

로봇과 PC 프로세싱을 연동한 그림 그리기 교육 소프트웨어의 설계 및 제작

김재수*, 정보경*, 배정환*, 귀 우*, 박재운^o

^o경북대학교 IT대학 컴퓨터학부

e-mail: kjs@knu.ac.kr*, jjaewoon2@gmail.com^o

Design and Implementation of Drawing Education Software using Robot and PC Processing

Kim Jae-Soo*, Chung Bo-kyung*, Bae Jung-Han*, Gwu-wu*, Park Jae-Woon^o

^oDept. of Computer, Kyungpook National University

● 요약 ●

영유아들은 호기심이 많아 어떤 것을 조작하며 경험을 통해 더욱 더 발달하게 되는데, 아이들의 인지발달을 더욱 정교하고 정확하게 발달할 수 있도록 도움이 되는 놀이를 통해 유도해 주어야 한다. 4차 산업시대를 맞아 로봇을 이용함으로써 호기심을 불러 일으켜 새로운 경험을 할 수 있도록 돕고, 더 나아가 소프트웨어에 대한 호기심까지 이끌어 낼 수 있을 것이라 기대한다. 본 논문에서는 무선 통신을 통해 프로세싱과 ROS를 연동하여 프로세싱에서 그린 그림을 로봇이 돌아다니며 바닥에 그릴 수 있는 S/W, H/W를 구현하였다.

키워드: 프로세싱(Processing), ROS(Robot Operating System), Wireless Communication

I. 서론

영유아들은 호기심이 많아 어떤 것을 조작하며 경험을 통해 더욱 더 발달하게 된다. 아이들의 인지 발달을 더욱 정교하고 정확하게 성장시키기 위해서는 재미있는 놀이를 통해 관심을 유도해 주어야 한다. 아이들은 직관적으로 보이는 것을 좋아하고 로봇이라는 새로운 놀이 도구의 등장으로 재미있는 경험을 할 수 있다. 또한 4차 산업시대에 맞게 소프트웨어와 하드웨어를 결합한 놀이로써 소프트웨어를 체험하게 하고 호기심을 일으켜 소프트웨어에 대한 관심을 고양시킬 수 있다.

본 논문에서는 무선 통신을 통해 프로세싱과 ROS(Robot Operating System)를 연동하여 프로세싱에서 그린 그림을 로봇이 돌아다니며 바닥에 동일한 그릴 수 있는 하드웨어와 소프트웨어를 구현하고 제작하였다. ROS와 프로세싱을 연동한 교육용 소프트웨어의 설계 및 제작에 필요한 하드웨어는 로봇과 이두이노, 프로세싱 PC가 있으며, 이들 하드웨어 사이의 통신은 RF(Radio Frequency) 통신과 Wi-Fi 통신을 사용하였다.

PC 프로세싱(Processing)이란 컴퓨터 프로그래밍의 본질을 시각적 개념으로 프로그래머가 아닌 사람들에게 교육할 목적으로 뉴미디어 아트, 시각 디자인 공동체를 위해 개발된 오픈소스 프로그래밍 언어이자 통합 개발 환경(IDE)이다.

ROS는 로봇 응용 프로그램을 개발할 때 필요한 하드웨어 추상화, 하위 디바이스 제어, 일반적으로 사용되는 기능의 구현, 프로세스간의 메시지 패싱, 패키지 관리, 개발환경에 필요한 라이브러리와 다양한

개발 및 디버깅 도구를 제공한다.

프로세싱으로 그림을 그려 통신 매개를 통해 필요한 데이터를 전송하면 ROS는 데이터 파싱을 통해 로봇을 움직이고, 그림을 그리는 구조로 동작한다.

II. 설계 및 구현

1. 설계 및 운영 환경

본 논문에서 사용된 하드웨어로는 공식적으로 지원되는 ROBOTIS의 TURTLEBOT 3, Arduino Uno가 있고, RF통신을 사용할 때는 이두이노를 활용해 프로세싱과 ROS를 연동하였다. ROS는 Ubuntu 16.04LTS, Ubuntu MATE(Raspberry PI) 위에서 동작하고 Wi-Fi 통신을 지원한다.

2. 시스템 구현

2.1 시스템 구성도

본 논문에서 구현한 시스템의 하드웨어 사이에 일어나는 통신 환경은 Fig. 1.과 같다. 로봇과 PC 프로세싱 사이는 Wi-Fi 통신이 이루어지며, 로봇과 이두이노 사이의 통신은 RF 통신이 이루어진다.

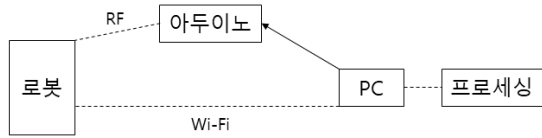


Fig. 1. The Communications between System Entities

2.2 로봇의 회전 각 계산과 이동

본 논문에서 사용한 로봇의 회전각을 계산하는 모형은 Fig. 2.과 같다. 로봇과 PC 프로세싱 사이는 로봇이 특정 $V_1(x_1, y_1)$ 으로 이동할 때 V_1 으로 가기위해 로봇이 바라보아야 하는 각도는 $\text{atan}(y_1/x_1)$ 이다. 이를 바탕으로 V_1 에서 다음 좌표 $V_2(x_2, y_2)$ 로 가기위해 로봇이 바라보아야 하는 각도는 $\text{atan}(y_2-y_1/x_2-x_1)$ 이다. 여기서 주의해야 할 점은, 원점에서 V_1 으로 갈 때는 로봇이 보는 각도가 0 (조건)이지만, V_1 으로 이동할 때 각도가 바뀌므로 V_2 로 이동할 때는 V_1 에서의 각을 고려해줘야 한다. Fig. 2.에서 로봇이 θ_1 을 바라보기 위해 총 회전해야 하는 각은 $\text{atan}(y_2-y_1/x_2-x_1) - \theta_1$ 이다. PC 프로세싱으로부터 로봇이 데이터를 받아 회전각을 계산하여 이동하는 과정에서 로봇이 데이터를 처리하는 과정은 Fig. 3.과 같다. 로봇이 데이터를 수신하게 되면 데이터 분석을 통하여 붓 관절을 동작시켜 로봇이 회전하면서 그림을 그리게 한다. 그림을 그리는 도중에 장애물은 피하면서 그림을 그린다.

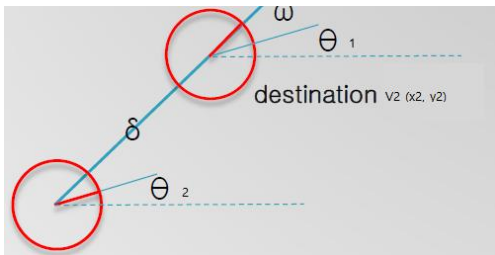


Fig. 2. The Direction Angle of a Robot

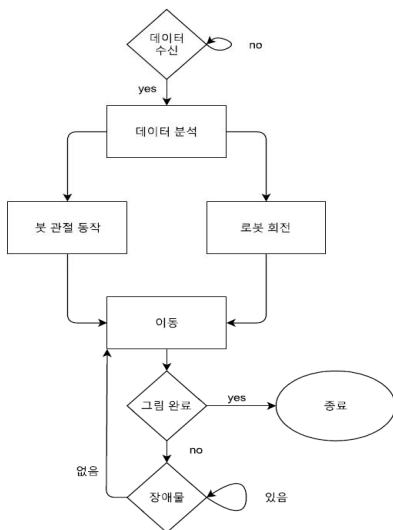


Fig. 3. The Diagram of Robot's Data Flow

2.3 User Interface 구현

그림판의 상위 Interface는 Fig. 4.와 같으며, 상위 Interface에서 화면의 임의의 지점을 클릭하면 하위 Interface로 전환이 된다.



Fig. 4. Upper(main) Interface

하위 Interface는 Fig. 5.과 같다. 하단의 메뉴바에서의 버튼은 좌측부터 A4 배율, A2 배율, 한 단계 롤백, 전체 삭제, 다음 단계 진행을 의미한다. 하위 Interface인 그림판에 그림을 그리고 가장 우측 버튼인 다음 단계 진행을 누르면 Fig. 5.과 같은 결과를 가진다.

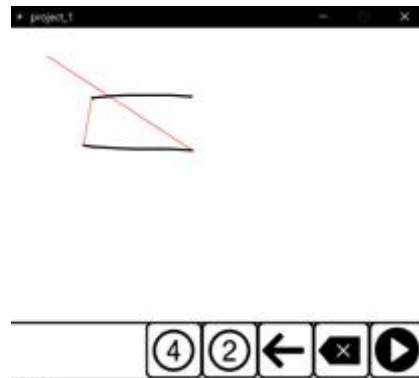


Fig. 6. Drawing Process in Lower Interface

2.3 Robot(Turtlebot3)

로봇이 그림을 그릴 수 있도록 Fig. 8.과 같이 Servo모터에 가이드를 제작하여 회전 운동을 수직 운동으로 바꿔주었다.



Fig. 8. Servo Motor Guide



Fig. 9. Drawing Outcome

프로세싱 그림판에서 Fig. 7.과 같이 그림을 그리면, 로봇이 그림을 그렸을 때의 결과는 Fig. 9.와 같다. Fig. 9.에서 좌측의 그림은 A4 배율, 우측의 그림은 A2 배율을 나타낸다.

III. 결론

본 논문에서는 사용자가 PC상에서 그림을 그리면 실제로 바닥에 그림을 그려주는 로봇을 구현하였다. 동작 테스트 결과 유사하게 그림을 그려주는 것을 확인할 수 있었다. 회전할 때의 회전 오차와 펜의 딜레이를 세밀하게 보정한다면 거의 똑같은 그림을 그릴 수 있다고 생각한다. 차후에 좀 더 크고 정확한 기계를 이용한다면 산업로의 확대 가능성도 고려할 수 있고, 그림을 그리는 장치도 PC에서 모바일까지 확장할 수 있을 것으로 기대한다.

REFERENCES

- [1] Turtlebot 3 manual, <http://emanual.robotis.com/>
- [2] ROS Documentation, <http://wiki.ros.org/>