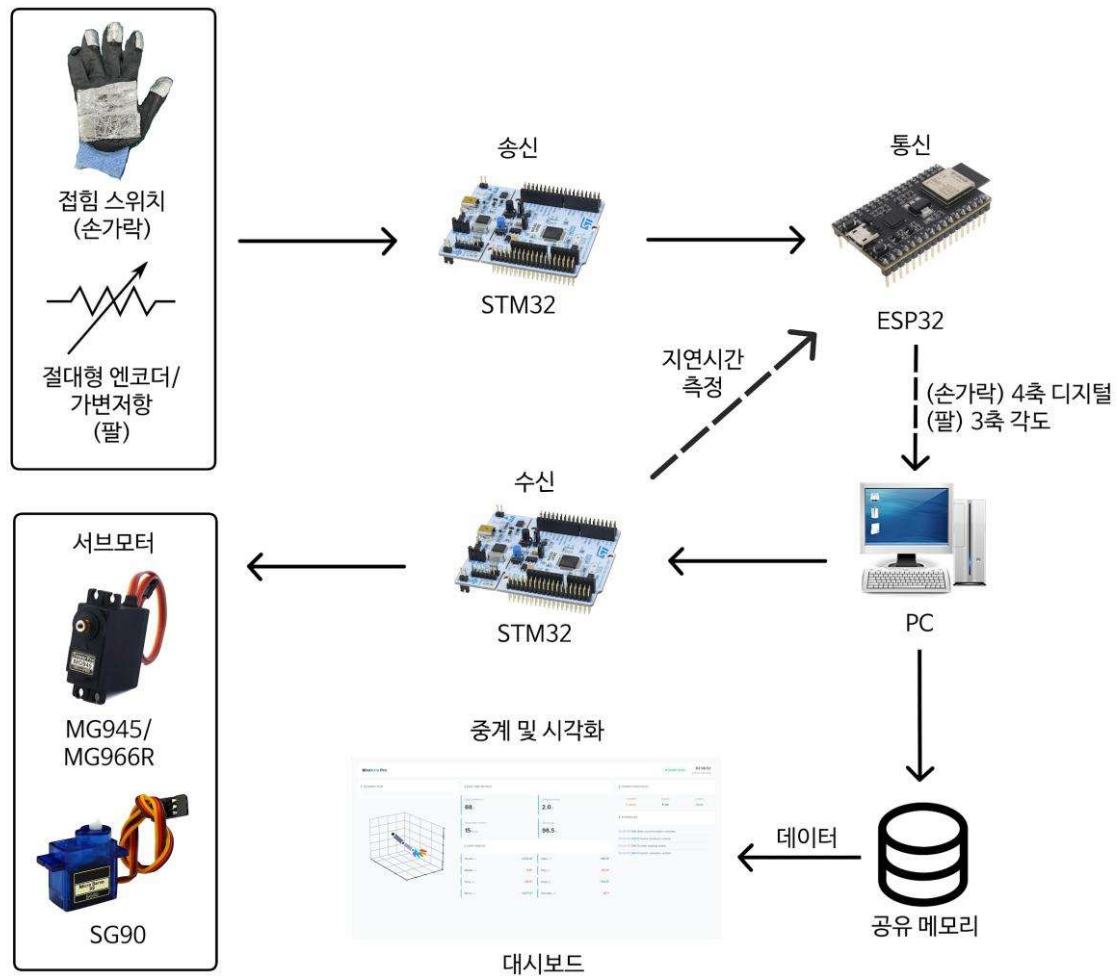


1. 시스템 전체 구성도



본 시스템은 인간의 동작을 감지하여 원거리의 로봇 팔에 실시간으로 동기화하기 위한 감지-패킷화-통신-제어-피드백의 5 단계 초저지연 프로세스를 기반으로 설계되었다.

송신단 (Wearable Transmitter): 사용자의 팔 움직임(3 축 각도)과 손가락 상태(4 축 디지털)를 수집한다. STM32 F446RE MCU 가 데이터를 처리하여 ESP32 의 블루투스 모듈을 통해 PC 로 전송하는 역할을 담당한다.

중계 및 시각화단 (PC Dashboard): Python 멀티프로세싱 아키텍처를 기반으로, 수신된 데이터를 Zero-Copy 방식의 공유 메모리(Shared Memory)에 저장하고 실시간 웹 대시보드에 시각화한다. 이와 동시에 로봇 제어 명령을 하향 전송하는 핵심 중계 역할을 수행한다.

수신단 (Robot Receiver): PC로부터 유선 시리얼로 수신된 명령 패킷을 STM32가 파싱 및 검증하여 총 7개의 서보모터(MG996R, SG90 등)를 구동 및 인간의 동작을 생체 모방 방식으로 재현하며, 명령 수행 후 ACK 신호를 송신단으로 역반환하여 지연시간을 측정한다.

2. 제품 전체 사진 및 동작 설명

2.1. 전체 시스템 외관

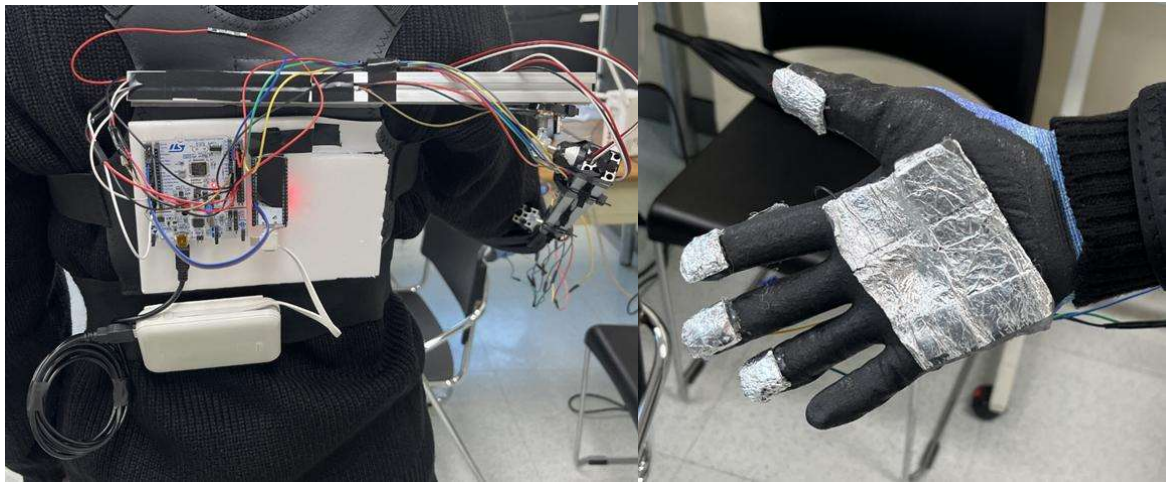
본 프로젝트인 '저지연 실시간 동작 모방 로봇팔'은 사용자의 팔 동작을 실시간으로 감지하는 웨어러블 송신부와 원격에서 해당 동작을 모방하는 로봇 팔 수신부, 그리고 두 시스템 간의 통신 중계 및 상태를 시각화하는 PC 대시보드로 구성된다.



'전체 시스템 구성. 사용자의 상체 움직임과 손가락의 굽힘을 로봇팔이 실시간으로 모방한다.'

시스템은 총 7축(어깨 2축, 팔꿈치 1축, 손가락 4축)의 제어를 수행하며, 블루투스 및 시리얼 통신을 결합한 하이브리드 통신 방식을 채택하여 End-to-End Latency를 평균 30ms 이내로 유지하도록 설계했다.

2.2. 송신부 (Wearable Controller) 상세 구성



송신부 하드웨어. STM32F446RE 보드와 ESP32 모듈, 각도 측정 센서가 장착되어 있다. 손부분은 호일 스위치 센서가 달려 있다.

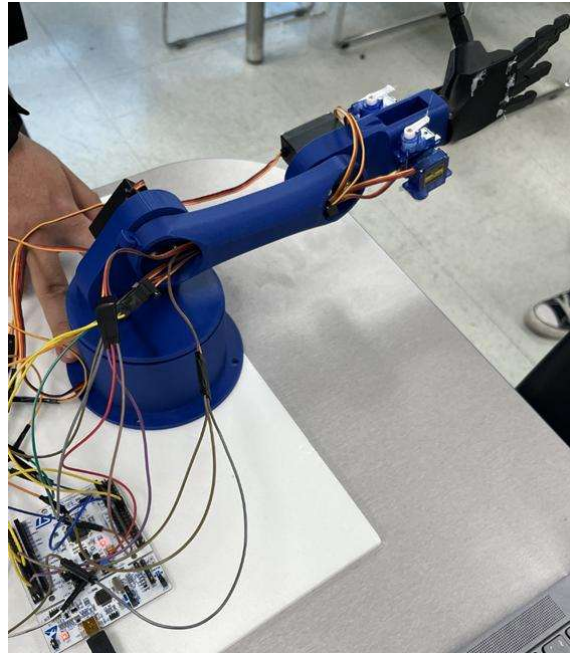
관절 각도 측정 (3-DOF): 어깨의 상하(Pitch), 좌우(Yaw) 움직임과 팔꿈치의 굽힘(Pitch)을 측정하기 위해 2 개의 절대형 엔코더와 1 개의 가변저항을 관절 회전축에 배치하였다. 각각 PWM 값과 저항 값을 추출하여 정밀한 각도 값을 계산했다.

손가락 굽힘 감지 (4-DOF): 엄지, 검지, 중지, 약지(새끼 포함)의 4 개 채널을 독립적으로 감지한다. 초기에는 힘 센서로 구부림을 감지하려 하였으나, 5 개의 힘 센서가 모두 불량으로 판정되어 그 대신 전도성 호일 스위치 방식을 고안하였다. 손가락 끝과 손바닥에 전도성 접지면을 구성하여, 주먹을 쥐었을 때 회로가 연결되는 스위치 원리를 이용해 0(굽힘)과 1(펴짐)의 디지털 상태를 인식한다.

웨어러블 본체: 견갑골의 hing 운동을 수용하기 위해 어깨 관절부에 레일형 슬라이딩 구조를 도입하여 착용감 및 측정 정밀도를 획기적으로 개선하였다.

제어 및 통신 모듈: 제어 통신 모듈은 STM32 과 ESP32 이 두가지 보드를 통해서 구현하였다. STM32 보드는 센서의 아날로그 및 디지털 신호를 수집하고 노이즈를 제거한 뒤, 15 바이트의 데이터 패킷을 생성한다. 또한, 파란색 사용자 버튼(B1)을 통해 현재 자세를 0 점으로 설정하는 '영점 조절' 기능을 수행한다. ESP32 보드는 STM32 로부터 UART 로 전달받은 패킷을 블루투스를 통해 PC 로 무선 전송한다. 특히 RTT(Round Trip Time) 측정을 위해 PC 로부터 되돌아오는 ACK 신호를 다시 STM32 로 전달하는 양방향 통신 브리지 역할을 수행한다.

2.3. 수신부 (Robot Arm) 상세 구성



로봇팔 수신부. 서보모터와 와이어 구동 방식의 로봇손으로 구성된다.

로봇팔 관절: 어깨와 팔꿈치 관절에는 높은 토크가 요구되므로 MG996R/MG945 고토크 서보모터를 사용하였다. 수신된 각도 데이터를 PWM 듀티비로 변환하여 1:1 매핑으로 제어된다.

로봇손: 인체의 전완근과 힘줄 구조를 생체 모방하여 설계하였다. 전완부에 위치한 4 개의 SG90 마이크로 서보모터가 와이어를 당겨 손가락을 굽히고, 탄성체를 이용해 다시 펴지는 구조를 갖는다. 송신부의 손가락 센서 값(0/1)에 따라 서보모터를 즉각 구동하여 손가락을 접거나 펴는 동작을 수행한다.

2.4. 시스템 동작 원리 및 데이터 흐름

이 '저지연 실시간 동작 모방 로봇팔' 시스템은 '센싱 → 패킷화 → 중계 및 처리 → 구동 → 피드백'의 5 단계 과정을 거쳐 실시간 동기화를 수행한다.

2.4.1. 데이터 패킷 구조 및 통신

송신부는 센서 데이터를 수집하여 15 바이트의 고정 길이 패킷을 생성한다. 패킷은 헤더(0xAA, 0x55), 3 축 각도 데이터(각 2 바이트), 4 개의 손가락 상태(각 1 바이트), RTT 측정용 타임스탬프, 그리고 데이터 무결성 검증을 위한 체크섬으로 구성된다.

2.4.2. 동작 프로세스

데이터 수집: STM32 는 가변저항의 ADC 값을 각도로 변환하고, 파란색 버튼이 눌린 시점의 각도를 오프셋(Offset)으로 저장하여 사용자의 체형에 맞는 영점을 설정한다.

무선 전송 및 중계: 생성된 패킷은 ESP32 를 통해 PC 의 Python 수신 프로세스로 전송된다. 이때 Python 프로그램은 멀티프로세싱 기반의 공유 메모리(Shared Memory)기법을 사용하여 데이터 복사 오버헤드를 최소화함으로써 고속 처리를 보장한다.

로봇 구동: PC 에서 유선 시리얼로 전달받은 데이터를 로봇 측 STM32 가 파싱한다. 체크섬 검증을 통과한 유효한 데이터에 한해 각 관절의 서보모터 PWM 신호를 갱신하여 즉각적인 움직임을 만들어낸다.

RTT 피드백: 저지연 성능을 검증하기 위해, 로봇팔은 명령을 수행한 직후 ACK 신호를 역순(로봇→PC→송신부)으로 반환한다. 송신부는 패킷 전송 시점과 ACK 수신 시점의 차이를 계산하여 전체 시스템의 지연시간을 실시간으로 측정한다.

2.4.3. 대시보드 모니터링

PC 에서는 별도의 대시보드(app.py)를 통해 현재 관절의 각도, 손가락 상태, 그리고 실시간 Latency 그래프를 시각화하여 시스템의 상태를 한눈에 모니터링할 수 있다.

3. 대시보드 소개



3.1. 전체 대시보드 소개

웹 대시보드에서는 로봇팔의 현재 상태를 확인할 수 있는 지표들을 확인하기 쉽도록 보여주는 것을 목표로 하였다. python 라이브러리 streamlit 을 사용해 웹 페이지를 제작하였다. PC 의 share_data 코드에서 ShareMemoryReader 를 불러와 공유 메모리를 설정하고 shm_reader.read() 함수를 사용해 실시간으로 데이터를 업데이트한다. 또한, 시스템의 무결성을 검증하고 통신 품질을 정량화하는 데 중점을 두었다

3.2. 대시보드 구역별 소개

3.2.1 로봇 시뮬레이션

3 축 그래프를 통해 로봇팔의 현재 상태를 시각화했다. 검은색 점은 어깨를 의미하는 기준점이며, 파란색 선은 상완(어깨~팔꿈치), 보라색 선은 하완(팔꿈치~손)을 의미한다. 그래프 하단에는 가장 최근에 전송한 각도 데이터를 표시했다.

그래프는 GRAPH_UPDATE_INTERVAL 이라는 상수를 통해 카운트 후 로딩하는 방식으로 지연을 방지했다.

3.2.2 실시간 성능

E2E Latency 는 센서에서 로봇까지 편도 지연 시간을 나타내며 원격 제어의 핵심 지표를 정량적으로 보여준다.

Packet Rate 는 초당 전송된 패킷 수를 의미하며, 시스템의 실시간 처리 능력을 나타낸다.

3.2.3 시스템 로그

초기 로그는 성공 시 [INFO] 공유 메모리 연결됨, [INFO] 궤적 추적 활성화를 표시한다.

실패 시에는 [INFO] 공유 메모리 연결 실패, [INFO] main.py 먼저 실행하세요를 표시한다.

이후 실시간으로 latency 를 [INFO] #1234 | E2E=20.0ms | RTT=40.0ms 와 같이 표시한다. #1234 는 패킷 카운트를 의미한다.

3.2.4 연결 상태

ESP32 와 PC 사이의 블루투스 연결, PC 와 대시보드 사이의 공유 메모리 연결, 현재 지연 시간을 표시한다.

3.2.5 핵심 지표

E2E Latency, Packet Rate, RTT, Total Pkts 값을 실시간으로 업데이트한다. Total Pkts 는 현재까지 전송받은 패킷의 누적 갯수를 의미한다.

3.2.6 그리퍼

손가락의 접힘 스위치 상태를 표시한다. 엄지, 검지, 중지, 약지 이렇게 네 개의 손가락을 표시한다. 소지는 약지와 함께 움직이므로 따로 측정하지 않았다. 만약 손가락이 손바닥에 닿았다면 ON, 아니라면 OFF 를 나타낸다.

4. 팀원별 업무 분장

성명 (학번)	담당	주요 업무 및 수행 내용
이호윤(202111055)	송신팀	송신부(STM32) 코드 구현 및 디버깅 웹 대시보드 코드 통합 및 최종 코드 관리
채예원(202310985)	송신팀	송신부(STM32) 데이터 패킷 정의 및 코드 구현 웹 대시보드 송수신 연결 상태 시각화 및 발표자료 제작
김다연(202310902)	수신팀	수신부 데이터 파싱 및 모터 제어 코드 구현 웹 대시보드 센서 각도 값 표 구현 및 발표자료 제작
안민철(202111048)	수신팀	로봇팔 디자인 및 수신부 하드웨어 설계/제작 수신팀 코드 통합 및 하드웨어 조립 주도
서형완(202110937)	통신팀	블루투스(ESP32) 통신 코드 구현 및 데이터 중계 웹 대시보드 3 축 그래프 시각화 및 최종 수정

5. 작업 일정표

주차	작업일정
9 주차	프로젝트 주제 확정 및 센서(엔코더, 압력센서) 수령 시스템 아키텍처 설계 및 하드웨어 구성 논의
10 주차	역할 분담(송신/수신/통신) 확정 및 하드웨어 1 차 설계 로봇팔 주요 부품 3D 프린팅 출력 및 수령
11 주차	각 팀별 단위 코드 개발(STM32 송신, ESP32 통신, PC 수신) 송수신 데이터 패킷 구조 정의 및 체크섬 도입
12 주차	서보모터(손가락 제어용) 추가 수령 및 하드웨어 조립 웨어러블 입력부 제작 및 영점 조절 기능 구현
13 주차	전체 시스템 통합 테스트 및 대시보드 시각화 연동 시연 영상 촬영 및 발표 자료 초안 작성
14 주차	건갑골 움직임 구현 등 하드웨어 최종 보완 지연 시간 최적화

6. 추가된 기술에 대한 상세 설명

6.1 초저지연을 위한 Zero-Copy IPC (공유 메모리)

통신 환경에서 블루투스 포트와 시리얼 포트가 단일 프로세스에 의해 점유되어 타 프로세스가 접근할 수 없는 문제를 해결하기 위해 멀티프로세싱(Multiprocessing) 아키텍처를 도입하고 Shared Memory 를 활용했다.

- **문제 인식:** Python 의 GIL(Global Interpreter Lock)로 인한 처리 지연 가능성 및 블루투스 포트의 단일 점유 문제를 해결해야 했다.
- **해결:** 수신 프로세스는 ESP32 로부터 블루투스 데이터를 수신 즉시 공유 메모리(Shared Memory)에 기록한다. 대시보드 시각화 프로세스와 로봇 제어 명령 프로세스는 이 공유 메모리를 별도의 데이터 복사(Zero-Copy) 과정 없이 직접 참조한다. 이는 소프트웨어

레벨에서의 지연 시간을 극단적으로 최소화하며, 고속 처리를 보장하는 핵심 기술이라고 볼 수 있다.

6.2 데이터 무결성 및 정밀 통신 신호 처리

- **Checksum 검증:** 15 바이트 패킷의 말단에 체크섬 비트를 포함하여, 무선 통신 환경에서 발생하는 비트 오류(Bit Error) 또는 패킷 손실을 필터링하고 유효 데이터에 한해서만 로봇을 구동하도록 하여 시스템의 안정성을 확보했다.
- **소프트웨어 디바운싱:** 접촉식 호일 스위치에서 발생 가능한 채터링(Chattering) 현상을 제거하기 위해, 연속 3 회 샘플링 시에만 유효 입력으로 판정하는 알고리즘을 송신단 펌웨어에 적용하여 신호의 신뢰성을 높였다.
- **정밀 RTT 측정 및 검증:** 시스템의 초저지연 성능(Ultra-Low Latency)을 정량적으로 증명하기 위해, 송신부에서 보낸 패킷이 로봇의 구동 명령을 거쳐 ACK 응답을 받아 돌아오기까지의 Round Trip Time (RTT)을 마이크로초(μ s) 단위로 실측했다. 이 RTT 값을 기반으로 실측 편도 지연 시간(E2E Latency)이 7.5ms 임을 입증하였으며, 이는 원격 제어 및 통신 분야에서 매우 우수한 수준이라고 볼 수 있다.

7. 시행착오 및 개선 과정

7.1. 시행착오 1: 힘 센서 불량으로 인한 손가락 접힘 인식 문제

7.1.1. 문제 상황

로봇팔의 손가락 제어를 위해 장갑에 힘 센서(Flex Sensor)를 부착하기 전, 브레드보드를 통해 센서 인식 테스트를 실시하였다. 그러나 구입한 모든 힘 센서에서 손가락의 구부림 정도에 따른 저항 값 변화가 거의 나타나지 않거나 무의미한 수치만 출력되는 현상이 발생하였다.

7.1.2. 원인 분석

관련 기술 자료 및 제조사 사양을 검토한 결과, 사용된 힘 센서의 길이가 짧을 경우 구부리는 정도로는 저항 값이 변하지 않을 수 있음을 확인하였다.

7.1.3. 대응 및 검증

추가적인 긴 힘 센서를 재구매하는 대신 손가락의 완전한 '접힘'과 '펴짐'을 명확히 구분할 수 있는 접촉식 디지털 스위치 방식으로 설계를 변경하였다.

- 손가락 끝마디와 손바닥의 맞닿는 지점에 전도성이 우수한 알루미늄 접지면을 구성
- 풀업(Pull-up) 회로 구성을 통해 손가락을 뺐을 때에는 신호가 1(High) 상태를 유지하도록 설계
- 손가락을 완전히 굽혀 알루미늄 면끼리 접촉될 때, 회로가 그라운드(GND)와 연결되어 신호가 0(Low)로 떨어지도록 구성

검증: 시리얼 모니터를 통해 손가락 접촉 시 신호가 1 에서 0 으로 즉각적으로 명확하게 전환되는지 확인하였으며, 플로팅(Floating) 현상이나 채터링(Chattering) 없이 동작의 신뢰성을 확보하였다.

7.1.4. 결과 및 의의

디지털 스위치 방식 적용 이후 손가락 접힘 인식이 안정적으로 동작하였으며, 센서 길이 및 특성에 따른 센서 인식 불안정 문제를 해소할 수 있었다. 또한, 구입한 힘 센서 모두 불량이라는 위기 속에서 기존 방식에 얽매이지 않고, 풀업 저항 원리를 응용해 추가 예산 투입 없이 주변 재료를 활용하여 문제를 극복할 수 있었다.

7.2. 시행착오 2: 절대형 엔코더 열 손상에 따른 관절 각도 측정 데이터 오류

7.2.1. 문제 현상

입력부 웨어러블 장치 제작 초기, 3 개의 절대형 엔코더 단품 테스트 시에는 모두 정상적인 각도 데이터를 출력하였다. 그러나 어깨 2 축 관절 부위를 고정하여 웨어러블 형태를 완성한 후 디버깅 과정에서 치명적인 데이터 오류가 발생하였다. 관절을 움직일 때 각도가 선형적으로 변하지 않고, 특정 각도인 25 도에서 65 도로 급격하게 수치가 튀는 등 비정상적인 데이터 변동이 발견되었다.

7.2.2. 원인 분석

처음에는 각도 변환 알고리즘이나 부동 소수점 연산 등 소프트웨어적인 코드 오류를 의심하여 소스 코드를 검토하였으나 코드 상의 결함을 찾지 못해 하드웨어 신호를 직접 분석하기로

하였다. 엔코더에서 출력되는 Raw PWM 신호를 시리얼 모니터로 모니터링한 결과, 특정 각도에서 듀티 사이클 자체가 불규칙하게 출력되는 것을 확인하였고, 이는 단순 코드 오류가 아닌 센서 자체의 결함임을 파악하였다. 근본 원인을 생각한 결과 부품 고정 과정에서 사용한 글루건의 열이 원인으로 분석되었다. 어깨 2 축의 기구적 고정을 위해 센서 인근에 고온의 글루건을 사용했는데, 이때 발생한 열이 절대형 엔코더 내부의 정밀 칩셋 또는 자기 센서 소자에 열 손상을 입혀 오작동을 유발한 것으로 판단되었다.

7.2.3. 대응 및 검증

프로젝트 일정 및 부품 수급 현황을 고려하여, 손상된 엔코더를 대체할 수 있는 가변저항을 활용한 각도 측정 방식을 채택하였다.

- 가변저항의 아날로그 전압 출력(0~3.3V)을 MCU의 ADC로 입력
- 읽어들인 아날로그 수치(0~4095)를 실제 관절 가동 범위인 각도 값으로 변환하는 맵핑 함수를 코드에 적용

검증: 가변저항으로 교체한 후 관절 회전 시 데이터 점핑 현상이 사라졌으며, 실제 움직임과 출력되는 각도 값이 일치하는 선형적인 데이터를 얻음으로써 동작의 신뢰성을 확보하였다.

7.2.4. 결과 및 의의

고가의 엔코더 없이도 가변저항을 활용하여 정밀한 관절 각도 피드백 시스템을 재구축하였으며, 로봇팔 제어의 안정성을 회복하였다. 특히 특정 부품의 손상이라는 돌발 상황에서 시스템의 원리(회전량 측정)를 파악하고, 이를 대체할 수 있는 최적의 솔루션을 찾아내어 프로젝트를 완수하였다.

7.3. 시행착오 3: 건갑골 유동성을 고려한 어깨 관절부 구조 설계 변경

7.3.1. 문제 현상

입력부 웨어러블의 어깨 2 축 관절을 본체에 완전히 고정한 형태로 제작하여 착용 테스트를 진행하였다. 사용자가 팔을 양옆으로 벌리거나 안으로 모으는 수평 내전 및 외전 동작을 수행할 때, 웨어러블 장치가 몸에서 들뜨거나 조이는 압박감이 발생하였다. 기구부의

구속으로 인해 실제 사용자의 어깨 가동 범위가 제한되고, 이로 인해 로봇팔로 전달되는 각도 데이터에 왜곡이 생기는 현상이 관찰되었다.

7.3.2. 원인 분석

어깨 관절의 움직임은 단순히 상완골(팔뼈)의 회전만으로 이루어지는 것이 아니라 견갑골이 흉곽을 따라 미끄러지듯 함께 움직이는 복합적인 구조임을 간과하였다. 고정식 2 축 구조는 회전 중심점이 고정되어 있어, 수평 운동 시 견갑골이 바깥쪽이나 안쪽으로 이동하는 횡 운동을 수용하지 못한다. 이로 인해 인체의 회전 중심과 장비의 회전 중심이 어긋나는 축 불일치 문제가 발생하였다.

7.3.3. 대응 및 검증

어깨 관절의 회전축 자체가 사용자의 움직임에 따라 유동적으로 변할 수 있도록 레일형 슬라이딩 구조를 도입하였다.

- 어깨 2 축 관절을 본체에 직접 고정하지 않고, 이너 브라켓을 추가하여 설계
- 해당 브라켓이 레일 역할을 수행하여, 견갑골의 움직임에 맞추어 관절 뭉치 전체가 좌우로 부드럽게 이동할 수 있도록 개조

검증: 개조 후 착용 테스트 결과, 팔의 수평 내전/외전 시 웨어러블 장치가 견갑골의 라인을 따라 자연스럽게 슬라이딩됨을 확인하였고 장치의 이탈이나 압박감 없이 인체의 가동 범위를 수용할 수 있게 되었다.

7.3.4. 결과 및 의의

사용자의 착용감을 획기적으로 개선하였으며, 인체와 웨어러블 기기의 밀착도를 높여 관절 각도 측정의 정밀도를 향상시켰다. 뿐만 아니라 단순한 기계적 매커니즘을 넘어 인체 해부학적 특성을 공학적으로 해석하고 이를 기구 설계에 반영하는 능력을 배양할 수 있었다.