**QTCreator 보고서**

20기 인턴 강은구

목차

1. HW1
   1. 주요 함수 및 명령어
      1. 사용 라이브러리(헤어 파일)
      2. Qt 주요 함수 및 명령어
   2. 코드 구조
      1. MainWindow 클래스
      2. intputAngle클래스
   3. 코드 설명
      1. intputAngle 클래스 세부 설명
      2. MainWindow 클래스 세부 설명
   4. 실행 결과
2. HW2

**HW1**

**-주요 함수 및 명려령어**

1. **라이브러리(헤더 파일)**
   1. **<QMainWindow>**  
      Qt에서 제공하는 기본 창 클래스  
      버튼, 레이블, 슬라이더 등 위젯을 포함하고, 이벤트를 처리하는 메인 윈도우 역할을 수행
   2. <**QTimer>**  
      일정한 간격으로 이벤트를 발생시키는 타이머 클래스  
      버튼을 누르고 있는 동안 관절이 계속 회전하도록 구현할 때 사용
   3. **<QPushButton>/<QLable>/<QSlider>/<QSpinBox>**  
      각각 버튼, 텍스트 출력용 레이블, 슬라이더, 숫자 입력용 SpinBox 위젯을 제어하기 위해 사용  
      UI에서 로봇팔 제어와 상태 표시 기능을 담당
   4. **<QPainter>/<QPen>**  
      paintEvent 안에서 그래픽을 그리기 위해 사용  
      QPainter는 선, 원, 도형을 그리는 클래스이며 뼤두은 선이 두께, 색상 등을 설정
   5. **<QFile>/<QTextStream>**  
      텍스트 파일을 열고, 쓰기/읽기를 할 때 사용  
      로봇 팡릐 각도를 저장하거나 불러오는 기능에 활용
   6. **<cmath>**  
      삼각함수(cos, sin)을 사용하기 위해 선언  
      관절 각도를 이용해 관절의 좌표를 계산
2. **Qt 주요 함수 및 명령어**
   1. **paintEvetgn(QPaintEvent \*event)**  
      QT에서 제공하는 이벤트 함수  
      창을 다시 그려야 할 때 자동으로 호출되며 QPainter를 사용한 그리기 작업을 해당 함수에 작성
   2. **connect()**  
      Qt의 Sinal-Slot 시스템을 연결하는 함수  
      버튼이 눌렸을 때 특정 함수를 실행하도록 연결하거나, SpinBox값이 바뀔 때 각도 변수를 업데이트 하도록 연결
   3. **update()**  
      화면 갱싱르 요청하는 함수  
      theta 값이 바뀌면 update()를 호출해서 paintEvent를 다시 실행
   4. **QTimer::start()/QtTimer::staop()**  
      타이머를 시작하거나 정지시키는 함수  
      버튼을 누르고 있는 동안 타이머가 동작해 주기적으로 관절이 회전
   5. **QFile::open()/QFile::close()**  
      파일을 열고 닫는 함수  
      상태 저장/불러오기 기능에서 사용
   6. **QTextStream입출력**  
      <<연산자로 파일에 쓰고, >>연산자로 파일에서 읽음  
      일반적인 C++의 cout, cin처럼 동작. 대상은 파일.

**-코드 구조**

**1.1 MainWindow 클래스**

1. 생성자 및 소멸자
   1. 생성자에서 UI초기화, 타이머 설정, 버튼-SLOT 연결 등을 수행
   2. 프로그램 시작 시 로봇팔의 초기 좌표를 설정 및 SpintBox와 변수 간의 연결을 진행
   3. 소멸자는 동적으로 할당된 리소스를 해제
2. paintEvent(QPaintEvent \*event)
   1. QPainter를 사용하여 로봇팔을 화면에 생성
   2. 각 관절의 각도(theta\_1\_3)을 기준으로 좌표를 삼각함수로 입력
   3. drawEllipse()와 drawLine()을 이용해 로봇팔 관절과 링크를 생성
   4. 각도가 변할 때마다 호출되어 시각적으로 반영
3. saveState()
   1. QFile과 QTextStream을 활용하여 현재 각 관절의 각도와 회전 방향을 텍스트 파일에 저장
   2. 저장된 형식은 [theta\_1, theta\_2, theta\_3, direction]형식
   3. 사용자가 언제든 버튼을 눌러 상태를 저장
4. loadState()
   1. 저장된 txt파일을 읽어 SpintBox와 내부 변수(theta)값에 반영
   2. 불러오기 시 paintEvent가 실행되어 저장된 각도로 이동
5. 버튼 제어 함수
   1. Joint1\_3을 선언하고 각각에 대해 cw/ccw버튼을 눌러 해당 각도의 증감을 실행
   2. QTimer::timeout와 연결하여 버튼을 누르고 잇는 동안 일정한 속도로 회전
   3. 버튼을 놓으면 타이머를 stop()

**1.2 mainWindow 클래스**

1. 생성자
   1. 특정 SpinBox와 로봇팔 각도 변수를 연결하기 위한 객체를 초기화
   2. MainWindow와 협력하여 UI입력을 내부 로직에 반영
2. setupConnection()
   1. SpinBox의 valueChanged(int) SIGNAL을 각도 변수에 SLOT으로 연결
   2. SpinBox값이 바뀌면 theta값을 자동으로 업데이트
   3. UI입력->내부 변수->paintEvent
3. getAngle()/setAngle()
   1. getAnlge()은 현재 관절 각도를 반환
   2. setAngle()은 txt파일에서 불러온 값과 초기값을 설정할 때 사용
   3. Degree<->Radian으로 변환

**-코드 설명**

텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

1. **inputAngle클래스**  
   :inputangle(), setupConnection
   1. inputAngle::inputAnle()  
      :해당 생성자는 QSpinBox와 로봇 팔 관절 각도 변수(theta)를 연결하는 역할을 합니다. 객체가 만들어질 때, 내부적으로 setupConnection()을 호출하여 spinBox값이 곧바로 theta에 반영되도록 준비합니다. 이를 통해 UI와 내부 계산 로직이 분리되면서도 동기화가 가능합니다. 결과적으로 사용자가 SpinBox에서 숫자를 바꾸면 자동으로 관절 각도가 업데이트 됩니다.
   2. intputAngle::setupConnection()  
      : SpinBox의 valueChanged(int) SIGNAL을 받아서 theta 변수에 적용하는 SLOT을 연결합니다. 입력 값은 degree단위이므로, 내부에서 라디안으로 변환하여 저장합니다.

텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

1. **MainWindow 클래스**  
   : MainWindow(QWidget \*parent = nullptr), ~MainWindow(), paintEvent(QPaintEvent \*event) override, swaveState(), loadState()

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷, 폰트, 메뉴이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* 1. **MainWindow::~MainWindow()**  
     : 생성자에서는 UI요소 초기화, SpinBox범위 설정, 버튼 및 슬라이더를 연결합니다. 각 관절 버튼에 pressed/released SIBNAL을 연결하여 버튼을 누르면 회전이 시작되고 떼면 멈추도록 하였습니다. QTimer를 설정해서 일정 주기마다 관절 각도를 갱신하였고 동시에 SpinBox값또한 업데이트 되도록 하였습니다. 슬라이더는 회전 속도를 조절하며 라벨을 통해 현재 속도를 사용자에게 보여줍니다 또한 파일 저장 및 불러오기 버튼과 각도 갱신 버튼을 연결합니다

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* 1. **Mainwindow::~MainWindow()**  
     : 동적으로 생성된 메모리를 해제하여 리소스 누수를 방지합니다.

텍스트, 스크린샷, 메뉴이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* 1. **MainWindow::paintEvent()**  
     : QPainter를 사용하여 로봇 팔의 각 관절과 링크를 시각적으로 표현합니다. theta값들을 삼각함수를 이용해 좌표(point\_1~4)를 계산하고, 원과 선을 이용해 로봇 팔을 표현합니다. 해당 함수는 SpinBox변경, 버튼 입력, 파일 불러오기 등으로 각도가 변할 때마다 자동으로 호출됩디다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* 1. **MainWindow::saveState()**: 현재 로봇팔 관절 각도를 txt파일에 저장합니다 내부 계산 단위에서 degree로 변환하여 저장합니다. 저장 성공 여부를 labelStatus\_2를 통해 표현하고 해당 load버튼을 통해 저장된 값을 불러올 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* 1. **MainWindow::loadState()**  
     :저장된 텍스트 파일(robot)arm)state.txt)에서 각도를 읽어 SpinBox와 theta변수에 반영합니다. 파일로부터 입력 받은 degree단위를 라디안으로 변환 후 각 값에 저장하며, 저장된 값을 토대로 로봇 팔의 각 관절이 회전합니다. 또한 labelStatus\_2를 통해 불러오기가 완료되었음을 표시합니다.

**-실행 결과**

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.