

| 2019-1 Take Out Mini Project

: 15기 미니프로젝트 아이디어 발표

작 품 명 : 조광 정도에 따른 가림막 개폐 장치

소 속 : 가천대학교 중앙동아리 Take Out

INDEX

| 01 작품 개요

| 02 디자인

| 03 사용 재료

| 04 개발 상황

| 05 기술적 요구사항

| 06 기술적 문제점 및 해결

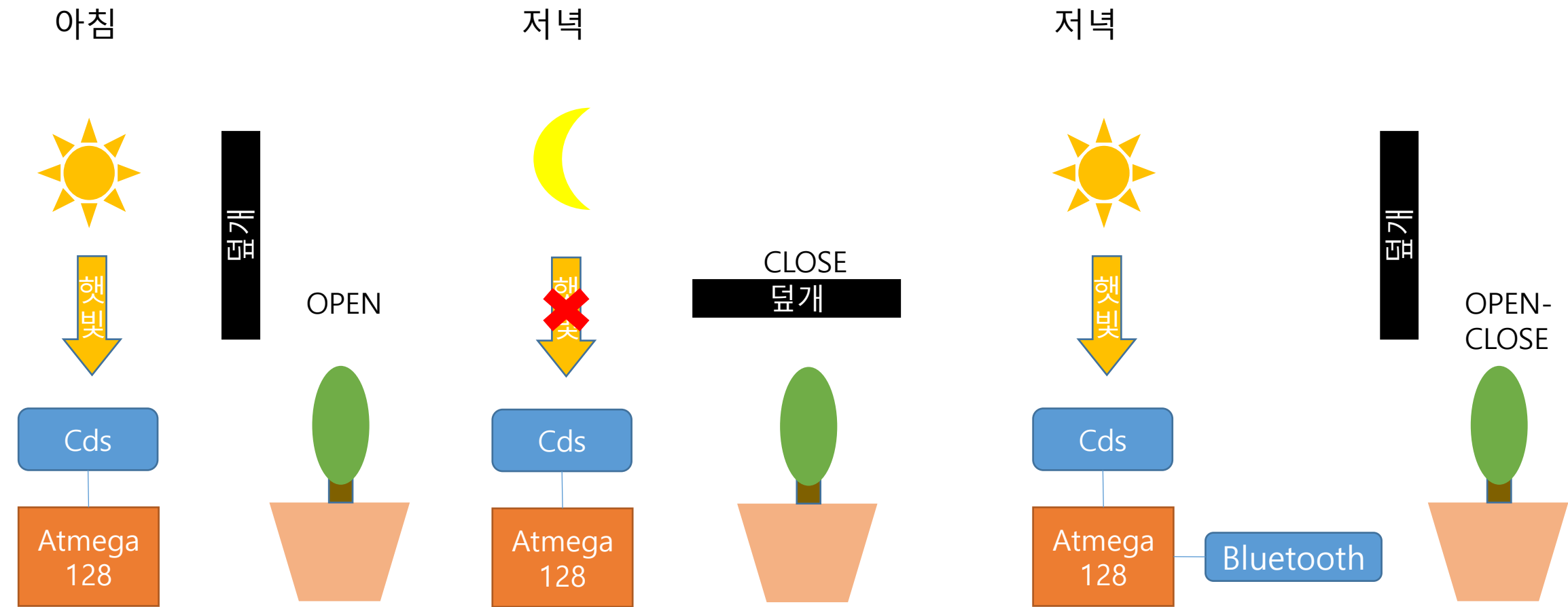
| 07 개발 코드 분석

팀 정보

	이름	학 교	전 공	학 년	E-mail	전화번호
팀장	유민상	가천대학교	전자공학과	2		
팀원1	윤재민	가천대학교	전자공학과	2		
팀원2	이우석	가천대학교	컴퓨터공학과	3		

작품 개요

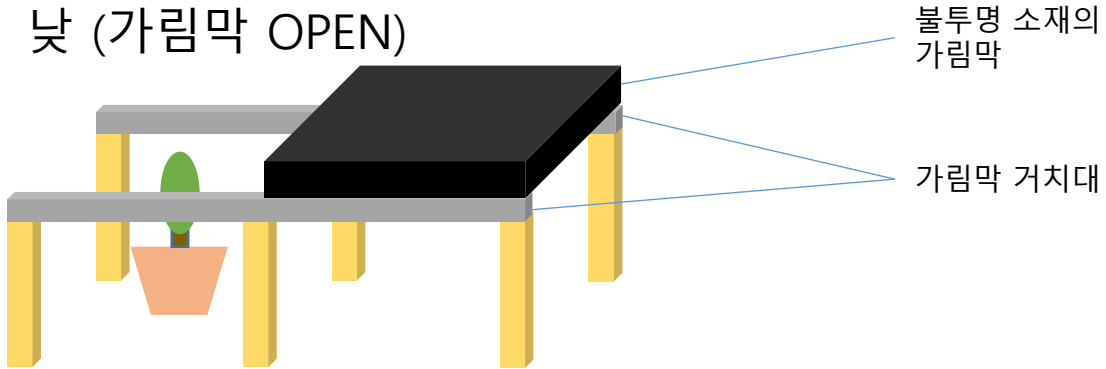
❖ 작품에 대한 전체적인 설명



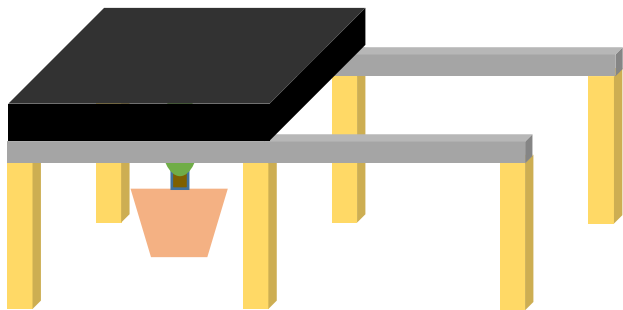
디자인

- ❖ 작품의 디자인 형태 및 컨셉
- ❖ 작품의 디자인 특성 및 이유 상세히 설명
ex. 착용의 편리함/심미성/착용감/소재의 특성 등

낮 (가림막 OPEN)



저녁 (가림막 CLOSE)



사용 재료

제품명	수량
황화카드뮴(Cds)	1
스텝 모터	1
모터드라이버	1
브레드보드	1
블루투스 모듈	1
저항 (10K)	1

개발 상황

- ❖ 스텝 모터와 Cds를 이용한 조광정도에 따른 자동개폐시스템
- ❖ UART 통신과 Appinventor를 이용한 수동개폐시스템

기술적 요구사항

- ❖ 1상 여자 방식을 사용한 스텝 모터 제어
- ❖ ADC를 통해 조광세기를 정함
- ❖ 조광세기에 level을 부여해 조광정도에 따라 개폐정도 결정
- ❖ Interrupt를 이용해 리모컨으로부터 신호 수신
- ❖ Uart 통신을 통한 블루투스 모듈 제어
- ❖ Appinventor를 사용한 리모컨 제작

기술적 문제점 및 해결

❖ 스텝 모터의 토크가 매우 약해 개폐시스템 작동 불가

- 기어비를 1:10으로 낮춰 적은 힘으로도 작동 가능하게 만들
- 기어비가 낮아진 탓에 작동시간이 10배로 증가

개발 코드 분석

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void UART1_init(void)
{
    UCSR1A |= (1 << U2X1); // 2배속 모드
    UCSR1B |= (1 << RXEN1) | (1 << TXEN1); // 송수신 활성화
    UCSR1C |= (1 << UCSZ11) | (1 << UCSZ10); // 8비트
    UBRR1H = 0;
    UBRR1L = 207; // 보율 9600
}

void UART1_transmit(char data)
{
    while( !(UCSR1A & (1 << UDRE1))); // 송신 가능 대기

    UDR1 = data; // 데이터 전송
}
```

```
int level;
int mode=0;
int open=3;

int level;
int initial_level=0;
int level_gap;

uint8_t step_data[]={0x01, 0x02, 0x04, 0x08};
int step_index=-1; // 현재상태

uint8_t stepForward(void) // 시계방향 회전
{
    step_index++;
    if(step_index>=4) step_index=0;

    return step_data[step_index];
}

uint8_t stepBackward(void) // 반시계방향 회전
{
    step_index--;
    if(step_index<0) step_index=3;

    return step_data[step_index];
}
```

개발 코드 분석

```
void movement(void)
{
    level_gap = level-initial_level;
    initial_level=level;

    if(level_gap >0)
    {
        for(level_gap; level_gap >0; level_gap --)
        {
            for(int i=0; i<5000; i++)
            {
                PORTC=stepForward();
                _delay_ms(5);
            }
        }
    }
    else
    {
        for(level_gap; level_gap <0; level_gap ++ )
        {
            for(int i=0; i<5000; i++)
            {
                PORTC=stepBackward();
                _delay_ms(5);
            }
        }
    }
    _delay_ms(200);
}
```

```
int read_ADC(void)
{
    while( ! ( ADCSRA & (1 << ADIF) ) );
    return ADC;
}

ISR(USART1_RX_vect)//UART1 수신시 인터럽트 발생
{
    char temp=UDR1;

    if(temp=='0') //자동모드
    {
        mode=0;
    }

    else if(temp=='1') //수동모드
    {
        mode=1;
    }

    else if(temp=='2') //열림
    {
        if(mode==1)
            open=1;
    }

    else if(temp=='3') //닫힘
    {
        if(mode==1)
            open=0;
    }
}
```

개발 코드 분석

```
int main(void)
{
    /* Replace with your application code */

    UCSR1B=(1<<RXIE1); //수신 완료 인터럽트 발생 허용 레지스터
    sei();

    int read;

    DDRB = 0xFF;
    DDRC = 0x0F;

    ADMUX |= (1 << REFS0) | (1 << MUX0 | 1 << MUX1);
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
    ADCSRA |= (1 << ADEN);
    ADCSRA |= (1 << ADFR);
    ADCSRA |= (1 << ADSC);

    UART1_init();
```

```
while (1)
{
    while(mode==0)
    {
        read = read_ADC();

        switch(read/100)
        {
            case 9 : PORTB = 0b11111110; level=9; break;
            case 8 : PORTB = 0b11111111; level=8; break;
            case 7 : PORTB = 0b01111111; level=7; break;
            case 6 : PORTB = 0b00111111; level=6; break;
            case 5 : PORTB = 0b00011111; level=5; break;
            case 4 : PORTB = 0b00001111; level=4; break;
            case 3 : PORTB = 0b00000111; level=3; break;
            case 2 : PORTB = 0b00000011; level=2; break;
            case 1 : PORTB = 0b00000001; level=1; break;
            default : PORTB = 0b00000000; level=0; break;
        }

        _delay_ms(200);

        movement();
```

개발 코드 분석

```
if(mode==1)//mode 1일 경우 루프 탈출
```

```
{  
  PORTB = 0b10101010;  
  _delay_ms(500);  
  PORTB = 0b01010101;  
  _delay_ms(500);  
  PORTB = 0b10101010;  
  _delay_ms(500);  
  PORTB = 0b01010101;  
  _delay_ms(500);  
}
```

```
break;  
}
```

```
}
```

```
while(mode==1)  
{
```

```
  if(open==0)  
  {  
    PORTB = 0b00000001;  
    level=1;
```

```
    movement();  
  }
```

```
  else if(open==1)  
  {  
    PORTB = 0b11111110;  
    level=9;
```

```
    movement();  
  }
```

```
  else if(open==3)  
  {  
    movement();  
  }
```

```
  if(mode==0)//mode 0일 경우 루프 탈출
```

```
  {  
    PORTB = 0b10101010;  
    _delay_ms(500);  
    PORTB = 0b01010101;  
    _delay_ms(500);  
    PORTB = 0b10101010;  
    _delay_ms(500);  
    PORTB = 0b01010101;  
    _delay_ms(500);  
  }
```

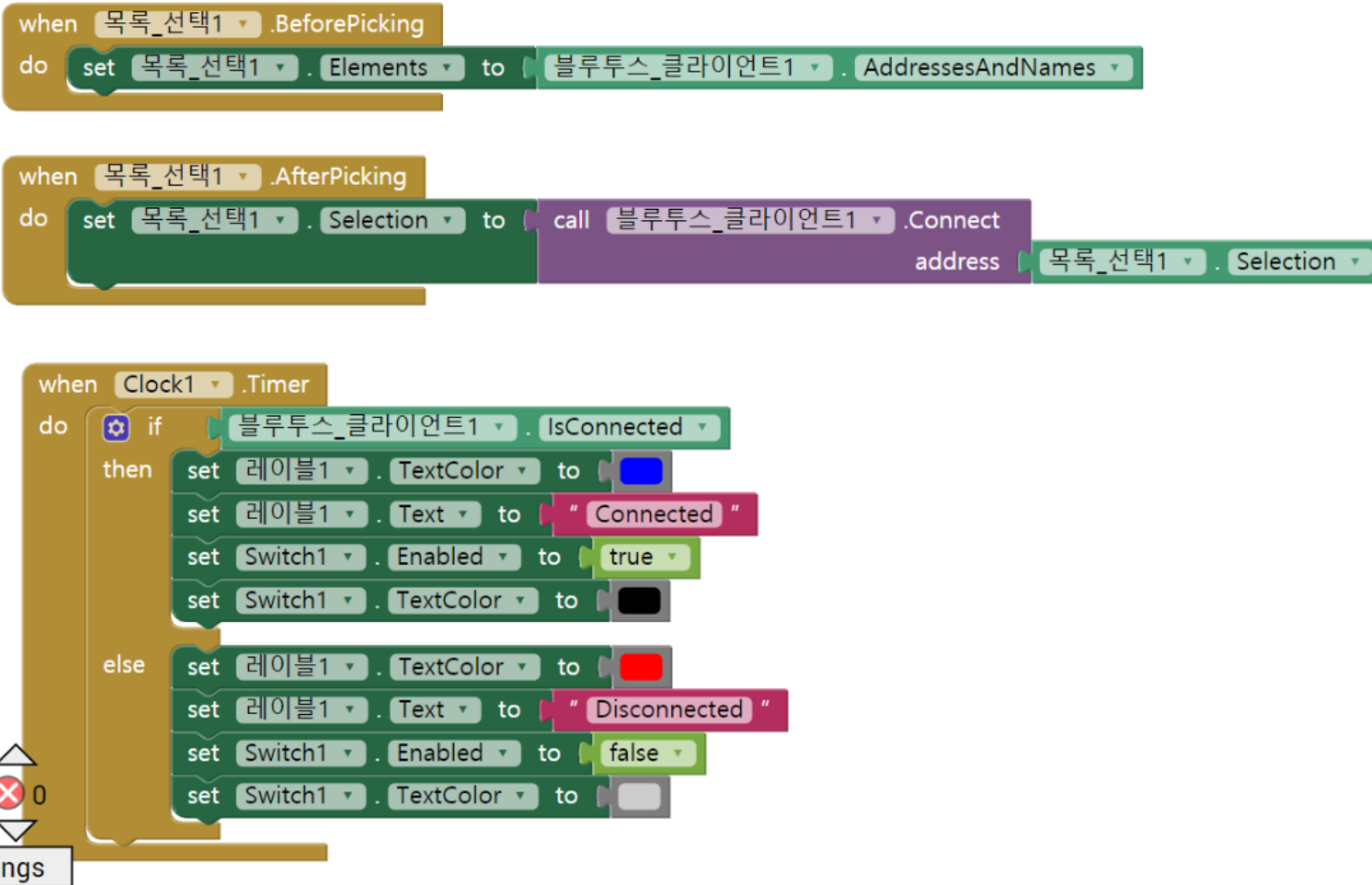
```
  break;  
}
```

```
}
```

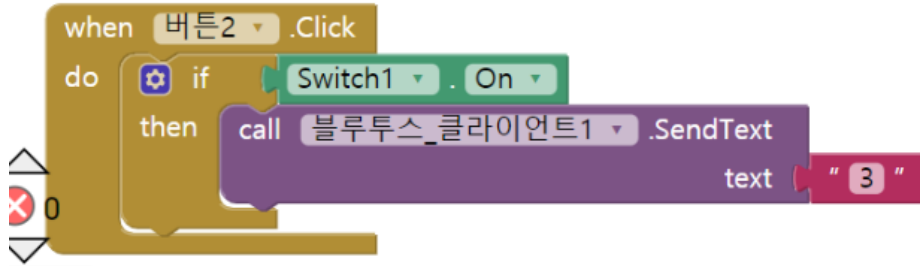
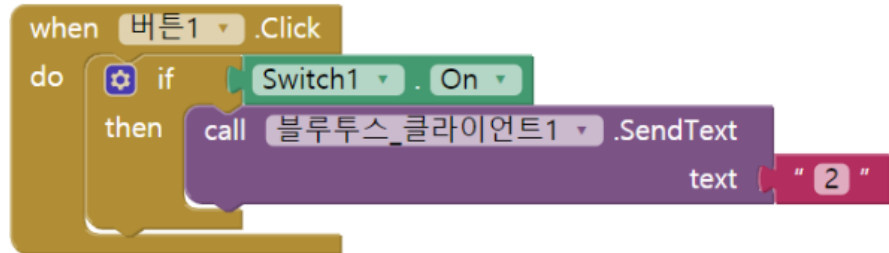
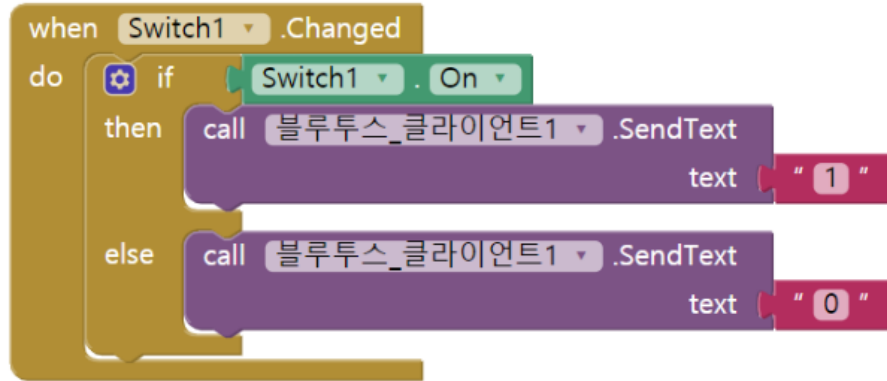
```
return 0;
```

```
}
```

개발 코드 분석 - Appinventor



개발 코드 분석 - Appinventor



추가 개발 계획

- ❖ 수동 작동으로 개폐 level 설정 가능
- ❖ 현재는 조광 세기가 강할수록 개폐 level이 높아짐
- ❖ 리모컨으로 조광 세기가 강할수록 개폐 level이 낮아질 수 있도록 하는 기능 추가