



LED 테트리스

조장 : 1260014 김 철 언
조원 : 1260027 신 중 혁
1260030 오 연 중
1260053 정 회 현

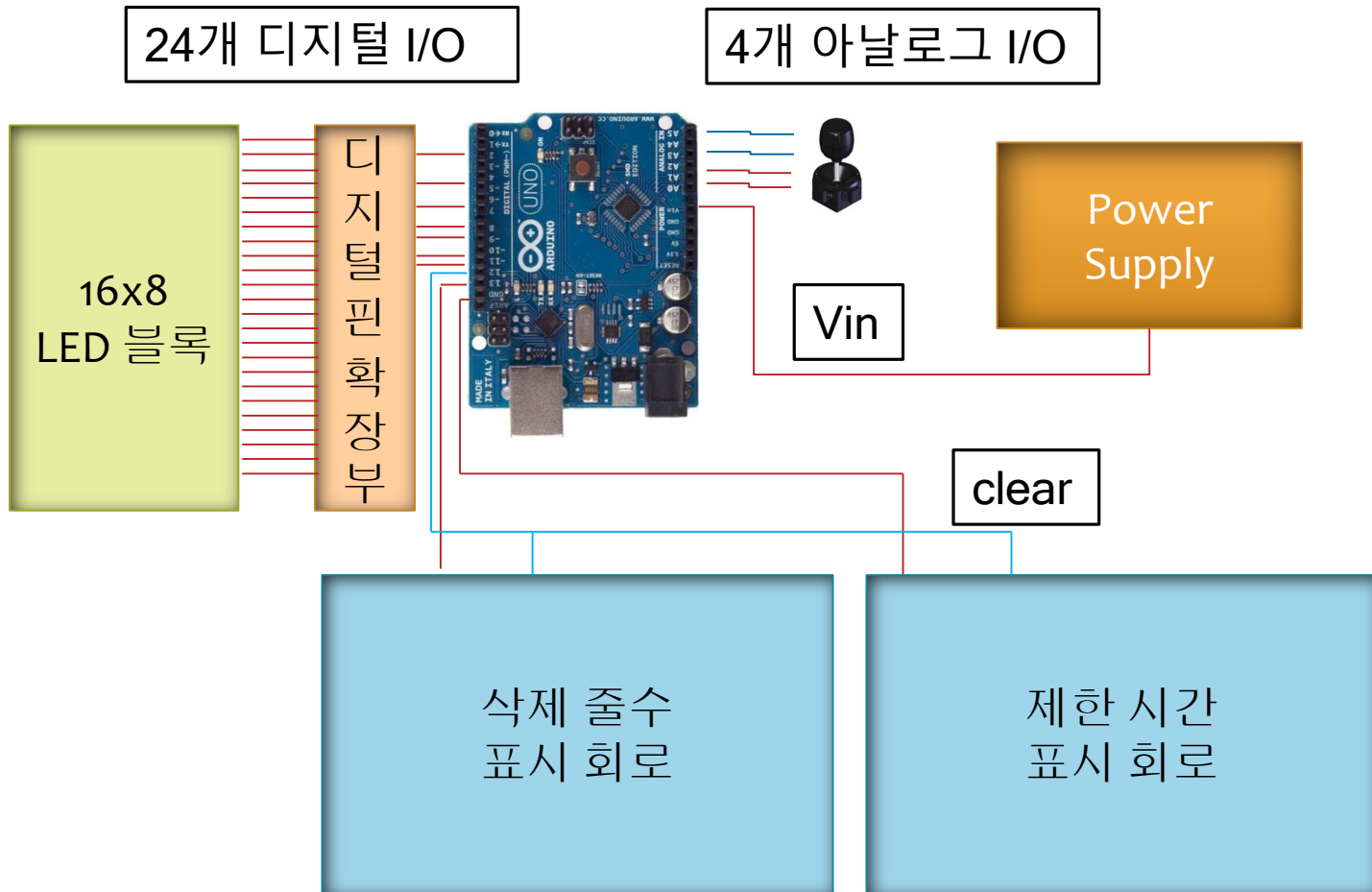
목차

- ▶ 시스템 개요
- ▶ 블록도
- ▶ Arduino Sketch 코드
 - Arduino Sketch 주요 코드
 - LED 출력 코드
- ▶ LED 출력
 - 잔상효과
 - shift register를 이용한 잔상효과 구현
- ▶ 삭제 줄 수 표시 회로
 - 줄 수 삭제 표시 원리
 - 증감연산자 방식을 이용한 삭제 줄 수 표시
- ▶ 제한 시간 표시 회로
 - 제한 시간 표시 원리
 - 99 계수기를 이용한 제한시간 표시
- ▶ 일정표
- ▶ 사용부품 및 가격

시스템 개요

- ✓ Arduino Board에서 블록 생성 및 하강
- ✓ 아날로그 스틱으로 블록 조종 신호 입력
- ✓ Arduino Board에 입력 및 내부에서 바뀌는 신호를 처리하여 LED로 출력
- ✓ 삭제된 줄 수를 증감연산자 역할을 통해 7-segment로 표현
- ✓ 제한 시간을 카운터로 출력

시스템 블록도



○○ Arduino Sketch 코드 작성 ○○

블록 선언 부

```
// 0
// 0
// 0
// 0
if (blocktype == 0)
{
    block[3][0]=1;
    block[3][1]=1;
    block[3][2]=1;
    block[3][3]=1;
}

// 블록을 선언하는 함수
void newBlock()
{
    check_gameover();

    blocktype = random(7);

    // 0
    // 0 0 0
    if (blocktype == 1)
    {
        block[2][0]=1;
        block[2][1]=1;
        block[3][1]=1;
        block[4][1]=1;
    }

    // 0
    // 0 0 0
    if (blocktype == 2)
    {
        block[4][0]=1;
        block[2][1]=1;
        block[3][1]=1;
        block[4][1]=1;
    }

    // 0 0
    // 0 0
    if (blocktype == 3)
    {
        block[3][0]=1;
        block[3][1]=1;
        block[4][0]=1;
        block[4][1]=1;
    }

    // 0 0
    // 0 0
    if (blocktype == 4)
    {
        block[4][0]=1;
        block[5][0]=1;
        block[3][1]=1;
        block[4][1]=1;
    }

    // 0
    // 0 0 0
    if (blocktype == 5)
    {
        block[4][0]=1;
        block[3][1]=1;
        block[4][1]=1;
        block[5][1]=1;
    }
```

버튼 입력 시 동작

```
//키를 아무것도 입력하지 않을 때
if (delays < millis())
{
    delays = millis() + delay_;
    movedown(); //0.5초마다 블록이 내려옴
}

//버튼 입력시
int button = readBut();

if (button == 1) //위-회전
    rotate();
if (button == 2) //오른쪽-오른쪽 이동
    moveright();
if (button == 3) //왼쪽-왼쪽 이동
    moveleft();
if (button == 4) //아래-아래로 이동
    movedown();
```

LED 출력 코드

LED 출력 과정 코드

```
void LEDRefresh()
{
    int i;
    int k;

    boolean tmpdisp[8][16];
    for (k=0;k<16;k++)
    {
        for(i=1;i<8;i++)
        {
            tmpdisp[i][k]=disp[i-1][k];
        }
        tmpdisp[0][k]=disp[7][k];
    }
}
```

```
for(i=0;i<8;i++)
{
    int j;

    if (i == 0)
        j = rowPin+7;
    else
        j = rowPin+i-1;
```

```
byte upper = 0;
int b;
for(b = 0;b<8;b++)
{
    upper <<= 1;
    if (!tmpdisp[b][i]) upper |= 1;
}
```

```
byte lower = 0;
for(b = 0;b<8;b++)
{
    lower <<= 1;
    if (!tmpdisp[b][i+8]) lower |= 1;
}
```

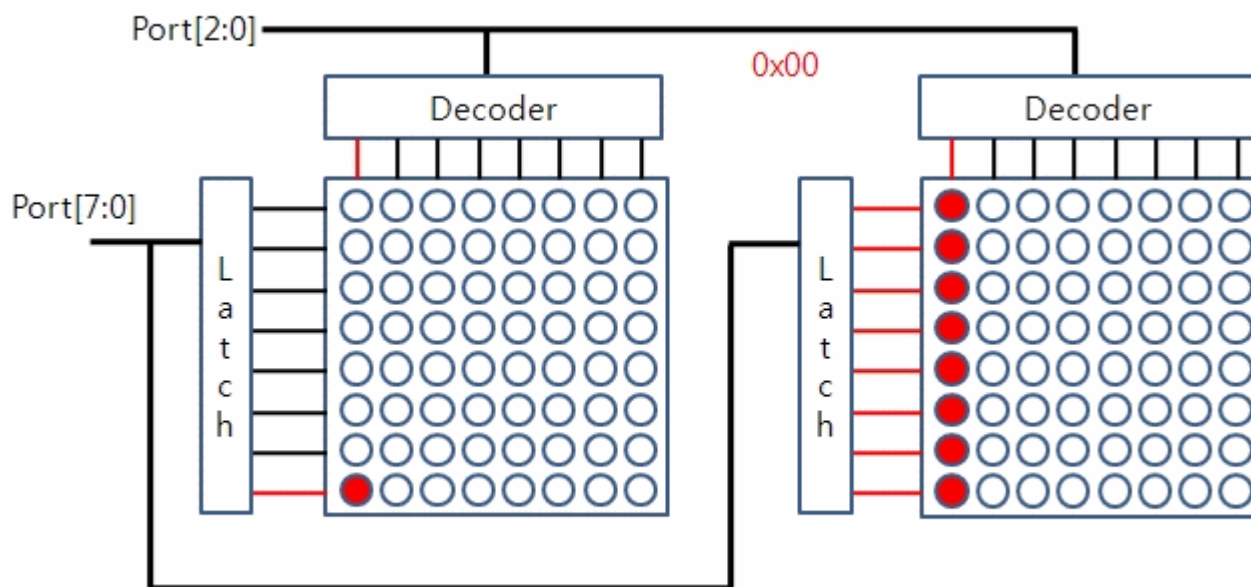
```
digitalWrite(j,LOW);
digitalWrite(latchPin, LOW);
shiftOut(dataPin, clockPin, LSBFIRST, lower);
shiftOut(dataPin, clockPin, LSBFIRST, upper);
digitalWrite(latchPin, HIGH);
digitalWrite(rowPin+i,HIGH);
delay(1);
}
digitalWrite(rowPin+7,LOW);
}
```



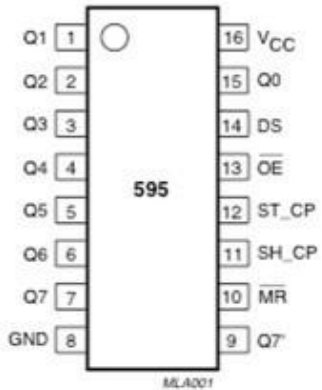
잔상효과



아두이노 보드를 이용하여 Delay를 매우 짧게 발생시켜 잔상이 남는 현상을 이용.



○○ Shift register를 이용한 잔상 효과 ○○



PINS 1-7, 15	Output Pins (병렬출력)
PIN 8	GND
PIN 9	Serial Out(다음IC로 데이터를 넘김)
PIN 10	Master Reclear, active low (+연결한다)
PIN 11	Shift register clock pin (아두이노와 연결)
PIN 12	Storage register clock pin (latch pin) (아두이노와 연결)
PIN 13	Output enable, active low (GND연결한다)
PIN 14	Serial data input (아두이노와 연결)
PIN 16	Positive supply voltage (Vcc)

74hc595 IC칩을 이용
8bit 시프트 레지스터
(병렬 출력)
+
아두이노 shiftOut() 함수

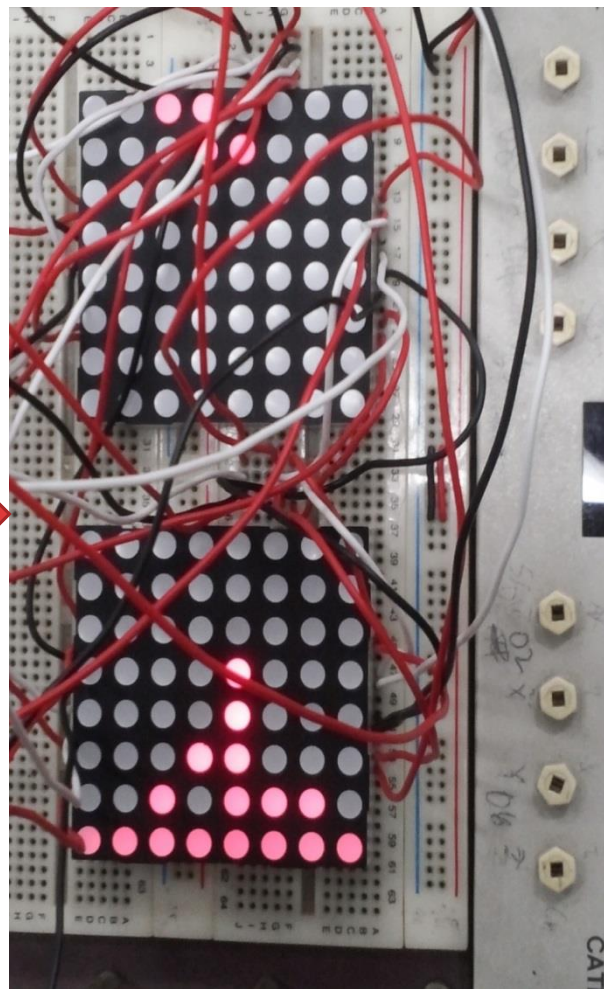
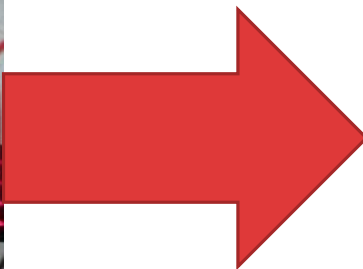
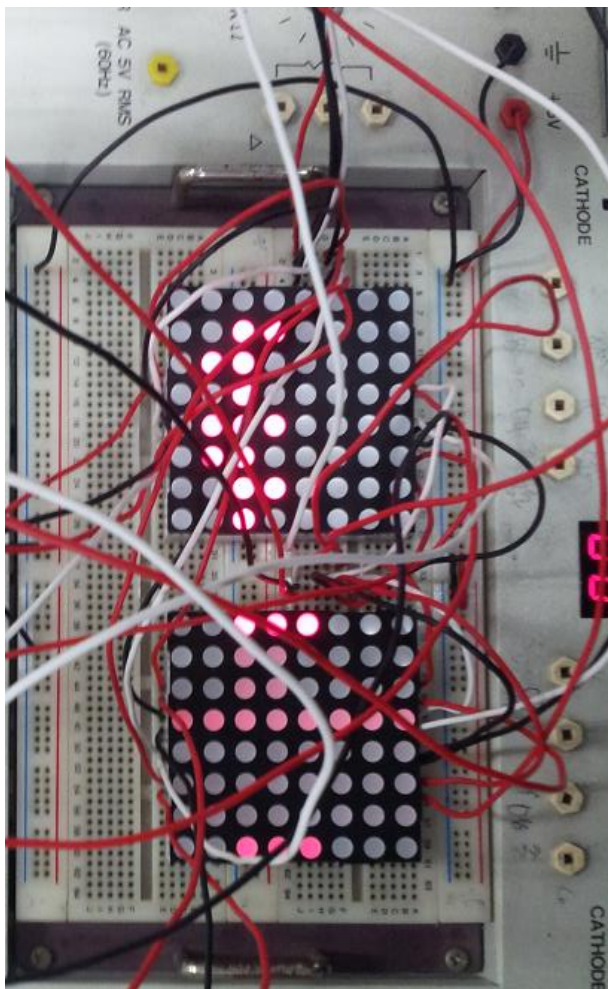
```
byte upper = 0;
int b;
for(b = 0;b<8;b++)
{
    upper <<= 1;
    if (!tmpdisp[b][i]) upper |= 1;
}
```

```
byte lower = 0;
for(b = 0;b<8;b++)
{
    lower <<= 1;
    if (!tmpdisp[b][i+8]) lower |= 1;
}
```

```
shiftOut(dataPin, clockPin, LSBFIRST, lower);
shiftOut(dataPin, clockPin, LSBFIRST, upper);
```



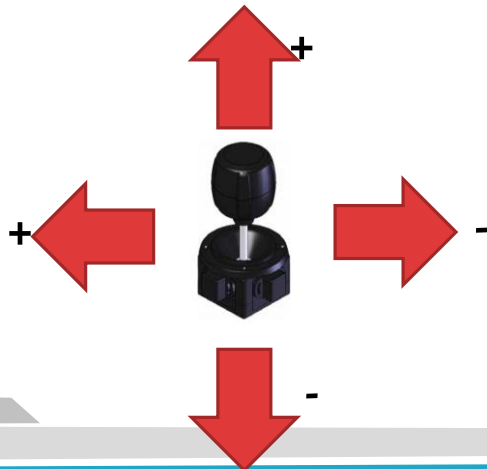

제작 결과 (1)



제작 결과 (2)

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  UD = analogRead(A0);  
  LR = analogRead(A1);  
  Serial.print("UD = ");  
  Serial.print(UD, DEC);  
  Serial.print(", LR = ");  
  Serial.println(LR, DEC);  
  delay(200);  
}
```

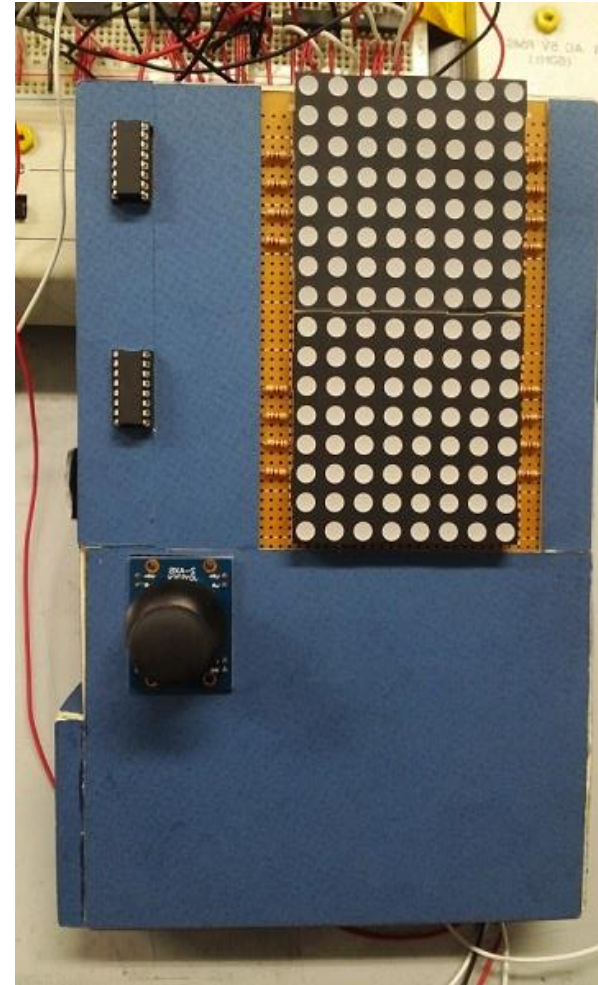
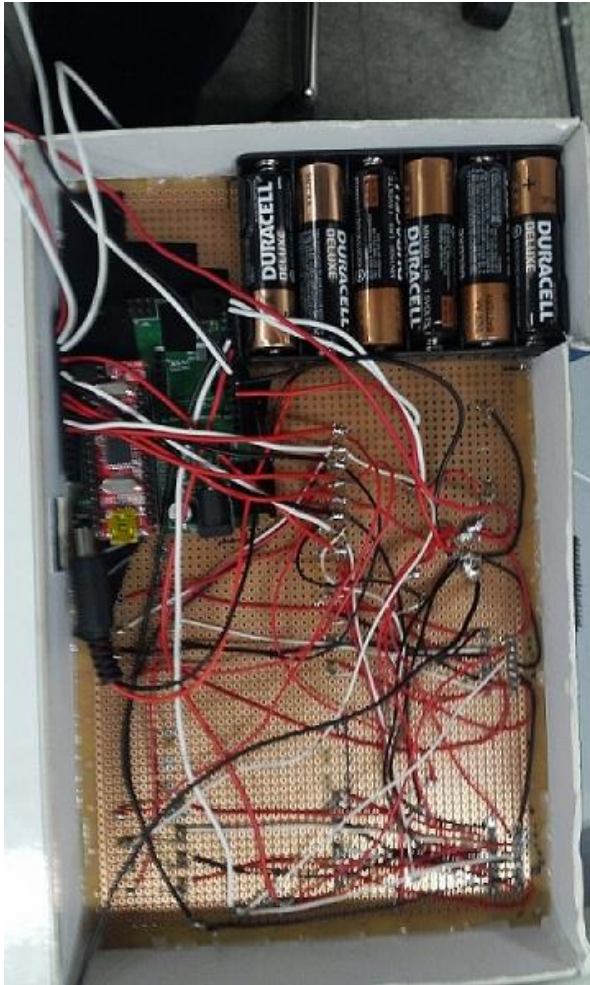
UD = 527, LR = 526
UD = 528, LR = 526
UD = 527, LR = 526
UD = 527, LR = 526
UD = 527, LR = 526
UD = 527, LR = 526
UD = 527, LR = 526
UD = 527, LR = 526
UD = 528, LR = 526
UD = 715, LR = 526
UD = 1022, LR = 526
UD = 1022, LR = 526
UD = 1022, LR = 526
UD = 1022, LR = 526



readBut()

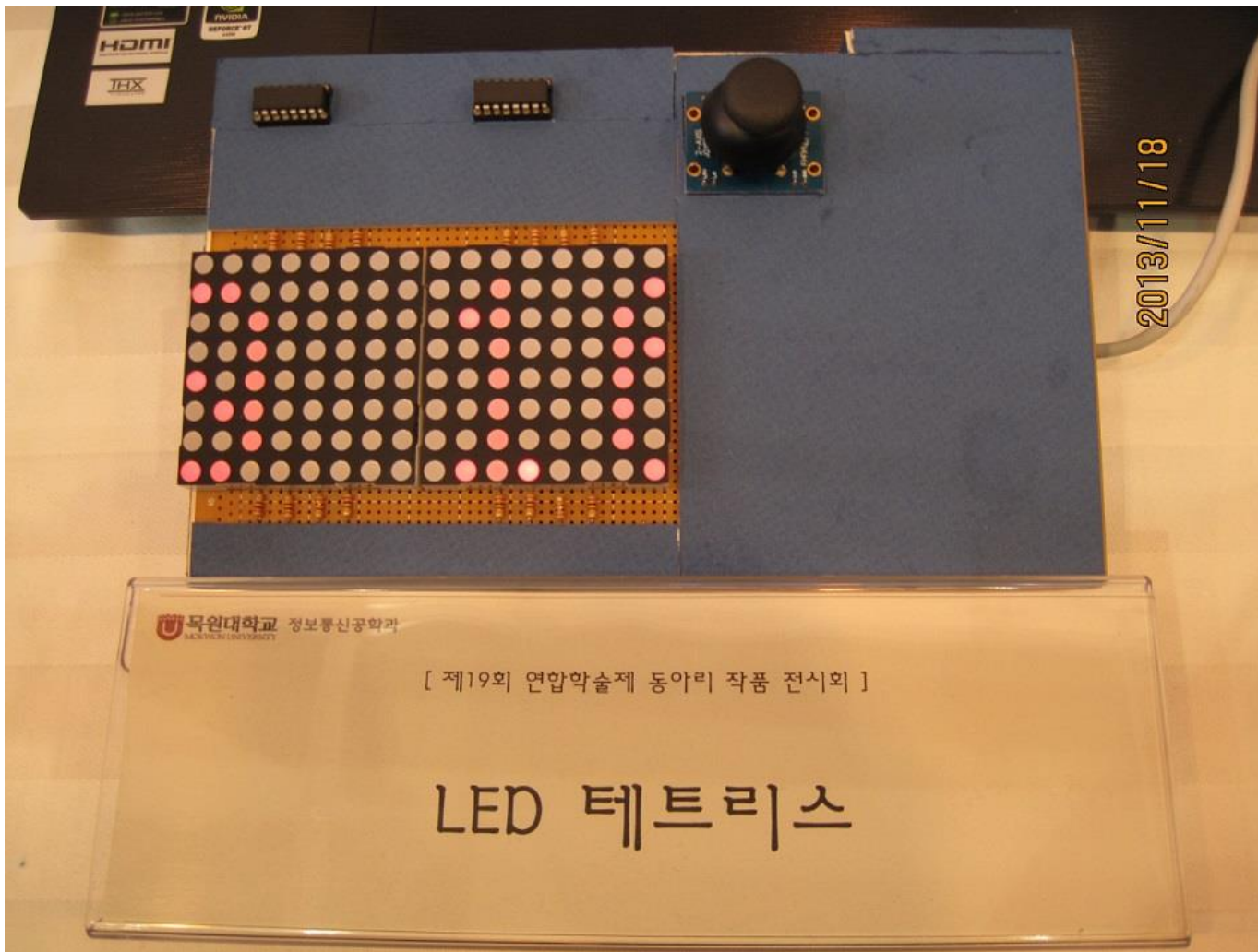
```
if (analogRead(A4) > 750)  
{  
  //left  
  bdelay = millis() + btsidedelay;  
  return 3;  
}  
  
if (analogRead(A5) < 350)  
{  
  //down  
  bdelay = millis() + btowndelay;  
  return 4;  
}  
  
if (analogRead(A4) < 350)  
{  
  //right  
  bdelay = millis() + btsidedelay;  
  return 2;  
}  
  
if (analogRead(A5) > 750)  
{  
  //up  
  bdelay = millis() + buttowndelay;  
  return 1;  
}
```

회로 제작 결과





학술제 발표



줄 수 삭제 원리

Arduino

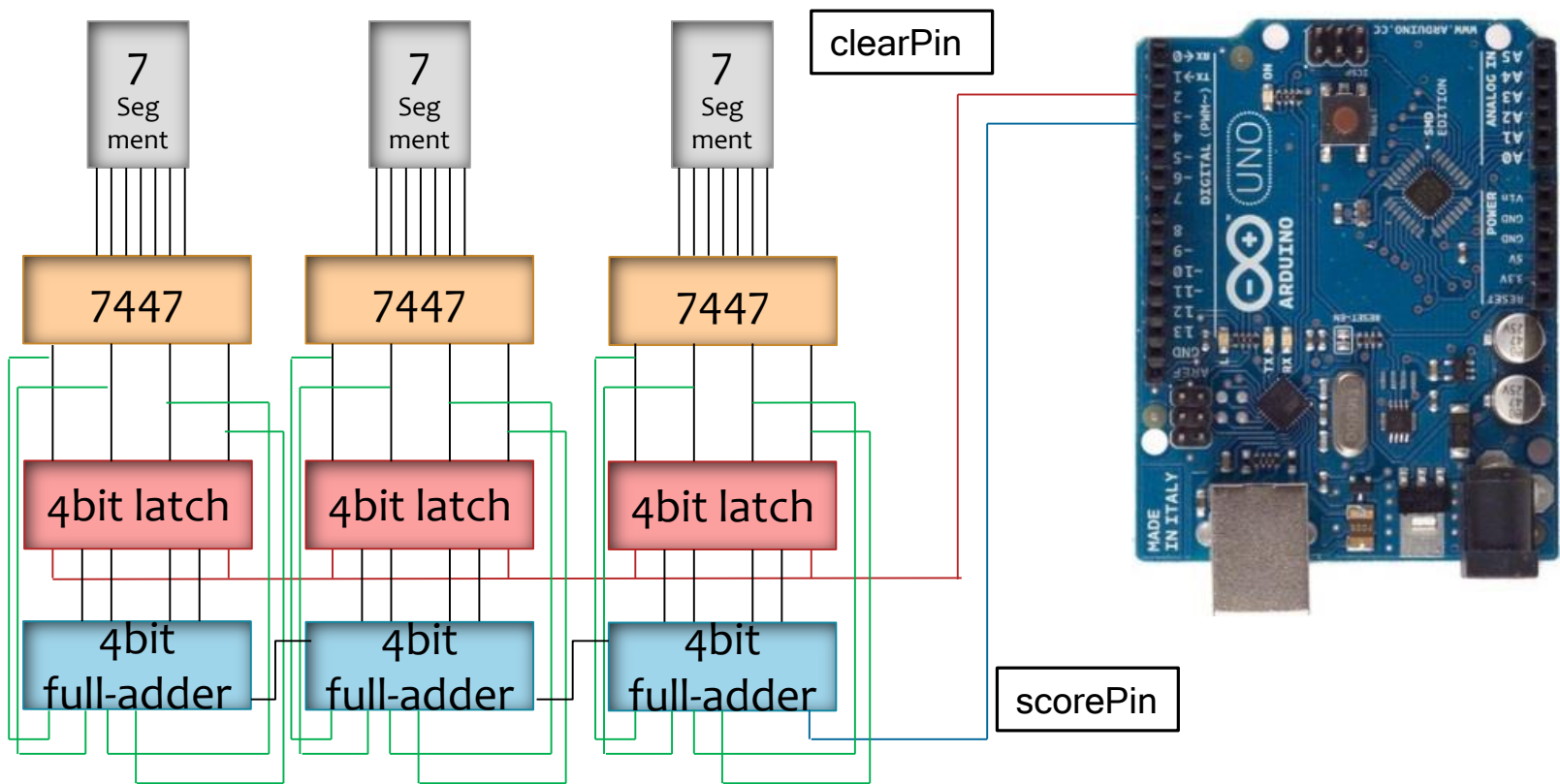
- ▶ 7-segment를 초기화할 핀 선언
- ▶ 줄 삭제 함수에 삭제된 핀의 개수를 증가하는 함수 선언

디지털 회로

- ▶ BCD to 7-segment 디지털 칩 7447 이용
- ▶ 4bit 가산기를 이용한 줄 수 계산
- ▶ 4bit latch를 이용한 결과값 저장
- ▶ 저장된 결과값을 7-segment로 전달



줄 수 삭제 블럭도





제한 시간 표시 원리



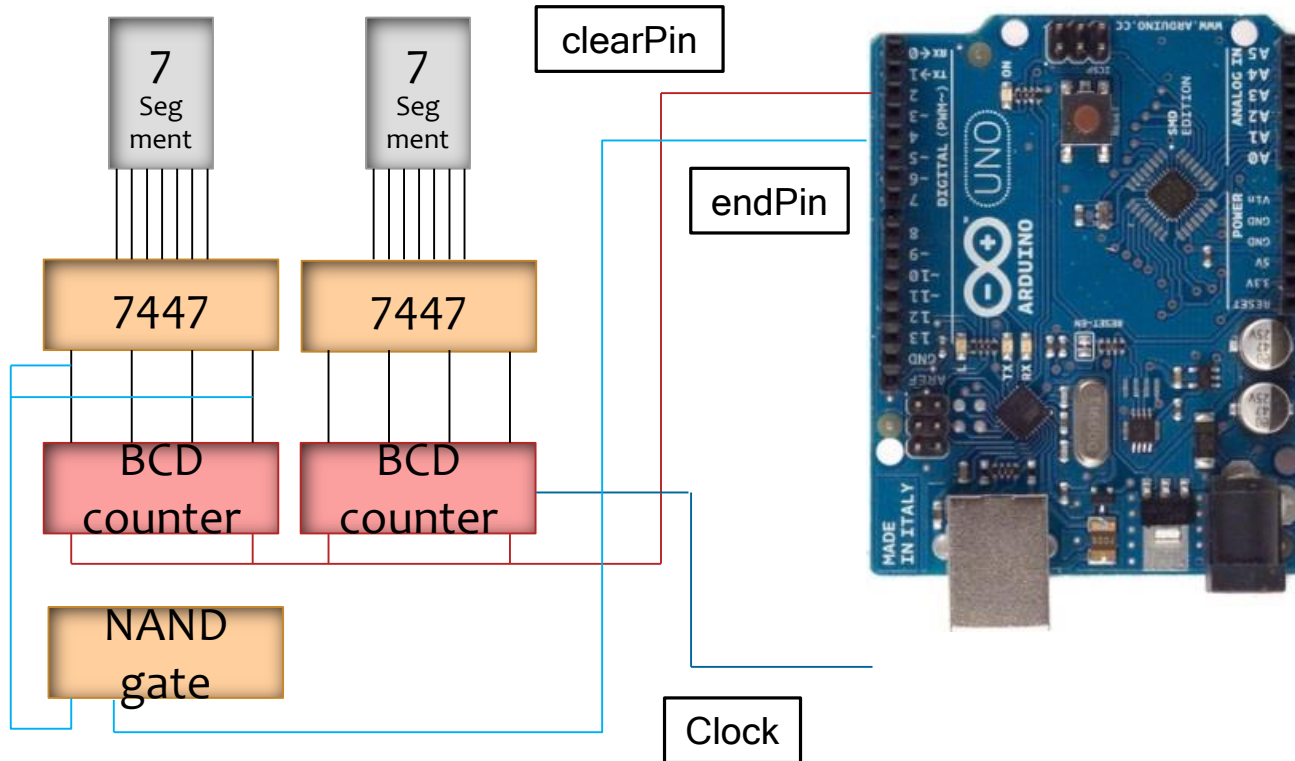
Arduino

- ▶ 7-segment를 초기화할 핀 선언
- ▶ 특정 핀에서 값이 입력되면 정지

디지털 회로

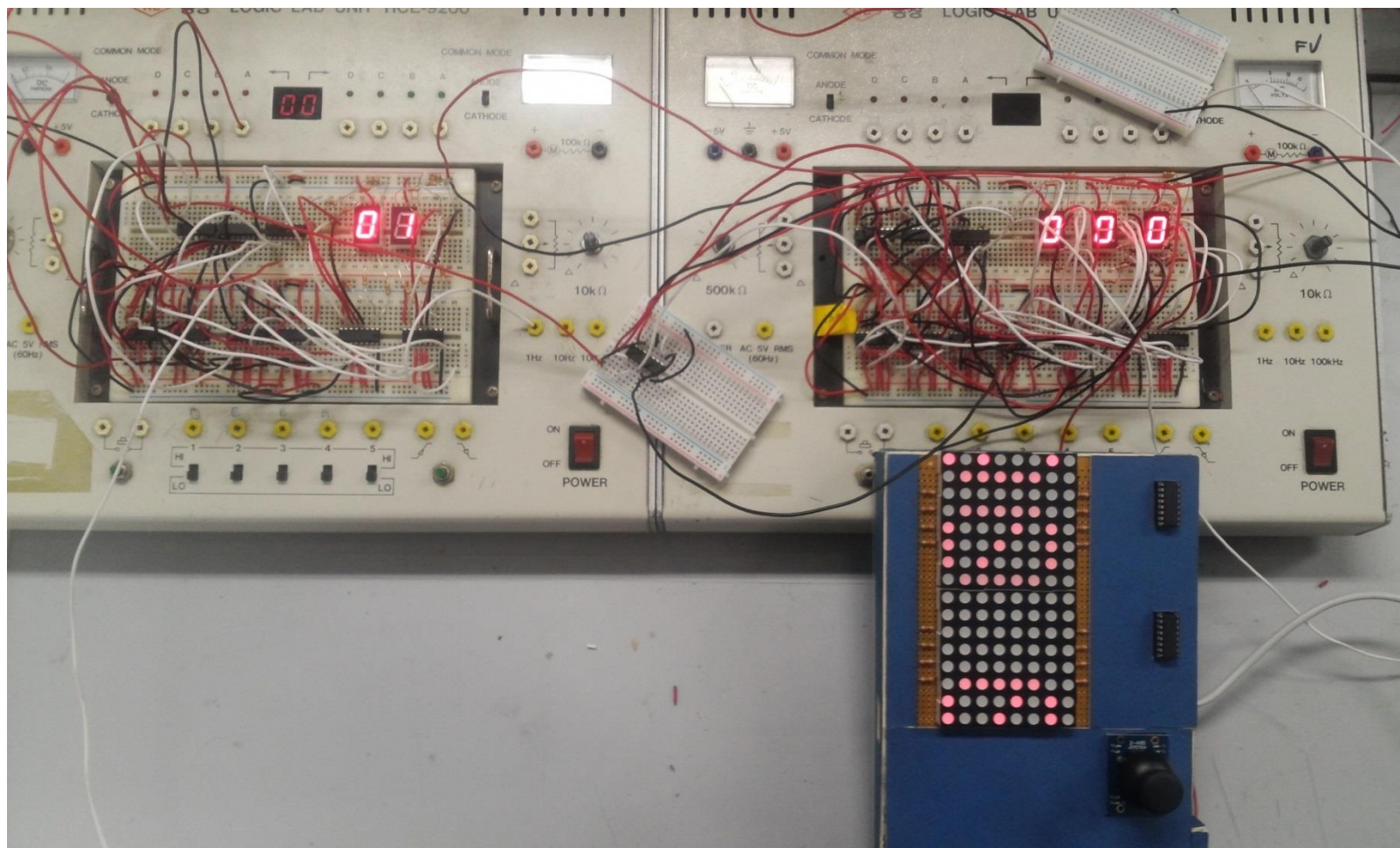
- ▶ 99 업카운터를 이용
- ▶ 90초에 도달하면 프로그램 정지
- ▶ Buffer를 의도적으로 발생하여 7-segment 출력 오류 해결

시간 제한 블록도



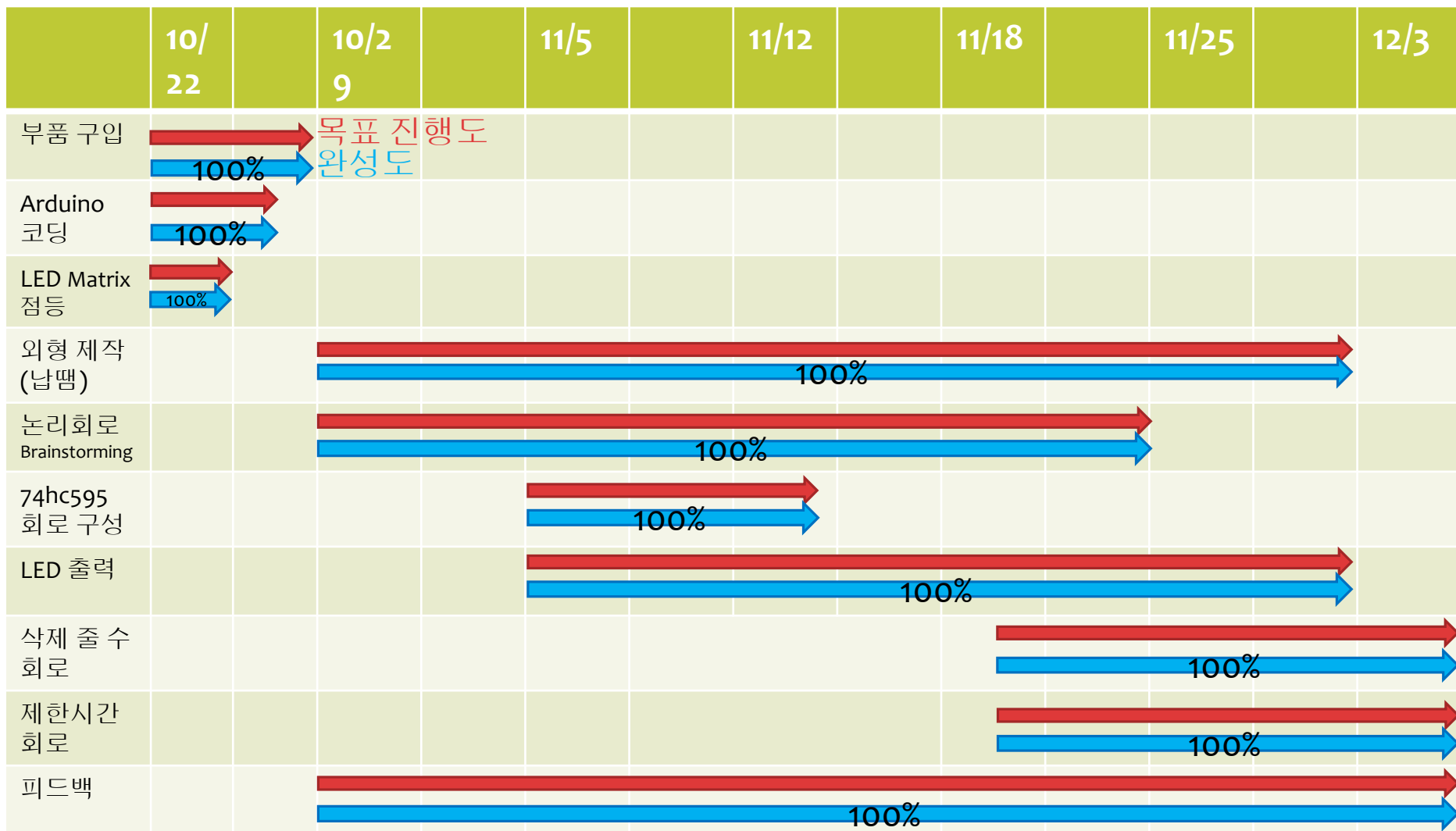


최종 제작 결과





일정표





사용 부품 List



부품	개수
브레드 보드	2 + 2(small)EA
8x8 LED Matrix (BVD-8518SG1)	2EA
Analog 2 axis joystick	1EA
74HC595	2EA
BCD 카운터(74HC76)	2*2EA
BCD to 7-segment IC(7447)	5EA
Resistor(195Ω)	7*5EA
4bit binary full-adder(7483)	3EA
4bit latch(74379)	3EA
7-segment(FND507)	5EA



Thank you for Listening!

