

코딩종합세트

UglyBot

어글리봇 스마트 코딩 교실



01 / 자동차 조립하기



02 / 작동하기 (레이싱 즐기기)



03 / 모바일 코딩

알씨카 만들기
라인따라가기
장애물 피하기

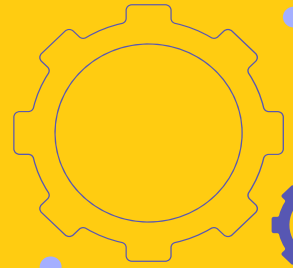


04 / PC 코딩

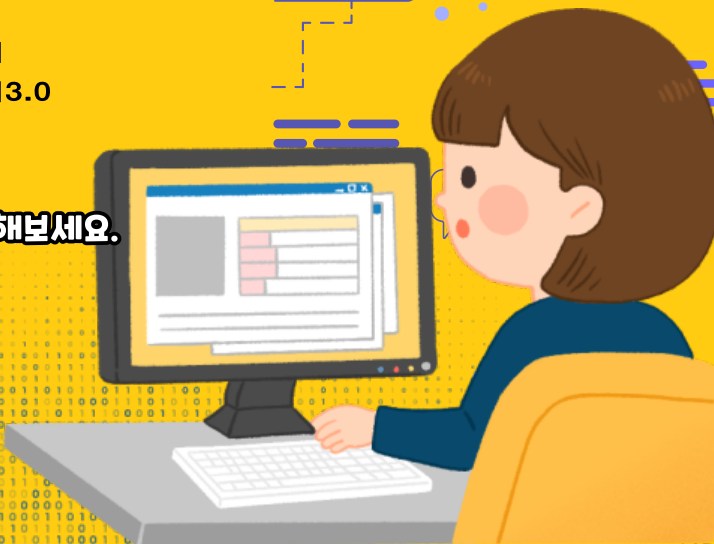
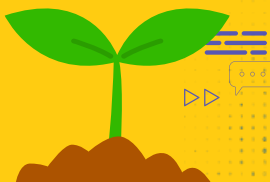
준비하기
스크래치3.0
엔트리
파이썬



05 / 이렇게 해보세요.

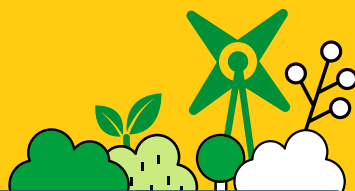


HTML



Chapter 1

자동차 조립하기



1. 구성품 확인

보드 1장



기어박스1팩



초음파센서 1개



바퀴2개



페이퍼1장

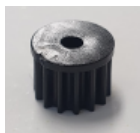


9V건전지

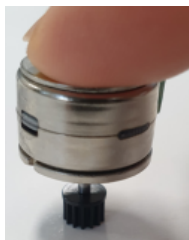


2. 기어박스 조립

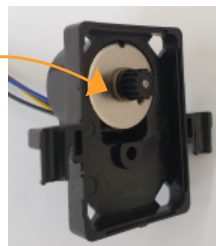
☐ 기어 박스 2개를 조립하여 준비한다.



① 기어(소)
평평한 면이
위로 오도록
놓기



② 기어(소) 구멍에 모터를 넣고
엄지로 지긋이 눌러서 완전히 꽂기



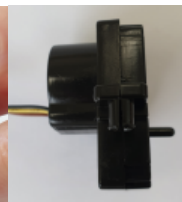
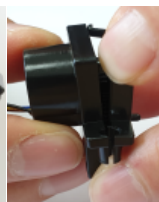
③ 스텝모터 박스(암)에 꽂기



④ 박스(수)
리벳 떼기



⑤ 박스(수)에
기어(대) 꽂기

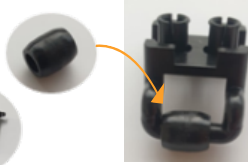


⑥ 박스(암)(수) 결합
(아랫 기둥을 꽂고 → 윗 기둥을 지긋이 누르면서
(암)에 꽂기 → 완성)

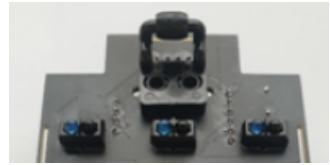
3. 캐스터 / 배터리 홀더 끼기



① 리벳 떼기



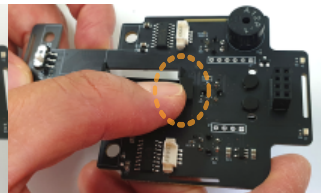
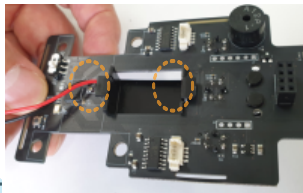
② 롤러 끼우기



③ 캐스터 끼기

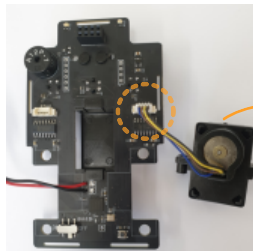


④ 배터리 홀더 끼우기
'ㄱ' 부분을 살짝 밀면서 끼는다.

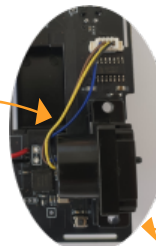


4. 기어박스 탑재 ☐ 기어박스 2개를 끼는다.

① 케이블 커넥터에 끼기



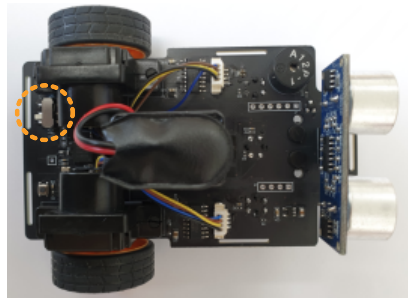
② 기어박스 끼기
& 밑면에서 리벳 끼기



5. 바퀴 탑재 & 초음파 센서 끼기 & 배터리 연결

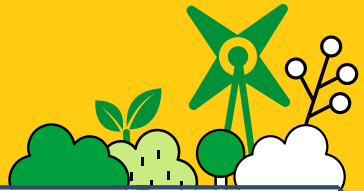
전원스위치

바퀴는 위아래로 지긋이
눌러주면 고정된다.



Chapter 2

작동하기 (레이싱즐기기)



1. 제이씨블록 앱 설치

- ① iOS (앱스토어), Android(플레이스토어) 에서 'jcblock'를 검색한다.
- ② 앱을 설치하고 실행한다.



2. 블루투스 연결

□ 안드로이드OS는 블루투스와 GPS 를 모두 켜 후 연결 시도 한다.

- ① 어글리봇 전원을 켜다.
- ② JCBLOCK 앱을 실행한다.
- ③ 블루투스 아이콘을 클릭한다.
- ④ 신호세기가 가장 큰 시리얼번호를 선택한다.
- ⑤ 연결 성공시 블루투스 아이콘 색상이 파랑으로변경되고, 어글리봇 부저에서 '삐리리' 소리가 난다.



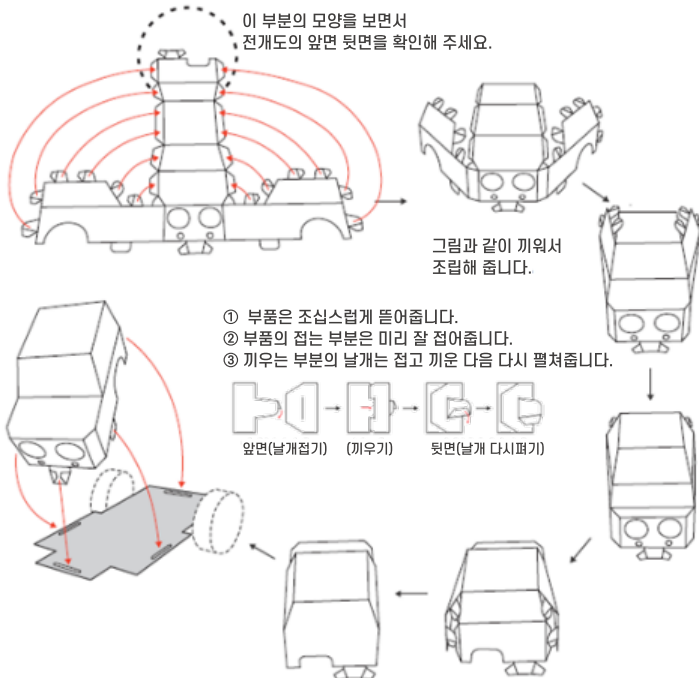
3. 예제 실행



4. 조종하기(알씨카 즐기기)

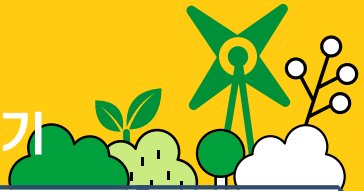


페이퍼 토이

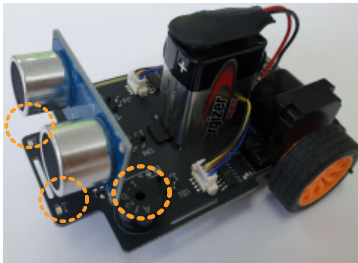


Chapter 3

모바일 코딩 알씨카만들기



제이씨블록 앱에서 체험한 '무선조종기' 예제를 직접 코딩하고 이해한다.



LED 부저



좌측 LED 경고음 (부저) 우측 LED

제이씨블록 기본 사용법



상태 정보

코딩 중에



전화가 오면 어떡하죠?
전원이 꺼지면 어떡하죠?
저장하지 않고 끝내면 어떡하죠?

걱정 마세요. 😊

앱 종료시 편집창에 쌓아둔 블록은
자동으로 tmp 파일로 저장됩니다.

☐ [예제] → [열기]에서 확인

tmp

1단계. 패드 디자인

무선조종기 인터페이스를 패드 기능을 이용하여 설계하고 디자인 한다.



프로그램 시작하기

그레이 패드 보이기 패드 디자인 선택

0번 조이스틱을 4:3에 놓기

1번 조이스틱을 4:10에 놓기

0번 버튼을 1:6에 놓기

0번 버튼을 텍스트위치로 설정

1번 버튼을 1:1에 놓기

2번 버튼을 1:12에 놓기

아이템 위치 지정

블루

블랙

그레이

회색큐브를 원하는 위치에 놓는다.

2단계. 변수 만들기



변수를 이용하여 단순하게 코딩 하도록 한다.

상태보기에서 변수 값 확인 가능

3단계. 변수 정의 & 모터 회전 설정



무한 반복하기

왼쪽모터 울(름) 0번 세로방향 조이스틱 / 1로 정하기

오른쪽모터 울(름) 0번 세로방향 조이스틱 / 1로 정하기

왼쪽모터 울(름) 1번 가로방향 조이스틱 / 3만큼 더하기

오른쪽모터 울(름) 1번 가로방향 조이스틱 / -3만큼 더하기

오른쪽모터를 왼쪽모터 세기로 돌리기

왼쪽모터를 오른쪽모터 세기로 돌리기

직진/후진 모터값 정의

제자리 회전 모터값(좌/우)

변수에 저장한 값으로 모터 세기 돌리기

0번 조이스틱

1번 조이스틱

3단계 "무한 반복하기" 블록 안에 순차적으로 4~5단계를 추가로 쌓도록 한다.

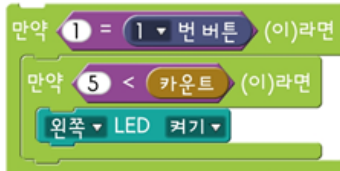
4단계. 경적 소리 삽입

0번 버튼을 누르면 부저음(라) 0.5초 소리 내기 구현한다.

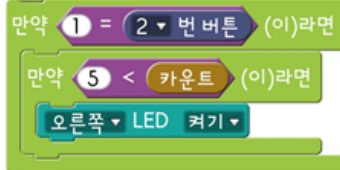


5단계. 방향 지시등

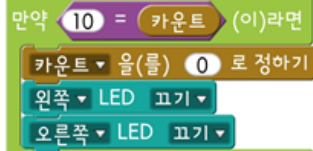
LED를 이용하여 방향 지시등을 코딩한다.



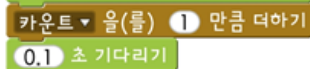
1번 누른상태에서
카운트5이하라면
→ 왼쪽 LED 켜기



2번 누른상태에서
카운트5이하라면
→ 오른쪽 LED 켜기



카운트=10 이면 카운트=0으로 초기화
→ 왼쪽 LED 끄기
→ 오른쪽 LED 끄기



카운트 1씩 증가하기
* 카운트는 LED 점멸 속도 조절을 위해 만든 변수



6단계. 코딩 저장하기

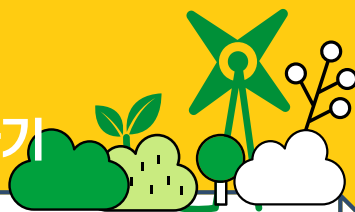


▢ [예제] → [열기]에서 확인

파일명

Chapter 3

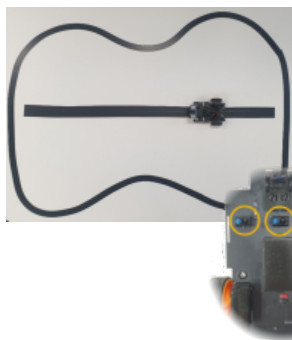
모바일 코딩 라인따라가기



적외선 센서를 이용하여 선을 따라 움직이는 라인트레이서를 구현 한다.

어글리봇은 3개의 적외선 센서가 탑재되어 중간 센서를 기준으로 선(검정색)을 트래킹하게 되며, 선을 이탈 할 경우에 좌/우 센서의 적외선 반사 값에 의해 선을 찾도록 자세를 보정하여 트래킹한다.

적외선 센서 반사값 : 검은색(0), 흰색(100)



1단계. 적외선 센서 & 변수 설정

사용할 센서를 설정하고 간결한 코딩을 위해 변수를 만든다.



프로그램 시작하기

라인기준값 > 음(를) 60로 정하기

왼쪽 > 적외선센서를 켜기 >

중간 > 적외선센서를 켜기 >

오른쪽 > 적외선센서를 켜기 >

무한 반복하기

왼쪽센서 > 음(를) 왼쪽 > 적외선 센서 값로 정하기

중간센서 > 음(를) 중간 > 적외선 센서 값로 정하기

오른쪽센서 > 음(를) 오른쪽 > 적외선 센서 값로 정하기

왼쪽모터 > 음(를) 20로 정하기

오른쪽모터 > 음(를) 20로 정하기

적외선 감도에 따라 라인 기준 값 설정

밝은 환경 → 값을 높게

어두운 환경 → 값을 낮게

어글리봇 모터 속도 값

속도가 빠르면 탈선을 높음

1단계 "무한 반복하기" 블록 안에 순차적으로 2~6단계를 추가로 쌓도록 한다.

2단계. 왼쪽센서에 라인(선) 있을 때

(왼쪽) 센서(1)이 선 위에 있을 때 왼쪽 모터를 감속하여 왼쪽으로 회전하도록 한다.

버튼	OFF	왼쪽 IR센서값	28
중간 IR센서값	100	오른쪽 IR센서값	100

만약 **왼쪽센서** < **라인기준값** (이)라면

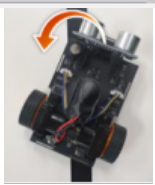
왼쪽모터 ▾ **을(를)** **-10** 만큼 더하기

이전상태 ▾ **을(를)** **1** 로 정하기

왼쪽센서 값으로 이탈 여부 체크

왼쪽 모터 감속 → **좌회전**

오른쪽 치우침 → **이전상태 =1**



3단계. 오른쪽센서에 라인(선) 있을 때

(오른쪽) 센서(3)이 선 위에 있을 때 오른쪽 모터를 감속하여 오른쪽으로 회전하도록 한다.

버튼	OFF	왼쪽 IR센서값	100
중간 IR센서값	100	오른쪽 IR센서값	34

만약 **오른쪽센서** < **라인기준값** (이)라면

오른쪽모터 ▾ **을(를)** **-10** 만큼 더하기

이전상태 ▾ **을(를)** **2** 로 정하기

오른쪽센서 값으로 이탈 여부 체크

왼쪽 모터 감속 → **우회전**

왼쪽 치우침 → **이전상태 =2**



4단계. 중간센서에 라인(선) 있을 때

버튼	OFF	왼쪽 IR센서값	99
중간 IR센서값	0	오른쪽 IR센서값	99

만약 **중간센서** < **라인기준값** (이)라면

이전상태 ▾ **을(를)** **0** 로 정하기

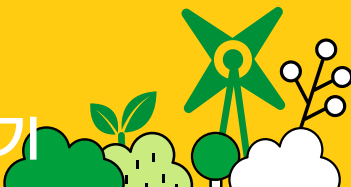
중간센서 값으로 이탈 여부 체크

치우침 없음 → **이전상태 =0**



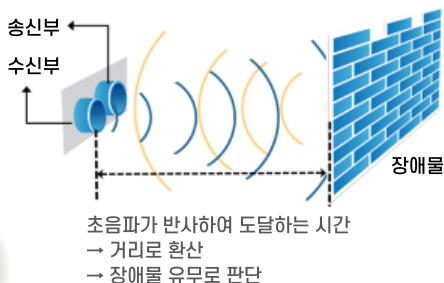
Chapter 3

모바일 코딩 장애물피하기



초음파 센서를 이용하여 장애물을 인지하고 장애물을 피하도록 구현한다.

초음파 센서는 음파를 만들어서 송출하는 송신부와 반사되는 음파를 받는 수신부로 나뉘어져 있다. 어글리봇은 초음파가 반사하고 수신부에 도달하는 시간을 거리로 환산하고 장애물이 있는지 판단한다.



1단계. 코딩 스케치

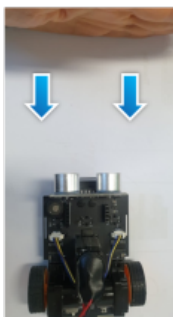
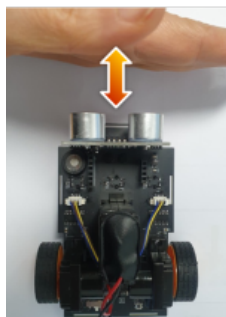
어글리봇의 움직임을 시뮬레이션하고 코딩을 설계하고 순서도를 생각한다.

장애물 인지

후진(탐색공간확보)

회전(길찾기 시도)

직진(장애물 없을 때)



2단계. 장애물 탐지 거리 설정

장애물 탐지 거리를 설정한다. 조건절에 따라 모터를 제어 하도록 설정한다.
초음파 특성 상 탐지거리는 가까울 수록 정확도가 높다.

프로그램 시작하기

무한 반복하기

만약 **초음파 센서 값** > **5** (이)라면

아니면

초음파 센서값 5 약5cm

초음파 센서값 21 약21cm

초음파 센서값 45 약45cm

3단계. 장애물이 없을 때

초음파 센서 값이 5 이상 일 경우에는 장애물 없음으로 인지하고 직진하도록 한다.

프로그램 시작하기

무한 반복하기

만약 **초음파 센서 값** > **5** (이)라면

왼쪽 ▾ 모터를 **50** 세기로 돌리기

오른쪽 ▾ 모터를 **50** 세기로 돌리기

아니면

초음파 센서값 21

장애물이 없을 때
→ 입력된 값으로 모터 회전하며 직진

4단계. 장애물이 있을 때

초음파 센서 값이 5 이하 일 경우에는 장애물 있음으로 인지하고 "정지 → 후진 → 회전" 순서대로 작동하여 새로운 길찾기를 시도한다.

초음파 센서값 5



왼쪽 ▾ 모터를 0 세기로 돌리기

오른쪽 ▾ 모터를 0 세기로 돌리기

모터 회전 세기 값을 '0' 으로 지정하여 0.1초 동안 정지한다.

0.1 초 기다리기

뒤 ▾ (으)로 10 cm 이동 10cm 후진

0.7 초 기다리기

시계방향 ▾ (으)로 90 도 회전 90도 시계 방향으로 회전

0.7 초 기다리기

5단계. 코딩 저장하기



☐ [예제] → [열기]에서 확인

파일명

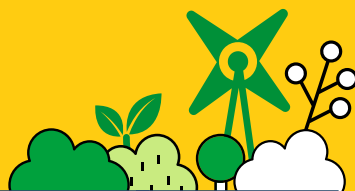
6단계. 실행하기

앱을 실행하고 어글리봇 초음파 앞에 손(장애물)을 대면 어글리봇은 정지 → 후진 → 회전 하며 길찾기를 시도한다.



Chapter 4

PC코딩준비하기



어글리봇을 하기의 PC용 코딩 언어를 사용 할 경우 블루투스 동글을 준비한다.

◇ 스크래치 ◇ 엔트리 ◇ 파이썬

1단계. USB 동글 연결



- ① USB 동글을 커넥터에 꽂는다.
- ② 동글의 빨간 LED가 천천히 깜빡거린다.
- ③ 어글리봇 전원을 켜다.
- ④ 왼쪽 초록색 LED가 깜빡거린다.

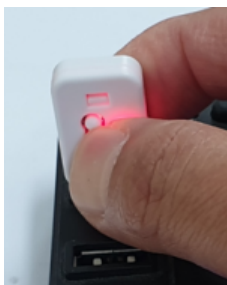


정상 연결 되면

- ⑤ 빨간 LED 깜빡임이 멈춘다.
- ⑥ 어글리봇 부저음이 빠리리~ 울리고, 왼쪽 초록색 LED 깜빡임이 멈춘다.

동글 초기화

동글에 저장된 어글리봇 블루투스 정보를 삭제하는 초기화 과정입니다.



● 만약, 정상 연결이 안되었을 경우

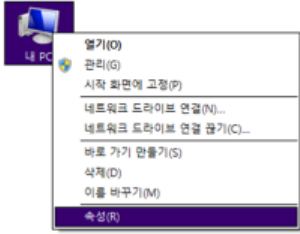
- ① 사진과 같이 동근버튼을 길게 누른다.
- ② 빨간 LED가 빠르게 깜빡거린다.
- ③ 잠시 후 빨간 LED가 천천히 깜빡거린다.
- ④ 위의 USB 연결 ③~⑤까지 진행 확인한다.

● 새로운 어글리봇과 연결하고자 할 경우

동글 초기화 후 새로운 어글리봇과 연결 시도한다.

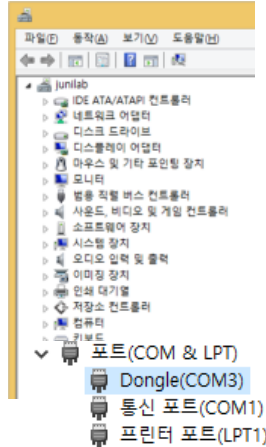
2단계. 환경 설정

① 내 컴퓨터 → 속성 → 장치관리자 실행



Dongle 드라이버가 미설치되었을
경우에만 3단계를 실행한다.

② 포트(COM&LPT)에서
Dongle(COM ●) 설치 여부 확인한다.

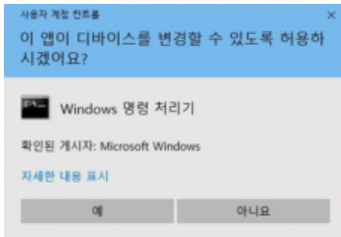


3단계. USB 드라이버 설치

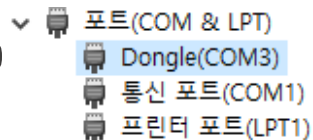
① www.junilab.co.kr → 공지사항 → [USB동글원도우드라이버] 게시물을 확인한다.

② 첨부 파일 DongleSetup 압축해제한다.

③ dogngleSetup.bat 파일을 더블클릭하여 실행한다.

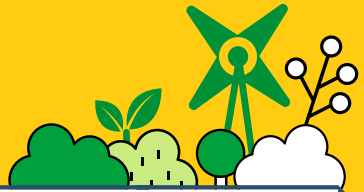


④ 장치관리자에서 드라이버를 확인한다.
(Dongle(COM●)에서 COM 번호는 자동으로 부여된다.)



Chapter 4

PC코딩엔트리

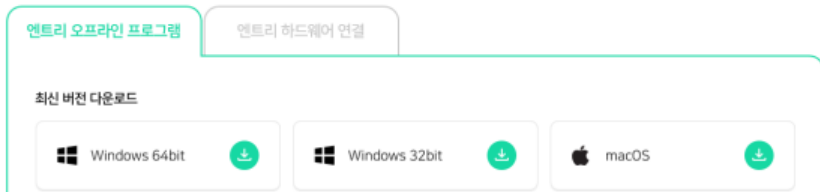


1단계. 엔트리 설치

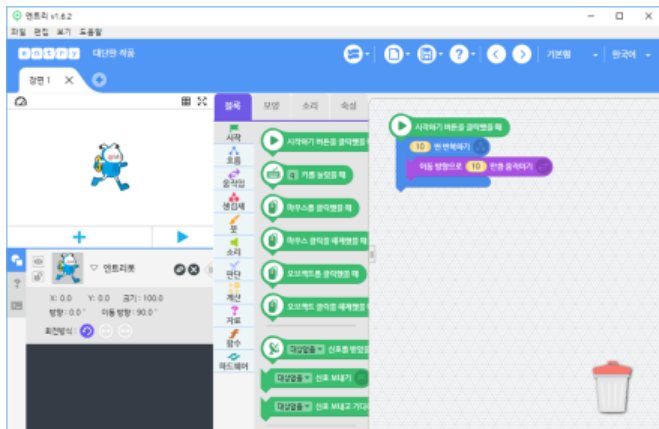
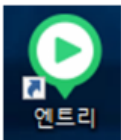
① <https://playentry.org/download/offline> 에서 컴퓨터 환경에 맞는 버전으로 설치한다.

다운로드

엔트리가 제공하는 공식 프로그램입니다.
누구나 자유롭게 다운로드 받아 사용할 수 있어요!



② 정상 설치 시 바탕화면에 아이콘이 생성된다. 아이콘을 클릭하여 프로그램이 정상동작하는지 확인 한다.



2단계. 엔트리&어글리봇 연결하기

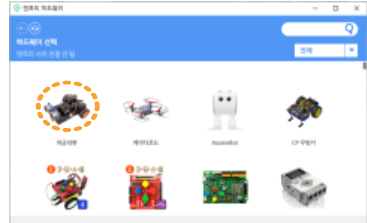
① 블록→하드웨어→"연결프로그램 열기"실행한다.



※연결 성공 화면은 닫지 않고
엔트리 코딩창으로 이동한다.

※연결 실패시 USB동글 드라이버를 별도 설치한다.
[PC코딩 준비하기 3단계] 참고

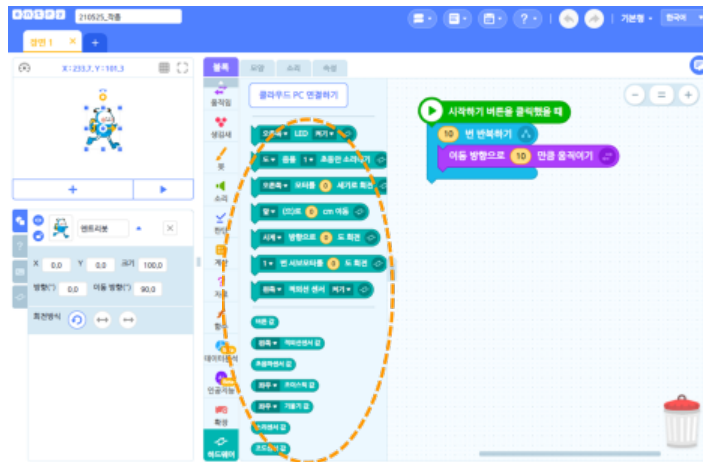
② 어글리봇을 선택한다.



③ "하드웨어>연결성공" 확인한다.



④ 코딩창에서 "블록 → 하드웨어" 메뉴에 단말기 관련 블록이 나타난다.



※어글리봇 관련 블록을
확인한다.

3단계. 코딩 하기

알씨카 만들기



① 왼쪽/오른쪽 모터 변수 정의

② 만일 화살표(↑) 눌렀을 때 직진
왼쪽모터=100, 오른쪽모터 =100

③ 만일 화살표(↓) 눌렀을 때 후진
왼쪽모터=-100, 오른쪽모터 =-100

④ 만일 화살표(→) 눌렀을 때 우회전
왼쪽모터=100, 오른쪽모터 =50
오른쪽 LED 켜기

⑤ 만일 화살표(←) 눌렀을 때 좌회전
왼쪽모터=50, 오른쪽모터 =100
왼쪽 LED 켜기

⑥ 만일 스페이스 눌렀을 때 부저음
왼쪽모터=50, 오른쪽모터 =100
왼쪽 LED 켜기

⑦ 모터 회전하기
조건별 저장된 모터 값으로 회전

시작하기 버튼을 클릭했을 때

계속 반복하기

왼쪽 모터 ▾ 를 0 (으)로 정하기 ?

오른쪽 모터 ▾ 를 0 (으)로 정하기 ?

만일 ▾ 위쪽 화살표 ▾ 키가 눌러져 있는가? (아)라면 ▴

왼쪽 모터 ▾ 를 100 (으)로 정하기 ?

오른쪽 모터 ▾ 를 100 (으)로 정하기 ?

만일 ▾ 아래쪽 화살표 ▾ 키가 눌러져 있는가? (아)라면 ▴

왼쪽 모터 ▾ 를 -100 (으)로 정하기 ?

오른쪽 모터 ▾ 를 -100 (으)로 정하기 ?

만일 ▾ 오른쪽 화살표 ▾ 키가 눌러져 있는가? (아)라면 ▴

오른쪽 모터 ▾ 를 (오른쪽 모터 ▾ 값 / 2) (으)로 정하기 ?

오른쪽 ▾ LED 켜기 ▾

아니면 ▾

오른쪽 ▾ LED 끄기 ▾

만일 ▾ 왼쪽 화살표 ▾ 키가 눌러져 있는가? (아)라면 ▴

왼쪽 모터 ▾ 를 (왼쪽 모터 ▾ 값 / 2) (으)로 정하기 ?

왼쪽 ▾ LED 켜기 ▾

아니면 ▾

왼쪽 ▾ LED 끄기 ▾

만일 ▾ 스페이스 ▾ 키가 눌러져 있는가? (아)라면 ▴

도 ▾ 음을 0.5 초동안 소리내기 ▾

왼쪽 ▾ 모터를 (왼쪽 모터 ▾ 값) 세기로 회전 ▾

오른쪽 ▾ 모터를 (오른쪽 모터 ▾ 값) 세기로 회전 ▾

라인따라가기

① 적외선 감도에 따라 라인 기준 값 설정
 밝은 환경 → 값을 높게
 어두운 환경 → 값을 낮게

② 센서 사용 선언
 이전상태(어글리봇 상태) 변수 정의

③ 왼쪽/오른쪽 모터 변수 기본값 정의

④ 오른쪽으로 치우칠 때
 왼쪽모터 속도 줄여 좌회전
 이전상태=1

왼쪽센서 20
 중간센서 100
 오른쪽센서 100



⑤ 왼쪽으로 치우칠 때
 오른쪽모터 속도 줄여 우회전
 이전상태=2

왼쪽센서 100
 중간센서 100
 오른쪽센서 9



⑥ 중간센서로 잘 따라 갈 때
 이전 상태=0

왼쪽센서 100
 중간센서 9
 오른쪽센서 100



⑦ 3개 센서 모두 선을 이탈 했을 때
 이전상태 값에 따라 치우침 방향 판단
 모터속도값 제어로 좌회전/우회전

왼쪽센서 100
 중간센서 100
 오른쪽센서 100



⑧ 모터 회전하기
 조건별 저장된 모터 값으로 회전



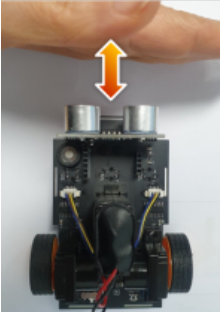
생략된 '⑦번' 조건

왼쪽센서 > 60 그리고 중간센서 > 60 그리고 오른쪽센서 > 60

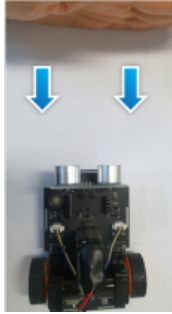
장애물피하기

하기 그림은 초음파 센서가 장애물을 인지 했을 때 길찾기 시도 과정이다.

장애물 인지



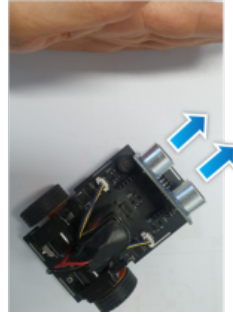
후진(탐색공간확보)



회전(길찾기 시도)



직진(장애물 없을 때)



① 초음파 센서값 변수 정의

② 센서값 10 기준으로 장애물 인지조건

③ 장애물이 없을 때 (초음파 센서>10)

왼쪽/오른쪽 모터 50세기로 직진

④ 장애물이 있을 때

멈춤 (왼쪽/오른쪽 모터 세기=0)

후진 (10cm 후진)

회전 (시계방향 90도 회전)

시작하기 버튼을 클릭했을 때

계속 반복하기

초음파 센서값 > 10 (으)로 경하기

만일 초음파센서 값 > 10 (이)라면

왼쪽 > 모터를 50 세기로 회전

오른쪽 > 모터를 50 세기로 회전

아니면

왼쪽 > 모터를 0 세기로 회전

오른쪽 > 모터를 0 세기로 회전

0.1 초 기다리기

뒤 > (으)로 10 cm 이동

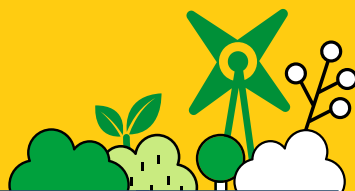
0.7 초 기다리기

시계 > 방향으로 90 도 회전

0.7 초 기다리기

Chapter 4

PC코딩스크래치3.0



1단계. JuniLink 설치

- ① <https://www.junilab.co.kr>>공지사항>[Scratch3.0설치파일] 게시물을 확인한다.
- ② "JuniLink.msi" 파일을 다운로드 후 실행하면 바탕화면에 2개의 아이콘이 생성된다.



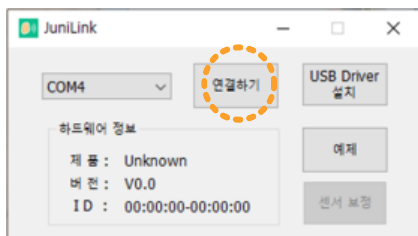
스크래치3.0 실행파일
(Scratch3.exe)



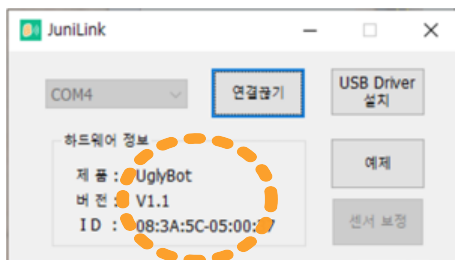
USB동글과 어글리봇 연결프로그램
(JuniLink.exe)

2단계. COM 포트 연결

- ① 아이콘 실행
- ② COM 선택 후 연결하기 실행



- ③ USB동글과 어글리봇이 정상 연결되면
어글리봇 정보가 출력된다.



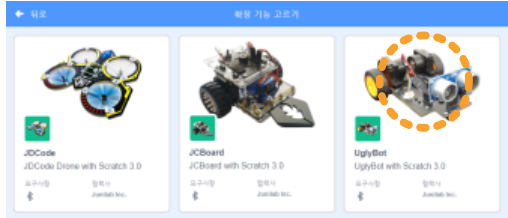
※COM 연결 성공 화면은
닫지 않고 스크래치를 실행한다.

3단계. 스크래치3.0 실행하기

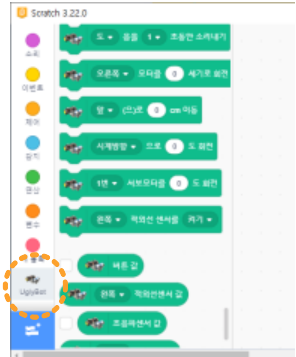
① 아이콘 실행 ② 확장하기 추가하기



③ 어글리봇 선택하기



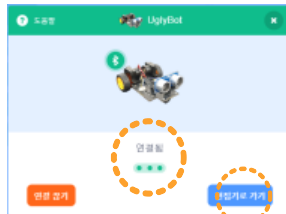
⑥ 어글리봇 블록 확인



④ 어글리봇 연결하기



⑤ 어글리봇 연결 성공확인

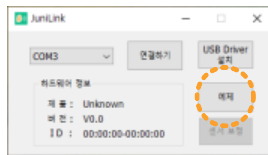


스크래치
예제보기
어글리봇의 다양한 예제를
볼 수 있어요.

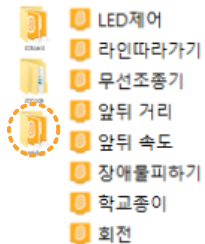
① 아이콘 실행



② 예제 버튼 클릭



③ 예제 선택



4단계. 코딩 하기

알씨카 만들기



① 왼쪽/오른쪽 모터 변수 정의

② 만일 화살표(↑) 눌렀을 때 직진
왼쪽모터=100, 오른쪽모터 =100

③ 만일 화살표(↓) 눌렀을 때 후진
왼쪽모터=-100, 오른쪽모터 =-100

④ 만일 화살표(→) 눌렀을 때 우회전
왼쪽모터=100, 오른쪽모터 =50
오른쪽 LED 켜기

⑤ 만일 화살표(←) 눌렀을 때 좌회전
왼쪽모터=50, 오른쪽모터 =100
왼쪽 LED 켜기

⑥ 모터 회전하기
조건별 저장된 모터 값으로 회전



라인따라가기

① 적외선 감도에 따라 라인 기준 값 설정
 밝은 환경 → 값을 높게
 어두운 환경 → 값을 낮게

② 센서 사용 선언
 이전상태(에글리봇 상태) 변수 정의

③ 왼쪽/오른쪽 모터 변수 기본값 정의

④ 오른쪽으로 치우칠 때
 왼쪽모터 속도 줄여 좌회전
 이전상태=1

왼쪽센서 20
 중간센서 100
 오른쪽센서 100



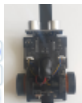
⑤ 왼쪽으로 치우칠 때
 오른쪽모터 속도 줄여 우회전
 이전상태=2

왼쪽센서 100
 중간센서 100
 오른쪽센서 9



⑥ 중간센서로 잘 따라 갈 때
 이전 상태=0

왼쪽센서 100
 중간센서 9
 오른쪽센서 100



⑦ 3개 센서 모두 선을 이탈 했을 때
 이전상태 값에 따라 치우침 방향 판단
 모터속도값 제어로 좌회전/우회전

왼쪽센서 100
 중간센서 100
 오른쪽센서 100



⑧ 모터 회전하기
 조건별 저장된 모터 값으로 회전



생략된 '⑦번' 조건

장애물피하기

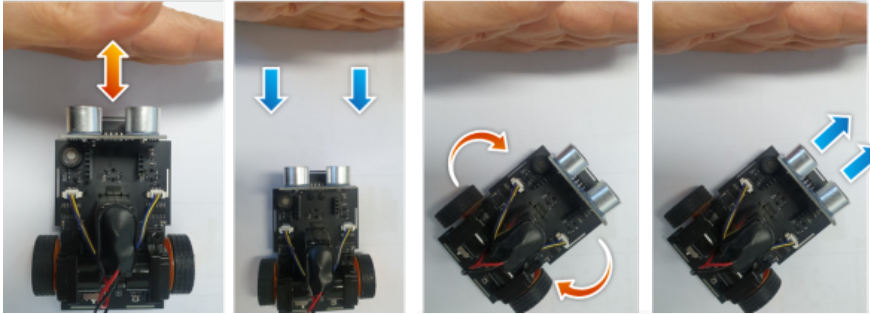
하기 그림은 초음파 센서가 장애물을 인지 했을 때 길찾기 시도 과정이다.

장애물 인지

후진(탐색공간확보)

회전(길찾기 시도)

직진(장애물 없을 때)



① 센서값 10 기준으로 장애물 인지조건

② 장애물이 없을 때 (초음파 센서>10)

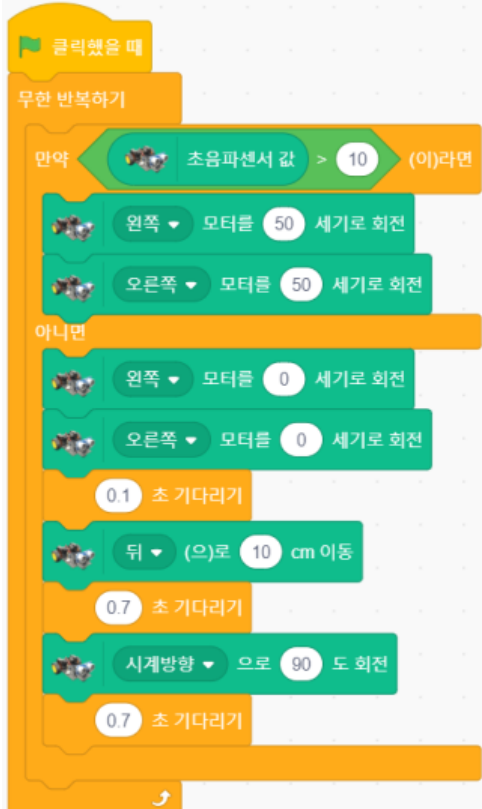
왼쪽/오른쪽 모터 50세기로 직진

③ 장애물이 있을 때

멈춤 (왼쪽/오른쪽 모터 세기=0)

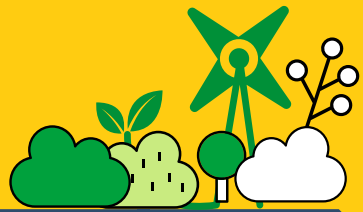
후진 (10cm 후진)

회전 (시계방향 90도 회전)



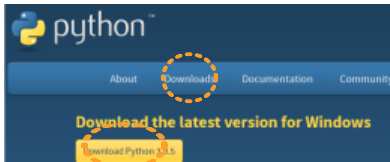
Chapter 4

PC코딩파이썬



1단계. 파이썬 다운로드

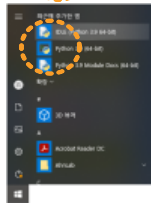
① <https://www.python.org>>Downloads



② 다운로드 실행



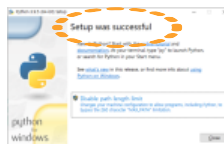
⑥ 설치 확인



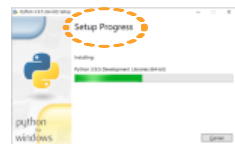
③ 설치하기



⑤ 설치 성공

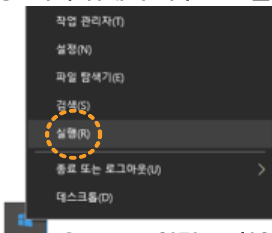


④ 설치 중

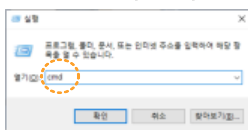


2단계. 어글리봇 라이브러리 설치

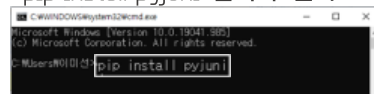
① 시작 위에서 마우스 오른쪽 클릭



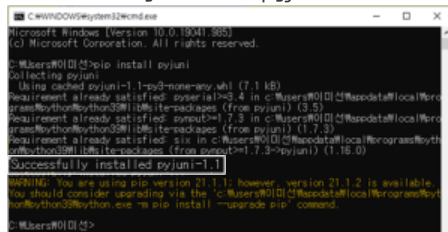
② 'cmd' 입력 → 확인



③ 'pip install pyjuni' 입력 후 엔터

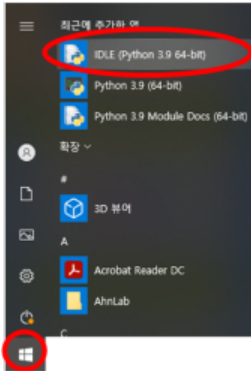


④ 'Successfully installed pyjuni-1.1' 성공 확인

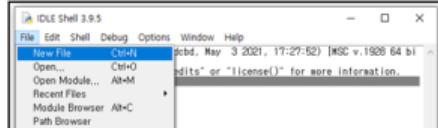


3단계. 어글리봇- 파이썬 실행하기

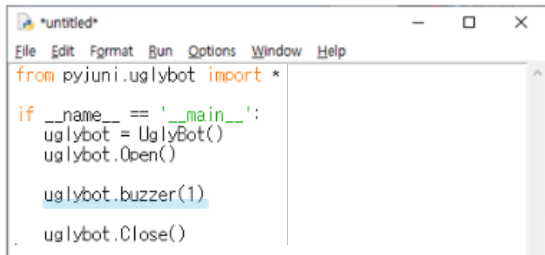
① IDLE 실행



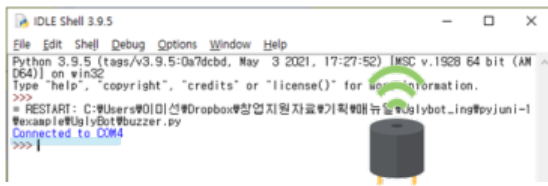
② New File 실행



③ 어글리봇 정상 연결 확인위해 코딩(부저 발생)

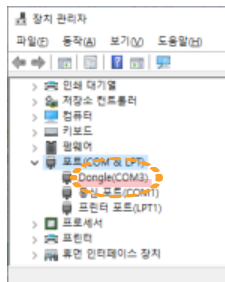
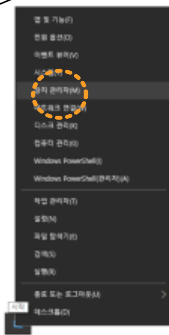


④ 어글리봇 정상 연결 확인위해 코딩(부저 발생)



정상연결이
안될 때
시리얼 포트를 수동으로
설정해주세요.

동글이 연결된 시리얼포트 번호 (COMO)를 확인하고
해당 번호를 Open("COMO") 함수에 기입한다.



```
from pyjuni.uglybot import *

if __name__ == '__main__':
    uglybot = UglyBot()
    uglybot.open("COM4")

    uglybot.buzzer(1)

    uglybot.Close()
```

4단계. 코딩 하기

```
from time import sleep
from pyjuni.uglybot import *
from pyjuni.jkeyevent import *
```

```
if __name__ == '__main__':
    uglybot = UglyBot()
    jkey = JKeyEvent()
    uglybot.Open()
```

```
while not jkey.isKeyEscPressed():
    leftMotor = 0 # 왼쪽모터 변수 =0
    rightMotor = 0 # 왼쪽모터 변수 =0
```

```
if jkey.isKeyUpPressed(): # ↑ 키 눌렀을 때 직진
    leftMotor = 100
    rightMotor = 100
```

```
if jkey.isKeyDownPressed(): # ↓ 키 눌렀을 때 후진
    leftMotor = -100
    rightMotor = -100
```

```
if jkey.isKeyRightPressed(): # → 키 눌렀을 때 우회전
    rightMotor = int(rightMotor / 2)
    uglybot.led(1)
else:
    uglybot.led(0)
```

```
if jkey.isKeyLeftPressed(): # ← 키 눌렀을 때 좌회전
    leftMotor = int(leftMotor / 2)
    uglybot.led(2)
else:
    uglybot.led(0)
```

```
if jkey.isKeyEnterPressed(): # Enter 키 눌렀을 때 부저울림
    uglybot.buzzer(2,0.05)
```

```
uglybot.motor(leftMotor, rightMotor) # 모터변수값으로 모터세기 정의
sleep(0.05)
```

알씨가 만들기



```

from time import sleep
from pyjuni.uglybot import *
from pyjuni.jkeyevent import *
leftir = 0
midir = 0
rightir = 0
preState = 0
def receiveData(packet):

```

라인따라가기



```

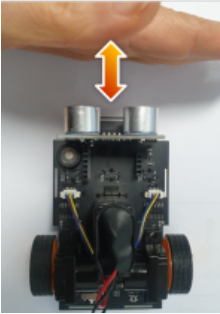
    global leftir # 글로벌 변수 사용
    global midir # 글로벌 변수 사용
    global rightir # 글로벌 변수 사용
    leftir = packet[8]; # 패킷 8번 자리에 좌측 근접센서값이 들어감
    midir = packet[9]; # 패킷 9번 자리에 중간 근접센서값이 들어감
    rightir = packet[10]; # 패킷 10번 자리에 우측 근접센서값이 들어감
if __name__ == '__main__':
    uglybot = UglyBot(receiveData)
    jkey = JKeyEvent()
    uglybot.Open()
    lineRef = 60
    uglybot.ir(7) # IR 센서 모두 켜기
    while not jkey.isKeyEscPressed(): # ESC 키 눌렀을 때까지 루프
        leftMotor = 20 # 왼쪽모터 세기. 값이 클수록 속도는 빨라지나 탈선 가능성이 높아짐
        rightMotor = 20 # 오른쪽모터 세기. 값이 클수록 속도는 빨라지나 탈선 가능성이 높아짐
        if leftir < lineRef: # 만약 왼쪽 센서값이 lineRef 미만일 경우 (오른쪽 치우침)
            leftMotor -= 10 # 왼쪽 모터 회전 속도 줄임.
            preState = 1 # 이전상태 값 '1'로 정의
        elif rightir < lineRef: # 만약 오른쪽 센서 값이 lineRef 미만일 경우(왼쪽 치우침)
            rightMotor -= 10 # 오른쪽 모터 회전 속도 줄임
            preState = 2 # 이전상태 값 '2'로 정의
        elif midir < lineRef: # 만약 중간 센서 값이 lineRef 미만 일 경우(치우침 없음)
            preState = 0 # 이전상태 값 '0'로 정의
        if leftir>lineRef and midir>lineRef and rightir>lineRef: # 3개센서 lineRef 이상일 때 (탈선 상태)
            if preState == 1: # 만약 이전상태 값이 '1'일 경우 오른쪽 치우침으로 판단됨
                leftMotor = 0 # 왼쪽 모터 세기 '0'으로 설정해 왼쪽으로 돌아가게 함
            if preState == 2: # 만약 이전상태 값이 '2'일 경우 오른쪽 치우침으로 판단됨
                rightMotor = 0 # 오른쪽 모터 세기 '0'으로 정의해 오른쪽으로 돌아가게 함
            uglybot.motor(leftMotor, rightMotor) # 모터 함수 호출
            sleep(0.05)
    uglybot.Close()

```

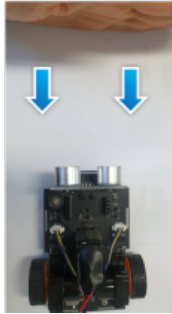
장애물피하기

하기 그림은 초음파 센서가 장애물을 인지 했을 때 길찾기 시도 과정이다.

장애물 인지



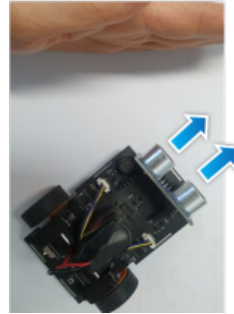
후진(탐색공간확보)



회전(길찾기 시도)



직진(장애물 없을 때)



```
from time import sleep
from pyjuni.uglybot import *
from pyjuni.jkeyevent import *
ultrasonic = 0 # 초음파 변수 정의
def receiveData(packet):
    global ultrasonic #글로벌 변수 선언
    ultrasonic = packet[11] # 패킷 11번에 초음파 데이터값 입력됨
if __name__ == '__main__':
    uglybot = UglyBot(receiveData)
    uglybot.Open()

    jkey = JKeyEvent()
    while not jkey.isKeyEscPressed():
        if ultrasonic > 10: # 만약 초음파 값(장애물 거리)이 10이하일 경우
            uglybot.motor(50, 50) #모터세기 50으로 직진
        else: # 초음파 값(장애물 거리)이 10 이상일 경우
            uglybot.motor(0, 0) # 멈춤
            sleep(0.2)
            uglybot.move(-10) # 10Cm 후진
            sleep(0.5)
            uglybot.rotation(90) # 90도 회전
            sleep(0.5)
            sleep(0.05)
    uglybot.Close()
```

TROUBLE SHOOTING

이렇게 해보세요.



1. 안드로이드 OS에서 블루투스 연결이 안돼요.

안드로이드 스마트폰은 설정에서 블루투스와 GPS(위치정보) 둘 다 켜주세요.

2. 갑자기 블루투스가 자주 끊어져요.

잘 되던 어글리봇이 갑자기 작동이 멈추거나 블루투스 연결이 불안하면 건전지 잔량이 없으니 알카라인 건전지로 교체 해주세요.

3. 바퀴가 잘 돌아가지 않아요.

기어박스에서 (소)(대)기어톱니가 맞물려 꽃아 졌는지 확인 해 주세요.
기어박스 분해는 핀셋으로 하시면 됩니다.

4. 바퀴가 쉽게 빠져요.

어글리봇을 옆으로 세워서 바퀴를 위에서 아래로 지긋이 눌러서 고정 시켜주세요.

5. 건전지를 교체해도 작동이 안됩니다.

어글리봇은 9V 사각 건전지 (알카라인) 건전지를 권장합니다.
ex)에너자이저, 듀라셀

더 많은 정보가 필요하시면 언제든지 연락 주세요. TEL)1599-4729, help@junilab.co.kr