|  |
| --- |
| **2019년 1학기 모바일로봇프로그래밍**  **팀 프로젝트 보고서** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 주제 | 오픈소스 기반의 meta OS인 ROS와 터틀봇, rpi카메라를 이용한  패턴인식 추종 프로그램 구현 | | | |
| 목표 | 강의에서 배운 내용을 이용해 직접 프로그램을 만들어보며 ROS 숙달 | | | |
| 주요내용 | 터틀봇에 설치된 rpi 카메라를 이용한 패턴 인식 (네모 모양)  패턴의 위치에 따라 twist값을 결정하는 추종 알고리즘  장애물 충돌예상 시 후진 | | | |
| 기대효과 | 노드 프로그램을 직접 만들어보며 ROS 프로그래밍 기법 이해  직접 터틀봇을 다뤄보며 로봇의 원리와 동작 과정 이해 | | | |
| 팀원  소개 | 소속(학과) | 학번 | 이름 | 담당업무 |
| 컴퓨터학부 | 2013105104 | 허영행 | 패턴 인식 및 거리계산  프로그램 구현 |
| 컴퓨터학부 | 2015114743 | 황은지 | Opencv 화면 출력 및  메인 함수 구현 |
| 컴퓨터학부 | 2013105107 | 홍혁진 | 거리 값을 이용한  추종 프로그램 구현 |

**(1) 과제 목표 및 내용**

* 과제 목표
* 패턴 인식을 이용한 터틀봇 추종 프로그램
* 과제 내용
* Linux OS 환경에서의 C++언어를 이용한 ROS 프로그램
* 추종 프로그램 , 장애물 인식 시 후진 프로그램 구현
* Rpi camera를 통해 실시간으로 전송되는 이미지에서 특정 패턴 인식
* 일정 크기 이상의 패턴만 세로중심선 출력
* 이미지의 세로중심선과 인식된 패턴의 세로중심선 사이의 거리 값을 측정
* 측정한 값에 따라 정해진 twist 값을 cmd\_vel로 publish
* Publish된 twist 값을 통한 터틀봇의 추종

**(2) 추진 방법**

* 상세기능 분석 및 설계
* 효율적인 프로젝트 관리를 하기 위한 Git 사용
* 짧은 개발 주기에 신속히 대처하기 위해 단계별 목표 수립
* 프로그램 개발
* 단계별 목표에 따른 순차적 개발
* 개발 과정에서 수시로 오류 정정 및 검증
* 개발 완료 후 테스트를 통한 프로그램의 완성도 향상
* 과제개발 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **개발 내용** | **추 진 일 정** | | | | | | | | | | | | | | |
| **4월**  **4주차** | | **5월**  **1주차** | | **5월**  **2주차** | | **5월**  **3주차** | | **5월**  **4주차** | | **6월**  **1주차** | | **6월**  **2주차** | **6월**  **3주차** | |
| 1 | 주제 선정 및 개발환경 구축 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | GitHub 숙달 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 상세 기능 분석 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 프로그램 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | 테스트 및 문제점 수정 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 문서화 작업 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**(3) 프로젝트 수행 결과**

* 구현 내용
* 패턴 인식 프로그램 ( turtle\_pattern\_detect.cpp )
* Rpi 카메라를 통해 들어오는 이미지를 subscribe해서 특정 패턴을 인식
* 패턴의 크기를 측정하여 일정 크기 이상의 패턴 선택
* 선택한 패턴의 중심점을 기준으로 세로선 출력
* 전체 이미지의 중앙에 세로선 출력
* 패턴 중심선과 전체 중심선사이의 거리를 추종 알고리즘 계산 노드에 publish
* 추종 알고리즘 계산 프로그램 ( turtle\_follow.cpp )
* 패턴 인식 프로그램에서 보내는 거리 계산 값이 담긴 msg를 subscribe
* 받은 값의 범위에 따라 twist 값을 여러 개의 조건문으로 미리 정의 해 놓음
* 터틀봇을 움직이게 하기위한 twist 값 publish
* 장애물 회피 프로그램 ( turtle\_scan\_sub.cpp)
* /scan topic을 이용하여 레이더 스캔 값 subscribe
* Turtle\_scan\_sub에서 -15도 ~ 15도 사이의 평균 장애물 거리 계산.
* cmd\_vel을 이용해서 속도 값 (음수) 전송.