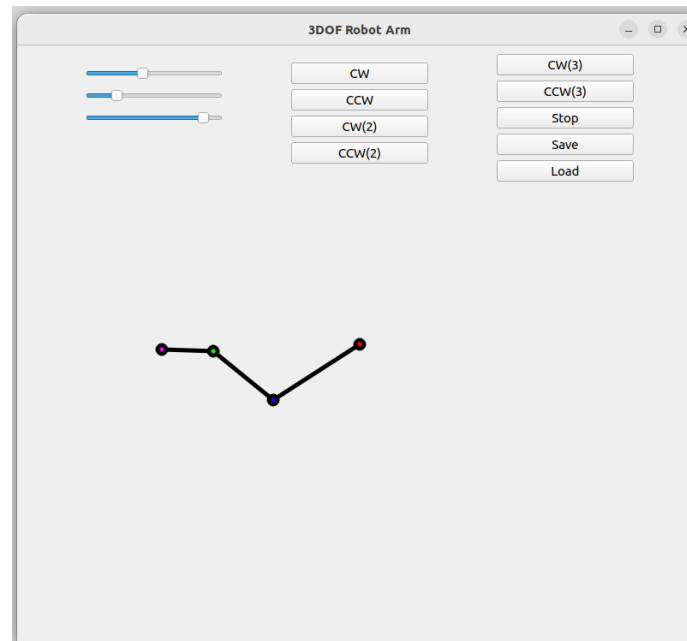


1. Ui 구성(실행 결과)



QT를 활용하여 다음과 같이 UI를 구성하였다. 우선 조건에 따라 3축 로봇 팔을 구성하였으며 중앙에 있는 빨간 점이 1축 파란 점이 2축 초록색 점이 3축이다. 각각의 관절에 대해 시계 방향과 반 시계 방향으로 회전하는 버튼이 있어야 한다는 조건이 있기에 1축부터 CW, CCW, 2축은 CW(2), CCW(2), 3축은 CW(3), CCW(3)으로 구성하였다. 그리고 각 관절의 수동 제어를 위해 제일 위쪽 슬라이드 바부터 1축, 2축, 3축을 각각 360도 제어 가능하도록 UI를 구성하였다. 팔 표시는 자료에 주어진 QPainter를 이용하여 시각적으로 나타내어 제어가 잘 되는지 직관적으로 판단할 수 있게 하였다. 마지막으로 나타낸 팔의 위치를 저장하고 불러올 수 있는 save/load 버튼을 추가하였다.

2. 코드 설명

우선 핵심적인 데이터를 저장하기 위해 각각의 축들의 각도와 회전 방향을 저장하기 위한 변수를 설정하고 부모 객체인 MainWindow가 파괴될 때, 자식(QTimer)가 자동으로 소멸되도록 포인터를 사용하여 메모리를 관리하였다.

슬라이드 바를 이용한 수동 제어의 경우 슬라이드의 값을 0~360으로 지정하여 해당 관절의 각도를 저장하고 이를 즉시 update()를 이용하여 UI에 표시되도록 하였다. 자동 회전 버튼의 경우 버튼이 눌린 것을 감지하면 관절 방향을 0/-1/1로 설정하여 타이머가 50ms 마다 updateAuto()

함수를 호출하도록 하였다.

로봇팔 표시의 경우 한 개의 축만 움직이는 경우 나머지는 움직이지 않아야 하지만 좌표상으로 나타나는 그래픽이기 때문에 현실과 별개로 선분과 선분이 계속하여 일정한 각도를 유지하도록 변환해 주어야 한다. 그렇기에 윈도우의 중앙을 로봇팔의 베이스로 잡고 링크의 길이를 정한 후 링크의 끝점의 좌표를 길이와 삼각함수의 연산을 이용하여 도출해서 3dof 로봇팔을 구현하였다. 또한 이렇게 표시되는 팔을 save하고 load 할 수 있도록 각각의 변수값들을 모두 저장하였다가 load 버튼이 눌리면 저장된 값들로 동기화 하도록 하였다.