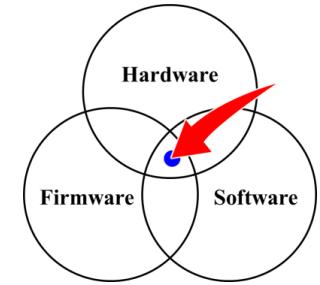


PORTFOLIO



새로운 도전에 주저하지 않는 개발자 김성호입니다.

이 름	김 성 호
이 메 일	Kims9872@naver.com
전화번호	010-2920-4176

목 차

▶ 자기소개

※ 명칭/페이지 번호를 누르면 해당 페이지로 이동

1. 개발자 김성호란? p4
2. 수행한 주요 프로젝트 p6

▶ 수행한 프로젝트(세부내용)

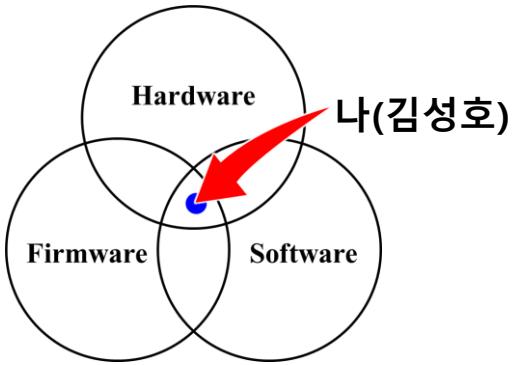
H/W - 검정색, F/W – 빨간색, S/W - 파란색

1. 햅틱 에어 마우스 H/W(p11~14), F/W(p14)
2. VR AR 콘텐츠 제작을 위한 다형성 햅틱 컨트롤러 H/W(p18~19), F/W(p19~23), S/W(p24~25)
3. 다중초점 블랙박스 시스템 개발 H/W(p28), F/W(p29), S/W(p30)
4. 고전압 스위칭 회로 개발 H/W(p33), F/W(p34), S/W(p35)
5. 필름형 고신축성 3D 터치 센서 개발 F/W(p38), S/W(p39)
6. VR 낚시 레저활동을 위한 액추에이터 F/W(p42)
7. HD 촉감 기술 기반 초실감 콘텐츠 재현 기술 개발 H/W(p45~46)

자기소개

1. 개발자 김성호란?

1. 개발자 김성호란?



이름	김성호
최종학력	석사 졸업
전공	컴퓨터공학/ HCI(haptics)
주요 성과	경쟁형 프로젝트(31억) 연구지속, 특히 등록 15건 및 출원 5건, 국내외학술대회 2건, KCI 1건 등
사용 프로그래밍 언어	C/C++, C#, MATLAB 등
개발환경	Visual studio, IAR workbench, Unity 등
자격증	정보처리기사, ISTQB, 1종보통

나의 강점

다양한 프로젝트 경험을 통해
복합적 개발 역량 향상

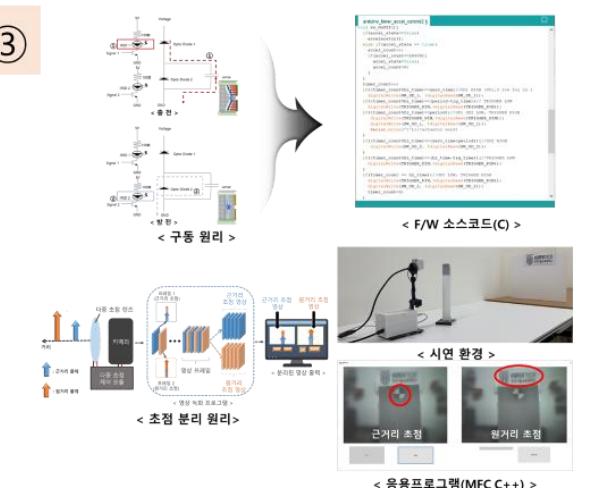
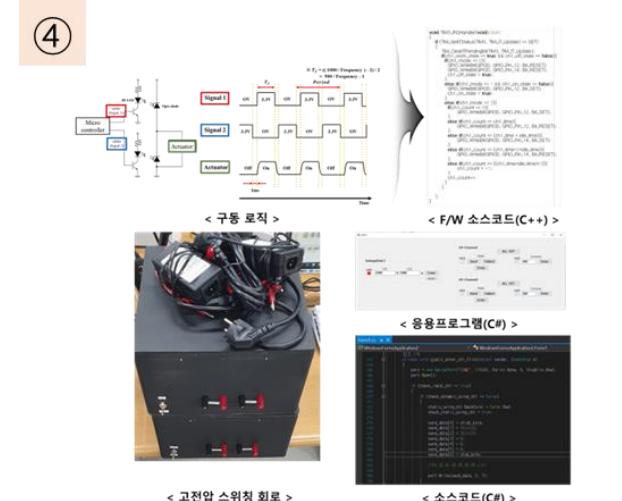
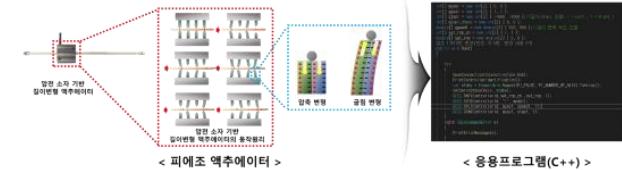
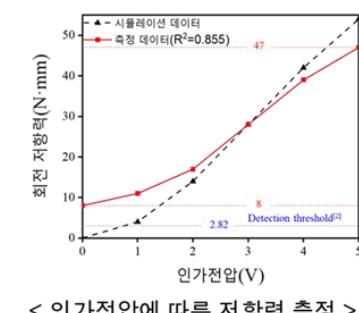
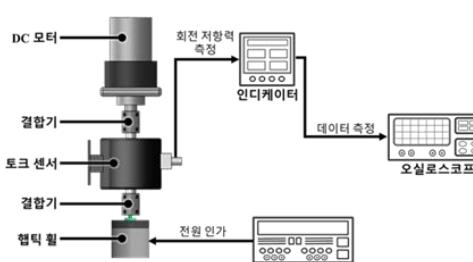
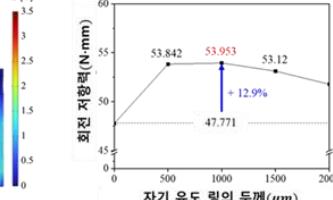
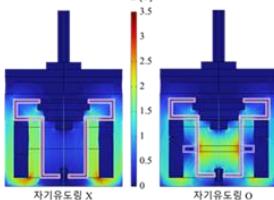
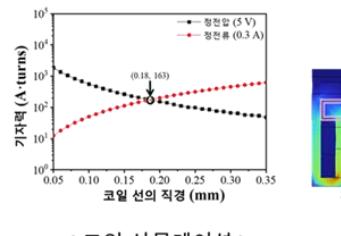
업무에서 가치

1. 타부서와 커뮤니케이션에서 효과적
2. 다양한 관점에서 문제 해결
3. H/W에 대한 이해를 바탕으로 S/W 개발
업무에서 빠른 투입 가능
4. 새롭고 복합적인 R&D 아이디어 제안 가능

3가지 분야(H/W, F/W, S/W)에 대한 지식 및 개발 역량을 갖춘 개발자

2. 수행한 주요 프로젝트

2. 수행한 주요 프로젝트



- ① 햅틱 에어마우스 ② 다형성 햅틱 컨트롤러 ③ 블랙박스 시스템 ④ 고전압 스위칭 회로 ⑤ 3D 터치 센서

< 수행한 주요 프로젝트 >

수행한 프로젝트

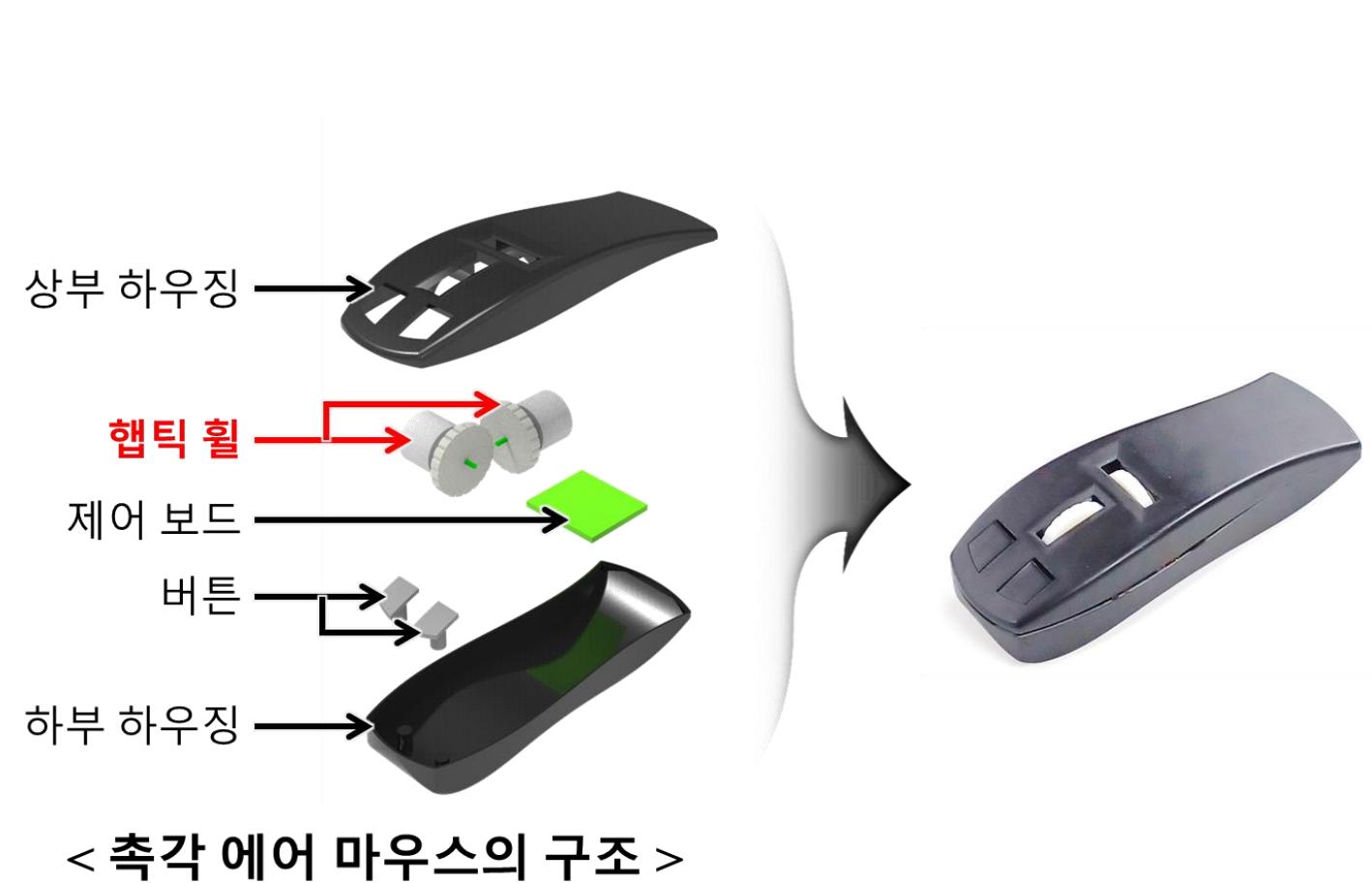
(세부내용)

1. 햅틱 에어 마우스

1. 햅틱 에어 마우스 – 개요

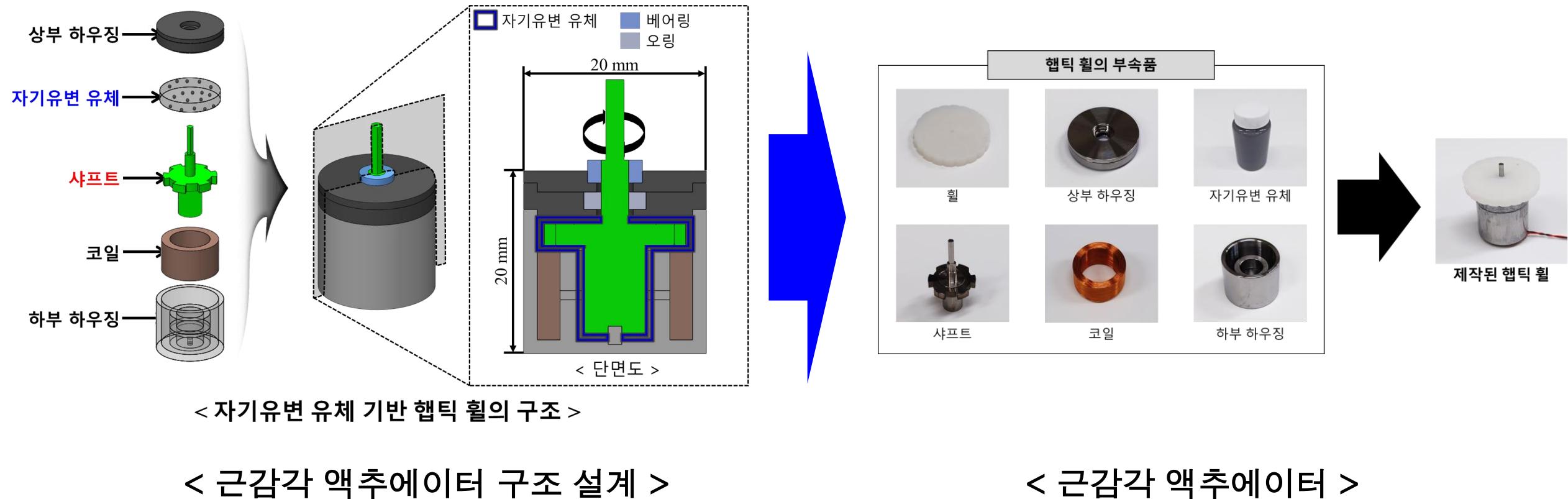
프로젝트 소개

작업 기간	2022. 01 ~ 2022. 12 (12개월)
인력 구성	석사과정 1명(본인)
프로젝트 목적	입체적인 3차원 상호 작용이 가능하며, 근감각을 줄 수 있는 촉각 에어 마우스 개발
프로젝트 내용	<ol style="list-style-type: none">근감각 액추에이터 개발촉각 에어 마우스 제작
주요 업무 및 상세 역할	<ol style="list-style-type: none">근감각 액추에이터 구조 설계 및 시뮬레이션구동 회로 설계 및 F/W 개발촉각 에어 마우스 하우징 설계
사용언어 및 개발 환경	C, C++ / Visual studio, Solidworks, IAR workbench



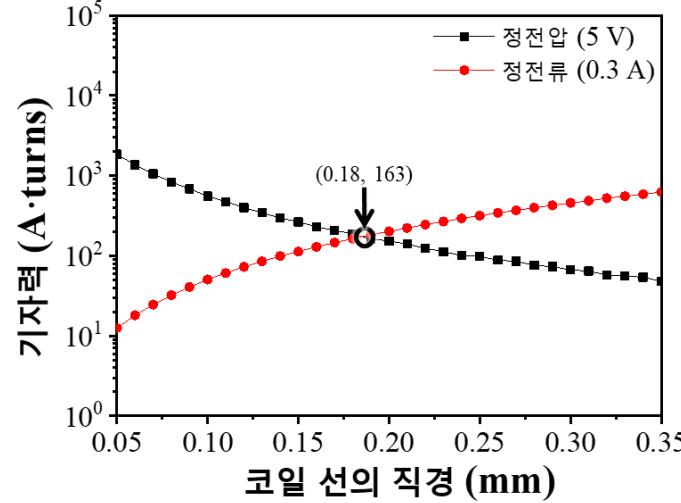
< 촉각 에어 마우스의 구조 >

1. 햅틱 에어 마우스 – H/W (근감각 액추에이터 설계)

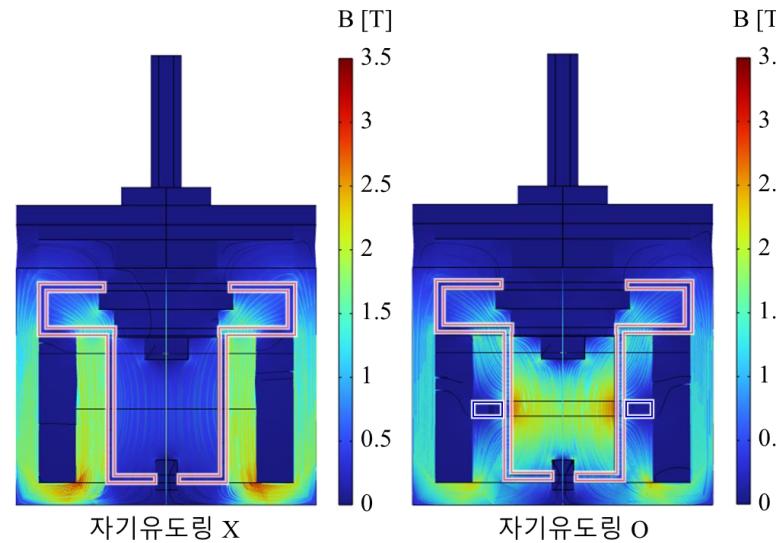


- Solidworks를 이용하여 passive motor인 근감각 액추에이터 구조 설계를 진행함
- 시뮬레이션을 통해 최적 성능을 가지는 구조를 찾고, 이를 바탕으로 햅틱 휠을 제작하였음

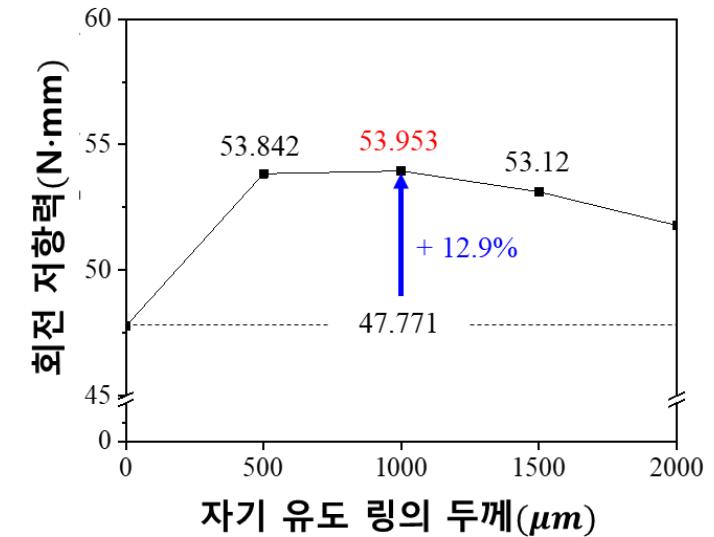
1. 햅틱 에어 마우스 – H/W (시뮬레이션)



< 코일 시뮬레이션 >



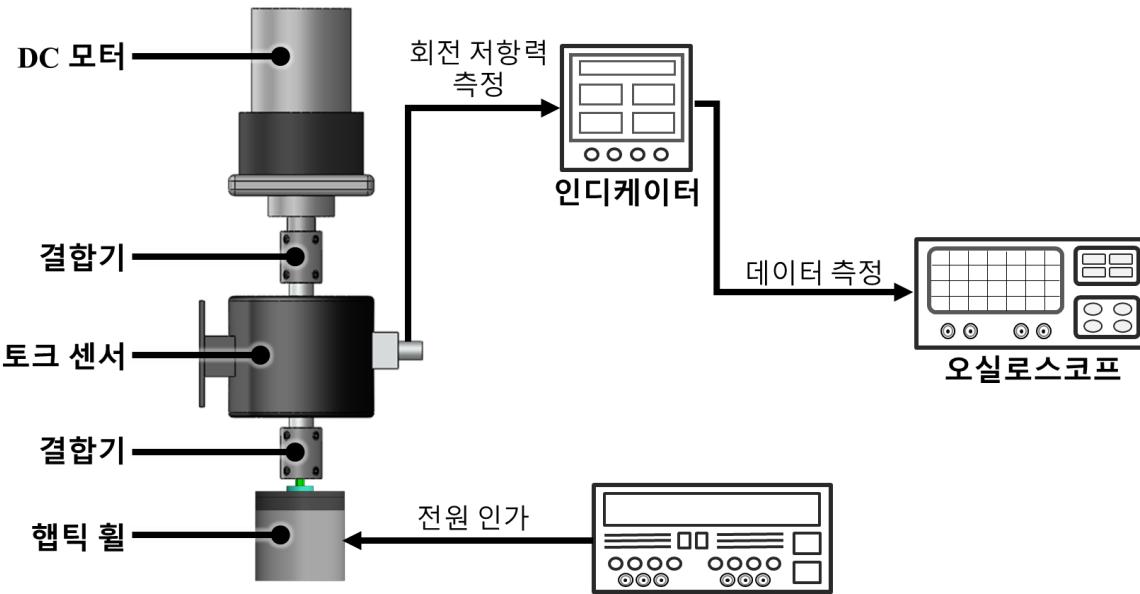
< 자속밀도 시뮬레이션 >



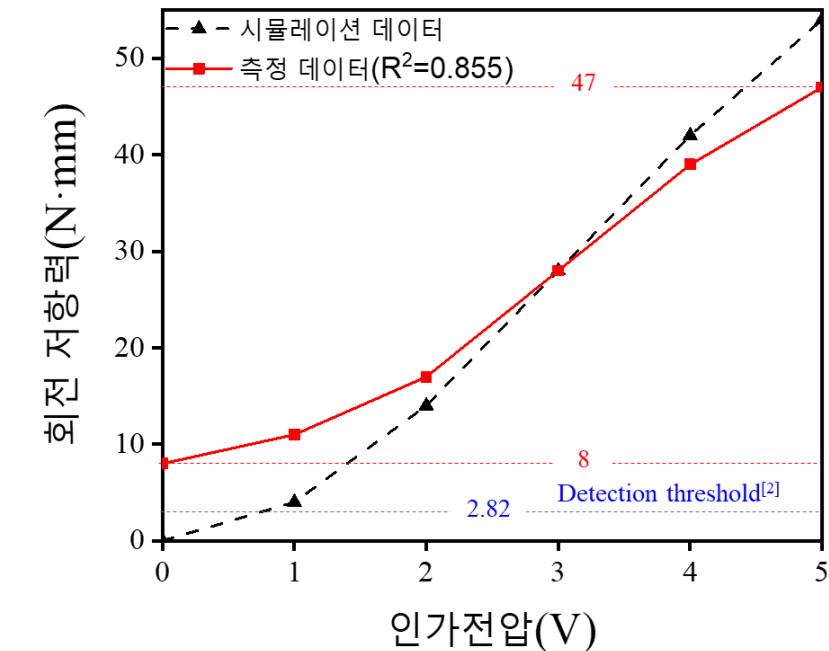
< 저항력 시뮬레이션 >

- ▶ 근감각 액추에이터의 최적 설계를 위해 MATLAB, COMSOL 을 이용하여 코일 시뮬레이션, 자속밀도 시뮬레이션, 저항력 시뮬레이션을 진행함

1. 햅틱 에어 마우스 – H/W (저항력 측정)



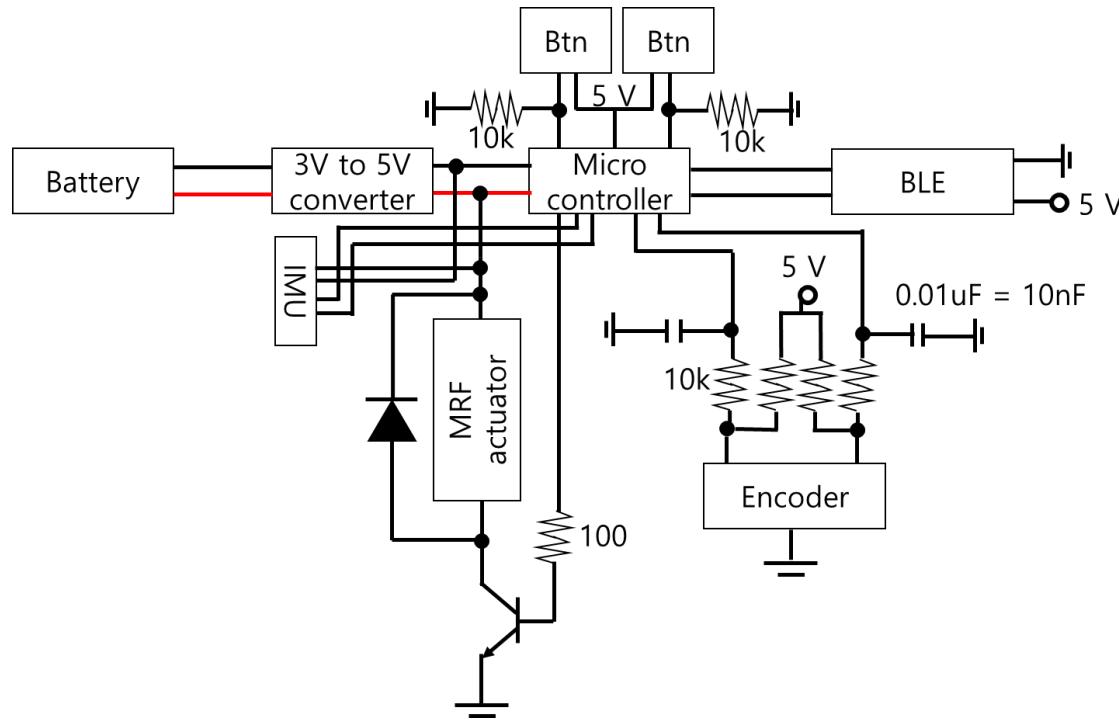
< 저항력 측정 실험 환경 >



< 인가전압에 따른 저항력 측정 >

- 근감각 액추에이터의 저항력 측정 실험 환경을 구축하고, 인가 전압에 따른 저항력을 측정함

1. 햅틱 에어 마우스 – 회로 설계 및 F/W 개발



< 회로 구성 >

```
***** Knob 2 Vibration(rdata: a) *****
if(state_knob2 == true){
    if(count_knob2 == time_knob2){
        digitalWrite(knob2, true);
    }
    else if(count_knob2 == time_knob2 * 2){
        digitalWrite(knob2, false);
        count_knob2 = 0;
    }
    count_knob2++;
}

***** Encoder 1(60Hz) *****
// if(state_encoder1 == true){
//     bsel.dataA = cse1.dataA;
//     bsel.dataB = cse1.dataB;
//     CheckBeforeStateEncoder1(bsel.dataA, bsel.dataB);
//     csel.dataA = digitalRead(enco1);
//     csel.dataB = digitalRead(enco2);
//     CheckCurrentStateEncoder1(cse1.dataA, cse1.dataB);
//     setRotationDirection1();
//     if(rotateCW1 == true){
//         knob_buf = 'J';
//         rotateCW1 = false;
//     }
//     else if(rotateCCW1 == true){
//         knob_buf = 'K';
//         rotateCCW1 = false;
//     }
// }
```

< 소스 코드>

➤ 기능

- 햅틱 휠의 회전 움직임을 encoder로 측정하여 데이터를 Bluetooth를 통해 PC로 전달

➤ 작업 내용

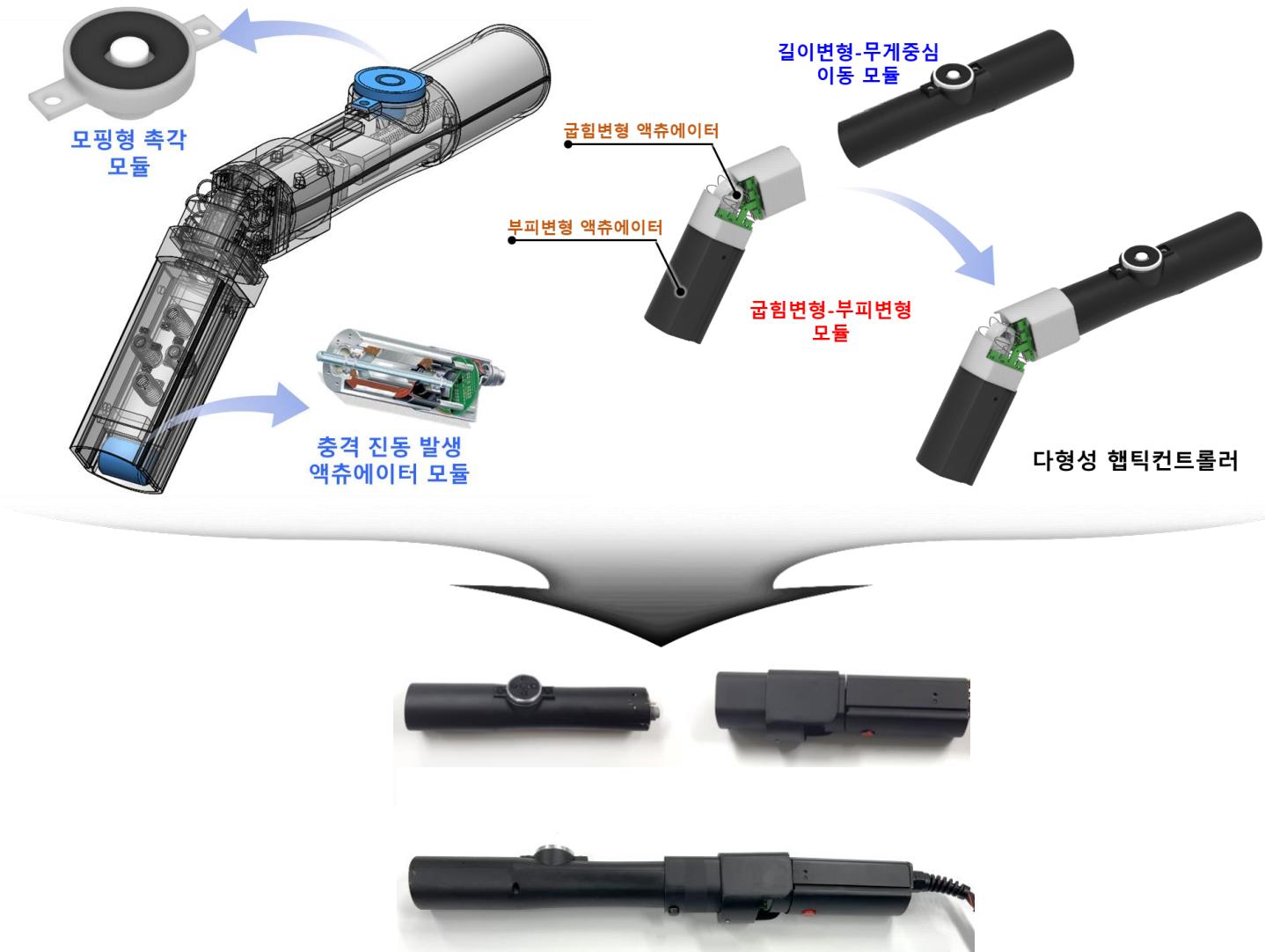
- 햅틱 휠의 rotation을 encoder로부터 수신받고, rotation 정보를 송신하는 코드 작성
- 사용자가 햅틱 휠을 회전시킬 때 발생하는 순간적인 역회전을 감지하지 못하도록 코드 작성함

2. VR·AR 콘텐츠 제작을 위한 다형성 햅틱 컨트롤러

2. 다형성 햅틱 컨트롤러 – 개요

프로젝트 소개

작업 기간	2020. 09 ~ 2022. 12 (28개월)
인력 구성	박사 1명, 박사과정 1명, 석사과정 2명(본인 포함)
프로젝트 목적	VR AR 콘텐츠 제어를 위한 다형성(길이변형, 무게중심 이동, 굽힘변형, 부피변형) 햅틱 컨트롤러 개발
프로젝트 내용	<ol style="list-style-type: none">다형성 햅틱 컨트롤러 개발모핑형 구동기 개발길이변형, 굽힘변형 등에 대한 사용자 인지 실험VR 애플리케이션 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">다형성 햅틱 컨트롤러 설계구동 회로 설계 및 F/W 개발굽힘, 부피변형 액추에이터 개발VR 애플리케이션 개발 및 컨트롤러 연동부품 구입 및 기타 사무 처리
사용언어 및 개발 환경	C, C++, C# / Visual studio, Unity, Solidworks, IAR workbench, Git



< 제작된 다형성 햅틱 컨트롤러 >

2. 다형성 햅틱 컨트롤러 – 적용 애플리케이션



< VR 복강경 수술 시뮬레이터>



< VR 화재 진압 시뮬레이터>



< VR 목공 시뮬레이터>



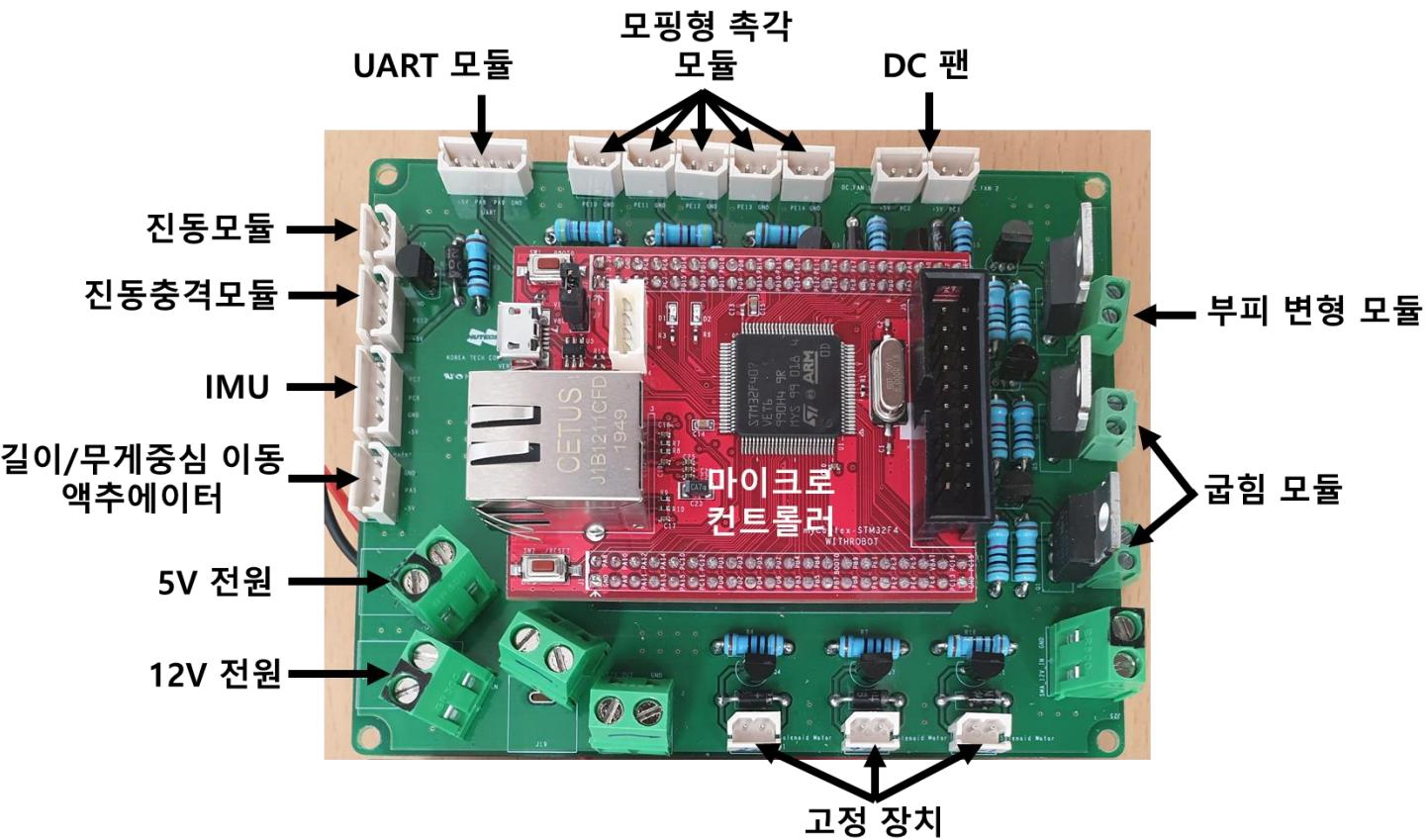
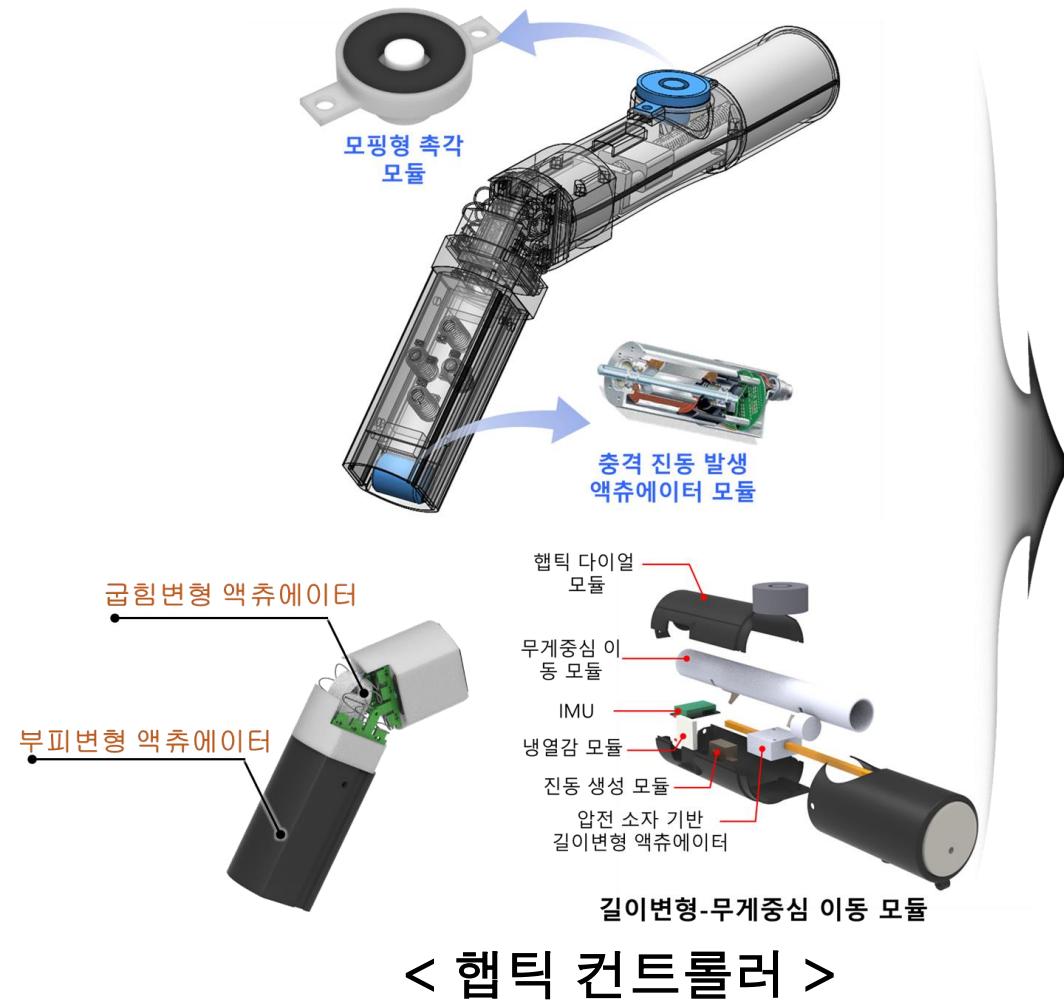
< VR 골프>



< VR 두더지잡기>

➤ 개발된 다형성 햅틱 컨트롤러는 여러 학교에서 개발한 VR 애플리케이션과 연동을 진행하였음

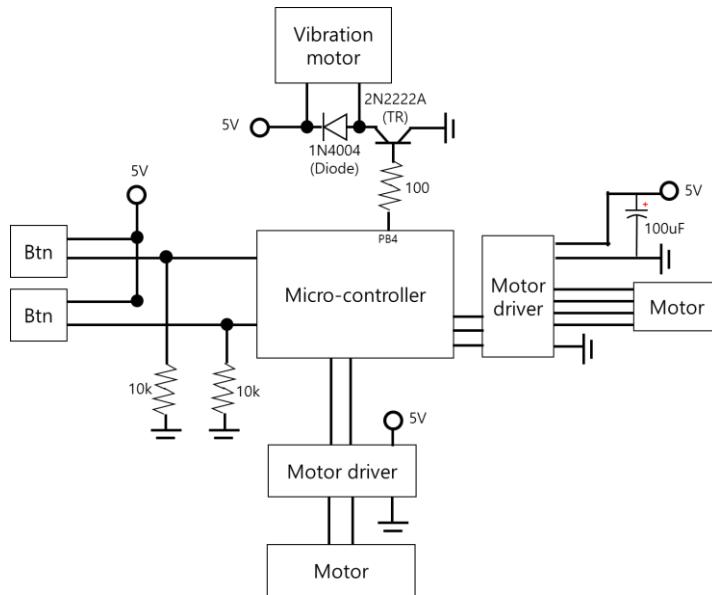
2. 다형성 햅틱 컨트롤러 - H/W (구동 회로 설계)



< 구동 회로 >

- ▶ 다형성 햅틱 컨트롤러의 부품을 제어하기 위한 회로 설계 및 개발 진행함
- ▶ 형상기억합금을 사용하는 굽힘변형, 부피변형 액추에이터는 5A까지 전류를 제한하였음

2. 다형성 햅틱 컨트롤러 – 소형화(회로 설계, F/W)



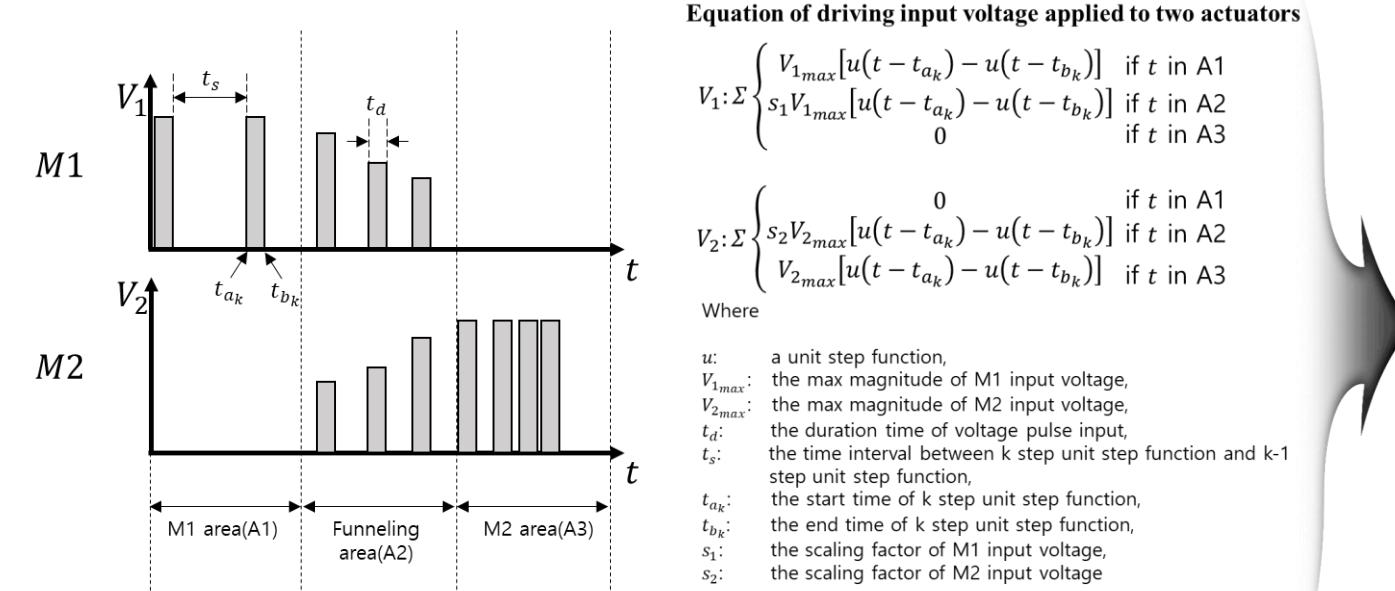
< 회로 구성 >

- 다형성 햅틱 컨트롤러 소형화 프로토 타입 개발을 위해 회로 설계, 펌웨어 개발 진행함
- 2개의 버튼이 동시에 눌렸을 때, debounce에 의한 노이즈를 제거하는 코드를 추가하였음

```
else if(button_count_2 > 4 && print_button == false && before_button_state!=3 && cur_button_state==2){  
    if(initial_button_active == false){  
        UART1Print("D");  
        initial_button_state = 2;  
        initial_button_active = true;  
        print_button = true;  
    }  
    // When we don't touch the button after touching button  
    if(print_button == true && before_button_state!=cur_button_state ){//A  
        if(initial_button_state == 1 && cur_button_state==0 && cur_button_state!=3 && cur_button_state!=2){  
            UART1Print("x");  
            button_count_1 = 0;  
            print_button = false;  
            initial_button_active = false;  
        }  
        else if(initial_button_state == 2 && cur_button_state==0 && cur_button_state!=3 && cur_button_state!=1 ){//D  
            UART1Print("y");  
            button_count_2 = 0;  
            print_button = false;  
            initial_button_active = false;  
        }  
        else if(initial_button_state == 3&& cur_button_state==0 && (cur_button_state!=1 && cur_button_state!=2)){//S  
            UART1Print("z");  
            button_count_3 = 0;  
            print_button = false;  
            initial_button_active = false;  
        }  
    }  
    before_button_state = cur_button_state;
```

< F/W 코드 일부 >

2. 다형성 햅틱 컨트롤러 – F/W (이동진동파)



The input signal of vibration motor for TVW

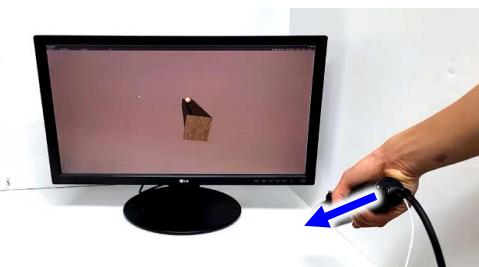
< 이동 진동파 원리>

```
void TVW_On() { // 이동진동파 제어 부분
    if (vib_dir == 1){
        virtual_obj_pos += temp_velo / 1000.0;

        if (virtual_obj_pos >= vib_distance) // Arrived in target point
            End_Move();
        if ((virtual_obj_pos - virtual_obj_prev_pos) >= temp_step) // virtual object pass through the temporary point
            virtual_obj_prev_pos = virtual_obj_pos;
        vib_1_amp -= 10;
        vib_2_amp += 10;
        burst_state = true;
    }
    if (burst_state) {
        if (burst_cnt < burst_time) {
            if (virtual_obj_pos < vib_distance * 1 / 10) {
                vib_1_amp = max_amp;
                vib_2_amp = min_amp;
                vib_state = true;
            }
            else if (virtual_obj_pos < vib_distance * 9 / 10) {
                if (vib_1_amp <= min_amp)
                    vib_1_amp = min_amp;
                if (vib_1_amp >= max_amp)
                    vib_1_amp = max_amp;
                if (vib_2_amp >= max_amp)
                    vib_2_amp = max_amp;
                if (vib_2_amp <= min_amp)
                    vib_2_amp = min_amp;
            }
            else if (virtual_obj_pos >= vib_distance) {
                vib_1_amp = min_amp;
                vib_2_amp = max_amp;
            }
        }
    }
}
```

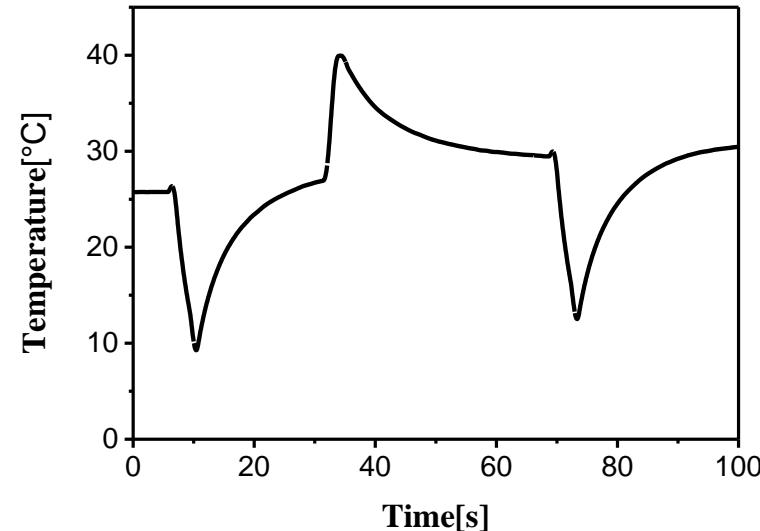
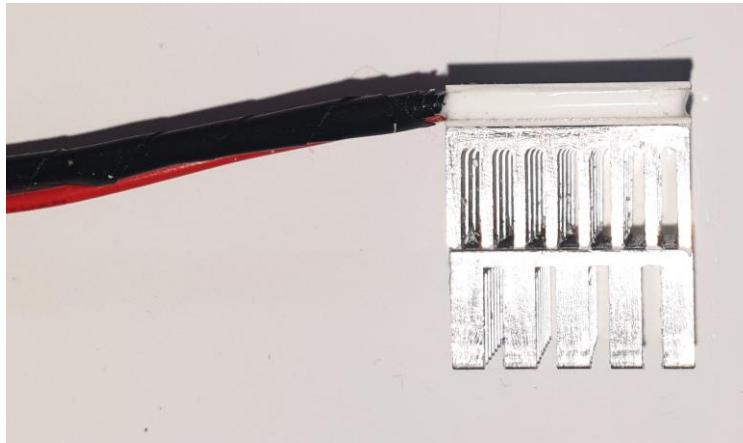
< 소스 코드>

→ : 진동 방향



- 기능
 - 컨트롤러 기울기에 따른 진동의 이동 표현
- 작업 내용
 - 이동진동 원리에 따라 2개의 진동모터로 진동의 이동을 표현할 수 있도록 구현

2. 다형성 햅틱 컨트롤러 – F/W (펠티어소자)



< 펠티어 소자 >

< 시간에 따른 펠티어 소자 온도 변화 >

< 소스 코드 >

```
if (TIM_GetITStatus(TIM3, TIM_IT_Update) == SET)
{
    TIM_ClearITPendingBit(TIM3, TIM_IT_Update);

    pelLcntTime++;
    //*****Pelteir 구동 영역 시작*****
    if(+(pelCntrPeriode == pelPeriode) { //펠티어 구동 주기
        adcVal1 = readPeltier1TempADCVal();
        adcVal2 = readPeltier2TempADCVal();
        currentSet[0][0] = (double)readMotor1CSADCVal() / 4095 / 0.14 * 3;
        currentSet[1][0] = (double)readMotor2CSADCVal() / 4095 / 0.14 * 3;

        //현재 온도 구하기
        curTemp1 = calTempFromThermistor(DEFAULT_ADC_MAX, adcVal1, DEFAULT_TO, DEFAULT_RO, DEFAULT_B, -1, DEFAULT_R2)+9.5;
        curTemp2 = calTempFromThermistor(DEFAULT_ADC_MAX, adcVal2, DEFAULT_TO, DEFAULT_RO, DEFAULT_B, -1, DEFAULT_R2);

        //PID 값 구하기 <-- 현재 온도 입력
        PIDVal1 = PLControl(pVal[0].targetTemp, curTemp1, pelFreq, &(pVal[0].pidData));
        PIDVal2 = PL_control(pVal[1].targetTemp, curTemp2, pelFreq, &(pVal[1].pidData));
        //PID 영향 조절 : PID제어 절대값이 PID_Percent% CI 정도로
        if(PIDVal1 > PID_Percent) { PIDVal1 = PID_Percent; }
        if(PIDVal1 < -PID_Percent) { PIDVal1 = -PID_Percent; }

        // If(PIDVal2 > PID_Percent) { PIDVal2 = PID_Percent; }
        // If(PIDVal2 < -PID_Percent) { PIDVal2 = -PID_Percent; }

        // [DEBUG] 목표 온도 도달 시간 측정 (오차 범위 이내 들어올 경우 종료)
        if(pVal[0].toggleReachedTargetTemp == 0) {
            if(isReachedTargetTemp(pVal[0].targetTemp, curTemp1, 1.0) == 0) {
                pVal[0].cntTime++;
            }
            else {
                pVal[0].toggleReachedTargetTemp = 1;
            }
        }
    }
}
```

➤ 기능

- PID 제어를 통해 펠티어 소자 제어(통각이 아닌 냉열감을 구현)
- 펠티어 소자의 쿨링 방법 중 공압, 유압 방식 적용 불가에 따른 펠티어 소자 구동 시간에 따른 쿨링 제어

➤ 작업 내용

- 사용자가 냉열감을 느낄 수 있는 온도와 시간을 설정하여 쿨링 시간을 감소(>40s → 약 20s)
- 펠티어 소자의 쿨링 시간을 줄이기 위해 2개의 방열판 사용(하우징 크기 제한 고려)

2. 다형성 햅틱 컨트롤러 – F/W(햅틱 디이얼)



< 햅틱 디이얼 사용 예시>

➤ 기능

- 햅틱 디이얼의 회전 움직임을 encoder로 측정하여 데이터를 알카노이드 애플리케이션으로 전달

➤ 작업 내용

- 햅틱 디이얼의 회전 움직임을 encoder로부터 수신받고, 회전 정보를 송신하는 코드 작성
- 사용자가 햅틱 디이얼을 회전시킬 때 발생하는 순간적인 역회전을 감지하지 못하도록 코드 작성함

```
//----- Encoder -----  
if(state_encoder == true){  
    bse.dataA = cse.dataA;  
    bse.dataB = cse.dataB;  
    CheckBeforeStateEncoder(bse.dataA, bse.dataB);  
    cse.dataA = GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_8);  
    cse.dataB = GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_9);  
    CheckCurrentStateEncoder(cse.dataA, cse.dataB);  
    setRotationDirection();  
    if(rotateCW == true){  
        UART1Print("U");  
        rotateCW = false;  
    }  
    else if(rotateCCW == true){  
        UART1Print("K");  
        rotateCCW = false;  
    }  
  
void CheckBeforeStateEncoder(bool dataA, bool dataB){ //CW 방향  
    if(dataA == false && dataB == false)  
        state_before = 1;  
    else if(dataA == true && dataB == false)  
        state_before = 2;  
    else if(dataA == true && dataB == true)  
        state_before = 3;  
    else if(dataA == false && dataB == true)  
        state_before = 4;  
}  
  
void CheckCurrentStateEncoder(bool dataA, bool dataB){ //CW 방향  
    if(dataA == false && dataB == false)  
        state_current = 1;  
    else if(dataA == true && dataB == false)  
        state_current = 2;  
    else if(dataA == true && dataB == true)  
        state_current = 3;  
    else if(dataA == false && dataB == true)  
        state_current = 4;  
}  
  
void setRotationDirection(){  
    if(state_current - state_before == 1 || (state_current - state_before == -3))  
        rotateCW = true;  
    else if(state_current - state_before == -1 || (state_current - state_before == 3))  
        rotateCCW = true;  
}
```

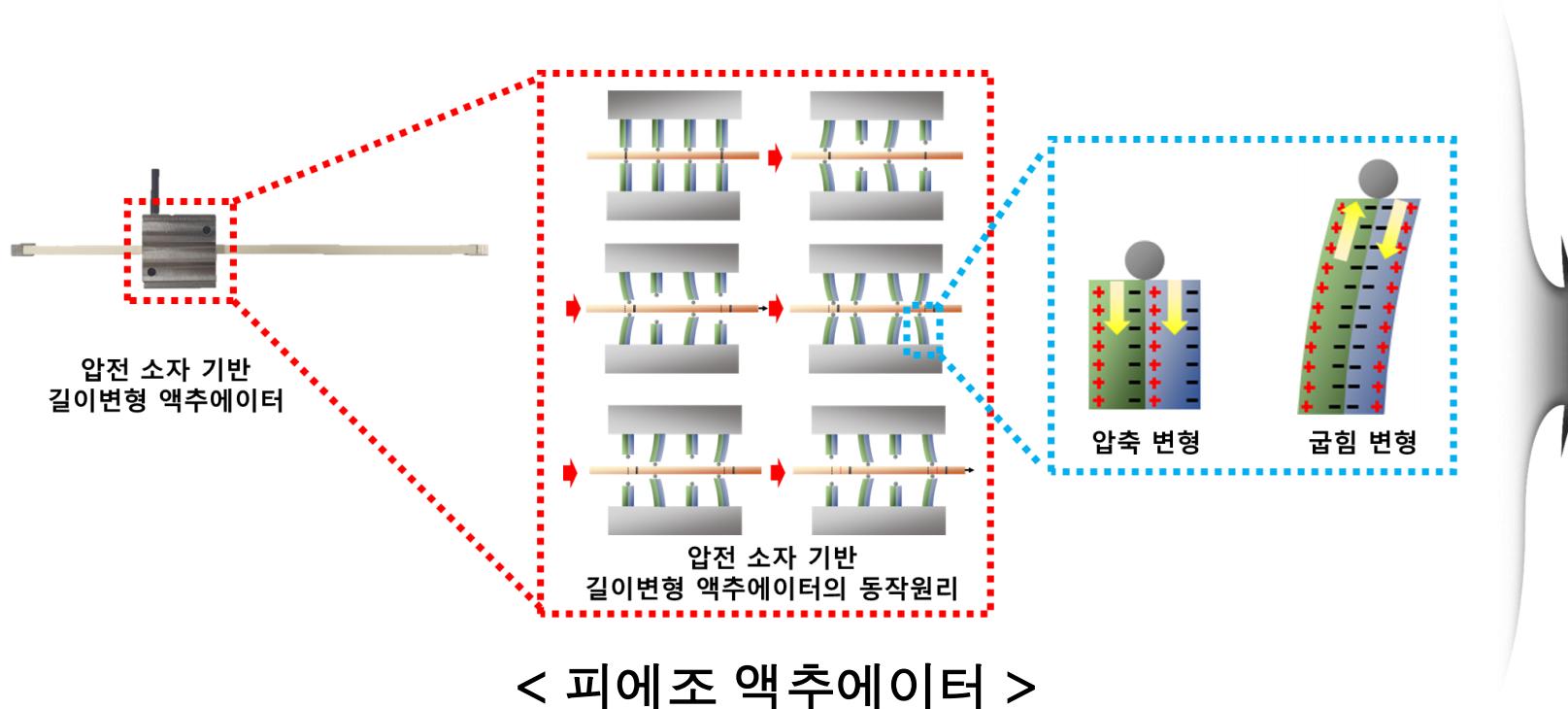
< 소스 코드>

2. 다형성 햅틱 컨트롤러 – F/W (통신 패킷 정의)



- Unity와 micro-controller간 통신을 위한 데이터 패킷 정의를 하였음
- 각 대학에서 개발한 Unity 기반 프로그램에 맞춰 사용자에게 줄 햅틱 피드백을 변경할 수 있도록 함

2. 다형성 햅틱 컨트롤러 – S/W (피에조 액추에이터)



```
int[] model = new int[2] { 0, 0 };
int[] axes1 = new int[2] { 1, 1 };
int[] step1 = new int[2] { -1000, -1000 };//길이(step) 조절( -= Left , + = Right )
int[] model_check = new int[2] { 0, 0 };
double[] speed1 = new double[2] { 500, 500 };//길이 변화 속도 조절
int[] set_rnp_ch = new int[2] { 1, 1 };
double[] set_rnp = new double[2] { 0, 0 };
점조 1개 0번 변경!만든 이 0명, 변경 내용 0개
public void Run()
{
    try
    {
        OpenConnection(ConnectionType.Usb);
        PrintControllerIdentification();
        var state = Enumerable.Repeat(PI_FALSE, PI_NUMBER_OF_AXIS).ToArray();
        SetServoState(Axis, state);
        GCS2.RNP(ControllerId, set_rnp_ch, set_rnp, 1);
        GCS2.SVO(ControllerId, "i", model);
        GCS2.DVL(ControllerId, axes1, speed1, 1);
        GCS2.OSM(ControllerId, axes1, step1, 1);
    }
    catch (GcsCommandError e)
    {
        PrintErrorMessage(e);
    }
}
```

< 소스 코드 >

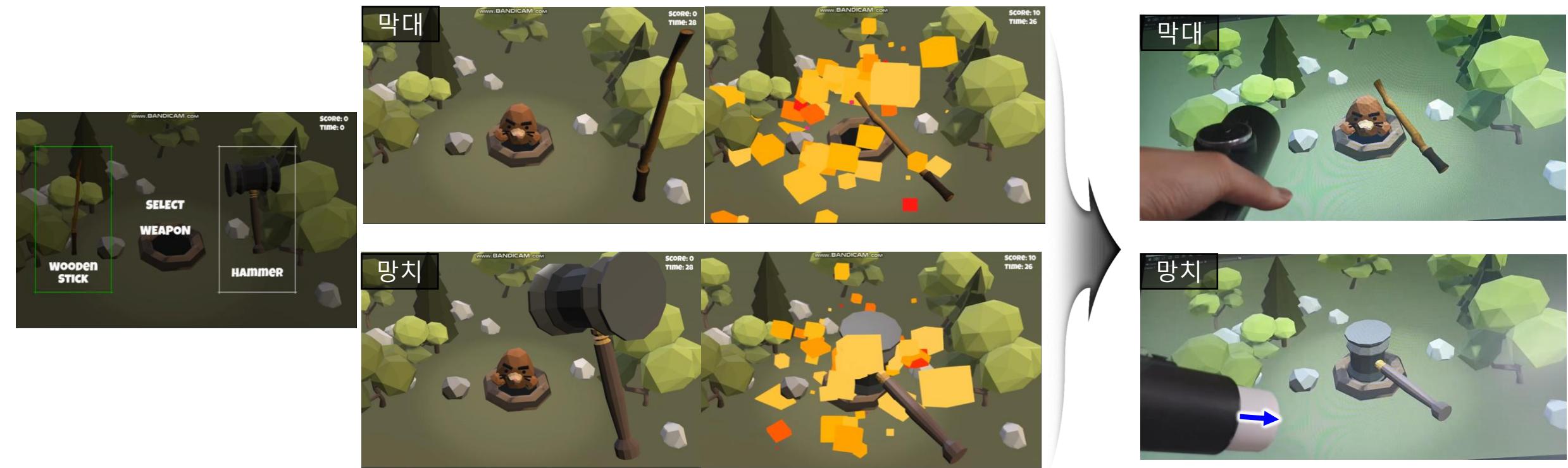
➤ 기능

- Openloop 구동 방식의 피에조 기반 리니어 액추에이터 구동 코드 작성 / Unity와 연동

➤ 작업 내용

- Openloop 구동 방식의 피에조 기반 리니어 액추에이터 구동 코드 작성(speed, step, direction 제어)
- 프로세스 호출로 Unity 프로그램 실행 시 동작하도록 구현(Unity에 dll 파일 적용 불가에 따른 해결 방안)

2. 다형성 햅틱 컨트롤러 – S/W (애플리케이션 연동)



< 두더지 잡기 애플리케이션 >

< 구현 >

➤ 기능

- 두더지잡기 도구에 따라 컨트롤러의 길이가 변하고, 두더지 잡는 이벤트 발생 시 진동이 울림

➤ 작업 내용

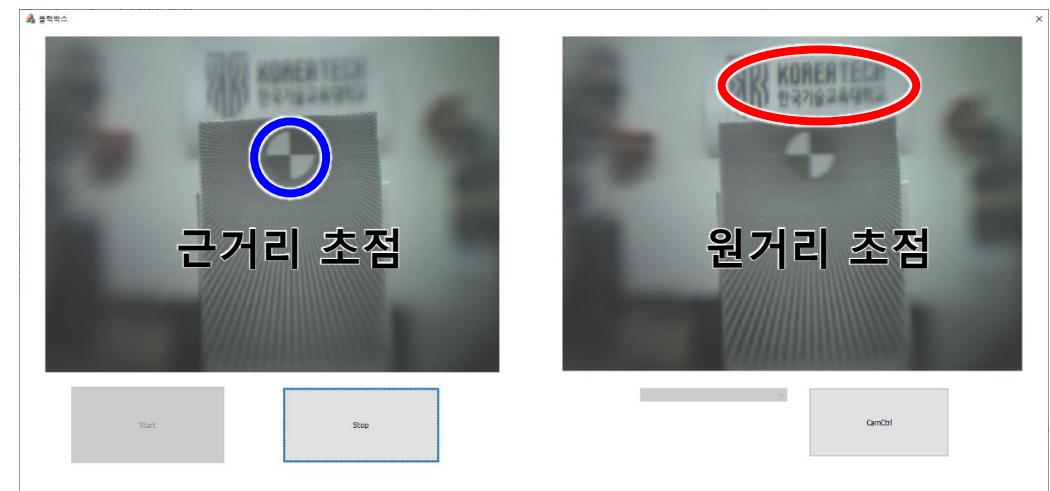
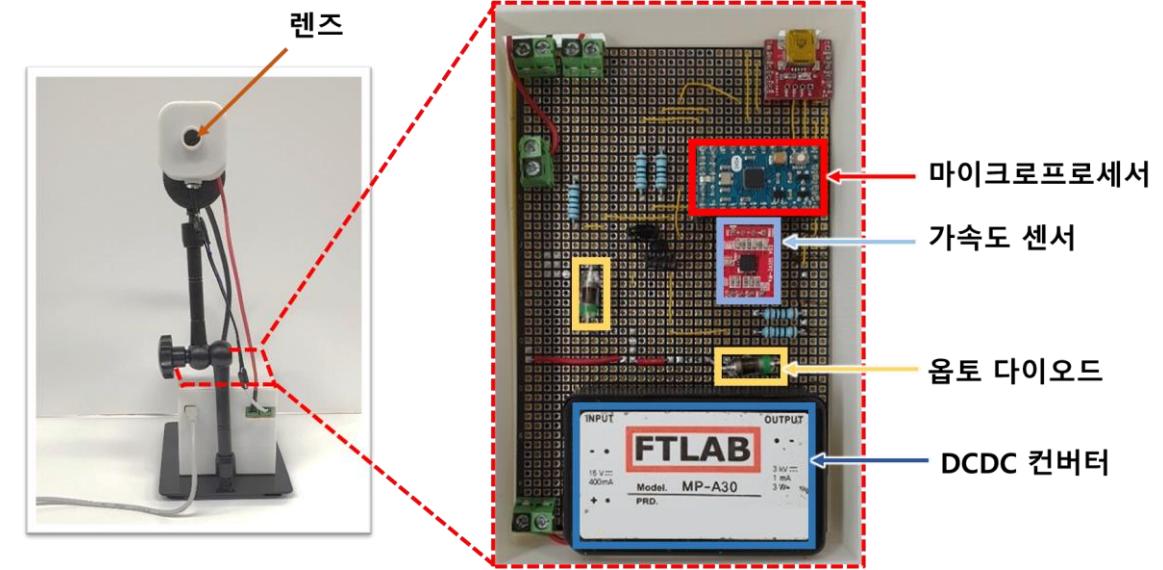
- 컨트롤러의 기울기에 따라 컨트롤러의 길이가 변하도록 구현함
- 두더지잡기 도구에 따라 사용자가 느끼는 진동 감각을 다르게 줄 수 있도록 구현함

3. 다중초점 블랙박스 시스템 개발

3. 다중초점 블랙박스 시스템 개발 - 개요

프로젝트 소개

작업 기간	2019. 03 ~ 2019. 11 (9개월)
인력 구성	학부생 2명(본인 포함)
프로젝트 목적	다중 초점을 가지는 블랙박스 시스템 개발
프로젝트 내용	<ol style="list-style-type: none">1. 폴리머 기반 렌즈의 구동 회로 개발2. 각 초점을 분리해서 보여주는 응용프로그램 개발3. 번호판 감지 시스템 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">1) 폴리머 기반 렌즈의 구동 회로 개발, F/W 개발2) 하우징 설계3) 응용 프로그램 내 초점 분리 기능, 충격 감지 이벤트 녹화 기능 구현
사용언어 및 개발 환경	C, C++, Visual studio MFC, Solidworks, IAR workbench, OpenCV



3. 다중초점 블랙박스 시스템 개발 – H/W (하우징)



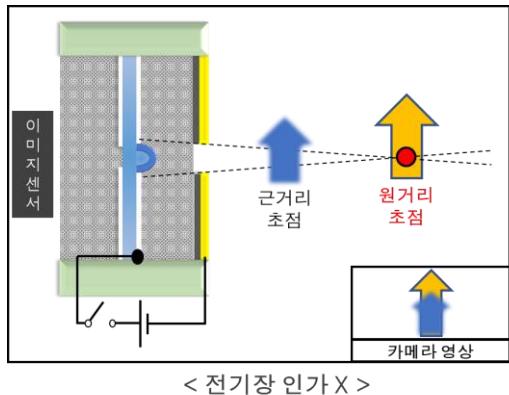
< 하우징 설계 >



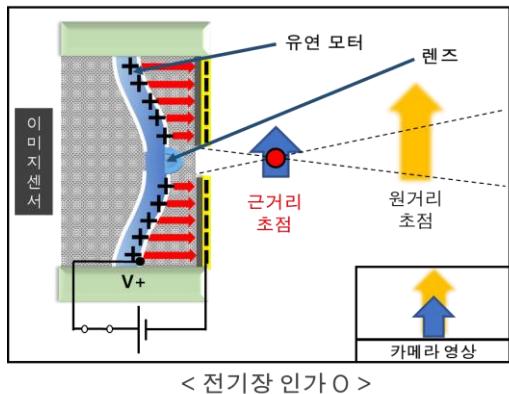
< 제작된 하우징 >

- Solidworks를 이용해 폴리머 렌즈의 하우징 및 회로 하우징 설계를 진행함

3. 다중초점 블랙박스 시스템 개발 – F/W 개발

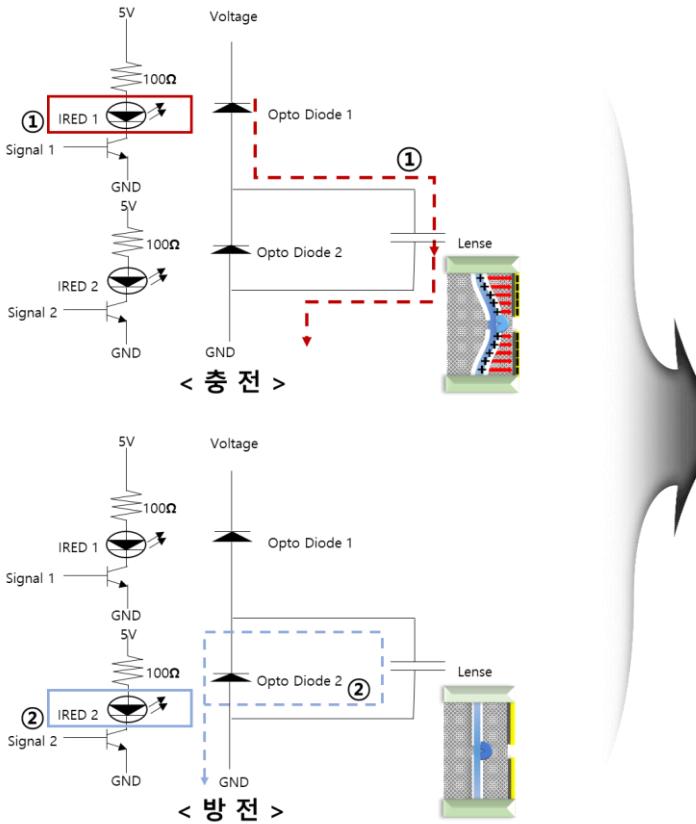


< 전기장 인가 X >



< 전기장 인가 O >

< 구동 원리 >



```
arduino_timer_accel_control2 §
void sw_OnOff(){
    if(accel_state==false)
        accelerator();
    else if(accel_state == true){
        accel_count++;
        if(accel_count==20000){
            accel_state=false;
            accel_count=0;
        }
    }
    timer_count++;
    if((timer_count%hz_time)==(zero_time)//OD1 HIGH (OD1,2 low for 2s )
        digitalWrite(SW_OD_1, !digitalRead(SW_OD_1));
    if((timer_count%hz_time)==(period-trg_time))// TRIGGER LOW
        digitalWrite(TRIGGER_PIN, !digitalRead(TRIGGER_PIN));
    if((timer_count%hz_time)==period){//OD1 OD2 LOW, TRIGGER HIGH
        digitalWrite(TRIGGER_PIN, !digitalRead(TRIGGER_PIN));
        digitalWrite(SW_OD_1, !digitalRead(SW_OD_1));
        Serial.write('t');//actuator work!
    }
    if((timer_count%hz_time)==(zero_time+period)){//OD2 HIGH
        digitalWrite(SW_OD_2, !digitalRead(SW_OD_2));
    }
    if((timer_count%hz_time)==(hz_time-trg_time))//TRIGGER LOW
        digitalWrite(TRIGGER_PIN, !digitalRead(TRIGGER_PIN));
    }
    if(timer_count == hz_time){//OD2 LOW, TRIGGER HIGH
        digitalWrite(TRIGGER_PIN, !digitalRead(TRIGGER_PIN));
        digitalWrite(SW_OD_2, !digitalRead(SW_OD_2));
        timer_count=0;
    }
}
```

< 소스 코드 >

➤ 기능

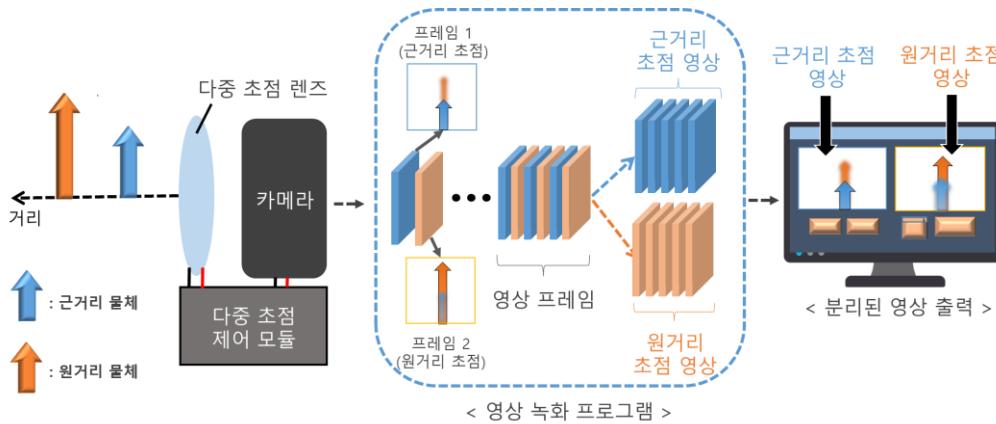
- 저전압(3.3 V)으로 고전압 신호(~3.5 kV)를 제어할 수 있는 기능 구현

➤ 작업 내용

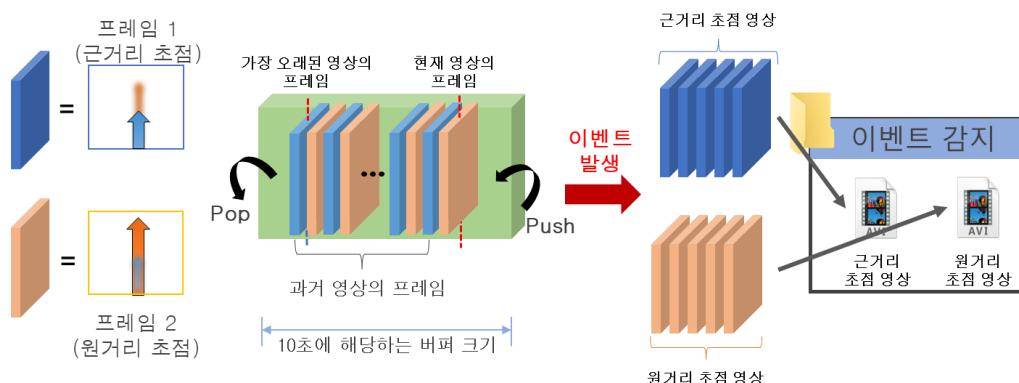
- 입력 전압, 주파수에 따라 고전압 스위칭 회로가 동작하도록 코드 작성

3. 다중초점 블랙박스 시스템 개발 – S/W 개발

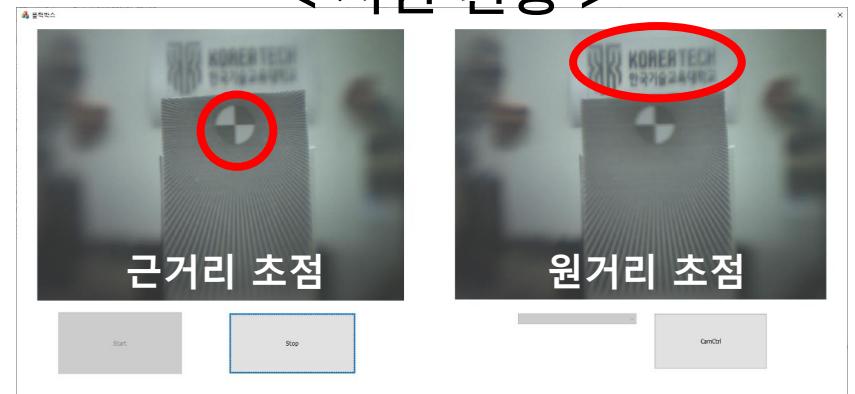
초점 분리 원리



이벤트 감지
자동 저장
원리



< 시연 환경 >



< 응용프로그램(MFC C++) >

➤ 기능

- 근거리 초점과 원거리 초점을 분리할 수 있는 기능 구현
- 이벤트 발생 시, 이벤트 기준 전후 10초까지 영상을 자동으로 저장하는 기능 구현

➤ 작업 내용

- Frame loss를 방지하기 위해 Queue 구조의 임시 버퍼 사용하여 각 기능 구현

4. 고전압 스위칭 회로 개발

4. 고전압 스위칭 회로 개발 – 개요

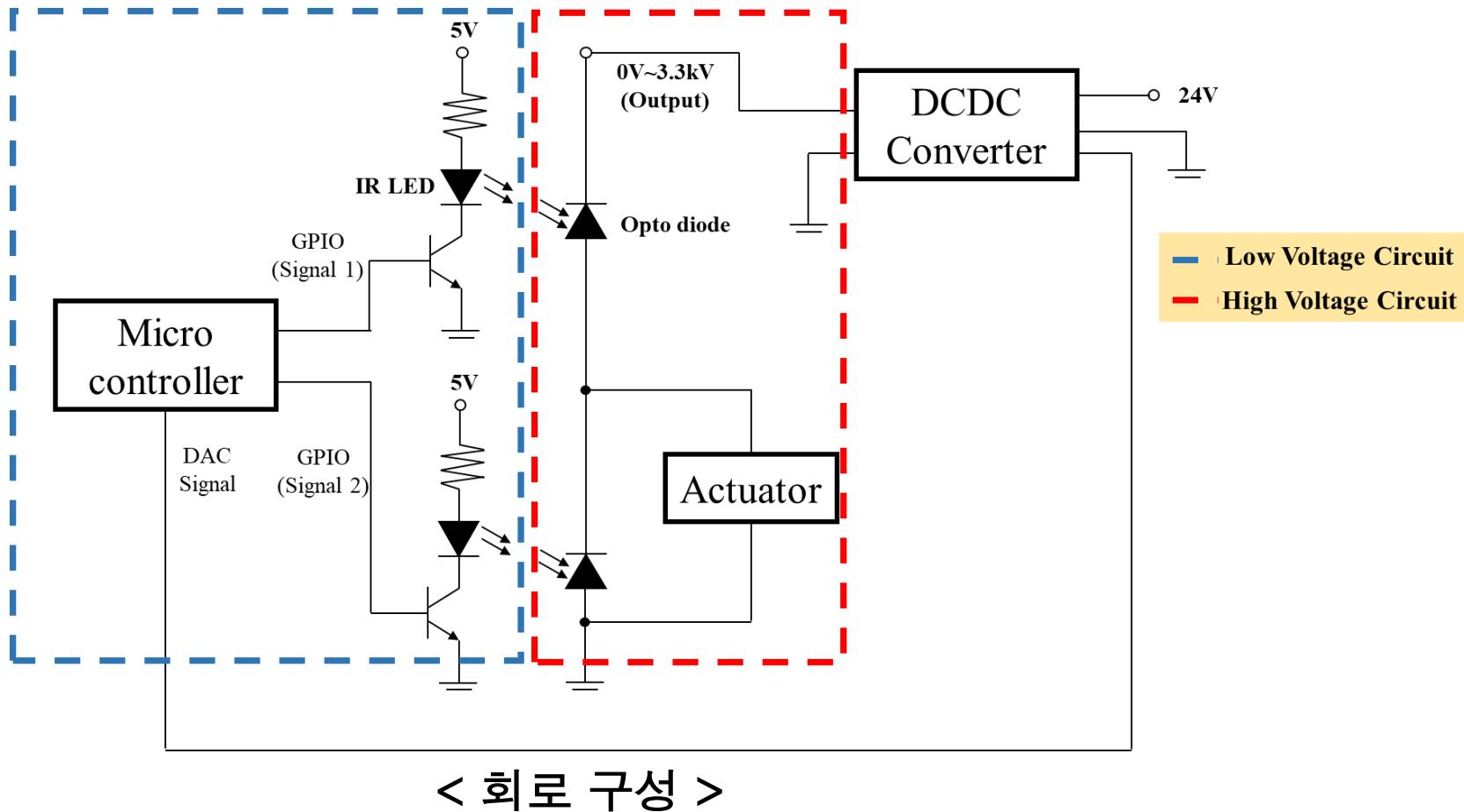
프로젝트 소개

작업 기간	2020. 04 ~ 2020. 04 (1개월 이내)
인력 구성	석사과정 1명(본인)
프로젝트 목적	폴리머 기반 렌즈를 테스트할 수 있는 고전압 스위칭 회로 개발(외주)
프로젝트 내용	<ol style="list-style-type: none">1. 고전압 스위칭 회로 개발2. 회로를 제어할 수 있는 응용프로그램 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">1) 고전압 스위칭 회로 개발2) 하우징 설계 및 제작3) 회로 제어 응용프로그램 개발
사용언어 및 개발 환경	C, C++, C# / Visual studio, Solidworks, IAR workbench



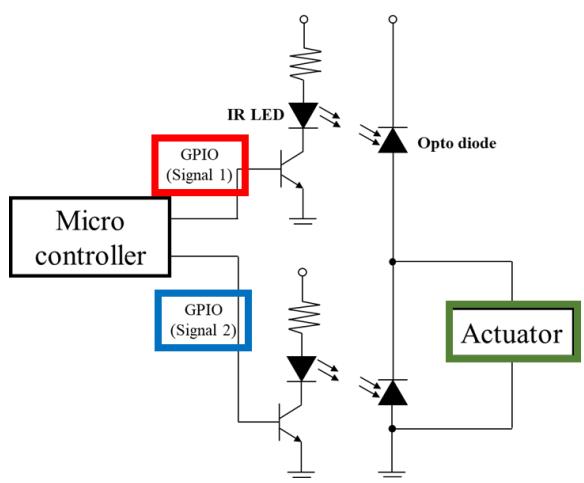
< 개발된 제품 >

4. 고전압 스위칭 회로 개발 - H/W (회로 설계)



➤ 저전압(3.3 V)으로 고전압(~5 kV)을 제어할 수 있는 회로 설계

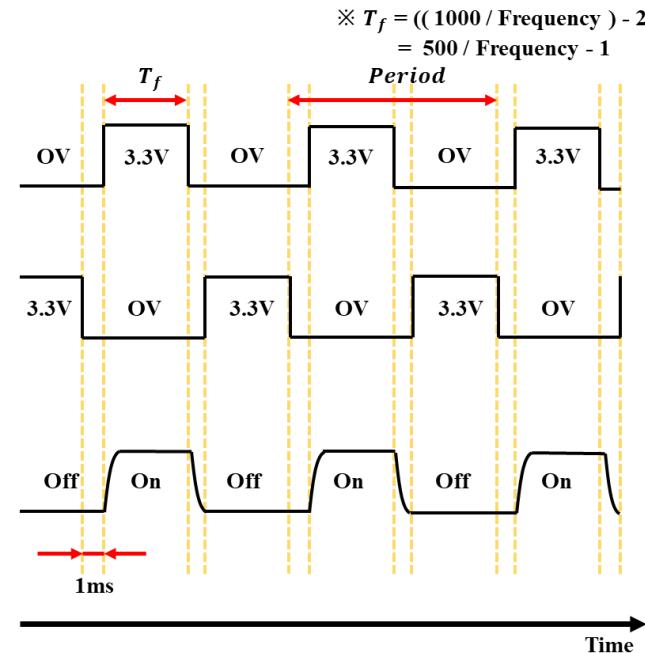
4. 고전압 스위칭 회로 개발 - F/W 개발



Signal 1

Signal 2

Actuator



< 구동 로직 >

```
void TIM3_IRQHandler(void)//ch1
{
    if (TIM_GetITStatus(TIM3, TIM_IT_Update) == SET)
    {
        TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
        if(ch1_work_state == true && ch1_off_state == false){
            if(ch1_mode == 0){
                GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
                GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
                ch1_off_state = true;
            }
            else if(ch1_mode == 1 && ch1_on_state == false){
                GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
                ch1_on_state = true;
            }
            else if(ch1_mode == 2){
                if(ch1_count == 0){
                    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
                }
                else if(ch1_count == (ch1_time + idle_time)){
                    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
                }
                else if(ch1_count == ((ch1_time+idle_time)*2)){
                    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
                }
                else if(ch1_count == ((ch1_time+idle_time)*2+1)){
                    ch1_count = -1;
                }
                ch1_count++;
            }
        }
    }
}
```

< 소스코드 >

➤ 기능

- 입력 주파수, 전압에 따라 고전압 신호를 스위칭

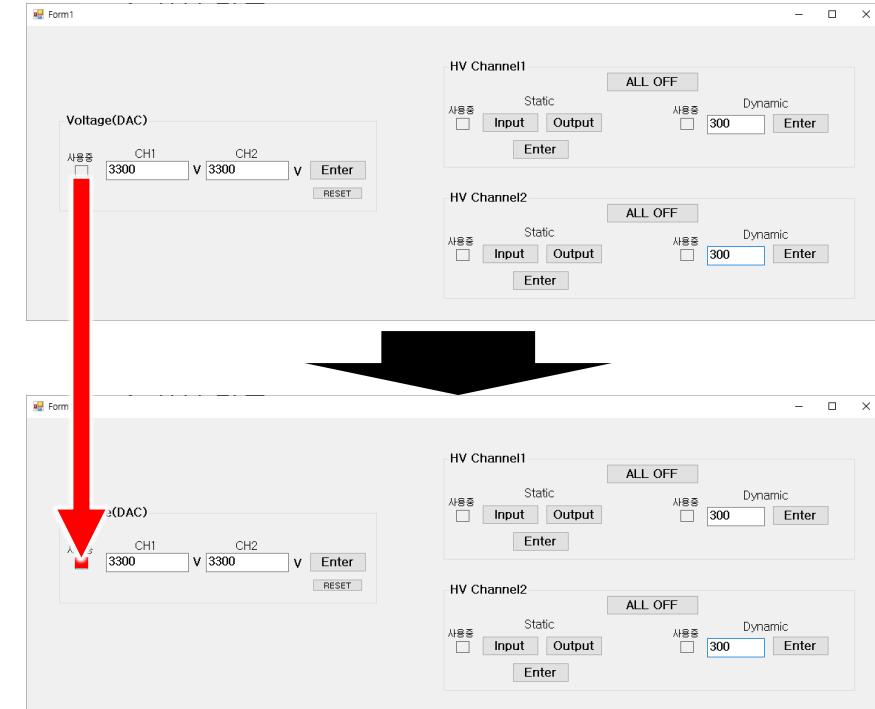
➤ 작업 내용

- 저전압 제어 코드를 작성하여 고전압 신호를 제어하도록 함
- 고전압에 의한 액추에이터 고장을 방지하고자, Signal 1, 2의 전압이 동시에 바뀌지 않고, 1 ms 간격을 두고 바뀌도록 함

4. 고전압 스위칭 회로 개발 – S/W 개발

```
Form1.cs  X | C# WindowsFormsApplication2 | WindowsFormsApplication2.Form1
163     창조 1개
164     private void static_enter_ch1_Click(object sender, EventArgs e)
165     {
166         port = new SerialPort("COM2", 115200, Parity.None, 8, StopBits.One);
167         port.Open();
168
169         if (check_input_ch1 == true)
170         {
171             if (check_dynamic_using_ch1 == false)
172             {
173                 static_using_ch1.BackColor = Color.Red;
174                 check_static_using_ch1 = true;
175
176                 send_data[0] = start_byte;
177                 send_data[1] = (Byte)2;
178                 send_data[2] = (Byte)1;
179                 send_data[3] = 0;
180                 send_data[4] = 0;
181                 send_data[5] = 0;
182                 send_data[6] = stop_byte;
183
184                 //FA 02 01 00 00 00 FB (in)
185
186                 port.Write(send_data, 0, 7);
187             }
188         }
189     }
```

< 소스코드 >



< 응용프로그램(C#) >

➤ 기능

- 값을 입력하고 enter를 누르면 응용프로그램 내 사각 박스가 빨간색으로 변경되고, 마이크로컨트롤러로 데이터 전달

➤ 작업 내용

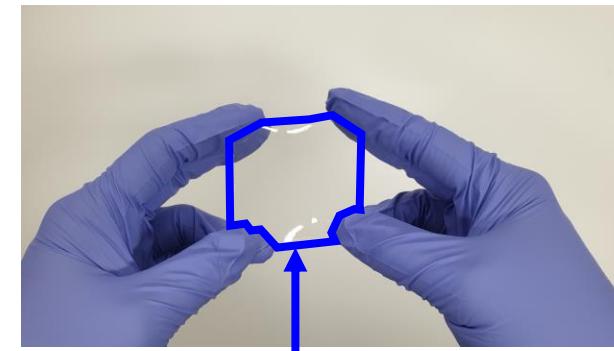
- 입력 전압, 주파수에 따라 회로가 동작하도록 통신 코드 작성

5. 필름형 고신축성 3D 터치 센서 개발

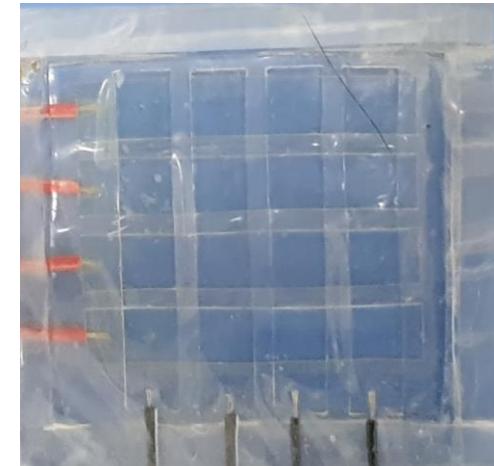
5. 필름형 고신축성 3D 터치 센서 개발 – 개요

프로젝트 소개

작업 기간	2020. 03 ~ 2020. 08 (6개월)
인력 구성	박사과정 2명, 석사과정 1명(본인)
프로젝트 목적	투명 고신축성 3D 터치 센서 개발
프로젝트 내용	<ol style="list-style-type: none">1. 투명 고신축성 3D 터치 필름 구조 설계2. 3D 터치 필름 구동을 위한 단일 칩 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">1) 3D 터치 센서 제작2) 센서 압력 측정 F/W 개발
사용언어 및 개발 환경	C, C++, C# / Visual studio, IAR workbench

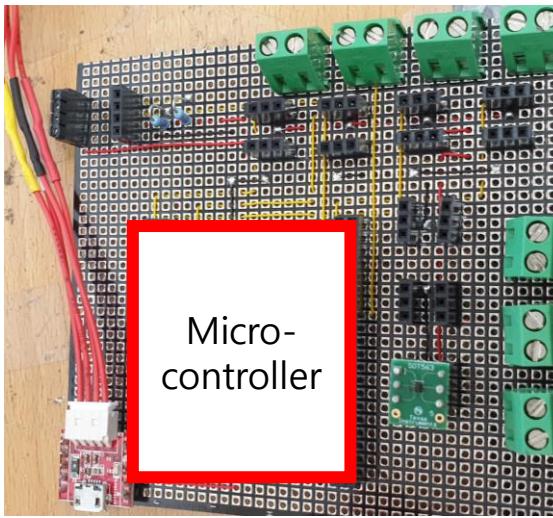


투명, 고신축성 폴리머



< 제작된 3D 터치 센서 >

5. 필름형 고신축성 3D 터치 센서 개발 – F/W 개발



Micro-controller

< 3D 터치 센서의 구동 회로 >

```
void update_Cal(uint8_t ch, int8_t offset){  
    uint8_t toggle = 0;  
    int cnt_TIM3 = 0;  
    uint8_t isRead = 0;  
    void TIM2_IRQHandler(void)//타이머 인터럽트 핸들러  
{  
        static float ch1_val = 0;  
        static float ch2_val = 0;  
        static float ch3_val = 0;  
        static float ch4_val = 0;  
  
        if (TIM_GetITStatus(TIM2, TIM_IT_Update) == SET)  
        {  
            TIM_ClearITPendingBit(TIM2, TIM_IT_Update);  
  
            if(cnt_TIM3 >15 && isRead ){  
                ch1_val = read_Cap(1);  
                ch2_val = read_Cap(2);  
                ch3_val = read_Cap(3);  
                ch4_val = read_Cap(4);  
  
                sprintf(buffer2, "%d,%f,%f,%f,%f\n", toggle, ch1_val, ch2_val, ch3_val, ch4_val );  
                UART1Print(buffer2);  
                isRead = 1;  
            }  
        }  
    }  
}
```

< 소스 코드 >

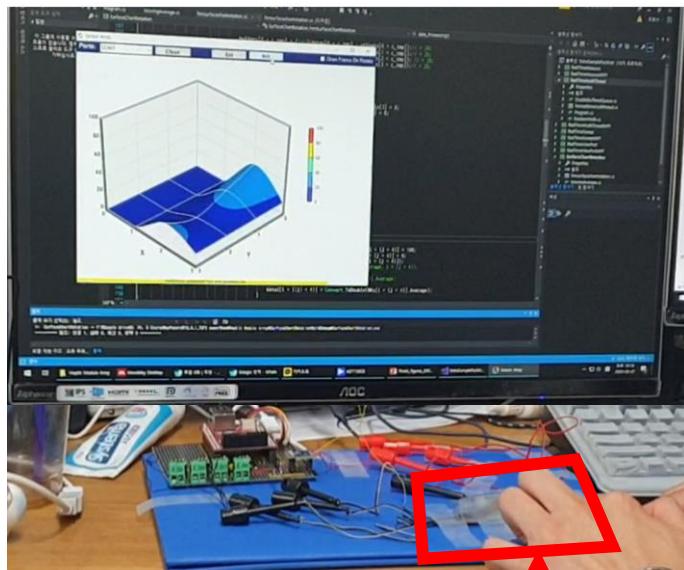
➤ 기능

- 3D 터치 센서의 capacitance 값을 받아 PC로 보냄

➤ 작업 내용

- 각 센서의 Capacitance 값을 읽을 수 있도록 회로 구성 후 capacitance값을 PC로 보내도록 함.

5. 필름형 고신축성 3D 터치 센서 개발 – S/W 개발



3D 터치 센서

< 3D 터치 센서 구동 >

- 기능
 - 센서의 노이즈 감소를 위해 Moving Average Filter 적용
- 작업 내용
 - Moving average filter를 적용하여 노이즈를 감소시키는 코드 작성

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace SurfaceChartRotation
{
    public class MovingAverage
    {
        private Queue<int> samples = new Queue<int>();
        private int windowSize = 3;
        private int sampleAccumulator;
        public int Average { get; private set; }

        public void ComputeAverage(int newSample)
        {
            sampleAccumulator += newSample;
            samples.Enqueue(newSample);

            if (samples.Count > windowSize)
            {
                sampleAccumulator -= samples.Dequeue();
            }

            Average = sampleAccumulator / samples.Count;
        }
    }
}
```

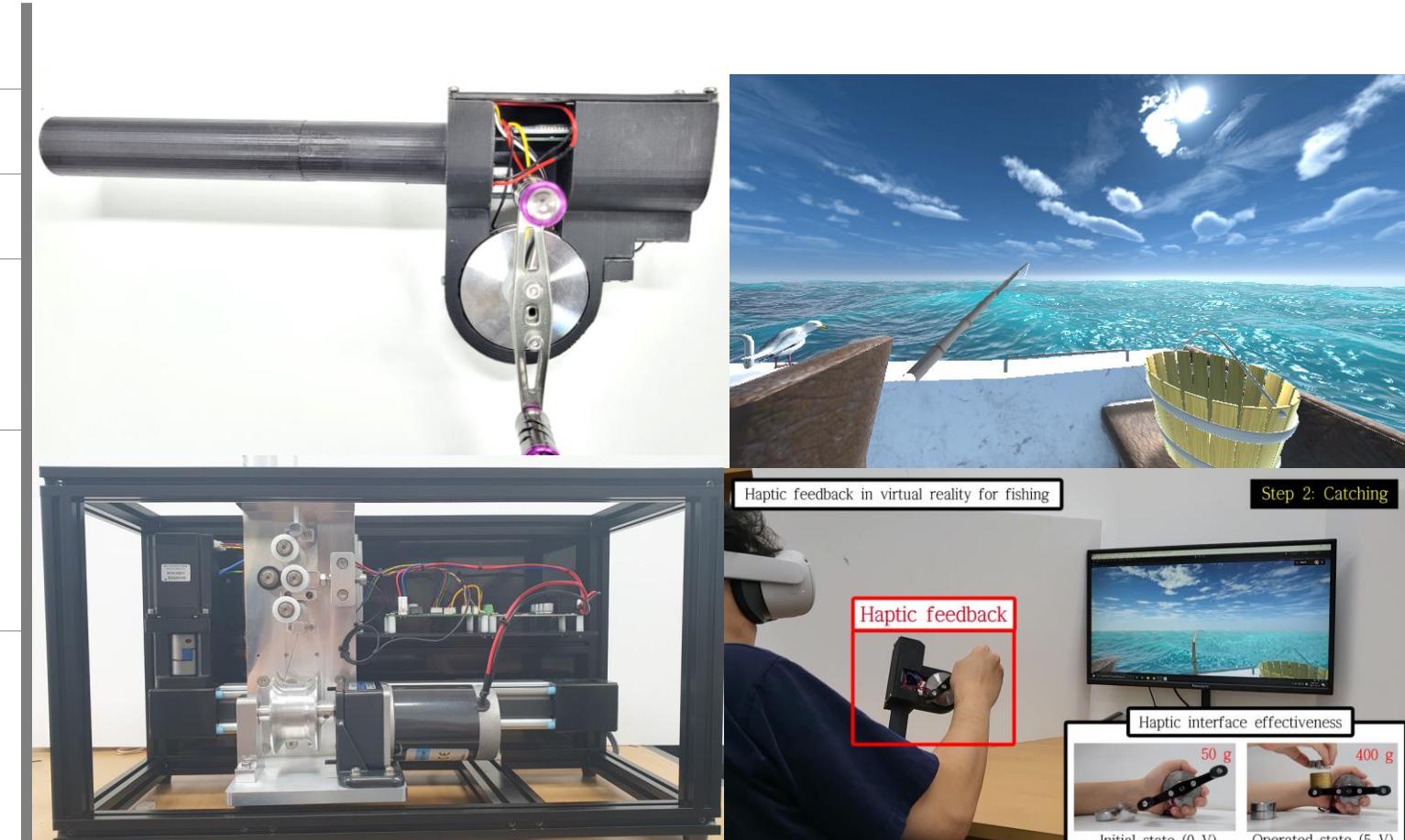
< 소스 코드 >

6. VR 낚시 레저활동을 위한 액추에이터

6. VR 낚시 레저활동을 위한 액추에이터 – 개요

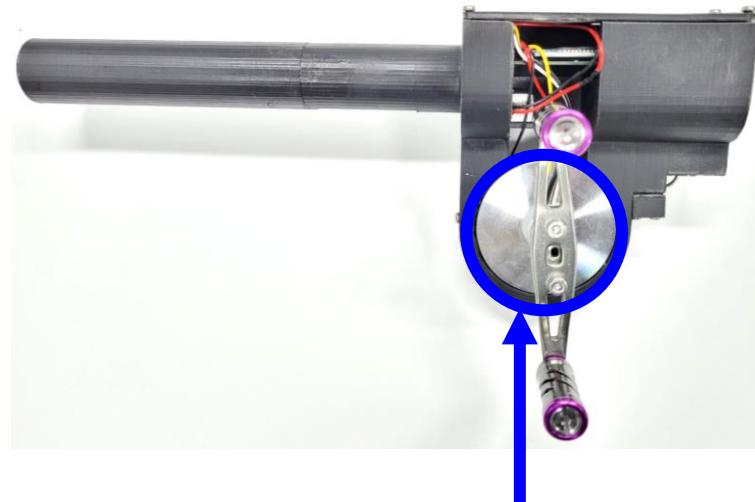
프로젝트 소개

작업 기간	2022. 01 ~ 2022. 12 (12개월)
인력 구성	박사과정 1명, 석사과정 1명(본인), 학부생 2명
프로젝트 목적	VR 낚시 레저활동을 위한 근감각 액추에이터 개발
프로젝트 내용	<ol style="list-style-type: none">근감각 액추에이터 개발DC 모터 모듈 개발VR 낚시 애플리케이션 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">근감각 액추에이터 개발근감각 액추에이터의 F/W 개발
사용언어 및 개발 환경	C, C++ / Visual studio, Solidworks, IAR workbench



< 개발된 H/W, S/W >

6. VR 낚시 레저활동을 위한 액추에이터 - F/W 개발



근감각 액추에이터

< 액추에이터 실물 사진 >

➤ 기능

- Unity 기반 VR 애플리케이션 내 물고기 종류에 따라 근감각을 다르게 줄 수 있도록, PWM 주파수 및 duty를 PC에서 입력하도록 함

< 소스 코드 >

```
void TIM5_IRQHandler (void)
{
    if (TIM_GetITStatus(TIM5, TIM_IT_Update) == SET)
    {
        TIM_ClearITPendingBit(TIM5, TIM_IT_Update);

        // ***** DC Signal code *****
        if(Weak_resitive_torque_on_flag == true){
            SetPWM_TIM4_CH1(25);
        }

        if(Normal_resitive_torque_on_flag == true){
            SetPWM_TIM4_CH1(30);
        }

        if(Strong_resitive_torque_on_flag == true){
            SetPWM_TIM4_CH1(50);
        }
        // ***** AC Signal code *****
        if(AC_signal_on_flag == true){

            AC_signal_flag = !AC_signal_flag;
            if(AC_signal_flag){
                SetPWM_TIM4_CH1(45);
            }
            else{
                SetPWM_TIM4_CH1(0);
            }
        }
    }
}
```

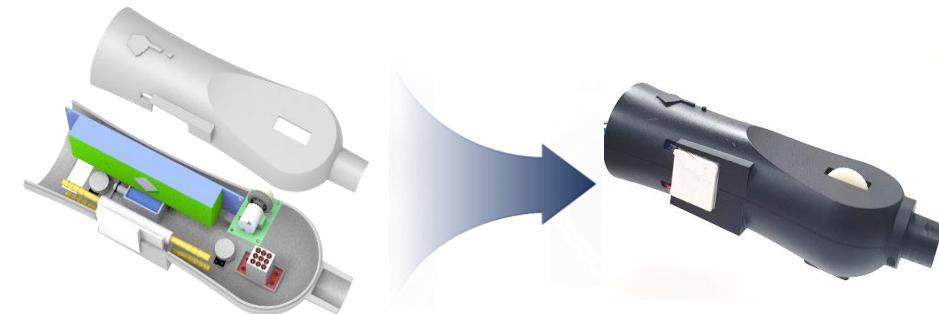
7. HD 촉감 기술 기반 초실감 콘텐츠 재현 기술 개발

7. HD 촉감 기술 기반 초실감 콘텐츠 재현 기술 개발 - 개요

프로젝트 소개

작업 기간	2019.12 ~ 2020.02 (3개월)
인력 구성	석사과정 1명, 학부연구생 2명(본인 포함)
프로젝트 목적	단순 진동 뿐만 아니라 형상, 냉열감 등을 사용자에게 제공할 수 있는 햅틱 컨트롤러 개발
프로젝트 내용	<ol style="list-style-type: none">햅틱 컨트롤러 구조 설계각 액추에이터 개발VR 컨텐츠 개발
주요 업무 및 상세 역할	<ol style="list-style-type: none">햅틱 컨트롤러 구동 회로 개발손목 근감각 장치 설계
사용언어 및 개발 환경	Solidworks

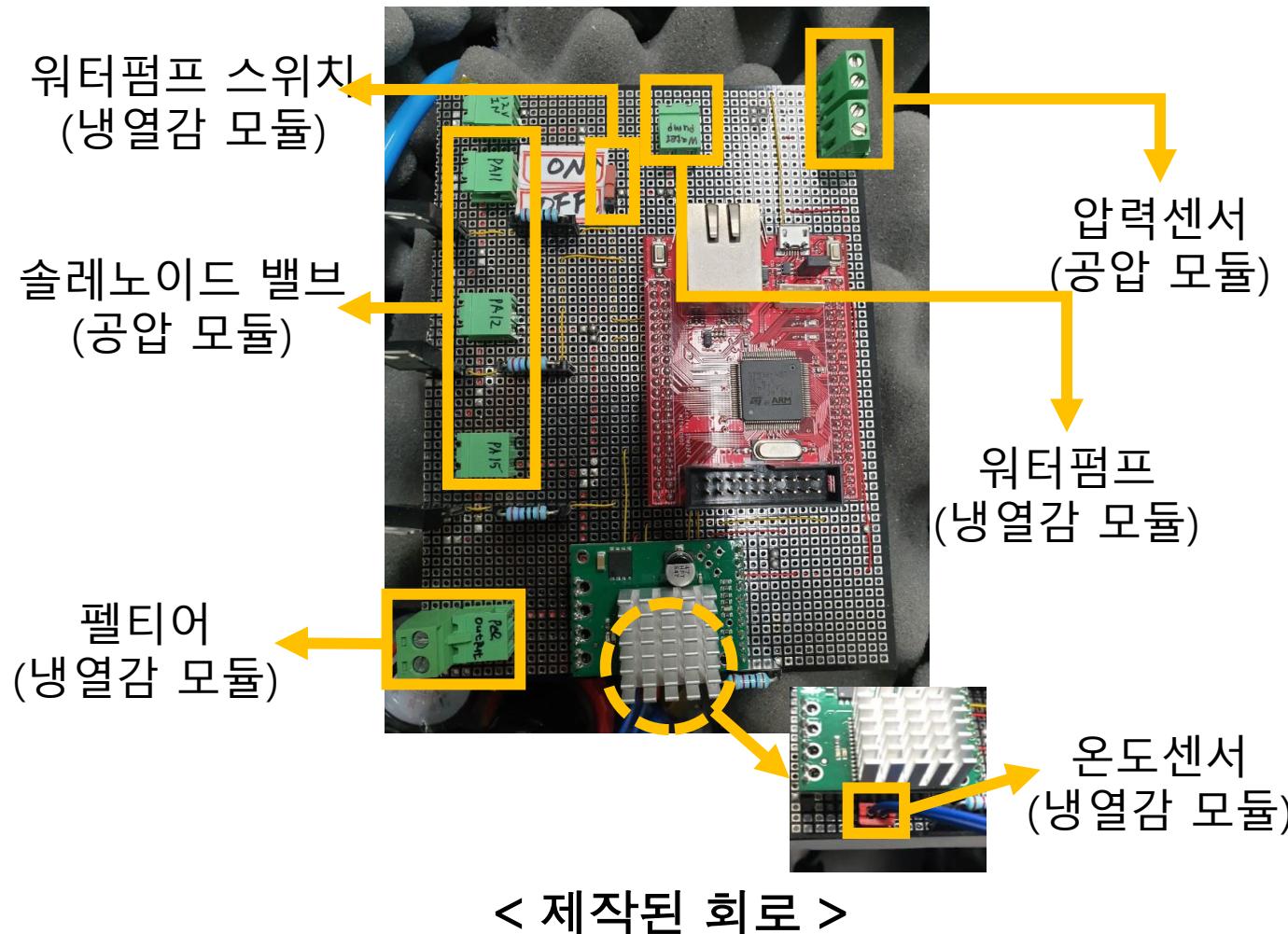
▶ 멀티 모달 햅틱 컨트롤러



▶ 멀티 모달 햅틱 컨트롤러에 사용되는 모듈

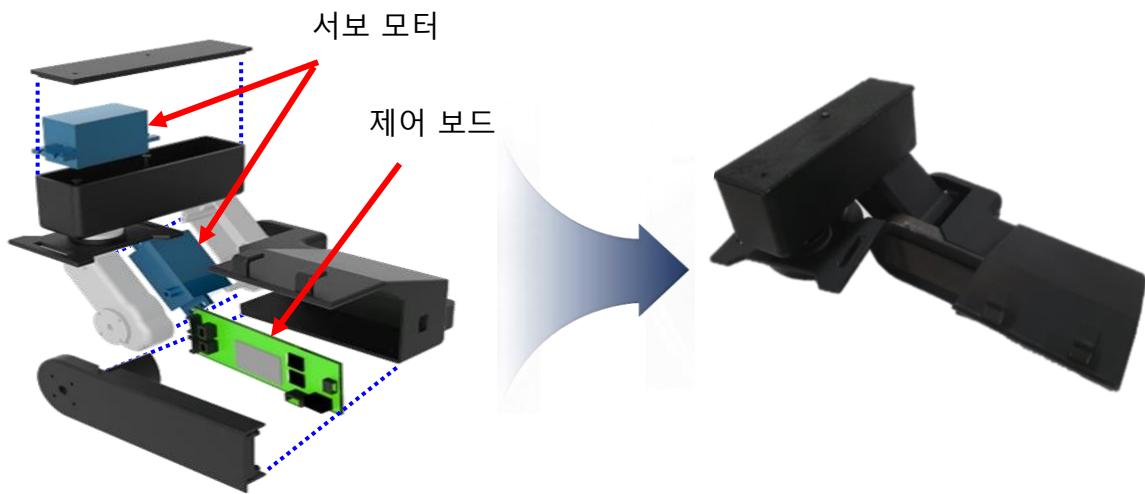


7. HD 촉감 기술 기반 초실감 콘텐츠 재현 기술 개발 – H/W (회로 개발)



▶ 각 액추에이터(냉열감 모듈, 형상 변형 모듈)를 제어할 수 있는 회로 구현

7. HD 촉감 기술 기반 초실감 콘텐츠 재현 기술 개발 – H/W (근감각 장치)



< 손목 근감각 장치 설계 >



< 제작된 손목 근감각 장치 >

- Solidworks를 이용해 VR 애플리케이션 내 상황에 따라 근감각을 줄 수 있는 손목 근감각 장치 개발을 진행함

감사합니다
