

제목 : 칼러센서를 통해 R,G,B 값을 추출한 여러가지 디스플레이(LED,LCD,OLED패널,LCD노트북)의 하드웨어, 소프트웨어적인 요소의 효용성 확인

이론적배경

ICT의 발달이 4차산업혁명이 되어 가속화되면서 디스플레이의 필요성은 강조되고 발전되고 있다. 삼성의 QLED, LG의 OLED, AMAZON의 KINDLE 등 다양한 디스플레이가 출현하였다. 이는 IOT와 5G와 맞물려 더욱 가속화되고 디스플레이의 필요성은 더욱 강조될 것이다. 하지만, 현재 20세 이하의 청소년들은 대부분 근시를 가지고 있으며, 이는 이전 세대와 다를 정도로 급격하게 늘어난 스마트폰, 전자기기의 이용 때문이라고 볼 수 있다. 그렇기에 다가올 디스플레이의 효용성을 알아보는 것이 필요해보인다.

실험 목적

- 여러가지 디스플레이(LED,LCD,OLED패널,LCD노트북)의 사용방식과 칼러센서의 사용방식을 익힌다.
- 디스플레이의 하드웨어적인 방식(색,모양,글자표현방식 등)이 R,G,B 값에 어떤 영향을 미치는지 분석한다.
- 디스플레이의 소프트웨어적인 방식(밝기,다크모드,블루라이트차단 등이 R,G,B 값에 어떤 영향을 미치는지 분석한다.
- 눈에 보이는 색과 R,G,B데이터 간의 연관성을 알아본다.

준비물

컬러센서 이용실험 (1~8) : 칼러센서, 아두이노 우노보드, 점퍼선

16x2캐릭터LCD 이용실험(1) : 16x2캐릭터LCD(컨트롤러 포함), 아두이노 우노보드, 점퍼선

8x8LED매트릭스이용실험(2,3) : 8x8LED매트릭스, 아두이노 우노보드, 점퍼선

기본 LED(파랑) 이용실험(3) : 기본 LED(파랑), 아두이노 우노보드, 점퍼선
OLED디스플레이 이용실험(4): OLED디스플레이, 아두이노 우노보드, 점퍼선
LCD노트북 이용실험(5~8) : LCD노트북

실험방법

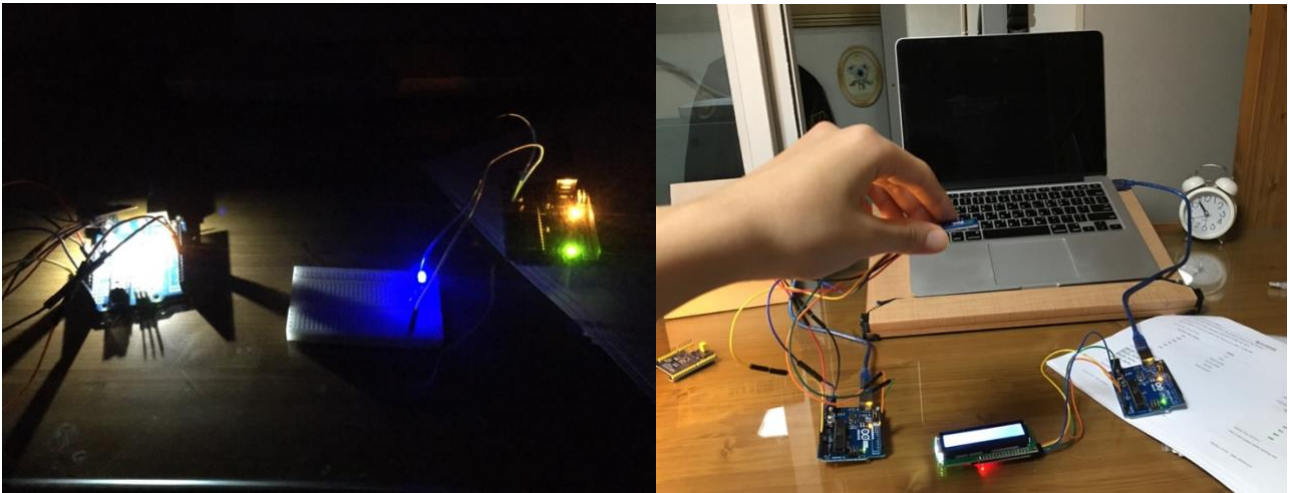
*실험1~4는 조건1,2하에 수행하고 실험5~8은 조건1,3,4하에 수행한다.

조건1. 암실

조건2. 디스플레이와 센서거리 25cm

조건3. 노트북과 센서거리 20cm

조건4. 디스플레이 밝기를 최소로 통일



**실험 1~8 중에 센서와 디스플레이모듈을 사용할시에 아래와 같은 방법을 따른다.

-점퍼선으로 아두이노와 컬러센서를 연결한다. 아두이노와 PC를 연결한다.

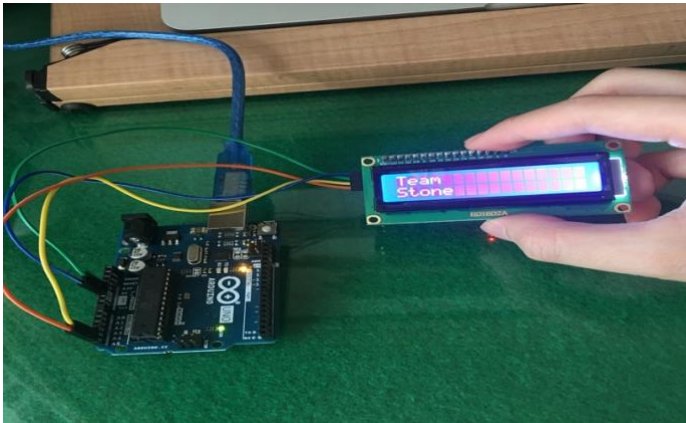
-아두이노 프로그램을 켜서 스케치파일을 업로드한다.(단,라이브러리의 추가가 필요)

-밝기를 변화할 시에 코드를 변환하여 준다.

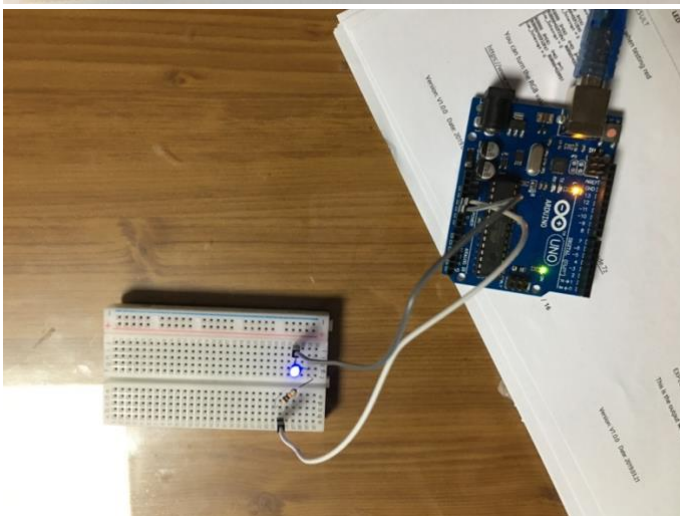
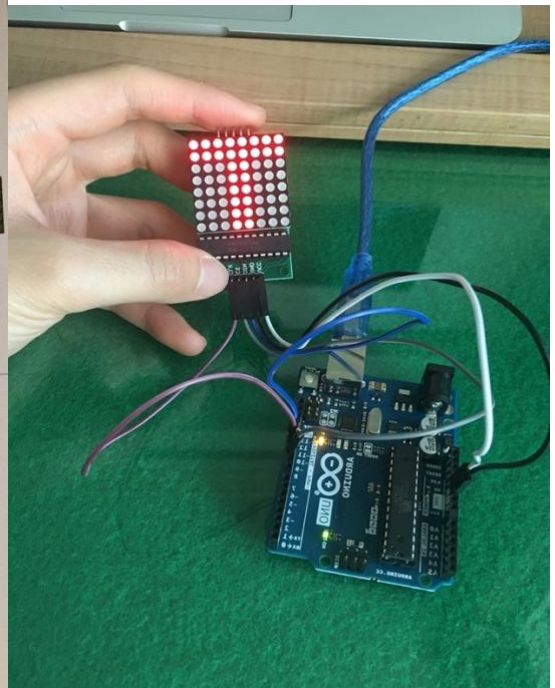
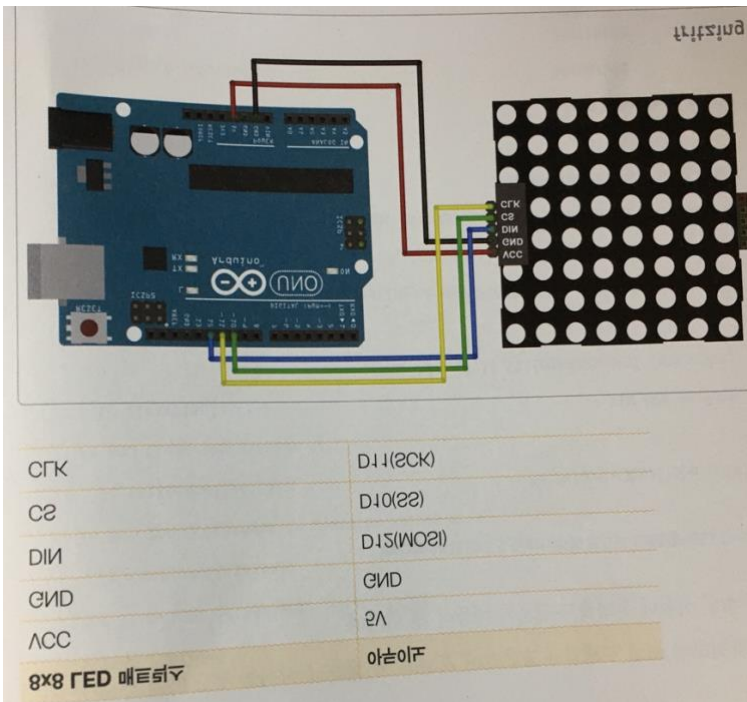
***각 실험의 구체적인 회로와 모습은 다음을 참고한다.(일부사진은 참고문헌(1)에서 발췌)

1. LCD

16x2 캐릭터 LCD	아두이노 우노 보드
VCC	5V
GND	GND
SDA	A4(SDA)
SCL	A5(SCL)

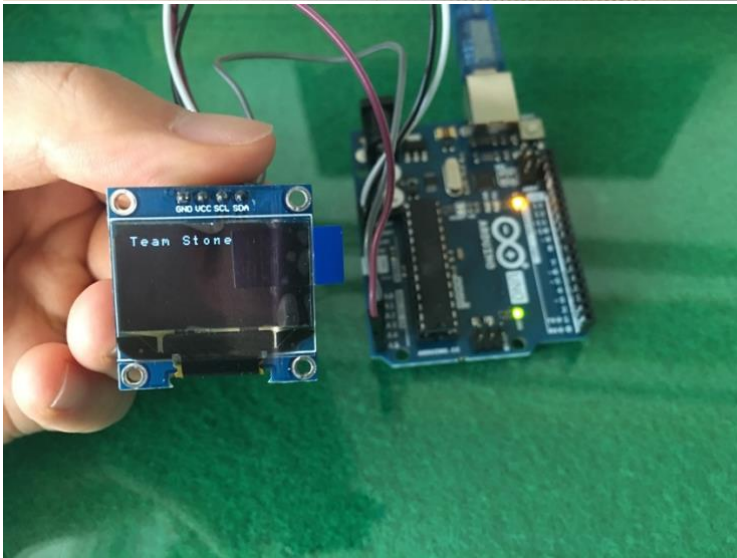
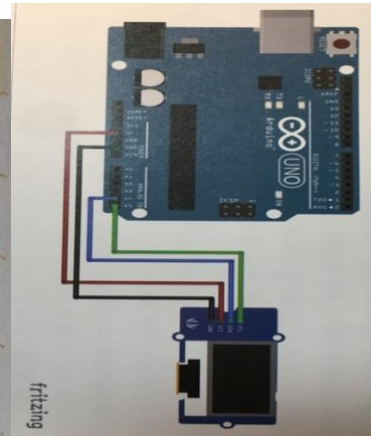


2. LED

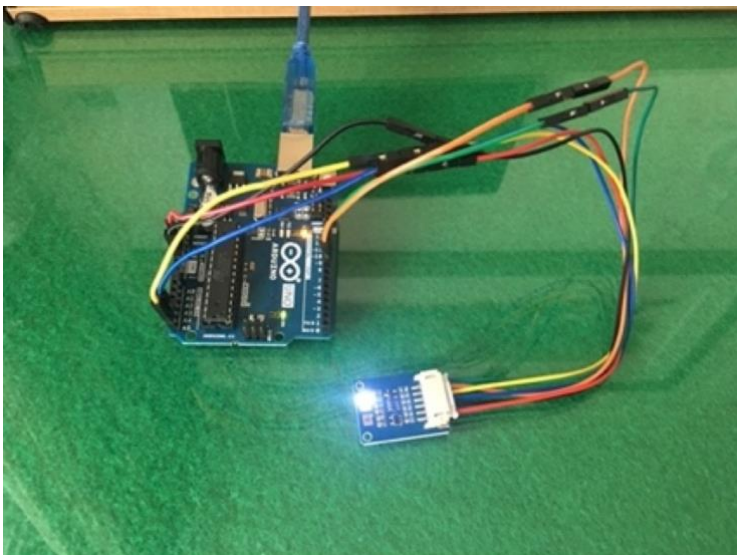


3. OLED

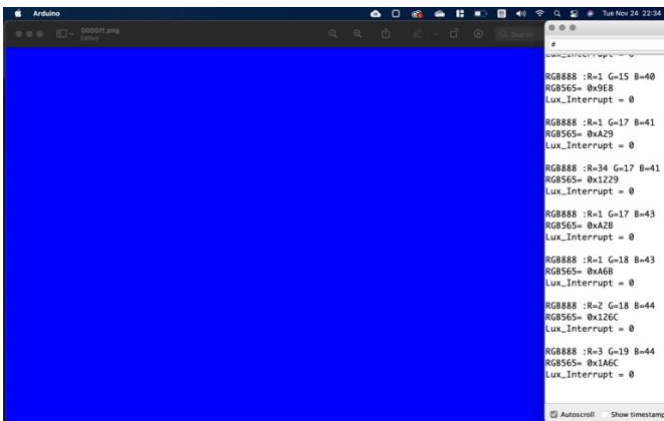
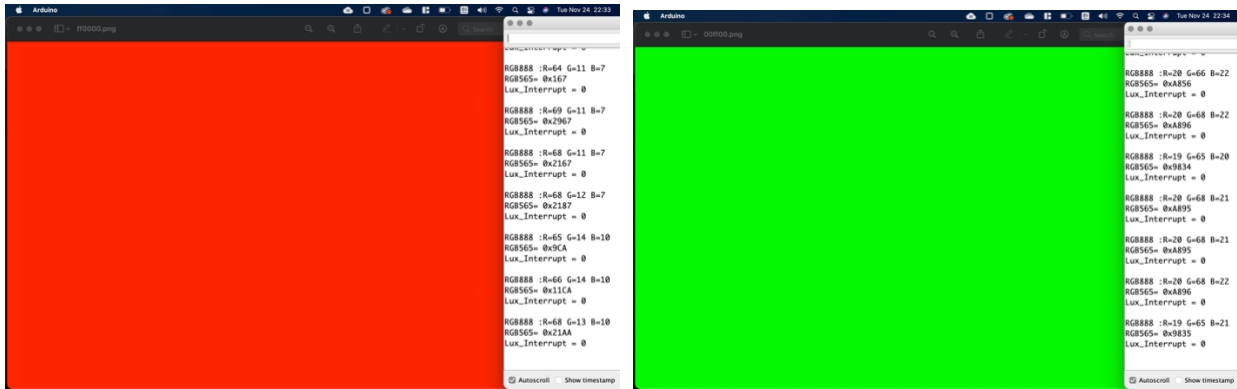
16x2 캐릭터 LCD	아두이노 우노 보드
VCC	5V
GND	GND
SDA	A4(SDA)
SCL	A5(SCL)



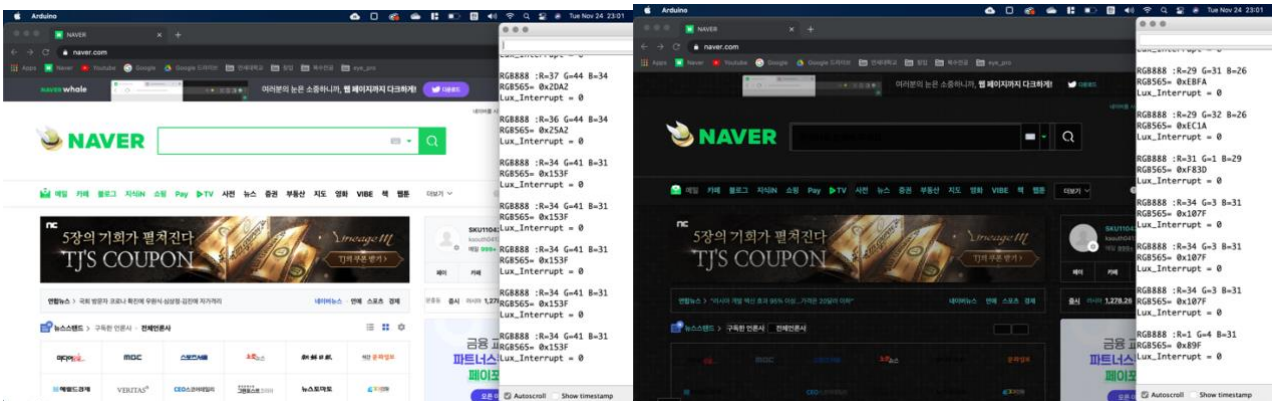
4. 칼러센서



5. R,G,B 원색



6. 다크모드 적용 전후



***코드는 다음을 참고한다.

1. LCD

```
#include <Wire.h> // i2C 통신을 위한 라이브러리
#include <LiquidCrystal_I2C.h> // LCD 1602 I2C용 라이브러리
LiquidCrystal_I2C lcd(0x27,16,2); // 접근주소: 0x3F or 0x27
void setup()
{
  lcd.init(); // LCD 초기화
  // Print a message to the LCD.
```

```

lcd.backlight();           // 백라이트 켜기
lcd.setCursor(0,0);        // 1번째, 1라인
lcd.print("Team");
lcd.setCursor(0,1);        // 1번째, 2라인
lcd.print("Stone");
}
void loop()
{
}

```

2. LED

```

#include "LedControl.h"           // 라이브러리는 첨부하였습니다.

LedControl lc=LedControl(12,11,10,1);    // 핀들을 접속 (DIN 핀,CLK 핀, CS핀, 주소) 여기서 주소는
                                          // 몇번째 모듈인지를 의미합니다.

```

```

void setup()

{

    lc.shutdown(0,false);    // 절전모드를 끄고,

    lc.setIntensity(0,15);    // 밝기를 설정합니다(0~15까지 밝기 설정가능합니다)

    lc.clearDisplay(0);      // 화면을 정리합니다

}

void loop()
{
    for (int col=0; col<8; col++)
    {
        lc.setLed(0,col,7,true);
    }
    for (int col=0; col<8; col++)

```

```

{
    lc.setLed(0,col,6,true);
}

for (int row=0; row<8; row++)
{
    lc.setLed(0,4,row,true);
}
for (int row=0; row<8; row++)
{
    lc.setLed(0,3,row,true);
}

}

```

//꺼지면서 올라가고 하는 동작이 반복됩니다

3. OLED

// 0.96인치 OLED Hello world 예제

```
#include <Wire.h>
```

```
#include <OzOLED.h>
```

```
void setup()
```

```
{
```

```
    OzOled.init(); //initialize Eduino OLED display
```

```
    OzOled.printString("Team Stone"); //Print the String
```

```
}
```

```
void loop(){
```

```
    OzOled.printString("", 0, 0); //Print the String
```

```
}// 0.96인치 OLED Hello world 예제
```

4. COLOR_SENSOR

```
#include "DEV_Config.h"
```

```

#include "TCS34725.h"
RGB rgb,RGB888;
UWORD   RGB565=0;

void setup() {
    Config_Init();
    if(TCS34725_Init() != 0){
        Serial.print("TCS34725 initialization error!!WrWn");
        return 0;
    }
    Serial.print("TCS34725 initialization success!!WrWn");

}

void loop() {
    rgb=TCS34725_Get_RGBData();
    RGB888=TCS34725_GetRGB888(rgb);
    RGB565=TCS34725_GetRGB565(rgb);
    Serial.print("RGB888 :R=");
    Serial.print(RGB888.R);
    Serial.print(" G=");
    Serial.print(RGB888.G);
    Serial.print(" B=");
    Serial.print(RGB888.B);
    Serial.print("WrWn");

    Serial.print("RGB565= 0x");
    Serial.println((RGB565), HEX);
    if(TCS34725_GetLux_Interrupt(0xff00, 0x00ff) == 1){
        Serial.print("Lux_Interrupt = 1WrWn");
    }else{
        Serial.print("Lux_Interrupt = 0WrWn");
    }
    Serial.print("WrWn");
}

```



```
DEV_Delay_ms(1000);
}
```

실험결과

실험1(밝기가 다른 두개의 LCD)

1[1]_밝음

	R	G	B
1	65	87	148
2	65	87	148
3	65	87	149
4	64	86	148
5	64	87	149
6	65	87	150
7	65	87	150
mean	64.7	86.9	148.9
std	0.5	0.4	0.9

1[2]_어두움

	R	G	B	
1	44	79	181	ignore
2	18	39	96	
3	19	40	98	
4	20	43	100	
5	21	44	103	
6	20	44	102	
7	20	44	102	
mean	19.7	42.3	100.2	
std	1	2.3	2.7	

실험2(밝기가 다른 두개의 LED_밝기는 0~15단계로 조절)

2[1]_15

	R	G	B
1	244	26	1 ignore
2	187	20	27
3	191	20	27
4	192	20	28
5	194	20	28
6	196	20	28
7	199	21	28
mean	193.2	20.2	27.7
std	4.2	0.4	0.5

2[2]_0

	R	G	B
1	238	21	18
2	236	21	19
3	226	20	17
4	231	21	17
5	234	21	18
6	236	22	18
7	238	22	18
mean	234.1	21.1	17.9
std	4.3	0.7	0.7

실험3(색상이 다른 두 LED)

3[1]_빨강

R G B

3[2]_파랑

R G B

1	29	41	128	ignore	1	65	87	148
2	32	43	127	ignore	2	65	87	148
3	20	77	211		3	65	87	149
4	20	77	209		4	64	86	148
5	21	78	213		5	64	87	149
6	22	80	217		6	65	87	150
7	22	79	216		7	65	87	150
mean	21	78.2	213.2		mean	64.7	86.9	148.9
std	1	1.3	3.3		std	0.5	0.4	0.9

실험4(OLED)

4[1]

	R	G	B	
1	34	30	18	ignore
2	31	28	17	ignore
3	7	2	20	
4	9	2	21	
5	10	4	23	
6	10	4	23	
7	9	3	21	
mean	9	3	21.6	
std	1.2	1	1.3	

실험5(LCD 노트북에 띄운 R, G, B 색상 그림과 화면을 꺼둔 노트북)

5[1]_R

5[2]_G

	R	G	B			R	G	B
1	64	11	7		1	20	66	22
2	69	11	7		2	20	68	22
3	68	11	7		3	19	65	20
4	68	12	7		4	20	68	21
5	65	14	10		5	20	68	21
6	66	14	10		6	20	68	22
7	68	13	10		7	19	65	21
mean	66.9	12.3	8.3		mean	19.7	66.9	21.3

std	1.9	1.4	1.6
5[3]_B			
	R	G	B
3	34	17	41
2	1	17	41
1	1	15	40
4	1	17	43
5	1	18	43
6	2	18	44
7	3	19	44
mean	1.5	17.3	42.5
std	0.8	1.4	1.6

std	0.5	1.5	0.8
5[4]_nolight			
	R	G	B
1	15	18	13
2	15	18	14
3	15	18	14
4	15	18	13
5	15	18	13
6	15	18	13
mean	15	18	13.3
std	0	0	0.5

실험6(LCD노트북의 밝기 변화)

6[1]_최대			
	R	G	B
1	176	211	206
2	176	210	206
3	175	210	206
4	176	210	206
5	174	209	205
6	176	210	206
7	179	213	208
mean	176	210.4	206.1
std	1.5	1.3	0.9

6[2]_중간			
	R	G	B
1	171	214	202
2	196	231	217
3	188	225	213
4	184	222	209
5	183	222	209
6	182	221	208
7	183	221	209
mean	183.9	222.3	209.6
std	7.5	5.1	4.6

6[3]_최소			
	R	G	B
1	32	9	3
2	34	11	4
3	34	11	5
4	34	11	5
5	34	11	5
6	34	11	5
7	31	6	34
mean	33.7	10.7	4.5
std	0.8	0.8	0.8

실험7('실험5조건에서 LCD노트북에서의 블루라이트 차단 비교

7[1]_R				7[2]_G			
	R	G	B		R	G	B
1	100	11	6	1	20	64	14
2	99	10	6	2	20	64	14
3	98	10	6	3	20	65	14
4	99	10	6	4	19	64	14
5	100	10	6	5	19	64	14
6	100	11	6	6	19	63	13
7	97	9	4	7	19	63	13
mean	99	10.1	5.7	mean	19.4	63.9	13.7
std	1.2	0.7	0.8	std	0.5	0.7	0.5

7[3]_B			
	R	G	B
1	5	13	23
2	5	13	23
3	4	12	21
4	3	9	19
5	2	7	18
6	2	7	18
7	2	7	17
mean	3.3	9.7	19.9
std	1.4	2.9	2.5

실험8(LCD노트북에서의 다크모드 on/off_네이버시작화면비교)

8[1]_on					8[2]_off				
	R	G	B			R	G	B	
1	29	31	26	ignore	1	37	44	34	
2	29	32	26	ignore	2	36	44	34	
7	1	4	31	ignore	3	34	41	31	

4	34	3	31	4	34	41	31
5	34	3	31	5	34	41	31
6	34	3	31	6	34	41	31
3	31	1	29	7	34	41	31
mean	33.3	2.5	30.5	mean	34.7	41.9	31.9
std	1.5	1	1	std	1.3	1.5	1.5

분석 및 결론

실험1

-1[1]의 R,G,B 평균값이 1[2]보다 높은 것으로 보아, 밝기가 높으며 낮은 것에 비해 R,G,B수치가 높게 나타난다.

-1[1]의 R,G,B 표준편차가 1[2]보다 큰 것으로 보아, 화면이 어두울 시에 빛의 광량 사이의 변동성이 크게 나타난다.

실험2

-실험 전 설정을 하면서 2[1]의 밝기를 2[2]의 밝기보다 높게 해두었다. 그러나 R,G,B수치는 모두 높지 않다. R(+40.9), G(+0.9), B(-9.8)으로 G수치가 비슷하긴 해도 더 낮으며 B수치가 올라간 것을 제외하면 R,G수치 모두 내려갔다. 그러나 변화량을 각각의 R,G,B 평균 수치로 나누면 이야기가 달라진다. 즉, $R(+40.9/193.2=+0.21)$, $G(+0.9/20.2=+0.04)$, $B(-9.8/27.7=-0.35)$ 으로 오히려 B의 변화가 크게 나타난다. 또한, R,G,B값간의 유사성을 서로의 차이인 표준편차로 알아보면 2[1]=79.8, 2[2]=101.2으로 밝기가 밝으면, R,G,B간의 차이가 적음을 알 수 있다. 결과적으로 우리가 보는 빛이 R,G,B의 절대적인 값이 아닌 상대적인 값이 중요하다고 결론을 내릴 수 있으며, 이는 우리가 보는 빛이 R,G,B 선형적인 결합이 아님을 간접적으로 드러낸다.

실험3

-2[1]과 3[1]을 비교해보면, 2[1]은 빨강빛으로 보이고 3[1]은 파랑빛으로 보이는 만큼 각각의 R, B수치가 높게 나타난다. 그러나 G값을 비교해보면 3[1]은 R값(21)에 비해 G값(78)이 크다. 이에 비해 2[1]은 G(20.2), B(27.2)로 비슷하다. 여기서 알 수 있는 것은 아무리 다른 수치의 비가 달라도 절대적인 한 수치의 값이 우리가 보는 색을 결정짓는다는 사실이다.

실험4

-다른 패널에 비해 밝기가 적음은 R,G,B값이 적게 나오므로써 설명된다. 이는 다른 패널(LCD)에 비해 글자를 표현하는 부분 이외에는 디스플레이의 밝기가 '0'이기 때문이다.

실험5

-우선 nolight인데도 R,G,B수치가 있는 이유는 color_sensor자체에 하얀색 LED가 존재하기 때문이다. 또한, nolight시 R,G,B수치가 13~18로 대략 비슷하기에 관찰할 빛의 R,G,B에 큰 영향을 미치지 않을 것으로 보인다.

-5[1~3]을 보면 아무리 순수한 R,G,B를 띄우더라도 다른 색요소가 존재하는 것을 볼 수 있다. 이는 앞서 설명한 color_sensor의 영향 때문인 것으로 설명된다. 즉, 5[1]은 순수한 빨간색이지만 G,B가 12.3,8.3으로 나타나는데 이가 color_snesor의 빛과 중첩되어 나타난다는 것이다. 다만 특이한 것은 5[3]의 B 수치가 42.5로 5[1,2]의 R,G 최대값인 66.9보다 적다는 것과 R수치가 1.5로 현저히 작다는 것이다.

실험6

-6[1]과 6[2]를 비교하면, 밝기가 더 적은 6[2]가 R,G,B수치가 모두 높은 아이러니한 결과를 보여준다. 그러나 6[2]의 변동성(7.5/5.1/4.6)이 6[1]과 6[2]의 대략 '1'정도 되는 값보다 크다는 것이 조금 이상하게 여겨진다. 물론, 평균이 아닌 각각의 값으로 보아도 거의 대부분의 R,G,B 값이 크다는 결과는 여전히 아이러니하다. 그렇기에 재실험의 필요성이 있다.

-그렇다고 하더라도 6[1]과 6[3]을 비교하면, 밝기와 R,G,B수치는 비례함을 알 수 있다. 그리고 밝기가 최대일 때는 상대적으로 G(210.4),B(206.1)값이 큰데에 비해 최소일 때는 R값(33.7)을 알 수 있다.

실험7

-5[1]과 7[1]을 비교하면 블루라이트 차단 기능을 이용했을시, R값은 66.9->99로 오히려 증가하였고 G값은 12.3->10.1으로 약간 감소하였고 B값은 8.3->5.7으로 마찬가지로 약간 감소하였다.

-5[2]와 7[2]을 비교하면 블루라이트 차단 기능을 이용했을시, R값은 19.7->19.4으로 거의 변화가 없고 G값은 66.9->63.9으로 약간 감소하였으며 B값은 21.3->13.7으로 제일 크게 감소하였다.

-5[3]과 7[3]을 비교하면 블루라이트 차단 기능을 이용했을시, R값은 1.5->3.3으로 오히려 증가하였고 G값은 66.9->9.7으로 감소하였고 B값은 42.5->19.9로 마찬가지로 제일 크게 감소하였다.

-결론적으로 블루라이트 차단 기능을 이용했을시, R값은 증가하는 편이며, G값은 감소하며, B값은 압도적으로 감소함을 알 수 있다. 그리고 G,B의 감소로 인해 오두워진 화면을 밝게 유지하기 위해 오히려 R수치가 올라갔다고 예측할 수도 있다.

실험8

-off->on했을 시, R(-1.4), G(-39.4), B(-1.4)으로 R,G,B값 모두가 감소하였음을 알 수 있다. 또한, G값이 압도적으로 줄어든 이유는 네이버 시작 화면이 녹색이기 때문이라고 예측할 수 있다.

*여러가지 디스플레이(LCD,LED,OLED패널,LCD노트북)의 R,G,B값을 칼러센서로 확인해보면서, 먼저 R,G,B의 상대적인 수치값이 영향을 미치는 경우와 절대적인 값이 영향을 미치는 경우가 있음을 살펴보았다. 이는 빛이 파동의 특성과 입자의 특성을 동시에 가짐을 간접적으로 드러낸다. 혹은 실제 빛과 이를 받아들이는 인간 눈의 한계를 보여주는 것일 수도 있다. 또한 비록 다른 조건에서 실험했지만 각각의 디스플레이는 각자의 특성에 따라 다르고 같은 종류의 디스플레이라도 개발자의 출력 방식에 따라 R,G,B값과

눈에 보이는 이미지가 확연히 달라질 수 있다는 것을 알 수 있다. 그리고 다크모드, 블루라이트차단효과는 확연하게 G,B값을 줄여줌을 알 수 있다.

토의

[실험의 환경]

- 밤에 불을 끄고 실험했으나 창문으로 빛이 들어왔기에 완전한 암실 환경에서 실험할 필요가 있다.
- 디스플레이와 센서간의 거리를 유지하려고 손으로 들고 있었기에 고정장치를 사용해 오차를 줄일 수 있을 것이다.
- 실험결과 중 'ignore'이라는 값이 있었다. 또한, 아두이노 스케치의 시리얼 모니터는 텍스트복사가 안됨으로 겪는 어려움 등이 있었다. 그래서 데이터 선별 프로그램을 이용할 필요성이 있어보인다.
- 차이가 큰 디스플레이로 실험을 했기에 디스플레이간 비교를 하여 결론을 내리기에는 한계가 있었다. 그렇기에 거의 비슷한 조건의 디스플레이를 이용할 필요성이 있어보인다.
- 칼라센서를 노트북에 연결해서 노트북 디스플레이를 통해 데이터를 확인하는 번거로움이 있었다. 이를 해결하기 위해 센서, 디스플레이, 배터리를 이용해 R,G,B 수치 탐지기를 만들 수도 있겠다.

[실험의 방향성 및 분석]

- R,G,B를 통해 인간이 보는 색을 표현하고 색온도를 나타내는 수식을 찾거나 직접 예상을 하여 수식의 각 요소들을 설명할 수 있을 것이다. Ex. $CT=A*(R^2+G^2+B^2)+B*((255-R)^2+(255-G)^2+(255-B)^2)+...$
- 직접 디스플레이를 구현하거나 프로그램을 이용해 눈에 편한 디스플레이를 제작할 수 있을 것이다.

참고문헌

- (1)아두이노 입문자에게 필요한 모든 것, 아두이노 101(한빛미디어/서영배)
- (2)메이커를 위한 아두이노의 모든것(프리렉/이준원)