

학습 체크리스트

파이썬

1. 개발 환경 설정

- 파이썬을 설치하고 버전을 확인할 수 있다.
- 파이썬의 설치 위치를 확인하고 버전이 여러 개일 때 제어할 수 있다.
- Visual Studio Code를 활용하여 파이썬 개발 환경을 설정할 수 있다.

2. 기본 문법과 자료형

- 키워드, 식별자, 변수, 자료형(숫자, 문자열, 불리언 등)을 이해하고 활용할 수 있다.
- 연산자(산술, 비교, 논리, 복합 대입 등)를 이해하고 활용할 수 있다.
- 문자열의 인덱싱, 슬라이싱, 길이 측정을 수행할 수 있다.
- 리스트와 딕셔너리를 생성하고 주요 메서드를 활용할 수 있다.
- C++과의 차이점을 명확히 이해할 수 있다.

3. 입출력과 문자열 처리

- print 함수의 sep 및 end 매개변수를 활용하여 출력 형식을 조정할 수 있다.
- input 함수를 사용하여 사용자 입력을 받을 수 있다.
- 문자열 포매팅(str.format, f-string)을 활용하여 출력 형식을 지정할 수 있다.

4. 제어문과 반복문

- 불리언 자료형과 조건문(if, elif, else)을 활용하여 논리 흐름을 제어할 수 있다.
- for 및 while 반복문을 사용하여 반복 작업을 수행할 수 있다.
- 리스트 컴프리헨션과 enumerate를 활용하여 반복 작업을 효율적으로 처리할 수 있다.

5. 함수와 모듈

- 함수를 자유롭게 정의하고 다양한 인자 전달 방식을 활용할 수 있다.
- 람다 함수와 재귀 함수를 작성하고 활용할 수 있다.
- 모듈과 패키지의 개념을 이해하고 import를 활용할 수 있다.

6. 클래스와 객체 지향 프로그래밍

- 클래스를 정의하고 객체를 생성하여 활용할 수 있다.
- 클래스 변수와 인스턴스 변수를 이해하고 활용할 수 있다.
- 상속과 다중 상속을 이해하고 메서드 오버라이딩을 활용할 수 있다.
- 프라이빗 변수와 프로퍼티 데코레이터를 활용하여 데이터 접근을 제어할 수 있다.

7. 예외 처리와 파일 입출력

- 예외 처리(try, except, else, finally)를 구현하고 사용자 정의 예외를 작성할 수 있다.
- 파일을 열고 읽고 쓰는 작업을 수행할 수 있다.

8. 표준 라이브러리 활용

- datetime 모듈을 사용하여 날짜와 시간을 처리할 수 있다.
- random 모듈을 활용하여 난수 생성 및 관련 작업을 수행할 수 있다.

- `__init__.py` 파일, 모듈, 패키지 사이의 관계를 알 수 있다.
- 제너레이터와 데코레이터를 활용하여 효율적인 코드를 작성할 수 있다.

9. 외부 라이브러리 활용

- 외부 라이브러리를 검색하고 설치할 수 있다. (pip)
- 외부 라이브러리의 설치 위치와 모듈을 자유롭게 import할 수 있다.
- 라이브러리를 추가할 수 있다.

OpenCV

1. OpenCV 설치 및 기본 사용법

- OpenCV를 설치하고 버전을 확인할 수 있다.
- `imread`, `imshow`, `waitKey`, `destroyAllWindows` 함수를 사용하여 이미지를 읽고 표시하며 창을 관리할 수 있다.
- `VideoCapture`와 `VideoWriter` 클래스를 활용하여 비디오를 읽고 저장할 수 있다.
- 파이썬과 C++로 설치하고 사용할 수 있다.

2. 이미지 및 비디오 처리

- `line`, `rectangle`, `circle`, `ellipse`, `putText` 함수를 사용하여 이미지에 도형과 텍스트를 그릴 수 있다.
- `freetype` 모듈을 통해 한글 폰트를 사용하여 텍스트를 표시할 수 있다.
- 이미지의 밝기와 대비를 조절하고 `saturate_cast`를 활용하여 픽셀 값을 안전하게 변환할 수 있다.
- `waitKeyEx` 함수와 마우스 콜백 함수를 사용하여 키보드 및 마우스 이벤트를 처리할 수 있다.

3. 이미지 분석 및 변환

- 히스토그램을 계산하고, 히스토그램 평활화와 스트레칭을 통해 이미지의 명암비를 조절할 수 있다.
- 비트 연산을 수행하여 이미지 합성 및 마스킹을 할 수 있다.
- `blur`, `GaussianBlur`, `medianBlur` 함수를 사용하여 다양한 블러링 효과를 적용할 수 있다.
- `warpAffine` 함수를 사용하여 이미지의 기하학적 변환을 수행할 수 있다.
- 미분 필터와 Canny 엣지 검출기를 활용하여 이미지의 경계를 추출할 수 있다.
- 허프 변환을 통해 이미지에서 선을 검출할 수 있다.
- 색상 공간 변환과 `inRange` 함수를 사용하여 특정 색상을 추출할 수 있다.
- 이진화(`threshold`, `adaptiveThreshold`) 및 모폴로지 연산을 수행하여 이미지의 형태를 분석할 수 있다.

4. 객체 검출 및 인식

- 템플릿 매칭을 통해 이미지 내에서 특정 패턴을 찾을 수 있다.
- 캐스케이드 분류기를 사용하여 얼굴 등 객체를 검출할 수 있다.
- HOG(Histogram of Oriented Gradients) 알고리즘을 이해하고 활용하여 객체를 인식할 수 있다.
- QR 코드 및 ArUco 마커를 검출하고 해석할 수 있다.

5. 머신러닝 및 딥러닝 기초

- OpenCV를 활용하여 머신러닝 및 딥러닝의 기초 개념을 이해하고 간단한 모델을 적용할 수 있다.

ROS 2 미들웨어 및 인터페이스

1. ROS 2의 개념 이해 및 설치

- ROS 2의 개념을 이해하고 설치할 수 있다. (topic, node, service, action)
- CLI 명령어(ros2 run, ros2 launch, ros2 topic, ros2 node, ros2 param, ros2 service, ros2 action)를 사용할 수 있다.
- rqt 도구를 활용하여 ROS 2 시스템을 모니터링하고 디버깅할 수 있다.
- 자유롭게 패키지를 생성하고 노드를 작성할 수 있다.
- 퍼블리셔와 서브스크라이버 노드를 작성하여 통신을 구현할 수 있다.
- QoS 설정을 이해하고 적용할 수 있다.
- 시간 인터페이스(Header)를 활용할 수 있다.
- 서비스 서버와 클라이언트를 작성하여 요청-응답 구조를 구현할 수 있다.
- 액션 인터페이스를 이해하고 액션 서버와 클라이언트를 작성할 수 있다.
- 런치 파일을 작성하고 패키지에 포함할 수 있다.
- 런치 파일에 여러 개의 노드를 포함시킬 수 있다.
- 런치 파일에 런치 콘피그를 통해 CLI에서 아규먼트를 변경시킬 수 있다.
- 파라미터를 노드에서 선언하고 쓸 수 있다.
- 파라미터를 외부에서 변경하도록 코드를 작성할 수 있다.
- 파라미터 파일을 생성하고 런치 파일을 작성하여 노드 실행을 관리할 수 있다.
- 런치 파일에 다른 런치 파일을 포함시킬 수 있다.
- 네임스페이스를 활용하여 노드와 토픽을 구조화할 수 있다.
- 로깅을 설정하고 사용자 정의 CLI 명령어를 작성할 수 있다.
- ROS 2의 미들웨어 계층과 DDS(Data Distribution Service)의 역할을 이해할 수 있다.
- 미들웨어 구현의 구조와 rmw, rcl, rosidl API의 기능을 이해할 수 있다.
- 미들웨어 패키지의 구성 요소와 그 역할을 설명할 수 있다.
- 미들웨어 구현 시 고려해야 할 사항과 선택 기준을 이해할 수 있다.
- 메시지(msg), 서비스(srv), 액션(action) 인터페이스의 개념과 역할을 이해할 수 있다.
- 기본 자료형과 복합 자료구조를 사용하여 커스텀 인터페이스를 정의할 수 있다.
- 인터페이스 정의 언어(IDL)를 활용하여 메시지를 정의하고 직렬화할 수 있다.
- 기존 인터페이스를 조회하고 활용할 수 있다.
- 커스텀 인터페이스를 작성하고 패키지에 포함시킬 수 있다.
- 인터페이스 변경 시의 빌드 및 배포 과정을 이해하고 수행할 수 있다.

터틀봇3 제어 및 시뮬레이션

1. 터틀봇3 하드웨어 및 소프트웨어 구조 이해

- 터틀봇3의 하드웨어 및 소프트웨어 구조를 이해하고 설명할 수 있다.
- 터틀봇3 패키지를 설치하고 환경을 설정할 수 있다.
- 터틀봇3를 키보드 또는 조이스틱으로 원격 제어할 수 있다.
- 터틀봇3의 센서 데이터를 수집하고 처리할 수 있다.
- Gazebo 시뮬레이터를 설치하고 터틀봇3 시뮬레이션 환경을 구성할 수 있다.
- Gazebo에서 터틀봇3를 시뮬레이션하고 제어할 수 있다.
- 터틀봇3에 OpenManipulator를 장착하고 제어할 수 있다.
- OpenManipulator의 동작을 프로그래밍하고 시뮬레이션할 수 있다.
- 터틀봇3와 OpenManipulator를 연동하여 복합 작업을 수행할 수 있다.
- SLAM(Simultaneous Localization and Mapping)을 구현하여 터틀봇3의 자율 주행을 실현할 수 있다.
- Navigation2 패키지를 활용하여 경로 계획 및 추적을 수행할 수 있다.
- Navigation2의 follow_waypoints 액션을 제어할 수 있다.

- 터틀봇3의 URDF(Unified Robot Description Format)를 수정하여 로봇 모델을 커스터마이징할 수 있다.
- RViz를 활용하여 터틀봇3의 상태와 센서 데이터를 시각화할 수 있다.
- 터틀봇3의 카메라 데이터를 처리하고 이미지 기반 응용 프로그램을 개발할 수 있다.
- 터틀봇3의 네트워크 설정을 구성하여 원격에서 제어하고 모니터링할 수 있다.

장치 통합: GPIO, USB, micro-ROS, Arduino Uno, Raspberry Pi Pico, 레벨 시프터, LED, 서보모터, DC 모터, LCD

1. GPIO 제어

- ROS 2에서 GPIO(General Purpose Input/Output) 핀을 제어하여 센서 및 액추에이터와 인터페이스할 수 있다.

2. USB 통신

- USB를 통해 외부 장치를 연결하고 ROS 2 노드와 통신할 수 있다.

3. micro-ROS 활용

- micro-ROS를 사용하여 마이크로컨트롤러를 ROS 2 네트워크에 통합할 수 있다.
- micro-ROS 지원 하드웨어 설정: micro-ROS를 지원하는 하드웨어 플랫폼을 선택하고 설정할 수 있다.
- 라즈베리파이-피코 보드에 micro-ros 펌웨어를 올릴 수 있다.
- micro-ROS 에이전트 구성: micro-ROS 에이전트를 설치하고 구성하여 마이크로컨트롤러와 ROS 2 간의 통신을 설정할 수 있다.
- 마이크로컨트롤러 노드 작성: 마이크로컨트롤러에서 퍼블리셔와 서브스크라이버 노드를 작성하여 ROS 2 네트워크와 데이터 교환을 할 수 있다.
- 센서 데이터 수집 및 전송: micro-ROS를 활용하여 센서 데이터를 수집하고 ROS 2 네트워크로 전송할 수 있다.
- 액추에이터 제어: micro-ROS를 통해 ROS 2 네트워크에서 마이크로컨트롤러의 액추에이터를 제어할 수 있다.
- micro-ROS 최적화: micro-ROS의 제한 사항과 최적화 방법을 이해하고 적용할 수 있다.

4. Arduino Uno와 ROS 2 통합

- Arduino Uno를 ROS 2 네트워크에 통합하여 센서 데이터 수집 및 액추에이터 제어를 할 수 있다.
- VSCode Arduino 확장 설치: Visual Studio Code에서 Arduino 확장을 설치하고 설정하여 Arduino 개발 환경을 구축할 수 있다.
- Arduino 코드 작성 및 업로드: VSCode를 활용하여 Arduino 코드를 작성하고 보드에 업로드할 수 있다.
- Arduino와 ROS 2 통신 설정: Arduino와 ROS 2 간의 시리얼 통신을 설정하여 데이터 교환을 할 수 있다.
- Arduino에서 ROS 2 메시지 퍼블리싱: Arduino에서 센서 데이터를 수집하여 ROS 2 토픽으로 퍼블리싱할 수 있다.
- ROS 2에서 Arduino 제어 명령 전송: ROS 2에서 제어 명령을 전송하여 Arduino의 액추에이터를 제어할 수 있다.

5. Raspberry Pi Pico 활용

- Raspberry Pi Pico를 활용하여 센서 및 액추에이터를 제어할 수 있다.
 - 레벨 시프터 사용: 3.3V GPIO를 가진 Raspberry Pi Pico에서 5V 센서를 안전하게 연결하기 위해 로직 레벨 컨버터를 사용할 수 있다.
- LED 제어: Raspberry Pi Pico의 GPIO를 통해 LED를 제어하고, PWM을 활용하여 밝기를 조절할 수 있다.
 - 서보모터 제어: Raspberry Pi Pico에서 PWM 신호를 생성하여 서보모터의 각도를 제어할 수 있다.
 - DC 모터 제어: Raspberry Pi Pico와 모터 드라이버를 사용하여 DC 모터의 속도와 방향을 제어할 수 있다.
 - LCD 디스플레이 제어: Raspberry Pi Pico를 통해 LCD 디스플레이에 텍스트나 그래픽을 출력할 수 있다.