

REPORT

전자종합설계

디지털로그 프린터

조 원 : B688057 정재훈

B688044 이성원

B688059 조영관

B684050 황지환

제 출 일 자: 2021.12.05

< 목 차 >

1. 서 론

1.1 설계의 필요성

1.2 설계 목표

2. 현실적 제한요소

2.1 안전성

2.2 심미성

2.3 신속성

2.4 정확성

3. 합성

3.1 개념 설계

3.2 상세 설계

4. 구현 및 제작

4.1 제작과정

4.2 웹페이지 코딩

4.3 모터 제어 코딩

5. 결론

5.1 결과

5.2 평가 및 향후개발 사항

1. 서론

1.1 설계의 필요성

(1) 문제 인식

최근 자연주의를 표방한 아날로그적 감성이 트렌드로 부상하고 있다. 여러 기업들이 포장지 속 브랜드 로고 등을 반듯한 모양의 활자체에서 친근하고 감성적인 손글씨로 변경하는가하면, 음식점이나 카페에서도 심심치 않게 캘리그래피로 적힌 문구들을 발견할 수 있다. 게다가, 코로나19로 인해 실내에서 즐기는 취미 활동이 증가하면서, 손글씨를 직접 배우려는 사람들도 증가하고 있다.

(2) 문제 정의

그래서 우리는 소형 휴대용 기기에 손글씨를 그리면 같은 비율의 손글씨를 더 크게 플로터로 그리는 제품을 생각해보았다. 먼저, 기존의 제품들을 살펴보면, 대부분 자신이 직접 손글씨를 그리면 그 그림을 인코딩하여 파일로 만들고, 최종적으로 그림파일을 플로터에 입력시켜 그리는 방식이었다. 하지만 우리 팀은 그림파일로 만드는 번거로운 과정 대신에 자신이 그리는 글씨나 그림 등을 플로터가 실시간으로 따라 그리는 제품을 설계하여 접근성을 더 높이면 어떨까 생각하였다.

1.2 설계 목표

스마트폰이나 태블릿PC에 그림 혹은 글씨를 그리면, 연동된 PHPoC 보드를 통해 XY 플로터를 작동시켜 해당 그림이나 글씨를 손글씨의 질감을 내면서 실시간으로 따라 그려주는 디지털 프린터를 만드는 것을 목표로 설계를 진행한다.

2. 현실적 제한요소

2.1 안전성

XY 플로터의 기본 몸체 프레임은 금속제로 제작되어 어느 정도의 충격에도 견딜 수 있으며, 제품의 동작에 필요한 스텝모터와 서보모터를 보호할 수 있는 내구성을 가졌다.

2.2 심미성

어디에서든지 간편하게 WIFI를 통해 기기를 동작시킬 수 있으며, HTML 언어와 CSS 코드 등의 시각적인 요소를 활용하여 어디에, 무엇을, 어떻게 해야 하는지 알 수 있도록 버튼으로 간단하게 제작하여 시인성을 높였다. 또한, XY 플로터에 다양한 필기구(볼펜, 연필, 분필, 붓) 등을 바꾸어 장착할 수 있어, 다양한 질감을 표현할 수도 있게 제작하였다.

2.3 신속성

사람이 손 또는 펜 등으로 웹페이지에 그리는 그림과, 그림의 좌표 정보를 받아 XY 플로터가 그리는 그림이 실시간으로 상호작용하여 신속한 출력물을 얻기 위해 실시간 채팅, 게임, 인터넷 방송 등에서 주로 사용되는 '웹소켓 통신'을 사용하였다.

2.4 정확성

정확성을 보다 높이기 위하여 플로터와 스텝모터의 동작원리를 분석, PHP 언어로 구현하였다. 스텝모터를 사용할 때는 더 정확하고 미세한 조정을 위하여 스텝모터로 구현 가능한 최대 분주비인 200분주비(스텝모터의 360° 를 1.8° 씩의 단계)로 나누었다.

3. 합성

3.1 개념 설계

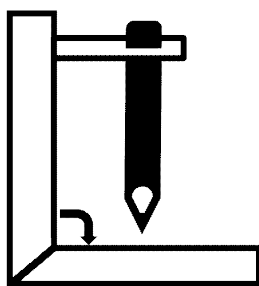
(1) 구현방식 설정

우리 팀은 목적에 맞는 제품을 구현하기 위해서 휴대용, 이동형, 플로터 이렇게 총 3가지로 구현 방식을 나누어 각각 방식의 장단점을 분석했다. 먼저, '휴대용'은 <그림 1>에서도 볼 수 있듯이 소형화하여 접이식으로 휴대가 가능한 방식을 생각했다. 이 방식은 펜이 고정되어 있고 캔버스가 움직여 그림을 그린다는 특징이 있다. 장점으로는 이름 그대로 휴대가 가능하여 어디서든 바로 꺼내서 그릴 수 있다는 점이다. 하지만 제품의 크기가 작아 다른 방식들 보다 상대적으로 작은 사이즈의 그림만 그릴 수 있다는 단점이 있다.

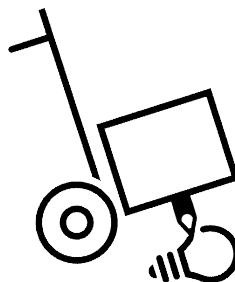
'이동형' 방식은 <그림 2>처럼 바퀴를 달아 몸체가 직접 움직이면서 입력된 그림을 그리는 방식으로 생각했다. 따라서 이 방식은 장소의 제약도 받지 않으며, 그릴 수 있는 그림의 사이즈가 가장 크다는 장점이 있다. 하지만 커버할 수 있는 범위가 넓다는 것은 그만큼 오차 범위 또한 넓고, 그림의 크기가 커질수록 오류가 많아진다는 단점으로도 볼 수 있다.

마지막으로 플로터 방식은 <그림 3>처럼 몸체와 캔버스가 고정되어 있고 레일을 통해 펜이 움직여 그림을 그리는 방식이다. 이 방식은 몸체가 고정되어 있으므로 다른 방식에 비해 튼튼하고 상대적으로 오류가 적다는 장점이 있다. 하지만 몸체가 커서 휴대용으로는 적합하지 않고 일정한 공간에서만 활용할 수 있다는 단점이 있다.

앞의 3가지 방식을 설계 목적에 맞게 나눠 정리해보면, 작동과정에서 오류의 기댓값은 플로터, 휴대용, 이동형 순으로 적고, 그릴 수 있는 그림의 크기는 이동형, 플로터, 휴대용 순으로 크다. 따라서 우리 팀은 최종적으로 부수적인 요소들을 포함해 오류가 상대적으로 적고, 웬만큼 큰 그림부터 작은 그림까지 범용적으로 활용이 가능한 XY 플로터 방식으로 설계하기로 결정했다.



<그림 1>



<그림 2>



<그림 3>

(2) 세부기능 정의

우리 팀이 구상한 제품의 작동 메커니즘을 간단히 설명하면, 먼저 스마트폰 또는 태블릿PC로 그림을 그릴 수 있는 캔버스를 포함한 웹사이트에 접속해 캔버스에 손가락 또는 펜으로 그

림을 그린다. 그 다음에 캔버스에 입력되는 좌표의 정보를 분석하여 모터로 전달하면 모터가 팬을 움직여 고정된 종이 위에 그림을 따라 그리도록 설계하기로 했다.

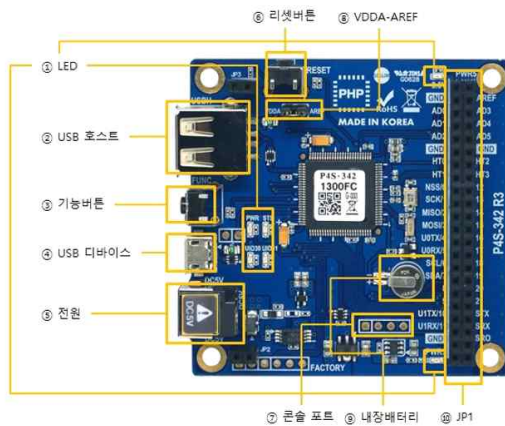
이와 같은 기능을 하기 위해서는 그림이 그려진 캔버스 좌표 정보를 스마트폰 또는 태블릿 PC에서 전달받은 뒤 해석하여 다시 모터에 전달할 컨트롤러와 실제로 팬을 움직일 모터가 필요하다. 정확한 좌표만큼 팬을 움직여야 하는 X, Y축에는 스텝각을 조정하여 정확한 위치를 나타낼 수 있는 스텝모터를 사용하고, X, Y축과는 다르게 팬을 종이에서 떼 상태를 유지하다가 그림의 좌표 정보를 받을 때만 팬을 붙이는 기능만을 필요로 하는 Z축에는 서보모터를 사용하기로 구상했다. 그리고 캔버스가 포함된 웹사이트도 HTML 언어와 자바 스크립트 등을 사용하여 직접 제작하기로 했다. 또한 앞에서 언급한 모든 작업은 실시간으로 동시에 동작하는 것처럼 보여야 하는데, 이를 위해서는 정보를 주고받을 때 소요되는 지연시간이 짧아야 한다. 따라서 우리 팀은 양방향에서 자유롭게 정보를 주고받는 데 특화되어 있는 ‘웹소켓 통신’을 사용하기로 했다.

3.2 상세 설계

(1) 각 부품 세부기능 및 사양

XY 플로터는 크게 PHPoC 보드, 스텝모터 제어보드, XY 플로터 몸체, 스텝/서보모터 이 4가지로 이루어져 있다. 이 중요한 부품들의 사양과 기능들에 대해 살펴보자.

I. PHPoC 보드(P4S-342)

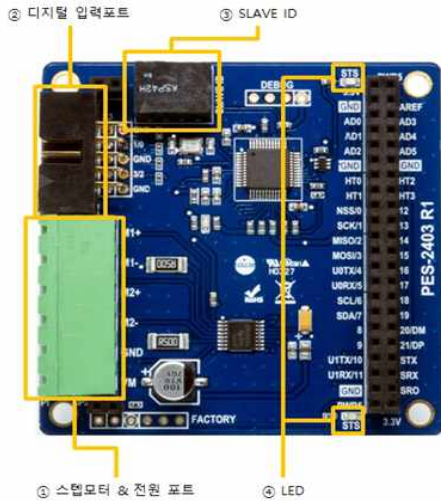


전원	입력전압 1	DC 5V (±0.5V)
	입력전압 2	DC 5V (±0.5V) - USB 디바이스 포트
	소비전류	평상시 - 약 85mA(※ USB 무선랜 어댑터 제외), 절전모드시 - 200uA 미만
	치수	66.5mm x 63.8mm x 13mm
	무게	약 27g (※ USB 무선랜 어댑터 제외)
인터페이스	UART	2 X UART포트(UART0 ~ 1), 통신속도: 1,200 bps ~ 230,400 bps
	네트워크	IEEE802.11b/g 무선랜 (Ralink RT3070/5370 chipset 무선랜 동글 필요)
	USB	USB 호스트 - USB 무선랜 어댑터 연결용 USB 디바이스 - PC 연결용
	디지털 입/출력	UIO: pin #0 ~ #21, #30(LED), #31(LED)
	아날로그 입력	ADC_CH0 ~ 5, AREF, 12-bit 분해능
	하드웨어 타이머	HT0 ~ 3, 토크/펄스/PWM출력모드, 캡처모드
	SPI	NSS, SCK, MISO, MOSI
	I2C	SCL, SDA
	SPC	STX, SRX, SRO
	내장배터리	3V(충전방식)
	무선랜 보안	WPA-PSK / WPA2-PSK, WPA-Enterprise (TLS/TLS/PEAP)
	온도	동작/저장온도 -20°C ~ 60°C
	환경	유럽 RoHS 규격 준수

< PHPoC 보드의 평면도 및 제품 사양표 >

PHPoC 보드는 ④USB 디바이스에 USB 케이블을 사용하여 컴퓨터에서 작성한 코드를 내려 받을 수 있다. PHPoC를 동작시키기 위해서 제품 사양표에 나와 있는 입력전압 DC 5V를 가해줘야 하므로, ⑤전원에 별도의 어댑터를 이용하여 5V의 전원을 인가해줄 것이다. ②USB 호스트에는 무선랜 동글을 사용하여 XY 플로터를 조작하기 위한 디바이스와 무선으로 연결할 것이다. 그리고 ⑩JP1에는 PHPoC를 통해 동작시키고 싶은 부품들의 결선도에 맞게 핀을 연결해주어야 한다.

II. 스텝모터 제어보드(PES-2405[S])



< 스텝모터 제어보드 평면도 >

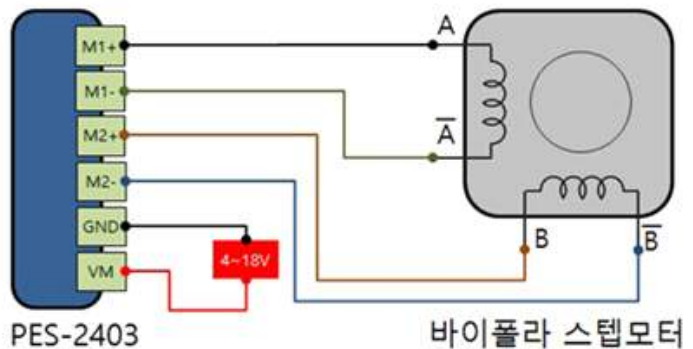
① 스텝모터 & 전원포트는 스텝모터를 사용할 때 사용하는 포트로, 스텝모터 결선도에 맞게 핀을 연결해주어야 한다.

② 디지털 입력포트에 스위치를 연결해주어 사용할 수 있다. ON/OFF를 하고 싶을 때 스위치를 연결해줄 수 있다.

③ SLAVE ID를 통해 사용하고 싶은 부품의 ID를 설정해주어 제어할 수 있다. 총 16개의 ID를 설정해줄 수 있다.

이 작품은 위에서 설명한 PHPoC 보드 1개와 스텝모터 2개를 동작시킬 스텝모터 제어보드 2개를 사용할 것이며, 이 2종류의 보드를 연결하여 1단은 PHPoC 보드, 2,3단은 스텝모터 제어보드로 총 3단으로 구성할 것이다.

III. 스텝모터 & 서보모터



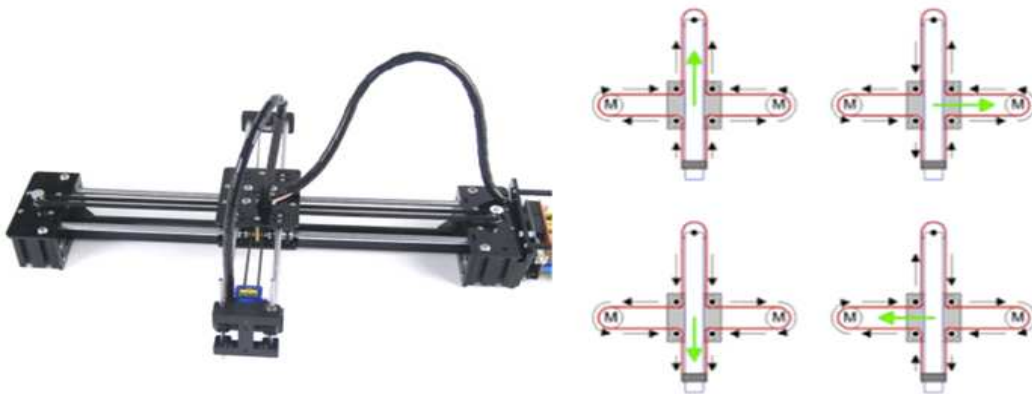
< 스텝모터 외형 및 결선도 >

앞서 말한 스텝모터를 사용하기 위해서는 스텝모터 제어보드를 사용해야 하며, 위의 결선도에 맞게 스텝모터 제어보드의 포트에 핀을 연결해주어야 작동시킬 수 있다. 또한, 스텝모터를 사용하기 위해서는 4~18V의 전원을 연결해줘야 하므로 9V 어댑터를 통해 전원을 인가해줄 것이다.

서보모터를 사용하기 위해서는 PHPoC 보드의 'JP1'부분의 결선도에 맞게 핀을 연결해줘야 한다. 서보모터는 PWM모드로만 사용하므로 PHPoC에 인가해준 5V만으로 서보모터를 동작시킬 수 있다. 따라서, 스텝모터를 동작시키기 위해 전원을 인가한 것과 다르게 추가 전원을 인가해줄 필요는 없다.

IV. XY 플로터 몸체

XY 플로터 몸체에는 2개의 스텝모터와 1개의 서보모터가 포함되어 있고, 서보모터에 펜을 장착한다. 2개의 스텝모터로 X와 Y축, 펜의 위치를 조작할 수 있고, 1개의 서보모터로는 Z축, 펜의 높이를 조작할 수 있다. 원하는 결과를 얻기 위해서는 X, Y, Z축을 제대로 조작할 수 있어야 하므로, 코드를 오류 없이 작성하여 스텝모터와 서보모터를 올바르게 동작시켜야 한다. 아래 XY 플로터 동작 방식에서 보는 것처럼, 스텝모터 2개를 정회전과 역회전 등 어떻게 조작하는지에 따라 상하좌우, 대각선 등으로 이동할 수 있으며, 이러한 일련의 과정들이 연속 되면 자연스러운 곡선도 그릴 수 있다.



< XY 플로터 외형 및 동작 방식 >

(2) 부품 구매 (예상 총 가격 : 약 37만 + α)

부품사진	부품명	수량	부품가격
	PHPoC 보드 (P4S-342)	1	약 5만원
	스텝모터 제어보드 (PES-2405[S])	2	약 10만원
	XY 플로터 몸체 (스텝/서보모터 포함)	1	약 20만원
	USB 케이블 (마이크로 5핀)	1	약 1만원
	9V 어댑터	2	약 1만원

4. 구현 및 제작

4.1 제작과정

(1) XY 플로터

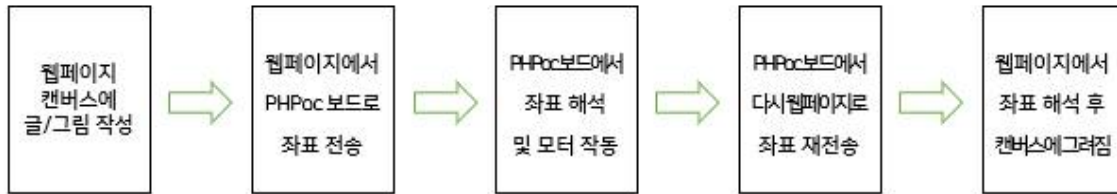


먼저, 디지털프린터의 출력 부분이라 할 수 있는 XY 플로터를 조립하였다. XY 플로터의 하측에 펜을 고정하는 부품과 서보모터를 결합하여 캔버스를 터치하면 서보모터가 동작해 종이에 펜을 맞당게 한다. 또한 좌, 우측에는 스텝 모터를 결합하고 XY 플로터의 사방을 하나의 레일로 연결한다. 2개의 스텝모터를 동작 펜이 부착된 부품을 움직여 종이에 그림을 그리게 된다. 그러면 캔버스와 종이에 똑같은 그림이 그려진다.



스텝모터와 연결되는 스텝모터 제어보드 2개의 @SLAVE ID를 왼쪽 사진과 같이 각각 1, 2로 설정해준 뒤, 스텝모터를 스텝모터 결선도와 같이 결선한다. 결합을 완료한 보드들은 XY 플로터의 좌측에 직접 부착하여 고정시키고 보드 케이스를 만들어 보드를 보호함과 동시에 심미성을 높였다.

4.2 웹페이지 코딩

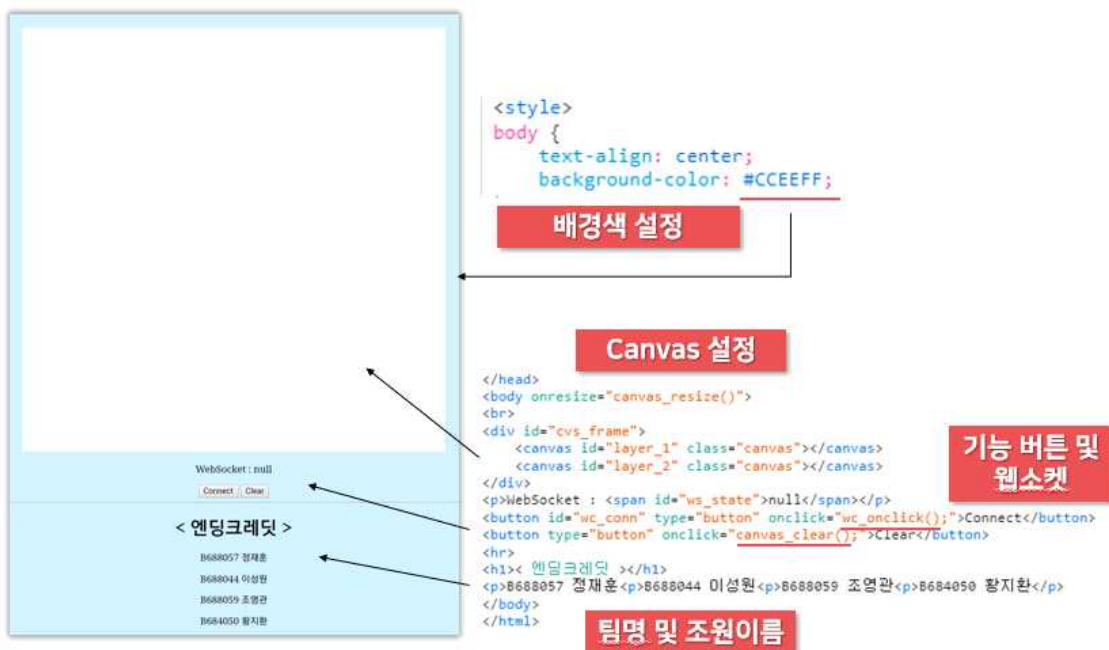


< 실제 동작 알고리즘 >

코드를 설명하기 전에 동작 알고리즘을 설명해 볼 것이다. 먼저 웹페이지에 있는 캔버스를 통해 글 혹은 그림을 작성해주게 되면, 웹페이지와 PHPoC 보드가 웹소켓으로 연결되어 PHPoC 보드로 좌표가 전송되게 되며, 보드에서는 좌표를 해석하고 스텝모터와 서보모터를 작동시킨다. 모터를 동작시키는데 사용된 좌표는 보드에서 다시 웹페이지로 전송되며, 최종적으로 아무것도 그려지지 않은 캔버스 위에 좌표에 따라 그려지게 된다. 이 과정은 빠른 속도로 실행되기 때문에 실시간으로 동작하는 것처럼 보인다.

1. 웹페이지 디자인

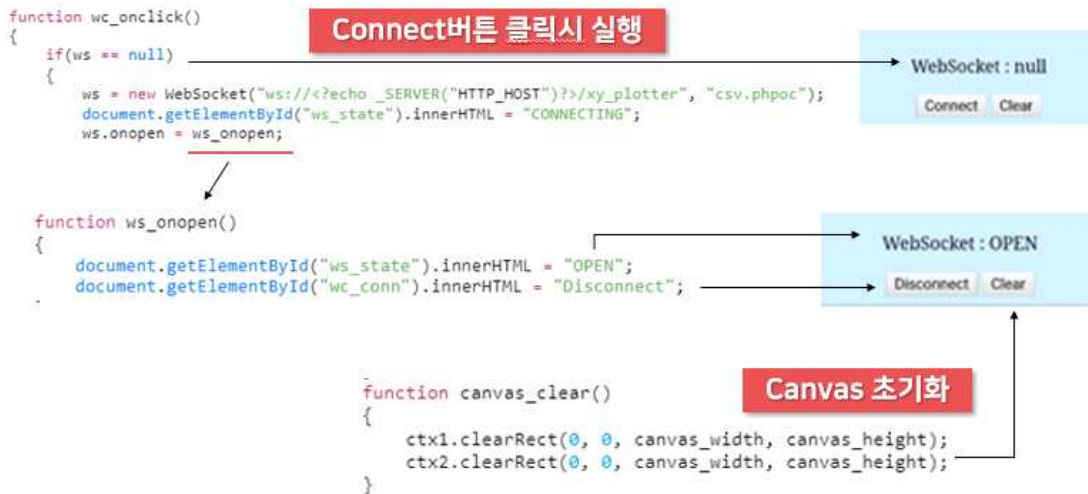
해당 웹페이지는 HTML 언어로 작성되었으며, 디자인 부분은 CSS를 사용하였다. 또한, 웹소켓 연결 및 보드로 좌표 전송 등은 모두 Javascript로 작성되었다.



< 웹페이지 디자인 및 캔버스 >

웹페이지에는 캔버스, 기능 버튼, 팀명 및 조원 이름을 작성해주었다. 바탕 배경색은 하늘색으로 설정해주었고, 웹소켓을 연결하기 위한 'Connect' 버튼과 만약 캔버스에 글 혹은 그림들이 작성되었다면 초기 상태인 흰 배경으로 만들어주기 위한 'Clear' 버튼을 작성해주었다. 기능 버튼들에 대해 더 상세히 알아보도록 하자.

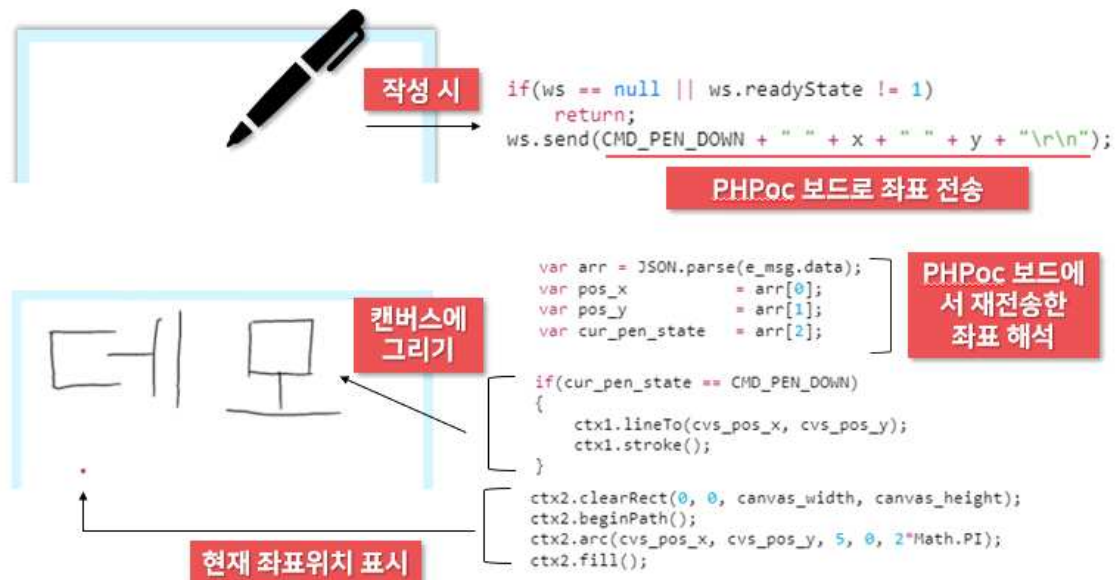
II. 기능 버튼



< 기능 버튼 동작 방식 >

기능 버튼은 2가지로, 'Connect' 버튼을 클릭하게 되면 아래 'wc_onclick()'이라는 함수가 실행된다. 이 함수는 웹 소켓 상태가 'null' 혹은 'CLOSE' 상태로 웹페이지와 보드가 웹소켓으로 연결되지 않았다면, 연결해주어 'OPEN' 상태로 만들어주는 함수이다. 위의 사진과 같이 'null' 상태였다면 웹 소켓이 'Connect' 버튼을 클릭함으로써 'OPEN' 상태가 된 것을 확인할 수 있다. 그리고 'Clear' 기능 버튼을 클릭하게 되면 'canvas_clear()'이라는 함수가 실행되며, 이 함수는 좌측상단 좌표인 (0,0)에서 좌측하단까지의 범위 내에 그려진 것을 지우는 함수이다.

III. 좌표 전송 및 캔버스에 그리기

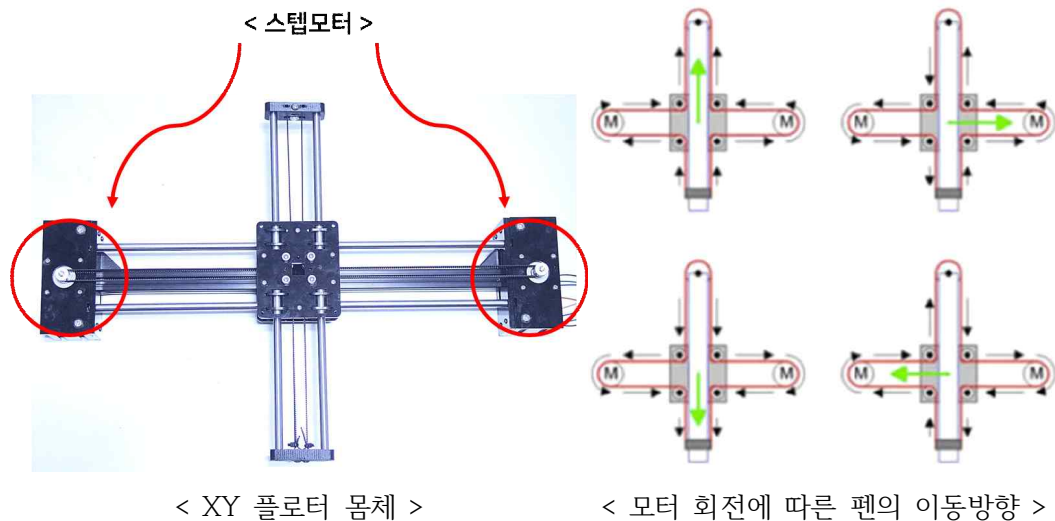


< 좌표 전송 및 캔버스에 그리기 >

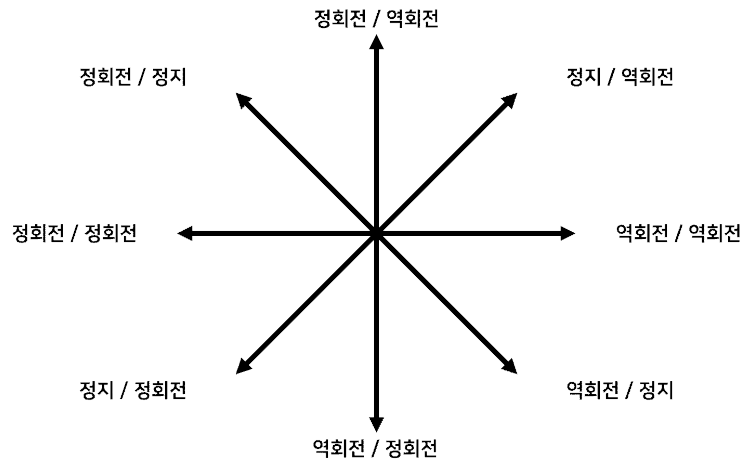
만약, 웹페이지로 작성한 캔버스에 손이나 마우스 등으로 그림을 그리게 되면, 캔버스에서 바로 그려지는 시스템이 아니다. 이 시스템은 캔버스에 그림을 그리게 되면, 계속 움직이는 좌표들을 실시간으로 PHPoC 보드로 전송하게 되며, 보드는 이 좌표들을 해석 및 계산하여 스텝모터를 움직인 좌표만큼의 회전량으로 회전하게 된다. 좌표뿐만 아니라 펜의 상태를 같이 전송하여, 서보모터를 동작시키게 되는데, 모터 동작의 자세한 부분은 뒤에서 다룰 것이다. 모터 동작시키는데 사용된 좌표 데이터를 보드에서는 다시 웹페이지로 전송하며, 웹페이지는 그 좌표를 통해 캔버스에 그려진다. 이 과정들은 매우 빠른 속도로 실시간으로 진행된다. 웹페이지와 보드 간 연결된 웹소켓의 특징으로 양방향 통신을 통해 서로 데이터를 빠른 속도로 주고 받을 수 있다. 위의 사진을 통해 보드로 보내는 펜 상태와 좌표 데이터 및 보드에서 보내온 데이터를 해석하고 캔버스에 그리게 되는 코드가 어떻게 작성되었는지 확인할 수 있다.

4.3 모터 제어 코딩

(1) 아이디어

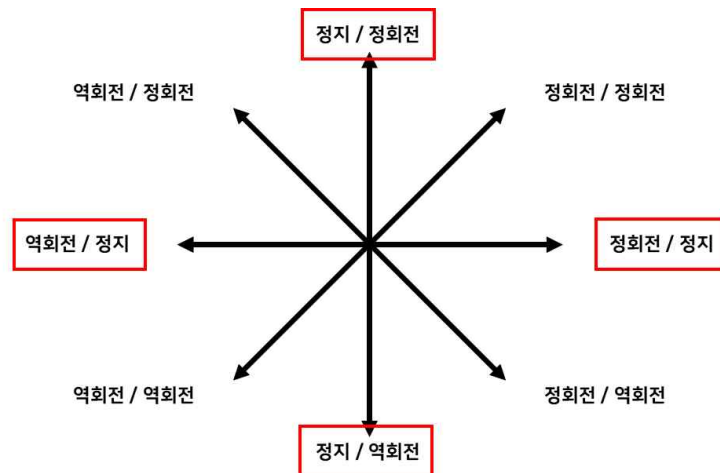


XY 플로터를 구현하기 위해서는 왼쪽의 그림처럼 스텝모터 2개가 사용된다. 오른쪽의 그림을 보면 펜이 있는 중심부가 움직이기 위해서는 다음과 같은 방식으로 움직여야 한다. 예를 들면, 오른쪽으로 가기 위해서는 왼쪽에 있는 모터(L_M)이 역회전, 오른쪽에 있는 모터(R_M)도 역회전, 간단하게 표현하면 (역회전, 역회전)의 입력이 있어야 한다. 또한, 오른쪽 위 대각선으로 가기 위해서는 L_M이 정지, R_M이 역회전 (정지, 역회전)의 입력이 있어야 하고, 위 방향으로 가기 위해서는 (정회전, 역회전)의 입력이 있어야 한다. 일반화를 시키면 대각선방향으로 움직일 경우 1개의 모터만으로 동작하고 상하좌우 방향은 모터 2개를 사용해야 한다. 도식화를 시켜보면 아래의 그림과 같이 표현할 수 있다.



< 모터 회전에 따른 펜의 이동방향을 도식화한 그림 >

그리고 이와 다르게, 일반적인 XY 좌표평면계에서의 X좌표와 Y좌표의 구현은 X축 모터 1개, Y축 모터 1개로 구현한다. 그리고 X축 모터의 회전 방향을 왼쪽, Y축 모터의 회전방향을 오른쪽 모터라고 하면, 아래의 그림처럼 표현할 수 있다.



< XY좌표평면에서 모터의 회전방향에 따른 펜의 이동방향 >

위에서 도식화한 두 개의 그림을 보면, 우리가 설계한 제품을 XY좌표평면으로 바꾸기 위해서는 시계방향으로 135° 회전하여야 한다는 것을 알 수 있다. 따라서 우리는 웹소켓 통신을 통해 전달받은 웹페이지의 X좌표와 Y좌표를 회전변환행렬 $\begin{pmatrix} \cos 135^\circ & -\sin 135^\circ \\ \sin 135^\circ & \cos 135^\circ \end{pmatrix}$ 을 통해 변환하여 원하는 방향으로 모터를 제어하였다.

(2) 구현 과정

```

57 function xy_goto($x, $y)
58 {
59     global $mode;
60
61     $r_x = (int)(-($x + $y)/sqrt(2));
62     $r_y = (int)(-($x - $y)/sqrt(2));
63
64     $x0 = (int)spc_request_dev(SID_X, "get pos") / $mode;
65     $y0 = (int)spc_request_dev(SID_Y, "get pos") / $mode;
66
67     $delta_x = $r_x - $x0;
68     $delta_y = $r_y - $y0;

```

← 웹소켓으로 전달받은 x 좌표, y좌표를 회전변환한 값

← "get pos" 명령어로 스텝모터의 카운터 값을 불러온다.

```

93
94     if($delta_x == 0)
95         spc_request_dev(SID_Y, "goto $r_y $speed_y $accel_y");
96     else if($delta_y == 0)
97         spc_request_dev(SID_X, "goto $r_x $speed_x $accel_x");
98     else
99     {
100         spc_request_dev(SID_X, "goto $r_x $speed_x $accel_x");
101         spc_request_dev(SID_Y, "goto $r_y $speed_y $accel_y");
102     }

```

카운터에서 불러온 값과 불러들인 값을 비교하여 '0'인 경우 모터의 정지를 판단, 대각선 방향의 제어를 명령한다.

< xy_goto 함수 부분 >

스텝 모터를 제어하는 데에는 내부함수인 'spc_request_dev(SID, "goto \$값")' 함수를 사용했다. 먼저, if절을 사용하여 대각선과 상하좌우를 구분하고, 'spc_request_dev(SID, "get pos")' 함수를 사용하여 스텝모터의 이전 카운터 값을 불러온 뒤, 웹소켓 통신으로 받아 변환한 값을 비교하여 0일 경우, 움직이지 않았음(모터의 정지)을 판단하고 아이디어대로 모터가 움직이도록 SID_X, SID_Y의 모터에게 명령한다.

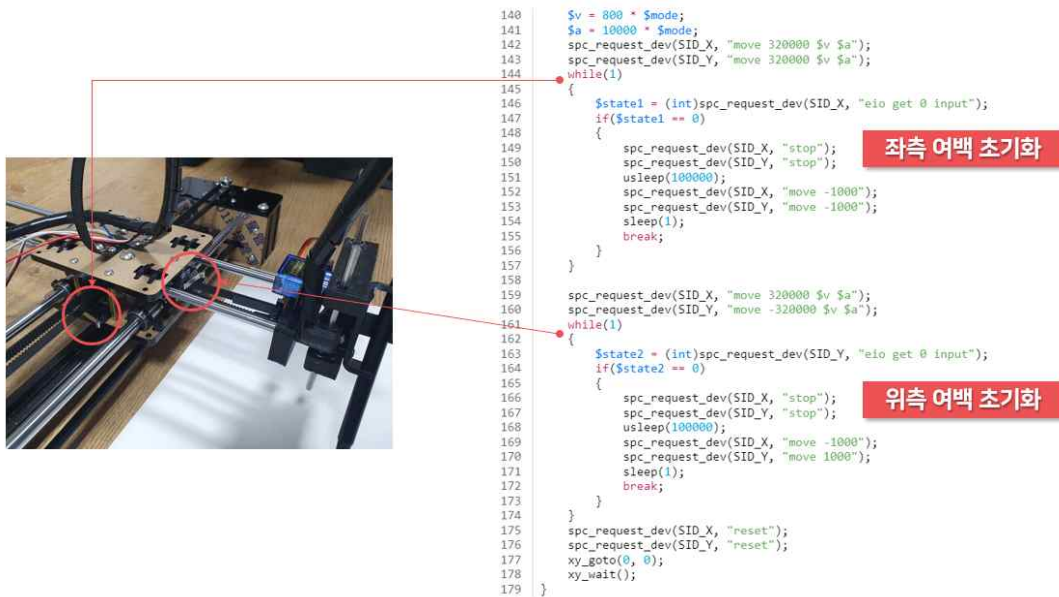
```

73     $speed_x = SPEED * abs($delta_x);
74     $speed_y = SPEED * abs($delta_y);
75
76     $r_x *= $mode;
77     $r_y *= $mode;
78     $speed_x *= $mode;
79     $speed_y *= $mode;
80
81     if($speed_x > SPEED_MAX * $mode)
82         $speed_x = SPEED_MAX * $mode;
83     if($speed_y > SPEED_MAX * $mode)
84         $speed_y = SPEED_MAX * $mode;
85
86     $accel_x = ACCEL_X * $speed_x;
87     $accel_y = ACCEL_Y * $speed_y;
88
89     if($accel_x > ACCEL_MAX * $mode)
90         $accel_x = ACCEL_MAX * $mode;
91     if($accel_y > ACCEL_MAX * $mode)
92         $accel_y = ACCEL_MAX * $mode;

```

< xy_goto 함수의 속도, 가속도 제어 부분 >

위의 코드는 스텝모터가 좀 더 정밀하게 움직일 수 있도록 속도(\$speed)와 가속도(\$accel)을 제한하는 과정이다. 제한된 속도와 가속도를 통해 spc_request_dev(SID, "goto \$값 \$speed \$accel")과 같이 입력하여 모터의 회전방향, 속도, 가속도를 명령하였다.



< 리미트 스위치를 이용한 초기화 >

추가적으로 리미트 스위치를 이용하여 PHPoC 보드의 'Reset' 버튼을 누를 시 좌측상단 좌표인 (0,0)으로 플로터를 초기화할 수 있도록 설정 해주었다.

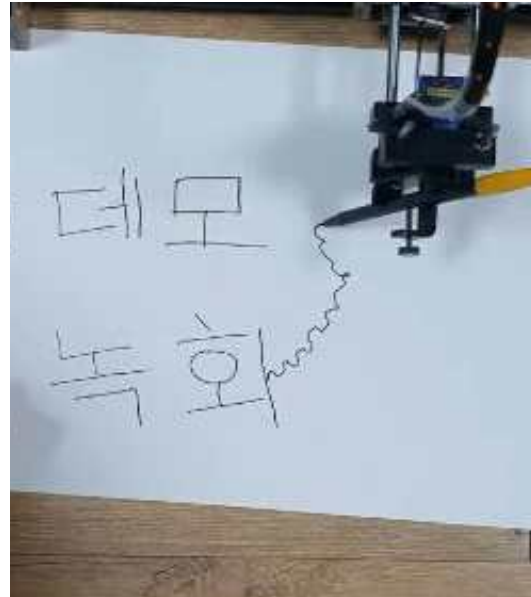
'Reset' 작동 시 플로터를 먼저 좌측으로 이동하도록 설정한 뒤 플로터의 몸체가 좌측 끝에 부딪히면 몸체에 부착해둔 리미트 스위치가 눌러 좌측 여백을 초기화된다. 그 다음 플로터가 위쪽으로 이동하게 되고 좌측 여백을 초기화 시킨 것과 같은 원리로 위측 여백을 초기화 시켜 'Reset' 작동 시 결과적으로 좌측상단 좌표 (0.0)으로 플로터가 초기화된다.

5. 결론

5.1 결과



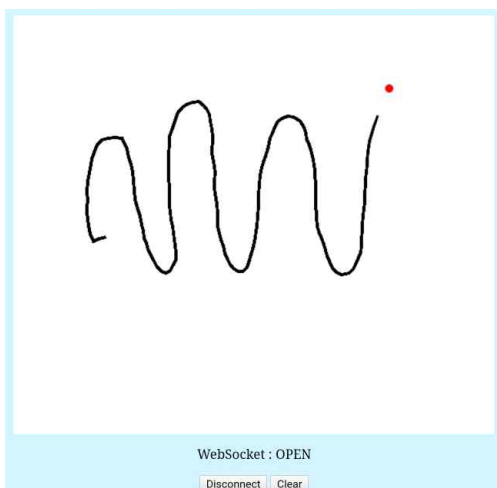
< 웹페이지 캔버스 >



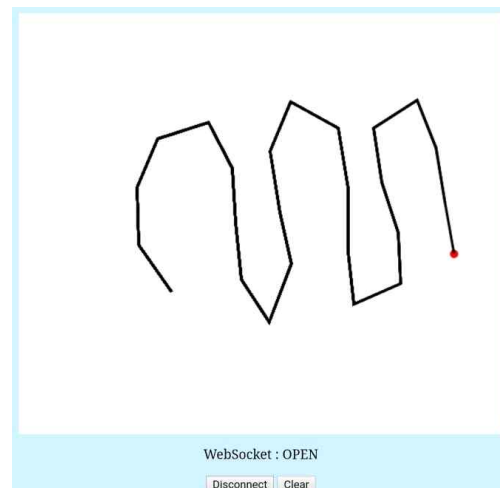
< 실제 플로터의 그림 >

위의 그림은 실제로 캔버스에 그림을 그려보고 웹페이지의 그림과 실제 플로터로 출력한 그림을 비교한 사진인데, 두 출력물을 비교했을 때 거의 차이가 없었다. 자세하게는, 글씨의 직선부분 뿐만 아니라 곡선 부분도 완벽하게 구현되었음을 확인하였고, '화'의 마지막 획에서 미세하게 변위를 주어 플로터를 동작시켰을 때에도 거의 오차가 없는 수준이었다. 또한, 왼쪽의 웹페이지가 소형 휴대기기라는 점을 참고하여 볼 때, 우리가 처음에 의도했던 소형 휴대기기에 그린 그림의 스케일을 키우는 목적도 달성하였다.

5.2 평가 및 향후개발 사항



< resolution : 20 >



< resolution : 200 >

우리는 웹페이지를 설계할 때, resolution 값을 20으로 조절하여 플로터의 해상도를 조절했는데, 결과적으로 큰 오차없이 만족스러운 결과를 보였다. resolution이 더 작으면 플로터의 동작 최소단위가 짧아져 과도한 CPU 성능이 요구되었고, 현재의 하드웨어 성능으로는 따라가기 벅차 동작속도의 딜레이와 오류가 다소 발생하였다. 그리고 반대로 resolution이 클 경우에는 위의 그림처럼 캔버스 상의 그림 중 곡선의 표현이 어려웠고 이에 따라 결과적으로 플로터가 그린 그림도 만족스럽지 못했다.

그리고 우리는 지금까지의 제품 시험 테스트를 통해 제품의 장점과 단점을 분석해보았는데, 일단 장점으로서는 앞서 여러 번 언급했듯이 휴대기기와 플로터가 웹소켓 통신을 통해 상호작용하여 실시간으로 결과물을 출력한다는 점이다. 따라서 사용자는 자신이 원하는 그림을 신속하고 정확하게 바로 추출할 수 있다. 하지만 이와 같은 방식에서는 필연적으로 단점이 존재하는데, 바로 수정이 불가능하다는 것이다. 그림이 실시간으로 그려지기 때문에 작은 부분이 잘못되어 수정하려고 해도 이미 결과물이 출력되었기 때문에 불가능할 수밖에 없다.

따라서 우리는 사용방식에 따른 문제점을 보완하기 위해 현재 제품처럼 웹페이지와 플로터가 상호작용하여 실시간으로 그리는 방식과 캔버스에 그린 그림 혹은 인터넷 상의 사진을 웹 서버에 업로드하면 플로터가 한 번에 그려주는 기존 방식을 버튼을 통해 설정하는 방법을 추가적으로 개발해 볼 계획이다.