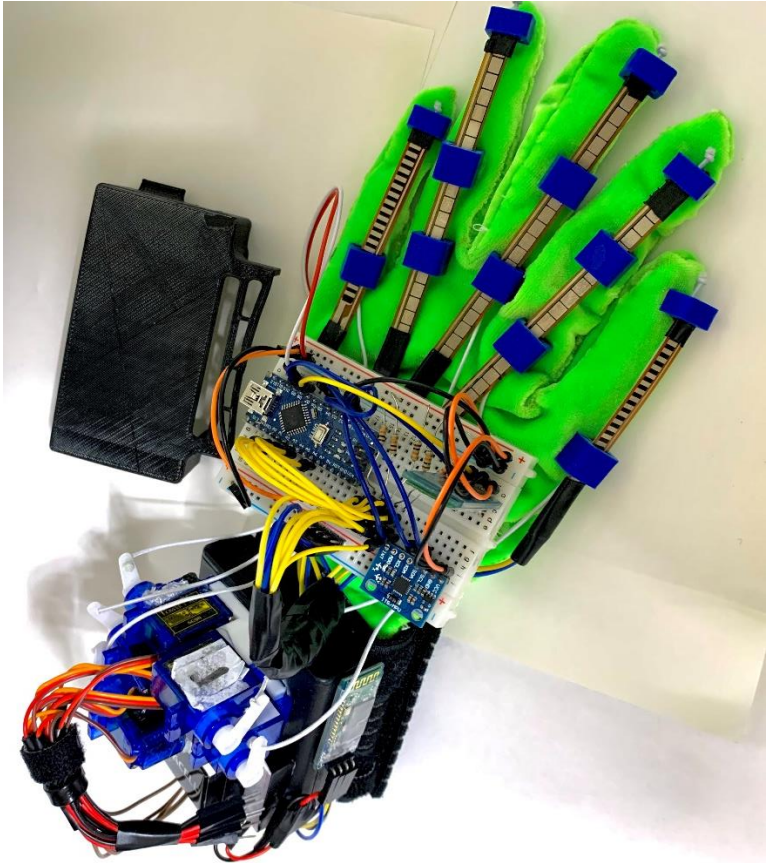


제18회 임베디드SW경진대회 개발완료보고서

[자유공모]

□ 개발 요약

팀명	VUHWAR
	
작품명	gloVR
작품설명 (요약)	뇌졸중 환자의 상지재활을 돕기 위한 장갑형 악력 컨트롤러
소스코드	https://github.com/qkrwjdan/2020ESWContest_free_1097/
시연동영상	https://www.youtube.com/watch?v=EcmVe9Gge1I

작품
기능설명

우리는 뇌졸중 환자의 효과적인 재활치료를 위해 게임 치료에 손가락 별 저항력을 적용시킨 장갑 형태 3D 게임 컨트롤러 gloVR을 제작했다. **gloVR은 Flex 센서, IMU 센서, 카메라 모듈을 이용하여 손가락의 움직임을 파악하고** Unity Engine과 블루투스 통신하여 값을 전달한다. 뇌졸중 환자의 재활을 돕기 위해 **탄성 줄을 이용하여 손에 저항력을 발생시키도록 설계하였다.** 사용자는 5개의 서보 모터를 통해 손가락마다 적용되는 저항력을 변경시킬 수 있다.

이는 사용자가 직접 조절하거나 게임 속 상황에 따라 달라질 수 있다.

gloVR 컨트롤러에 맞는 게임(캐치볼 게임)을 자체 제작하여 재활효율을 극대화할 수 있도록 하였다. 컨트롤러의 움직임(손의 위치값(x,y,z), 손가락 별 구부린 정도, 손의 회전각)을 파악하여 화면상에 나타낼 수 있게 하였고, 날라오는 야구공을 잡는 게임을 개발하였다.

gloVR의 게임 콘텐츠는 Unity 3D 엔진으로 설계되어 VR 게임으로 발전할 수 있는 장점이 있다. **VR 게임과 gloVR 컨트롤러를 통한 상지 재활은 뇌졸중 환자의 재활 효율을 더욱더 높여줄 수 있을 것이다.**

□ 개발 개요

○ 개발 작품 개요

2019년 기준 국내 뇌졸중 환자는 69 만 명이고, 매년 10.5 만 명의 환자가 발생하는 것으로 추이되었다. 이 점에 더하여 뇌졸중으로 인한 사망률이 감소하며 재활에 대한 수요가 증가하고 있다. 뇌졸중 환자의 90%는 편마비가 나타나 상지 재활이 필수적이다.

우리는 이 점을 파고들어 효율적인 뇌졸중 재활치료를 위해 게임 치료에 손가락 별 저항력을 적용시킨 장갑 형태 3D 게임 컨트롤러 gloVR을 제작했다. **gloVR은 Flex 센서, IMU 센서, 카메라 모듈을 이용하여 손가락의 움직임을 파악**하고 Unity Engine과 블루투스 통신하여 값을 전달한다. 뇌졸중 환자의 재활을 돕기 위해 **탄성 줄을 이용하여 손에 저항력을 발생시키도록 설계하였다**. 사용자는 5개의 서보 모터를 통해 손가락마다 적용되는 저항력을 변경시킬 수 있다. 이는 사용자가 직접 조절하거나 게임 속 상황에 따라 달라질 수 있다.

gloVR 컨트롤러에 맞는 게임(캐치볼 게임)을 자체 제작하여 재활효율을 극대화할 수 있도록 하였다. 컨트롤러의 움직임(손의 위치값(x,y,z), 손가락 별 구부린 정도, 손의 회전각)을 파악하여 화면상에 나타낼 수 있게 하였고, 날라오는 야구공을 잡는 게임을 개발하였다.

gloVR의 게임 콘텐츠는 Unity 3D 엔진으로 설계되어 VR 게임으로 발전할 수 있는 장점이 있다. **VR 게임을 통한 상지 재활은 뇌졸중 환자의 재활 효율을 더욱더 높여줄 수 있다.**

○ 개발 목표

우리는 3D 게임과 로봇 치료법을 결합하여 치료사의 치료를 대체하고, 효과적으로 상지기능 회복을 도와주는 게임과 장갑 형태의 컨트롤러를 제작하기로 하였다. gloVR 사용을 통해 **의료인력 결핍을 극복하고, 뇌졸중 환자의 재활효과를 극대화하는 것이 목표이다.**



완제품 예상도

○ 개발 작품의 필요성

뇌졸중은 실생활에서 자주 사용되는 상지기능 마비를 동반한다. 일상생활로의 복귀를 위해서는 상지기능 회복을 위한 작업치료가 필수적이다. 현재 상지 재활치료의 수요가 증가하는 추세이기 때문에 그에 따른 의료인력이 부족하다. 따라서 치료사의 치료를 대체할 수 있는 시스템이나

기계의 필요성이 대두되고 있다.

현재 로봇 치료법, 건축 제한 운동 치료법, 가상 현실, 운동 연상 등의 다양한 상지 재활기법 등이 사용되며 연구되고 있다. 하지만 이러한 치료법들 모두 단독으로는 치료 환경이 단조롭고, 흥미나 과제 다양성, 치료 참여도가 떨어지는 단점이 있다. 3D 게임을 통한 치료법은 시각적인 자극을 통해 광범위한 운동 관련 대뇌조직을 활성화해 재활치료의 효과를 높일 수 있다.

COVID-19의 확산에 따라 비대면 치료 서비스가 필요한 시점이다. 물리치료 또한 비대면 서비스 전환 추세에 맞추어 변화할 것이다.

gloVR 타겟시장은 다양한 분야가 복합적으로 연관된 시장으로서 각각의 시장 전망을 조사했다. 확장 중인 Healthcare VR, 신경 재활치료 디바이스 시장에서 가장 활성화된 시장은 각각 환자 치료 관리(재활치료, 컨디션 케어), 뇌졸중 재활 치료이다. 즉, 뇌졸중 신경 재활치료용 Healthcare VR 기기인 gloVR의 타겟 시장의 전망은 밝다고 볼 수 있다.

- 1) Healthcare VR 시장 전망 : 전 세계 Healthcare VR 시장은 2018년 10.1억 달러에서 연평균 성장률 28.15%로 증가하여, 2023년에는 34억 8,456만 달러에 이를 것으로 전망되었다. - Fig. 1
- 2) 글로벌 Healthcare VR 시장의 용도별 시장 전망 : 환자 치료 관리는 Healthcare VR 시장 중 가장 활성화된 분야. 연평균 성장률 14.50%로 증가, 2023년에는 4억 2,270만 달러- Fig. 2
- 3) 뇌졸중 신경 재활치료 디바이스에 대한 글로벌 시장 전망 : 자료를 보면, 신경 재활치료 디바이스 시장은 연평균 성장률 12.5%를 기록할 것이며, 그중에서 뇌졸중 관련 디바이스 기술은 47%의 높은 점유율을 차지한다.- Fig. 3

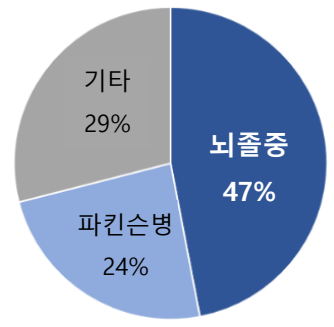
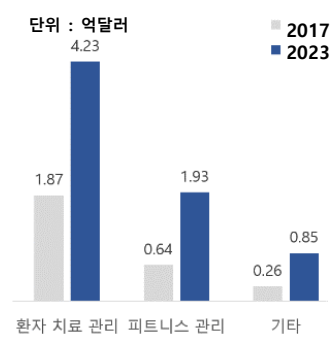
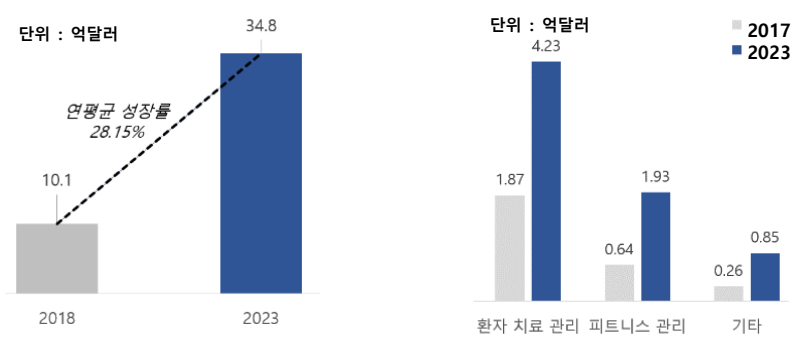


Fig 1. 출처 : Markets and Markets(2017)

Fig 2. 출처 : TechNavio(2018)

Fig 3. 출처 : Transparency Market Research (2020)

□ 개발 환경 설명

○ Hardware 구성

1) 개발 목표 및 고려 사항

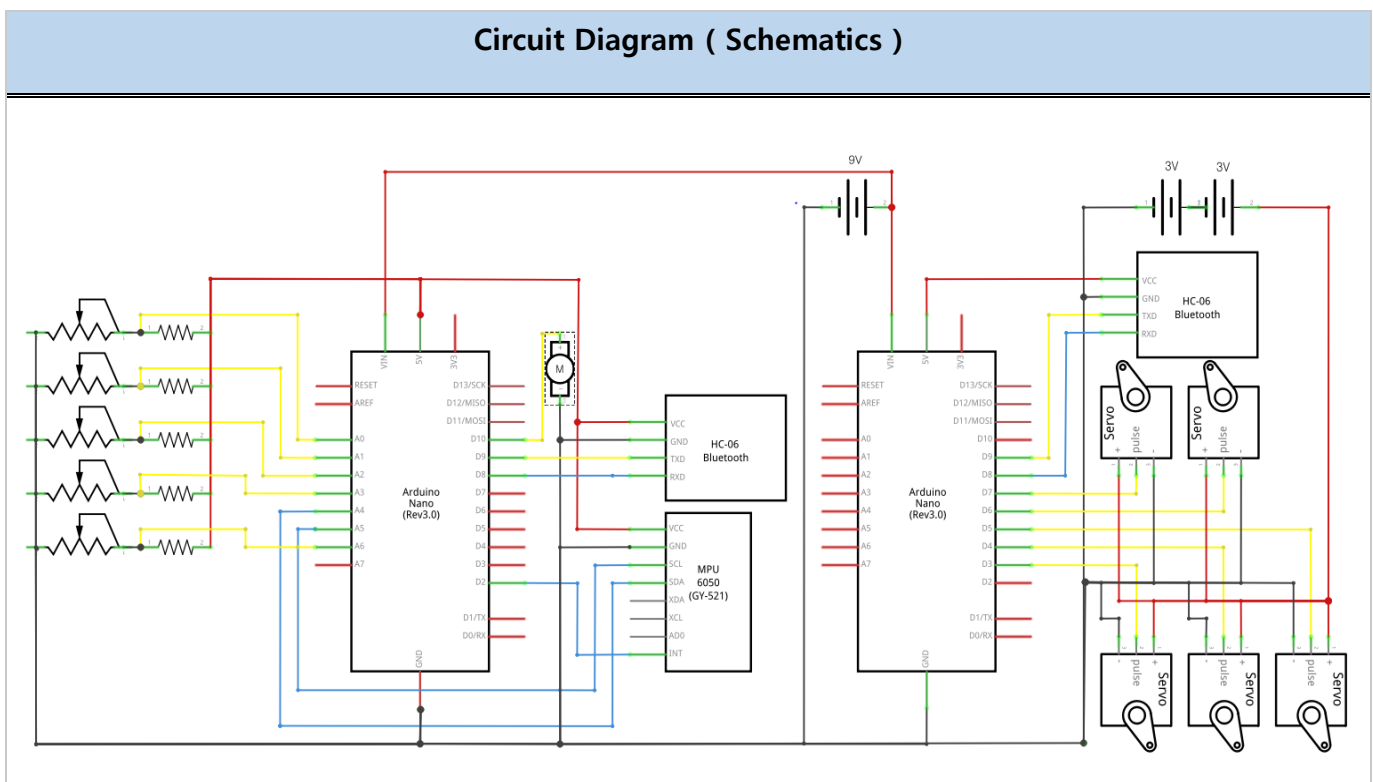
- (1) 개발 목표 : “내 손같이 편안한 착용감으로 재활치료를 보조하는 핸드 컨트롤러 gloVR”
- (2) 경량화, 착용감 : 3D Printer 를 이용하여 설계 - 출력 - 수정 작업을 거친다.
- (3) 하드웨어 간소화 : 소프트웨어적으로 대체 가능한 기능은 적극 활용한다. 무게를 줄여 사용감을 높이고, 단가를 낮춘다. 필수적인 기능을 수행하되, 불필요한 기능은 지양한다.

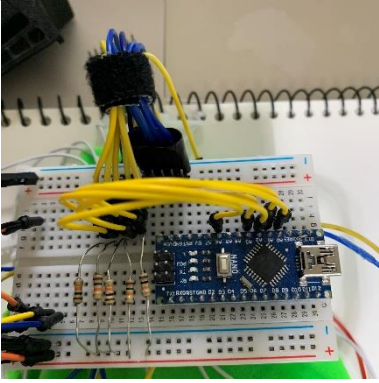
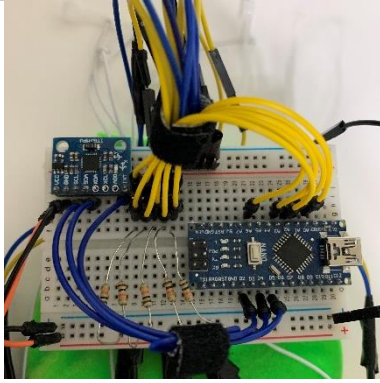
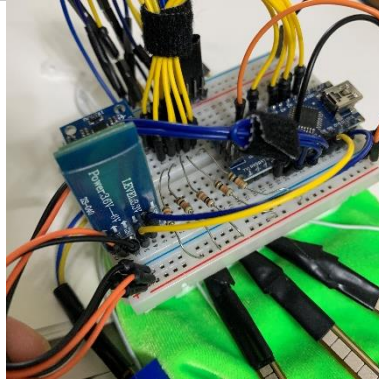
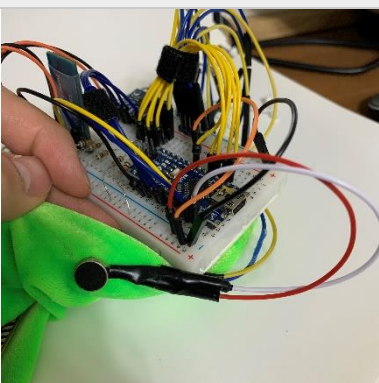
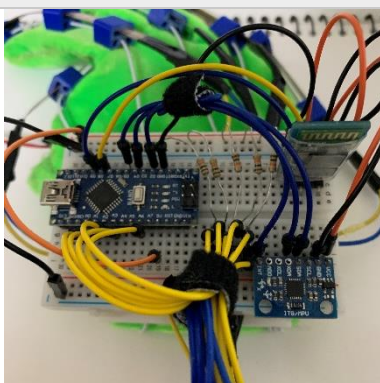

2) 부품 선정

파트	제품명 및 사진	장단점
MCU	<p>[Arduino Nano]</p> 	<p>장점: 높은 휴대성, 간단한 회로연결, 낮은 전력 소모</p> <p>선정이유: Arduino 제품이 높은 접근성을 가지고 있으며, 특히 Nano는 휴대성이 높다.</p>
Sensor	<p>[MPU-6050]</p> 	<p>장점: 저렴한 가격, RPY를 이용한 회전표현</p> <p>선정이유: Gyro 센서 중 가장 간단한 6축 센서로써, DMP 기능을 이용해 RPY의 정확한 값을 구할 수 있다.</p>
	<p>[SEN-08606]</p> 	<p>장점: 높은 휴대성, 간단한 회로연결</p> <p>선정이유: 손가락과 비슷한 길이의 센서로 따로 신호선 없이 힘 정도를 측정할 수 있다.</p>
	<p>[APC900-FHD]</p> 	<p>장점: FHD의 높은 화질, 높은 휴대성, 다양한 OS와의 호환성, 드라이버 없이 작동, 높은 데이터 접근성</p> <p>선정이유: Window와 Mac에서 드라이버 없이 작동</p>
Actuator	<p>[SG-90]</p> 	<p>장점: 저렴한 가격, 높은 휴대성, 간단한 각도제어,</p> <p>선정이유: Arduino에 호환되는 RC servo 모터 중 가장 작은 모터이며, 적절한 출력 토크 스펙을 갖는다.</p>

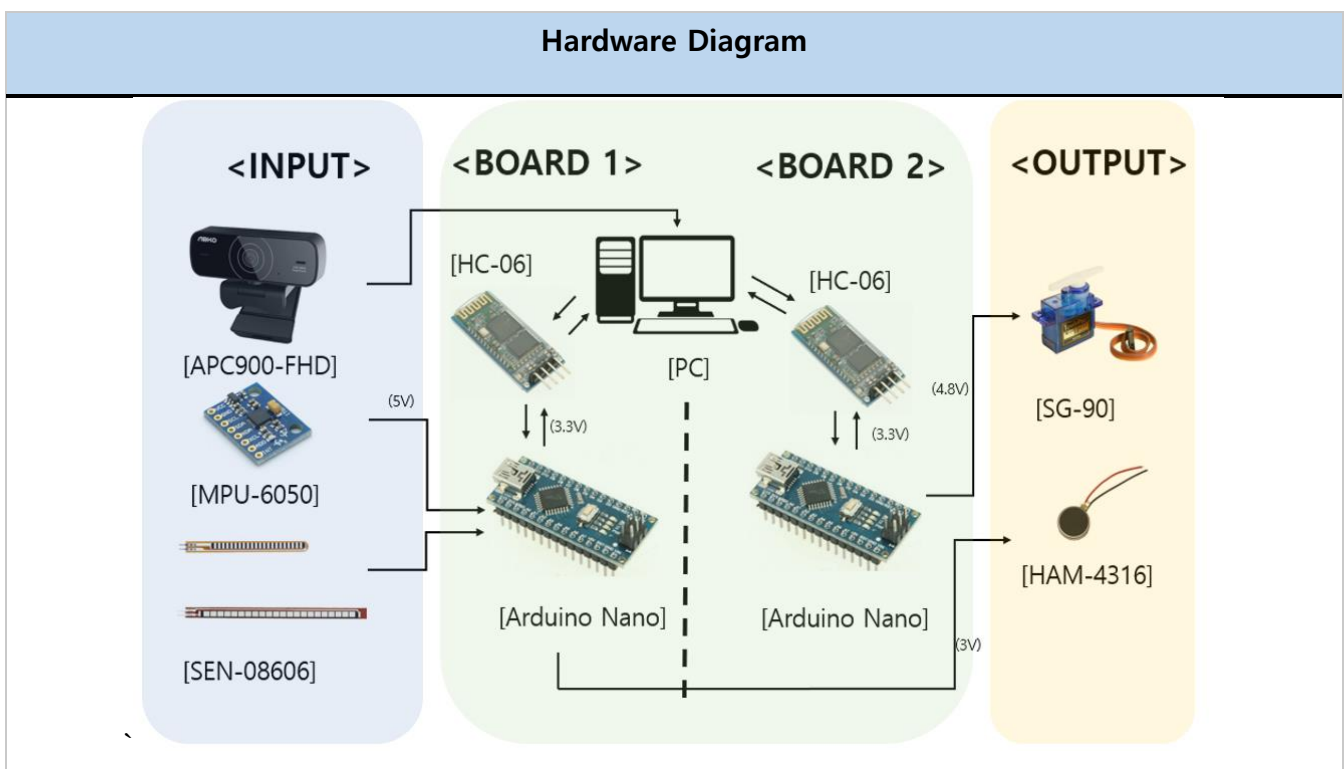
	<p>[HAM-4316]</p> 	<p>장점: 저렴한 가격, 높은 출력, 낮은 전력소모</p> <p>선정이유: 손바닥에 붙일 수 있을 정도로 크기가 작고, 손에서 느낄 수 있을만큼 충분한 진동세기를 가진다.</p>
통신	<p>[HC-06]</p> 	<p>장점: TX/RX 통신, 2.1mbps의 전송속도, 양방향 통신</p> <p>선정이유: 다양한 블루투스 모듈 중 양방향 통신이 되며, PC와의 연결이 가장 안정적으로 되는 제품이다.</p>
Power	<p>[9V battery] [1.5V battery]</p> 	<p>장점: 높은 접근성, 저렴한 가격, 높은 호환성, 안정성</p> <p>9V Nano 전원 공급</p> <p>1.5V batter pack : Servo motor 전원 공급</p>

3) 회로 배선

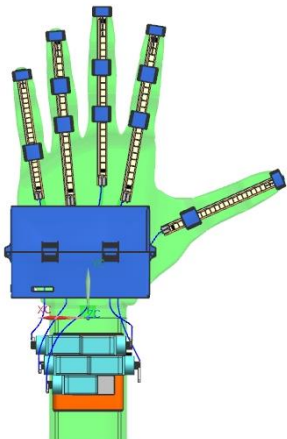
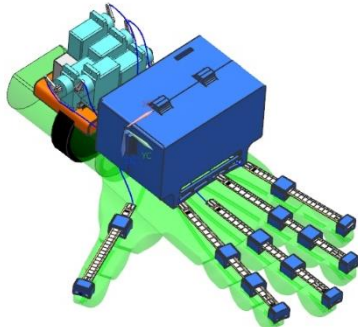
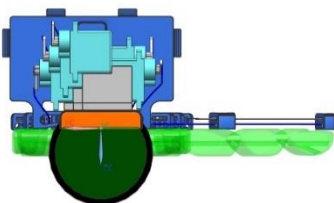
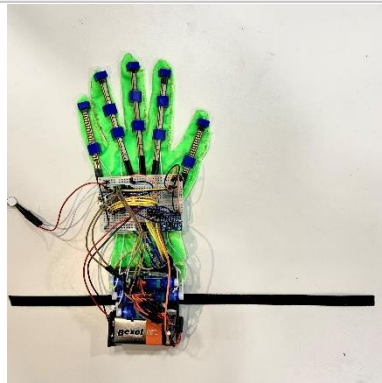
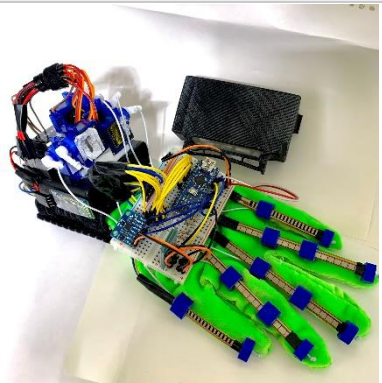
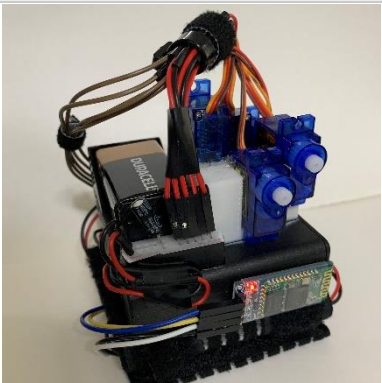
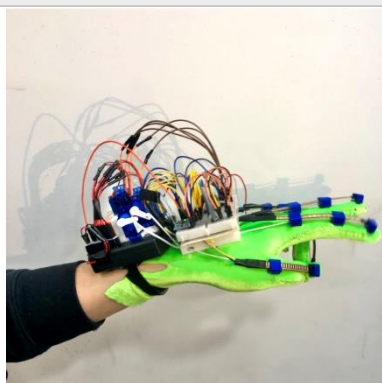
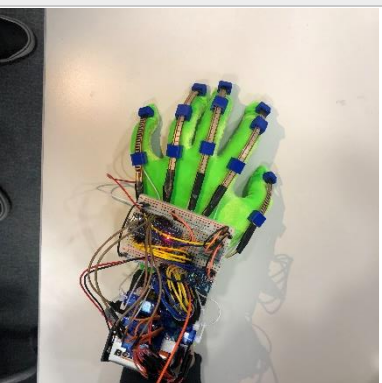
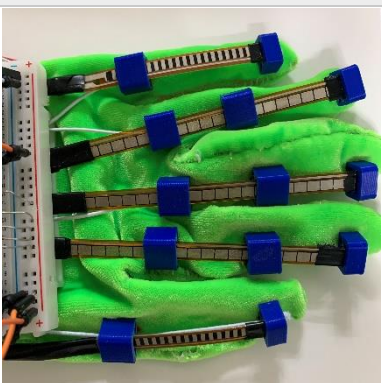


Flex Sensor 배선	MPU6050 배선	HX 06 배선
		
진동 모터 배선	메인 회로 완성 사진	Sg - 90 배선
 (손바닥 면에 부착)		 전압 안정화 : 캐패시터 (100uF)

4) 하드웨어 시스템 구성



5) 하드웨어 구조 이미지

Hardware Structure			
Graphics	Top View	Isometric View	구동부
			Back View
			
실물	Top View	Isometric View	구동부(서보, 배터리케이스)
			
	착용 - 1	착용 - 2	Finger Part
			

○ Hardware 기능 (제어 방법 등 서술)

1) MCU 1, 2 : 하드웨어의 메인 시스템은 PC, Arduino Nano 2 대, HC-06 2 대로 이루어져 있다.

- **Arduino Nano 1(main)** – flex sensor, gyro sensor 입력, Unity 와 통신(hc - 06), 진동모터 구동

- **Arduino Nano 2(ServoControl)** – Unity 와 통신(hc – 06), 서보모터 구동

2) INPUT

(1) Webcam : PC 와 직접 연결되어 OpenCV 를 통해 손의 위치를 찾는데 사용된다.

(2) Gyro sensor : I2C 통신과 DMP(Digital Motioning Processing)기능을 사용하여 상보필터와 칼만필터의 사용없이 안정된 Roll, Pitch, Yaw 의 값을 측정한다.

(3) Flex sensor : analog pin 에 연결하여 구부림 정도에 따른 저항값을 통해 힘 정도를 측정한다.

3) OUTPUT – PWM 제어 구동

(1) Servo motor : digital pin 에 연결하여 PWM(Pulse width Modulation) 제어를 이용한다. 손가락 끝에 연결된 고무줄이 모터까지 연결되어 모터의 회전각에 따라 손가락에 가해지는 장력이 달라진다. 안정적인 작동을 위해 개별적으로 1.5V 배터리 4 개를 연결해주었다.

(2) 진동모터 : 캐치볼 게임에서 공을 잡을 때마다 진동이 울려 공을 잡았다는 것을 알려준다.

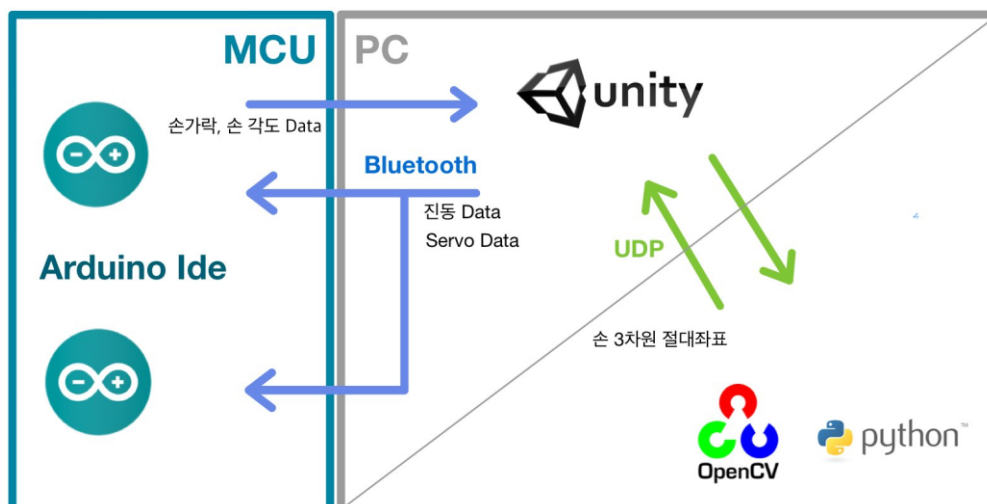
4) 장갑 디자인 – 스판덱스

(1) 사용자의 손의 크기는 모두 다르고, 편안한 착용감을 위해 Soft Structure 재질을 선택하였다.

(2) OpenCV 색 검출을 위해 주로 사용되는 Green 컬러 원단을 사용하였다.



○ Software 구성



1) 아두이노 제어 프로그램

- (1) 다른 역할을 하는 두 보드에서 사용되는 기능을 통합 관리 및 개발하기 위해 DataHandler 라이브러리를 제작했고, 보드에 연결된 센서와 액추에이터의 제어를 위한 프로그램을 개발하였다.
- (3) PC 와 블루투스 모듈간 통신을 위한 코드를 작성하였다.

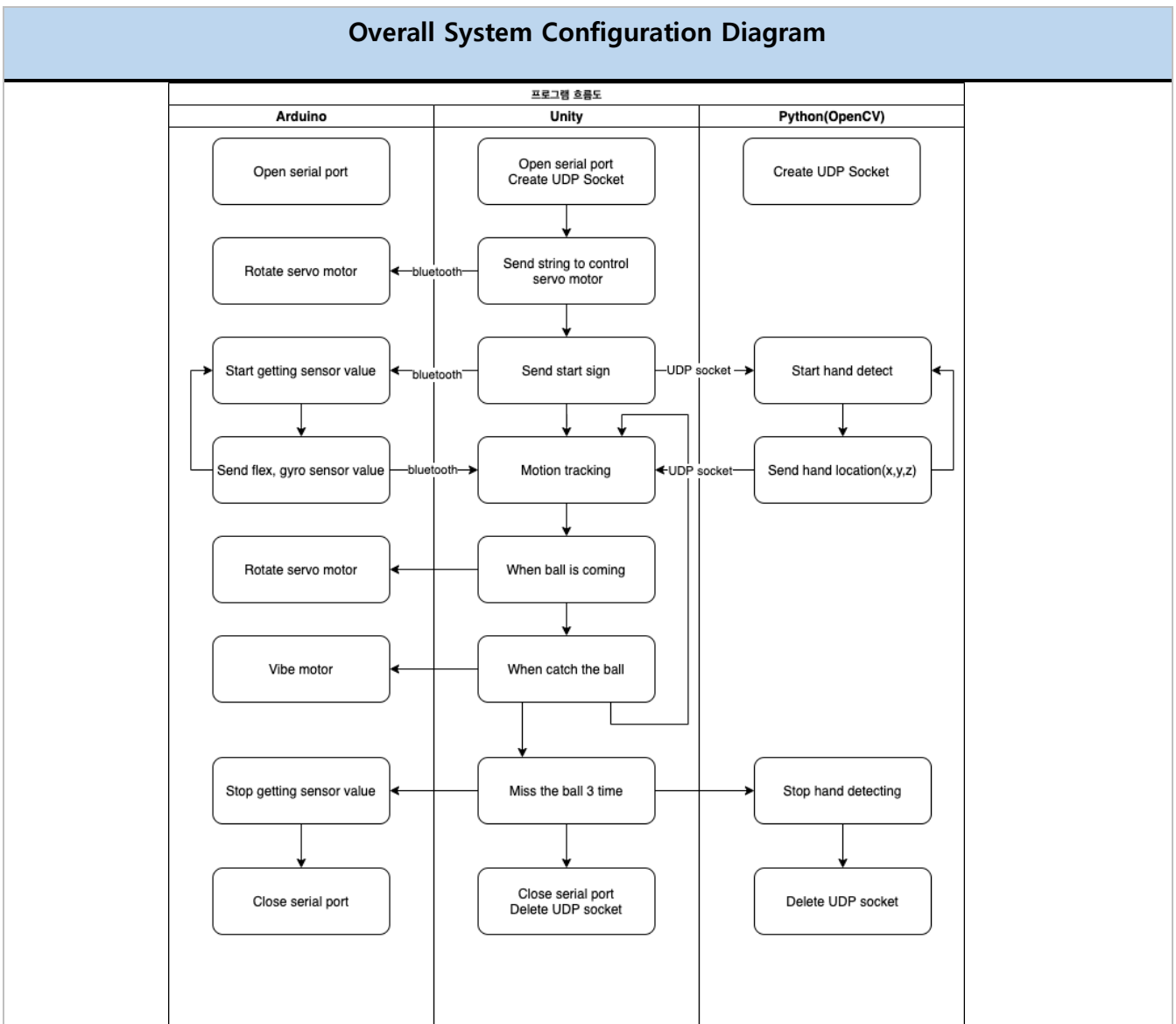
2) Python 프로그램

- (1) OpenCV 와 카메라를 제어로, 3 차원의 HandDetection 을 수행하는 프로그램을 개발하였다.
- (2) Socket 모듈을 사용하여 Unity 와 상호작용하는 UDP 통신을 구축하였다.

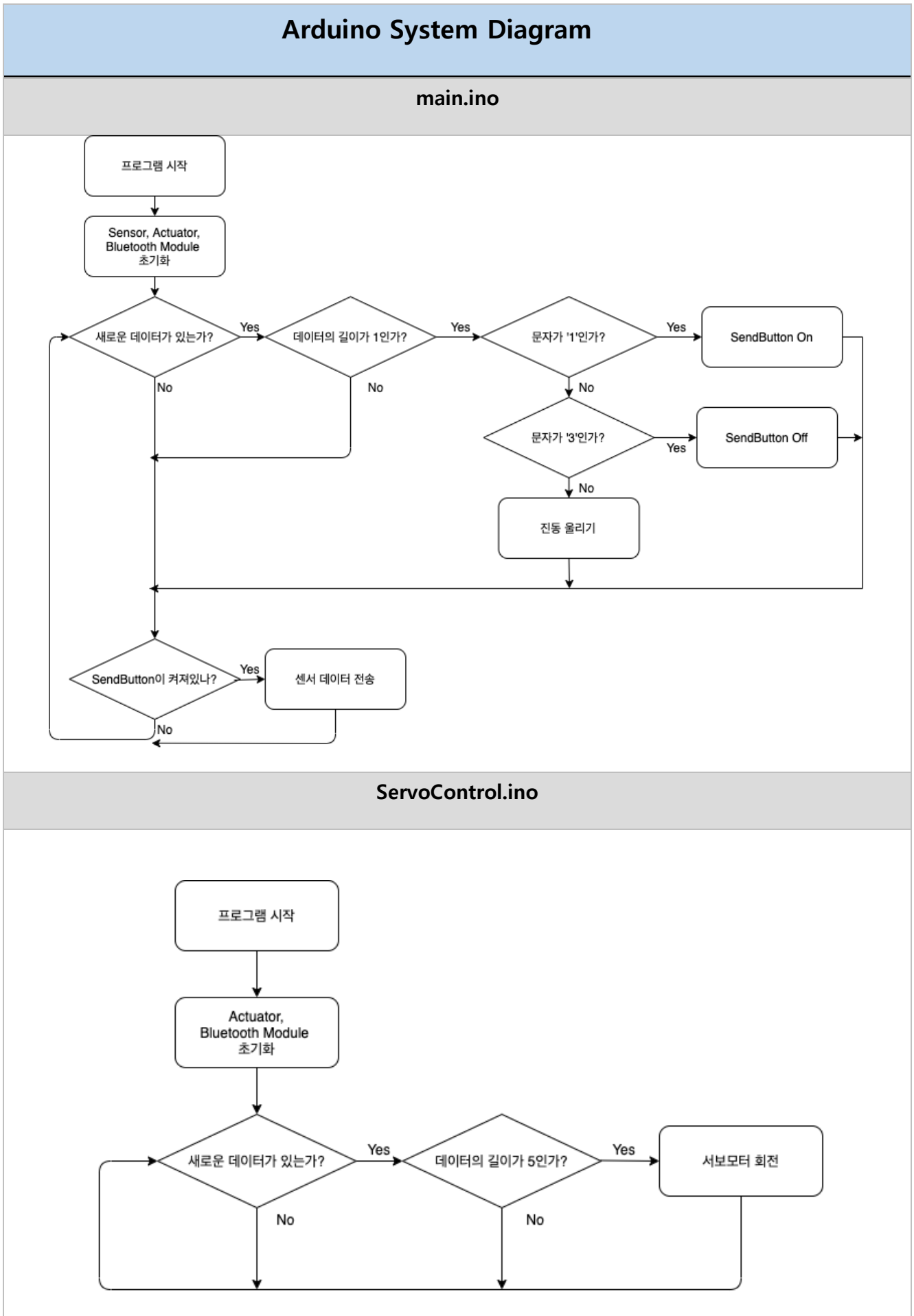
3) Unity Game : Unity contents 제작, Arduino 와 Bluetooth 통신, OpenCV 와 UDP 통신

○ Software 설계도 (흐름도 및 클래스 다이어그램 등 / 개발언어에 따라 선택)

1) 전체 시스템 구성도

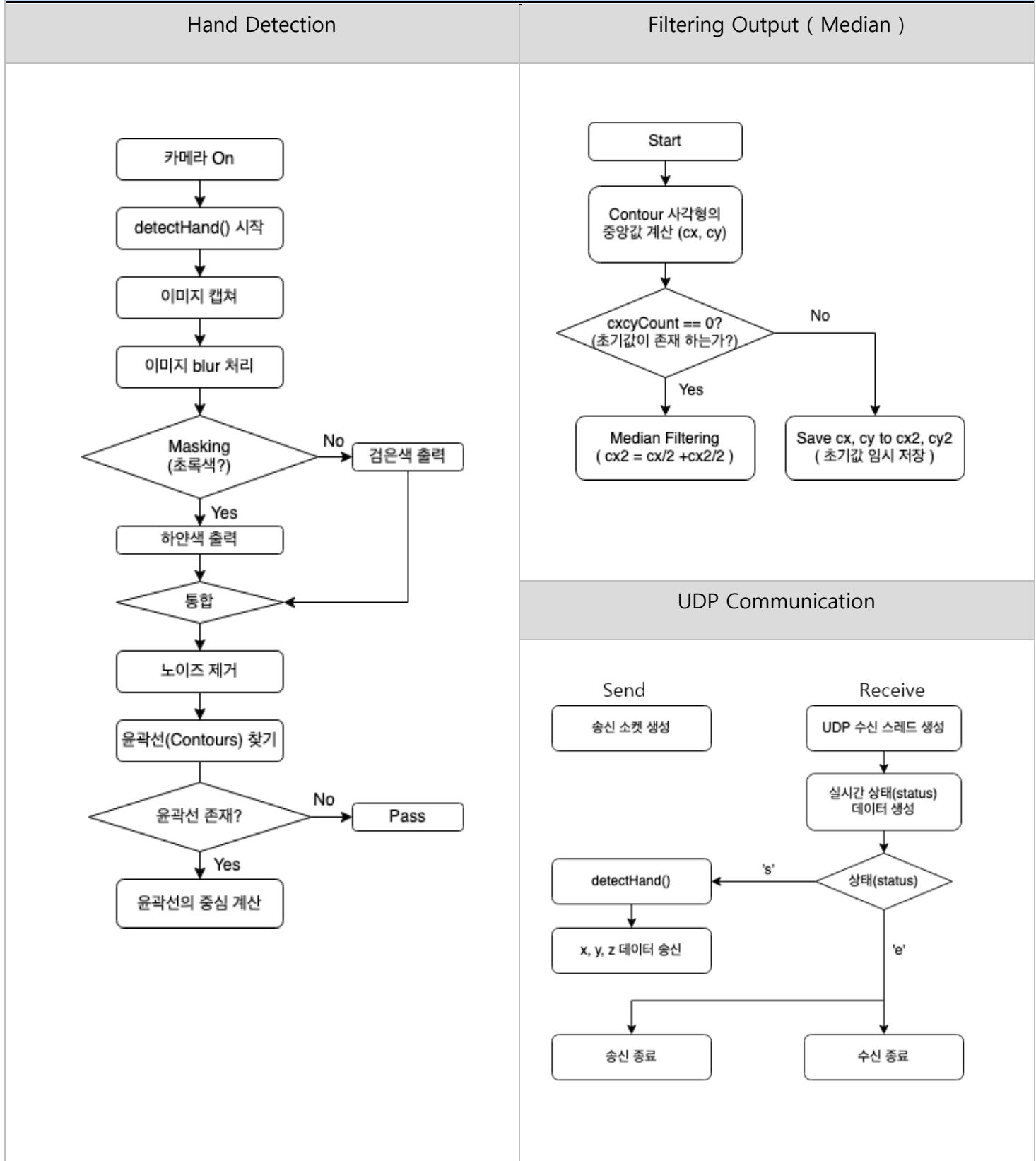


2) 아두이노 시스템 구성도

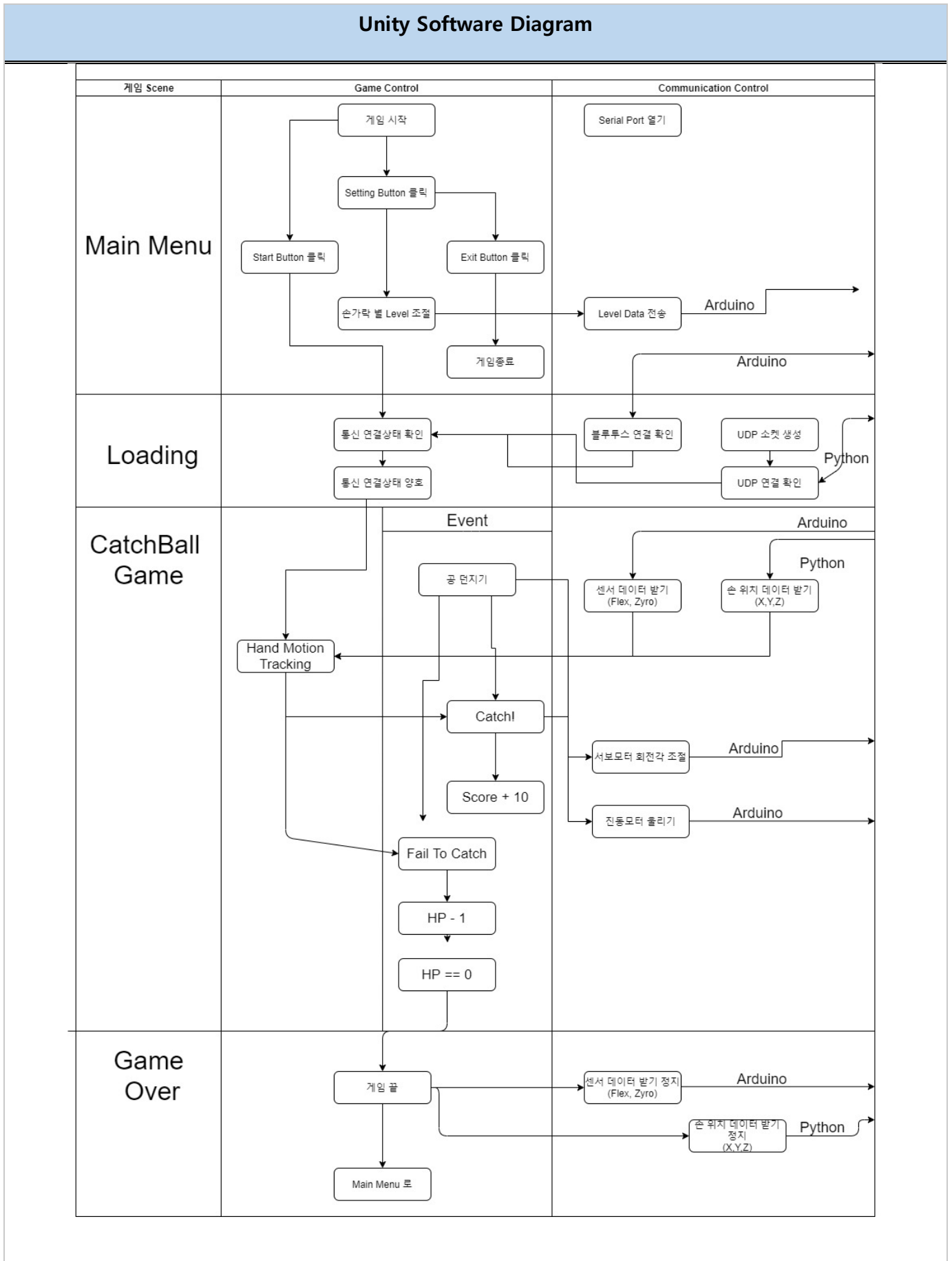


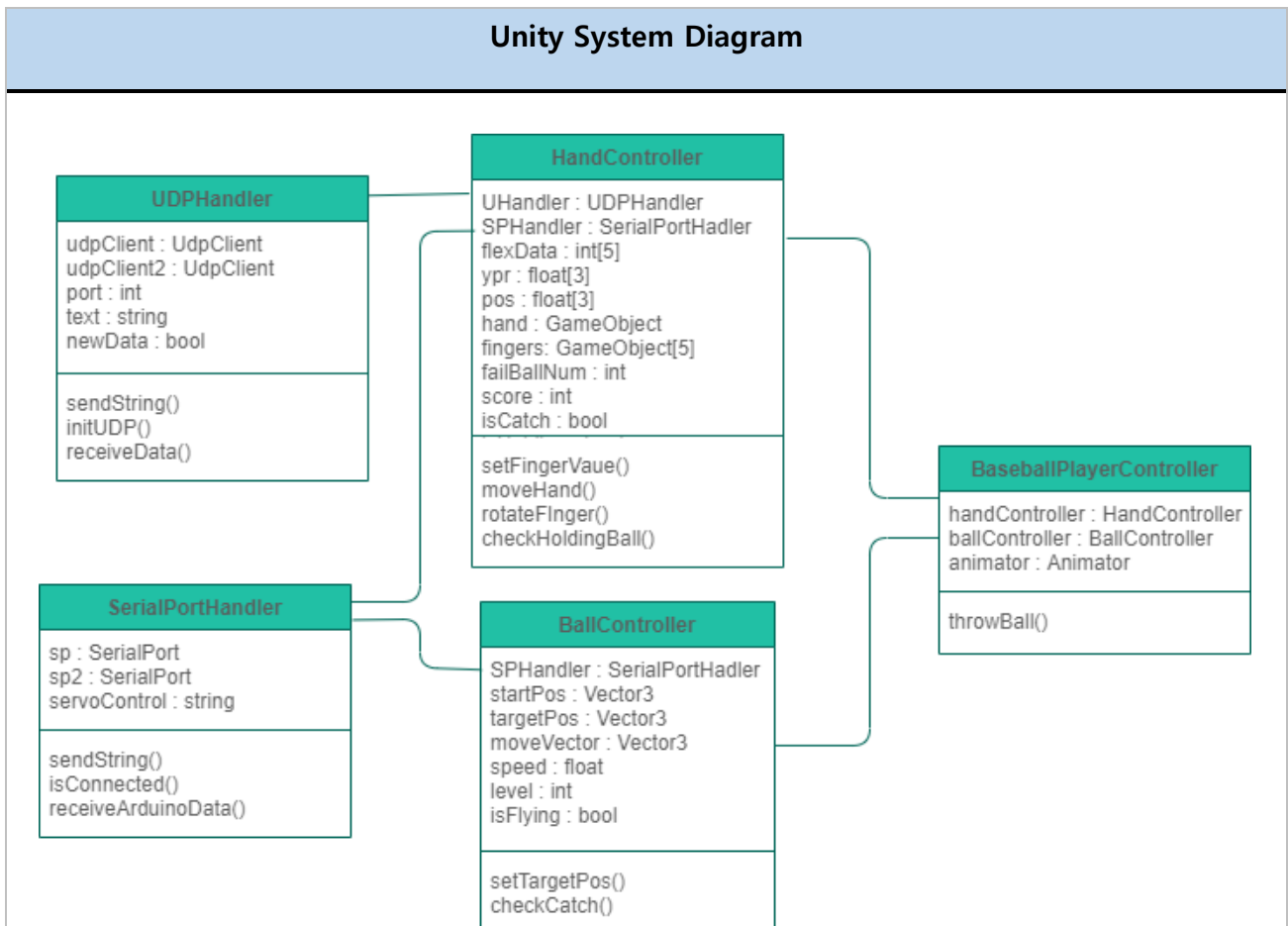
3) OpenCV 시스템 구성도

OpenCV System Diagram



4) Unity 시스템 구성도





○ Software 기능

1) Hand Detection 기능

(1) Arduino

- **핸드 모션 tracking** : 플렉스 센서를 활용하여 손가락 움직임 데이터를 얻고, 손등에 부착된 자이로 센서를 통해 손의 Roll Pitch Yaw 데이터를 얻는다.

- **Noise Filtering** : 아날로그 센서의 필터링은 필수적이지만 실시간으로 핸드 모션을 그래픽에 반영해야 하는 만큼 반응시간이 길어지는 것을 지양해야 한다.

Cut off 값을 조절하며 필터링 정도(지연시간과 비례) 반응시간 사이에서의 적정점을 찾았다.

	LowPassFilter + Moving Average Filter	LowPassFilter
Serial Plotter		
Result	계단 현상, Long Latency	곡선, Short Latency

Low Pass Filter(LPF) 만으로도 충분히 노이즈를 제거하고 우수한 반응 속도를 보여 완성품에는 LPF 가 적용되었다. 다음은 DataHandler.h 에서 구현된 필터링 메소드이다.

<pre>// Low Pass Filter filteredValue[i] = filteredValue[i] * (1 - alpha) + flexValueArr[i] *alpha; // 손가락 범위 제한 if (filteredValue[i] <= flexMin[i]) filteredValue[i] = flexMin[i]; else if (filteredValue[i] >= flexMax[i]) filteredValue[i] = flexMax[i]; // 50 ~ 180 deg 범위로 변환 angleValue[i] = map((int)filteredValue[i], flexMin[i], flexMax[i], 50, 180); angleValue[i] *= -1; angleValue[i] += 230;</pre>	<p>alpha : 실험적으로 얻은 FilterDegree 값 filteredValue[5] : 필터링 된 데이터 배열, setup 문에서 초기값을 입력 받는다. flexValueArr[5] : 5 개의 flex sensor 로부터 얻은 데이터 배열. angleValue[5] : 필터링 후 map 함수를 통하여 50 ~ 180 deg 범위로 변환한다. flexMax[5], flexMin[5] : 미리 측정한 플렉스 센서의 최대값 및 최소값 배열.</p>
--	---

- Bluetooth Serial Communication

블루투스 모듈을 통해서 유니티와 아두이노 간 통신을 구현했다. 이때, 두 응용 프로그램 간 원활한 통신을 위하여 각 프로토콜을 정의하였다. 아두이노에서 유니티와 주고받는 신호는 다음과 같다.

1. Arduino To Unity

1Byte (Byte)	5Byte (uint8_t)	21Byte (String)					1Byte (Byte)	
Start Byte (Value : 200)	Flex Sensor Data * 5 (Range : 50 ~ 180)	GyroSensor - Yaw (Range : -3.2 ~ 3.2)	'\n'	GyroSensor - Pitch (Range : -3.2 ~ 3.2)	'\n'	GyroSensor - Roll (Range : -3.2 ~ 3.2)	'\n'	End Byte (Value : 201)

2. Unity To Arduino

(1) Start, Vibe, End Signal

3Byte (String)		
Start Byte (Value : 's')	CASE : '1' -> Start '2' -> Vibe '3' -> End	End Byte (Value : 'e')

(2) Servo Controll Signal

1Byte (String)	5Byte (String)	1Byte (String)
Start Byte (Value : 's')	손가락 별 서보모터 회전 각도 새끼 - 약지 - 중지 - 검지 - 엄지 (Range : '0' ~ '3')	End Byte (Value : 'e')

1. 아두이노 -> 유니티 데이터

5 개의 Flex Sensor 값과 Gyro Sensor 값을 보낸다.

Float 형식의 Flex, Gyro Sensor 값을 소수 둘째자리까지

문자열로 변환한 뒤 전송한다.

2. 유니티 -> 아두이노

아두이노는 유니티로부터 3 종류의 신호와 1 종류의 데이터를 받는다.

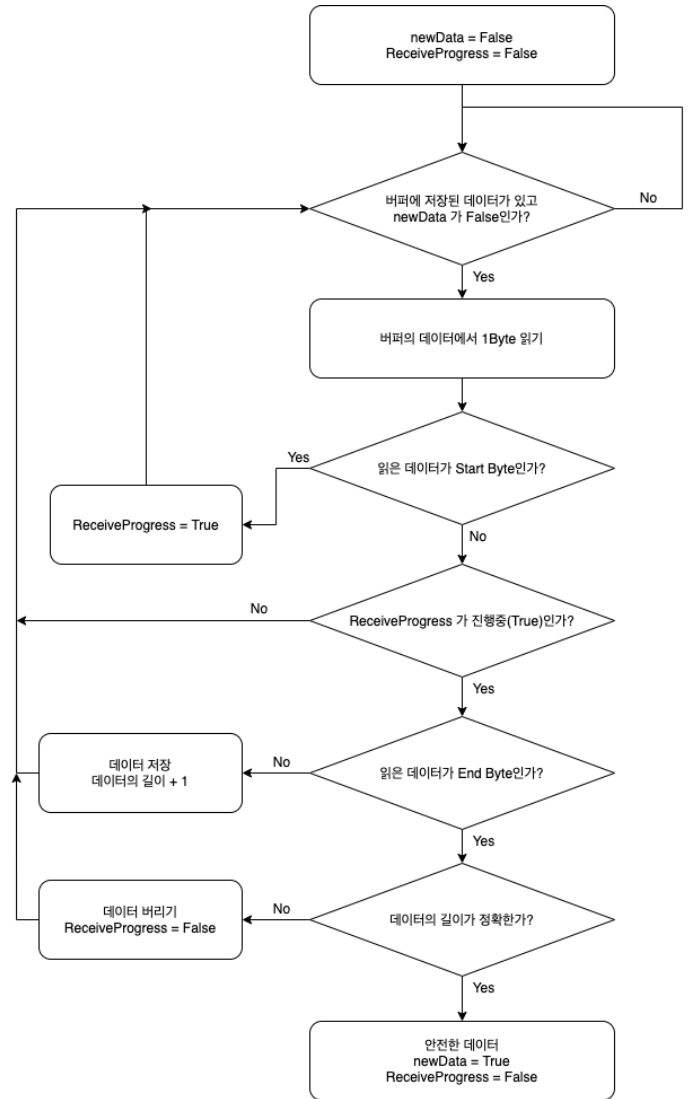
① 시작, 진동, 끝 신호

- '1': 데이터 송신 시작
- '2': 진동모터 진동
- '3': 데이터 송신 중지

② 5 개의 서보모터를 제어하기 위한 문자열 : 5 개의 서보모터의 Level 데이터를 수신한다. 수신 데이터는 '0' ~ '3'이고, 새끼 - 약지 - 중지 - 검지 - 엄지 순으로 연결된 서보모터의 회전 Level 에 대한 데이터를 담고있다.

③ 확실한 수신을 위한 알고리즘

안전한 데이터를 수신받기 위해 우측과 같은 알고리즘을 작성했다.



(2) OpenCV

- **핸드 좌표 tracking** : 컨트롤러의 절대좌표값을 감지하기 위해 OpenCV 의 함수를 이용하였다. 먼저 blur() 함수를 통해 이미지의 노이즈를 최소화하였다. 이후 초록색을 Masking 하여 초록색인 부분은 하얀색으로, 나머지 색깔은 검은색으로 나타내었다. 여러 HSV 색(빨간색, 파란색, 초록색)을 실험해본 결과 초록색을 가장 잘 Masking 하였다.

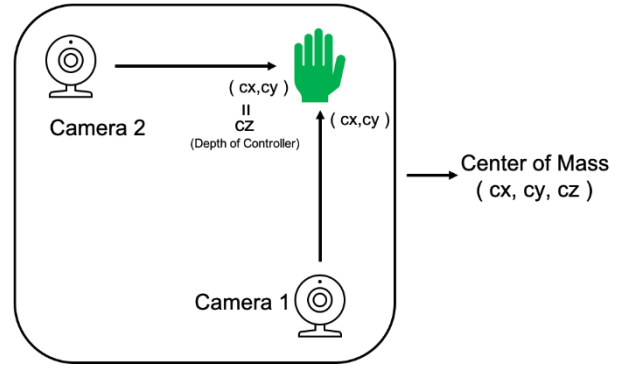


Before Masking



After Masking

흰색과 검은색 픽셀을 구분하는 Contour Area 를 생성시켜 Contour 의 최대 범위를 사각형으로 변환하고, 중점을 Center 좌표(cx, cy)로 출력하였다. 화면의 깊이는 두 번째 카메라를 90 도로 설치하여 두 번째 카메라의 cx 값을 이용하여 측정하였다.



- **Filtering** : OpenCV 의 경우 Integer 형으로 출력되고, Unity3D 엔진의 경우 Float 형으로 입력할 수 있기 때문에 화면상에서 끊기는 현상이 발생했다. 이를 보완하기 위해 이전 출력값과 현재 출력값의 중간값을 이용하는 Median Filter 식을 이용하였다.

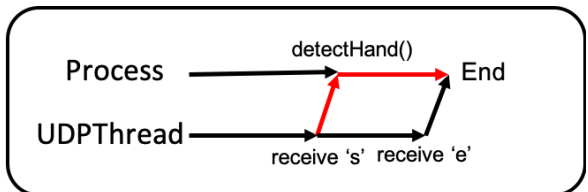
실행초기에는 이전 출력값이 존재하지 않기 때문에 같은 값을 임시로 넣어 필터링하였다.

$$cx2 = (cx + cx2) / 2$$

cx : 현재 출력 값, cx2 : 이전 출력 값

- UDP Socket Communication

수신 : detectHand() 함수를 실행·종료시키기 위해 실시간으로 데이터를 수신할 수 있는 과정이 필요했다. UDPThread 클래스를 이용하여 detectHand()의 프로세스가 진행되는 동안 recvfrom() 함수를 통해 실시간 데이터를 수신할 수 있게 하였다. 문자 's'를 받게 되면 detectHand() 함수를 실행하고, 문자 'e'를 받게 되면 프로세스가 종료되도록 설계하였다.



송신 : detectHand() 함수 안에서 출력된 값을

Unity 로 전송하는 Socket 을 만들어 sendto() 함수를 통해 전송하도록 하였다.

3) Unity (Interface) – 캐치볼 게임 제작



- (1) 아두이노, OpenCV 와 통신할 수 있는 블루투스 통신과 UDP 통신을 구현했다.
- (2) 전달받은 데이터를 통해서 사용자의 손 움직임을 화면상에 나타내는 Hand Motion Tracking 을 구현했다.
- (3) 아두이노와 OpenCV 프로그램의 흐름을 제어한다.

○ 프로그램 사용법 (Interface)

Game Manual	
1	<div style="display: flex; justify-content: space-around;">   </div> <p>처음 gloVR 프로그램을 실행하면 캐치볼 게임을 시작하는 Start 버튼, 서보 모터의 값을 설정하는 Setting 버튼, 게임을 종료하는 Exit 버튼으로 총 3개의 선택지가 있다.</p> <p>Setting 화면에서는 손가락 별로 서보 모터의 값을 조절할 수 있다. 서보모터의 값을 설정하는 이유는 사용자의 힘에 따라 맞춤형 서비스를 제공하기 위해서이다.</p>
2	<div style="text-align: center;">  </div> <p>모든 통신이 정상적으로 되는 것이 확인되고나면 게임을 시작한다.</p>
3	<div style="display: flex; justify-content: space-around;">   </div> <p>야구선수와 점수를 나타내는 점수판, 남은 기회가 몇 번인지를 뜻하는 야구공 이미지가 있다. 야구선수가 손을 향해 임의로 공을 던진다.</p>
4	<div style="display: flex; justify-content: space-around;">   </div> <p>공을 잡는데 성공하면 점수가 10 점 올라간다.</p> <p>공을 놓치면 야구공이 하나씩 줄어든다.</p>

5	
공을 3 번 놓치고 나면 게임이 종료된다.	

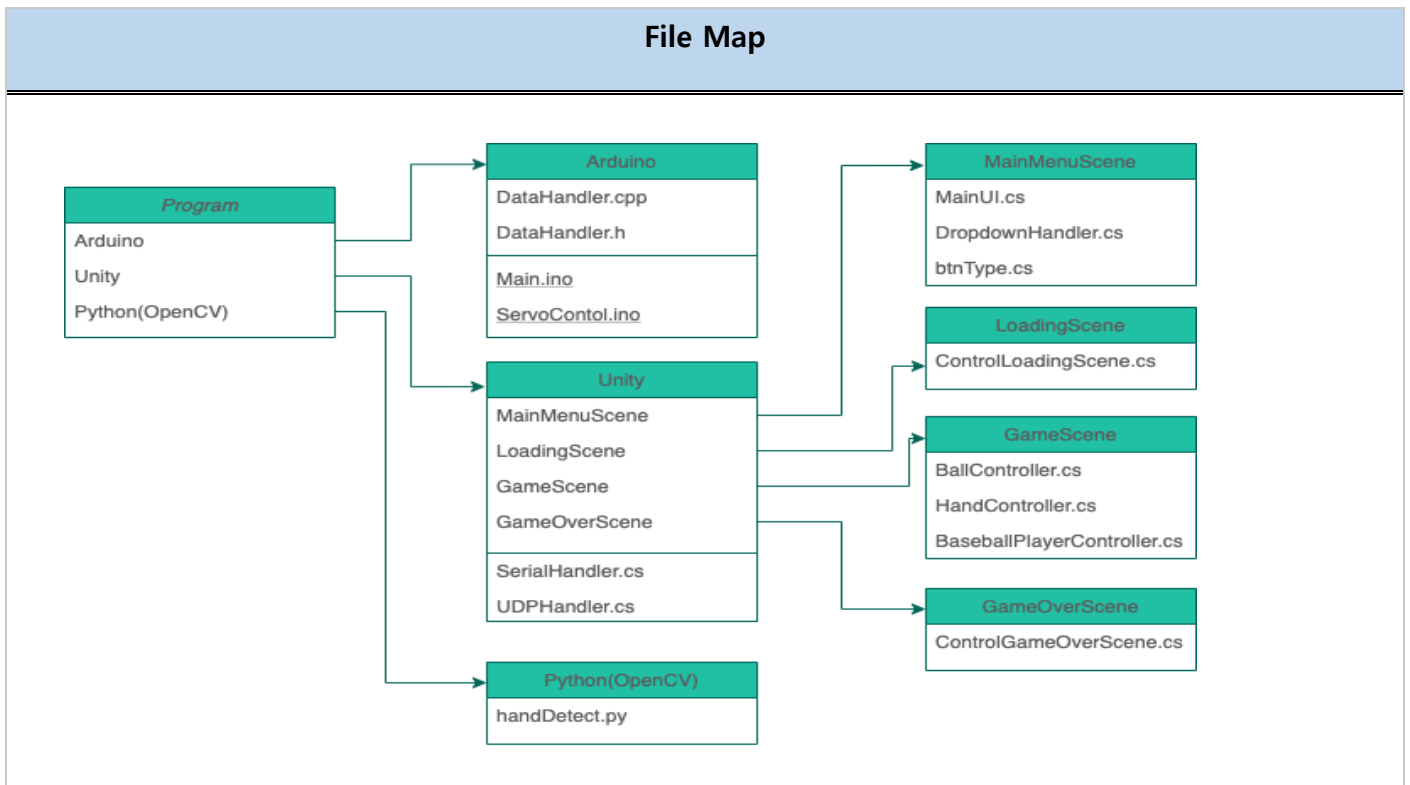
○ 개발환경 (언어, Tool, 사용시스템 등

개발언어	사용
	Unity3D Scripts Arduino Studio UDP Communication Serial Communication
	UDP Communication OpenCV

라이브러리	사용
	Detecting Glove Controller (x, y, z location)
개발도구	사용
	Unity3D (v.2019.4.8f1) Make 3D CatchBall Game UDP Communication
	Arduino Studio (v.1.8) Control Servo, Vibe, Gyro, Flex, Bluetooth Serial Communication

□ 개발 프로그램 설명

○ 파일 구성



1) Arduino

- main.ino : Flex Sensor와 Gyro Sensor, Vibration motor와 연결된 나노보드에 올라가는 파일
- ServoControll.ino : Servo motor와 연결된 나노보드에 올라가는 파일
- (library) DataHandler.cpp/DataHandler.h : 두 나노에 쓰이는 여러가지 함수 및 변수들을 통합 관리

및 개발하는 라이브러리

- (library) MPU6050.cpp/ MPU6050.h : Gyro Sensor를 제어하는 라이브러리

2) OpenCV

- handDetect.py : OpenCV라이브러리 및 socket 모듈을 사용하여 3차원 HandDetecting과 UDP통신을 수행하는 파이썬 프로그램

3) Unity

- HandController.cs – 손의 움직임, 손가락 회전, 손 회전과 같은 손과 관련된 이벤트 처리하는 파일
- BallController.cs – 날아오는 공을 제어하는 스크립트. 매번 공의 단계가 0부터 3까지 무작위로 설정되는데, 이 단계에 맞춰 서보 모터의 값을 변경한다.
- MainUI.cs / DropdownHandler.cs / btnType.cs – 메인 메뉴의 UI를 제어하는 스크립트.
- ControlLoadingScene.cs / ControlGameScene.cs – 로딩씬, 게임씬을 제어하는 스크립트. 블루투스 통신, UDP통신이 연결되어있는지 확인하거나 통신을 종료한다.
- SerialPortHandler.cs – 아두이노와 연결되는 블루투스 통신을 제어하는 스크립트. 게임 전반에서 사용되는 데이터 송수신을 위한 함수 및 변수를 다룬다.
- UDPHandler.cs – 파이썬 프로그램과 연결되는 UDP 통신을 제어하는 스크립트.○ 함수별 기능

1) Arduino – DataHandler.h

InitFlex()	Flex Sensor 핀설정, 필터링을 위한 초기값 입력
FilterDeg()	Arduino sketch 에서 Low Pass Filter 민감도를 입력 받음
InitServo()	Servo Motor 핀설정, 초기 Postioning
InitVibe()	진동 모터 핀설정
GetFlexData()	Flex Sensor 값 입력, FiltFlexData() 호출 후 손가락 각도값 반환

FiltFlexData()	GetFlexData() 내부에서 호출되어, 센서값 Low Pass Filter 로 필터링
SendData()	송신 버퍼에 Flex, gyro 데이터를 각각 입력한다.
ReceiveData()	폴링 방식으로 수신 감지, 수신 버퍼를 생성한다.
RotateServo()	수신 버퍼에 해당하는 값을 Servo Motor 에 pwm 신호로 출력한다.
MakeVibe()	진동모터를 구동한다.
TurnVibeOn()	진동모터 on/off
ClearArr()	수신 버퍼를 클리어한다.

- main.ino, ServoControl.ino

InitMPU()	MPU6050 핀설정 및 이니셜라이징, I2C 통신을 위한 객체 생성, I2C 통신 setting
getYPR()	연산을 통하여 Yaw Pitch Roll 데이터를 입력받는다.
dmpDataReady()	MPU6050 센서에 인터럽트 신호를 입력한다.

2) OpenCV - hand.py

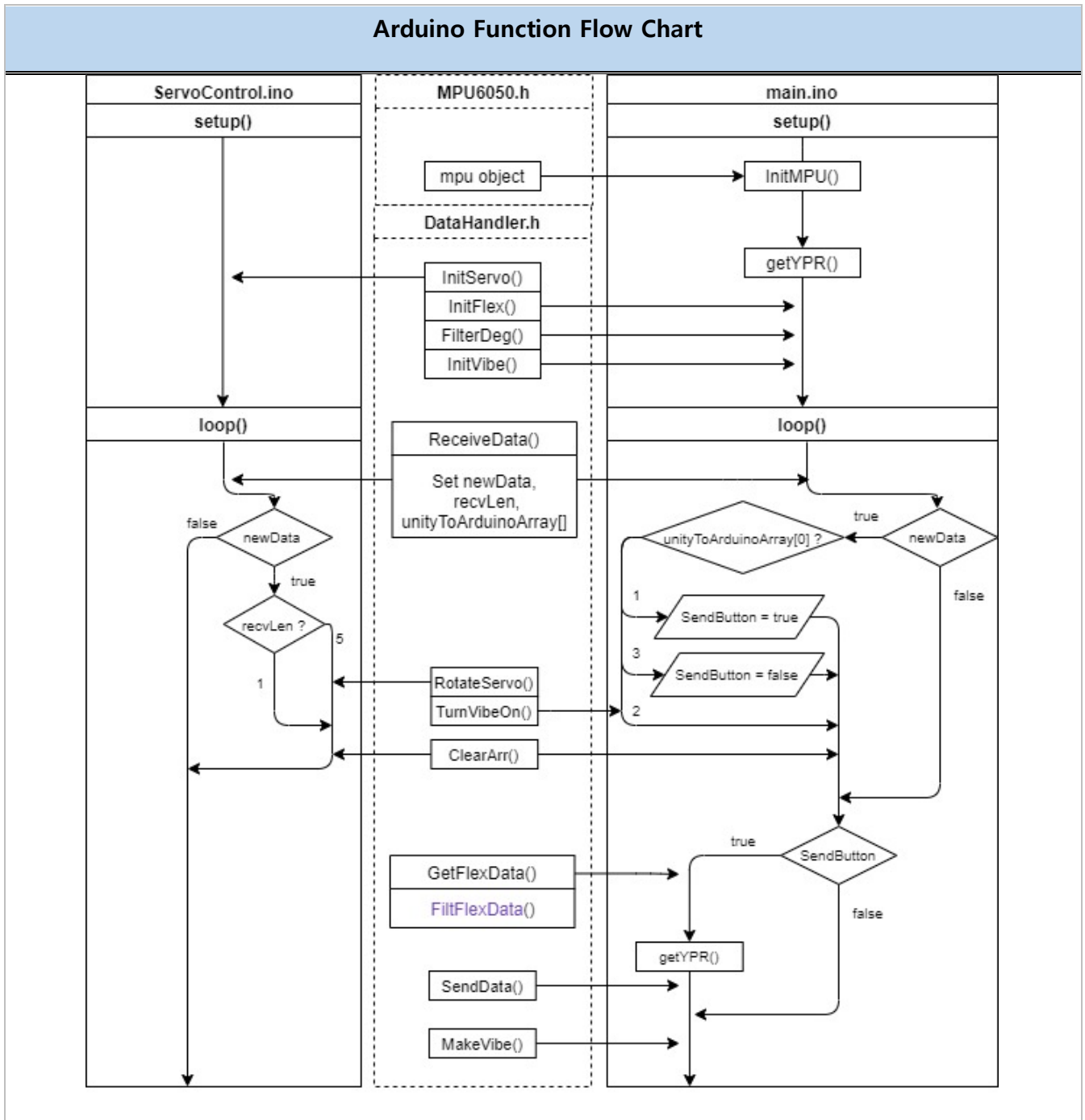
detectHand()	컨트롤러의 x,y,z 값을 Unity 에 송신 및 출력한다.
ESC()	키보드 'esc' 입력 시 프로세스가 강제 종료된다.
MF()	중간값 필터링의 출력값을 반환해준다.

3) Unity

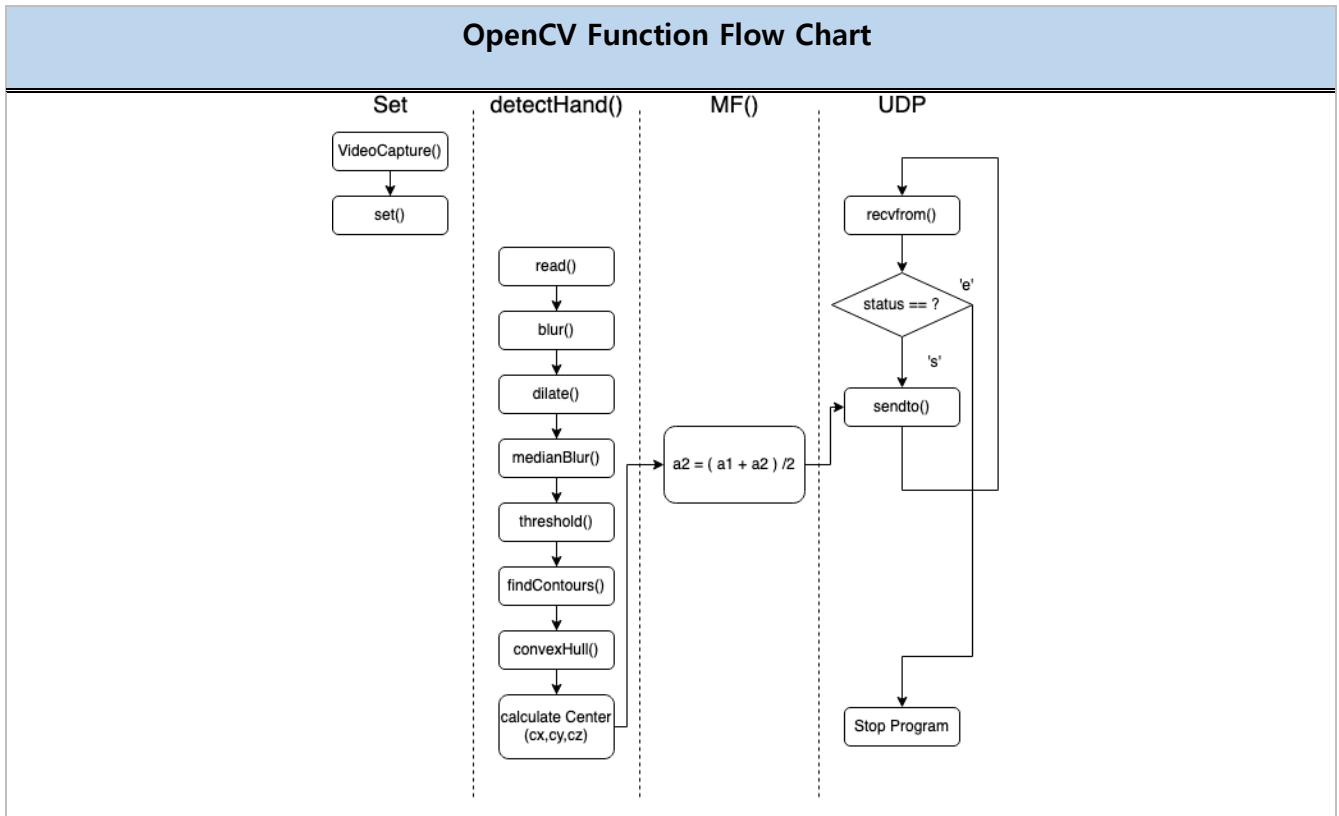
moveHand()	손의 x, y, z 값을 받아 그 위치로 이동시킨다.
rotateFinger()	flex 센서 값을 받아 손가락이 구부러진 정도를 설정한다.
checkCatch()	날아가는 공을 사용자가 잡았는지 확인한다.
SendString()	UDPHandler.cs 에서는 파이썬으로, SerialPortHandler 에서는 아두이노로 매개변수로 주어진 문자열을 전송한다.
ReceiveArduinoData()	아두이노에서 송신된 데이터를 수신한다.
IsConnected()	블루투스 통신이 원활하게 이루어지는지 확인한다.
ReceiveData()	파이썬에서 송신된 데이터를 수신한다.

○ 주요 함수의 흐름도

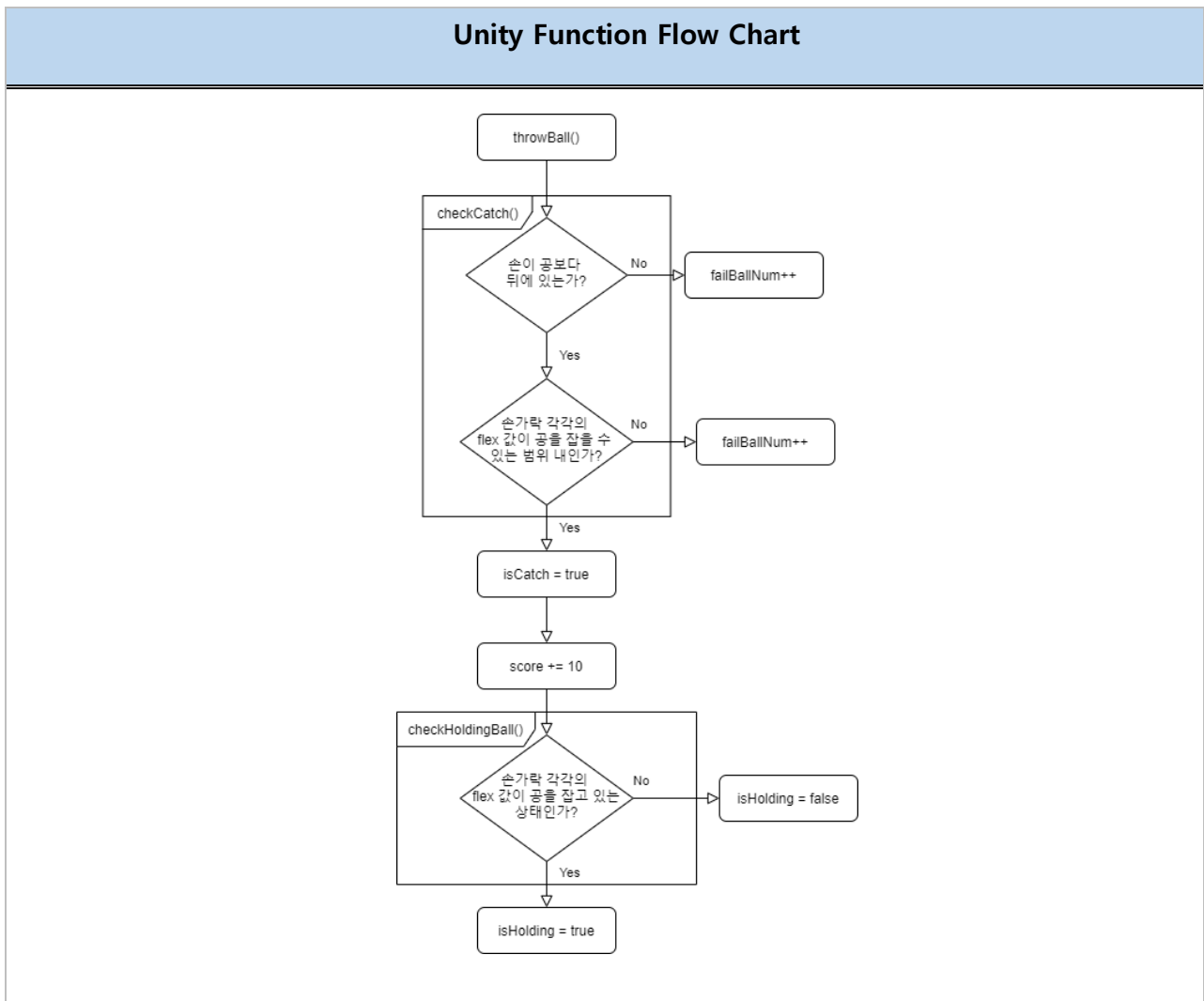
1) Arduino



2) OpenCV



3) Unity



○ 기술적 차별성

- 1) 서보모터와 고무줄을 이용하여 손가락에 저항력이 발생할 수 있도록 설계하였다. 게임 속 설정을 통해 손가락 별 서보모터의 각도를 조절하여 저항력의 크기를 각기 다르게 설정할 수 있다.
- 2) 진동모터를 이용하여 게임 속 감각을 끌어올렸다. 사용자의 현실감 높은 재화에 초점을 두고 제품을 설계했다.
- 3) 스테레오 카메라를 이용하여 컨트롤러의 위치를 출력할 수 있는 프로그램을 설계하였다. gloVR 사용자는 HMD 를 구매하지 않고도 손쉽게 플레이 환경을 세팅할 수 있다.

□ 개발 중 발생한 장애요인과 해결방안

○ MCU 성능의 한계

- Arduino 의 특성상 많은 Sensor 와 Actuator 가 들어갈 경우 서로 충돌이 일어나기 쉽다.
- 통신 규약을 설정하여 원활한 데이터 교환을 통한 통신 속도 개선해주었다.
- 외부전력의 개수를 2 개로 나눠 분할된 전력공급을 이용, 각 하드웨어 올바른 전력을 전달해주었다.
- Multi Thread 를 지원하지 않아 servo motor 와 mpu6050 이 각각 인터럽트를 사용할 때 충돌이 일어나 servo motor 가 오작동을 일으켰다. 나노 보드를 추가해주어 servo 모터 회로를 분리하였다.

○ 센서, 모터 데이터 값에 노이즈가 발생하여 안정적이고 신뢰도 높은 데이터를 얻기 어려웠다.

- 하드웨어적(LPF), 소프트웨어적(LPF, Moving average filter)를 적용하여 값의 데이터 플로팅 해본 결과, 안정적인 데이터를 얻었다.
- 그러나 과도한 필터링은 지연 시간을 최소화 해야 하는 RTOS System 의 성능을 저하하는 요인이 되므로, 실험적으로 적절한 필터 정도를 찾게 되었다.

○ HC - 06의 데이터 송수신 과부하

- HC-06 은 센서값을 유니티로 송신하고 있는 과정에서 유니티에서 오는 데이터를 정확하게 수신하지 못했다. 하여 아두이노에서 수신하는 데이터에 안전성을 추가하기 위해 데이터를 검증하는 알고리즘을 추가하여 이 문제를 해결하였다.

○ 웨어러블 기구는 로봇이나 기계 부품을 모델링하는 것에 비해 제약조건이 많아 난이도가 높았다.

- 사람마다 다른 손의 크기와 모양 때문에 전체적인 하드웨어를 인체공학적으로 설계해야만 하는 문제점이 있다.

- 다양한 사람들이 제품을 이용할 수 있도록 형태변환이 가능한 Soft design 을 이용했다.

- 경량화를 위해 3D 프린터로 플라스틱 재질의 제품을 제작했고, 불필요한 구조는 최소화하였다.

- 신체와 닿는 부분의 착용감을 극대화 하기 위해서 10 번 이상 설계, 출력, 피드백을 반복하며 3D 모델링을 진행하였다.

○ Gyro Sensor를 통한 절대 좌표 생성

- 초기 모델 구축 시 컨트롤러의 위치값을 산출하기 위해 Gyro Sensor 를 사용하는 것으로 설계했다. 하지만 Gyro 의 정확한 제어가 불가능했고, Stereo Camera 와 OpenCV 를 통해 이를 해결하였다.

□ 개발결과물의 차별성

gloVR 은 뇌졸중 환자를 위한 VR 장갑 컨트롤러와 프로그램이다. 현재 서울대병원, 단국대병원 등 국내 20 개 의료기관과 파트너십을 체결한 네오펙트사의 VR 글러브 컨트롤러 'Smart Glove'가 존재한다. 'Smart Glove'와 비교하여 gloVR 만이 가진 강점은 여러 가지가 있다.

- 1) 손가락마다 스프링의 길이를 조절하여 환자의 상태에 맞춰 재활치료가 가능하다. 상태가 좋아질수록 높은 저항력으로 게임을 진행하여 빠르게 상지기능을 회복할 수 있다.
- 2) gloVR 컨트롤러에 맞는 콘텐츠를 직접 제작한다. 스포츠 분야, 음악 분야, 실생활 분야 등의 장갑 컨트롤러를 이용한 다양한 게임 콘텐츠가 계획되어 있다. 제작한 콘텐츠들은 계속 쌓이고, 사용자들은 점점 더 많은 gloVR 콘텐츠를 즐길 수 있다.
- 3) 가격 경쟁력을 갖춘다. 현재 'Smart Glove'의 납품가는 약 1000 만원으로 측정된다. 이와 달리 gloVR 은 재활에 필요한 최소한의 센서와 전자부품으로 제작비와 재료비가 줄어든다.

○ 우수성

- 1) 재활 효과가 좋다.

게임을 통한 치료법은 시각적인 자극을 통해 광범위한 운동 관련 대뇌조직을 활성화해 재활치료의 효과를 높일 수 있다.

- 2) 꾸준한 재활 치료를 유도한다.

기존 재활치료는 단순한 작업만을 반복하여 흥미와 동기부여가 떨어지고 시간이 지날수록 환자의 치료 참여도가 떨어지는 단점이 있다. 게임 재활치료를 통해서 성취감과 흥미를 느껴 꾸준히 재활을 지속하도록 돕는다.

- 3) 설계구조가 단순하여 설치와 유지보수가 쉽다.

- 4) 가볍다.

가벼운 소재를 사용하여 손목에 무리가 없도록 하였다. (3D printing)

- 5) 치료에 사용되는 소프트웨어를 별도로 제작한다. 장치에 최적화된 소프트웨어는 장치의 성능을 최대화한다.

- 6) VR 기기와 분리해서도 사용할 수 있다. 단순 운동을 위한 장치로서 사용할 수 있다.

□ 개발 일정

내용	날짜	6월				7월				8월				9월				10월			
		1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주
기능설계		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
HW기능구현		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
SW기능구현		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
시제품 제작		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
기능 개선		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
통신 규약 설정		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
최종제품제작		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
제품 테스트		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			
제품 정리, 개선		[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]				[Green Bar]			

HW(Arduino)
정광서, 공성현

SW(통신)
박정무

SW(OpenCV)
유은광

SW(Unity)
김영서

□ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	박정무	<p>Arduino, Unity, OpenCV간의 통신 프로토콜 생성.</p> <p>BlueTooth, UDP, socket을 이용한 데이터 송수신.</p> <p>C++ 기반의 Datahandler 파일을 생성하여 Sensor 값과 Actuator 값의 형태를 정하고 통신속도를 최적화.</p>
2	팀원	유윤광	<p>Python 기반의 OpenCV를 이용해 장갑의 위치 파악.</p> <p>노트북 웹캠과 카메라 한 대가 보여주는 두 영상을 이용하여 장갑의 공간 위치 파악 프로그래밍 제작</p> <p>UDP, socket, Thread 를 이용한 Python-Unity 데이터 송수신</p>
3	팀원	김영서	<p>Unity를 이용한 캐치볼 게임 제작.</p> <p>게임의 배경과 디자인을 결정하고 제품의 데이터 값을 받아서 게임상에서 손을 움직여 날아오는 공을 잡을 수 있도록 프로그래밍.</p> <p>원활한 게임 플레이를 위한 UI 디자인.</p>
4	팀원	정광서	<p>임베디드 시스템 제어(보드 및 센서 제어)</p> <p>장갑의 3d modeling과 전체적인 하드웨어 디자인.</p> <p>C++ 기반의 Datahandler 파일을 생성하여 Sensor 값과 Actuator 값의 형태를 정하고 통신속도를 최적화.</p>
5	팀원	공성헌	<p>Arduino를 이용한 Actuator 설계.</p> <p>Gyro 센서와 Servo 모터의 이용을 위한 프로그래밍 및 구조설계.</p> <p>장갑의 3d modeling과 전체적인 하드웨어 디자인.</p>