주니어 임베디드SW 챌린저

-1차 기술 지원 교육-

2019. 07. 13

- 컬러 센서
- 모터제어
- •로봇 제어 기초
- 함수
- 반복문
- 조건문
- 변수

목차

- •프로그래밍 기초
- Python 개발 환경 구축

• 로봇 제어 응용 • 정확한 회전

• 사각형 그리기

• 검은 선 카운트

•라인트레이싱

마무리







•구축 순서

- 1. Python 설치
- 2. EV3Dev 이미지 파일 다운로드
- 3. 부팅 SD 카드 제작
- 4. EV3Dev 부팅





1. Python 설치

Python 프로그래밍을 하려면 PC에 Python 프로그램을 설치해야 한다.

<u>https://www.python.org/downloads/</u> 에 접속하여 파이썬 최신 설치파일을 다운로드한다.





Python 설치파일을 실행하면 아래와 같은 화면이 나타난다.

이 때, 먼저 'Add Python 3.7 to PATH' 옵션을 체크하고 'Install Now'를 클릭한다.



FUNERS



2. EV3Dev 이미지 파일 다운로드

EV3에는 기본적으로 레고에서 제공하는 펌웨어가 설치되어 있다.

Python을 이용해서 프로그래밍을 하려면 먼저 이미지 파일을 다운 받아 부팅 SD카드를 만든 다음, SD카드를 EV3의 SD 슬롯에 삽입한 후 EV3를 실행해야 한다.

<u>https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3</u>에 접속하여 SD card

image를 다운로드 한다.



Minimum specifications Micro SDHC (min. 4GB, max. 32GB) with Application Performance Class A1. Windows 10 or Mac OS computer. Visual Studio Code version 1.31 or above. You can now use your EV3 Brick to unleash the power of Python programmin MicroPython. Simply install the EV3 MicroPython image onto any micro SD ca up your EV3 Brick from it to start programming straight away. Switching bac standard LEGO® MINDSTORMS® EV3 firmware is just as simple. We have al full documentation and sample API code to get you started.

FUNERS

How to get started

Download and flash the EV3 MicroPython image onto a micro SD card
 Insert your micro SD card into the SD card slot on the EV3 Brick and tu
 Download, install, and launch the free Visual Studio Code editor on you

Install and activate the LEGO Education EV3 extension

5. Connect the EV3 Brick to your computer and start to code

You will find a detailed description of the above steps in the EV3 MicroPython documentation PDF (link below).

Links to documentation and firmware

Download the EV3 MicroPython documentation (incl. API, sample code).

Download the EV3 MicroPython micro SD card image



3. 부팅 SD카드 제작

부팅 SD카드를 제작하기 위해선 'Etcher'라는 프로그램을 설치해야 한다.

- ① <u>https://etcher.io/</u>에 접속하여 다운로드한다.
- ② 다운로드 한 balenaEtcher-Setup-1.5.39-x64를 실행하여 설치를 진행한다.

FUNERS



Python 개발 환경 구축

'Etcher'를 실행하였으면 아래와 같은 순서대로 진행한다.

- ① SD카드 리더기를 이용하여 SD카드(8Gb~32Gb)를 PC에 연결한다.
- ② Select image를 클릭하여 1에서 다운로드한 파이썬 SD Card image 파일을 선택한다.

FUNERS

③ SD카드의 경로가 제대로 설정 되었는지 확인한다. 경로가 다를 경우 Change를 클릭하 여 올바른 SD카드 경로를 설정해준다.





④ '경로를 설정하고 Flash!를 클릭한다.

⑤ 설치가 완료되면 Flash Complete!라는 화면으로 바뀐다.





4. EV3Dev 부팅

EV3의 SD카드 슬롯에 minid SD카드를 삽입하고, EV3 전원을 켠다. (Enter 버튼 누름) 부팅이 완료되면 아래와 같은 EV3 화면을 확인 할 수 있다.



FUINERS



5. Visual Studio Code

<u>https://code.visualstudio.com/</u>으로 접속하여 설치파일을 다운로드 한다.

Code editing. **Redefined.**

Free. Open source. Runs everywhere.



By using VS Code, you agree to its license and privacy statement.

B	EXTENSIONS		볼 …	app.ts	۷	ww.ts	×	package.json	README.md		Ξ	
	@popular				import ap import de	p from ' bugModule	/app 2 = r e	'; equire('debug'));			
Q � ⊛		1.2.2 ♀356K ↑ for Visual Studio icrosoft rthon 0♀211K ↑ hting, Debugging on Jayamanne	Code (p., Install		<pre>import ht const deb // Get po const por app.set[]'</pre>	tp = requ ug = debu rt from (t = norma port', po	uire(ugModu enviro alizef ort[];	'http'); ule('node-expre onment and stor Port(process.er	ess-typescript:se e in Express. w.PORT '3000'	:rver'););		
¢		bugger for citre ebug your JavaScr icrosoft JS Diagno. IC++ 0.7	ipt code Install		<pre>// create const ser server.li server.on server.on</pre>	 CSSImp CSSSup export export import import Messag 	ortRul portsR s Script ePort	e ule s				Fa
		0.6.39 Q.99K 1 ch Go language si tehoban Lint 0.10Q.88K 1 tegrates ESLint inf rk Baeumer	upport f Install		/** * Normal */ function let por	<pre>normalize normalize t = parse</pre>	izePort onst port(Port()	t ort: number [strin (val: any): num val, 10);	g boolean mber string boole	an {		

FUINERS

FUNERS

1) Visual Studio Code 설치





🗙 설치 - Visual Studio Code	-		×
사용권 계약 계속하기 전에 다음 중요한 정보를 읽어 보세요.			N
다음 사용권 계약을 읽어 주세요. 설치를 계속하려면 먼저 니다.	이 계약 조건에	동의해야	합
MICROSOFT 소프트웨어 사용권 계약서			Â
MICROSOFT VISUAL STUDIO 코드			
본 사용권 계약은 Microsoft Corporation(또는 거주 지역에 귀하 간에 체결되는 계약입니다. 본 사용권 계약은 위에 5 용됩니다. 본 계약은 다른 조건이 있는 경우를 제외하고 4 Microsoft 서비스 또는 업데이트에도 적용됩니다.	따라 계열사 중 경시된 소프트웨 논프트웨어에 대) 하나)과 어에 적 한	~
 ● 계약에 동의함(A) ○ 계약에 동의 안 함(D) 			
< 뒤로(B)	다음(N) >	2	취소
🗙 설치 - Visual Studio Code	-		×
추가 작업 선택 어떤 작업을 추가로 수행하시겠습니까?			X
설치 프로그램에서 Visual Studio Code을(를) 설치하는 동안 후 [다음]을 클릭하세요.	안 수행할 추가 ?	작업을 선택	한
추가 바로 가기:			
바탕 화면 바로 가기 만들기(D)			

□ "Code(으)로 열기" 작업을 Windows 탐색기 파일의 상황에 맞는 메뉴에 추가

□ Code을(를) 지원되는 파일 형식에 대한 편집기로 등록합니다.

✓ PATH에 추가(다시 시작한 후 사용 가능)

□ "Code(으)로 열기" 작업을 Windows 탐색기 디렉터리의 상황에 맞는 메뉴에 추가

< 뒤로(B) 다음(N) >

취소

기타:





FUINERS

2) Visual Studio Code 실행

실행 시 주의 할 점
 실행 아이콘을 오른쪽 클릭을 하여
 "관리자 권한으로 실행"을 선택한다.



FUNERS

3) Extention 설치

N	파일	!(<u>F</u>)	편집(<u>E</u>)	선택	영역(<u>S</u>)	보기(<u>V</u>)	이동	;(<u>G)</u> [
D		EXTE	NSIONS: M/	ARKETF	PLACE		×	•••
Ŷ		EV3	MicroPytł	non	2			
°		E educat	Proje LEGO	0® M ct ten ® Edu	INDSTOF nplates an cation	rd docum	.0.2 G entati	o 192 ion nstall
¢	1	li	micro micro langx	o pyth obit ex m	i on 0.0.1 ct for vsco	ode	န Ir	947 Istall



4) EV3-MicroPython Project 생성

 확장 기능 설치가 완료되면 좌측 메 뉴 중 레고 마인드스톰 아이콘을 클 릭 후 Create a new project 를 클릭 한다.

 프로젝트 이름을 입력하고 Enter 키를 누르면 프로젝트를 저장할 폴더를 선택 하게 된다.











- 프로젝트 폴더 열기 및 폴더 닫기 상 단의 File 메뉴를 클릭을 하면 된다.
- 위에서 생성한 프로젝트를 Open Folder..를 클릭하여 연다.







- 현재 열린 프로젝트 폴더 내부에 launch.json 파일이 있는데 program
 경로를 아래와 같이 변경한다.
- "/home/robot/\${workspaceRootFolder Name}/\${relativeFile}"
- 경로 변경 이후 활성화된 파일의 코드를 실행하게 된다.

탐색기	() launchijson 🗙
▲ 영경 있는 편집기 × () launch.json vscode ▲ MY_PROJECT	<pre>1 { 2 // Use IntelliSense to learn about possible attributes. 3 // Hover to view descriptions of existing attributes. 4 // For more information, visit: https://go.microsoft.com/fwlink/?linkid=83038?</pre>
viscose 0 extensions.json 0 launch.json A extensions	5 "version": "0.2.0", 6 "configurations": [7 { 8 ["norm": "Download and Run"
gitignore main.1.py	9 "type": "ev3devBrowser", 10 "request": "launch". 11 "program": "/home/robot/\${workspaceRootFolderName}/\${relativeFile}"
• нан ру	

5) 간단한 코드 작성

main.py를 더블 클릭하여 열고

brick.sound.beep()

- 도구모음의 File -> save 저장
- F5 -> 실행

1 #!/usr/bin/env pybricks-micropython 2 3 from pybricks import ev3brick as brick 4 from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, 5 | | | | | | | | | InfraredSensor, UltrasonicSensor, GyroSensor) 6 from pybricks.parameters import (Port, Stop, Direction, Button, Color, 7 | | | | | | SoundFile, ImageFile, Align) 8 from pybricks.tools import print, wait, StopWatch 9 from pybricks.robotics import DriveBase 10 11 brick.sound.beep()

FUNERS



6) EV3와 PC를 연결하기

1. PC 와 EV3를 USB 케이블을 통하여 연결

https://www.youtube.com/watch?v=OZHH5t4BjNI

2. PC 와 EV3를 블루투스로 연결

https://youtu.be/niziYwWA_7U

https://youtu.be/mdwFQ2sr-mU







• 변수 - 컨테이너, 가방, 그릇, 풍선

- 다양한 데이터를 보관할 수 있는 공간
- 변수 안에 넣어둔 데이터는 언제 어디서 열어 봐도 항상 동일하다.
- 변수의 내용물은 언제든지 바꿀 수 있다.

3. 대소문자를 구분해서 사용해야 한다.

- - 1. 변수명은 영문자(A-Z, a-z), 밑줄문자(_), 숫자를 조합하여 구성한다.
- 변수 선언 시 유의 사항

2. 변수명의 첫글자는 반드시 영문자나 밑줄문자()로 시작해야 한다.

4. 이미 다른 용도로 사용되고 있는 키워드 또는 예약어는 변수명으로 사용할 수 없다.

FUNERS

•변수 선언하기

height = 185
weight = 85
print(height)
print(weight)
print(height+weight)

test.py ▷ ... from pybricks import ev3brick as brick from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor) from pybricks.parameters import (Port, Stop, Direction, Button, Color, SoundFile, ImageFile, Align) from pybricks.tools import print, wait, StopWatch from pybricks.robotics import DriveBase height = 185 weight = 85 print(height) print(weight) print(height+weight) PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Starting: brickrun --directory="/home/r Started. 185 85 270





• 조건문 – 특정한 조건에 따라 실행 여부가 결정된다.

- If 조건 : 만약 ~이면
- elif 조건 : 아니면 만약 ~이면
- else : 위 조건들이 아닌 모든 경우
- 불량품이 발생 했을 때 상점의 대처
 - if 구매 후 3일 이전: 환불
 - elif 구매 후 3일 이후: 교환
 - elif 구매 후 7일 이후: 환불/교환 불가
 - else : 경찰 부름

FUINERS

•if - elif – else 사용하기

number = 12

```
if number < 5:

print("5보다 작아요")

elif number == 5:

print("5입니다")

elif number > 5:

print("5보다 커요")

else:

print("알 수 없어요")
```

```
🕹 test.py ♭ ...
     #!/usr/bin/env pybricks-micropython
     from pybricks import ev3brick as brick
     from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
 4
                                      InfraredSensor, UltrasonicSensor, GyroSensor)
     from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                      SoundFile, ImageFile, Align)
     from pybricks.tools import print, wait, StopWatch
     from pybricks.robotics import DriveBase
     number = 12
     if number < 5:
         print("5보다 작아요")
     elif number == 5:
         print("5입니다")
     elif number > 5:
         print("5보다 커요")
         print("알 수 없어요")
         OUTPUT DEBUG CONSOLE TERMINAL SEARCH
PROBLEMS
Starting: brickrun --directory="/home/robot/my project" "/home/robot/my project"
Started.
5보다 커요
```





• 반복문 – 특정한 조건에 따라 반복 여부가 결정된다.

• while 조건: 조건을 만족 하는 동안 반복

• for 변수 in 리스트: 리스트 안의 모든 경우 만큼 반복(중복 X)

•불량품이 발생 했을 때 상점의 대처

• while 물건이 불량품인 경우: 교환

FUINERS

•while 사용하기

count = 0

while count < 10: print(count) count = count + 1

print("열 번 끝")

🕏 test.py ▶
<pre>1 #!/usr/bin/env pybricks-micropython</pre>
3 from pybricks import ev3brick as brick
4 from pybricks.ev3devices import (Motor, TouchSensor, ColorSen
5 InfraredSensor, UltrasonicSe
6 from pybricks.parameters import (Port, Stop, Direction, Butte
7 SoundFile, ImageFile, Align
8 from pybricks.tools import print, wait, StopWatch
9 from pybricks.robotics import DriveBase
10
11 count = 0
12
13 while count < 10:
14 print(count)
15 count = count +1
16 print("열만 픝")
1/
10
20
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SEARCH
9
1
2
3
4
5
6
7
8
9
열번 끝

FUINERS

• for 사용하기

for count in range(0,10): print(count)

print("열 번 끝")

🕏 test.py 🕨
<pre>1 #!/usr/bin/env pybricks-micropython 2 5 from pybricks import outbrick as brick</pre>
 from pybricks import ev3brick as brick from pybricks.ev3devices import (Motor, TouchSensor, Cold infraredSensor, Ultrasor from pybricks.parameters import (Port, Stop, Direction, E from pybricks.tools import print, wait, StopWatch from pybricks.robotics import DriveBase for count in range(0,10):
12 print(count) 13 14 print("열번 끝") 15 16 17 18 19
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SEARCH Starting: brickrundirectory="/home/robot/my_project" "/home/ started.





•혼합문 – 조건문과 반복문이 동시에 쓰이는 경우

• 불량품이 발생 했을 때 상점의 대처

• while 물건이 불량품인 동안:

if 교환 횟수가 5번 이하: 교환

else : 환불

FUNERS

•혼합문 사용하기

count = 0

```
while count < 10:
if count%2 == 0:
print(count, "= 짝 수입니다.")
else:
print(count, "= 홀 수 입니다.")
count = count+1
```

```
print("열 번 끝")
```

```
test.py ▷ ...
     from pybricks import ev3brick as brick
     from pybricks.ev3devices import (Motor, TouchSensor, ColorSens)
                                     InfraredSensor, UltrasonicSe
     from pybricks.parameters import (Port, Stop, Direction, Butto
                                     SoundFile, ImageFile, Align)
     from pybricks.tools import print, wait, StopWatch
     from pybricks.robotics import DriveBase
     count = 0
     while count < 10:
         if count%2 == 0:
             print(count, "= 짝 수 입니다.")
            print(count, "= 홀 수 입니다.")
         count = count +1
     print("열번 끝")
       OUTPUT DEBUG CONSOLE TERMINAL SEARCH
ROBLEMS
Starting: brickrun --directory="/home/robot/my project" "/home/robot
Started.
  = 짝수 입니다.
 = 홀 수 입니다.
  = 짝수 입니다.
 = 홀 수 입니다.
 = 짝수 입니다.
 = 홀 수 입니다.
  = 짝수 입니다.
 = 홀 수 입니다.
 = 짝수 입니다.
```

9 = 홀 수 입니다. 열번 끝



• 연산자

- 연산자를 활용하면 변수에 저장된 값을 다양하게 조작하여 사용할 수 있다.
- 프로그램 상에 괄호 없이 두 개 이상의 연산자를 한 문장에서 혼합하여 사용하게 되면, 연산 자의 우선 순위에 따라 실행된다.

연산자	의미	예시	연산 결과
+	덧셈	4 + 2	6
-	뺄셈	4 – 2	2
/	나눗셈	4 / 2	2
*	곱셈	4 * 2	8
%	나머지	4 % 2	0

<산술 연산자>

<관계 연산자>

연산자	의미	예시	연산 결과
<	a가 b보다 작다	4 < 2	false
>	a가 B보다 크다	4 > 2	true
==	a와 b가 같다	4 == 2	false
!=	a와 b가 같지 않다	4 != 2	true
<=	a가 b보다 작거나 같다	4 <= 2	False
> =	a가 b보다 크거나 같다	4 >= 2	true

FUINERS

<대입 연산자>

연산자	의미	예시	연산 결과
+=	덧셈 누적	a+=1	a=a+1
-=	뺄셈 누적	a-=1	a=a-1
/=	나눗셈 누적	a/=1	a=a/1
=	곱셈 누적	a=1	a=a*1

<논리 연산자>

연산자	의미	예시	연산 결과
and	양쪽의 값이 모두 true인 경우 true	(2<4) and (5>8)	false
or	어느 한 쪽만 true인 경우 ture	(2<4) or (5<8)	true
not	True면 false, false면 ture	not(4<20)	false



함수

- 입력을 받아 출력을 내보내는 것
- 불필요한 코드의 반복을 삭제해
 주는 편리한 도구
- def 함수이름(입력1, 입력2,):
- 함수의 3요소
 - 함수 이름
 - 입력
 - 출력 -> 없을 수도 있다



FUINERS

•adder() 함수 만들기	def = 함수 정의 A adder = 함수 이용	♥ test 1 2 3 4 5 6 7 8	<pre>py > #!/usr/bin/env pybricks-micropython from pybricks import ev3brick as brick from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor</pre>
def adder(a, b): return a+b ##여기까지 함수 정의	a = 입덕1 b = 입력2 return = 출력	9 10 11 12 13 14 15 16	<pre>trom pybricks.robotics import DriveBase def adder(a,b): return a+b print(adder(1,1)) print(adder(2,3)) print(adder(2,3)))</pre>
print(adder(1, 1)) ##호출 print(adder(2, 3)) ##호출 print(adder(adder(1, 1), adde	er(2, 3) <mark>)</mark>) ##호출	10 17 18 19 20 21	
		PROBLE Start Start 2 5 7	MS OUTPUT DEBUG CONSOLE TERMINAL SEARCH ing: brickrundirectory="/home/robot/my_project" "/home/robot/ ed.

FUNERS

• delay() 함수 만들기 1

import time

t = time current_time = t.time() print(current_time)

while True: if t.time() - current_time > 5: break

print(t.time())

```
test.py ▷ ...
      #!/usr/bin/env pybricks-micropython
      from pybricks import ev3brick as brick
      from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
  4
                                        InfraredSensor, UltrasonicSensor,
      from pybricks.parameters import (Port, Stop, Direction, Button, Co
                                        SoundFile, ImageFile, Align)
      from pybricks.tools import print, wait, StopWatch
      from pybricks.robotics import DriveBase
      import time
      t = time
      current time = t.time()
 14
      print(current time)
      while True:
          if t.time() - current time > 5:
              break
      print(t.time())
         OUTPUT DEBUG CONSOLE TERMINAL
                                         SEARCH
PROBLEMS
Starting: brickrun --directory="/home/robot/my project" "/home/robot/my
Started.
1551738981.010175
1551738986.010261
```

FUNERS

• delay() 함수 만들기 2

import time

def delay(second):
 t = time
 current_time = t.time()
 print(current_time)

while True: if t.time() - current_time > second: break

def = 함수 정의 시작

wait = 함수 이름

second = 입력

print(t.time()) ##여기까지 함수 정의

delay(5) ##함수 호출

	<pre>#!/usr/bin/env pybricks-micropython</pre>
2	
	from pybricks import ev3brick as brick
	<pre>from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,</pre>
	InfraredSensor, UltrasonicSensor,
	from pybricks.parameters import (Port, Stop, Direction, Button, Co
	SoundFile, ImageFile, Align)
	from pybricks.tools import print, wait, StopWatch
	from pybricks.robotics import DriveBase
10	
11	import time
12	
13	<pre>def delay(second):</pre>
14	
15	t = time
16	current_time = t.time()
17	<pre>print(current_time)</pre>
18	
19	while True:
20	<pre>if t.time() - current_time > second:</pre>
21	break
22	
23	<pre>print(t.time())</pre>
24	
25	delay(5)
26	
PROBLEN	AS OUTPUT DEBUG CONSOLE TERMINAL SEARCH
Starti	.ng: brickrundirectory="/home/robot/my_project" "/home/robot/my_
Starte	ed.
155173	99139.419999
155173	39144.420068





•프로그래밍이란?

- 한 편의 연극을 만드는 것.
- •메모리 = 무대
- 코드 = 대본

- 배우가 소품들을 가지고 연기를 한다.
- 객체가 변수들을 가지고 함수로 동작한다.

• 배우의 특징을 고려

• 장면 하나하나를 구체적으로 상상

어떻게? 왜?

- 언제? 어디서? 누가? 무엇을?
- •시간의 흐름에 따라 논리적으로

•대본을 잘 쓰려면?

- 적절한 변수, 함수, 객체의 사용
- •장면 하나하나를 구체적으로 상상
- 언어 문법에 알맞게
- 순차 진행에 따라 논리적으로
- 프로그래밍을 잘 작성하려면?







피	人
2	



•로봇 프로그래밍

- 프로그램 작성 방법에서 실행해 보았던 코드에 사용된 함수에 대해 알아본다.
- 기본 설정
 - #!/usr/bin/env pybricks-micropython : EV3에서 프로그램을 실행 시키기 위해서는 반드시 제일 첫 행에 입력해야한 다.
 - from pybricks ... : EV3를 사용하기 위한 모든 public 속성/함수를 사용하기 위해 반드시 필요하다.

모든 프로그램 에 입력한다.





모터 제어

FUNERS

•모터 제어

• 모터 객체를 생성하고 객체 안의 함수를 이용하여 제어한다.

•모터 제어 함수

- Motor(출력 포트) 모터 객체
- run(speed) 모터 켜기
- run_time(speed = -1050~1000, time(ms), Stop Type(), wait(True or False)
- Stop Type
 - Stop.BRAKE :모터의 전원이 제거되었다가 다시 전원이 들어옴.
 - Stop.HOLD : 모터의 전원이 제거되지 않고 현재 위치를 유지하려고 붙잡고 있음.
 - Stop.COAST: 모터의 전원이 제거되어 자연스럽게 멈춤.

• Wait

- True : 모터 회전 후 다음 코드 실행
- False : 모터 회전 명령 후 다음 코드 바로 실행



•모터 하나를 5초 동안 회전 시키기

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B) #모터 300의 속도로 5초간 회전 motorB.run_time(300,5000,Stop.BRAKE,True)

FUINERS

• 두개의 모터를 5초 동안 회전 시키기

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B) motorC = Motor(Port.C) #모터 300의 속도로 5초간 전진

```
motorB.run_time(300,5000,Stop.BRAKE,False)
motorC.run_time(300,5000,Stop.BRAKE,True)
```

FUNERS

•모터 제어

• 모터 객체를 생성하고 객체 안의 함수를 이용하여 제어한다.

•모터 각도 제어 함수

- Motor(출력 포트) 모터 객체
- run_angle(speed = -1050~1000, rotation_anagle(), Stop Type(), wait(True or False)
- Stop Type 타입
 - Stop.BRAKE :모터의 전원이 제거되었다가 다시 전원이 들어옴,
 - Stop.HOLD : 모터의 전원이 제거되지 않고 현재 위치를 유지할려고 붙잡고 있음,
 - Stop.COAST: 모터의 전원이 제거되어 자연스럽게 멈춤
- Wait 타입
 - True : 모터 회전 후 다음 코드 실행
 - False : 모터 회전 명령 후 다음 코드 바로 실행



• 전진 360도

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#모터 300의 속도로 360도 회전

motorB.run_angle(300,360,Stop.BRAKE,False)
motorC.run_angle(300,360,Stop.BRAKE,True)

• 후진 360도

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#모터 300의 속도로 - 360도 회전

motorB.run_angle(300,-360,Stop.BRAKE,False)
motorC.run_angle(300,-360,Stop.BRAKE,True)



• 회전하기

• 로봇을 회전시키는 방법



FUNERS

#모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#포인트 턴

motorB.run_angle(300,360,Stop.BRAKE,False)
motorC.run_angle(-300,360,Stop.BRAKE,True)

brick.sound.beep()
wait(1000)

#스윙 턴

motorB.run_angle(300,360,Stop.BRAKE,True)

brick.sound.beep()
wait(1000)

#커브 턴

motorB.run_angle(300,360,Stop.BRAKE,False)
motorC.run_angle(100,360,Stop.BRAKE,True)

brick.sound.beep()
wait(1000)



컬러 센서



• 컬러 센서 활용

• 컬러 센서로 색상의 값을 측정하는 방법.







·컬러 모드(COL-COLOR)

• 컬러 센서가 감지하는 표면의 색상에 따라 다른 숫자 값이 반환된다.

숫자	색상	숫자	색상	숫자	색상	숫자	색상
0	None	1	Black	2	Blue	3	Green
4	Yellow	5	Red	6	White	7	Brown



•반사광 모드(COL-REFLECT)

- 컬러 센서를 반사광 모드로 작동시키면, 센서는 현재 측정하고 있는 표면이 빛에 반사되는 정도를 감지하여 측정값을 0~100사이의 값으로 반환한다.
- 어두운 영역과 밝은 영역을 구분할 때는 두 영역의 경계로 판단할 수 있는 **문턱값(threshold)**을 정해야 한다.

→ 가장 일반적으로 문턱값을 정할 떄는 **두 표면의**

빛 값을 측정한 후 평균값을 계산하는 방법을 활용한다.





•컬러 센서

• 컬러 센서 객체를 생성하고 객체 안의 함수를 이용하여 제어한다.

•센서 값 제어

- ColorSensor(입력 포트) 컬러센서 객체를 생성
- 센서 모드 지정 및 센서가 감지하고 있는 값

reflection(), color(), ambient()



• 컬러센서가 빨간색을 인식 할 때 까지 전진 후 정지

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#컬러 센서 선언

color sensor = ColorSensor(Port.S3)

#붉은색이 감지되지 않을 동안 반복

while color_sensor.color() != 5: #하얀색이 감지되면 if color_sensor.color() == 6: motorB.run(300) motorC.run(500) else:

motorB.run(500)
motorC.run(300)



• 밝으면 빠르게 어두우면 느리게

#!/usr/bin/env pybricks-micropython

#모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#컬러 센서 선언 color_sensor = ColorSensor(Port.S3)

#취소 버튼을 누를때까지 반복

while True:

motorB.run(color_sensor.reflection()*100)
motorC.run(color_sensor.reflection()*100)



로봇 제어 응용





•정확한 회전 연습

- •모터의 회전 각을 이용
- 포인트 턴과 스윙 턴으로 90도 180도 360도 정확하게 회전하기
- 포인트 턴과 스윙 턴 각각의 특징과 차이점을 확인
- 포인트 턴의 회전 각도 * 2 = 스윙 턴의 회전 각도

로봇 제어 응용



•사각형을 그리며 움직이고 제자리로 돌아오기

- 연속적인 움직임을 연습
- 직진 회전 직진 회전 직진 회전 직진 회전
- 직진과 회전 사이의 대기시간을 통해 관성 제거하기
- 반복문을 사용하여 코드의 반복 줄이기

FUINERS

•사각형 돌기

#!/usr/bin/env pybricks-micropython

모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#함수 선언

def turn_right(angle:int): motorB.run_angle(300,180,Stop.BRAKE,False) motorC.run_angle(-300,180,Stop.BRAKE,True)

#반복원

for count in range(0,4):
 motorB.run_time(300,2000,Stop.BRAKE,False)
 motorC.run_time(300,2000,Stop.BRAKE,True)
 wait(500)
 turn_right(360)

로봇 제어 응용



•N번 째 검은 선에서 정지하기

- 컬러 센서를 이용하여 검은 선의 개수를 세며 전진하기
- 센서가 검은 선을 통과하는 시간(또는 모터 회전 각도)가 필요
- 반복문을 사용하여 코드의 반복을 제거



• 4번 째 검은선에서 멈추기

#!/usr/bin/env pybricks-micropython

모터 선언

motorB = Motor(Port.B)
motorC = Motor(Port.C)

#컬러 센서 선언

color_sensor = ColorSensor(Port.S3)

반복문

for count in range(0,4):
 motorB.run(300)
 motorC.run(300)

#반사값이 15보다 작으면 whi]

while True: if color_sensor.reflection() < 15: break

brick.sound.beep()
wait(500)

motorB.stop(Stop.BRAKE)
motorC.stop(Stop.BRAKE)





•라인트레이싱

• 컬러 센서를 활용한 라인트레이싱

		white	grey
	1	우회전	좌회전
	2	좌회전	우회전

FUNERS

•10 초 동안 라인트레이싱

- <u>30</u> = 문턱 값
- (흰색의 value + 검은색의 value) / 2

/usr/bin/env pybricks-micropython

#모터 선언 motorB = Motor(Port.B) motorC = Motor(Port.C)

컬러 센서 선언 color_sensor = ColorSensor(Port.S3) current time = time()

while time() - current_time < 10: if color_sensor.reflection() < 30: motorB.run(300) motorC.run(0) else : motorB.run(0) motorC.run(300)



•참고 사이트

마무리

- EV3 Micro Python 학습 및 API : <u>https://education.lego.com/en-us/support/mindstorms-</u> ev3/python-for-ev3
- Visual studio code 설치 관련: <u>https://youtu.be/cqtRqsl6xMc</u>
- EV3와 PC의 연결 관련 : <u>https://youtu.be/TNXqizQTZhs</u>

•교육, 기술 지원 및 경기장 구매 문의

퓨너스 (<u>www.funers.com</u>) T.070-8670-8911