

LED CUBE

이름 오상민
이메일 tkdals9630@naver.com

목차

- I. 서론
 - 1. 선정 및 개발 목적
 - 2. LED 디스플레이의 장점과 기술동향
 - 2.1 기술동향
 - 2.2 LED 디스플레이의 장점
- II. 본론
 - 1. LED
 - 1.1 LED정의
 - 1.2 LED구조
 - 2. DECODER
 - 3. FLIP-FLOP
 - 3.1 LATCH
 - 3.2 D GATE LATCH
 - 3.3 D FLIP-FLOP
 - 4. TRANSISTOR
 - 4.1 TRANSISTOR의 정의
 - 4.2 TRANSISTOR 동작원리
 - 5. 시스템 구성도
 - 6. 하드웨어 설계
 - 6.1 Main board 회로도
 - 6.2 MCU 회로도
 - 6.3 IC 회로도
 - 7. 소프트웨어 설계
 - 7.1 주요 파일
 - 7.2 주요 코드
 - 8. 결과
- III. 결론
- 참고문헌

I. 서론

1. 선정 및 개발 목적

디스플레이는 우리 일상 많은 곳에 다양한 형태로 있다. 그 중 LED는 환경친화적이고 쉽게 제어 가능한 부품 중 하나이다. 그러한 LED디스플레이로 배우면서 제작 가능한 것을 찾아 보다가 우연히 LED큐브가 눈에 들어왔고 제작을 시도 하게 되었다. LED큐브의 장점으로서는 단순한 디스플레이가 아닌 3D 디스플레이가 가능하다는 점이고 뿐만 아니라 평면으로도 다양한 것들을 표현할 수 있다는 것이다.

2. 기술동향 과 LED 디스플레이의 장점

2.1 기술 동향

1993년 Nichia의 Nakamura에 의하여 고 휘도 청색 LED 제조 기술 이 개발되면서 LED로 백색광을 만들 수 있게 됨에 따라 조명 분야 응용이 가능하게 되었다. LED는 저전력 소모, 긴 수명의 장점을 가지고 있으나, 아직 백열등이나 형광등에 비하여 가격이 높아서 광범위한 사용은 이루어지지 않고 있다. 그러나 Si 결정을 기판으로 사용 하는 기술의 개발 등을 통하여 LED 가격이 지속적으로 하락하는 추세를 보이고 있고, 정부의 에너지 절감 및 이산화탄소 배출 감축 정책에 따라 백열등의 생산과 판매를 2014년부터 전면적으로 금지할 계획으로 있어서 조만간 LED 조명의 사용이 크게 증가할 것으로 전망된다.

2.2 LED 디스플레이의 장점과 단점

- LED 디스플레이의 장점

장점	내용	비고
적은 전력 소모량	필라멘트로 인한 열 손실이 없어 소비전력 대비 밝기가 우수하다.	
충격에 강하고 안전하다	기존 광원에 비해 깨지거나 다칠 확률이 적고, 저 전압 사용으로 비교적 안전하다.	
반영구적 수명	약 5만 시간의 수명으로 반영구적 수명을 가지고 있다	
소형화	광원이 작아 소형화, 경량화가 가능하다	
광범위한 확장성	집단 제어가 가능해 광원을 다양한 모양으로 구현 가능하다	
자유로운 색 조절	파장을 자유롭게 조절하여 여러가지 색 표현이 가능하다	
환경 친화적	무 수은 광원으로 환경 친화적이며, CO2 발생량을 크게 줄일 수 있다.	

- LED 디스플레이의 단점

단점	내용	비고
낮은 가격 경쟁력	휘도 당 LED가격이 높고, 드라이버 방열, 휘도 보정에 가격이 상승한다.	
연색성	광학적으로 선명한 단색 광을 보여줌으로써 연색성이 좋지 않다.	
조명 전환	점 광원 형태로 대면 조명 전환이 어렵다	
열 특성	온도가 상승함에 따라 휘도 감소 및 색상 변화가 일어난다.	

II. 본론

1. LED

1.1 LED의 정의

LED(Light-Emitting Diode)는 반도체의 일종으로 발광다이오드라고 한다.

일반 다이오드는 반도체의 p-n접합에 바탕을 두고 있다. p-n 다이오드에서 전류는 p-type(양극-anode)면에서 n-type(음극-cathode)면으로만 흐르는 것으로 전류의 흐름에 대한 방향성이 일정하다.

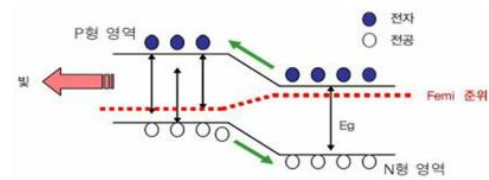


그림1-1 다이오드 발광의 원리

LED 발광다이오드 또한 그림1-1처럼 p-n접합부의 구조로 이루어져 있으며, p-type 반도체에서는 +전하(양극-anode)를 갖는 정공과 n-type 반도체에서는 -전하(음극-cathode)를 갖는 전자가 서로 충돌하며 빛이라는 에너지를 발현하게 된다.

1.2 LED의 구조

LED램프(그림1-2)를 구성하는 LED chip은 그림1-2 과 같이 여러 층의 반도체 물질로 구성되어 있다. LED에 직류전압을 공급하면 active layer층 에서 빛을 내며, 그 빛은 직접 방출되거나 반사에 의해 방출되고 연속 스펙트럼을 가지는 광원과는 다르게 특정한 파장으로 광색을 표출한다.

LED는 적색, 청색, 백색 LED로 구분된다. 청색에서 적색에 이르는 모든 파장과 백색을 생산하는 LED는 높은 휘도를 얻기 위해서 두 가지 화합물 (AlGaInP, InGaN)이 널리 사용된다.

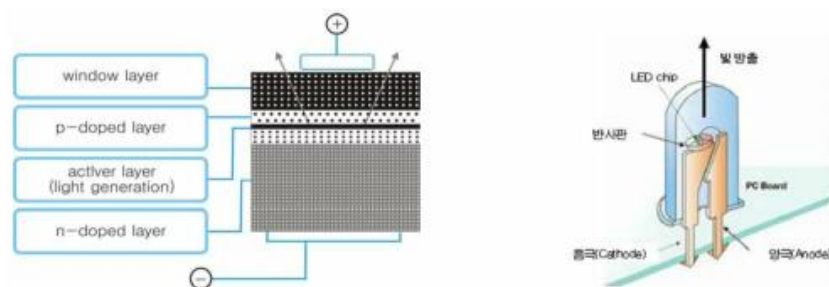


그림1-2 LED의구조

2. DECODER

Decoder란 n 비트의 2진 코드(code) 값을 입력으로 받아들이며 최대 2^n 개의 서로 다른 정보로 바뀌 주는 조합 회로를 말한다. 입력 값에 따라 선택된 하나의 출력선이 나머지 출력선들과 반대 값을 갖는다. 그림 2-1에 입력선이 2개, 출력선이 $2^2 = 4$ 개인 2-to-4 decoder를 나타내었다.

그림에서 예를 들어 AB 입력 값이 01일 경우에는 출력선 D1만이 1이고 나머지 출력선 D0, D2, D3은 모두 0이 되며, 나머지 입력값의 조합에 대해서도 한 출력선이 나머지 출력선들과 다른 값을 가짐을 확인할 수 있다.

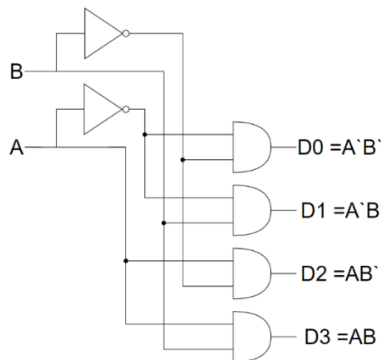


그림2-1 AND 게이트 논리회로

입력		출력			
A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

표2-1 디코더 AND 게이트 진리표

2-to-4 decoder의 또 다른 형태로 그림 4-2와 같

은 decoder가 있다. 그림 2-1의 decoder에 대한 진리표와 그림 2-2의 decoder에 대한 진리표를 비교해 볼 때 출력 값이 0은 1로, 1은 0으로 바뀌었음을 알 수 있는데, 이 decoder 역시 입력 값의 각 조합에 대해 한 출력선이 나머지 출력선들과 반대값을 가짐을 확인할 수 있다. 또한 회로를 비교해 보면 그림 2-1의 AND 게이트들이 그림 2-2에서는 NAND 게이트로 바뀌었음을 알 수 있다. 실제로 회로 구성에 사용되는 decoder는 주로 그림 2-2와 같은 형태가 많다. 그 이유는 AND 게이트를 칩 내부에 구현할 때는 NAND 게이트 뒤에 NOT 게이트를 연결한 형태로 구현되며, 따라서 그림 2-2의 NAND 게이트를 사용한 decoder의 게이트가 더 적고 경제적이며 성능 측면에서도 NOT 게이트의 신호전달 지연시간을 줄일 수 있기 때문이다.

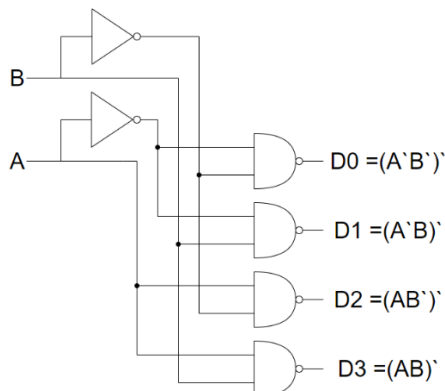


그림2-2 NAND 게이트 논리회로

입력		출력			
A	B	D0	D1	D2	D3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

표2-2 디코더 NAND 게이트 진리표

3. FLIP-FLOP

3.1 LATCH

Latch 는 한 비트의 정보를 데이터가 바뀌기 전까지 계속 유지하는 회로이다. 따라서 출력 Q 을 0 또는 1 로 상태 전이가 필요하다. latch 종류에 따라 하나 또는 두개의 입력을 사용한다.

논리 회로 시스템 설계에서 경우에 따라 latch 의 입력을 반영할 시점을 조절할 필요가 있다. 즉, 입력 신호가 들어와도 입력 시기를 조절하여 Q 의 상태변화가 없도록 하는 제어 신호가 있고, 이것으로 입력을 무시하거나 또는 출력에 반영할 수 있다. 이때 사용하는 신호가 게이트(gate, 또는 enable)이다. 표시는 E(enable, E, EN, G 등으로 표시)로 하고, 입력 신호와 별도의 제어 입력이 같이 있다. 이 신호가 존재하는 latch 를 Gated latch 이라고 한다. E 신호가 없을 경우는 입력의 상태가 바로 반영된다.

3.2 D GATE LATCH

D latch 는 SR 의 상태 전이를 유도하는 SR 입력이 01 또는 10 만이 존재한다. SR latch 의 입력 SR=11 과 SR latch 의 입력 SR=00 은 Q 와 Q' 출력이 같은 상태가 된다. 기본적으로 Flip-Flop 에서 Q 와 Q' 출력이 상반된 상태가 나와야 한다. 따라서 Q 와 Q' 출력이 같은 논리는 피해야 한다. D latch 는 입력이 하나이므로 SR latch 의 금지된 상태가 되지 않도록 S 와 R 이 항상 반대의 Logic 이 되도록 D 입력의 NOT 게이트를 사용 하면 된다.

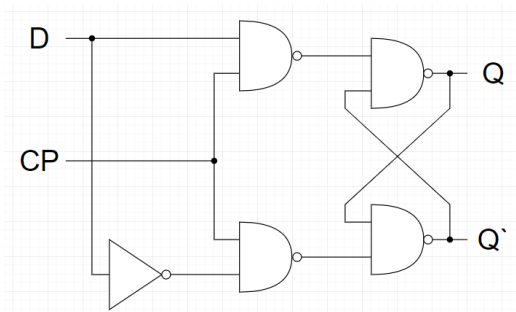


그림3-1 D latch 논리회로

CP	D	Q(next)
Rising edge	0	0
Rising edge	1	1
None	X	Q(previous)

표 3-1 D latch 진리표

3.3 D FLIP-FLOP

D Flip-Flop 은 입력 D 의 값을 클럭의 Edge 에서 캡처해서 Q 에 반영한다. Edge 가 발생하지 않는 시간에는 Q 가 변하지 않고 유지한다.

D Flip-Flop 의 동기를 위한 Clock 은 Rising edge 와 Falling edge 를 사용한다. D Flip-Flop 의 출력 상태 변화는 클럭의 Edge 에서만 이루어진다. 즉, S=R=0 일 경우, 입력 D 와 CP 에 의해 상태변화가 되고, S 또는 R 이 하나만 1 일 때 비동기로 즉각 Q 의 상태변화가 일어난다. SR=11 이면 출력 QQ=11 이므로 사용하지 않는다.

4. Transistor

4.1 Transistor의 정의

Transistor는 현대 전자기기를 구성하는 굉장히 흔한 기본 부품 중 하나이며 Transistor출현으로 더 작고 값싼 전자기기가 개발 되었다. Transistor는 크게 접합형 transistor와 전계효과 transistor로 구분된다 Transistor는 보통 입력단, 공통단, 그리고 출력단으로 구성 되어 있다. 입력단과 공통단 사이에 전압 또는 전류를 인가 하면 공통단과 출력단 사이의 전기전도도가 증가하게 되고 이를 통해 그들 사이의 전류 흐름을 제어하게 된다.

4.2 Transistor 동작원리

대표적인 Transistor의 동작원리는 증폭 작용과 스위칭 작용이 있다.

- 증폭 작용

베이스 전류 입력 - 콜렉터/이미터 전류 출력: $i_c = \beta i_b$

일정 배율을 가지고 동작하는 것을 forward-active (또는 간단히 active) 모드 라고 한다. 이 모드는 입력 전류에 대해 일정 비율로 출력 전류로 나타난다. 그림4-1 Transistor active모드에서 입력 전류의 몇 배가 출력 전류로 나타난다. 따라서 다음과 기본 회로도 같은 수식으로 표현이 가능하다.

$$I_C = \beta_F I_B$$

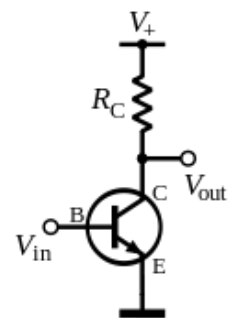
입력 전류의 β_F 만큼 전류 증폭이 된다. 만약 출력 전압이 필요하다면 저항 등을 사용하여 출력 전류를 흘려 주면 옴의 법칙에 따라 전압으로 변환 된다. 그림4-1 의 증폭 예에서 입력 전류 I_B 는 입력 전압 V_{in} 에 의 결정된다. 그리고 콜렉터에 흐르는 전류 I_C 는 저항 R_C 에 의해 V_{out} 의 출력 전압으로 변환 된다.

- 스위칭 작용

스위치 작용은 동작 모드 중에서 cut-off와 saturation 모드에서 동작 한다. 논리 상태가 0에서 1로 또는 1에서 0으로 변할 때 순간적으로 active 모드가 나타날 수는 있지만 순간적으로 짧은 시간 동안이다. 0초 동안 순간적으로 상태가 변할 수 없기 때문이다. 이것은 이상적일 때나 가능한 일이지, 관성이 존재해 실제 현상에서는 불가능 하다.

Transistor의 스위치 동작 시, 다음과 같은 상황을 논리회로에서 이용한다:

- 스위치 켜짐 (ON): 포화모드(saturation mode) 상태로 출력 전류는 외부회로에 의해 제한
- 스위치 꺼짐 (OFF): 차단모드(cut-off mode) 상태로 출력 전류 거의 0



5. 시스템 구성도

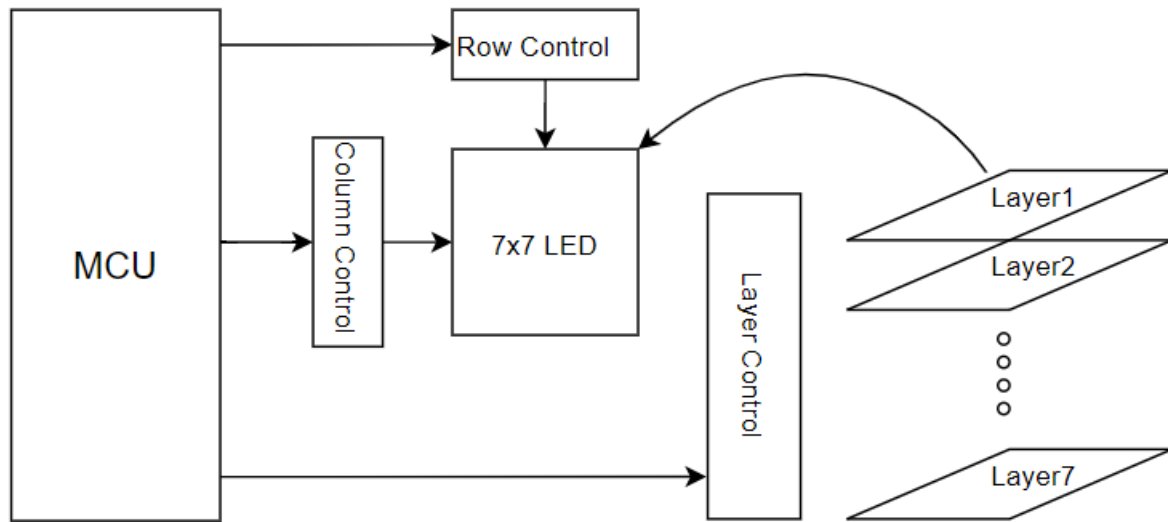


그림5-1 시스템 블록도

- MCU: Atmega328p
- Row control: 74hc574(Flip-Flop)
- Column Control: 2N3904(Transistor)
- Layer control: 74hc138(Decoder)
- LED: 5mm 2.2V 80mW

6. 하드웨어 설계

6.1 Main board 회로도

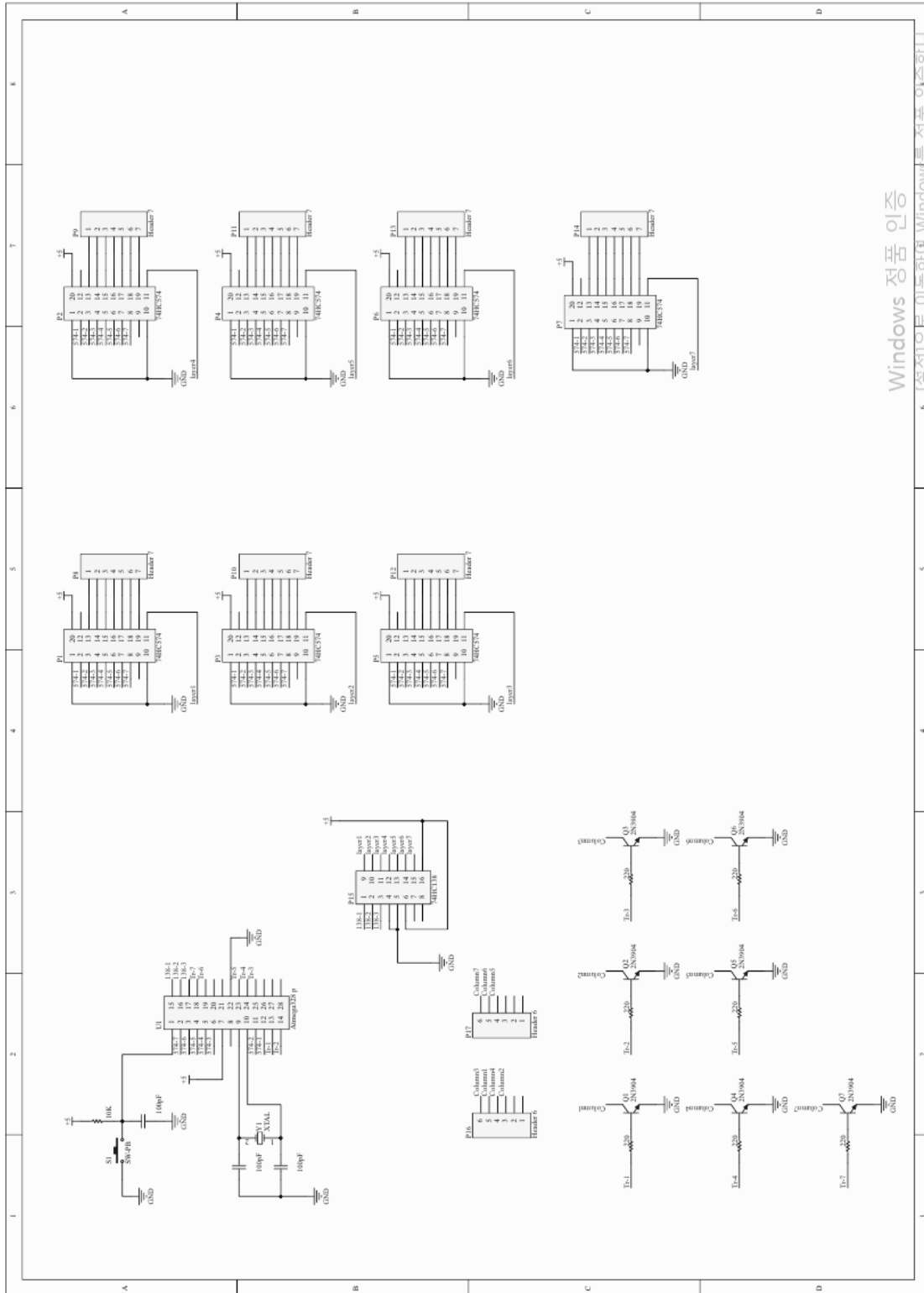


그림6-1 Main Board 회로도

6.2 MCU 회로도

- ATmega 328p
 - High Performance, Low Power AVR® 8-Bit Microcontroller
 - 23 Programmable I/O Lines
 - Operating Voltage: 1.8 - 5.5V
 - Memory Size: Flash- 32K Bytes, EEPROM- 1K Bytes, RAM- 2K Bytes, Interrupt Vector Size- 2 instructions words/vector

- 회로도

- Pull-up reset switch: 1번 핀의 리셋 스위치
- Clock crystal: 9,10번 핀에 16MHz crystal과 캐패시터

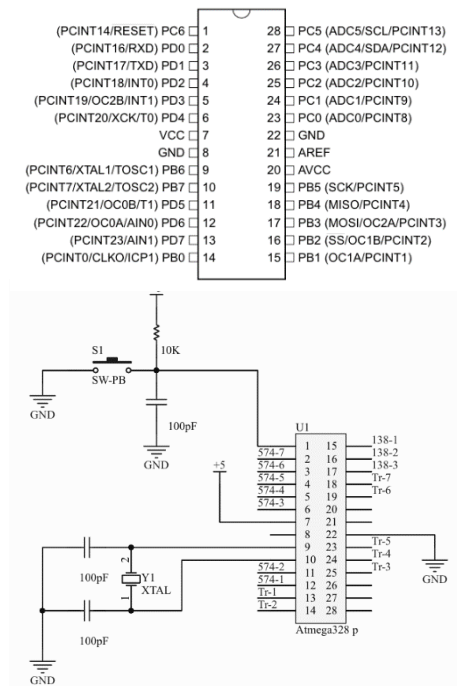


그림6-2 ATmega 328p 상세 회로도

6.3 IC 회로도

- 2N3904
 - NPN switching transistor
 - Collector-emitter voltage: max. 40V
 - Collector current: max. 200mA
- 2N3904 회로도
 - Emitter-base: ATmega328p data pin.
 - Collector: 각 레이어 LED의 그라운드 선을 collector에 연결한다.

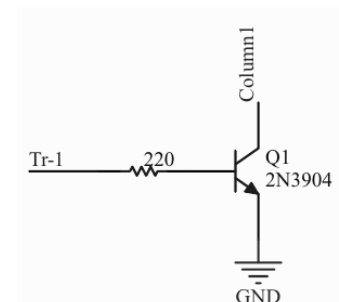


그림6-3 Transistor 연결 회로도

- 74hc 574

- Operating Voltage Range: 2.0 to 6.0 V
- Low Input Current: 1.0 mA
- Outputs Directly Interface to CMOS, NMOS, and TTL
- Clock rising edge에 데이터를 내보냄
- Output은 Enable핀에 영향을 받지 않지만 Enable핀이 HIGH일 때 high-impedance 상태가 됨

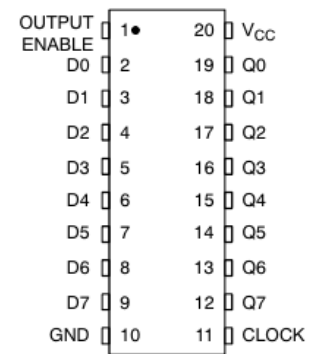
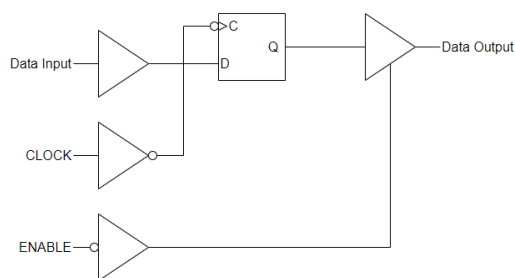


그림6-4 74hc574 핀 정보

- 74hc 574 Function Table



FUNCTION TABLE

Inputs			Output
OE	Clock	D	Q
L		H	H
L		L	L
L	L,H,	X	No Change
H	X	X	Z

X = Don't Care
Z = High Impedance

그림6-5 74hc 574 논리회로 및 Function Table

- 74hc 574 회로도

- 각 74hc574의 데이터 핀에 ATmega328p의 데이터 핀을 연결한다.
- 해더 핀에 각층의 LED열을 연결해 준다.

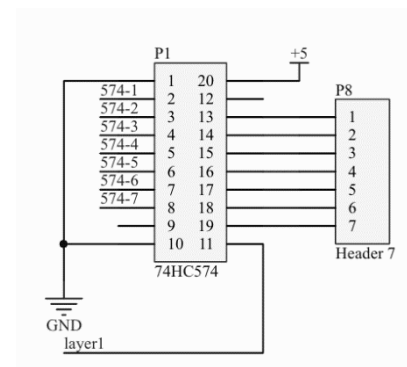


그림6-6 74hc574 연결 회로도

- 74HC138

- Ideal for memory chip select decoding(3to8)
- Three enable inputs: two active LOW(E1 and E2) and one active HIGH (E3)Outputs Directly Interface to CMOS, NMOS, and TTL
- Data input voltage: $V_I = \text{GND to VCC}$

- 74HC138 Function Table

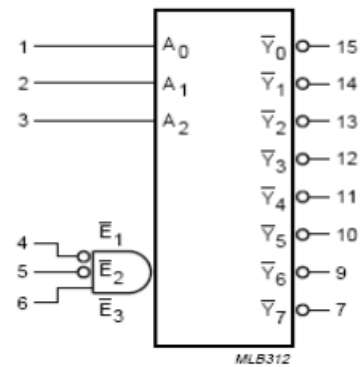


그림6-7 74hc138 핀정보

INPUTS						OUTPUT							
E1	E2	E3	A0	A1	A2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Notes: H = HIGH voltage level L = LOW voltage level X = don't care

표6-1 74hc138 진리표

- 74HC138 회로도

- 각 OUTPUT 핀이 74HC574의 Clock핀으로 들어가 layer를 조절한다.
- 74HC574의 Clock핀이 rising edge에 데이터가 출력되기 때문에 원하는 layer에서 다른 layer로 바뀔 때 데이터가 출력된다.

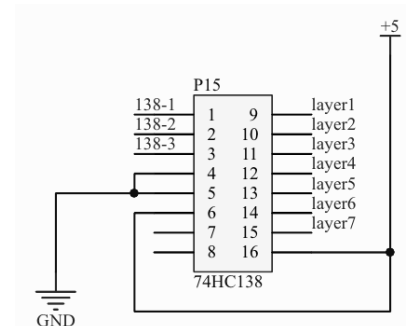


그림6-8 74hc138 연결 회로도

7. 소프트웨어 설계

7.1 주요 파일

	ledControl.h 헤더파일과 함수 정의	
main.c 실제 동작 부분	init.c Control함수	pattern.c 여러가지 패턴

그림7-1 주요 파일 관계도

7.2 주요 코드

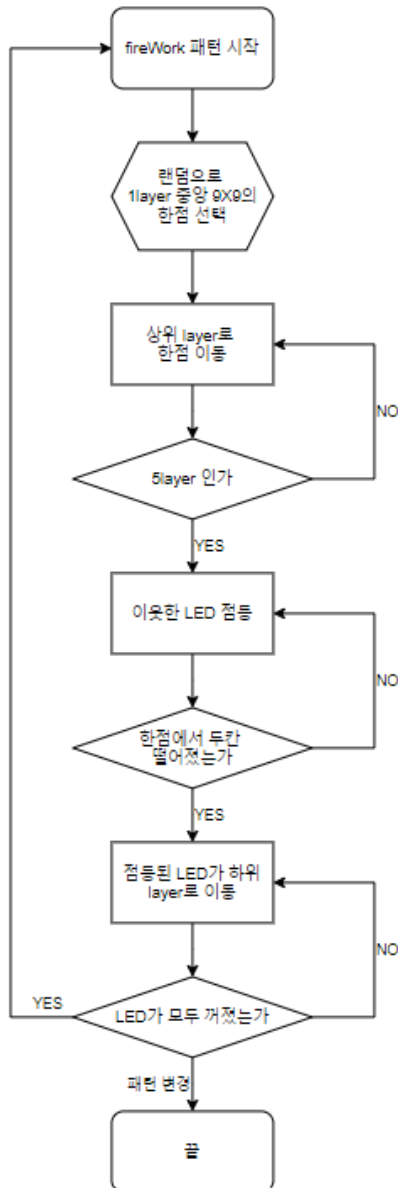


그림7-2 fireWork 패턴 흐름도

- init함수: 사용하는 레지스터를 정리하고 초기화하는 함수

```
void init(void){
    DDRC= 0x38;                //pc0,pc1,pc2 digitalout
    DDRD= 0xff;                // 1111 1111
    DDRB= 0x3f;                // 0011 1111

    PORTC= 0x04;                //PC1,PC2 PULL UP INPUT
    PCICR |= (1<<PCIE1);
    PCMSK1 |= (1<<PCINT10);
    sei();

    rowTotalControl(0x00);      //init
    PORTB= 0x3f;
    PORTD = (PORTD&0x7f)|0x80;
}
```

- rowTotalControl함수: 모든 layer에 입력 받은 data값을 보내준다.

```
void rowTotalControl(int data){
    PORTD = (PORTD&0x80)|(data&0x7f);
    for(int i=0;i<7;i++){
        PORTC = (PORTC&0xc7)|((layerA[i]&0x07)<<3);
    }
}
```

- layerTotalControl함수: 입력 받은 data배열 값에 따라 layer(74hc574)의 clock이 변할 때 data를 보내준다.

```
void layerTotalControl(int *data){
    for(int i=0;i<8;i++){
        PORTC = (PORTC&0xc7)|((layerA[i]&0x07)<<3);
        PORTD = (PORTD&0x80)|(data[i]&0x7f);
    }
}
```

- columnTotalControl함수: 입력 받은 data값으로 transistor를 제어한다.

```
void columeTotalControl(int data){
    PORTB = (PORTB&0xc0)|(data&0x3f);
    PORTD = (PORTD&0x7f)|((data&0x40)<<1);
}
```

- pointControl함수: 74hc138의 input인 An값중 쓰지 않는 값을 이용해서 한 층을 제어한다.

```
void pointControl(unsigned char layer, unsigned char row,unsigned char
colume,unsigned int temp_t1){
    PORTB = (PORTB&0xc0)|(colume&0x3f);        // select tr 1bit~6bit
    PORTD = (PORTD&0x7f)|((colume&0x40)<<1);    // select tr 7it

    PORTC = (PORTC&0xc7)|((layer&0x07)<<3);      // select layer
    PORTD = (PORTD&0x80)|(row&0x7f);
    PORTC |= 0x38;
    delay_time(temp_t1);
}
```

8. 결과

사진 첨부

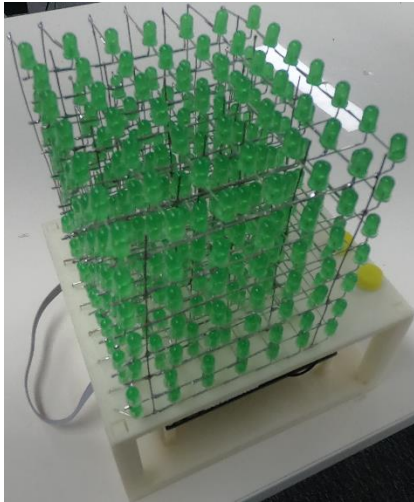


그림8-1 LED큐브 외형 사진 및 동작 사진

- 코드: <https://github.com/goyen/LEDCube.git>
- 시연 동영상: <https://youtu.be/7OKG3AEI6L0>

III. 결론

한정된 핀을 활용하여 많은 수의 LED를 제어하기 위해 74hc138 디코더를 이용했다. 각 면을 층, 행, 그리고 열로 나누어 제어를 했고, 연속적인 화면을 위해 행을 빠르게 반복하면서 구현을 했다. 즉, 7개의 핀으로 행을 바꿔 가면 제어한 것이다.

하드웨어를 제작하면서 디스플레이의 특성상 보여지는 것이 중요하다고 생각했고, LED간 간격을 일정하게 맞추기 위해 3D프린터를 이용하는 등 많은 노력을 했다.

층을 제어하는 D플립플롭은 이전 값을 저장하고 있기 때문에 층별로 제어를 하려면 이전 층의 데이터를 지워야 하는 문제가 있었고, 그래서 행을 바꿔 가면 제어하는 방식을 선택했다. 쉽게 여러가지 패턴을 만들기 위해 init.c라는 파일을 만들어 각 층, 행, 열을 제어 할 수 있는 함수를 만들어 정의했고, 가독성을 위해 만들어진 패턴들을 모아 하나의 파일을 만들었다.

참고문헌

- 발광다이오드 조명에 관한 연구: 이인호 저 서울 산업대학교 논문 2008
- LED 조명 구동회로: 정항근 저 청문각 2015
- LED 조명 시장 동향 분석 및 전망: 김정훈 저 파인테크닉스 기업 부설 연구소 논문
- Wikipedia - [Teansistor](#), [LED](#)
- Datasheet: 74hc138, 74hc574, ATmega328p