

2021년도 임베디드시스템 팀프로젝트 최종보고서

(프로젝트 명 : 자이로센서를 이용한 자세교정 프로그램)

2021. 06. 15.

팀 명 : 4인분 같은 2인분

팀원명단 : 남유정, 황진주

1. 프로젝트 목표 및 내용

가. 프로젝트 목표(개요) 및 변경내용

구 분	당초계획	변경내용	변경사유
목표(개요)	자이로 센서값을 이용한 등,허리 자세교정 프로그램	x	x

나. 프로젝트 개념도 또는 개요

1. 그래픽 디바이스에 팀로고를 출력한다.
2. 인터럽트 버튼을 통해 **인터럽트 신호**를 보내게 되면, **멜로디**가 출력된다.
3. 사용을 위한 **매뉴얼**이 **그래픽 디바이스**에 출력된다. (자세교정모드, 스트레칭모드)
4. **자세교정모드** 선택을 위한 버튼인 **버튼**을 누른다.
5. 그래픽 디바이스의 안내에 따라 원하는 부위를 고른다. (거북목 교정, 일자허리 교정)
6. 교정이 시작되면 프로세싱은 **자이로 센서값**을 받아 현재 모습을 **3D 인체 모델**로 보여준다.
7. 올바르지 않은 자세를 취하면 **해당 부위가 빨강**게 변하고 그렇지 않으면 **초록색**으로 유지된다.
8. 재시작 후 1~4의 과정으로 **스트레칭모드**를 선택한다.
9. 목, 허리 중 원하는 스트레칭 모드를 버튼을 눌러 선택한다.
10. 그래픽 디바이스에 그려진 **이미지**에 맞게 동작을 따라한다.
11. 동작이 완료된 후 다시 메인 화면으로 돌아오게 된다.

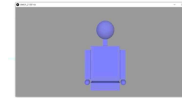
다. 프로젝트 세부(혹은 단계별) 목표 및 추진한 내용

세부 목표	추진 내용
물체 모델링 및 자이로 센서값 연동	<ul style="list-style-type: none"> - 프로세싱에서 인체 모형을 본 딴 3D 입체 모델 제작 - 행렬 스택을 이용하여 신체 중 특정 부위(목, 허리)만 움직이도록 구성 - 시리얼 통신 값에 따라, 제어 모드인 경우에만 제어 부위의 색상 변경 - myahrs+ 자이로 센서를 이용한 pitch, yaw, roll값 측정 - 아두이노에서 정제한 자이로 센서값 pitch, yaw, roll 값 시리얼 통신 - '올바른 자세에'에 대한 각도 측정 후 프로세싱 제어 코드 적용
그래픽 디바이스 이미지 제작 및 멜로디 작성	<ul style="list-style-type: none"> - 동작에 맞는 메뉴얼, 이미지 등을 제작 - 제작 이미지를 흐름에 맞도록 순서를 제어하는 모드들을 작성 - 특정 상황에 맞는 멜로디 함수 startPlay()를 제작
인터럽트 발생 동작 및 7세그먼트 출력 작성	<ul style="list-style-type: none"> - 인터럽트 버튼을 통해 프로그램 시작/종료를 알 수 있도록 구성 - 인터럽트 발생 시 멜로디가 나와 프로그램이 시작/종료 됨을 알림 - 인터럽트가 발생해야만 각 버튼 제어가 가능하도록 작성 - 7세그먼트의 출력이 회전 : 신호 입력 대기 - 7세그먼트의 출력이 멈춤 : 프로그램 가동 중, 신호 입력 받지 않음
아두이노 모듈간 연결 (인터럽트, 버튼핀, 그래픽디바이스, 문자LCD)	<ul style="list-style-type: none"> - 현재 진행되는 모드, 상황에 대해 문자 LCD에 출력 - 버튼 동작에 따라 프로그램 제어 모드가 변경되어, 상황에 맞는 그래픽 이미지를 출력하도록 함 - 버튼 제어로 설정된 프로그램 제어 모드가 전송 모드인 경우에만 자이로 센서값을 전송하도록 함 - 스트레칭 프로그램이 종료될 시 메인화면으로 복귀하여 재실행 가능

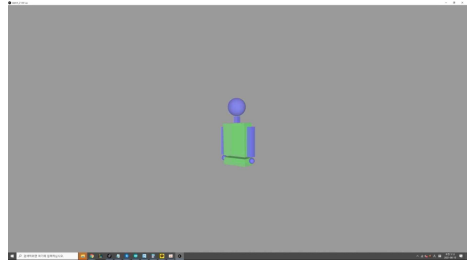
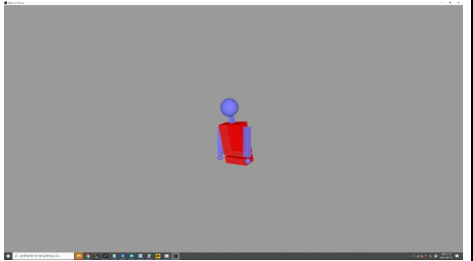
2. 프로젝트수행 결과

가. 프로젝트 결과물

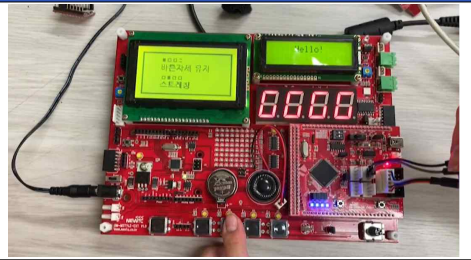
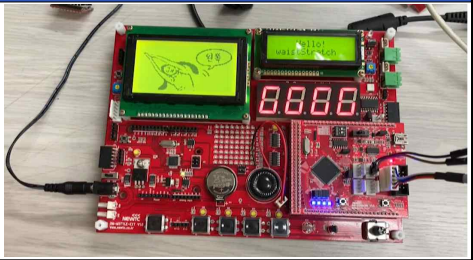
1) 결과물 동작 상태 확인용 사진



◀ 자세교정 모드 입력 전 디폴트 출력
방향키 조작을 통해 시점 변환 가능

자세교정 프로그램(허리)	
	
<ul style="list-style-type: none"> - 허리교정 프로그램 가동 시 프로세싱의 모습 - 올바른 자세인 경우 허리 부분이 녹색으로 출력 	<ul style="list-style-type: none"> - 허리교정 프로그램 가동 시 프로세싱의 모습 - 자세가 올바르지 못 할 때 빨간색으로 출력

자세교정 프로그램(목)	
	
<ul style="list-style-type: none"> - 목교정 프로그램 가동 시 프로세싱의 모습 - 올바른 자세인 경우 머리와 목이 녹색으로 출력 	<ul style="list-style-type: none"> - 목교정 프로그램 가동 시 프로세싱의 모습 - 자세가 올바르지 못 할 때 빨간색으로 출력

메뉴 선택 및 스트레칭	
	
<ul style="list-style-type: none"> - 그래픽 디바이스에 표기된 버튼에 따라 메뉴선택 - □■□□ : 핀번호 15번을 누르면 해당 모드 이동 	<ul style="list-style-type: none"> - 스트레칭 모드 진입에 따른 그래픽 디바이스 출력 - 모드에 맞는 애니메이션 출력

2) 결과물 구동 소스 코드

버튼 입력에 따른 모드 변화 (모드선택 제어)			
pullBtn1()	pullBtn2()	pullBtn3()	pullBtn4()
<pre>// 스트레칭 or 교정 if (startMode == 1) { icon = menu1; menuMode++; } // 자세 교정 선택 시 if (menuMode > 1) { icon = menu2; // 교정부위 선택지 활성화 partMode = 1; } // 교정부위 선택지 출력 GLCD.DrawBitmap(icon, 0, 0);</pre>	<pre>// 스트레칭 선택 시 if (menuMode > 1) { icon = menu3; // 교정부위 선택지 활성화 partMode = 2; } // 메뉴모드 활성화 menuMode = 2; // 스트레칭 부위 선택지 출력 GLCD.DrawBitmap(icon, 0, 0);</pre>	<pre>// 교정모드 일 때 if (partMode == 1) { // 목 교정 모드 설정 partSelectMode = 1; } // 스트레칭모드 일 때 if (partMode == 2) { // 목 스트레칭 모드 설정 partSelectMode = 3; } }</pre>	<pre>// 교정모드 일 때 if (partMode == 1) { // 허리 교정 모드 설정 partSelectMode = 2; } // 스트레칭모드 일 때 if (partMode == 2) { // 허리 스트레칭 모드 설정 partSelectMode = 4; } }</pre>

모드값에 의한 프로그램 제어 (Loop)
<pre>// 버튼 제어 모드(프로그램 시작, 메뉴선택, 작동부위 선택) : startMode, menuMode, partMode // 모드 컨버팅을 이용한 음악 반복재생 제어 : 음악재생 후 재생 모드를 0으로 해줌으로 다시 재생되지 않음. if (startMode == 1 && startMusicMode == 1) { startSound(); startMusicMode = 0; }</pre> <pre>// 버튼 입력에 따른 프로그램 제어 시의 모드 제어 // 핀이 HIGH되고, 이전값과 달라진 경우 == 버튼을 눌렀다 뗐을 경우 // 해당버튼 번호와, 프로그램 시작여부, 모드선택 여부에 따라 버튼 동작 여부가 갈림 if (digitalRead(14) == HIGH && prev1 != digitalRead(14)) { if (startMode == 1) pullBtn1(); } if (digitalRead(15) == HIGH && prev2 != digitalRead(15)) { if (startMode == 1) pullBtn2(); } if (digitalRead(16) == HIGH && prev3 != digitalRead(16)) { if (partMode != 0) pullBtn3(); } if (digitalRead(17) == HIGH && prev4 != digitalRead(17)) { if (partMode != 0) pullBtn4(); }</pre> <pre>if (partSelectMode == 1) neckBraces(); // 목교정 프로그램 실행 if (partSelectMode == 2) waistBraces(); // 허리교정 프로그램 실행 if (partSelectMode == 3) neckStretching(); // 목스트레칭 프로그램 실행 if (partSelectMode == 4) waistStretching(); // 허리스트레칭 프로그램 실행</pre>

스트레칭 애니메이션 출력	
<pre>// 허리 스트레칭 애니메이션 lcd.setCursor(1, 1); lcd.write("waistStretch"); // 현재 모드 문자 LCD에 출력 icon = waist1; // 각 단계에 맞는 이미지 출력 및 시간 조정 GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = waist2; // 이미지는 헤더 파일로 정의되어 있음 GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = waist3; GLCD.DrawBitmap(icon, 0, 0); delay(3000); ... // 중략 icon = endStretching; GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = menu1; GLCD.DrawBitmap(icon, 0, 0); partSelectMode = 0; // 모드 재설정이 가능하도록 초기화</pre>	<pre>// 목 스트레칭 애니메이션 lcd.setCursor(1, 1); lcd.write(" neckStretch"); // 현재 모드 문자 LCD에 출력 icon = neck1; // 각 단계에 맞는 이미지 출력 및 시간 조정 GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = neck2; // 이미지는 헤더 파일로 정의되어 있음 GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = neck3; GLCD.DrawBitmap(icon, 0, 0); delay(3000); ... // 중략 icon = endStretching; GLCD.DrawBitmap(icon, 0, 0); delay(3000); icon = menu1; GLCD.DrawBitmap(icon, 0, 0); partSelectMode = 0; // 모드 재설정이 가능하도록 초기화</pre>

모델 출력 및 화면 회전	
<pre>// 방향키로 모델을 회전시킴 (keyPress함수) if (key == CODED) { if (keyCode == UP) { y += 10; } // 위쪽 else if (keyCode == DOWN) { y -= 10; } // 아래쪽 else if (keyCode == LEFT) { x -= 10; } // 왼쪽 else if (keyCode == RIGHT) { x += 10; } // 오른쪽 } lights(); // 물체에 조명을 넣어 입체감을 넣음 translate(width / 2, height / 2, 0); // 창의 중점을 원점으로 rotateX(radians(y)); // 키보드 값에 의해 연산된 값을 X기준 회전 rotateY(radians(x)); // 키보드 값에 의해 연산된 값을 Y기준 회전</pre>	<pre>// 허리 교정 모드인 경우 if (mode[1]) { // 모델이 사용자의 등 움직임과 동일하게 움직임 rotateX(-(rot - 273) / 90.0); if (go) // 올바른 자세 fill(125, 225, 125, 200); // 하반신, 등 초록색 변경 else { // 올바르지 못한 자세 fill(255, 0, 0, 200); // 하반신, 등 빨간색 변경 } } else if (!mode[1]) { // 4번 모드(등)가 아닐 때 fill(125, 125, 255, 200); // 선택되지 않은 영역이므로 곤색 출력 }</pre>

데이터 수신 및 값 처리	
<pre>String input = port.readStringUntil("\n"); //println(input); if(modebutton==0){ // 목교정 모드 mode[0] = true; // draw의 목, 허리 제어할 부분 기준 정립 mode[1] = false; // draw의 목, 허리 제어할 부분 기준 정립 } else if(modebutton == 1){ // 허리 교정 모드 mode[0] = false; // 목 제어 불가능 mode[1] = true; // 허리 제어 가능 } // roll, pitch, yaw 값이 들어올 것을 알림 if (input.charAt(0) == 's') { cnt = 0; // 입력 개수 카운팅 초기화 } // 제어 부위 설정값이 들어올 것을 알림 else if (input.charAt(0) == 'b') { buttonInput = true; // 제어 부위 입력 조건문 활성화 } // 제어 부위 조작 모드 활성화 else if (buttonInput == true) { // 프로세싱에 맞도록 값을 정제 modebutton = input.charAt(0) - '0' - 3; buttonInput = false; // 입력 완료 후 입력 대기상태로 전환 }</pre>	<pre>else { val[cnt] = float(input); if (cnt >= 2) { // 3개 값이 모두 들어온 경우에만 배열에 저장 rotX += radians(prevX - val[1] + 360); rotY += radians(prevY - val[2] + 360); rotZ += radians(prevZ - val[0] + 360); // 올바른 자세일 때 if (val[1] + 360 > 270 && val[1] + 360 < 280) { go = true; // 올바른 자세임을 표시 } else { // 올바른 자세가 아닐 때 go = false; // 올바르지 못한 자세임을 표시 } // 회전값 및, 각 x, y, z에 대치되는 결과값을 저장 rot = val[1] + 360; prevX = val[1]; prevY = val[2]; prevZ = val[0]; } cnt++; // 입력받은 데이터 카운팅 증가 }</pre>

나. 참여 인원별 역할 및 수행 소감

성 명	역할 구분	수행 역할	수행 소감
남유정	팀장	<ul style="list-style-type: none">- 인체 모델링- 자이로 값에 의한 모델링 움직임 및 색상변환- 멜로디 사운드 출력- 그래픽 디바이스 로고 출력- 텍스트 LCD 출력- PPT 제작	구현 전 미리 했던 설계와 실제 출력이 달 랐던 부분에 대해 고쳐나가면서 작동원리 에 대해 정확히 알게 되었고 이 기능들을 캡스톤을 통해 좀 더 발전시켜 완벽에 가 까운 하나의 상품을 만들어보고 싶다는 생 각이 들었다.
황진주	조원	<ul style="list-style-type: none">- 자이로 센서 연결 및 시리얼 통신 코드 작성- 올바른 자세 각도 설정- 인터럽트를 이용한 실행- 버튼 입력기반 모드 전환, 이미지 출력 지점 연결- 그래픽 이미지 제작- 스트레칭 이미지 애니메이션화- 7세그먼트 출력	프로그램 흐름 작성 시 단순한 설계에도 많은 분기점이 존재한다는 걸 몸소 느끼게 되었다. 이보다 복잡한 게임이나 프로그램 에서의 분기 설정에 대해 공부를 해보고 싶다는 생각이 들었다.

다. 활용방안

활용분야	활용방안	비고
게임	박자, 이미지에 맞추어 동작을 따라하는 리듬게임 추가	향후
모바일 연동	블루투스 연동을 통한 모바일 제어 추가	
리플레이	교정 데이터와 리플레이 영상을 통해 나의 자세를 되돌아 봄	
스트레칭	스트레칭 안내를 통해 몸의 피로도를 낮추어줌	현재
자세 교정	올바른 자세를 잡을 수 있도록, 현재 자세를 실시간으로 점검함	