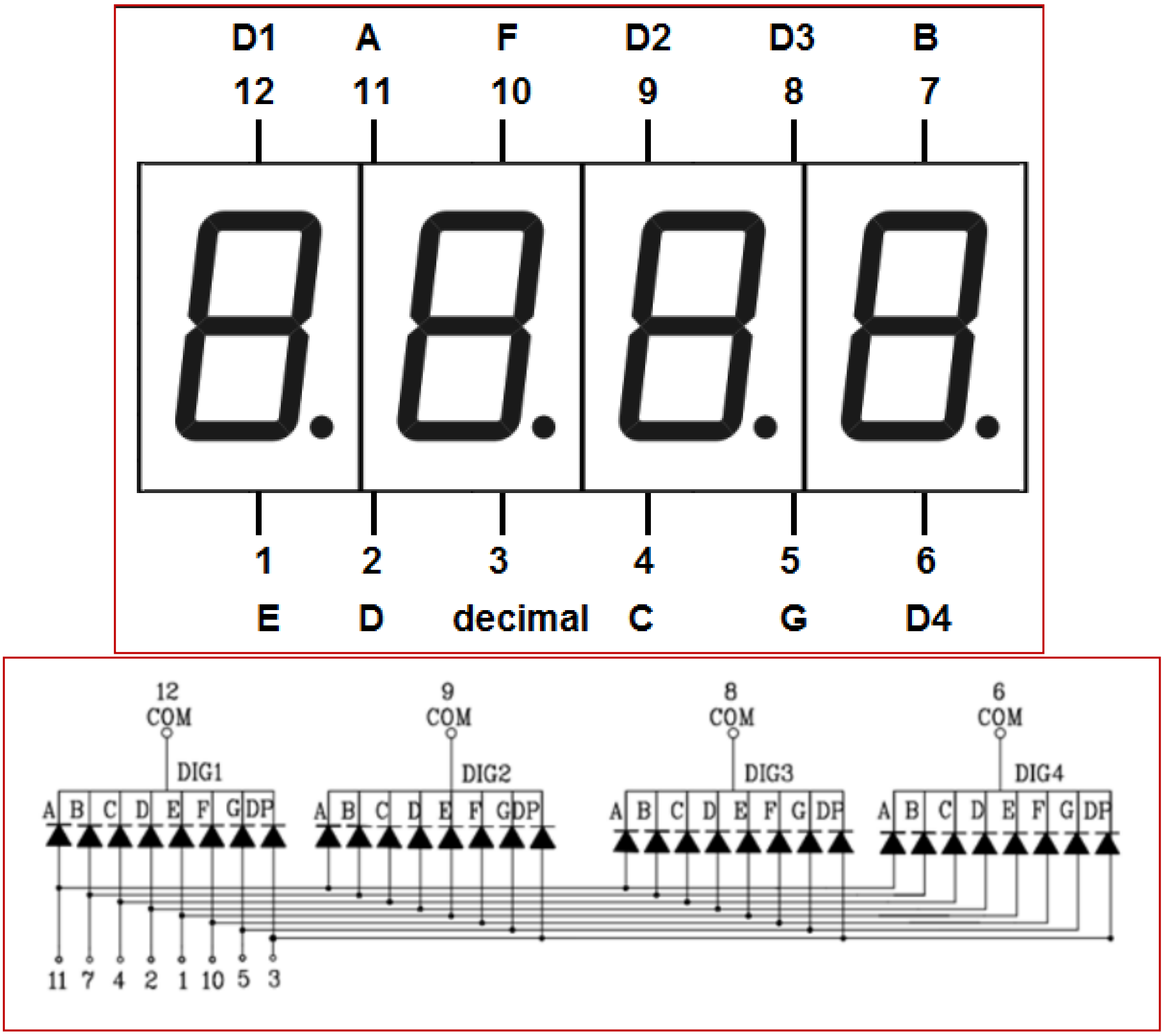
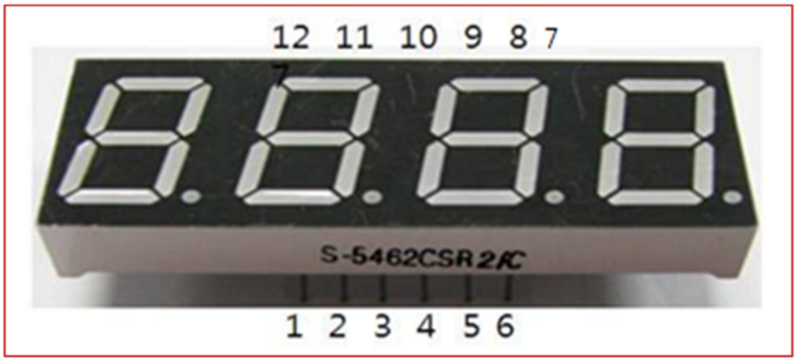
실험 5. 4 digit 7-segment

5.1 4 digit 7-segment



[4 Digit 7-Segment 회로]

- SH5461AS(CC) 사용

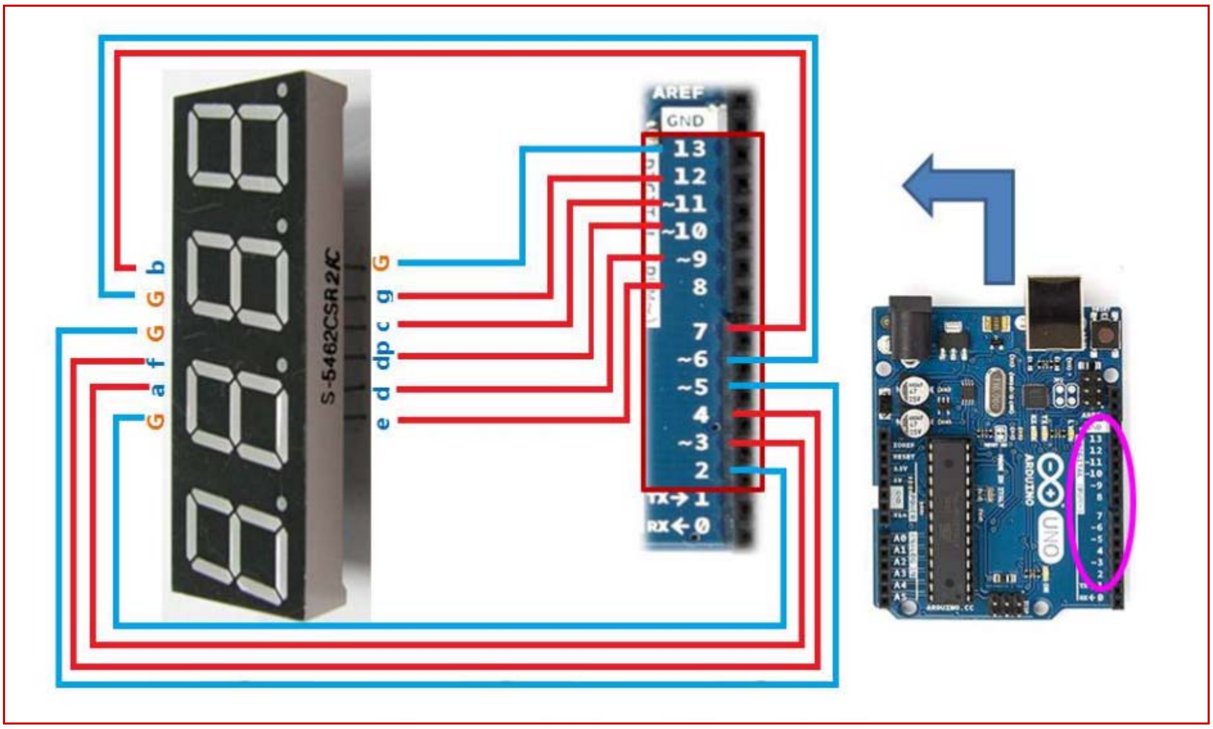


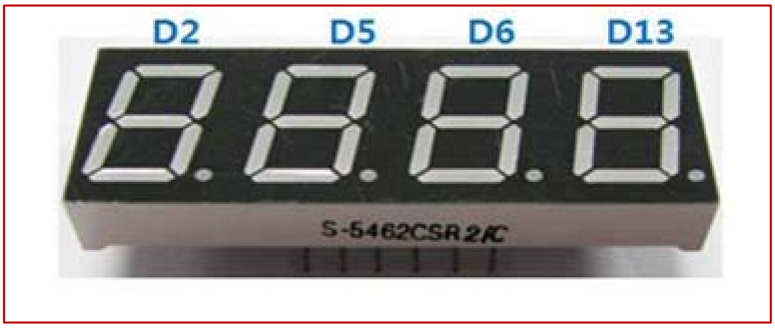
- Common Pin: 6, 8, 9 12번
  - -> 이 핀으로 4개의 FND 중 하나만 선택함
  - -> 사람의 눈으로 느낄 수 없을 만큼의 빠른 속도

- 8개 LED 제어: 1, 2, 3, 4, 5, 7, 10, 11
  - -> 아래는 왼쪽부터 오른쪽으로 위는 오른쪽부터 왼쪽으로
  - -> 위의 그림 참조

- 7-Segment Operation 방법
  - -> Static Method: Hardware 복잡, Software 간단
  - -> Dynamic Method: Hardware 간단, Software 복잡, Driver 줄임

5. 2　4 digit 7-segment 결선

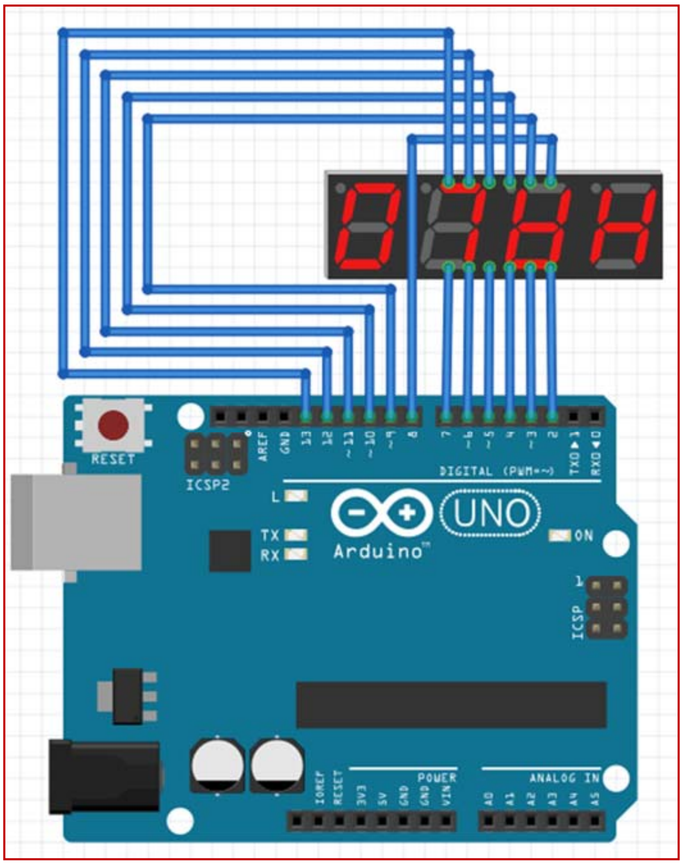




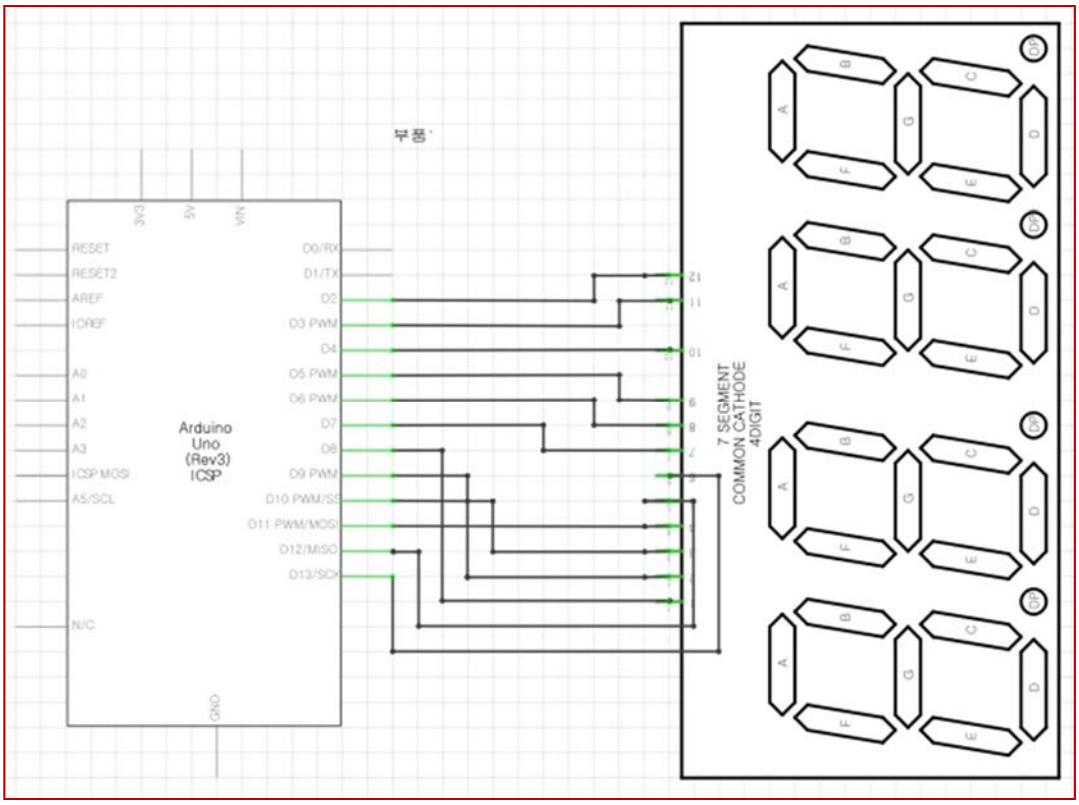
- **Arduino Pin 과 4 Digit 7-Segment connection**
  - 2->seg1(12), 5->seg2(9), 6->seg3(8), 13->seg4(6)
  - 3->ledA(11), 7->ledB(7), 11->ledC(4), 9->ledD(2), 8->ledE (1), 4->ledF(10), 12->ledG(5), 10->DOT(3)

EX. 1) 4 digit 7-segment에 1234 Display하기

- 반드시 실험을 해보고 결과를 교수님한테 검사를 받는다.

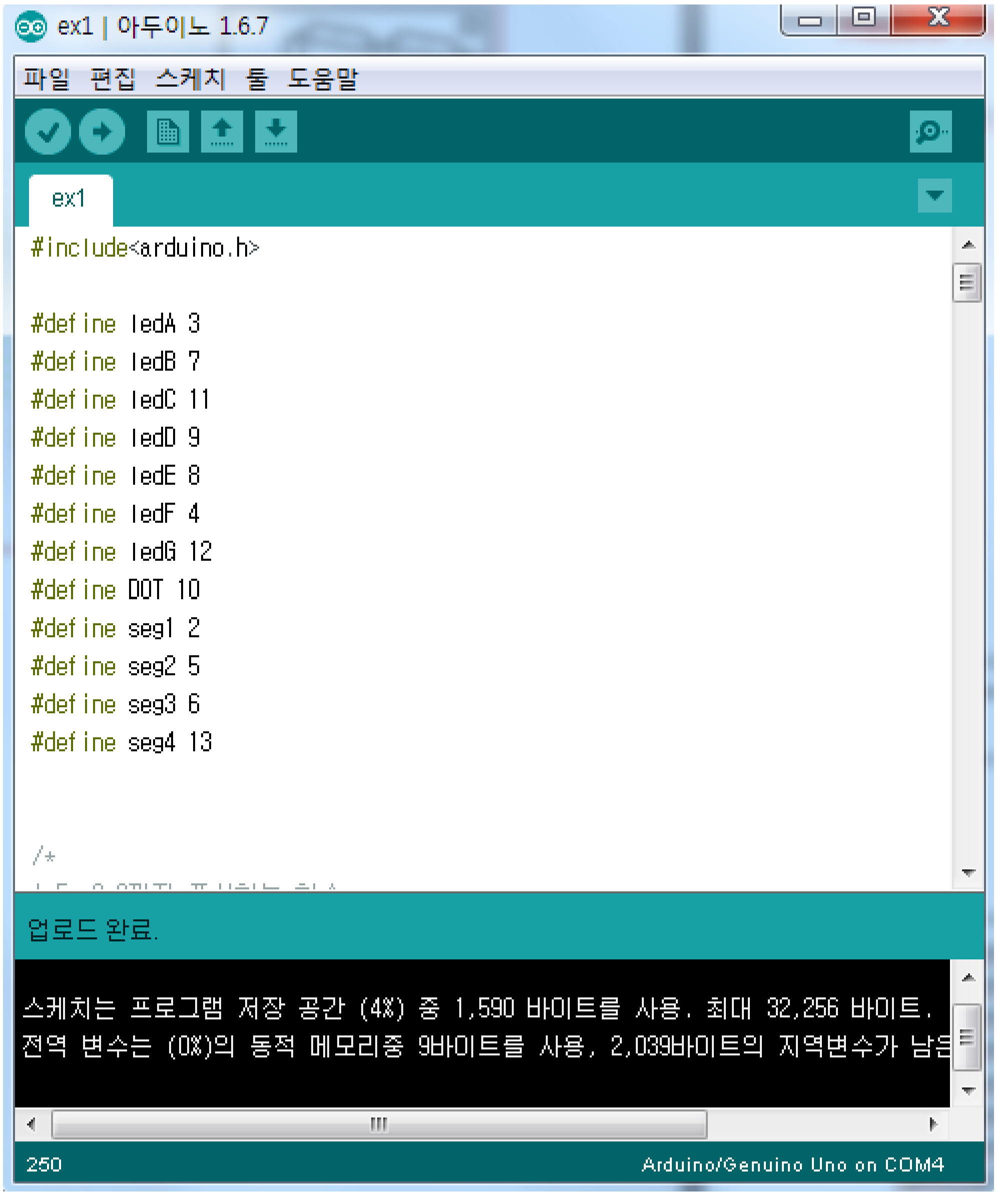


[배선도]

부품
3V3
5V
VIN
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
ICSP MOSI
A5/SCL
N/C
Arduino
Uno
(Rev3)
ICSP
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
12
11
10
9
8
7
7 SEGMENT
COMMON CATHODE
4DIGIT
A
B
C
D
E
F
G
DP

[회로도]

프로그램 입력



```
#include<arduino.h>

#define ledA 3
#define ledB 7
#define ledC 11
#define ledD 9
#define ledE 8
#define ledF 4
#define ledG 12
#define DOT 10
#define seg1 2
#define seg2 5
#define seg3 6
#define seg4 13


/*
```

```
스케치는 프로그램 저장 공간 (4%) 중 1,590 바이트를 사용. 최대 32,256 바이트.
전역 변수는 (0%)의 동적 메모리중 9바이트를 사용, 2,039바이트의 지역변수가 남은
```

Program File

// Header File 역할 – 숫자, 문자 Display 및 Clear, Set 등

```
#include<arduino.h>

#define ledA 3
#define ledB 7
#define ledC 11
#define ledD 9
#define ledE 8
#define ledF 4
#define ledG 12
#define DOT 10
#define seg1 2
#define seg2 5
#define seg3 6
#define seg4 13


/*
A~F, 0~9까지 표시하는 함수
7개의 LED를 HIGH(켜짐)/LOW(꺼짐)으로 설정하여 표시한다.
*/
void charfnd(int n) // 문자 출력 함수
{
    switch(n)
    {
        case 1:
            //Displays A
            digitalWrite(ledD, LOW);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, HIGH);
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(DOT , LOW);
```

```
   break;

case 2:
   //Displays B
   digitalWrite(ledD, HIGH);
   digitalWrite(ledE, HIGH);
   digitalWrite(ledF, HIGH);
   digitalWrite(ledG, HIGH);
   digitalWrite(ledA, HIGH);
   digitalWrite(ledB, HIGH);
   digitalWrite(ledC, HIGH);
   digitalWrite(DOT , LOW);
   break;

case 3:
   //Displays C
   digitalWrite(ledD, HIGH);
   digitalWrite(ledE, HIGH);
   digitalWrite(ledF, HIGH);
   digitalWrite(ledG, LOW);
   digitalWrite(ledA, HIGH);
   digitalWrite(ledB, LOW);
   digitalWrite(ledC, LOW);
   digitalWrite(DOT , LOW);
   break;

case 4:
   //Displays D
   digitalWrite(ledD, HIGH);
   digitalWrite(ledE, HIGH);
   digitalWrite(ledF, HIGH);
   digitalWrite(ledG, LOW);
   digitalWrite(ledA, HIGH);
   digitalWrite(ledB, HIGH);
   digitalWrite(ledC, HIGH);
   digitalWrite(DOT , LOW);
   break;

case 5:
```

```
          //Displays E
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;

        case 6:
          //Displays F
          digitalWrite(ledD, LOW);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;
    }
}

void fnd(int n) // 숫자 출력 함수
{
    switch(n)
    {

          case 0:
          //Displays 0
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, LOW);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
          digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 1:
//Displays 1
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 2:
//Displays 2
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, LOW);
digitalWrite(DOT , LOW);
break;

case 3:
//Displays 3
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;
```

```
case 4:
//Displays 4
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 5:
//Displays 5
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 6:
//Displays 6
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 7:
//Displays 7
digitalWrite(ledD, LOW);
```

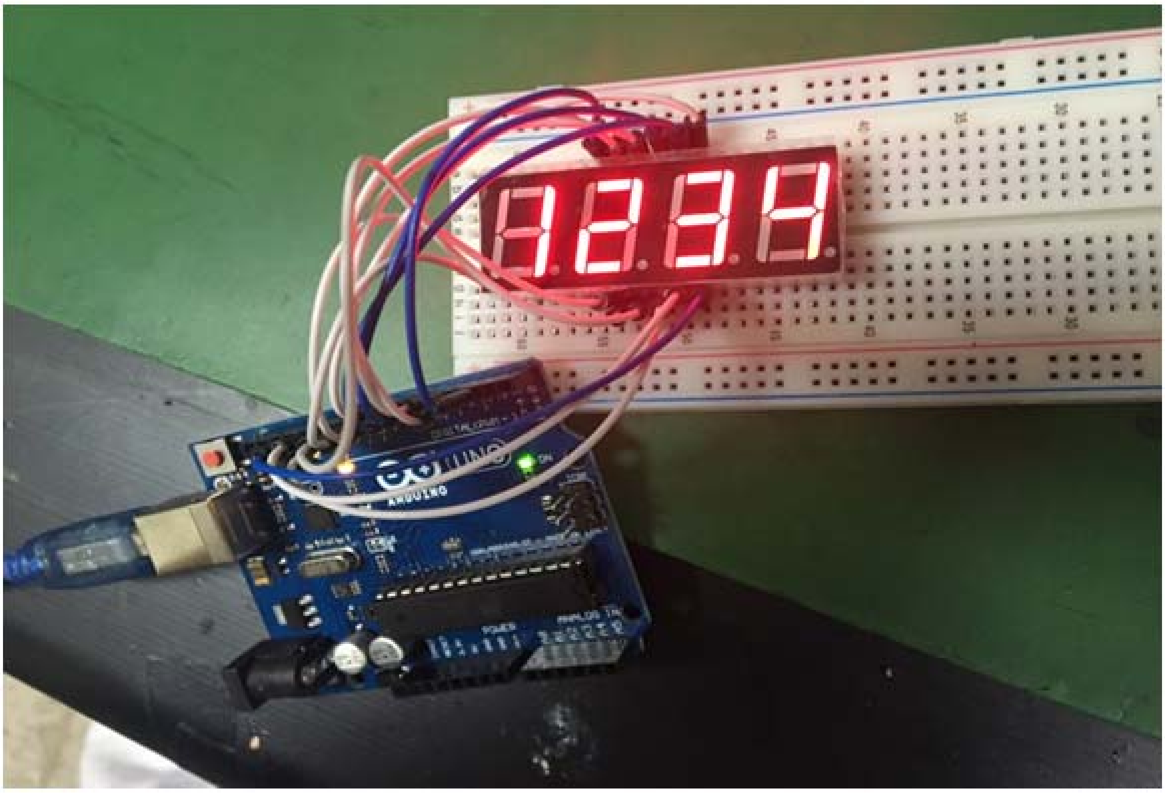
```
      digitalWrite(ledE, LOW);
      digitalWrite(ledF, LOW);
      digitalWrite(ledG, LOW);
      digitalWrite(ledA, HIGH);
      digitalWrite(ledB, HIGH);
      digitalWrite(ledC, HIGH);
      digitalWrite(DOT , LOW);
      break;

      case 8:
      //Displays 8
      digitalWrite(ledD, HIGH);
      digitalWrite(ledE, HIGH);
      digitalWrite(ledF, HIGH);
      digitalWrite(ledG, HIGH);
      digitalWrite(ledA, HIGH);
      digitalWrite(ledB, HIGH);
      digitalWrite(ledC, HIGH);
      digitalWrite(DOT , LOW);
      break;

      case 9:
      //Displays 9
      digitalWrite(ledD, HIGH);
      digitalWrite(ledE, LOW);
      digitalWrite(ledF, HIGH);
      digitalWrite(ledG, HIGH);
      digitalWrite(ledA, HIGH);
      digitalWrite(ledB, HIGH);
      digitalWrite(ledC, HIGH);
      digitalWrite(DOT , LOW);
      break;

  }
}
void clr() //led clear 함수
{
  //Clears the LED
  digitalWrite(ledA, LOW);
```

```
   digitalWrite(ledB, LOW);
   digitalWrite(ledC, LOW);
   digitalWrite(ledD, LOW);
   digitalWrite(ledE, LOW);
   digitalWrite(ledF, LOW);
   digitalWrite(ledG, LOW);
   digitalWrite(DOT , LOW);
}
void set() // led set 함수
{   //Set the LED
   digitalWrite(ledA, HIGH);
   digitalWrite(ledB, HIGH);
   digitalWrite(ledC, HIGH);
   digitalWrite(ledD, HIGH);
   digitalWrite(ledE, HIGH);
   digitalWrite(ledF, HIGH);
   digitalWrite(ledG, HIGH);
   digitalWrite(DOT , HIGH);
}
```

// 1234 Display Program

```
void setup()

{
   //각각의 핀들을 OUTPUT으로 설정해 준다.
   pinMode(ledA, OUTPUT);
   pinMode(ledB, OUTPUT);
   pinMode(ledC, OUTPUT);
   pinMode(ledD, OUTPUT);
   pinMode(ledE, OUTPUT);
   pinMode(ledF, OUTPUT);
   pinMode(ledG, OUTPUT);
   pinMode(DOT, OUTPUT);
   pinMode(seg1, OUTPUT);
   pinMode(seg2, OUTPUT);
   pinMode(seg3, OUTPUT);
   pinMode(seg4, OUTPUT);

}
```

```
void loop()
{
    /* 각 세그먼트 숫자 출력*/
     digitalWrite(seg1,LOW);
     fnd(1);
     delay(5);
     digitalWrite(seg1,HIGH);

     digitalWrite(seg2,LOW);
     fnd(2);
     delay(5);
     digitalWrite(seg2,HIGH);

     digitalWrite(seg3,LOW);
     fnd(3);
     delay(5);
     digitalWrite(seg3,HIGH);

     digitalWrite(seg4,LOW);
     fnd(4);
     delay(5);
     digitalWrite(seg4,HIGH);
 }
```

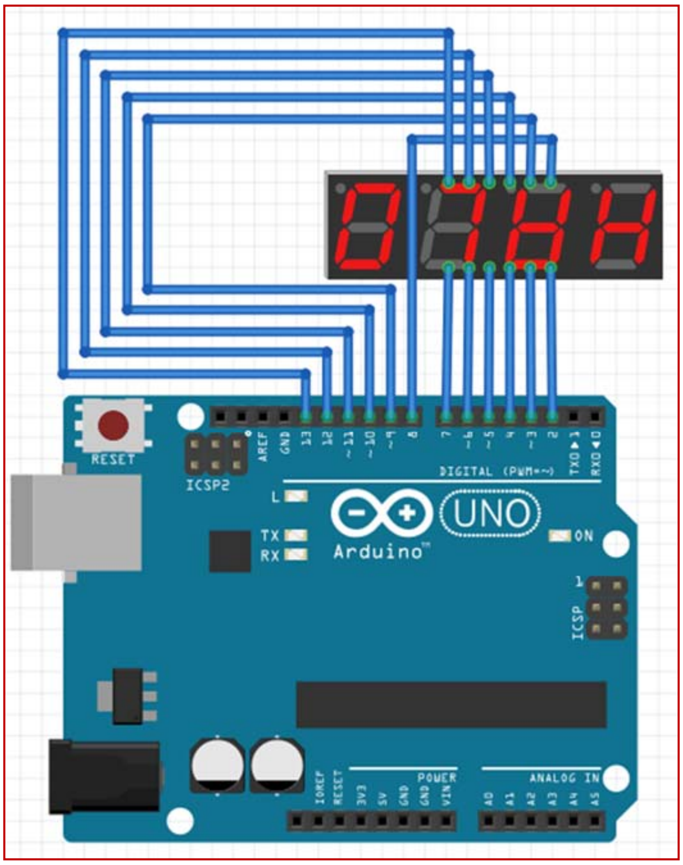


[프로그램 결과]

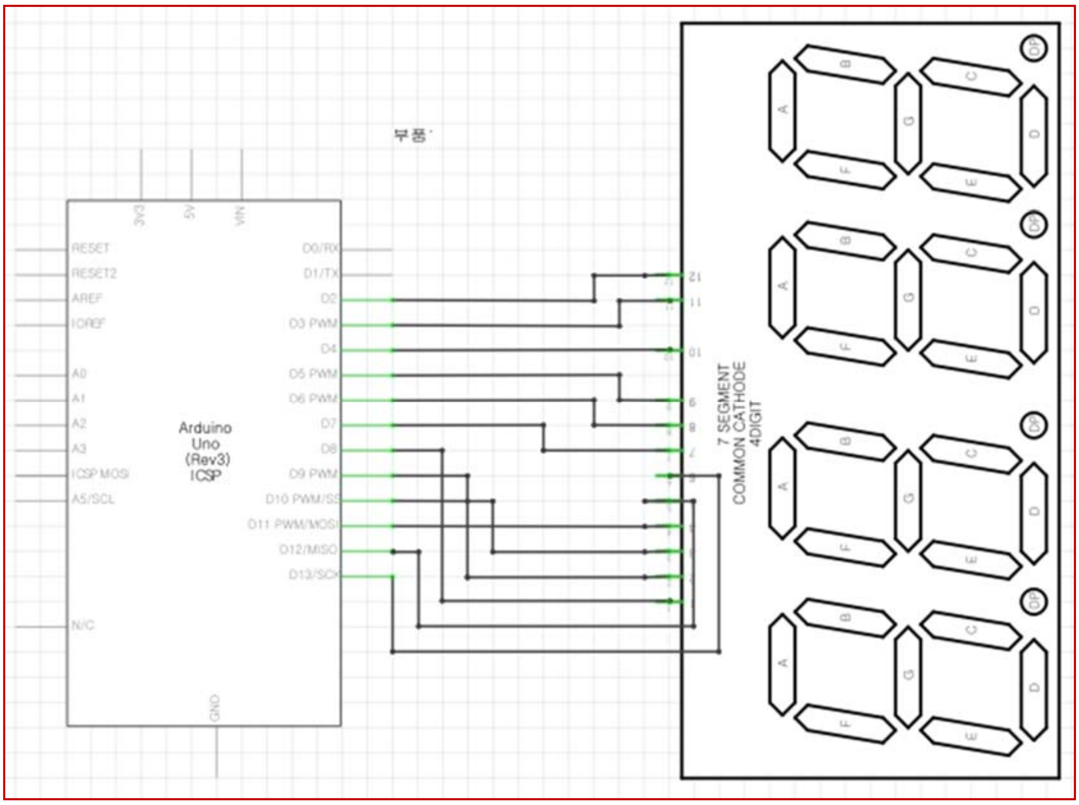
EX. 2) Serial Communication을 이용한 4 digit 7-segment에 Display하기

- 시리얼통신을 통해 4자리 숫자를 입력할 때 4 digit 7-segment의 4 자리 숫자를 출력하는 프로그램을 작성함.

- 반드시 실험을 해보고 결과를 교수님한테 검사를 받는다.

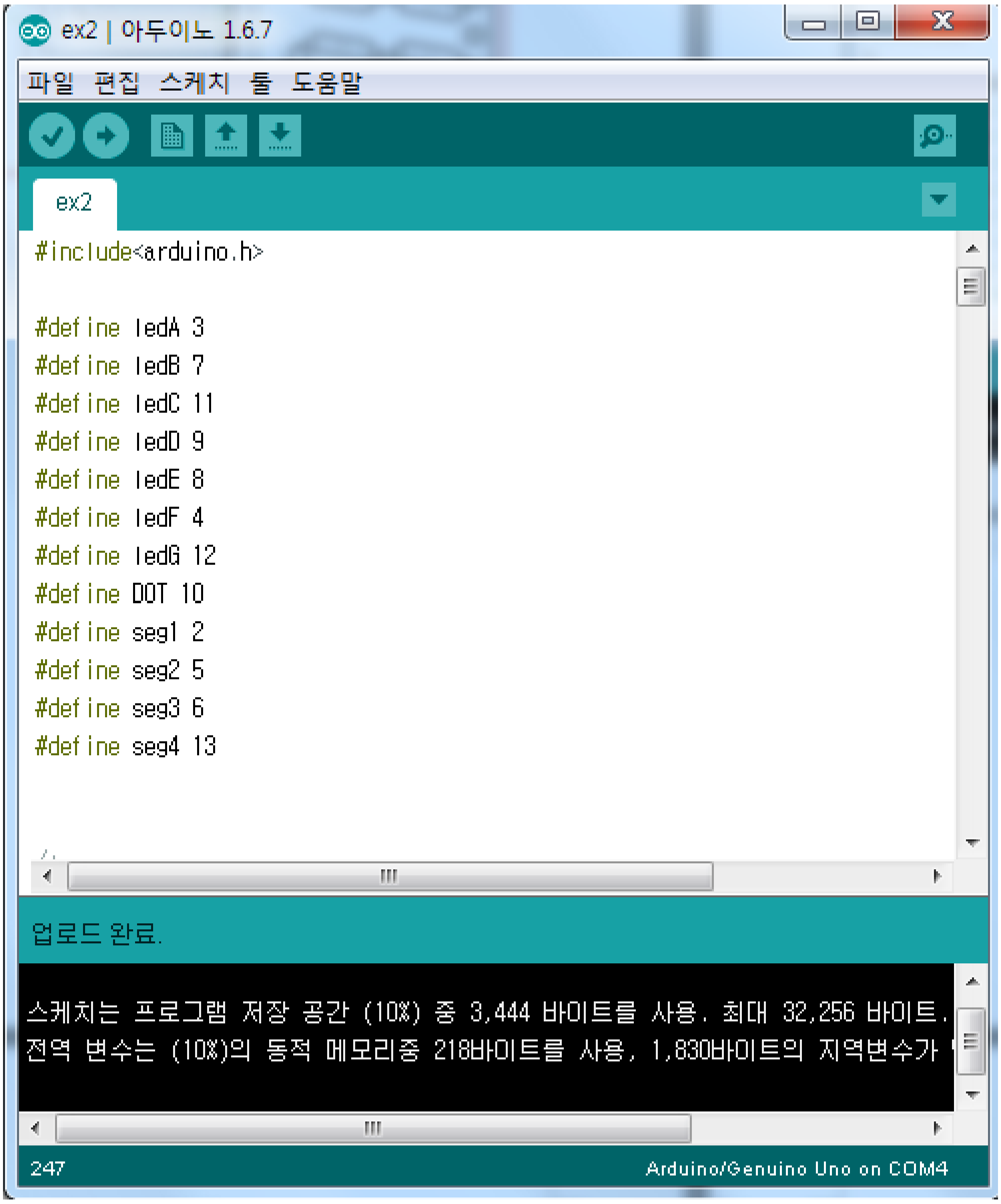


[아두이노 배선도]


부품
3V3
5V
VIN
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
ICSP MOSI
A5/SCL
N/C
Arduino
Uno
(Rev3)
ICSP
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
12
11
10
9
8
7
6
7 SEGMENT
COMMON CATHODE
4DIGIT
A
B
C
D
E
F
G
DP

[회로도]

프로그램 입력



```
#include<arduino.h>

#define ledA 3
#define ledB 7
#define ledC 11
#define ledD 9
#define ledE 8
#define ledF 4
#define ledG 12
#define DOT 10
#define seg1 2
#define seg2 5
#define seg3 6
#define seg4 13
```

업로드 완료.

```
스케치는 프로그램 저장 공간 (10%) 중 3,444 바이트를 사용. 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리중 218바이트를 사용, 1,830바이트의 지역변수가
```

247 Arduino/Genuino Uno on COM4

Program File

// Header File 역할 – 숫자, 문자 Display 및 Clear, Set 등

```
#include<arduino.h>

#define ledA 3
#define ledB 7
#define ledC 11
#define ledD 9
#define ledE 8
#define ledF 4
#define ledG 12
#define DOT 10
#define seg1 2
#define seg2 5
#define seg3 6
#define seg4 13


/*
A~F, 0~9까지 표시하는 함수
7개의 LED를 HIGH(켜짐)/LOW(꺼짐)으로 설정하여 표시한다.
*/
void charfnd(int n) // 문자 출력 함수
{
  switch(n)
  {
    case 1:
      //Displays A
      digitalWrite(ledD, LOW);
      digitalWrite(ledE, HIGH);
      digitalWrite(ledF, HIGH);
      digitalWrite(ledG, HIGH);
      digitalWrite(ledA, HIGH);
      digitalWrite(ledB, HIGH);
      digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;

  case 2:
    //Displays B
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;

  case 3:
    //Displays C
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, LOW);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, LOW);
    digitalWrite(ledC, LOW);
    digitalWrite(DOT , LOW);
    break;

  case 4:
    //Displays D
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, LOW);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;
```

```
        case 5:
          //Displays E
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;

        case 6:
          //Displays F
          digitalWrite(ledD, LOW);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;
    }
}

void fnd(int n) // 숫자 출력 함수
{
    switch(n)
    {

          case 0:
          //Displays 0
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, LOW);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 1:
//Displays 1
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 2:
//Displays 2
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, LOW);
digitalWrite(DOT , LOW);
break;

case 3:
//Displays 3
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;
```

```
case 4:
//Displays 4
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 5:
//Displays 5
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 6:
//Displays 6
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 7:
//Displays 7
```

```
            digitalWrite(ledD, LOW);
            digitalWrite(ledE, LOW);
            digitalWrite(ledF, LOW);
            digitalWrite(ledG, LOW);
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(DOT , LOW);
            break;

            case 8:
            //Displays 8
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, HIGH);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, HIGH);
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(DOT , LOW);
            break;

            case 9:
            //Displays 9
            digitalWrite(ledD, HIGH);
            digitalWrite(ledE, LOW);
            digitalWrite(ledF, HIGH);
            digitalWrite(ledG, HIGH);
            digitalWrite(ledA, HIGH);
            digitalWrite(ledB, HIGH);
            digitalWrite(ledC, HIGH);
            digitalWrite(DOT , LOW);
            break;

    }
}
void clr() //led clear 함수
{
    //Clears the LED
```

```
   digitalWrite(ledA, LOW);
   digitalWrite(ledB, LOW);
   digitalWrite(ledC, LOW);
   digitalWrite(ledD, LOW);
   digitalWrite(ledE, LOW);
   digitalWrite(ledF, LOW);
   digitalWrite(ledG, LOW);
   digitalWrite(DOT , LOW);
}
void set() // led set 함수
{   //Set the LED
   digitalWrite(ledA, HIGH);
   digitalWrite(ledB, HIGH);
   digitalWrite(ledC, HIGH);
   digitalWrite(ledD, HIGH);
   digitalWrite(ledE, HIGH);
   digitalWrite(ledF, HIGH);
   digitalWrite(ledG, HIGH);
   digitalWrite(DOT , HIGH);
}
```

//3545 Display Program

```
int a[5]= {};   // 각 segment 숫자를 저장할 배열
int b;

void setup()
{
   //각각의 핀들을 OUTPUT으로 설정해 준다.
   pinMode(ledA, OUTPUT);
   pinMode(ledB, OUTPUT);
   pinMode(ledC, OUTPUT);
   pinMode(ledD, OUTPUT);
   pinMode(ledE, OUTPUT);
   pinMode(ledF, OUTPUT);
   pinMode(ledG, OUTPUT);
   pinMode(DOT, OUTPUT);
   pinMode(seg1, OUTPUT);
```

```
  pinMode(seg2, OUTPUT);
  pinMode(seg3, OUTPUT);
  pinMode(seg4, OUTPUT);

Serial.begin(9600);// 시리얼 통신 설정
}

void loop()
{
  if(Serial.available()) //시리얼 통신을 통해 값을 입력 받으면 실행
  {
    for(b=0;b<4;b++) // 배열 a[] 0~3까지 값 입력
    {
      a[b]=Serial.read()-48; /* serial.read() 함수 문은 값 입력시 아스키 코트로 변환 함으로
                              우리가 써야할 0~9 까지의 아스키코드 값은 48~57 이다.
                             따라서 48을 빼 주어야 우리가 입력한 숫자로 입력된다.*/
      Serial.println(a[b]); // a[0]~a[3] 까지의 값이 제대로 입력 되었는지 시리얼 모니터를 통해 확인
     }

   Serial.println("start"); // for 문이 끝났는지 시리얼 모니터링
  }
   /* 각 세그먼트 숫자 출력*/
     digitalWrite(seg1,LOW);
     fnd(a[0]);
     delay(5);
     digitalWrite(seg1,HIGH);

     digitalWrite(seg2,LOW);
     fnd(a[1]);
     delay(5);
     digitalWrite(seg2,HIGH);

     digitalWrite(seg3,LOW);
     fnd(a[2]);
     delay(5);
     digitalWrite(seg3,HIGH);

     digitalWrite(seg4,LOW);
```

```
        fnd(a[3]);
        delay(5);
        digitalWrite(seg4,HIGH);
 }
```

- Program 설명

- if(data == 1), if(data == '1')

  -> HIGH 와 char 형 data 수신

- Serial.available();

  -> Serial data 수신 시 0 보다 큼

- Serial.read();

  -> Rx : 수신된 Serial data 1 byte 읽어옴

->Serial.read() 함수를 이용하여 컴퓨터로부터 값을 입력 받으면 ASCII Code 값으로 입력 됨

예) 12 입력시 개별 문자로 인식하여 ASCII Code 값 '49', '50' 으로 입력 됨

- Serial.parseInt();

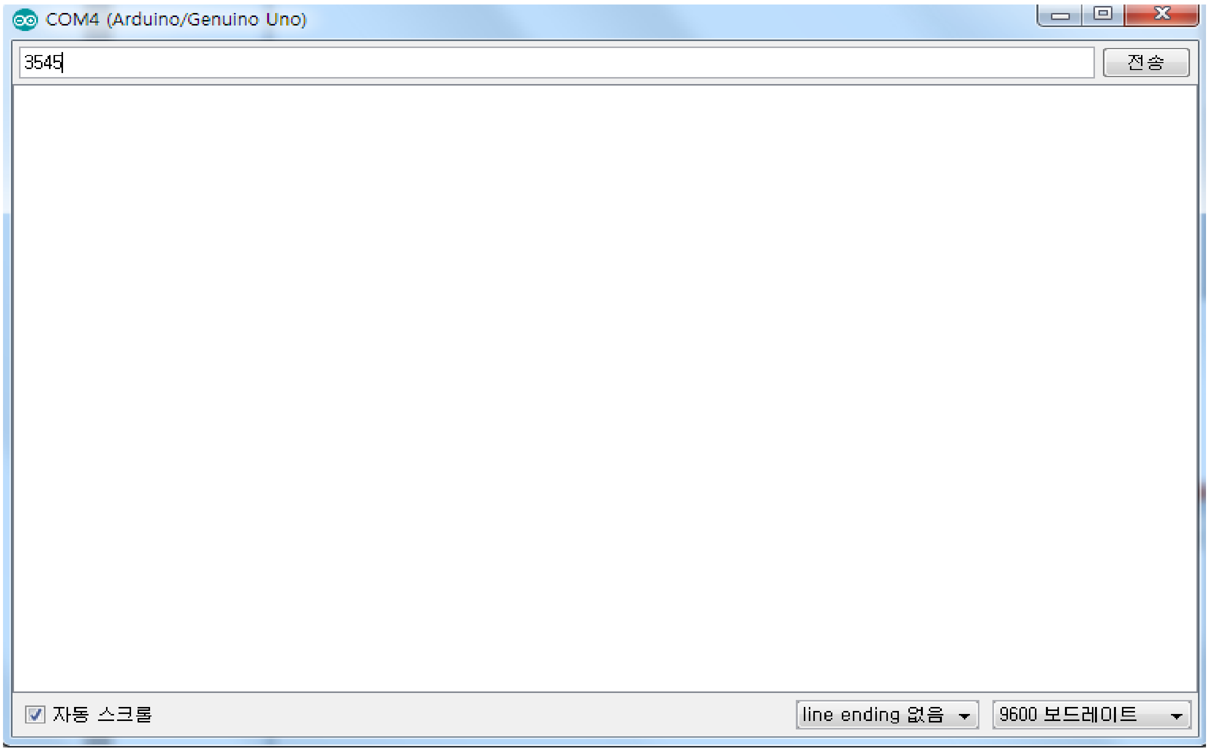
  -> 위 함수는 ASCII 값이 아닌 정수 값이 입력됨

  -> 정수 값을 입력 받아서 PWM 등에 바로 적용 가능

- Serial.println("LED1 ON");

  -> Tx : Serial 로 data 전송

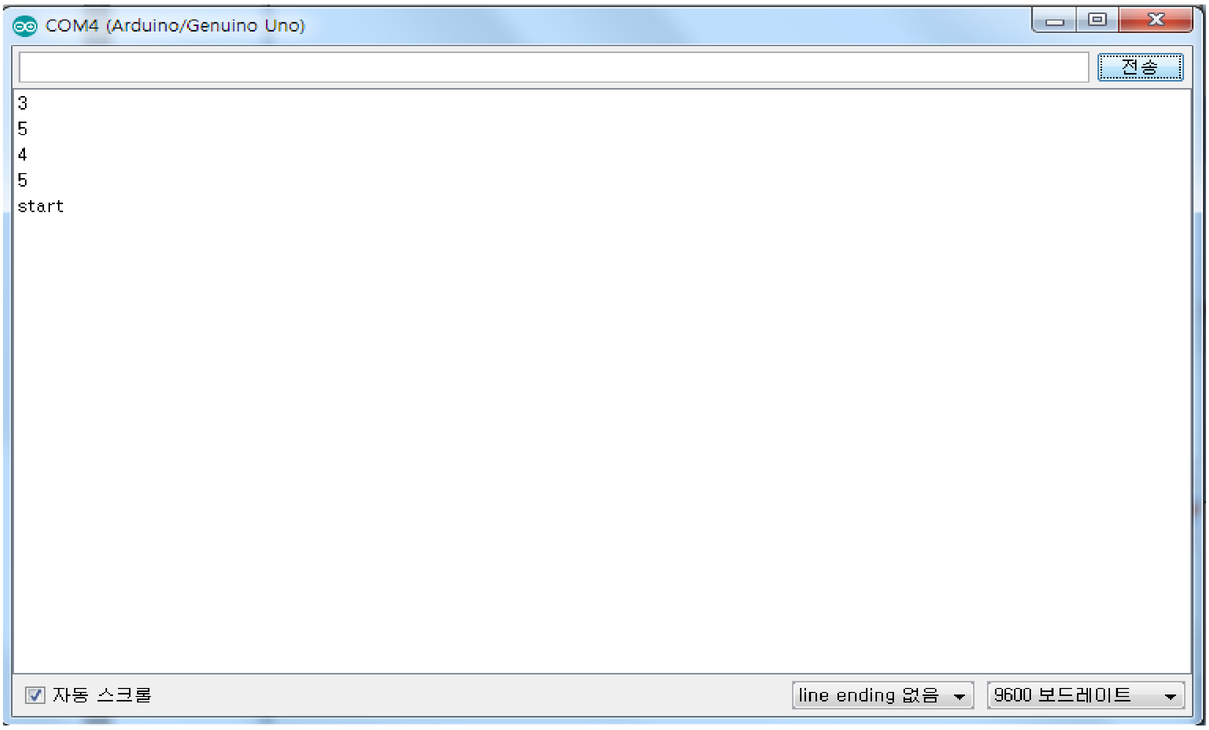
Serial 값 입력



COM4 (Arduino/Genuino Uno)

```
3545
```

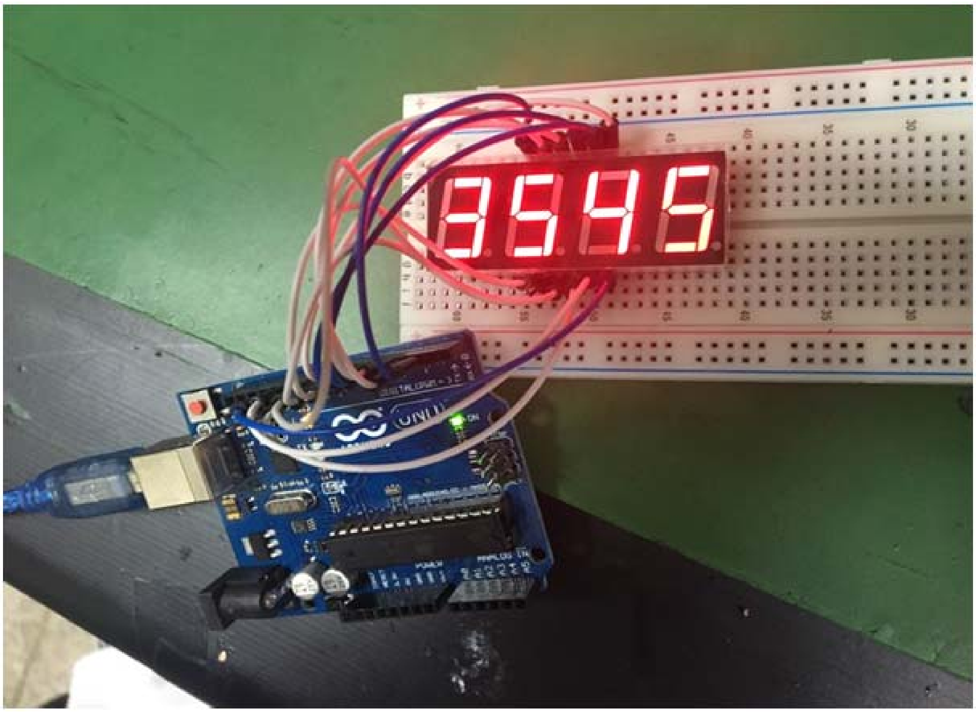
Serial 값 입력 후 Serial Monitor 출력 값



COM4 (Arduino/Genuino Uno)

```
3
5
4
5
start
```

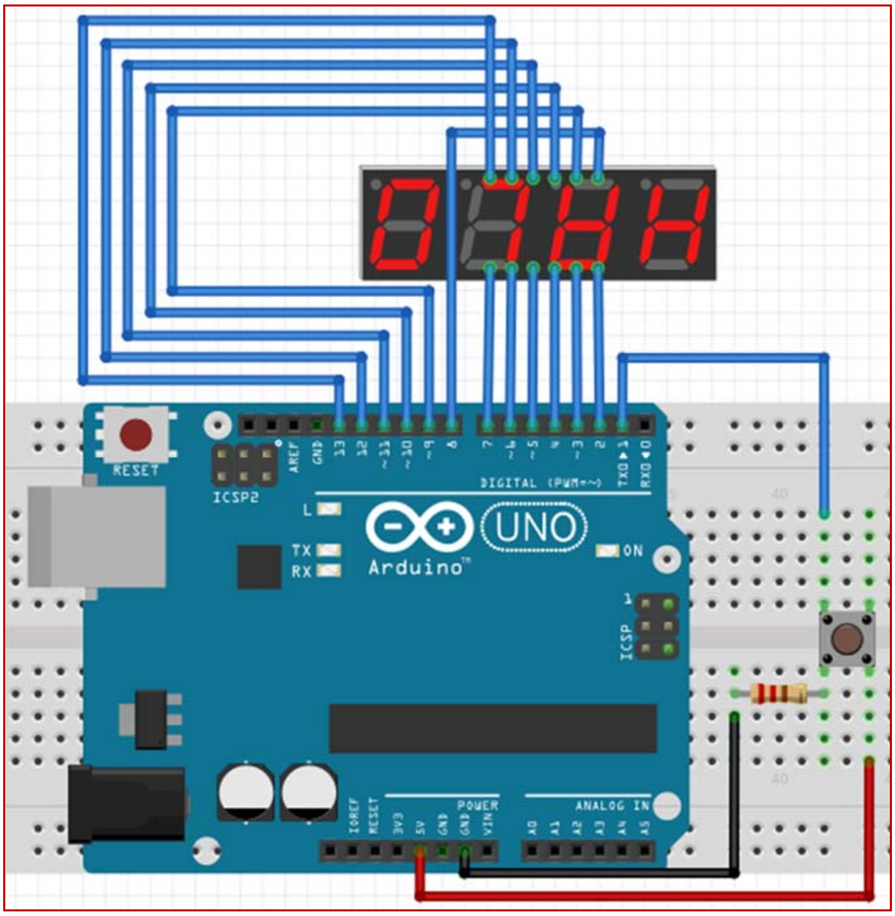
실험 결과



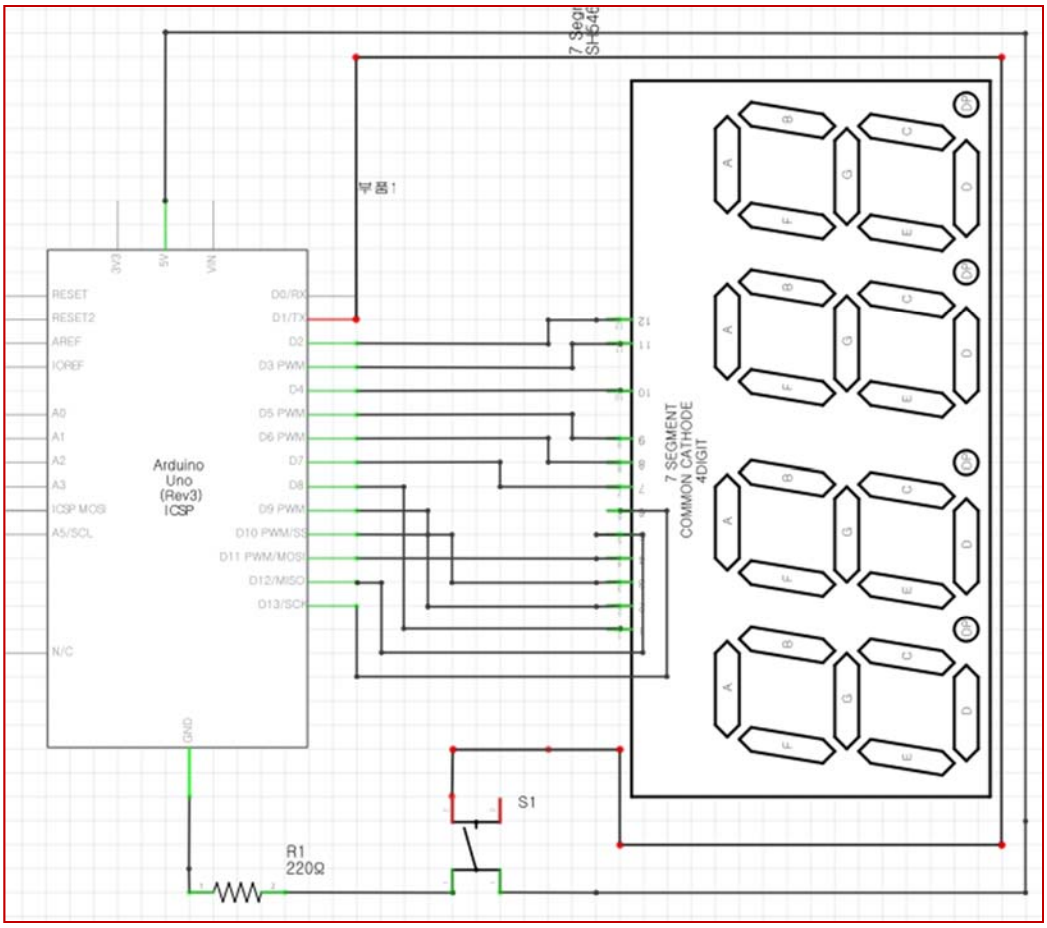
EX. 3) Switch를 이용한 4 digit 7-segment에 Display 하기

- 4 digit 7-segment를 처음에 0000으로 출력하도록 하고 디지털 입력 1번 Pin에 스위치를 연결하여 스위치를 누를 때마다 숫자가 1씩 증가하는 프로그램을 작성함.

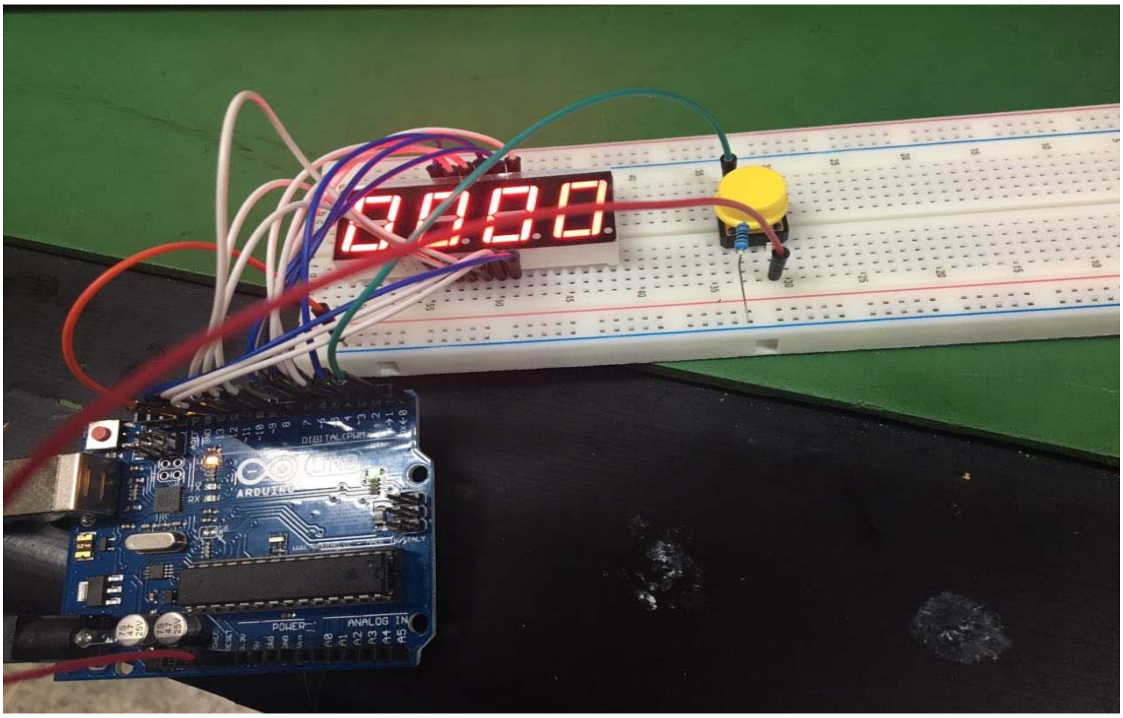
- 반드시 실험을 해보고 결과를 교수님한테 검사를 받는다.



[Switch를 이용한 배선도]

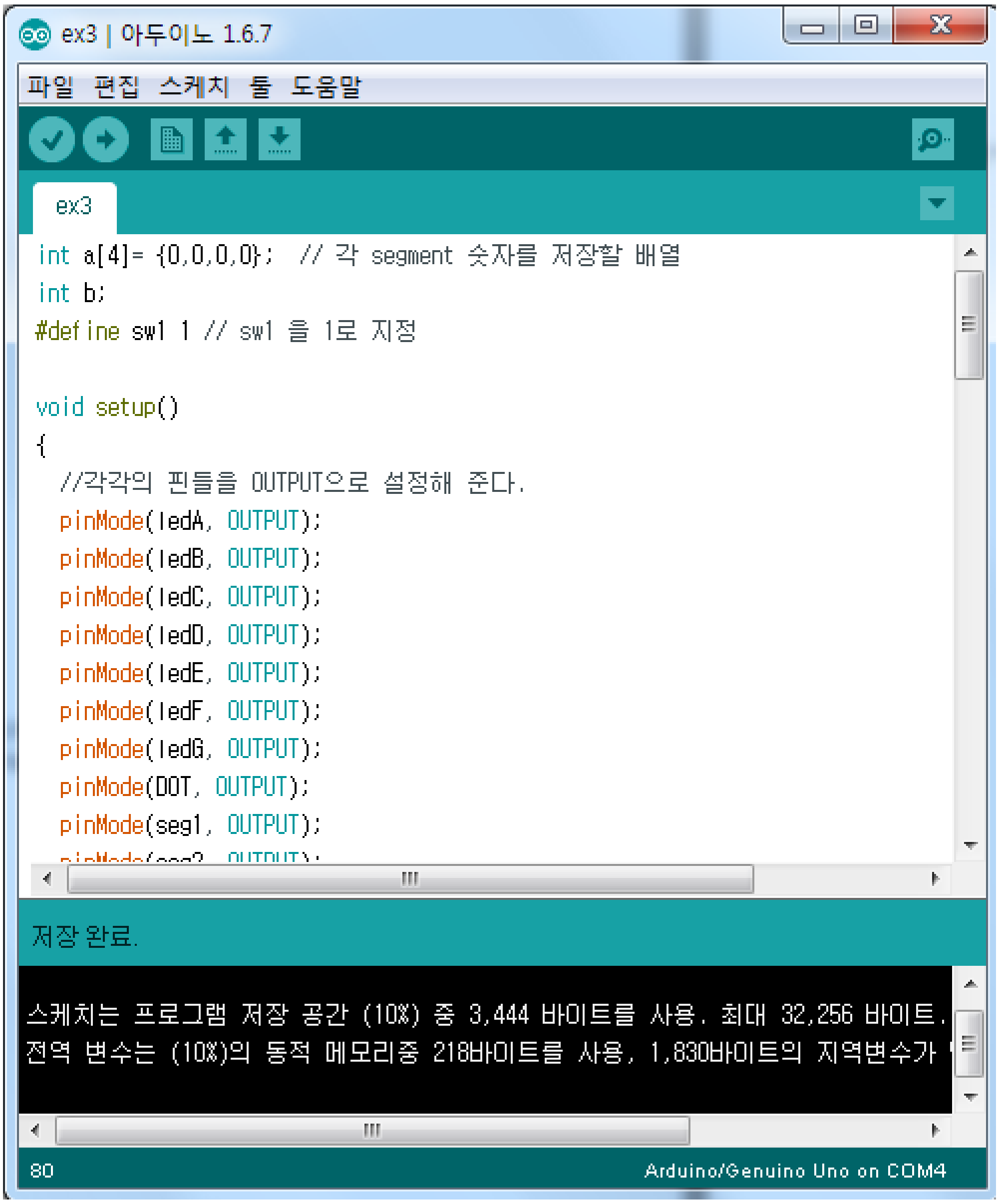


[회로도]



[결선 회로- 초기값 0000]

프로그램 입력



```
int a[4]= {0,0,0,0};  // 각 segment 숫자를 저장할 배열
int b;
#define sw1 1 // sw1 을 1로 지정

void setup()
{
  //각각의 핀들을 OUTPUT으로 설정해 준다.
  pinMode(ledA, OUTPUT);
  pinMode(ledB, OUTPUT);
  pinMode(ledC, OUTPUT);
  pinMode(ledD, OUTPUT);
  pinMode(ledE, OUTPUT);
  pinMode(ledF, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(DOT, OUTPUT);
  pinMode(seg1, OUTPUT);
  pinMode(seg2, OUTPUT);
```

저장 완료.

```
스케치는 프로그램 저장 공간 (10%) 중 3,444 바이트를 사용. 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리중 218바이트를 사용, 1,830바이트의 지역변수가
```

Program File

// Header File 역할 – 숫자, 문자 Display 및 Clear, Set 등

```
#include<arduino.h>

#define ledA 3
#define ledB 7
#define ledC 11
#define ledD 9
#define ledE 8
#define ledF 4
#define ledG 12
#define DOT 10
#define seg1 2
#define seg2 5
#define seg3 6
#define seg4 13


/*
A~F, 0~9까지 표시하는 함수
7개의 LED를 HIGH(켜짐)/LOW(꺼짐)으로 설정하여 표시한다.
*/
void charfnd(int n) // 문자 출력 함수
{
  switch(n)
  {
    case 1:
      //Displays A
      digitalWrite(ledD, LOW);
      digitalWrite(ledE, HIGH);
      digitalWrite(ledF, HIGH);
      digitalWrite(ledG, HIGH);
      digitalWrite(ledA, HIGH);
      digitalWrite(ledB, HIGH);
      digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;

  case 2:
    //Displays B
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;

  case 3:
    //Displays C
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, LOW);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, LOW);
    digitalWrite(ledC, LOW);
    digitalWrite(DOT , LOW);
    break;

  case 4:
    //Displays D
    digitalWrite(ledD, HIGH);
    digitalWrite(ledE, HIGH);
    digitalWrite(ledF, HIGH);
    digitalWrite(ledG, LOW);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledB, HIGH);
    digitalWrite(ledC, HIGH);
    digitalWrite(DOT , LOW);
    break;
```

```
        case 5:
          //Displays E
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;

        case 6:
          //Displays F
          digitalWrite(ledD, LOW);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, LOW);
          digitalWrite(ledC, LOW);
          digitalWrite(DOT , LOW);
          break;
    }
}

void fnd(int n) // 숫자 출력 함수
{
    switch(n)
    {

          case 0:
          //Displays 0
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, LOW);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 1:
//Displays 1
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, LOW);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 2:
//Displays 2
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, LOW);
digitalWrite(DOT , LOW);
break;

case 3:
//Displays 3
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, LOW);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;
```

```
case 4:
//Displays 4
digitalWrite(ledD, LOW);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, LOW);
digitalWrite(ledB, HIGH);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 5:
//Displays 5
digitalWrite(ledD, HIGH);
digitalWrite(ledE, LOW);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 6:
//Displays 6
digitalWrite(ledD, HIGH);
digitalWrite(ledE, HIGH);
digitalWrite(ledF, HIGH);
digitalWrite(ledG, HIGH);
digitalWrite(ledA, HIGH);
digitalWrite(ledB, LOW);
digitalWrite(ledC, HIGH);
digitalWrite(DOT , LOW);
break;

case 7:
//Displays 7
```

```
          digitalWrite(ledD, LOW);
          digitalWrite(ledE, LOW);
          digitalWrite(ledF, LOW);
          digitalWrite(ledG, LOW);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
          digitalWrite(ledC, HIGH);
          digitalWrite(DOT , LOW);
          break;

          case 8:
          //Displays 8
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, HIGH);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
          digitalWrite(ledC, HIGH);
          digitalWrite(DOT , LOW);
          break;

          case 9:
          //Displays 9
          digitalWrite(ledD, HIGH);
          digitalWrite(ledE, LOW);
          digitalWrite(ledF, HIGH);
          digitalWrite(ledG, HIGH);
          digitalWrite(ledA, HIGH);
          digitalWrite(ledB, HIGH);
          digitalWrite(ledC, HIGH);
          digitalWrite(DOT , LOW);
          break;

   }
}
void clr() //led clear 함수
{
   //Clears the LED
```

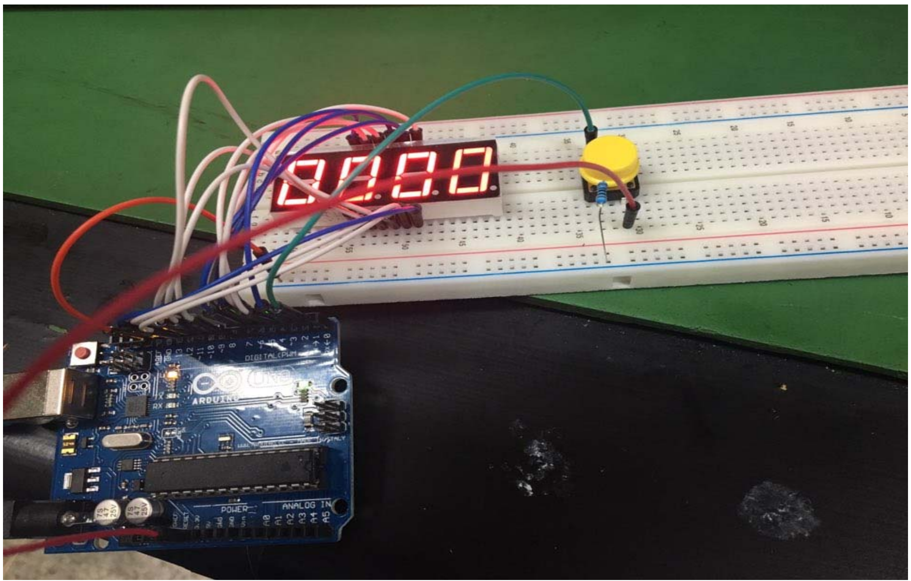
```
  digitalWrite(ledA, LOW);
  digitalWrite(ledB, LOW);
  digitalWrite(ledC, LOW);
  digitalWrite(ledD, LOW);
  digitalWrite(ledE, LOW);
  digitalWrite(ledF, LOW);
  digitalWrite(ledG, LOW);
  digitalWrite(DOT , LOW);
}
void set() // led set 함수
{   //Set the LED
  digitalWrite(ledA, HIGH);
  digitalWrite(ledB, HIGH);
  digitalWrite(ledC, HIGH);
  digitalWrite(ledD, HIGH);
  digitalWrite(ledE, HIGH);
  digitalWrite(ledF, HIGH);
  digitalWrite(ledG, HIGH);
  digitalWrite(DOT , HIGH);
}
```

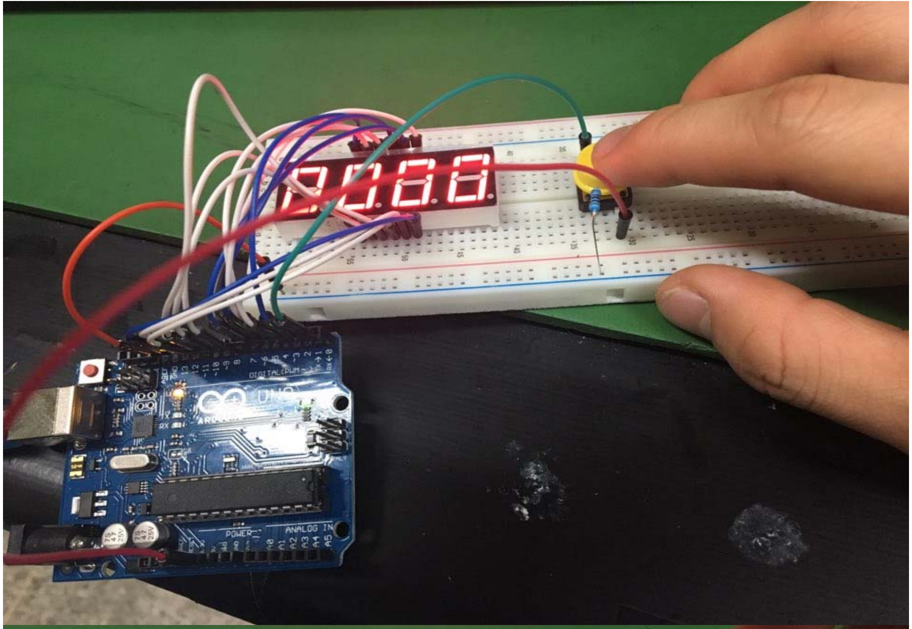
//Switch를 누를 때마다 1씩 증가히는 Display Program

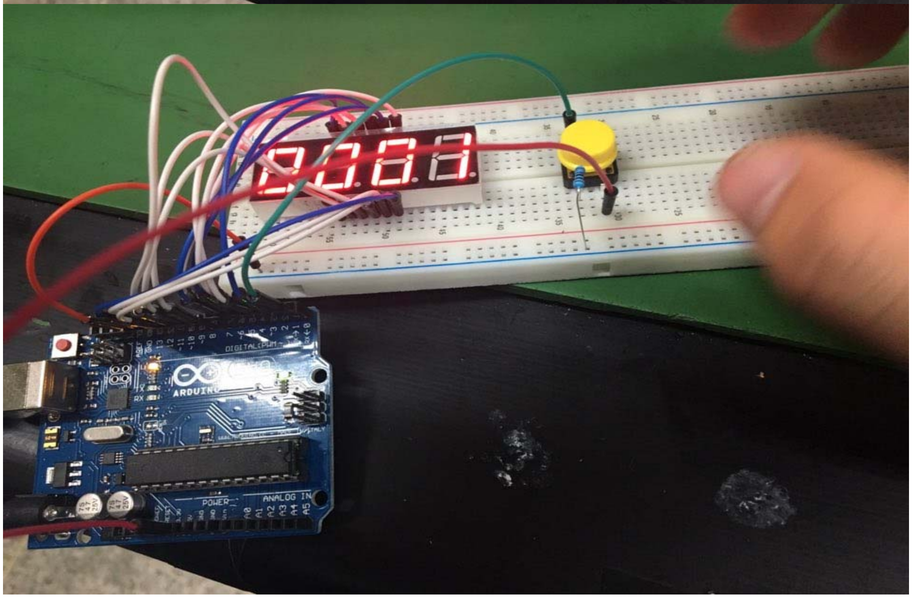
```
int a[4]= {0,0,0,0};   // 각 segment 숫자를 저장할 배열
int b;
#define sw1 1 // sw1 을 1로 지정

void setup()
{
  //각각의 핀들을 OUTPUT으로 설정해 준다.
  pinMode(ledA, OUTPUT);
  pinMode(ledB, OUTPUT);
  pinMode(ledC, OUTPUT);
  pinMode(ledD, OUTPUT);
  pinMode(ledE, OUTPUT);
  pinMode(ledF, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(DOT, OUTPUT);
  pinMode(seg1, OUTPUT);
```

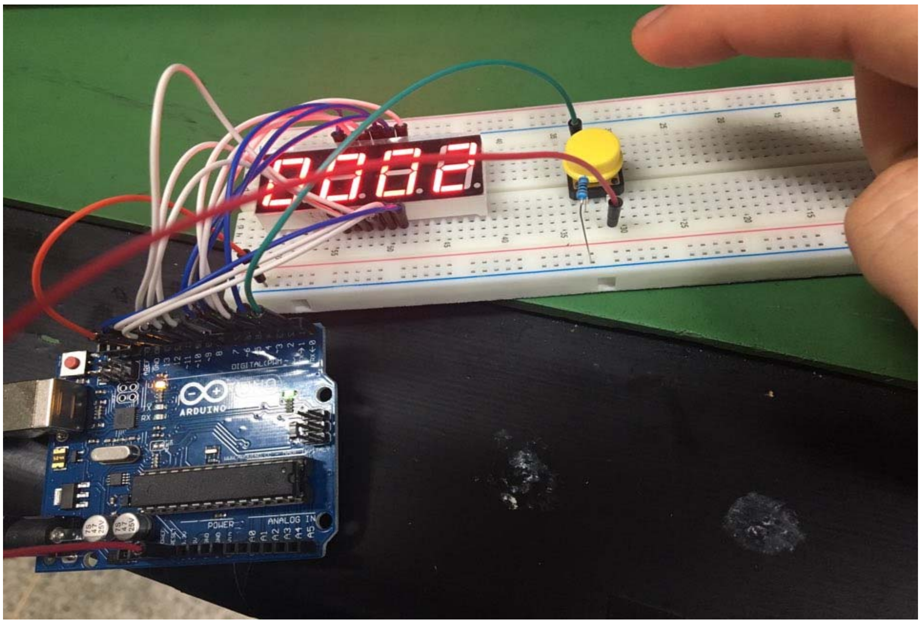
```
   pinMode(seg2, OUTPUT);
   pinMode(seg3, OUTPUT);
   pinMode(seg4, OUTPUT);
   pinMode(sw1, INPUT); // 1번포트에 값을 입력받는다.

Serial.end(); /* Serial.end는 Serial통신을 사용하지 않도록 하는 함수입니다.
Serial.begin()이 Serial 통신을 사용하기 위해 사용되는 것이면,
Serial.end()는 Serial에 사용되는 핀(디지털 0, 1번핀)을 일반 디지털 핀으로 사용하게 하는 함수*/
}

void loop()
{
    int sw1_state = digitalRead(sw1);   //스위치가 연결된 1번포트의 값을 읽어온다.

    if(sw1_state==1) // 스위치가 눌릴 때 마다 segment의 숫자를 증가 시킨다.
    {
      ++a[3];
      if(a[3]>9)
      {
       a[3]=0;
       ++a[2];
      }
      if(a[2]>9)
      {
       a[2]=0;
       ++a[1];
      }
      if(a[1]>9)
      {
       a[1]=0;
       ++a[0];
      }
      if(a[0]>9)
      {
       a[0]=0;
      }
    }
    // segment 숫자 출력
      for(b=0;b<5;b++) /* sw1의 입력이 여러번 들어가는것을 방지
```

```
                                        delay  역할*/
        {
        digitalWrite(seg1,LOW);
        fnd(a[0]);
        delay(5);
        digitalWrite(seg1,HIGH);

        digitalWrite(seg2,LOW);
        fnd(a[1]);
        delay(5);
        digitalWrite(seg2,HIGH);

        digitalWrite(seg3,LOW);
        fnd(a[2]);
        delay(5);
        digitalWrite(seg3,HIGH);

        digitalWrite(seg4,LOW);
        fnd(a[3]);
        delay(5);
        digitalWrite(seg4,HIGH);
        }
    }
```









[Switch 를 누를 때마다 숫자가 증가 함]

Challenge 5-1)

4 digit 7-segment에 ABCD를 Display하는 프로그램을 작성하여 실험을 한다.

Challenge 5-2)

Serial Communication을 이용하여 Serial Monitor로 4 digit 7-segment에 4321을 입력하여 4321을 Display하는 프로그램을 작성하여 실험을 한다.

Challenge 5-3)

4 digit 7-segment 와 스위치 1개를 사용하여 처음에는 0000을 출력을 하고 스위치를 누를 때마다 2씩 숫자가 증가하는 프로그램을 작성한다.

GoldenBell 5-1)

4 digit 7-segment에 Indu를 Display하는 프로그램을 작성하여 실험을 한다.

GoldenBell 5-2)

Serial Communication을 이용한 4 digit 7-segment에 4321을 입력하면 1234로 Display하는 프로그램을 작성하여 실험을 한다.

GoldenBell 5-3)

Switch를 이용한 4 digit 7-segment Display에서 처음에는 0000을 출

력하도록 하고 디지털 입력 1번 Pin에 스위치를 연결하여 스위치를 누를 때마다 숫자가 1씩 증가하는 프로그램을 작성한다.
이때 Display되는 순서는 LSB의 4 digit 7-segment부터 Display하여 MSB의 4 digit 7-segment방향으로 Display를 한다. 즉 앞의 예제와 Display 하는 순서가 반대 방향 임.
합산 값이 10을 넘으면 Display되는 순서는 MSB의 4 digit 7-segment부터 Display하여 LSB의 4 digit 7-segment방향으로 Display를 한다. 즉 위와 순서가 반대 방향 임.

GoldenBell 5-4)

Switch3개와 4 digit 7-segment Display를 이용하여 초시계를 만들어 본다.
처음에는 0000을 출력하도록 하고 스위치를 1을 누르면 1초씩 증가한다. 스위치 2를 누르면 증가하던 초시계가 멈춘다. 이때 스위치 1을 다시 누르면 이어서 1초씩 증가하고 스위치 2를 누르면 멈춘다. 이와 같은 동작을 반복하다가 스위치 3을 누르면 현재의 값이 Reset되어 0000을 출력한다.