

The image shows a white rectangular portfolio cover with a thin black border. Two black binder clips are attached to the top edge. In the top-left corner, there is a thin black diagonal line. In the bottom-right corner, there is a thin black diagonal line. The word "PORTFOLIO" is written in a black, handwritten-style font in the center. Below it, a red rounded rectangle contains white text. A small horizontal black line is positioned below the red rectangle.

# PORTFOLIO

IT융합부품소재공과대학  
메카트로닉스공학과  
김우석

# PORTFOLIO

---

## Profile

---

**Education** 2013. 03 ~ 2020.02 동의대학교 메카트로닉스공학과 졸업

---

**Work Experiences** 워킹 홀리데이 호주(2017-01~2018~01)

---

**Project**

- Bio Safety Helmet
- Segway
- 모션제어시스템(Wanners bros)

---

**Certification**

- 정보통신기사
- 정보처리기사
- TOEIC 810
- 한국사 2급
- 컴퓨터활용능력 1급

## Index

### BIO SAFETY HELMET

- 2019년 1학기
- 역할 : S/W프로그래밍
- 참여인원 : 7명
- 전공과목 : 캡스톤디자인
- 타전공수강 : 사물인터넷
- 성과 : A, 아이디어 난장대회 참가
- Raspberry pi 사용(언어 python)

### SEGWAY

- 2018년 2학기
- 역할: 팀장(프로젝트 총괄 및 하드웨어 설계)
- 참여인원 : 4명
- 전공과목 : 캡스톤디자인
- 성과 : A+
- Arduino-Mega 사용(언어 C/C++)

### 모션제어시스템 (WANER BROS.)

- 2018년 2학기
- 역할: 팀장(프로젝트 총괄 및 CAD 디자인)
- 참여인원 : 5명
- 전공과목 : 모션제어시스템
- 교내프로그램 : 전공 추수학습
- 성과 : A
- MFC사용(언어 C/C++)

## PORTFOLIO

### 기여도

#### 작품설명 (기획포인트)

요즘 주목을 받고 있는 IOT(사물인터넷)기술을 기반으로 근로자가 사용하는 안전모에 온도 센서와 심박센서를 장착하여 실시간으로 착용자의 체온과 심박을 측정한다. 수집되는 데이터를 Raspberry zero를 통해 주기적으로 IOT서비스에 적합한 초경량 통신 프로토콜인 CloudMQTT서버로 송신하여 스마트폰APP 또는 컴퓨터를 통해 정보를 확인할 수 있다. 실시간으로 입력되는 데이터를 기반으로 안전모의 착용유무도 판별이 가능하며, 심박이나 체온이 정상범위를 벗어나는 경우, 알람과 LED를 통해 주변 작업자 및 관리자가 즉각적으로 인지하고 조치를 취할 수 있도록 한다.

BIO SAFETY HELMET

## Bio Safety Helmet

20142975  
20133020  
20133044  
20133070  
20142984  
20133047

김민석  
최유완  
조근준  
김우석  
박경용  
박창민

## Bio Safety Helmet

### Bio Safety Helmet 이란?



BPM



36.5°C



근로자의 심박수와 체온 정보를 바탕으로 몸의 이상상태 및 안전모의 착용 유무를 판별하여 실시간으로 사무실과의 데이터 송수신이 이루어짐으로써 작업현장에서의 안전사고율을 줄여주는 장비

## 선정배경 및 목표

안전모 미착용으로 인한 사망자가 17%

조성대 | 승

**"안전모 쓰지 않아 근로자 사망"...法, 업체대표에 실형**

대구청

(울산=뉴스1) 김기열 기자 | 2019-03-19 13:28 송고

기사보기

네티즌의견

좋아요 0개

공유하기

트윗

카카오

인쇄 | 확대 | 축소



## 선정배경 및 목표

### 목표

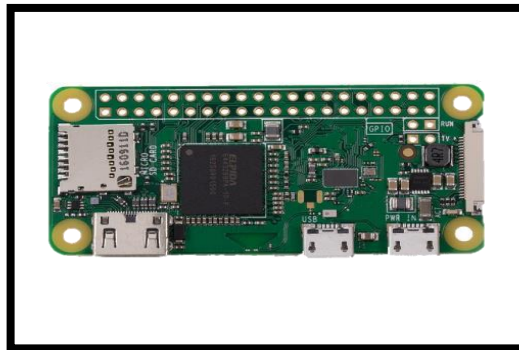
1. 라즈베리파이 ZERO W를 통해 심박센서 및 온도센서의 값을 받아올 수 있는 알고리즘을 구현한다.
2. ZERO W에 수집되는 데이터를 주기적으로 CloudMQTT서버로 송신함으로써 스마트폰APP 또는 컴퓨터를 통해 확인할 수 있도록 한다.
3. 심박이나 체온이 정상범위를 벗어나는 경우, 알람을 통해 작업자 및 관리자가 즉각적으로 알 수 있도록 한다.

## 구성요소 - MCU



### Raspberry pi 3 B+ (알람)

- CPU : BCM2837 64-bit / 1.4GHz
- GPIO : 40핀
- 메모리 : 1 GB
- 규격 : 85.60 × 56.5 mm, 45g



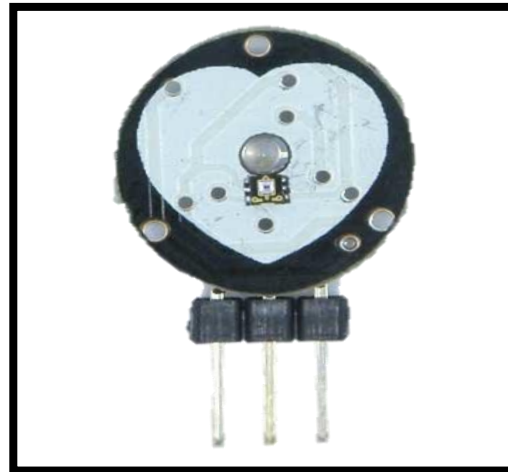
### Raspberry pi zero w (헬멧)

- CPU : BCM2835 64-bit / 1.0GHz
- GPIO : 40핀
- 메모리 : 512MB
- 규격 : 65 × 30mm, 9g
- Bluetooth 및 Wifi 통신 지원



## 구성요소 - 센서

### SEN-11574 (심박센서)



- 3V 또는 5V 동작
- 노이즈 방지 회로 내장
- 아날로그 출력
  - >> ADC 모듈을 이용하여 I2C 통신 가능

>> 내장된 광학앰프에서 컷볼의 미세혈관에 적외선을 방출하여 반사되는 빛의 양을 감지하고, 혈액의 흐름에 따라 빛의 양이 변화 하는 것을 이용하여 맥박을 측정

## 구성요소 - 센서

### MLX90614 (온도센서)



- 3V에서만 동작
- 비 접촉 온도 측정 방식
- ADC가 내장되어있음
- PWM 및 I2C통신방식 지원

>> 온도를 측정하고자 하는 대상에 적외선을 방출하여 그 대상으로부터 반사되는 적외선 빛의 파장을 온도로 환산하여 측정



## 구성요소 - 통신



### MQTT란?

- Message Queuing Telemetry Transport의 약자로 통신 장비, 모바일, 스마트폰 기기에 최적화된 가벼운 메시징 프로토콜로 IoT장비에 현재 가장 많이 사용되고 있는 프로토콜이다.



## PORTFOLIO

### • 주된 학습

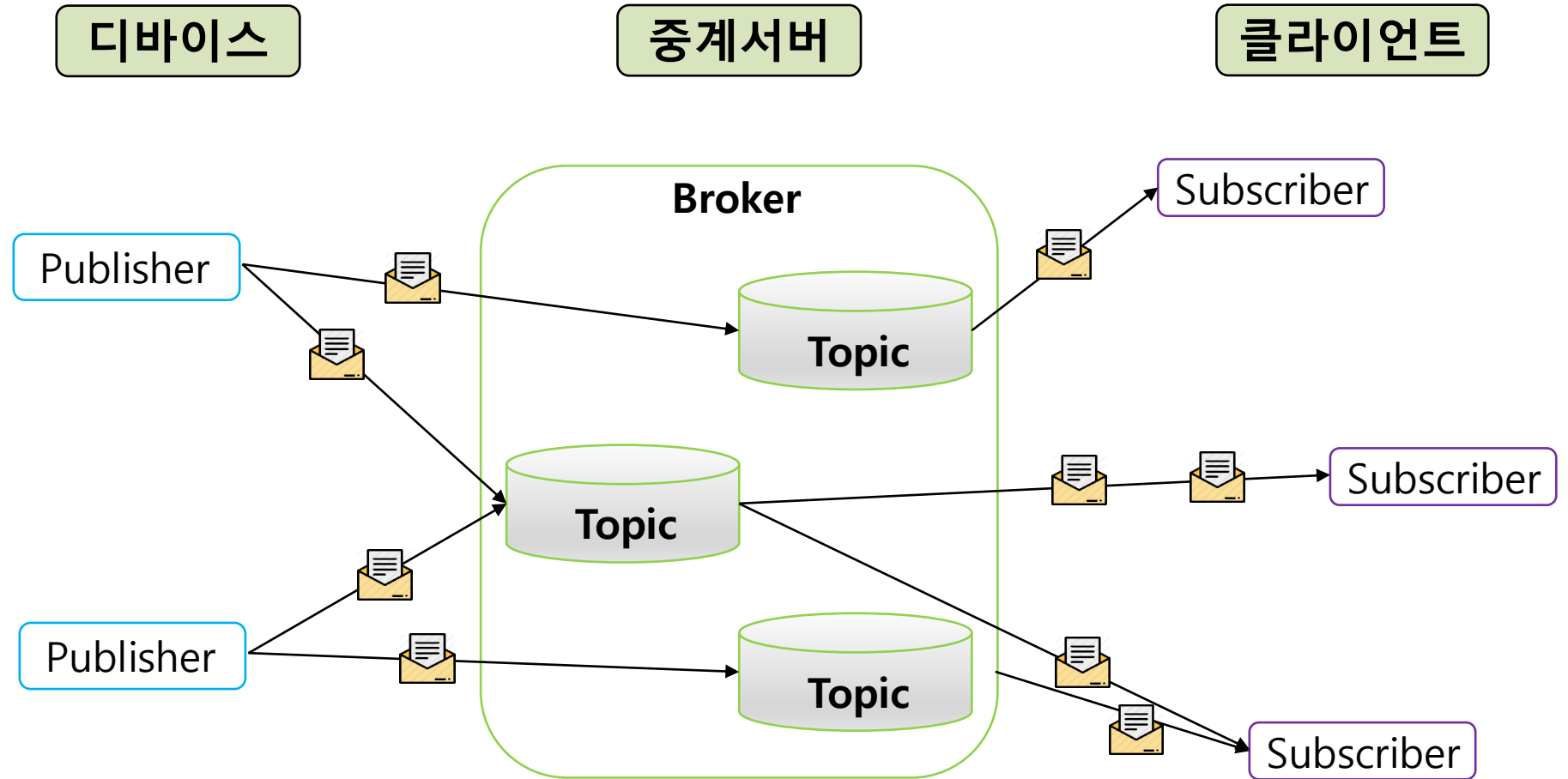
IoT기술에 사용되는 프로토콜은 여러가지가 있지만 이번 프로젝트에서는 사물인터넷 과목을 가르치고 계신 교수님이 추천 해 주신 MQTT 프로토콜에 대해 학습하여 프로젝트를 진행하였다.

MQTT 프로토콜은 메시지를 발행 (publishing) 하고, 관심 있는 주제를 구독(subscribe) 하는 것을 기본 원칙으로 한다.

Publisher과 Subscriber은 모두 Broker에 대한 클라이언트로 작동한다. Publisher는 토픽을 발행하기 위한 목적으로 Subscriber은 토픽을 구독하기 위한 목적으로 Broker 서버에 연결한다.

하나 이상의 Pub와 Sub가 브로커에 연결해서 토픽을 발행 하거나 구독할 수 있다. 또한 다수의 클라이언트가 하나의 주제를 구독할 수도 있다.

## 구성요소 - 통신



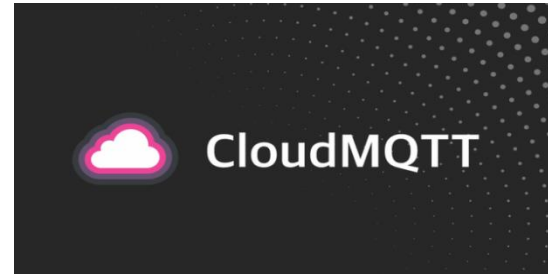
## 구성요소 - 통신

MQTT는 시스템에 참여하는 장치들의 처리능력, 네트워크 대역폭, 메시지 오버헤드 등 주변상황에 맞게 시스템이 동작할 수 있도록 **3단계 QoS(Quality of Service)**를 제공한다.

QoS value	Description
0	메시지는 한번만 전달. 전달여부를 확인하지 않음
1	메시지는 한번 이상 전달. Publsih를 송신하면 Puback 메시지에 의해 확인 응답이 가능하다.
2	메시지는 한번만 전달. 메시지의 핸드셰이킹 과정을 추적. 높은 품질을 보장하지만 성능의 희생을 따른다.

0에 가까울수록 메시지 처리에 대한 부하가 적은 대신 메시지 손실 위험이 높아진다.  
2에 가까울수록 메시지 손실위험은 줄어들지만 메시지 처리 부하가 급격히 늘어난다.  
(보통 0~1 정도의 QoS를 사용함)

## 구성요소 - 통신



### CloudMQTT

- Amazon에서 제공하는 MQTT 프로토콜 통신방식의 Cloud 서비스로, MQTT를 컴퓨터에 직접 깔지 않아도 온라인에서 MQTT를 사용할 수 있다.

### 중계서버(Broker) 역할

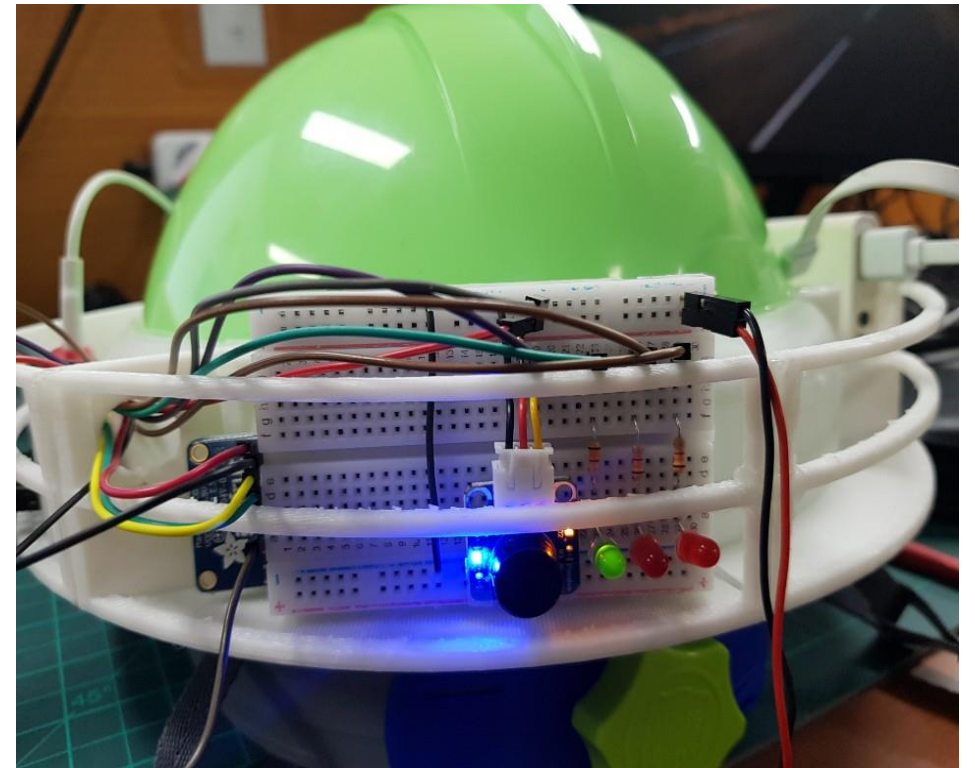
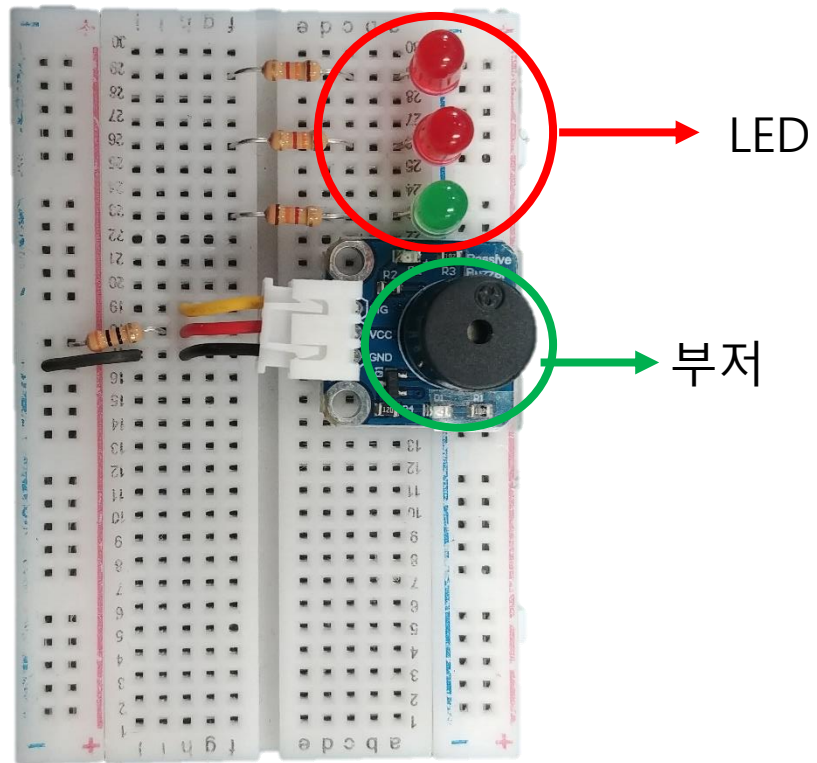


## 하드웨어 구성 - 헬멧

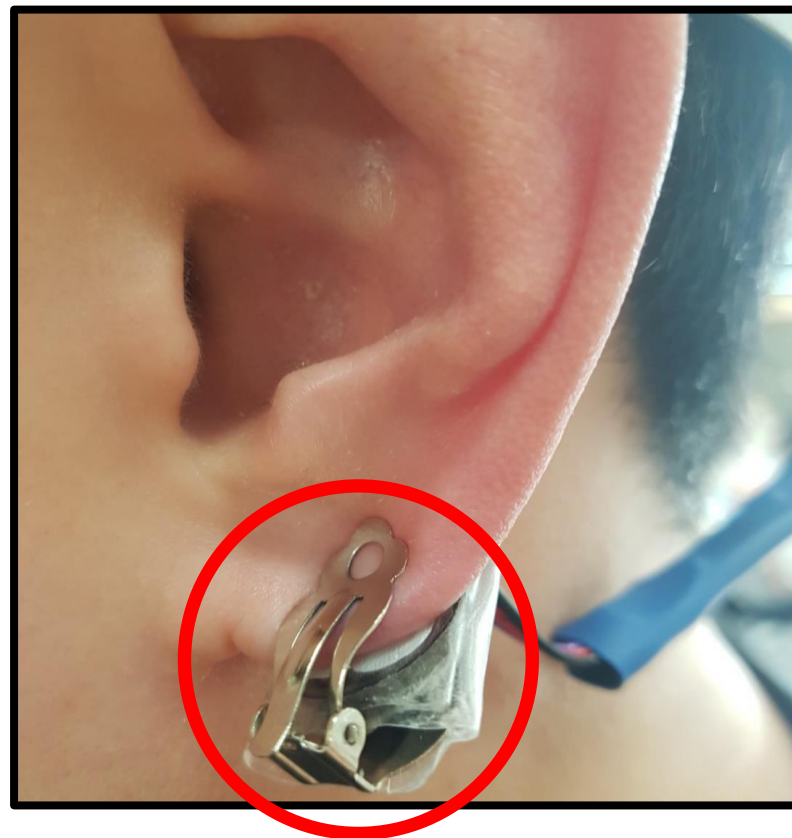
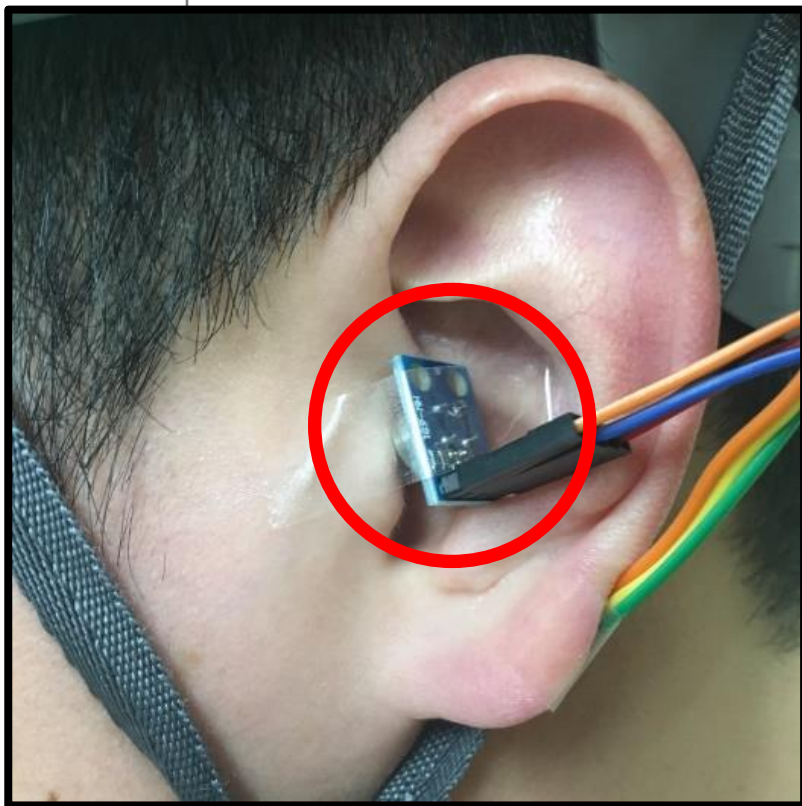




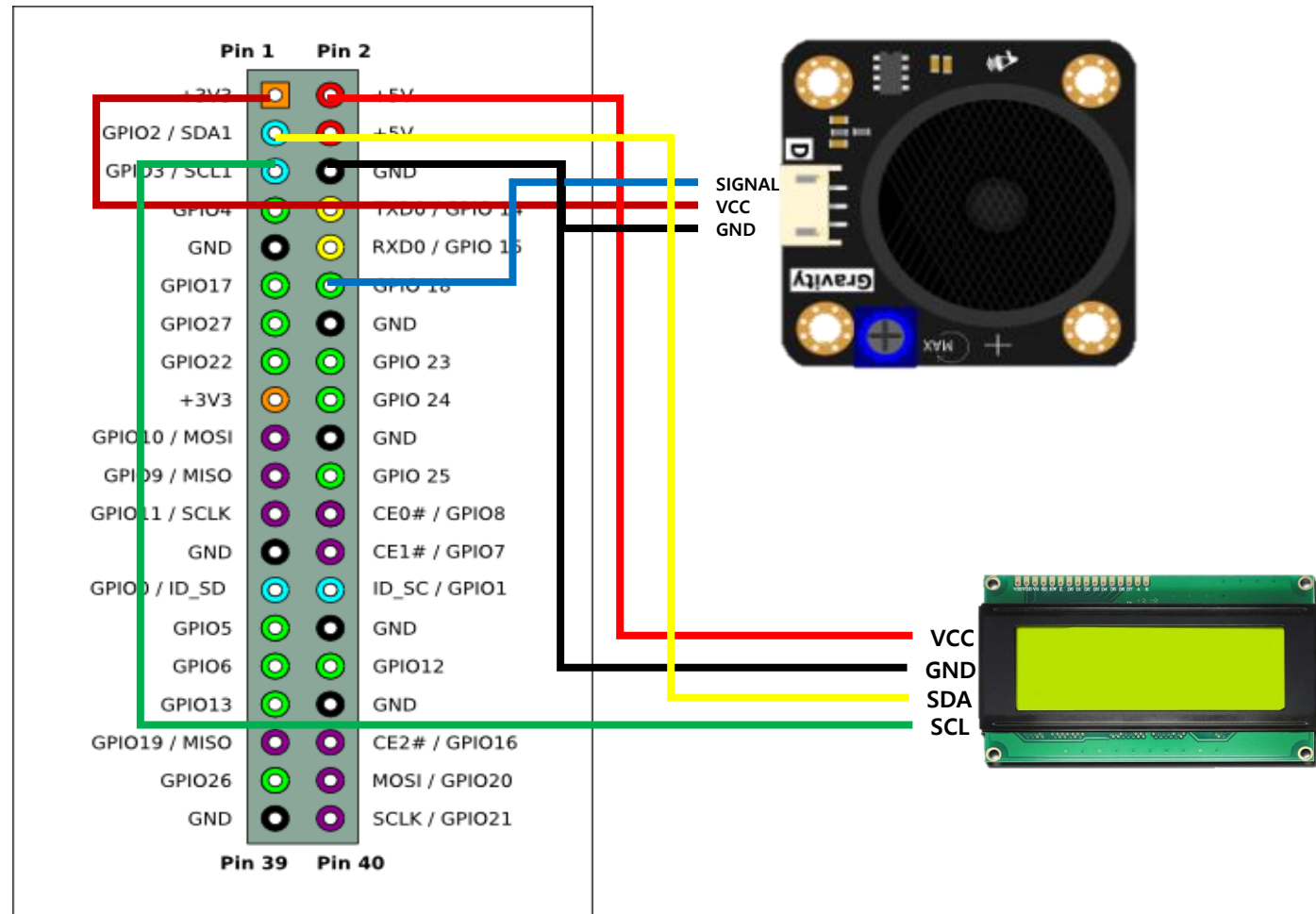
## 하드웨어 구성 - 헬멧



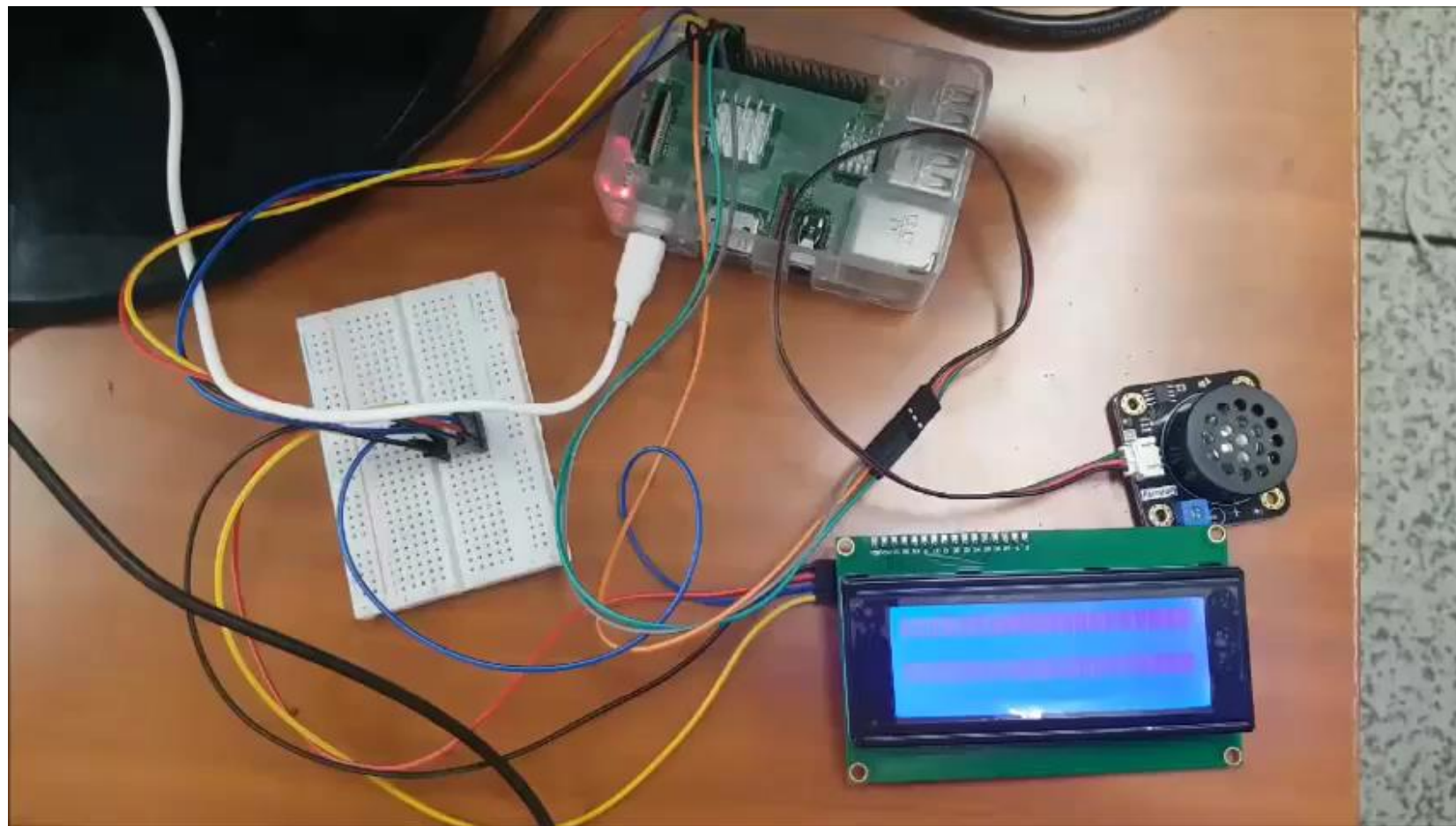
## 하드웨어 구성 - 헬멧



## 하드웨어 구성 - 알람

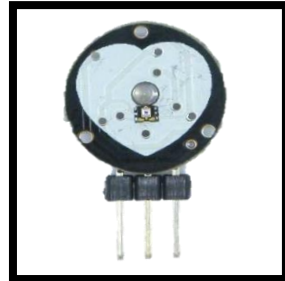


## 알람 작동 영상



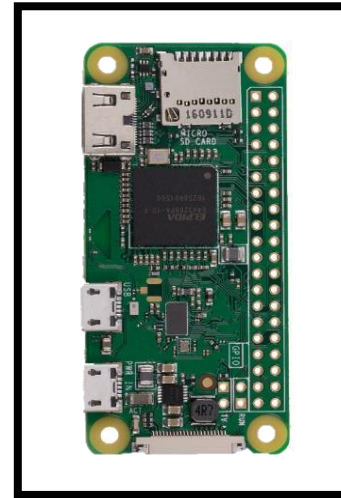
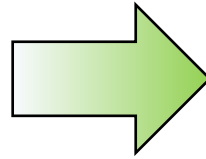


## 시스템 알고리즘



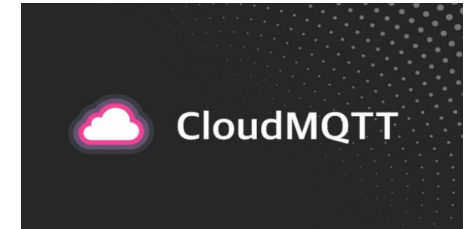
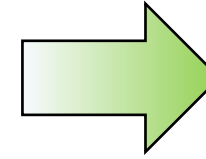
### Senser

- 생체정보 수집



### Zero W

- 데이터 변환 후 서버로 송신
- 근로자의 이상 상태 유무 판별



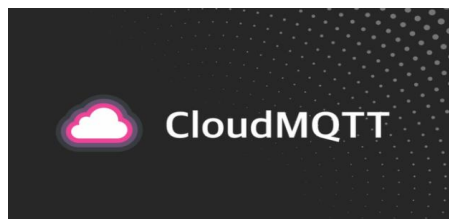
Topic1 : BH/temp  
Topic2 : BH/Pulse  
...

### CloudMQTT

- Topic 생성
- 데이터 저장 및 실시간 출력

## 시스템 알고리즘

CloudMQTT  
(컴퓨터)



Topic1 : BH/temp1  
Topic2 : BH/temp2  
...

subscribe

LTE

data



스마트폰

App을 통해  
실시간으로 갱  
신되는 데이터  
확인 가능

subscribe

Wifi

data



Alarm

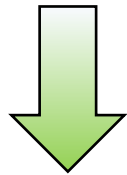
실시간으로  
갱신되는 데이  
터로 이상상태  
유무 판별

# PORTFOLIO

## 시스템 알고리즘

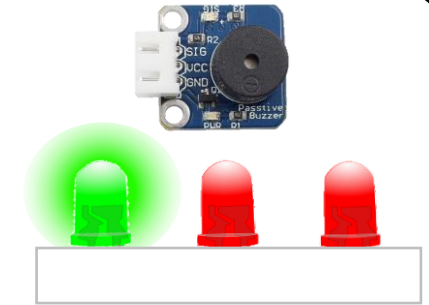
### 평상시

심박수 : 60~100  
체온 : 38.5°C ↓

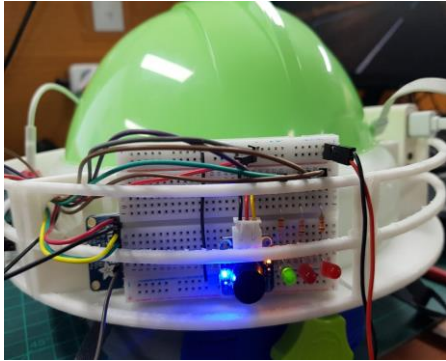


초록색 LED On

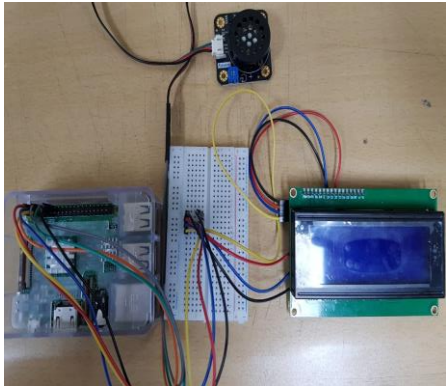
HELMET



DESK

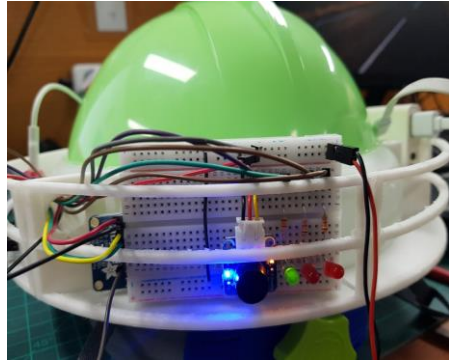


HELMET 장비

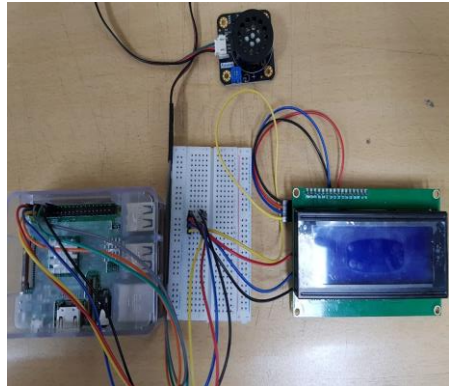


DESK 장비

# PORTFOLIO



HELMET 장비

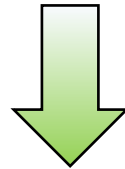


DESK 장비

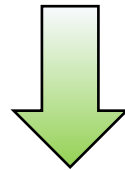
## 시스템 알고리즘

### 이상상태

심박수 : 50 ↓ or 100 ↑  
체온 : 38.5°C ↑

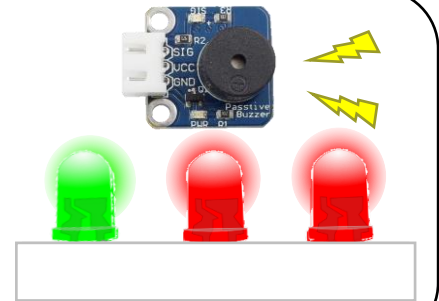


카운팅 15회

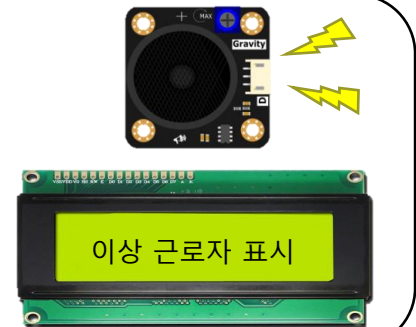


비상알람 동작

HELMET

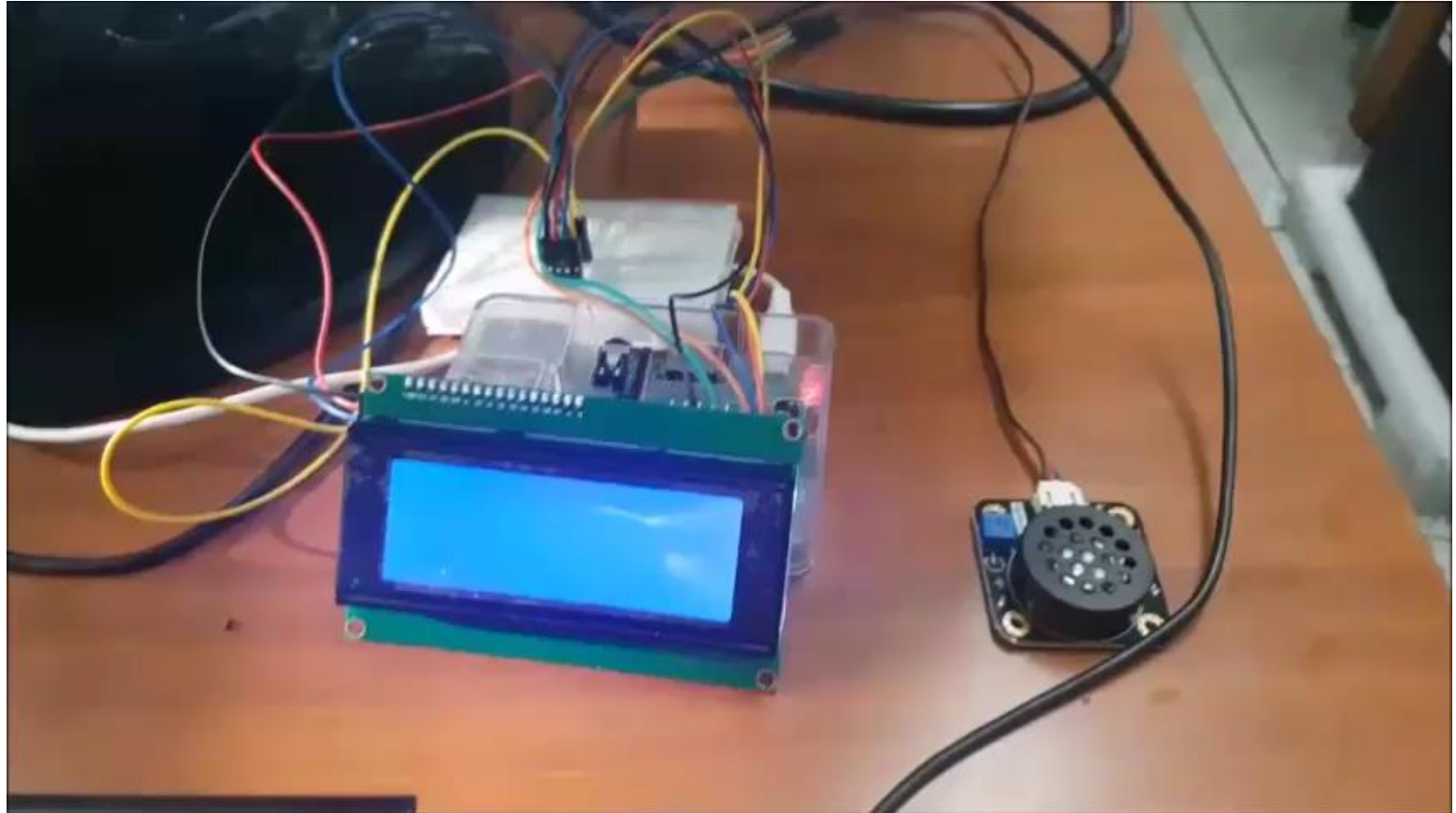


DESK





## 시연 동영상



## 기대효과 및 활용방안



안전모에서 전송되는 데이터를 바탕으로  
근로자들의 건강상태를 효율적으로 관리



현장 근로자들의 안전사고율 감소 및  
응급상황 발생시 빠른 대처 가능



화재 현장



수색 & 구조 현장

건설 현장뿐 만 아니라 작품에 사용된 기술을 바탕으  
로 현장, 상황 등에 맞춰 다양하게 활용 가능

## 기대효과 및 활용방안



GPS



카메라 영상

안전모에 GPS 또는 카메라를 탑재할 수 있다면  
사고가 발생한 사람의 위치와 상황을  
좀 더 쉽게 파악할 수 있어 더욱 빠른 조치 가능



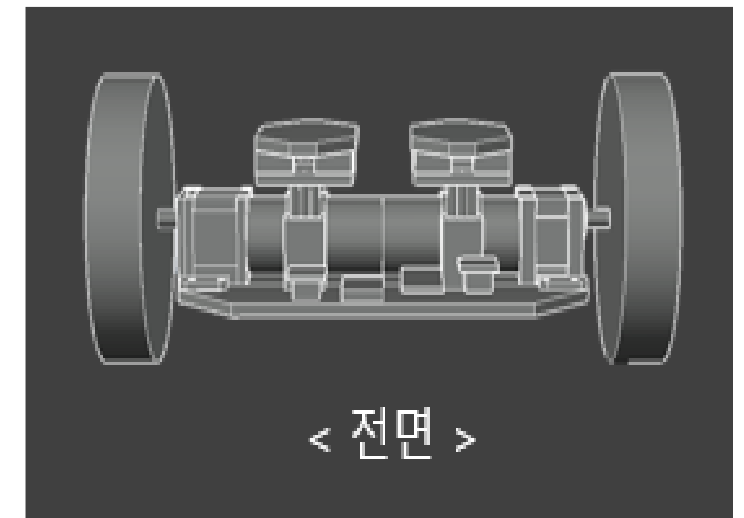
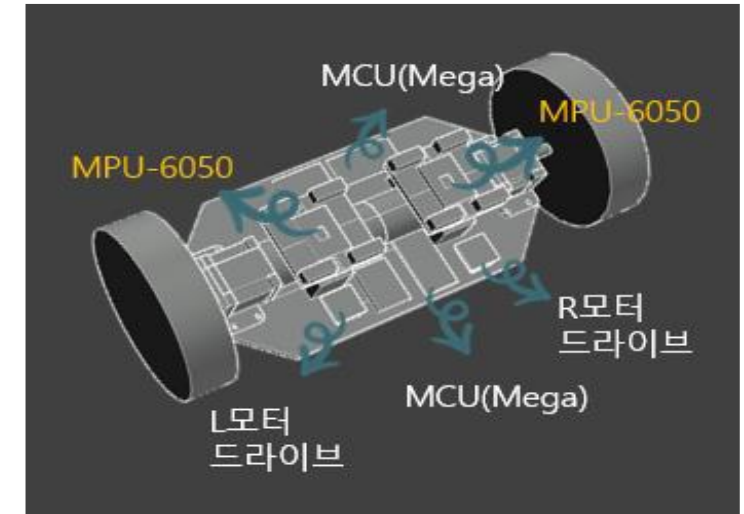
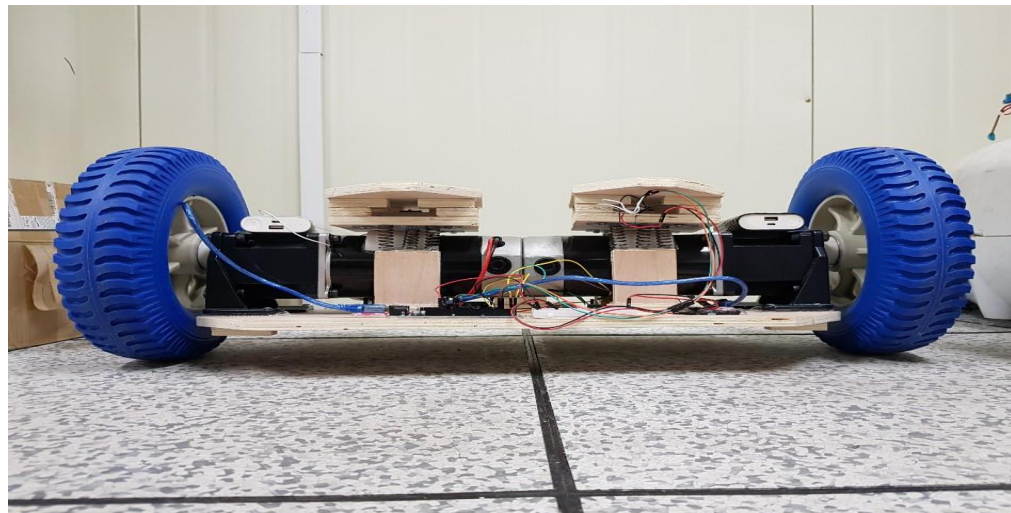
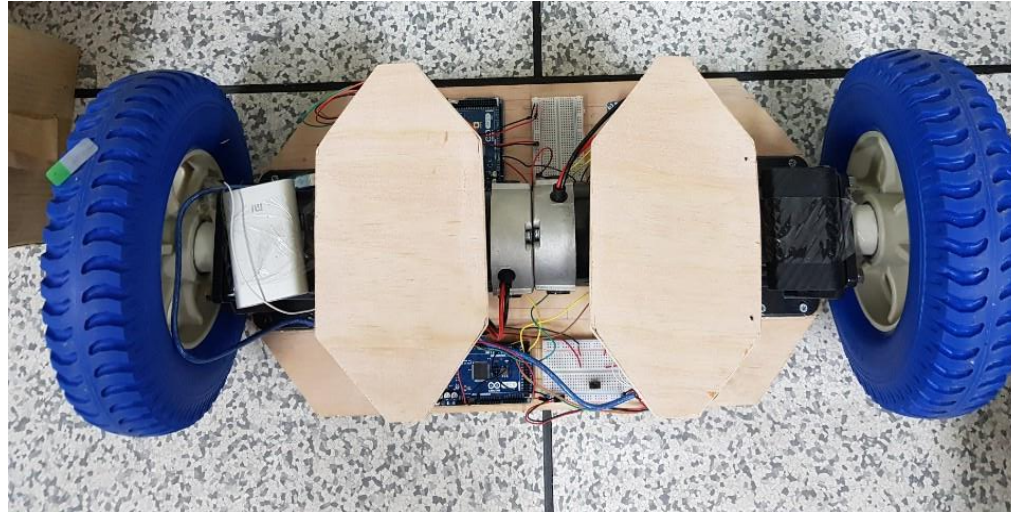
## PORTFOLIO

### 기여도

#### 작품설명 (기획포인트)

몸의 기울어짐에 따라 속도가 변하고, 양쪽 바퀴의 속도에 차이를 주어 방향을 제어하고 차체가 넘어지지 않도록 밸런싱을 하는 휴대가 가능한 1인용 이동수단.

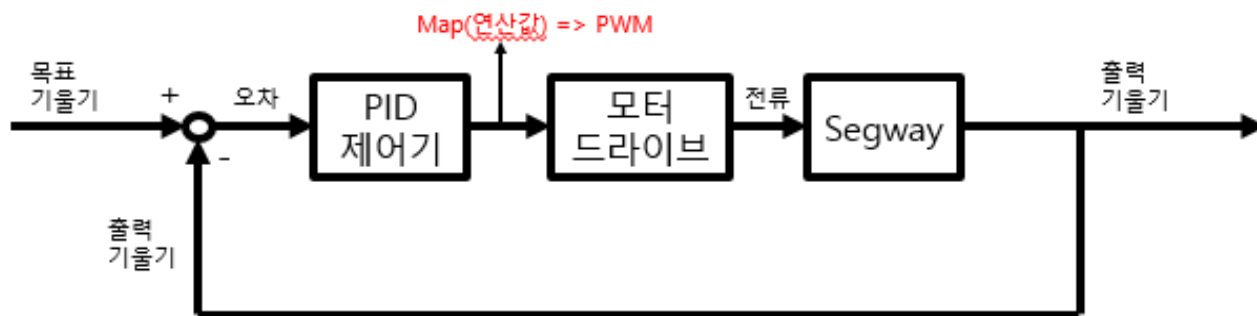
### SEGWAY



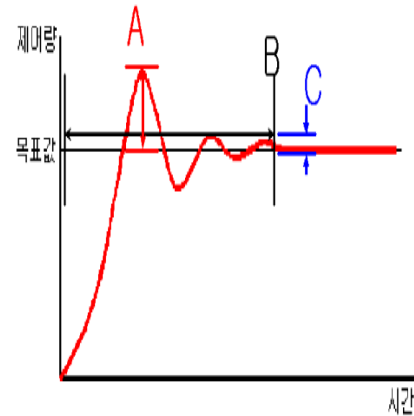
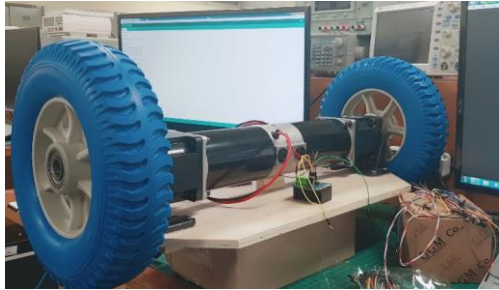
## 프로젝트 목표

개발 및 판매 당시 새로운 기술을 이용한 이동수단의 발명은 호평을 받았지만, 고가의 판매가로 인하여 상용화에 실패하였다. 하여 상용화를 위한 생산단가를 낮추는 방법으로 나무와 스프링 등 저가의 재료를 이용하여 하드웨어를 구성하고, 소프트웨어적으로 동작알고리즘을 세밀하게 구성하여 부족한 안정성을 보완하고 기존의 제품과 비교해도 뒤쳐지지 않을 만큼의 성능을 발휘할 수 있도록 제작하는 것을 목표로 하였다.

## 기술

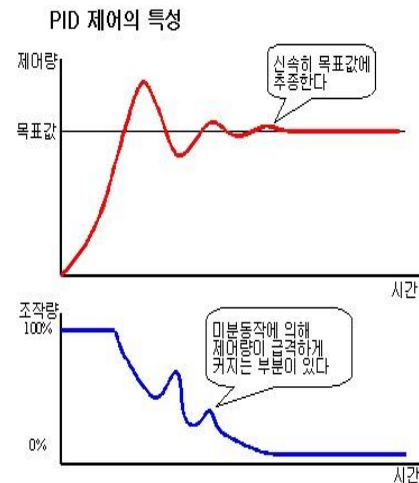


## PORTFOLIO



## PID 제어의 특성

- A. **Overshoot – 오버 슈트** : 목표 값을 기준으로 오차가 최고로 커지는 부분의 값을 말한다. (이 값이 너무 커지면 시스템에 무리를 줄 수도 있다.)
- B. **목표 값 도달 시간** : 목표 값의 몇 %에 들어가면 제어가 완료된 것으로 보는데 그 범위안에 들어가서 제어가 완료되었다고 보는 시간을 말한다.
- C. **정상상태 오차** : 제어량이 목표량의 일정범위에 도달하여 제어는 끝났지만 없어지지 않고 남아 있는 오차를 말한다.



## PID 제어 각 요소의 역할

P 제어 : 목표 값 도달 시간(B)을 줄인다.

I 제어 : 정상상태 오차(C)를 줄인다.

D 제어 : 오버 슈트(현재치의 급변이나 외란-A)을 억제한다.

## PORTFOLIO

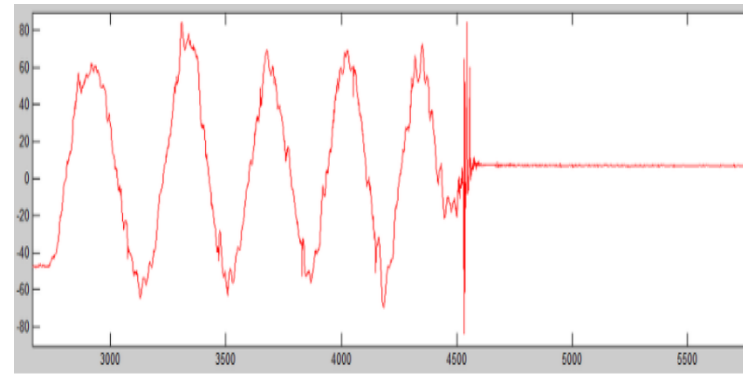
## • 학습내용

MPU-6050 센서에 대해 학습  
하고 장단점을 알아보았다.



MPU-6050

## 가속도 센서



&lt;가속도센서 값을 pitch로 변환한 후 출력&gt;

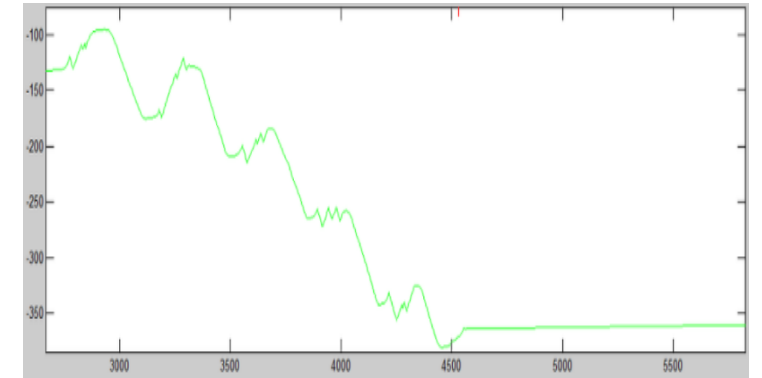
시간이 오래 지나도 오차가 누적되지 않는다.

짧은 시간의 변화에 대해서는 불안정한 값을 준다.

장점

단점

## 자이로 센서



&lt;자이로센서 값을 pitch로 변환한 후 출력&gt;

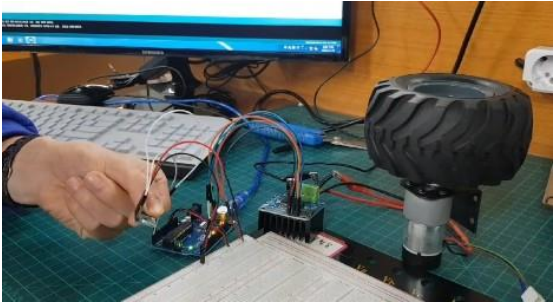
짧은 시간의 변화에 대해서는 안정적이고 정확한  
데이터를 제공한다.

오랜 시간에 걸친 적분에 의해 방향에 대한  
누적오차가 발생한다.

## PORTFOLIO

## • 학습내용

프로젝트에서는 상보 필터와 칼만필터를 조합해서 얻은 Complementary 값을 사용하였다.

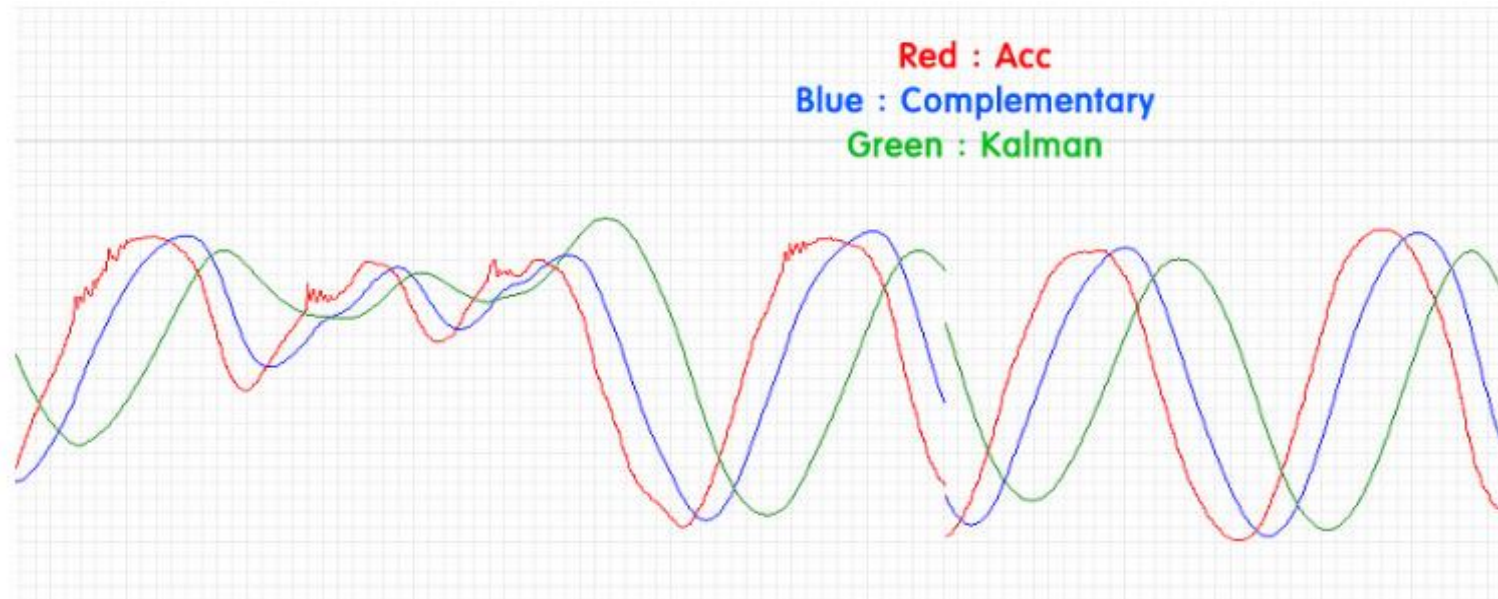


## ■ 상보 필터

다른 주파수 특징을 가진 센서로부터의 노이즈를 최소화 시키기 위해서 사용된다.

## ■ 칼만 필터

과거의 정보와 새로운 측정값을 사용하여 측정값에 포함된 잡음을 제거시켜 최적의 값을 추적하는데 사용하는 알고리즘.





### ■ 문제점

1. 처음에 선정 했던 DC 36V 250W 모터가 사람이 탑승하여 운행함에 있어 무리가 있을 것 같다는 판단이 있었다.
2. 자이로센서에서 검출된 각속도 값을 이용하여 pitch값을 도출한 경우, 기울기의 변화에 대해 서는 가속도센서에 비해 보다 정밀한 값을 출력하지만, 시간이 지남에 따라 적분으로 인한 오차가 누적되어서 드리프트 현상이 발생하는 것을 알 수 있었다.
3. 발판을 하중을 가하면 기울어지도록 하드웨어를 설계함에 있어 아이디어적으로 토의가 필요했다.
4. 주문한 바퀴에 베어링이 없어 제작이 필요했다.

### ■ 해결

1. DC 24V 300W 3000RPM과 기어박스(감속 비 1:10)를 추가시킨 모터로 변경하기로 결정.( 2개: 약 32만원)
2. 상보필터와 칼만필터라는 것을 사용해서 보다 안정적인 기울기 값을 도출해내기로 했다.
3. 생산단가를 낮추는 방법으로 나무와 스프링으로 저가의 재료를 이용하여 하드웨어를 구성했다.
4. 3D 모델링을 통해 베어링을 제작해 사용했다.

## PORTFOLIO

## 기여도

## 작품설명 (기획포인트)

N404보드 4축 모션제어보드를 이용한 PC기반 3축데스크탑 로봇을 이용하며, MFC를 이용하여 프로그래밍을 하며, 대칭 S자형 가감속 프로파일을 사용한다. CAD로 2D 모델링을 하고 각 좌표를 선정해 값을 기입한다.

먼저, 전체적인 틀을 잡기 위해 큰 방패와, 중앙을 지나는 링의 틀을 잡는다. 대문자 WB를 그리고 중앙링의 WARNER BROS.부분을 그린다.

## 모션제어 프로젝트

## B-5조

20133070 김우석  
20133020 최보웅  
20142988 김도형  
20163019 박현진  
20163035 조문영

# 목차

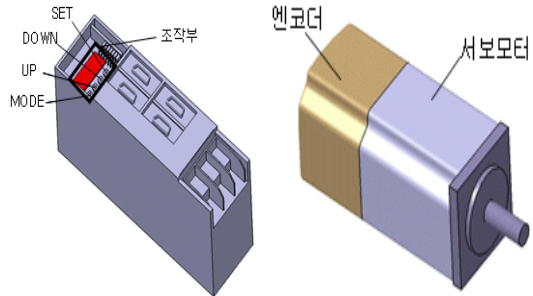


1. 모션제어기 시스템 구성
2. 주제 및 설계
3. 문제점 및 해결
4. 구동 사진 및 영상
5. 완성 사진

## PORTFOLIO

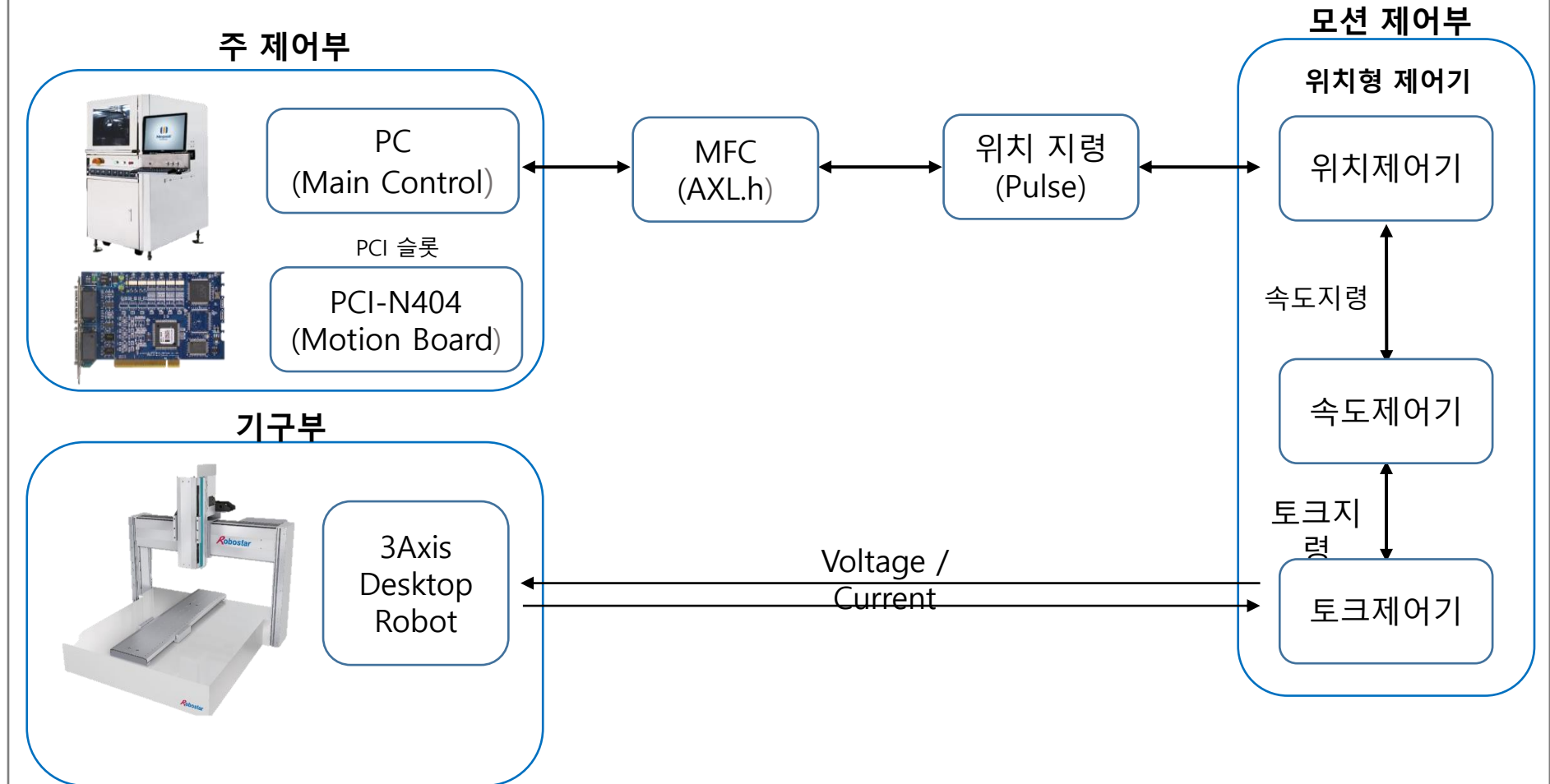
### • 학습내용

PC에서 MFC로 모션 프로그램밍을 하고 모션 프로그램밍 데이터를 모션 제어부로 보낸다. 모션 제어부에서는 모션 제어기를 이용하여 서보 데이터를 서보 드라이버와 교환하고 3축 로봇에 전압과 전류를 보내서 동작시킨다. 3축 로봇의 엔코더 값과 센서의 값들을 서보 드라이버로 받아서 제어한다.

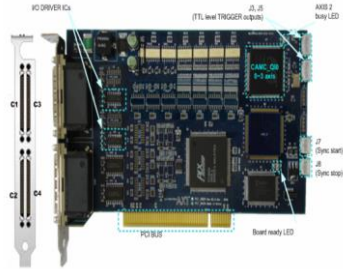
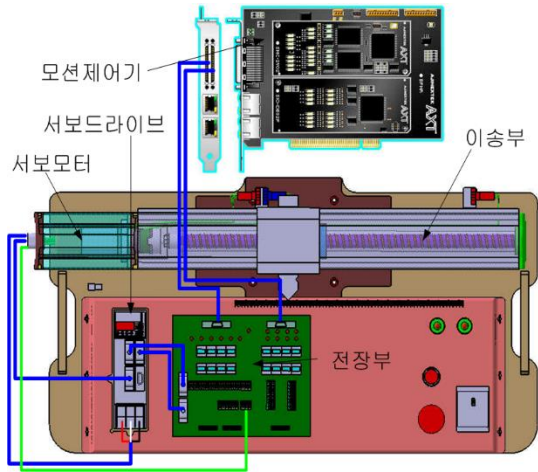


서보드라이브와 서보모터

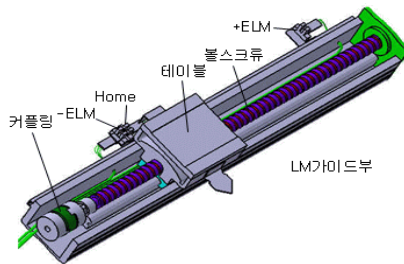
# 1. 모션제어기 시스템 구성



## PORTFOLIO



PCI-N404보드와 커넥터



직선 이송부

# 1. 모션제어기 시스템 구성

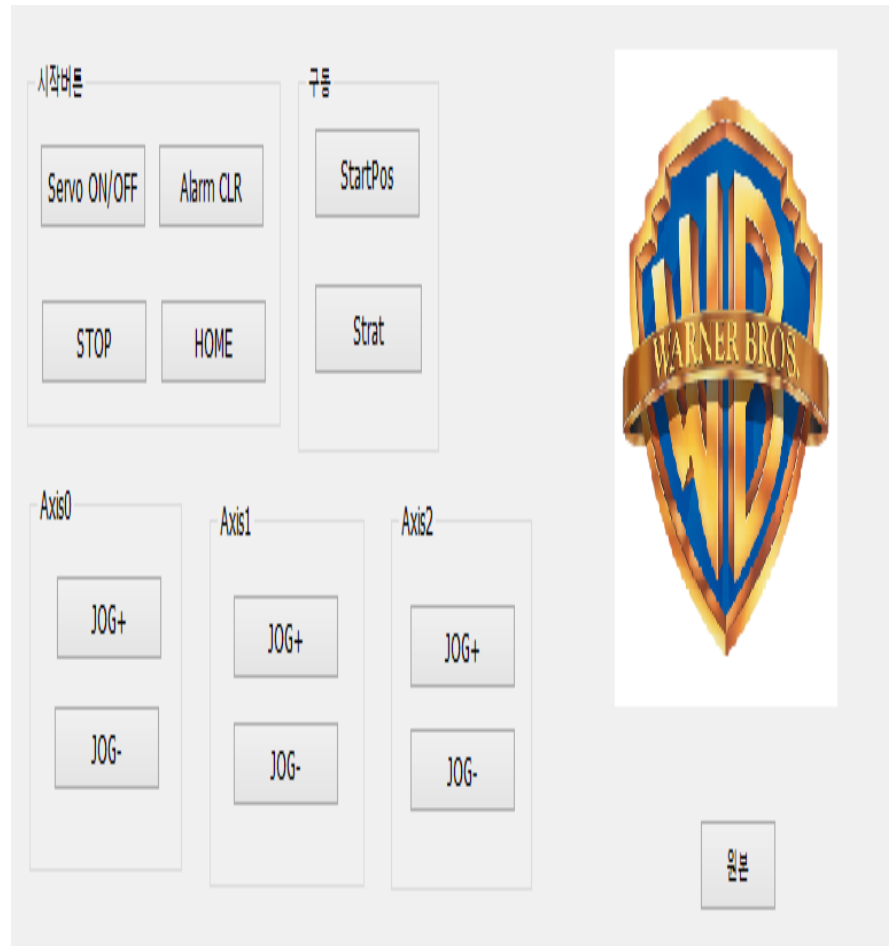
- CAMC-QI 장착  
(4축 펄스 출력, PCI Half size보드)
- 정상동작 상태 확인을 위한 LED, 커넥터, 내부 TTL Level Trigger 신호 연결용 헤더 점퍼로 구성
- Max. 10Mpps 고속정밀 모터 제어를 쉽고 간단하게 구현 가능
- 특정 위치에서의 트리거 및 인터럽트를 생성, 그리고 서보모터 드라이브의 잔여 펄스를 제거하는 기능
- 직선 이송부(Linear Motion Part): 직선운동이 발생하는 부분
- 회전운동을 직선운동으로 변환하는 볼 스크류 이송부
- 서보모터와 스크류 축을 연결하는 커플링
- 기구 부 신호
  - 1) 홈 센서: 테이블의 원점을 나타내는 센서
  - 2) 엔드리밋 센서: 테이블이 기구적으로 갈 수 있는 한계위치를 나타내는 센서로 -ELM와 +ELM가 있다

## 2. 주제 및 설계

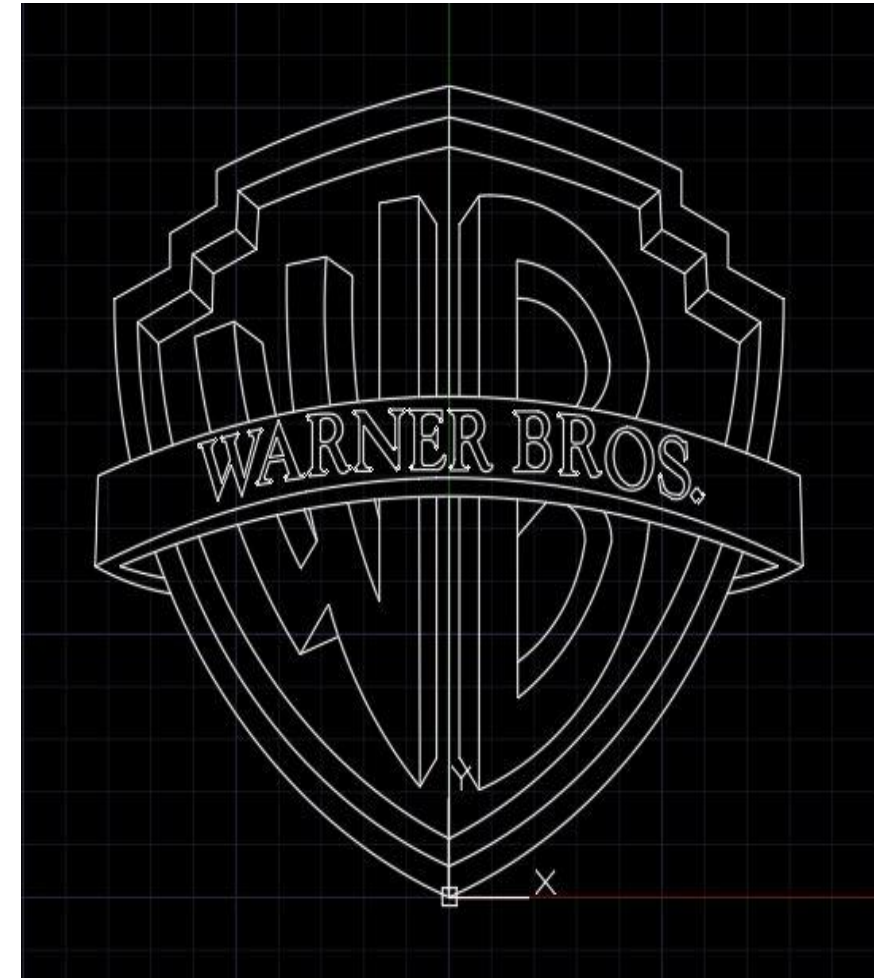


## 2. 주제 및 설계

〈구동 TOOL〉

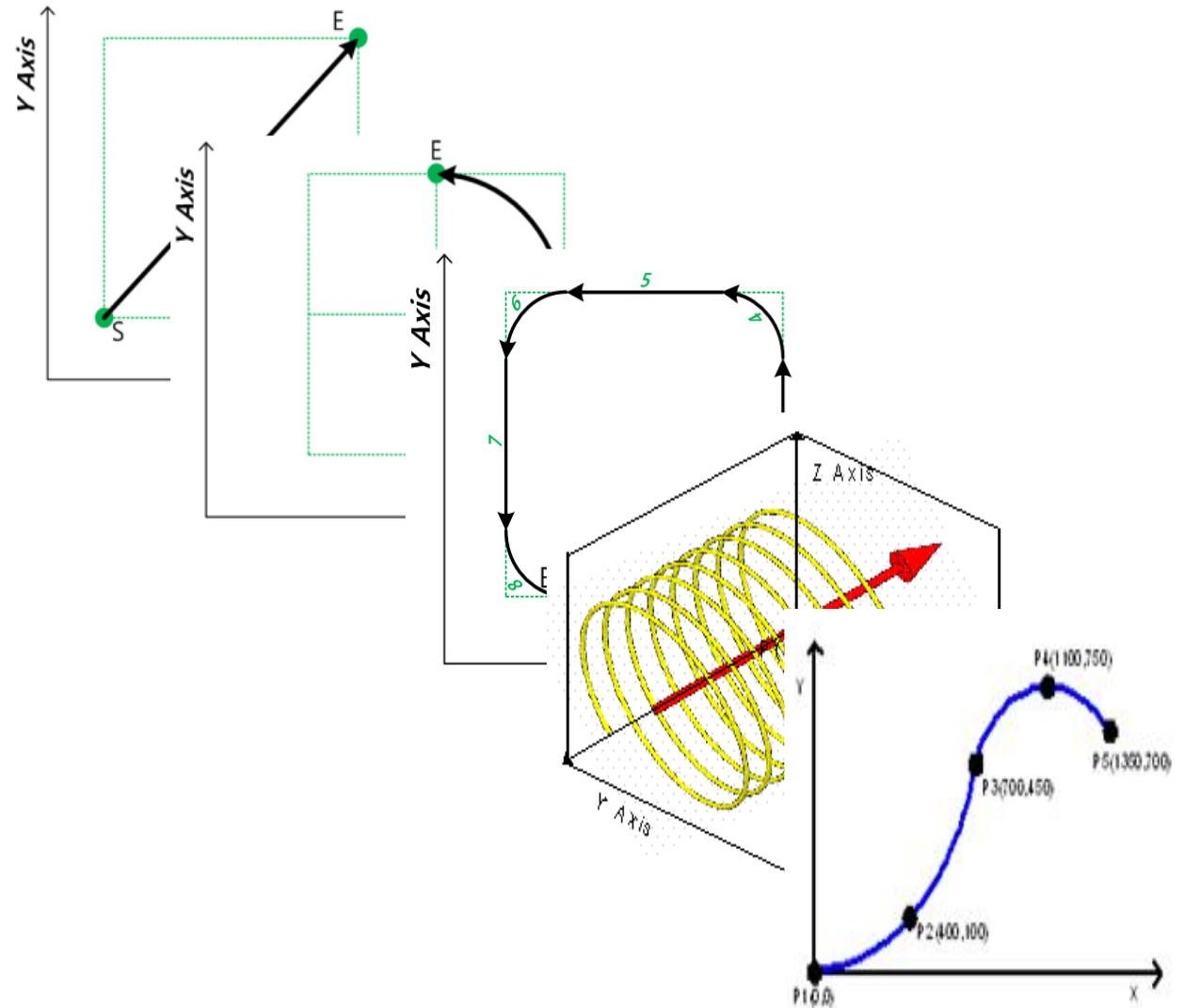


〈CAD 그림〉



## 2. 주제 및 설계

보간
직선보간
원호 보간
연속보간
헬리컬보간
스플라인 보간





## PORTFOLIO

## • 학습내용

유연한 모션제어를 위해 대칭 S자형 가감속 프로파일을 사용하며, 형상을 디자인하기 위해 라이브러리의 함수를 이용하여 직선 보간 구동, 원호 보간 구동, 오버드라이버 구동 등을 사용한다.

## 변수 설정

AXIS0, 1, 2 사용

AXIS2 브레이크 해제

Limit → LOW

InPosition → HIGH

ServoAlarm → LOW

Unit/Pulse → 1/1000(mm)

Min Velocity → 1

출력펄스형태 → TwoCwCcwHigh

엔코더 신호 → 4체배

AbsRelMode → 상대위치 구동

Profile Mode → 대칭S자구동

## 2. 주제 및 설계

## GUI

HOME

StartPos

Start

## 모션 구동

보간 큐 초기화

좌표계 축 설정

좌표계 구동 모드 설정

보간 노드 등록 시작

AxmLineMove  
AxmCircleRadiusMove

보간 노드 등록 종료

보간 구동 시작

## 2. 주제 및 설계

### 〈START 버튼을 눌렀을 때 동작코드〉

```
void C모션B5조Dlg::OnBnClickedStart()
```

```
{
    UpdateData(TRUE);
    long AXIS[3];
    double Endpos[3];
    double LEndpos[3];
    DWORD uShortDistance=0;
    long ICoordinate=0;
    long IPosSize=3;
    double Radius;
    double LVelocity=15;
    double LAccel=15;
    double Velocity=15;
    double Accel=15;

    AXIS[0] = 0;
    AXIS[1] = 1;
    AXIS[2] = 2;
    AxmContiWriteClear(ICoordinate);
    AxmContiSetAxisMap(ICoordinate, IPosSize, AXIS);
    AxmContiSetAbsRelMode(ICoordinate, 1);

    AxmContiBeginNode(ICoordinate);
```

직선보간 속도

직선보간 가속도

원호보간 속도

원호보간 가속도

### 〈작은 글씨를 쓸 때 낮춰준 속도〉

```
LVelocity=5;
LAccel=5;
Velocity=5;
Accel=5;
//작은
```

### 〈원호보간〉

```
Endpos[0] = -11.6; Endpos[1] = -44.67; Endpos[2] = 0; Radius = 299.44;
AxmCircleRadiusMove(ICoordinate, AXIS, Radius, Endpos, Velocity, Accel, Accel, 0, uShortDistance);
```

0축 위치

1축 위치

2축 위치

반지름

### 〈직선보간〉

```
LEndpos[0] = 0; LEndpos[1] = 0; LEndpos[2] = -10;
AxmLineMove(ICoordinate, LEndpos, LVelocity, LAccel, LAccel);
```

0축 위치

1축 위치

2축 위치

### 〈축 이동〉

```
LEndpos[0] = 0; LEndpos[1] = 0; LEndpos[2] = 10;
AxmLineMove(ICoordinate, LEndpos, LVelocity, LAccel, LAccel);
```

```
LEndpos[0] = -17.3; LEndpos[1] = 6.12; LEndpos[2] = 0;
AxmLineMove(ICoordinate, LEndpos, LVelocity, LAccel, LAccel);
```

```
LEndpos[0] = 0; LEndpos[1] = 0; LEndpos[2] = -10;
AxmLineMove(ICoordinate, LEndpos, LVelocity, LAccel, LAccel);
```

### 3. 문제점 및 해결

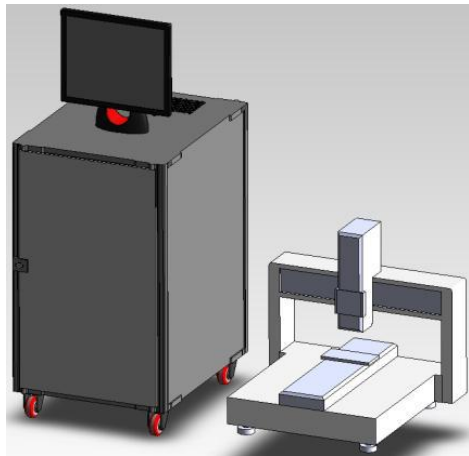
#### ▶ 문제점

1. 속도 문제- 링 안의 작은 글씨를 그리는 속도를 같은 속도로 빠르게 하면 글씨가 잘 나오지 않는 문제 발생
2. 펜 문제- 펜의 굵기가 굵으면 글씨가 나타나지 않고 얇으면 방패의 그림이 연하게 나타나는 문제 발생

#### ▶ 해결

1. 속도문제- 큰 그림을 그릴 때와 글씨를 그릴 때의 속도에 차이를 두어 문제 해결
2. 펜 문제- 여러 굵기의 펜을 사용하여 그림을 그려보고 알맞은 펜으로 선정

# PORTFOLIO

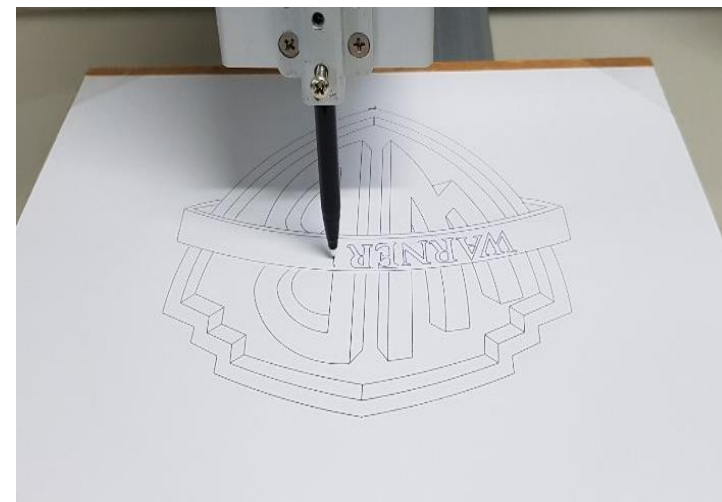
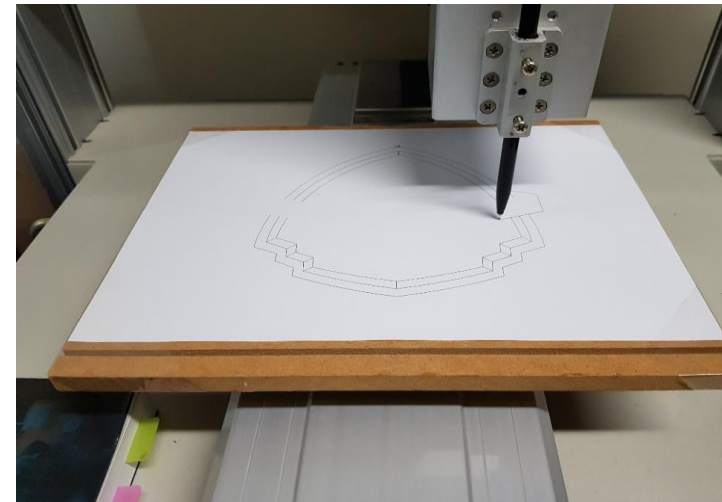


3축 데스크탑 로봇

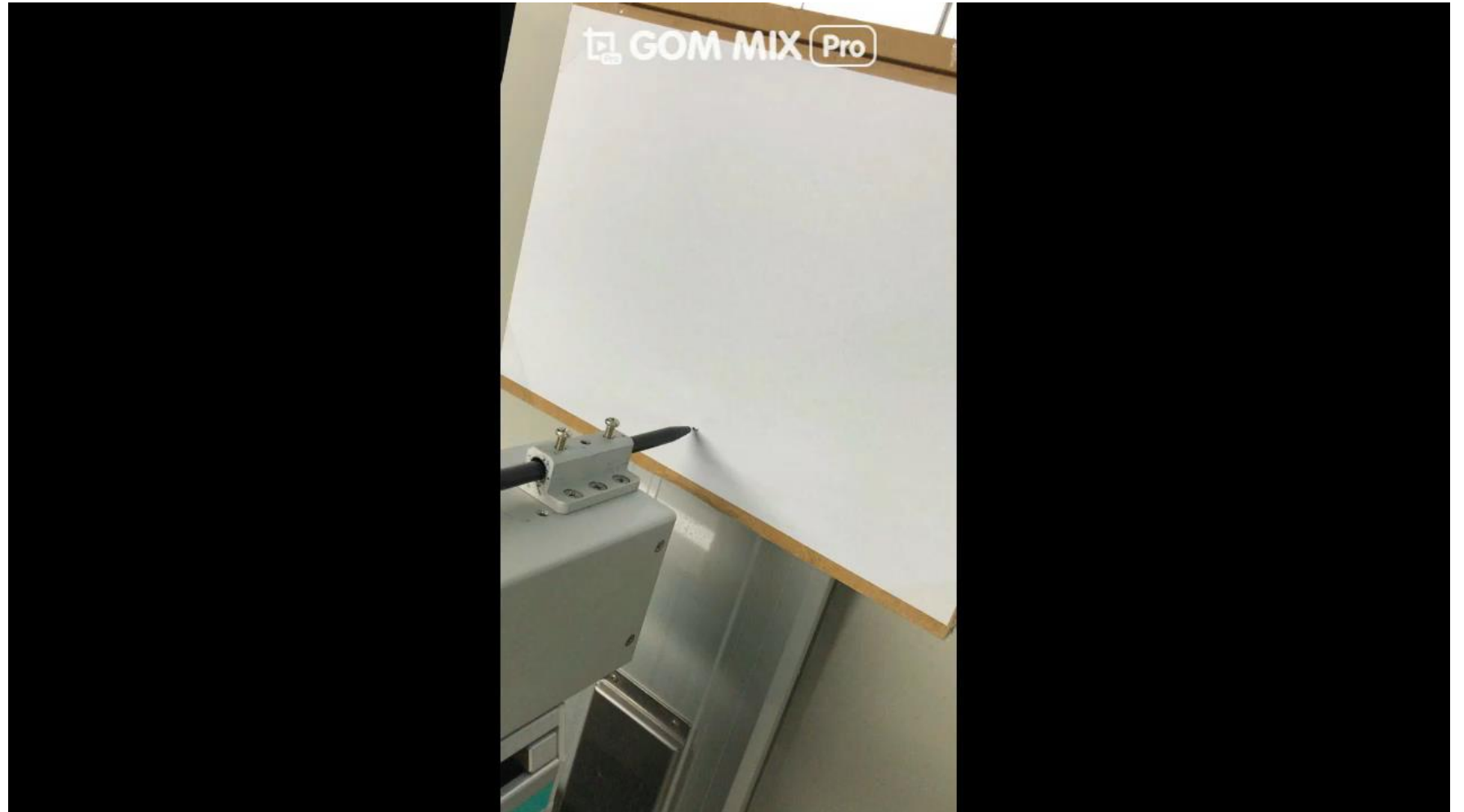
## 구동 메뉴얼 및 구동사진

모션 B-5조

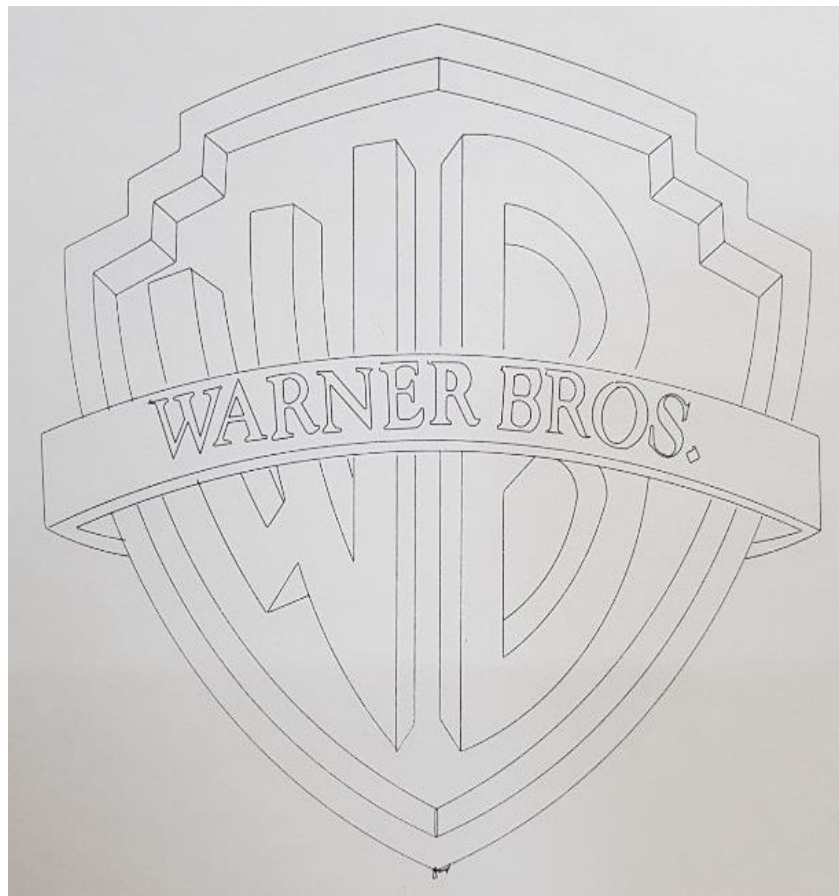
3축동시제어				시작 버튼	
상대 위치 구동		절대 위치 구동		Servo On/Off	
				Alarm CLR	
				Stop	
				Home	
AXIS_0					
0축 위치		0축 가속		상대 위치 구동	JOG+
0축 속도		0축 감속도		절대 위치 구동	JOG-
AXIS_1					
1축 위치		1축 가속도		상대 위치 구동	JOG+
1축 속도		1축 감속도		절대 위치 구동	JOG-
AXIS_2					
2축 위치		2축 가속도		상대 위치 구동	JOG+
2축 속도		2축 감속도		절대 위치 구동	JOG-
구동					
Strat Pos					
직선				원	
				Start	
				확인	
				취소	



## 4. 구동 영상



## 5. 완성사진





## ■ 문제점

1. 속도 문제- 링 안의 작은 글씨를 그리는 속도를 같은 속도로 빠르게 하면 글씨가 잘 나오지 않는 문제 발생
2. 펜 문제- 펜의 굵기가 굵으면 글씨가 나타나지 않고 얇으면 방패의 그림이 연하게 나타나는 문제 발생

## ■ 해결

1. 속도문제- 큰 그림을 그릴 때와 글씨를 그릴 때의 속도에 차이를 두어 문제 해결
2. 펜 문제- 여러 굵기의 펜을 사용하여 그림을 그려보고 알맞은 펜으로 선정

THANK YOU