



피지컬 컴퓨팅 프로젝트

〈최종 보고서〉

과목명	피지컬 컴퓨팅
담 당	최 인 경 교수님
제출일	2021 년 6 월 18 일
조	3 조
팀원	이수빈 2019050982 강은서 2019066671 차세진 2019033590 김은정 2019093418

HANYANG UNIVERSITY

〈 목 차 〉

1. 아이디어 배경 및 목적

1-1 아이디어 배경

1-2 아이디어 소개

1-3 아이디어 목적

2. 구현계획

2-1 필요 부품

2-2 회로 설계안

2-3 인터랙션 구조도

3. 구현 내용

3-1 코드 구현

3-2 하드웨어 구현 및 목업

4. 결론

4-1 최종 결과물

4-2 활용방안 및 기대효과

5. 참고문헌

1. 아이디어 배경 및 목적

1-1 아이디어 배경

인터랙션(interaction)이란 의미 자체로는 ‘상호작용’을 뜻하며, 상호작용이란 두 대상간 소통 혹은 의미 전달이 이루어지는 과정을 뜻한다. 이 프로젝트를 통해 인터랙션 기기를 제작하고자 하였으며, 상호작용을 뜻하는 인터랙션 그 자체의 의미에 초점을 맞추어 인터랙션 기기를 통해 이루어지는 제품/기기-사용자 상호작용 뿐만 아니라 사용자-사용자 상호작용에도 주목하였다.

대부분의 현대인은 하루 종일 스마트폰이나 컴퓨터와 같은 다양한 기기들과 인터랙션을 진행한다 해도 과언이 아닐 정도로, 여러 시스템과 사용자 간의 인터랙션을 늘 활발히 이루어지고 있지만, 사람 간의 인터랙션에서는 부족한 양상을 보이며 디지털 방식 표현에 익숙해져 스스로의 감정이나 뉘앙스를 표현하지 못한다는 문제가 발생한다.

여기서 우리는 이러한 문제들을 해결하고자 ‘감정’ 과 ‘사람 간 상호작용’ 이라는 두 가지 키워드에 집중하였다. 즉, 사람 간의 감정 공유를 돕고 스스로의 감정을 되돌아볼 수 있는 인터랙션 기기로 접근하게 된 것이다. ‘감정 공유’ 를 쉽게 유도할 수 있는 사용자의 행동으로는 무언가를 함께 먹거나 마시기를 택했으며, 더 구체적으로는 ‘차를 함께 마시는 행위’ 를 통해 감정 공유/상호작용을 이끌어내는 것을 설정하였다. 정신건강에 유익하며 감정 진정에 도움이 되는 차의 특성이 개발하려는 기기의 기능(감정 공유 및 치유 기능)과 일맥상통하기 때문에 음료로 ‘차’를 택하게 되었다.

1-2 아이디어 소개

인터랙션 기기의 이름은 ‘tea-motion’ 으로, 감정 공유 및 치유의 매체가 되는 차 ‘tea’ 와 감정을 뜻하는 ‘emotion’ 을 합친 용어이다. 차를 마시는 시간을 통해 감정의 치유와 공유가 이루어 진다는 아이디어의 의미를 담고자 하였으며, 사용자의 감정을 대변해주는 티백을 의미하기도 한다. tea-motion 은 스스로의 감정 치유와 상대방과의 감정 공유 라는 두 가지 기능을 가진다. 혼자 또는 함께 이루어지는 감정의 상호작용을 흥미롭게 나타내어 의미를 전달하고자 하였다.

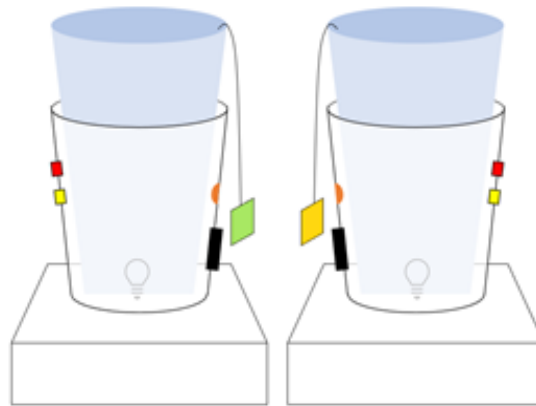
- 기능 1 - 스스로의 감정 치유

사용자가 표현한 감정에 대해 치유 효과가 있는 LED 색상을 통해 감정 치유가 이루어 진다. 컵 홀더 밑부분에 부착된 LED 색상이 차가 담긴 컵에서 은은하게 표현되는 것을 통해, 감정을 대신 표현하는 대리인의 역할을 하여 사용자가 자신의 감정을 정확히 인지할 수 있도록 한다.

- 기능 2 - 상대방과의 감정 공유

상대방이 표현한 감정을 가져옴으로써 감정 공유가 이루어지며, LED 색상으로 시각화 되어 있는 감정이 서로 섞여 새로운 색으로 표현되는 모습을 보여줌으로써 감정 상호 작용을 돕고자 한다.

위 두 가지 기능에 따라 사용자 시나리오를 설정한 뒤 구현을 진행하였다. 전체적인 인터랙션은 감정이 표현되어 있는 티백(tea-motion)을 선택하고 이를 인식하게 되면 감정에 대한 컬러테라피 색상이 컵받침 LED에 표현되는 것으로 진행된다. 시나리오 속에 등장하는 1번 버튼은 빨간 버튼, 2번 버튼은 노란 버튼을 뜻한다. (아래 그림 참고)



< 사용자 시나리오 1 - 스스로의 감정 치유 >

1. 오늘의 감정을 고려해 감정에 맞는 tea-motion 을 선택해 컵 안에 담는다.
2. 1번 버튼을 누르고 나의 tea-motion 을 인식한다.
3. 엄지 손가락 부분의 버튼 (CDS) 으로 tea-motion 을 결정한다.
4. 감정을 나타내는 색깔이 LED 로 표현되는 인터랙션이 발생한다.

< 사용자 시나리오 2 - 상대방과의 감정 공유 >

1. 2번 버튼을 누르고 상대방의 tea-motion 을 인식한다.
2. 엄지 손가락 부분의 버튼 (CDS) 으로 tea-motion 을 결정한다.
3. 상대방의 감정을 나타내는 색깔과 나의 감정을 나타내는 색깔에 대해 LED 로 표현되는 인터랙션이 발생한다. (나와 상대방의 감정이 다를 경우 - 혼합된 색깔을 표현하는 인터랙션 / 나와 상대방의 감정이 같을 경우 - 무지개 빛 색깔을 나타내는 인터랙션)

< 사용자 시나리오 3 - 차(음료)를 마시는 경우 >

1. 음료를 마시기 위해 컵을 기울인다.
2. LED 밝기가 눈에 편한 밝기 (밝기 감소) 로 변화하는 인터랙션이 발생한다.

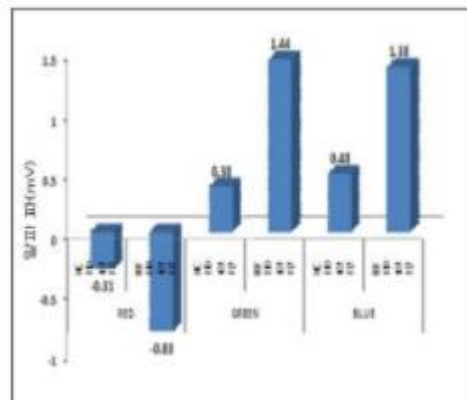
1-3 아이디어 목적

tea-motion 의 궁극적인 목적은 사람 간의 감정 공유 (상호작용) 를 돕고, 스스로의 감정 치유에 도움이 되는 것이다. 따라서 세부 목적은 2 가지로 나눌 수 있다. LED 색상을 활용한 컬러테라피로 감정 치유 기능을 더하고, 감정을 대신 표현하고 소통에 도움을 주는 감정 대리인의 역할을 수행하는 것이다.

- 컬러테라피

컬러테라피는 ‘색채 치료’ 또는 ‘색채 요법’ 이라고도 하며, 고유한 파장의 색상이 주는 영향을 치료에 활용하는 것이다. 컬러테라피의 효과에 관하여, 빛과 색상의 정신병리학적 의미를 연구하는 펠릭스 도아취(Félix Deutch)는 “빛의 작용은 육체를 구성하는 요소 뿐 아니라 정신을 구성하는 요소에도 영향을 미친다” 며 그 효과에 대해 언급한 바 있다.

컬러테라피 효과에 대하여 과학적인 입증 자료를 분석해 본 결과, 색상과 뇌는 서로 연관되어 있다는 것을 알 수 있었다. 망막-뉴런-뇌로 거쳐가는 과정에서 시상하부가 자극되며, 이 시상하부는 뇌하수체와 직접적으로 연결이 되어 있어 사람의 정신 감정에 영향을 준다는 것이다. 실제로 한 실험에서, 스트레스가 높은 집단이 낮은 집단 보다 상대적으로 적색에서는 긴장되고 녹색과 청색에서는 이완되는 효과를 받았음이 나타났다. (아래 그래프 참고)



(컬러테라피가 스트레스와 뇌파변화에 미치는 영향 -
Comparison of α wave for each group showing different stress levels)

다양한 색채들은 빛의 파장에서 비롯된 것이기 때문에, 파장이 다른 각 색채들은 다른 컬러테라피 효과를 가진다. 대표적인 그것의 증거로, 우리들은 매일 자신의 기분이나 몸 상태에 따라 다양한 색상을 선택하는 과정을 반복한다는 것을 들 수 있다. 따라서 사용자의 선택에 따라 변화되는 LED 색상을 통해 다양한 치유 효과를 제공하고자 하였다. 이와 같이 조명 컬러 테라피를 활용한 사례들은 다음과 같다.

〈Living Color Mini〉	바닥이나 테이블에 놓고 터치 휠 리모컨을 사용해 기호에 따라 1,600 가지 컬러로 변경이 가능한 컬러테라피 제품
〈mini candle〉	초가 흔들리는 움직임을 조명에 부여하여 자연스러운 분위기를 유도해 사용자에게 휴식과 안정을 가져다 주는 제품.
〈Vase〉	자연친화적 테라피 테이블 조명. 식물과 라이트, 컬러 테라피를 함께 할 수 있으며 화병으로 사용

위 사례들을 종합해보면, 사용자의 컨디션이나 날씨, 온도 등에 따라 원하는 컬러를 선택하여 자신이 원하는 환경으로 바꿀 수 있다는 공통점이 있다.

- 감정대리인

감정 대리인이란, 개인의 감정을 외부에서 대리해 주는 것을 뜻한다. 타인의 감정을 대신 느껴주고, 표현해주며 그와 맞게 관리해주는 모든 서비스와 대상이 감정 대리인이라고 할 수 있다. 이러한 감정 대리인이 각광 받게 된 것은 디지털 시대에 들어서면서 부터 이다. 디지털 시대 사람들 (디지털 원주민) 의 특징으로는, 자신이 노출되거나 표현되는 것을 꺼려하며, 현실에서 사람들과의 관계 맺음과 감정 표현을 어려워 한다는 것이다. 그렇기 때문에 자신의 감정을 직접 소모할 필요가 없는 감정 대리인에 매력을 느끼곤 한다. 대표적인 감정 대리인의 예시로는 메신저 이모티콘, 관찰 예능, 기분 맞춤 플레이리스트 서비스가 있으며, 감정 대리인 안에서도 3 가지의 유형으로 나뉘볼 수 있다.

〈감정 대리인의 유형 3 가지〉

1. 감정 대행인 : 개인의 감정을 대신 느껴주는 상품, 매체
2. 감정 대변인 : 감정을 대신 표현해 줌.
3. 감정 관리인 : 감정을 관리해주고 조절해주는 모든 서비스
(감정 대리인 트렌드 - 박세린, 최라인 (2019))

tea-motion 의 경우, 감정을 나타내는 색상을 LED 조명에서 표현한다는 점에서 감정 대변인의 역할을 하며, LED 조명 색상을 통해 해당 감정에 대해 치유 효과를 제공한다는 점에서 감정 관리인의 역할을 수행한다고 볼 수 있다.

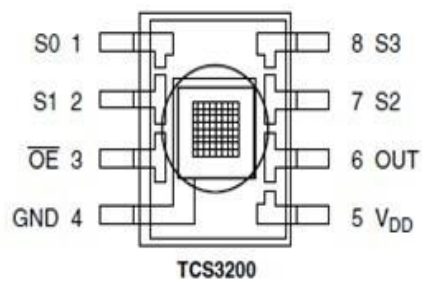
감정 대변인 유형은 타인을 통한 자신의 감정 표현이라 할 수 있다. 말이나 직접적인 제스처 대신 이모티콘, 프로필을 통해 현재 자신의 감정을 간접적으로 표현한다. 이와 같이 사람들은 감정 대변인을 내세워 직접 표현하기 어려운 감정을 표현한다. 대표적인 예시로는 액자형 관찰 예능, 메신저 이모티콘, 프로필 속 글귀 등이 있다.

감정 관리인 유형은 단순히 표현을 넘어선 감정 조절에 목적이 있다. 부정적인 감정을 해소하고 감정을 통한 색다른 감정 경험을 제공하는 것이 바로 감정 관리인의 역할이다. 감정 관리인의 사례로는 이야기 후 책을 골라주는 큐레이션 책방 서비스와 음원 큐레이션 서비스를 포함한 감성 큐레이션, ‘힐링’ 기반 서비스가 대표적 예라 할 수 있다.

2. 구현계획

2-1 필요 부품

- 컬러 센서 TCS3200



컬러센서는 들어오는 빛의 파장을 읽어 이를 토대로 값을 내어준다. 대상 물체의 색을 읽기 위하여 흰색 LED 가 같이 부착되어 있으며 LED 가 빛을 발산하면 물체에 반사된 빛을 읽어 들이는 형태이다. 이 센서는 총 10 개의 핀을 가지고 있으며, 각 핀은 다음과 같은 기능을 가지고 있다.

핀 이름	기능
VCC	전원부 +
GND	전원부 -
S0, S1, S2, S3	출력 scale 및 포토 다이오드 타입 제어
OUT	신호 출력부
LED	LED 제어판

센서는 포토다이오드 배열(8x8) 총 64 개의 다이오드로 구성되어 있으며, 이 다이오드는 각각 16 개씩 서로 다른 컬러 필터들로 덮혀 있다. 빛은 3 원색으로 이루어져 있기 때문에 RED, GREEN, BLUE 그리고 CLEAR 필터가 덮여 있다.

이 다이오드에서 감지된 광은 주파수로 변환하고, 50% duty cycle square wave(구형파) 형태로 출력한다. 이 때, 센서의 출력은 1 개이기 때문에 S2, S3 핀을 이용하여 원하는 채널을 선택한 후 주파수 정보를 얻어와야 한다. 아래 표는 S2, S3 의 값에 따른 필터 값이다.

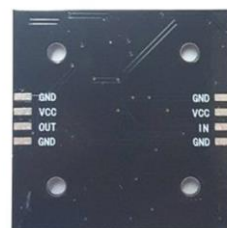
S2	S3	Photodiode type
L	L	RED
L	H	BLUE
H	L	Clear (no filter)
H	H	GREEN

센서는 빛의 색과 강도에 따라 출력하는 square waveform(구형파 : 전압 또는 전류의 시간적인 변화가 직사각형 또는 그 반복 도형이 되는 특수 파형)의 형태가 다르다. 이러한, 출력의 범위는 일반적으로 2Hz 에서 500Khz 가 되고 S0, S1 핀을 이용하여 스케일링 할 수 있다.

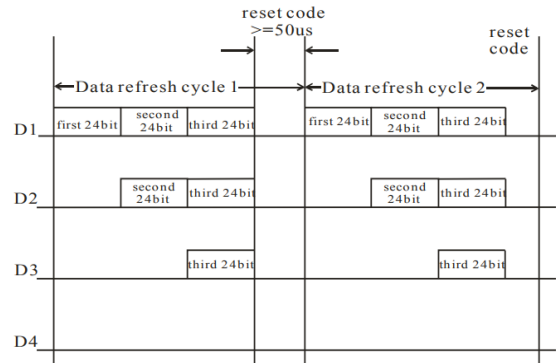
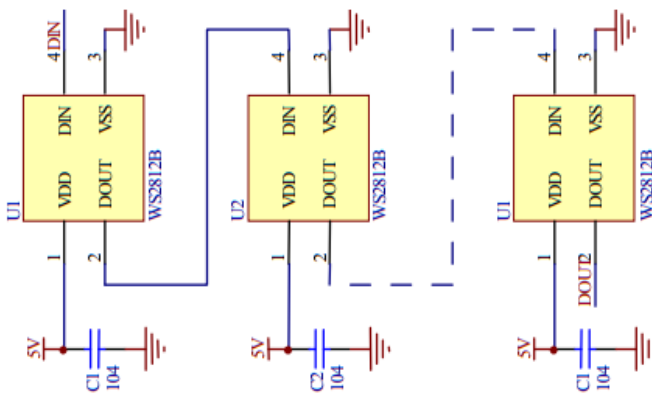
S0	S1	Output frequency scaling
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

이러한 원리를 가진 TCS3200 을 이용하여서 Teamotion 태그의 색상을 인식하였다. 센서 내 S2, S3 에 의해 선택된 필터를 통해 나오는 주파수를 S0, S1 핀을 통해 스케일링 하여 Teamotion 태그의 색상을 RGB 형태로 출력하였다.

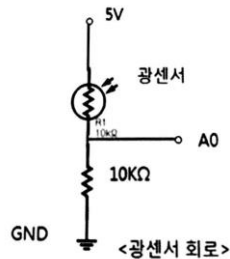
-
-
- 네오픽셀도트매트릭스



이는 WS281x 칩이 내장된 RGB LED 로 신호선 하나로 각각의 LED 에 대한 개별제어 (색상, ON, OFF)가 가능하다. 원리는 아래의 그림처럼 디지털 핀을 통해 데이터를 보내주게 되는데, 이때 첫 번째 24bit 데이터를 첫번째 픽셀이 받고, 그 데이터는 그 다음의 픽셀을 받는 형식으로 데이터가 전달된다. 디지털 핀 하나로 제어가 되다 보니 HIGH 와 LOW 신호의 길이가 정확해야 한다. 네오픽셀의 컬러는 RGB 네오픽셀은 adafruit 에서 라이브러리를 제공한다. 네오 픽셀을 사용하여, 자신의 감정 컬러 색상 표현과 다른 사람과의 감정 공유의 인터랙션이 나타나는 LED 를 발현 하였다.



- CDS 빛 센서

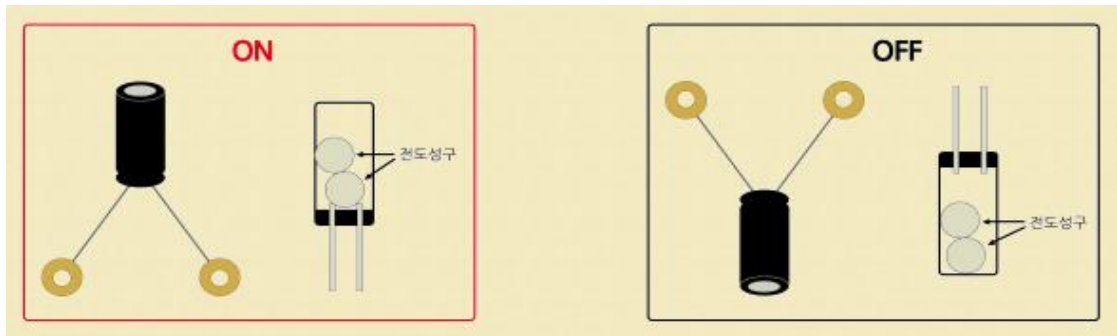


빛 센서는 황화카드뮴으로 구성된 센서이다. 아두이노에 회로를 구성하게 되면, 빛의 양으로 인해 저항값이 바뀔 때 동시에 전류의 세기도 마찬가지로 바뀌게 되고, 일정하지 않은 전자의 양이 아날로그 값으로 표현이 된다. 그 뒤에 정수로 변환시켜 빛의 양을 알아볼 수 있게 시리얼 모니터를 통해 확인할 수 있게 된다. CDS 는 극성이 존재하지 않으므로 예외적으로 GND 와 5V 를 구분하지 않아도 된다. CDS 회로를 구성할 때 전류의 값이 불안정하므로 10k 저항을 구성 시킨다. CDS 는 자신의 색상이나 다른 사람의 색상을 확정 시킬 때 사용하는 센서이다.

- 푸시 버튼

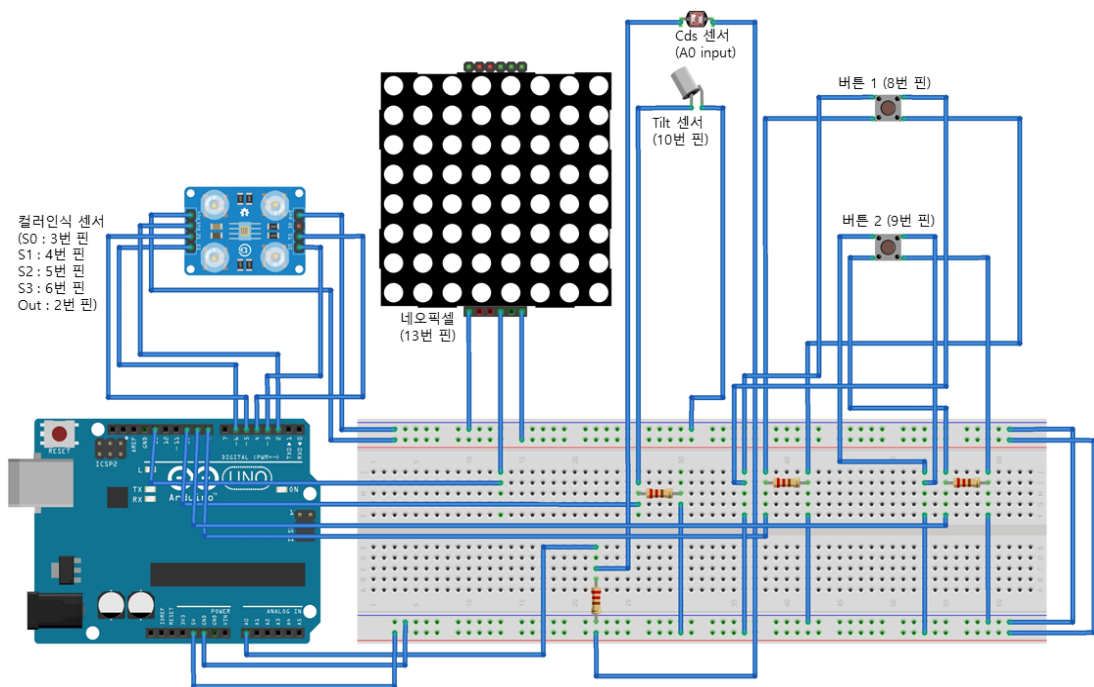
푸시 버튼은 버튼을 누르는 동안 HIGH 값을 가지며 땔 때 LOW 값을 갖는다. 네 개의 단자로 구성되며 두 단자간 서로 연결되어 있다. 버튼은 2 개를 사용하였으며, 하나는 자신의 teamotion 태그 색상을 인식하기 위해 다른 하나는 상대방의 teamotion 태그 색상을 인식하기 위해서 사용한다.

- tilt 센서



기울기 센서는 중력을 사용하여 전기적 신호를 개폐한다. 센서 안에는 전기가 통하는 구슬이 들어있으며, 그 구슬의 위치에 따라 HIGH 또는 LOW 신호를 출력한다. 기울기 센서가 위를 향했을 때 기울기 센서 속에 있는 구슬이 양쪽 단자 사이를 연결시켜 주기 때문에 전류가 흐르게 되고, 반대로 기울기 센서가 아래로 향했을 때는 양쪽 단자가 연결되지 않기 때문에 전류가 흐르게 되지 않는다. 전기의 단락, 개방 여부만 확인할 수 있으며 얼만큼 기울어졌는지 기울기 정도까지 확인할 수 없다.

2-2 회로 설계안



- 컬러센서
 - OUT : 2 번 pin
 - S0 : 3 번 pin
 - S1 : 4 번 pin
 - S2 : 5 번 pin
 - S3 : 6 번 pin
- 푸시버튼

- 자신의 컬러 인식 버튼 1 : 8 번 pin
- 상대방의 컬러 인식 버튼 2 : 9 번 pin
- tilt 센서 : 10 번 pin
- 네오픽셀 도트 매트릭스 : 13 번 pin
- CDS 빛센서 : A0

2-3 인터랙션 구조도

- Input

Button1 : 자신의 색상을 인식하기 위해 Color Sensor 을 on 하는 버튼

Button2 : 상대방의 색상을 인식하기 위해 Color Sensor 을 on 하는 버튼

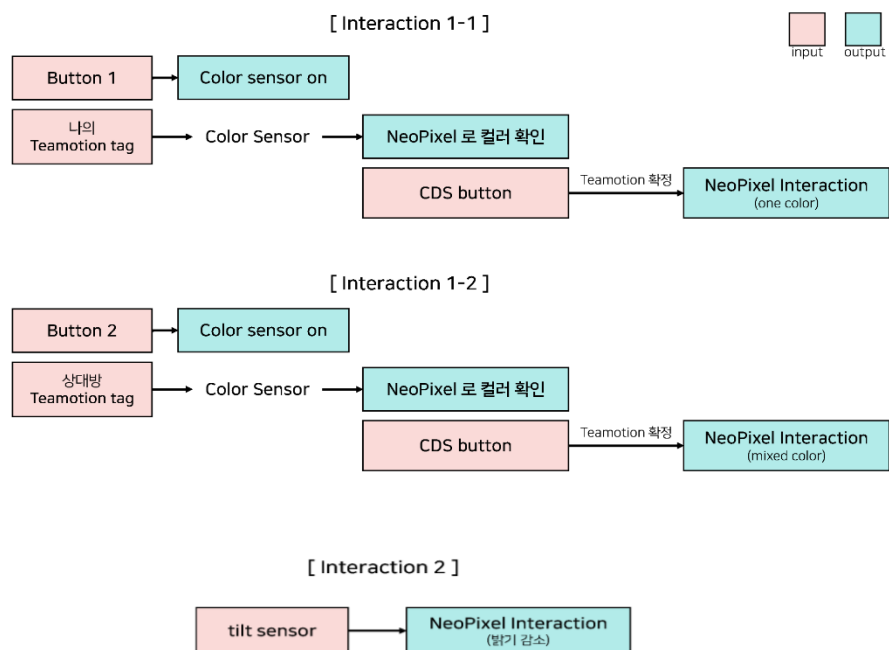
CDS : 색상을 확정 시키는 센서

Tilt : 컵의 기울기 여부를 확인하는 센서

컬러 센서 : Teamotion 의 태그를 인식하는 센서

- Output

NeoPixel : 색상이 나타나는 LED



- Interaction 1-1 “혼자 사용할 때”

Button1 을 누르게 되면, Color sensor 가 on 이 된다. 나의 Teamotion tag 를 on 상태인 Color sensor 에 인식하게 되면, NeoPixel 로 컬러를 확인 할 수 있고, 컬러가 자신이 선택한 컬러일 경우, CDS 를 이용하여 Teamotion 을 확정 짓는다. 확정을 지으면, NeoPixel 에서 확정 지은 색상이 서서히 밝아졌다가 서서히 어두워지는 인터랙션을 두 번 반복한 후 계속 그 색상을 보여준다.

- Interaction 1-2 “함께 사용할 때”

Button2 을 누르게 되면, Color sensor 가 on 이 된다. 상대방의 Teamotion tag 를 on 상태인 Color sensor 에 인식하게 되면, NeoPixel 로 컬러를 확인 할 수 있고, 컬러가 상대방의 컬러와 일치할 경우, CDS 를 이용하여 색상을 확정 짓는다. 상대방의 색상이 나의 색상과 같은 경우 레인보우 컬러를 2 번 보여주고 다시 자신의 컬러로 돌아오는 인터랙션을 보여준다. 만일, 상대방의 색상이 나의 색상과 다를 경우, 상대방과 나의 색상이 섞이면서 상대방의 색상을 보여줬다가 다시 나의 색상으로 돌아오는 인터랙션을 보여준다.

- Interaction 2 “컵을 기울일 때”

컵을 기울이게 되면 tilt sensor 가 기울었다고 판단하여, 약 6 초 동안 NeoPixel 의 밝기를 감소하였다가 다시 밝아지는 인터랙션을 보여준다.

3. 구현 내용

3-1 코드 구현

- 기본 setup

```
#include <Adafruit_NeoPixel.h>

#define PIN 13
#define NUM_LEDS 25
Adafruit_NeoPixel RGB_LED = Adafruit_NeoPixel(25, 13, NEO_GRB);
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB + NEO_KHZ800);
int cds = A0;
#define S0 3
#define S1 4
#define S2 5
#define S3 6
int btn1= 7; //인식
int btn2= 8; //저장1 (바꾼다)
int btn3 = 9; //저장2 (섞는다)
int tilt = 10;
int bv1 = 0;
int bv2 = 0;
int bv3 = 0;

int color = 0;
int bcount = 0;
int myColor = 0;
int yourColor = 0;
// 1 : red, 2 : yellow, 3 : blue, 4 : green

int colorprint_save_number[3];
int mya, myb, myc;
int ua, ub, uc;
|
#define sensorOut 2
int frequency = 0;
int red, blue, green;
```

adafruit 에서 제공하는 Adafruit_NeoPixel 라이브러리를 사용하기 위해 include 했다.

네오 픽셀을 사용하기 위해 객체를 생성한다. 첫 번째 인자값은 네오 픽셀의 개수, 두 번째 인자 값은 네오 픽셀이 연결된 아두이노 핀 번호, 세 번째 인자 값은 네오 픽셀의 타입에 따라 바뀌는 flag 를 의미한다.

cds 센서, 네오 픽셀의 센서, tilt 센서, 버튼 까지 모두 신호 핀에 연결한 값과 각 함수에서 사용되는 변수들이 전역 변수로 정의한다.

*앞서 언급된 버튼 1 과 버튼 2 는 코드 상에서 변수 btn2 와 btn3 으로 구현되었다.

```

void setup() {
  pinMode(tilt, INPUT);
  RGB_LED.begin(); // RGB_LED 불러옵니다.
  RGB_LED.setBrightness(100); // RGB_LED 밝기조절
  RGB_LED.clear(); // RGB_LED 초기화합니다.

  strip.setBrightness(100);
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'

  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(sensorOut, INPUT);
  pinMode(btn1, INPUT);
  pinMode(btn2, INPUT);
  pinMode(btn3, INPUT);
  pinMode(cds, INPUT);
  digitalWrite(s0, HIGH);
  digitalWrite(s1, LOW);
}

```

처음 네오픽셀을 사용할 때는 네오픽셀 객체의 begin 함수를 실행 해줘야 한다. begin 함수의 경우 내부에 단순히 아두이노 디지털 핀을 out 으로 설정하고 LOW 가 출력이 되도록 한다.

setBrightness 함수로 전체 네오 픽셀의 전체 밝기(0~255)를 먼저 설정한다. 컵 안에 넣었을 때 색상이 뚜렷하게 구분되면서 은은한 빛을 내는 밝기 정도(100)을 설정했다. setup 의 strip.show()함수는 NeoPixel 이 이전 프로그램에 의해 불이 켜진 경우 모든 NeoPixel 을 초기 "꺼짐"상태로 초기화한다. 모든 센서들을 pinMode 설정으로 설정한 핀이 INPUT, OUTPUT 인지 선언해준다. 출력 주파수가 20% scaling 하기 위해 s0, s1 핀을 HIGH,LOW 로 선언한다.

- CDS 센서를 활용한 컬러 변수 저장 기능 코드 구현

```

int cds_btn1() {
  cds = analogRead(A0);
  mapping = map(cds, 600, 850, 0, 7);
  while(mapping > 4) {
    cds = analogRead(A0);
    mapping = map(cds, 600, 850, 0, 7);
  }
}

```

led 형광등이 있는 환경에서 진행되었다. cds 센서의 값을 analogRead 로 읽어 cds 변수로 저장한다. 센서를 가렸을 때와 가리지 않았을 때 해당 값을 출력해본 결과, 약 600 ~ 1000 으로 인식 되었다. 중간 중간 튀는 값을 보정하기 위하여 map 함수를 사용하였다. 센싱 된 값보다 조금 더 넓은 범위로 mapping 해주었다. 주변 환경에 따라 너무 밝거나 어두울 경우를 대비하였다. 주로 값이 환경에 따라 상승세는 많이 보였지만, 하락세는 잘 보이지 않기 때문에 밝게 인식되는 값의 범위를 더 늘려주었다. 해당 값을 0~7 단계로 mapping 하였다. mapping 의 구간을 너무 좁거나 넓게 지정하게 되면 가려도 인식을 잘 하지 못하게 되기 때문이다. 컵을 잡았을 때 가려지는 cds 센서로 인하여 떨어지는 값으로 인하여 mapping 값이 4 보다 작아진다. while 문을 빠져나오게 된다. while 문 안에는 컬러 인식 센서(tcs3200)가 색상을 인식하여 해당 값을 return 하는 코드가 구현되어 있다.

- 컬러인식 센서(TCS3200)를 활용한 컬러 인식 코드 구현

```

digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
frequency = pulseIn(sensorOut, LOW);
Serial.print("R= ");
Serial.print(frequency);
red = frequency;
Serial.print(" ");
delay(100);

digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
frequency = pulseIn(sensorOut, LOW);
Serial.print("G= ");
Serial.print(frequency);
green = frequency;
Serial.print(" ");
delay(100);

digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
frequency = pulseIn(sensorOut, LOW);
Serial.print("B= ");
Serial.print(frequency);
blue = frequency;
Serial.println(" ");

```

S2 와 S3 의 값이 LOW, LOW 일 때 HIGH, HIGH 일 때, LOW,HIGH 일 때 주파수 값을 읽는다. 각 각 R G B 의 주파수 값을 각 색상 변수에 기록한다.

```

if(blue > red && blue > green){
    Serial.println("(Yellow Color)");
    colorprint(255, 255, 0);
    color = 1;
}
else if (blue<red && blue<green)
{
    Serial.println("(Blue Color)");
    colorprint(0, 0, 255);
    color = 2;
}
else if (red<blue && red<green)
{
    Serial.println("(Red Color)");
    colorprint(255, 0, 0);
    color = 3;
}
else if (green<red && green<blue)
{
    Serial.println("(Green Color)");
    colorprint(0, 255, 0);
    color = 4;
}
else{
    Serial.println("I don't know");
}

```

티모션에 사용되는 색상은 빨강 노랑 초록 파랑이다. 각 색상 별로 센싱한 값을 비교를 진행했다. S2, S3 핀으로 인식하여 변수에 저장한 색상 별 주파수 값을 이용하여 각 색상 별 대소 비교 식을 구하였다. 이 구문이 최종적으로 어떤 색인지 구별한다. 노랑은 빨강과 초록 주파수의 값이 파랑의 주파수 값보다 작다고 센싱 되었다. 파랑 색의 주파수 값은 빨강과 초록 주파수의 값보다 작다고 센싱 되었다. 이처럼 각 색상들은 다른 색에 비해서 작은 값으로 인식 되었다. 그렇게 주파수 값을 최종적으로 색상으로 구분하고, 해당 사항을 color 변수에 저장한다. 저장한 color 값을 return 하여 다른 함수에서 인식된 색상이 무엇인지 구별할 수 있다.

- 컵을 기울이면(tilt 센서) 밝기 감소 인터랙션
- loop 함수 내에서 tilt 센서가 기울어졌다고 판단이 되면 해당 함수가 실행된다.

```
void interaction_tilt(int a,int b,int c){

    for(int i = 100 ; i > 0 ; i -=4){
        for(int j = 0 ; j < NUM_LEDS; j++){
            strip.setBrightness(i);
            strip.setPixelColor(j, strip.Color(a, b, c));
        }
        strip.show();
        delay(30);
    }
    delay(5000);
    for (int i=0; i<100; i+=4) {
        for(int j=0; j<NUM_LEDS; j++) {
            strip.setBrightness(i);
            strip.setPixelColor(j, strip.Color(a, b, c));
        }
        strip.show();
        delay(30);
    }

    void interaction_1(int a, int b, int c) {
        for (int i=0; i<100; i+=4) {
            for(int j=0; j<NUM_LEDS; j++) {
                strip.setBrightness(i);
                strip.setPixelColor(j, strip.Color(a, b, c));
            }
            strip.show();
            delay(30);
        }
        for (int i=100; i>0; i-=4) {
            for(int j=0; j<NUM_LEDS; j++) {
                strip.setBrightness(i);
                strip.setPixelColor(j, strip.Color(a, b, c));
            }
            strip.show();
            delay(30);
        }
    }
}
```

이중 for 문을 두 번 사용하여 천천히 밝기를 줄였다가 늘리는 방식을 채택 하였다. 각 for 문 중 첫 번째 for 문은 밝기 값을 조절할 int i 변수를 위하여 사용되었고 두 번째 for 문은 모든 네오 픽셀의 픽셀번호를 지정하기위해 사용되었다. interaction_1 과 interaction_tilt 는 같은 원리로 구현되었다.

- 상대방과 teamotion 을 공유할 때
 - 다른 감정일 경우 mix 인터랙션

```
void mixColor(int myColor, int yourColor){
  switch(myColor){
    case 1: // 1 : yellow
      mya = 255;
      myb = 255;
      myc = 0;
      break;
    case 2: // 2 : blue
      mya = 0;
      myb = 0;
      myc = 255;
      break;
    case 3: // 3 : red
      mya = 255;
      myb = 0;
      myc = 0;
      break;
    case 4: // 4 : green
      mya = 0;
      myb = 255;
      myc = 0;
      break;
  }

  switch(yourColor){
    case 1: // 1 : yellow
      ua = 255;
      ub = 255;
      uc = 0;
      break;
    case 2: // 2 : blue
      ua = 0;
      ub = 0;
      uc = 255;
      break;
    case 3: // 3 : red
      ua = 255;
      ub = 0;
      uc = 0;
      break;
    case 4: // 4 : green
      ua = 0;
      ub = 255;
      uc = 0;
      break;
  }
}
```

mixColor 함수는 나의 색상 값(1,2,3,4 중 하나) 너의 색상 값 (1,2,3,4 중 하나) 를 인자로 받는다.
switch 문은 각 숫자가 의미하는 색상의 r,g,b 값으로 변수에 저장한다.

```
if(yourColor == myColor){
  rainbow();
}

else{
  int r,g,b;
  r = (mya+ua)/2;
  g = (myb+ub)/2;
  b = (myc+uc)/2;
  colorinteraction(ua, ub, uc, r, g, b);
  Serial.println("out of colorinteraction");
  colorprint2(r, g, b, mya, myb, myc);
}
```

서로 같은 색상인지 다른 색상인지 if 문을 사용하여 조건을 따진다. 같은 색상 일 경우 rainbow 함수를 호출한다.

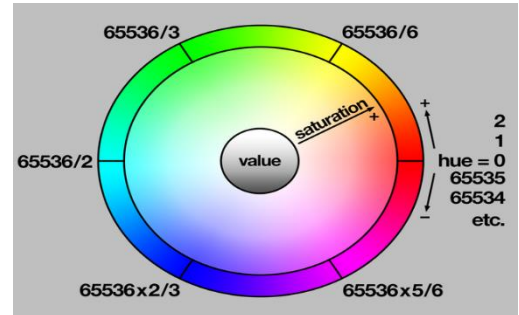
```

void rainbow() {
    int n, e, w;
    for (long firstPixelHue = 0; firstPixelHue < 2*65536 ; firstPixelHue += 256) {
        int k = 0;
        int pixelHue = firstPixelHue + ( k*65536L / strip.numPixels());
        for (int i = 0; i < strip.numPixels(); i++) {
            strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
            strip.show();
        }
        k++;
    }

    n = 255;
    e = 0;
    w = 0;
    int x = mya;
    int y = myb;
    int z = myc;

    while (!(n==x && e==y && w==z)) {
        if (n>x) {
            n -= 1;
        }
        else if (n<x) {
            n += 1;
        }
        if (e>y) {
            e -= 1;
        }
        else if (e<y) {
            e += 1;
        }
        if (w>z) {
            w -= 1;
        }
        else if (w<z) {
            w += 1;
        }
        colorprint(n, e, w);
    }
}

```



NeoPixel 라이브러리에서 색조는 16 비트 숫자로 표현된다. 빨간색의 경우 0에서 시작하여 먼저 노란색 (약 65536/6 또는 10922 주기)으로 증가하고 녹색, 청록색 (32768의 중간 지점에서), 파란색, 자홍색 및 다시 빨간색으로 증가한다. 자신의 코드에서 모든 색조 관련 변수가 오버플로 또는 언더플로 되도록 허용할 수 있다. ColorHSV 함수는 색조, 채도 값을 인수로 받아 HSV에서 RGB 값으로 변환한다.

첫 번째 for 문에서 이러한 hsv 값을 주고 두 번째 for 문에선 전체 네오픽셀이 해당 컬러 값으로 변하게 한다. 두 번의 레인보우 인터랙션이 끝난 후 while 문을 통해 현재 네오픽셀의 색상 값이 원래 자신의 색상 값인 mya, myb, myc 변수와 같아질 때까지 변수 값을 더하고 뺀다.

- 그 밖에 사용된 함수

```

void colorprint(int a, int b, int c){
    RGB_LED.setPixelColor(0, a, b, c);
    RGB_LED.show();
    RGB_LED.setPixelColor(1, a, b, c);
    RGB_LED.show();
    RGB_LED.setPixelColor(2, a, b, c);
    RGB_LED.show();
}

```

colorprint 함수는 나타내야 하는 색상의 r, g, b 값을 a, b, c 변수로 받아서 전체 네오픽셀에 출력하는 함수이다. setPixelColor 함수는 바꾸고 싶은 픽셀 index와 나타내야 하는 색상의 r, g, b 값을 인자로 받아서 해당 픽셀의 색상을 바꿔주는 함수이고, show 함수로 최종적으로 바뀌게 된다. 따라서 r g b 값으로 0~24 번째 픽셀 모두의 색상을 변경해주는 함수이다.

```

void colorprint_save(int myColor) {
    // 1 : red, 2 : yellow, 3 : blue, 4 : green
    if(myColor==1){
        colorprint_save_number[0] = 255;
        colorprint_save_number[1] = 255;
        colorprint_save_number[2] = 0;
    }
    else if(myColor==2){
        colorprint_save_number[0] = 0;
        colorprint_save_number[1] = 0;
        colorprint_save_number[2] = 255;
    }
    else if(myColor==3){
        colorprint_save_number[0] = 255;
        colorprint_save_number[1] = 0;
        colorprint_save_number[2] = 0;
    }
    else if(myColor==4){
        colorprint_save_number[0] = 0;
        colorprint_save_number[1] = 255;
        colorprint_save_number[2] = 0;
    }
}

void printMyColor(int myColor){
    if(myColor==1){
        colorprint(255, 255, 0);
    }
    else if(myColor==2){
        colorprint(0, 0, 255);
    }
    else if(myColor==3){
        colorprint(255,0,0);
    }
    else if(myColor==4){
        colorprint(0,255,0);
    }
}

```

두 함수 모두 myColor(1,2,3,4 中 하나의 값) 을 인자로 받아서 숫자가 어떤 색상을 의미하는지 판별한다. 예를 들어, 빨간색 = 1 이라 사전에 정해두었기에 myColor ==1 맞다면 빨간색으로 네오픽셀 값을 모두 프린트하거나 colorprint_save_number 에 차례로 빨간색의 rgb 값을 저장한다.

```

void colorprint2(int a, int b, int c, int x, int y, int z){
    //abc : 현재색바뀐색, xyz : 돌아갈원래색
    //색 바꾸는 인터랙션
    while(!(a==x && b==y && c==z)){
        if(a>x){
            a -= 1;
        }
        else if(a<x){
            a += 1;
        }
        if(b>y){
            b -=1;
        }
        else if(b<y){
            b +=1;
        }
        if(c>z){
            c -=1;
        }
        else if(c<z){
            c +=1;
        }
        colorprint(a, b, c);
    }
}

```

colorprint2 함수는 mixColor 에서 중간색으로 바뀌었다가 다시 자신의 색으로 돌아갈 시에 자연스러운 인터랙션을 위한 함수이다. 현재 바뀐 색의 rgb 값과 원래 자신의 색 rgb 값을 인자로 받는다. 해당 값들이 같아 질때까지 1 씩 더해지고 빠지는 while 문이 실행된다.

- loop 함수

```
void loop(){
    bv2=digitalRead(btn2);
    bv3=digitalRead(btn3);

    if (bv2==1 || bv3==1){
        if (bv2==1){ //색 변경 코드 , 단순변심을 위한 색 바꾸기 기회
            myColor = cds_btn1();
            colorprint_save(myColor);
            for(int i = 0; i < 2; i++) {
                interaction_1(colorprint_save_number[0],colorprint_save_number[1],colorprint_save_number[2]);
                delay(300);
            }
            for (int i=0; i<100; i+=4) {
                for(int j=0; j<NUM_LEDS; j++) {
                    strip.setBrightness(i);
                    strip.setPixelColor(j, strip.Color(colorprint_save_number[0],colorprint_save_number[1],colorprint_save_number[2]));
                }
            }
            strip.show();
            delay(30);
        }
        printMyColor(myColor);
    }
    if (bv3==1){ //실제로 해야하는 인터랙션
        yourColor = cds_btn1();
        mixColor(myColor, yourColor);
    }
}

int digitalstate = digitalRead(tilt);
int analogstate = analogRead(tilt);
int tiltmap=0;
tiltmap = map(analogstate, 100, 300, 0, 7);
if(tiltmap<3 && digitalstate==HIGH ){
    interaction_tilt(colorprint_save_number[0],colorprint_save_number[1],colorprint_save_number[2]);
    delay(3000);
}
else{
    Serial.println("notilt");
}
}
```

색상 값 인식 시작 버튼 2 와 3 을 digitalRead 를 통하여 값을 읽는다. 버튼 2 번이나 3 번을 눌렀을 조건과 그 중 2 번인 경우 그 중 3 번인 경우를 각 if 문 조건으로 확인한다. 조건문 안에 cds 버튼 함수(색상 저장하는 기능)를 실행하여 해당 함수의 return 값(color 1,2,3,4)를 myColor / yourColor 변수에 저장한다.

- 버튼 2 번을 눌렀을 경우

자신의 색상 저장 및 확정 코드를 실행한다. 저장된 myColor 변수 값에 따른 rgb 값을 colorprint_save 함수를 통하여 colorprint_save_number 배열에 저장하고 이 값을 index 로 접근해 밝기 조절 interaction_1 함수를 실행한다. 그 후로 다른 동작이 있기 전까지 자신의 color 색상을 유지하기위한 함수를 실행한다.

- 버튼 3 번을 눌렀을 경우

cds_btn 함수를 통하여 yourColor 에 상대의 컬러 값을 저장한다. 나와 상대의 컬러 값을 mixColor 함수의 인자로 보내어 같은 색상일 때와 다른 색상일때의 인터랙션을 실행시킨다.

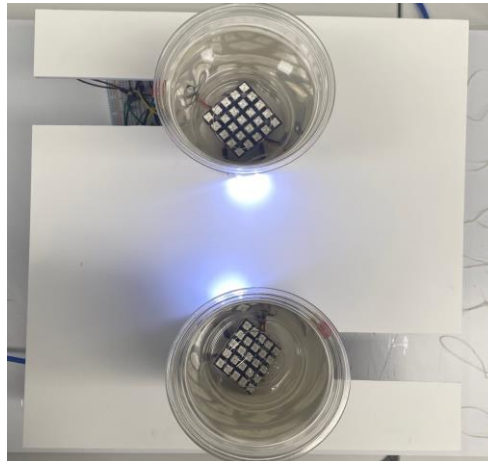
- 컵을 기울였을 경우

tilt 센서를 digital 과 analog 값으로 읽는다. digital 값은 티 모션 태그를 인식할 때나 살짝 이동할 때도 HIGH 값으로 인식하기 때문에, analog 값을 매핑 하여 두 조건이 모두 충족될 때 조건문을 통과하게

한다. 자연스러운 밝기 조절이 되는 `interaction_tilt` 함수를 호출한다. 이때, 자신의 색상은 `colorprint_save_number` 배열에서 `index` 로 접근하여 값을 인자로 보낸다.

3-2 하드웨어 구현 및 목업

- 네오픽셀 위치



수중에서의 빛의 색의 밝기, 수중에서의 빛의 확산율, 탈 부착의 용이함, 불편하지 않은 정도를 기준으로 수중에서 가장 효과적인 색감을 나타내는 LED 위치는 수면 아래 부분에 위치한 LED 가 빛이 가장 자연스럽게 확산되고, 색이 물에 선명하게 비쳤다. 그래서, 컵홀더 안쪽에 LED 를 위치하여서 수면 아래 부분에 LED 가 위치하도록 하였다.

- 버튼과 CDS

버튼과 CDS 는 사용자가 컵을 잡았을 때 자연스럽게 버튼을 누를 수 있도록, 나의 teamotion 태그를 인식하기 위해 컬러 센서를 on 하는 버튼 1 은 검지 위치에 상대방의 teamotion 태그를 인식하기 위해 컬러 센서를 on 하는 버튼 2 는 중지 위치에 부착하였다. CDS 의 위치는 엄지 위치에 부착하였다. CDS 는 엄지손가락으로 버튼을 누르기에 불편하다는 점을 고려하여서 많은 힘을 주지 않아도 입력이 되는 CDS 를 선택하였다.

- 컬러센서 위치

컬러센서는 나의 teamotion 태그나 상대방의 teamotion 태그를 인식해야 하기 때문에 컵홀더 바깥쪽에 부착했으며, 사용자가 음료를 마실 때 방해받지 않게 하기 위해서 하단 쪽에 위치하였다.

- tilt 센서 위치

tilt 센서는 컵을 기울였을 때, tilt 센서 내에 있는 구슬이 움직일 수 있도록 컵 홀더 내부에 세로로 부착하였다.

- 그 외

모든 점선들이 컵 받침 밑으로 나와서 미관상으로 좋지 않고, 점선 때문에 책상에 잘 세워지지 않는 것을 보완하기 위해서 포맥스라는 소재로 책상을 만들어 점선이 밑으로 빠질 수 있는 공간을 만들었다.

Teamotion 같은 경우, '감정을 선택한다'라는 의미를 가질 수 있도록, 태그에 표정 이모티콘을 그려 넣었다. 그 반대편에는 컬러센서가 인식할 수 있는 색종이를 붙였다.

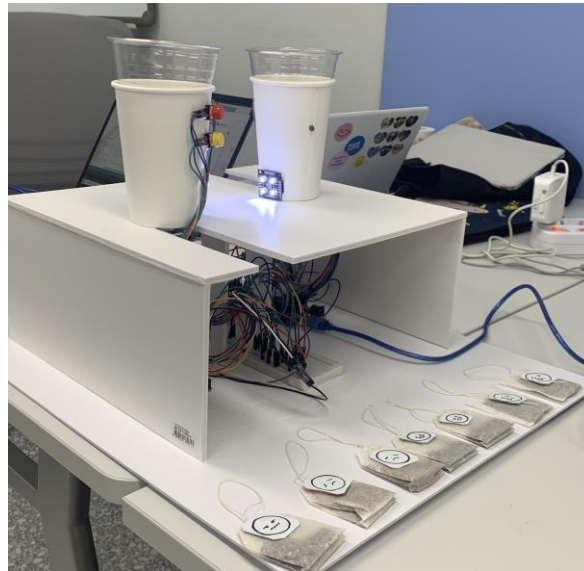
4. 결론

4-1 최종 결과물

- 최종 결과물 사진

최종 결과물에 대한 사진과 내용은 다음과 같다.

〈사진 1. 전체적인 결과물〉



→ 책상, teamotion 티백들, 그리고 컵 홀더와 컵으로 구성된 결과물의 전체적인 모습

〈사진 2. 컵 받침 결과물〉



→ 위와 같이 버튼 1, 버튼 2, cds 센서, tcs3200 컬러센서, tilt 센서 (컵 내부), 네오픽셀 (컵 내부) 로 이루어진 컵 홀더의 최종 결과물 모습

〈사진 3. 사용 시의 전체적인 모습〉



→ 위와 같이 개인의 teamotion 을 인식하는 모습, 그리고 상대방과 함께 서로의 teamotion 을 인식하는 모습

- teamotion 별 감정과 완화 기능

컬러테라피에 대한 여러 논문을 자료조사 하였다. 다양한 자료들 중, 다음과 같은 내용의 자료를 참조하였다. <표 1>의 컬러의 상징성과 감성적 치유 효과에 대한 내용을 참고하여 최종적인 teamotion 의 컬러와 감정을 매치할 때, 해당 컬러에 대한 감정 완화 기능을 녹여 제작하였다.

컬러명	연상 감정 및 상징	감성적 치유 효과
빨간색	피, 불, 정열, 애정, 활력, 야만, 집착, 위험, 혁명, 삶, 힘, 생명력	무기력, 피해의식, 집착, 미련, 과거지향적, 에너지의 고갈, 면역력 저하
주황색	자유, 변화, 호기심, 원기, 적극성, 만족, 풍부, 유쾌, 긍정, 낙관	자학, 성 집착, 공포, 트라우마, 신경쇠약, 우울증, 쇼크, 오만, 자기과시, 초조, 심한 분노
노란색	쾌활, 부드러움, 다정함, 경박, 주의력, 집중력, 융통성, 아이디어, 향상심	의기소침, 우울증, 부정적 생각, 낮은 자긍심, 두려움, 공포, 신경질, 변덕, 의심
초록색	평화, 자연, 안정, 안전, 신선함, 균형, 휴식, 편안함, 성실, 공평	시기, 질투, 소유욕, 고집, 침체, 무기력증, 감정적 진정, 균형 회복
파란색	이성, 진실, 냉정, 희망, 우울, 고독, 성실, 신의, 정직	긴장완화, 흥분·초조 완화, 정서 안정
남색	냉정, 완벽, 권위, 직관, 경건함, 우울, 통찰력	정신적, 중독성 강한 감정 정화
보라색	영적 지도자, 예술성, 고귀함, 정신적 봉사	우울증, 정신적인 공허함, 영적인 능력 개발

<표 1. 컬러의 상징성과 감성적 치유 효과>

teamotion 은 다음과 같이 총 4 가지 종류의 감정 (긴장, 무기력, 행복, 슬픔) 들로 구성된다.



컬러테라피 자료들을 참조해 보았을 때, 긴장을 완화해주는 색은 파란색, 무기력 및 에너지 고갈을 완화해주는 색은 빨간색, 감정의 진정과 균형 회복을 도와주는 색은 초록색, 우울증과 부정적인 생각을 완화해주는 색은 노란색인 것을 확인할 수 있었다. 따라서 해당 감정에 대한 완화 효과 및 감정에 대한 치유 효과와 긍정적인 감정 유지를 위하여 다음과 같이 색을 매치하였다.

< 선택 가능 감정 4가지 >

컬러	연상 감정 및 상징	감성적 치유 효과
빨간색	애정, 집착, 위험	무기력 완화, 집착 ↓, 에너지 향상
노란색	다정함, 쾌활, 부드러움	자긍심 상승, 변덕 완화, 우울증 완화
초록색	휴식, 편안함, 안정	균형 회복, 시기 질투 완화, 무기력증 완화
파란색	우울, 고독, 진실, 희망	정서 안정, 흥분 초조 완화

사용자는 색을 선택하는 것이 아닌, 감정에 대한 얼굴 모양을 보고 자신에게 현재 해당되는 감정을 선택하여 해당 감정에 대한 치유 효과 색상을 컵받침을 통해 확인할 수 있다.

4-2 활용방안 및 기대효과

● 종이컵을 이용한 컵 홀더 형태

- 쉬운 상용화와 간편한 관리가 가능하다.

- 사용자는 컵을 재사용하는 것이 아닌, 컵 홀더를 계속해서 사용하기 때문에 위생 문제에 대한 걱정 및 부담을 줄일 수 있다.

● 컬러테라피의 감성적 치유 및 심리적 치유 효과

- 색채 자극에 있어 스트레스가 높은 집단이 낮은 집단에 비해 상대적으로 컬러에 따른 이완 혹은 긴장의 반응이 크게 나타난 한 연구 결과를 통해, 색채를 통한 인체의 심신의 조화를 바로잡는

컬러테라피는 효과가 있다고 할 수 있다. teamotion 은 위와 같은 컬러테라피를 적용시킨 제품이기 때문에 감성적 치유에 대한 효과를 기대할 수 있다.

- 감정관리인의 역할을 하는 서비스 수요가 증가하는 것을 미루어 보아 현대 사회인들은 자신의 감정을 해소하고 통제하는 것에 어려움을 겪는다는 점을 엿볼 수 있다. 따라서 teamotion 을 통한 자신의 감정 표현과 상대방의 감정 인식의 행위는 이와 같은 현대인들의 미숙한 감정 표현에 대한 문제점을 해결해줄 수 있다. 자신의 감정을 정확히 인지하고 공유하는 행동 유도를 기대할 수 있고, 시각화 된 감정이 서로 섞이는 모습을 보여줌으로써 감정의 상호작용에 도움을 미친다.

● 디자인 1) 조명의 밝기 조절 디자인

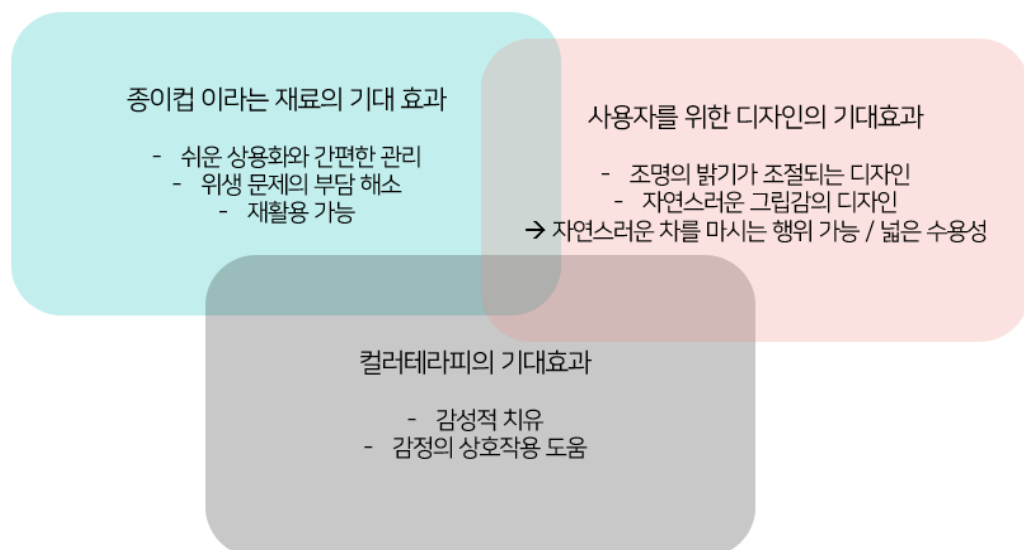
- 사용자가 차를 마시는 행위에 대해 조명의 불빛으로 인한 불편함을 느끼지 않도록 디자인 함으로써 자연스러운 음료 섭취 행동이 가능하다.

● 디자인 2) 자연스러운 그립감을 지닌 버튼 설계 및 배치 디자인

- 사용자가 컵을 잡는 모션에 대해 불편함을 느끼지 않도록 손가락 위치에 맞춰 버튼과 센서를 배치하여 자연스럽게 컵을 잡는 행동이 가능하다.

⇒ 위와 같은 2 가지 디자인을 통해 teamotion 의 사용자는 실내 어디든 시간이나 외부 환경에 많은 방해 받지 않고 자연스럽게 차를 마시는 행위와 감정을 치유 및 공유할 수 있어, 넓은 범위에서의 수용성을 갖는다.

결론적으로 teamotion 은 다음과 같은 기대효과를 갖는다.



5. 참고 문헌

- 임효선 외 4 명 (2018). 컬러테라피를 활용한 LED 반신욕 디바이스 개발연구. 한국디자인학회 학술발표대회 논문집, 288-289
- 이경재 외 2 명 (2012). LED 조명 테라피 제품군 분석을 통한 활용 방안 연구. 디지털디자인학연구, 12(4), 415-427
- 권예지 외 5 명 (2015). 한국의 디지털 원주민과 디지털 이주민. 한국방송학보, 29(2), 5-40
- 박세린, 최라인 (2019). 감정 대리인 트렌드. 마케팅, 53(5), 48-59
- 컬러테라피 활용 예시-컬러테라피 나에게 필요한 컬러는 무엇? (2019). 현대케피코 사보 KEFICO PLUS , 1, 2 월호 원문 발췌
- 홍인석 외 5 명 (2017). 커피전문점 고객을 위한 스마트 감성조명 디바이스 설계 및 구현. 한국 HCI 학회 학술대회, 192-195
- 박세린, 최라인 (2019). 감정 대리인 트렌드. 마케팅, 53(5), 48-59
- 유병숙 (2019), 감정치유를 위한 자연치유의 통합적 접근, 한국자연치유교육학회지 제 3 권 제 1 호, 103-125
- 홍근주 외 4 명 (2009), 컬러테라피가 스트레스와 뇌파변화에 미치는 영향, 대한피부미용학회지 제 7 권 제 1 호
- 김일균 (2019), 당신대신 울고 웃어주는 감정 대리인, 문화로 읽는 사회, 64-67