

경기도 자율주행 경진대회 교육 워크숍

Subject : LiDAR

Automation Lab.



목차

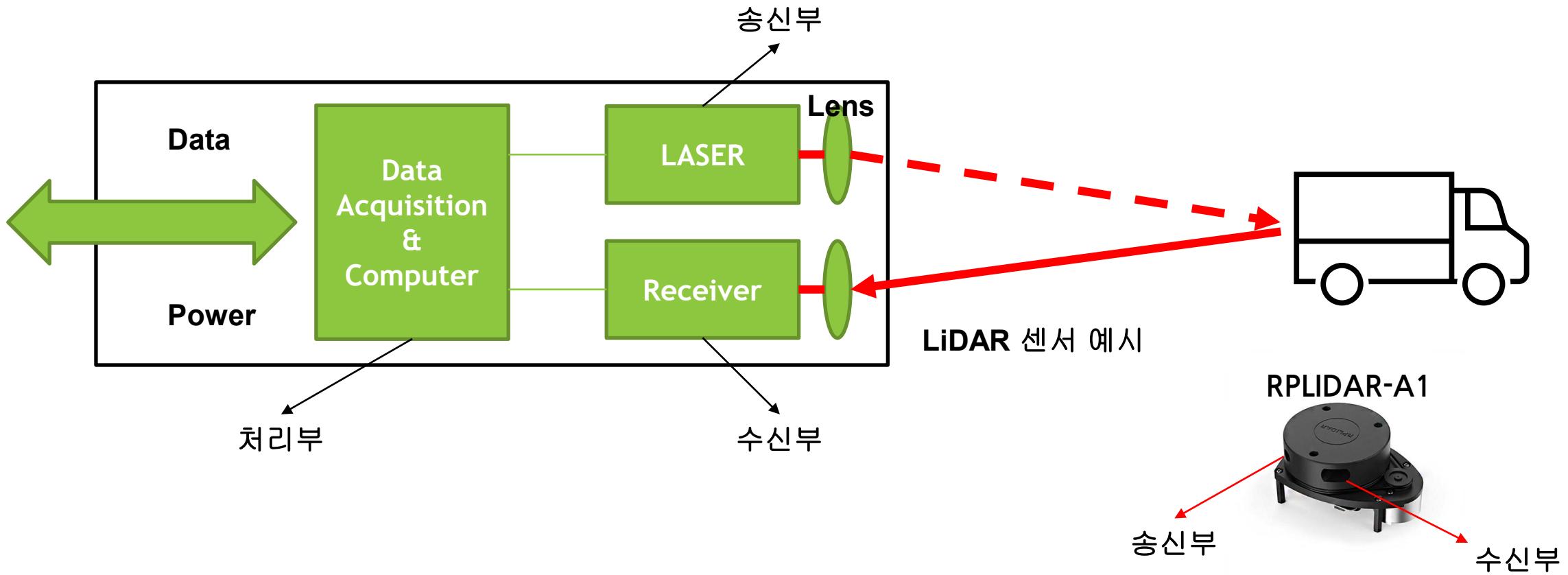
- **INTRODUCTION**

- **EXERCISE**



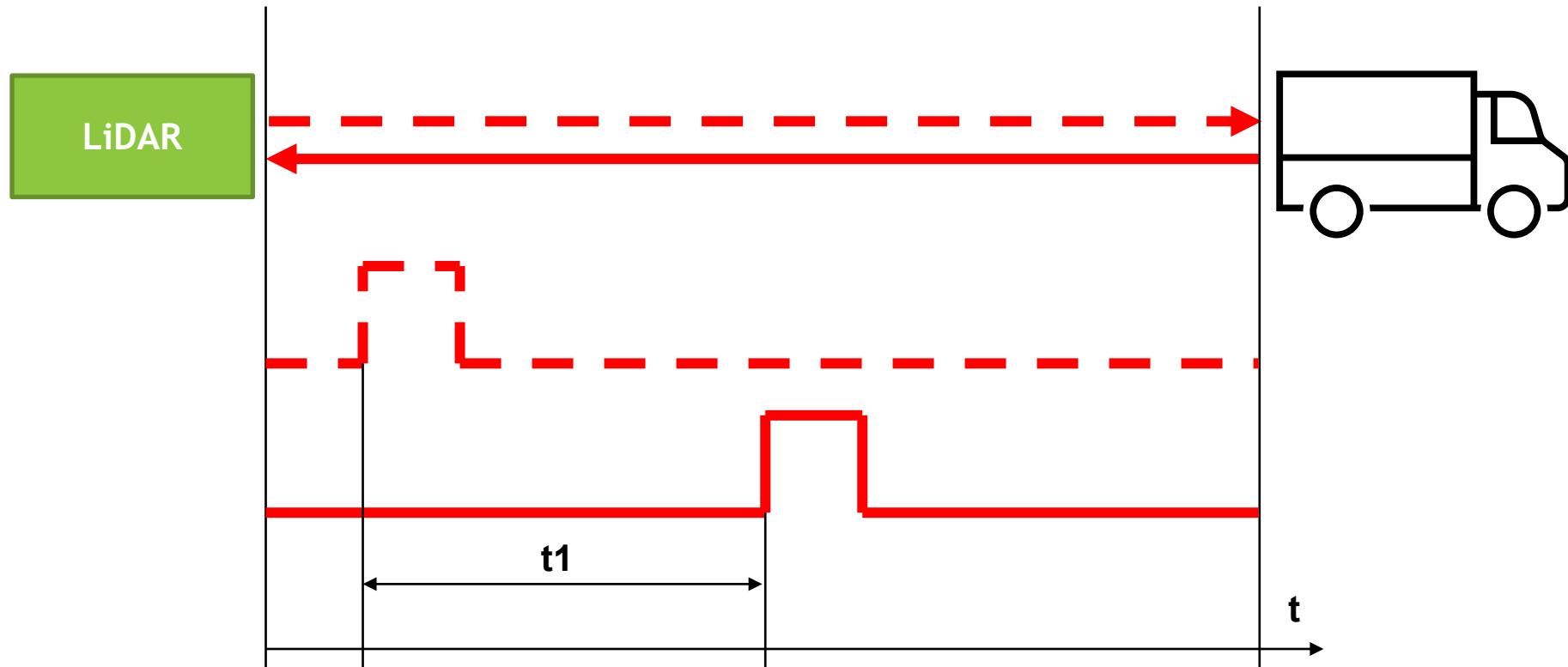
Introduction

■ LiDAR(Light Detection And Ranging)



Introduction

■ Distance Measurement ● Time-Of-Flight(ToF)



Introduction

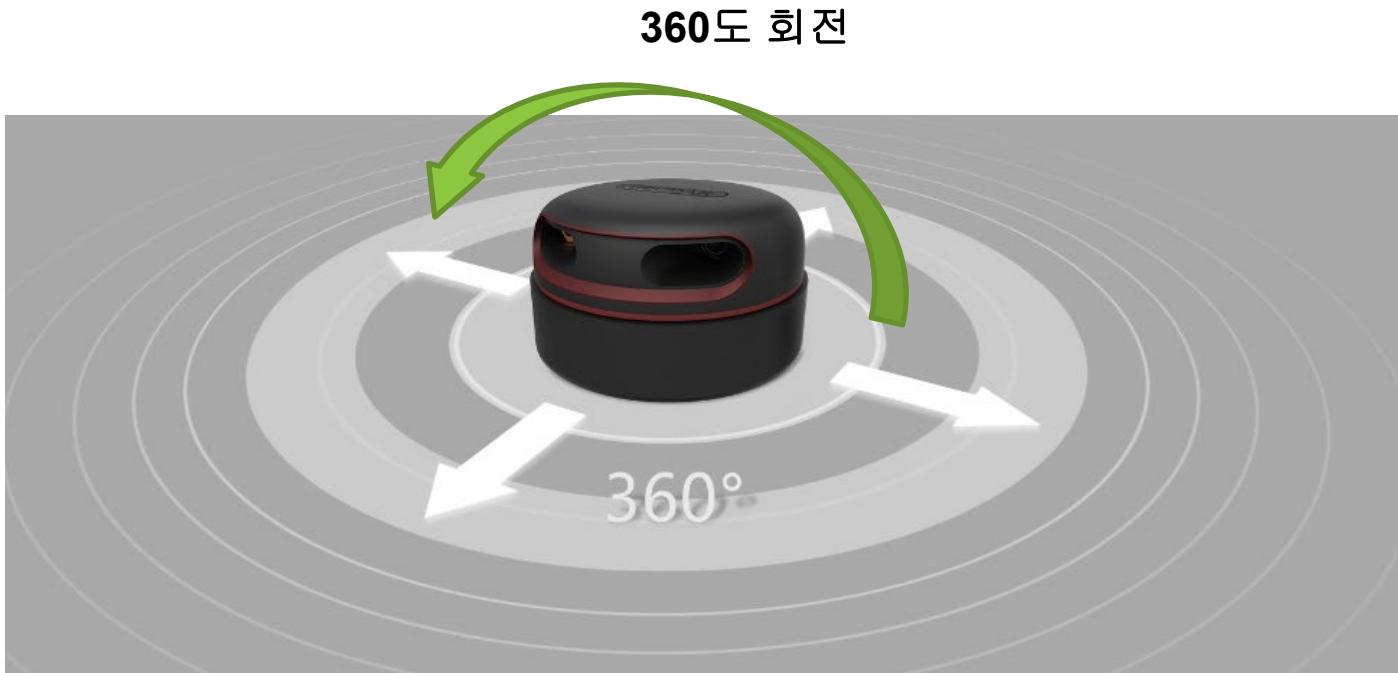
■ RPLIDAR-A1(LiDAR)



- 제품 사양
 - 거리 범위 : 0.15m ~ 6m
 - 각도 범위 : 0 ~ 360 degree
 - 각도 해상도 : 0.45 ~ 0.9 degree

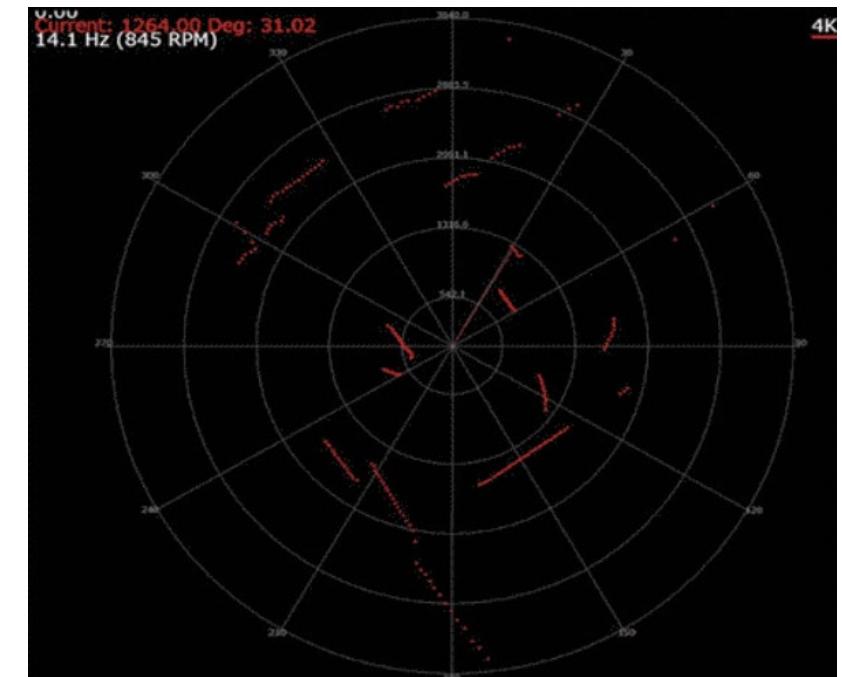
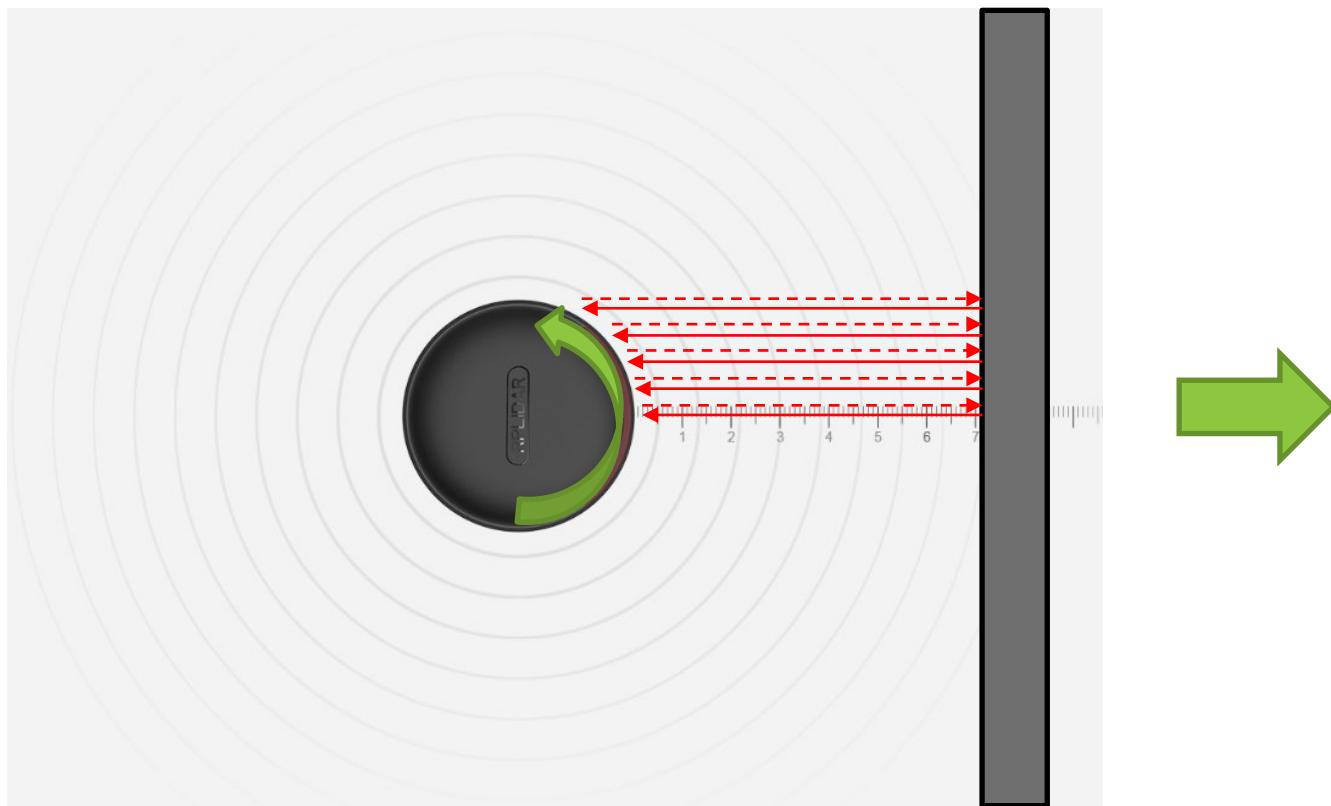
Introduction

■ LiDAR Operating



Introduction

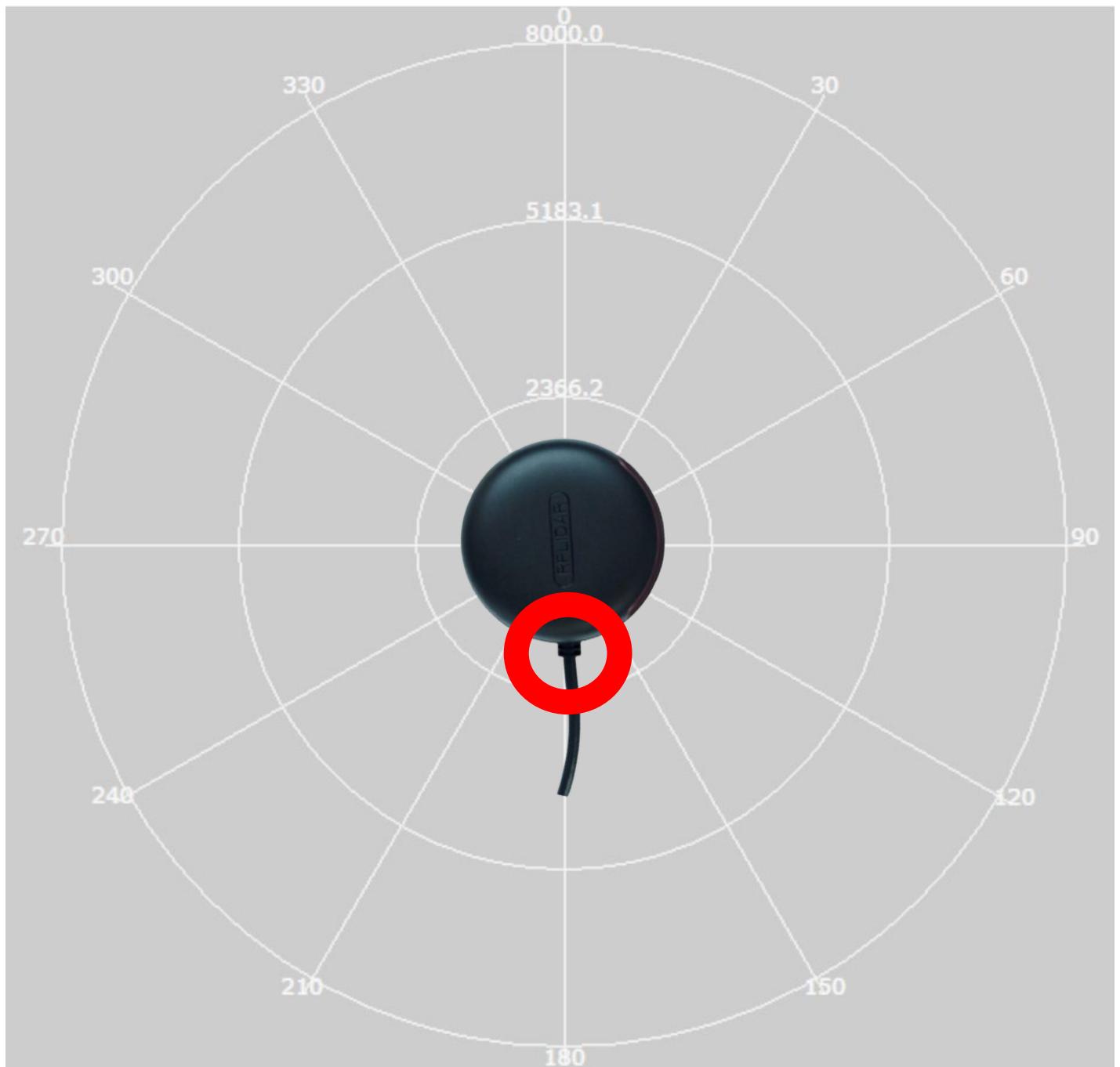
■ LiDAR Operating



Introduction

■ LiDAR Operating

- 측정 각도



목차

- INTRODUCTION

- EXERCISE



Exercise 1

■ LiDAR 응용프로그램 연동하기

- Step 1 : Hardware 연결

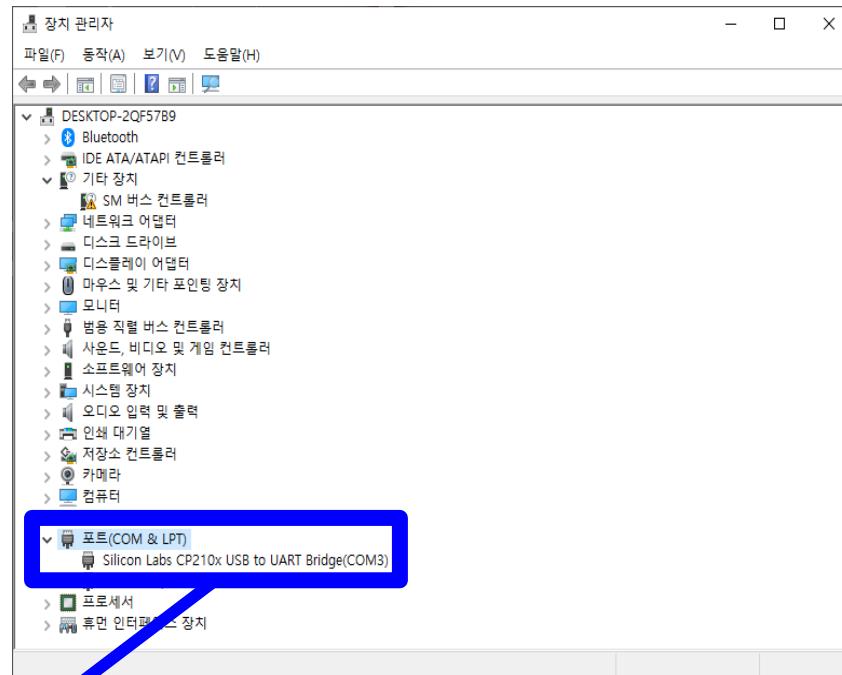
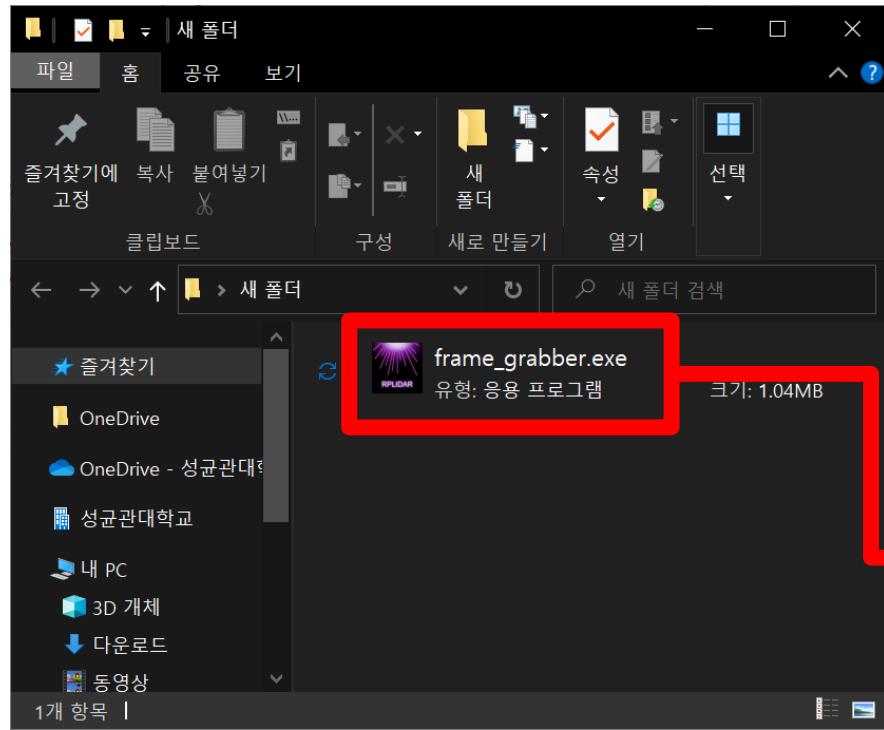
→ LiDAR의 전원 USB 케이블을 허브 또는 PC에 연결



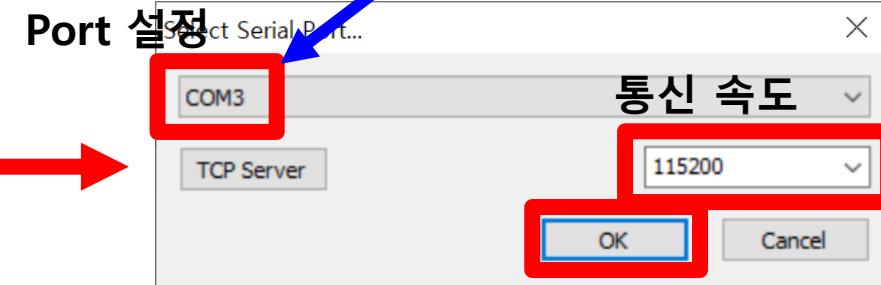
Exercise 1

■ LiDAR 응용프로그램 연동하기

● Step 2 : 프로그램 실행 및 연결



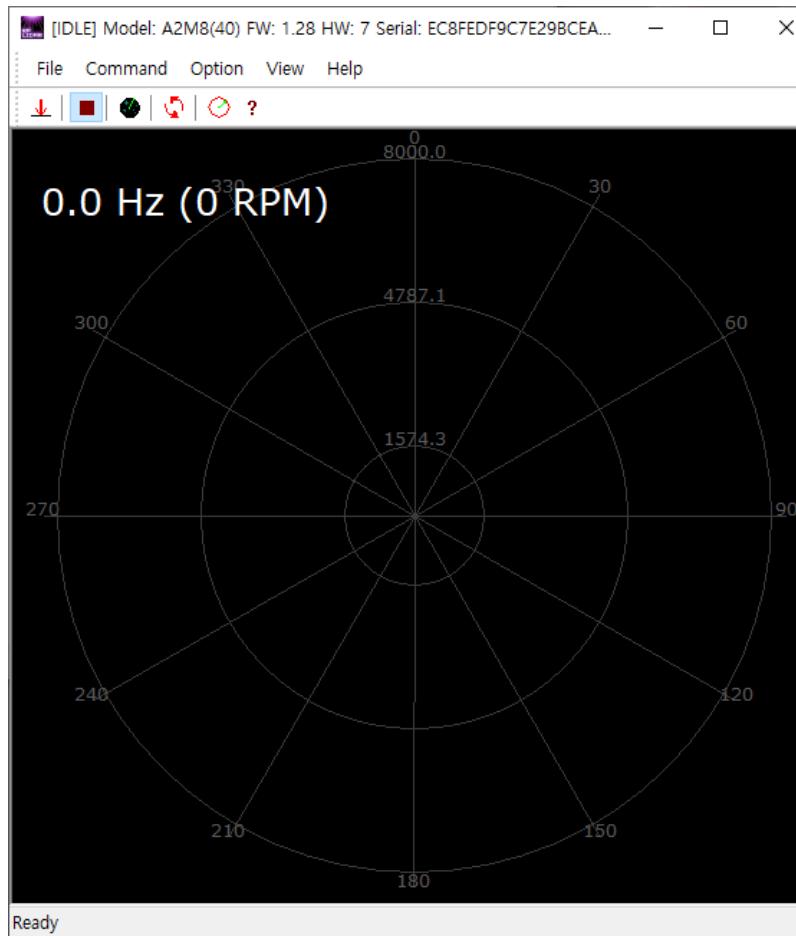
장치 관리자 – 포트(COM & LPT)
CP210x USB Port 번호 확인



Exercise 1

■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행

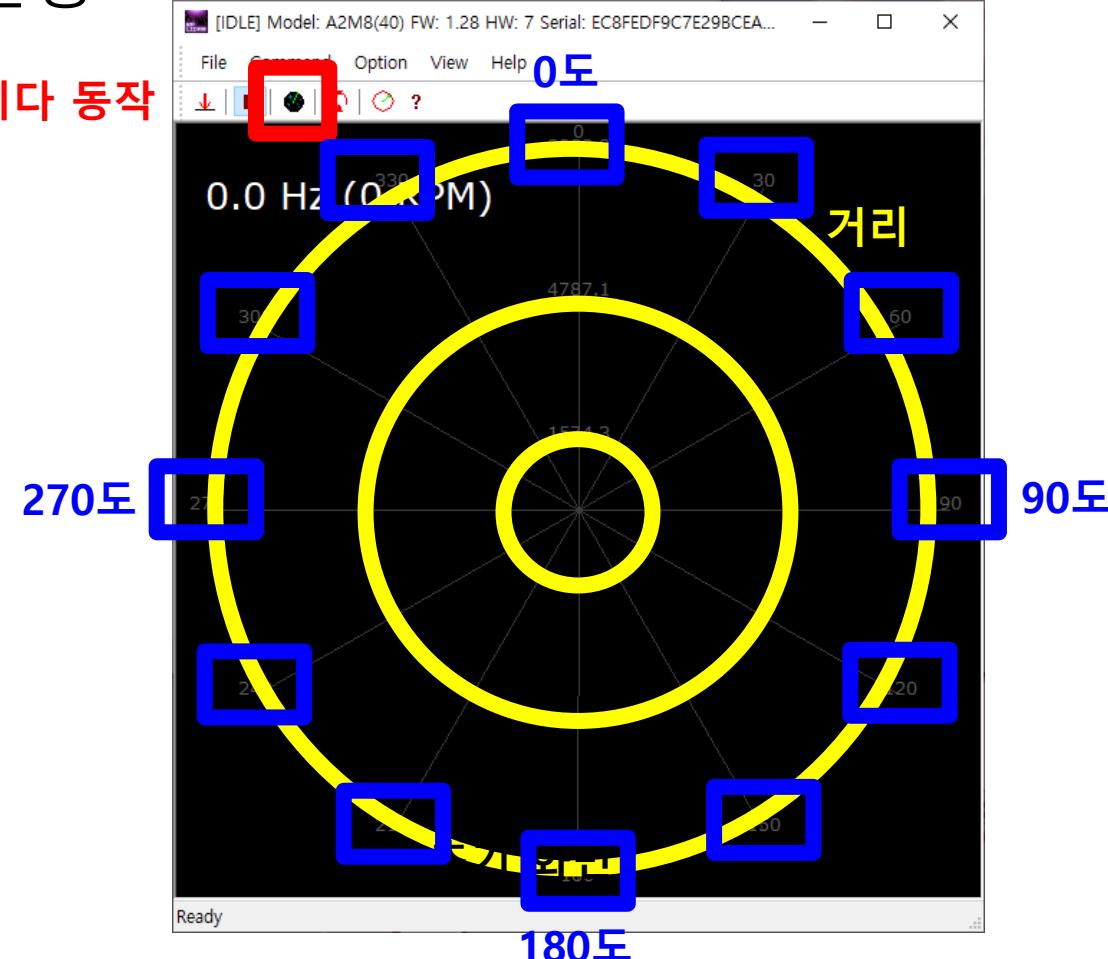


Exercise 1

■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행

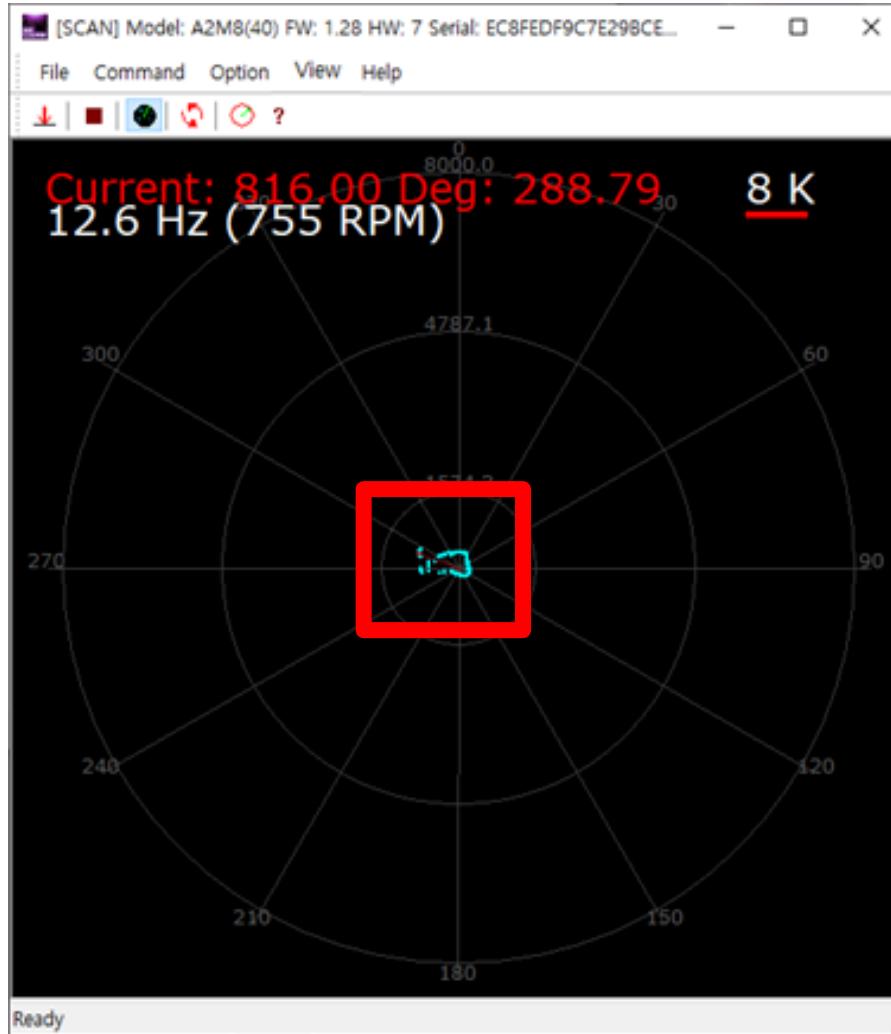
클릭 시, 라이다 동작



Exercise 1

■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행



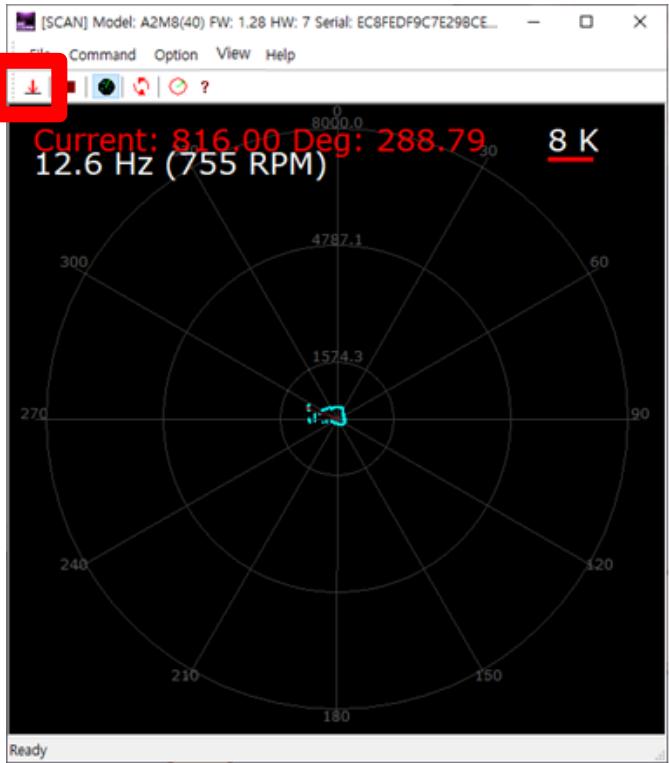
라이더 동작 시,
다음과 같이 물체 감지

Exercise 1

■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행

Dump data



LiDAR 감지 정보 저장

The figure shows a Windows Notepad window titled 'LiDAR DATA - Windows 메모장'. It contains the following data:

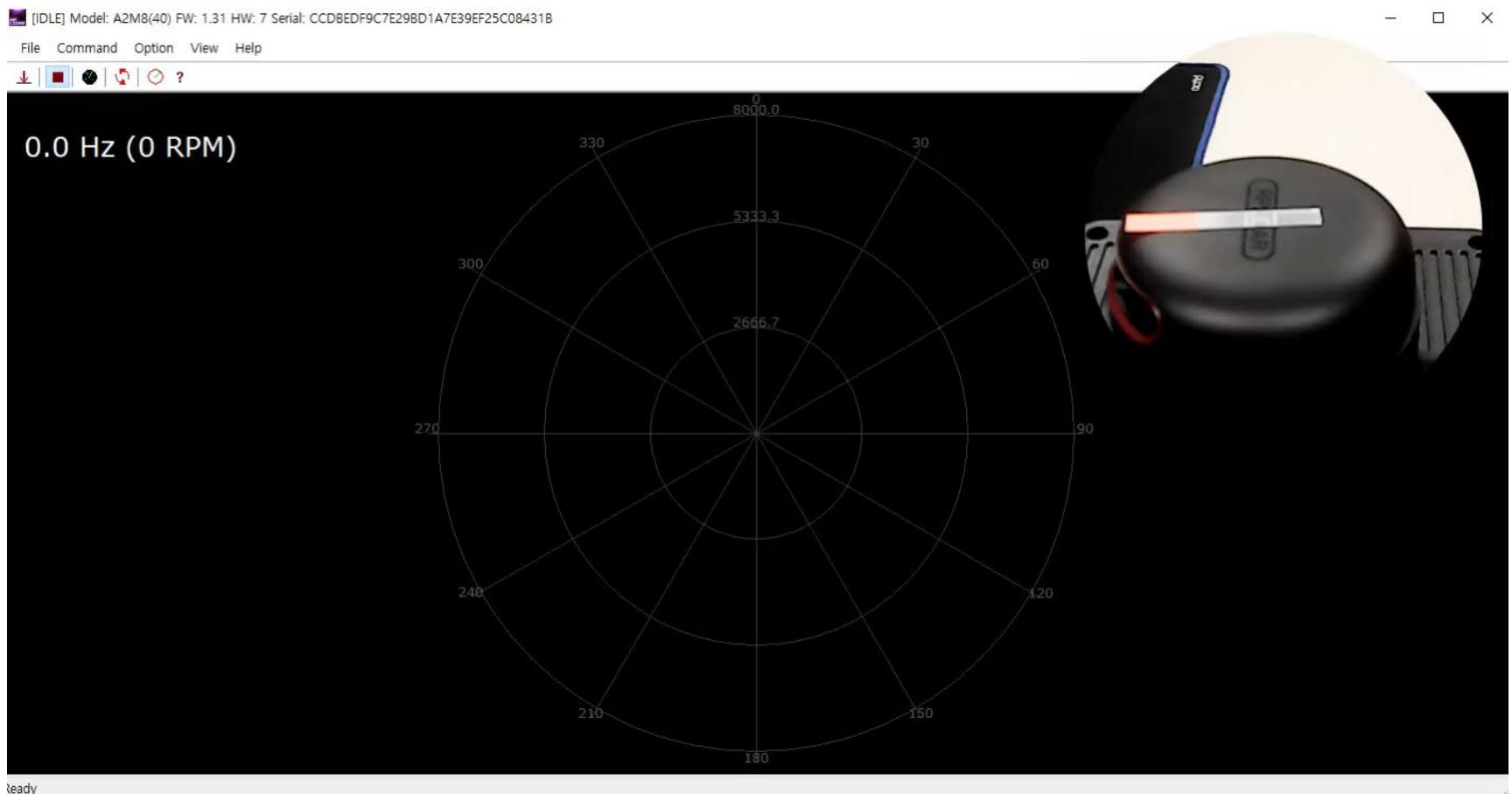
Angle	Distance	Quality
356.2921	306.0	188
356.7316	306.0	188
357.1710	306.0	188
357.6050	307.0	188
358.0444	307.0	188
358.4674	307.0	188
358.9014	307.0	188
359.3408	307.0	188
359.7803	307.0	188
0.2142	307.0	188
0.6372	307.0	188
1.0767	307.0	188
1.4502	308.0	188
1.8732	308.0	188
2.3071	308.0	188

0도 ~ 365도에 따른 거리 정보 확인 가능

Exercise 1

■ LiDAR 응용프로그램 연동하기

- 결과 영상



Exercise 2

■ LiDAR 기본 함수 사용

```
import Lib_LiDAR as LiDAR

if __name__ == "__main__":
    env = LiDAR.libLidar('COM11')
    env.init()

    env.getState()
    |
    count = 0

    for scan in env.scanning():
        count += 1
        print('%d: Got %d measurements' % (count, len(scan)))
        if count == 100:
            env.stop()
            break
```



Exercise 2

■ LiDAR 기본 함수 사용

```
import Lib_LiDAR as LiDAR

if __name__ == "__main__":
    env = LiDAR.libLidar('COM11')
    env.init()

    env.getState()
    |
    count = 0

    for scan in env.scanning():
        count += 1
        print('%d: Got %d measurements' % (count, len(scan)))
        if count == 100:
            env.stop()
            break
```

LiDAR 초기화

{'model': 40,

'firmware': (1, 28),

'hardware': 7,

'serialnumber': 'ECE8

Field Name	Description	Examples / Notes
model	RPLIDAR model ID	The model ID of the RPLIDAR being used
firmware_minor	Firmware version number, the minor value part	The decimal part of the version number
firmware_major	Firmware version number, the major value part	The integer part of the version number
hardware	Hardware version number	
serialnumber[16]	128bit unique serial number	When converting to text in hex, the Least Significant Byte prints first

Exercise 2

■ LiDAR 기본 함수 사용

```
import Lib_LiDAR as LiDAR

if __name__ == "__main__":
    env = LiDAR.libLidar('COM11')
    env.init()

    env.getState()
    |
    count = 0

    for scan in env.scanning():
        count += 1
        print('%d: Got %d measurements' % (count, len(scan)))
        if count == 100:
            env.stop()
            break
```

LiDAR 상태 확인

('Good' 0)

Field Name	Description	Examples / Notes
status	RPLIDAR State	Value definition : 0: Good 1: Warning 2: Error When the core system detects some potential risk that may cause hardware failure in the future, the status value will be set to Warning(1). But RPLIDAR can still work as normal. When RPLIDAR is in the Protection Stop state, the status value is set to Error(2).
error_code	The related error code that caused a warning/error.	

Exercise 2

■ LiDAR 기본 함수 사용

```
import Lib_LiDAR as LiDAR

if __name__ == "__main__":
    env = LiDAR.libLidar('COM11')
    env.init()

    env.getState()
    |
    count = 0

    for scan in env.scanning():
        count += 1
        print('%d: Got %d measurments' % (count, len(scan)))
        if count == 100:
            env.stop()
            break
```

0: Got 135 measurments
1: Got 135 measurments
2: Got 123 measurments
3: Got 117 measurments
4: Got 112 measurments
5: Got 110 measurments
6: Got 98 measurments
7: Got 93 measurments
8: Got 87 measurments
9: Got 89 measurments
10: Got 96 measurments
11: Got 93 measurments

LiDAR 종료

Exercise 2

■ LiDAR 기본 함수 사용

1) LiDAR 데이터 읽기

```
def scanning(self):
```

2) 특정 Angle 범위내 데이터만 출력

```
def getAngleRange(self, scan, minAngle, maxAngle):
```

3) 특정 Distance 범위내 데이터만 출력

```
def getDistanceRange(self, scan, minDist, maxDist):
```

4) 특정 Angle과 Distance 범위내 데이터만 출력

```
def getAngleDistanceRange(self, scan, minAngle, maxAngle, minDist, maxDist):
```

5) LiDAR Motor의 RPM 변경 및 확인

```
def setRPM(self, rpm):
```

```
def getRPM(self):
```

6) LiDAR 종료

```
def stop(self):
```

7) LiDAR 초기화

```
def init(self):
```

8) LiDAR 상태 확인

```
def getState(self):
```



Exercise 2

■ LiDAR 기본 함수 사용

1) 특정 Angle 범위내 데이터만 출력

```
def getAngleRange(self, scan, minAngle, maxAngle):
```

- Scan

- scanning() 함수를 통해 얻은 결과 데이터들

- minAngle

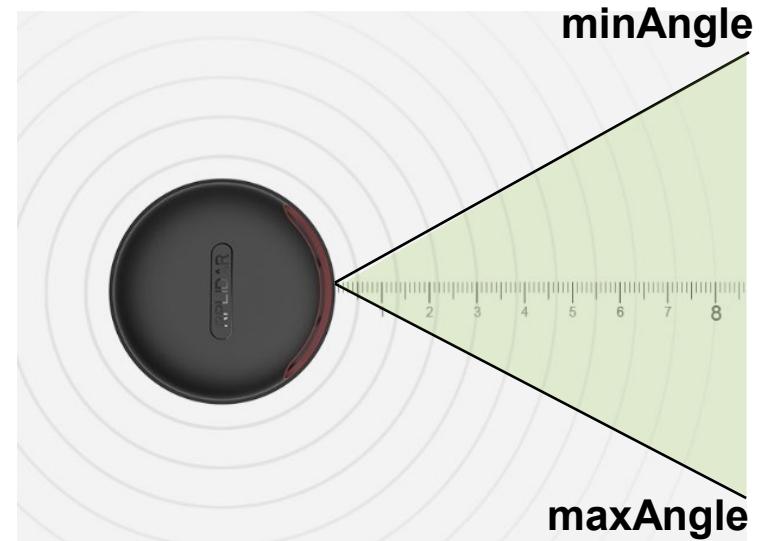
- 검색할 각도의 최소 값(0 이상)

- maxAngle

- 검색할 각도의 최대 값(360 이하)

- Return: List 형태의 검색 결과

- 입력 받은 데이터 중, 설정한 조건에 만족하는 데이터만 출력



Exercise 2

■ LiDAR 기본 함수 사용

2) 특정 Distance 범위내 데이터만 출력

```
def getDistanceRange(self, scan, minDist, maxDist):
```

- Scan

scanning() 함수를 통해 얻은 결과 데이터들

- minDist

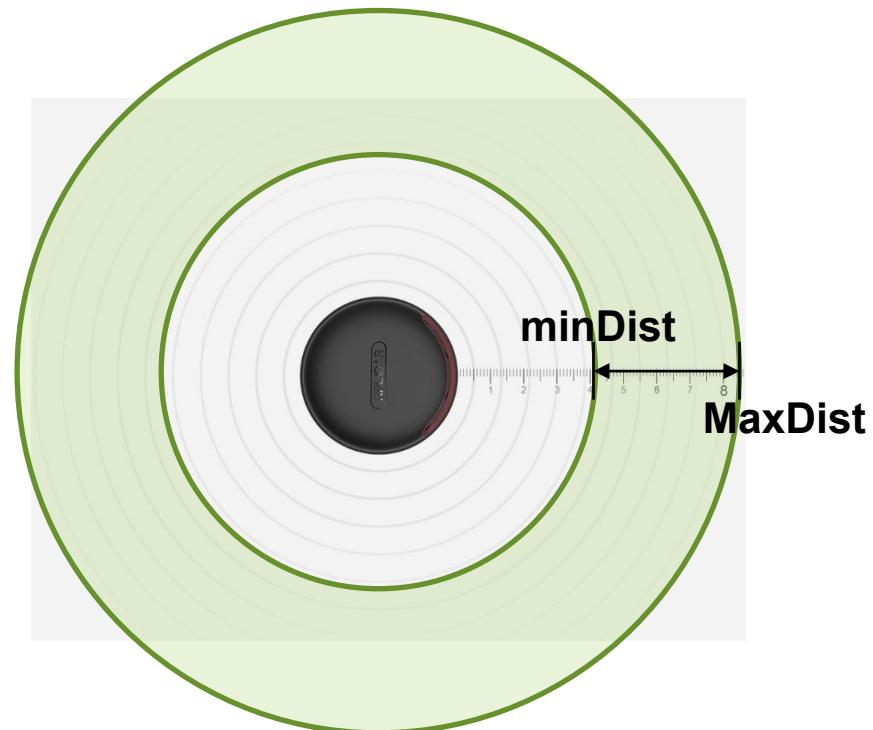
- 검색할 거리의 최소 값(150 이상)

- maxDist

- 검색할 거리의 최대 값(600 이하)

- Return: List 형태의 검색 결과

- 입력 받은 데이터 중, 설정한 조건에 만족하는 데이터만 출력



Exercise 2

■ LiDAR 기본 함수 사용

3) 특정 Angle과 Distance 범위내 데이터만 출력

```
def getAngleDistanceRange(self, scan, minAngle, maxAngle, minDist, maxDist):
```

- Scan

- scanning() 함수를 통해 얻은 결과 데이터들

- minAngle

- 검색할 각도의 최소 값(0 이상)

- maxAngle

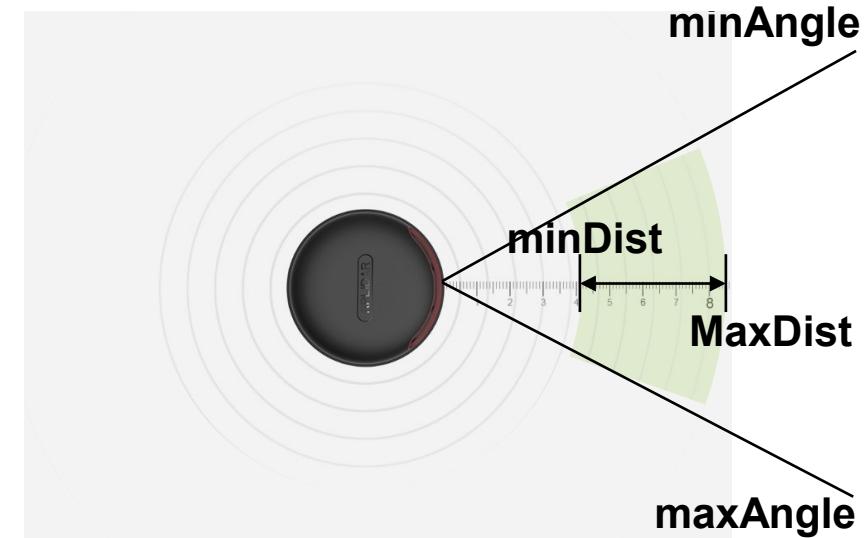
- 검색할 각도의 최대 값(360 이하)

- minDist

- 검색할 거리의 최소 값(150 이상)

- maxDist

- 검색할 거리의 최대 값(600 이하)



- Return: List 형태의 검색 결과

- 입력 받은 데이터 중, 설정한 조건에 만족하는 데이터만 출력

Exercise 2

■ LiDAR 기본 함수 사용

4) LiDAR Motor의 RPM 변경 및 확인

```
def setRPM(self, rpm):
```

- **rpm**

- 라이다 모터의 회전 속도
- 최소값 0, 최대값 1023
- Default : 660

- **Return:** 없음

```
def getRPM(self):
```

- **Return:** rpm(Int)

- 현재 설정된 rpm 값을 반환



Exercise 2

■ LiDAR 기본 함수 사용

5) LiDAR 종료

```
def stop(self):
```

- 함수 실행 시, LiDAR 동작 종료 후, 연결 해제
- **Return:** 없음

Exercise 2-1

■ LiDAR 기본 함수 사용

- 2-1 : 180도 ~ 210도 내의 LiDAR 정보만 출력



Exercise 2-1

```
# LiDAR Lib
import Lib_LiDAR as LiDAR

if __name__ == "__main__":
    env = LiDAR.LibLidar('COM11')
    env.init()
    count = 0

    for scan in env.scanning():
        count += 1
        scan = env.getAngleRange(scan, 180, 210)
        print(scan)
        if count == 100:
            env.stop()
            break
```



Exercise 2-2

■ LiDAR 기본 함수 사용

- 2-2 : 150mm ~ 300mm 내의 LiDAR 정보만 출력



Exercise 2-2



The image shows a screenshot of PyCharm IDE running a Python script to interface with a LiDAR sensor. The project structure on the left includes files like Lib_LIDAR.py, LIDAR_Exercise_2_1.py, LIDAR_Exercise_2_2.py (which is selected), LIDAR_Exercise_2_4.py, LIDAR_Exercise_2_5.py, and main.py. The Run tab at the bottom displays the output of the script, showing a series of coordinates:

```
if (__name__ == "__main__") > for scan in env.scanning():
    l[320.015825 250.
      [328.75 245.5 ]
      [331.4375 239.5 ]
      [333.671875 237.5 ]
      [336.90625 237.75 ]
      [339.140625 237.25 ]
      [342.375 235.75 ]
      [345.109375 226.25 ]
      [347.796875 224.25 ]
      [350.4375 224. ]
      [353.09375 228.25 ]]
```

Process finished with exit code 0

At the bottom, the status bar shows: Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console, Services, Run selected configuration, 12:54 CRLF UTF-8 4 spaces Python 3.9.

Exercise 2-3

■ LiDAR 기본 함수 사용

- 2-3 : 330도 ~ 350도 내의 LiDAR 정보 중에 200mm~ 250mm 내에 포함된 정보만 출력



Exercise 2-3



A screenshot of the PyCharm IDE interface. The title bar shows "pythonProject - LIDAR_Exercise_2_3.py". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The toolbar has icons for file operations. The Project tool window shows a "pythonProject" folder containing files: Lib_LIDAR.py, LIDAR_Exercise_2_1.py, LIDAR_Exercise_2_2.py, LIDAR_Exercise_2_3.py (selected), LIDAR_Exercise_2_4.py, LIDAR_Exercise_2_5.py, and main.py. The code editor shows the content of LIDAR_Exercise_2_3.py, which contains a list of empty brackets:

```
[  
]  
[]  
[]  
[]  
[]  
[]  
[]  
[]
```

 Below the code editor, the status bar says "Process finished with exit code 0". The bottom navigation bar includes Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console, Services, and a checkbox for "Run selected configuration". The bottom right corner shows the time as 6:15 and the Python version as 3.9.

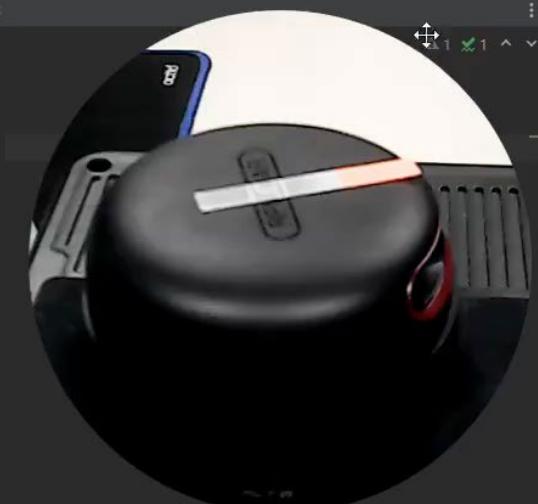
Exercise 2-4

■ LiDAR 기본 함수 사용

- 2-4 : 처음 시작은 660prm으로 LiDAR를 동작 시키고, 30번 동작 후에는 모터의 RPM을 1000rpm으로 변경



Exercise 2-4



The image shows a PyCharm IDE interface with a project titled "pythonProject". The project contains files: Lib_LiDAR.py, LiDAR_Exercise_2_4.py, LiDAR_Exercise_2_5.py, and main.py. The "LiDAR_Exercise_2_4.py" file is currently selected. In the bottom panel, a terminal window displays a traceback for a connection error:

```
Run: LiDAR_Exercise_2_4 ×
Traceback (most recent call last):
  File "C:\Users\Cloud\PycharmProjects\pythonProject\LiDAR_Exercise_2_4.py", line 5, in <module>
    env = LiDAR.libLidar('COM6')
  File "C:\Users\Cloud\PycharmProjects\pythonProject\Lib_LiDAR.py", line 7, in __init__
    self.lidar = RPLidar(port)
  File "C:\Users\Cloud\AppData\Local\Programs\Python\Python39\lib\site-packages\rplidar.py", line 135, in __init__
    self.connect()
  File "C:\Users\Cloud\AppData\Local\Programs\Python\Python39\lib\site-packages\rplidar.py", line 148, in connect
    raise RPLidarException('Failed to connect to the sensor')
rplidar.RPLidarException: Failed to connect to the sensor due to: could not open port 'COM6': FileNotFoundError(2, '지정된 파일을 찾을 수 없습니다.', None, 2)

Process finished with exit code 1
```

At the bottom of the terminal, it says "Packages installed successfully: Installed packages: 'pyserial' (today 오전 11:21)".

Exercise 2-5

■ LiDAR 기본 함수 사용

- 2-5 : 50번 LiDAR Scanning 작업 후, LiDAR 종료



Exercise 2-5



The image shows a black LIDAR sensor mounted on a circular base, with a smartphone placed above it. The sensor has a red laser beam pointing towards the phone.

pythonProject - LIDAR_Exercise_2_5.py

pythonProject > LIDAR_Exercise_2_5.py

File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - LIDAR_Exercise_2_5.py

Project pythonProject C:\Users\Cloud\PycharmProjects\pythonProject

- Lib_LIDAR.py
- LIDAR_Exercise_2_4.py
- LIDAR_Exercise_2_5.py
- main.py

External Libraries

Scratches and Consoles

main.py

LIDAR_Exercise_2_4.py

LIDAR_Exercise_2_5.py

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Run: LIDAR_Exercise_2_5

39: Got 109 measurements
40: Got 114 measurements
41: Got 114 measurements
42: Got 117 measurements
43: Got 109 measurements
44: Got 112 measurements
45: Got 116 measurements
46: Got 109 measurements
47: Got 106 measurements
48: Got 109 measurements
49: Got 118 measurements

Process finished with exit code 0

Version Control Run TODO Problems Python Packages Python Console Services

Packages installed successfully: Installed packages: 'pyserial' (today 오전 11:21)

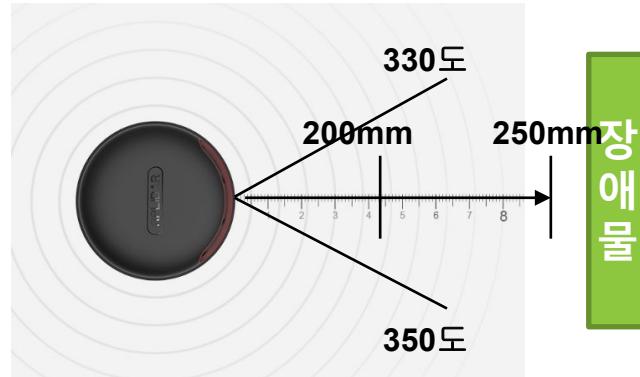
14:18 CRLF UTF-8 4 spaces Python 3.9

Exercise 3

■ LiDAR 물체 감지 후 동작

- 특정 각도 및 거리에서 물체가 감지되지 않은 경우 'Go'를 출력하고 물체가 감지되면 'Stop'을 출력하고 LiDAR 정지
 - 각도(330도~350도), 거리(200 ~ 250mm)

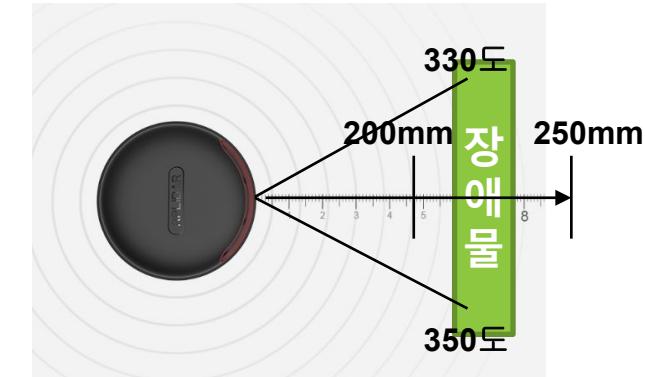
Go 출력



Go 출력



Stop 출력 -> LiDAR 종료



Exercise 3

The screenshot shows a PyCharm interface with the following details:

- File Menu:** PC, File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Title Bar:** pythonProject - LiDAR_Exercise_3.py
- Project Tree:** pythonProject C:\#Users\Cloud\PycharmProjects\pythonProject
Lib_LIDAR.py
LIDAR_Exercise_2_1.py
LIDAR_Exercise_2_2.py
LIDAR_Exercise_2_3.py
LIDAR_Exercise_2_4.py
LIDAR_Exercise_2_5.py
LIDAR_Exercise_3.py
main.py
- Code Editor:** Shows the content of LIDAR_Exercise_3.py with line numbers 1 to 23. The file contains:

```
go
go
go
go
go
go
go
go
[[331.09375 245.      ]
 [334.953125 247.     ]]
stop
```
- Run Tool Window:** Shows the command "LiDAR_Exercise_3" and its output:

```
go
go
go
go
go
go
go
go
[[331.09375 245.      ]
 [334.953125 247.     ]]
stop

Process finished with exit code 0
```
- Bottom Status Bar:** Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console, Services.
Packages installed successfully: Installed packages: 'pyserial' (today 오전 11:21)
- Bottom Right:** Notifications icon, showing a notification about a package update.

Thank You!

Automation Lab.



Exercise 4

■ LiDAR 물체 각도 구하기

- Step 1 : 프로그램 실행



Exercise 5

■ LiDAR에서 물체 감지되면 시리얼로 명령 전달

- Step 1 : 프로그램 실행



Exercise 6

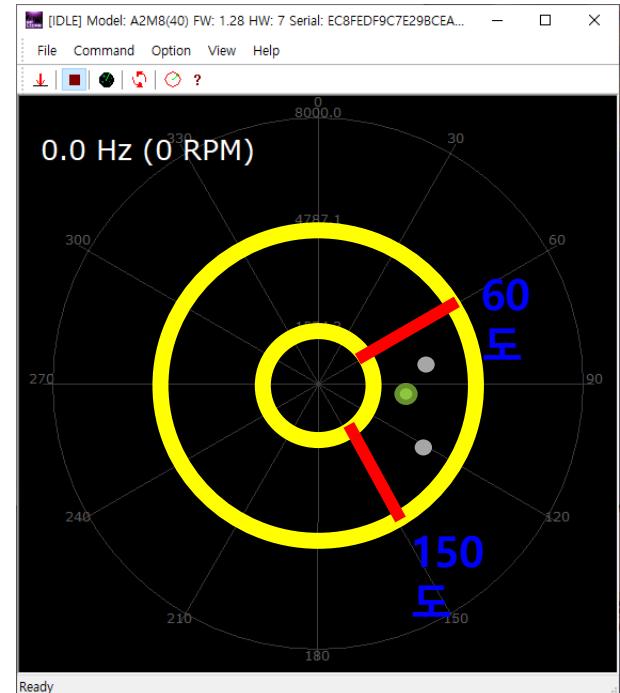
■ LiDAR에서 물체 감지되면 시리얼로 명령 전달

- Step 1 : 프로그램 실행



Exercise 7

- #1 LiDAR에서 특정한 거리 데이터 중에서 60-150도 사이에 있는 정보들 중에서 가장 가까운 위치의 값을 반환하는 함수 구현
- #2 가장 먼 거리의 값을 반환하는 함수 구현



Exercise 8

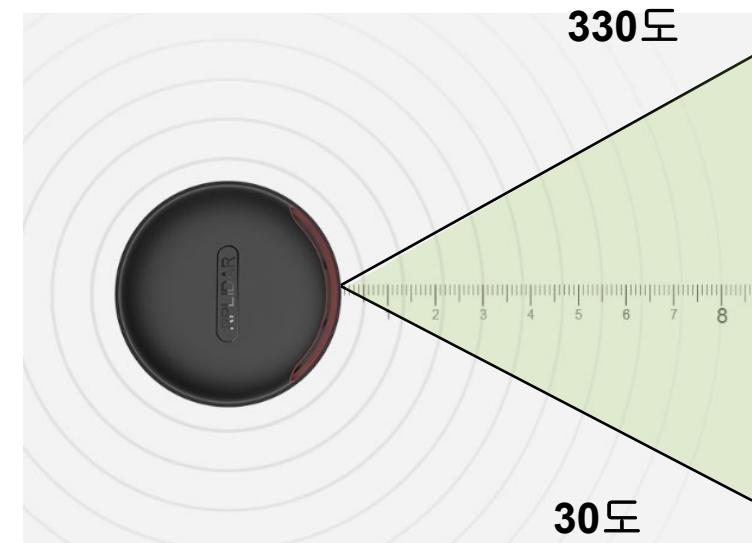
■ 물체추적 감시 시스템

1. 라이다의 위치를 앞뒤로 움직일 때, (310~360도) 에 있는 물체를 감지하여 현재 라이다가 앞쪽 또는 뒤로 이동 중인 상태를 표시한다.
2. 라이다가 앞으로 이동할 경우 → "Front" 출력
3. 라이다가 뒤로 이동할 경우 → "Rear" 출력
4. 라이다가 움직이지 않을 경우 → "Mid" 출력
(1mm 이내의 변화는 움직이지 않는 것으로 간주함)

Exercise 9

■ 검색 범위

1. GetAngleRange 함수를 변경하여 초록색 범위와 같이 330~360도, 0~30도 내의 물체를 감지하는 함수를 구현하시오.



The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and a project name "pythonProject - C:\Users\HHK\Desktop\CAR\lidar_test.py". The left sidebar displays the Project structure with files "lidar_test.py", "rplidar.py", and "C:\...\lidar_test.py". The main code editor window contains the following Python script:

```
from rplidar import RPLidar
lidar = RPLidar('COM5')

info = lidar.get_info()
print(info)

health = lidar.get_health()
print(health)

for i, scan in enumerate(lidar.iter_scans()):
    print('%d: Got %d measurements' % (i, len(scan)))
    print(scan)
    if i > 10:
        break

lidar.stop()
lidar.stop_motor()
lidar.disconnect()
```

The bottom run tab shows the output of the script:

```
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3012/3000. Cleaning buffer...
Too many bytes in the input buffer: 3017/3000. Cleaning buffer...
Too many bytes in the input buffer: 3017/3000. Cleaning buffer...
Too many bytes in the input buffer: 3017/3000. Cleaning buffer...

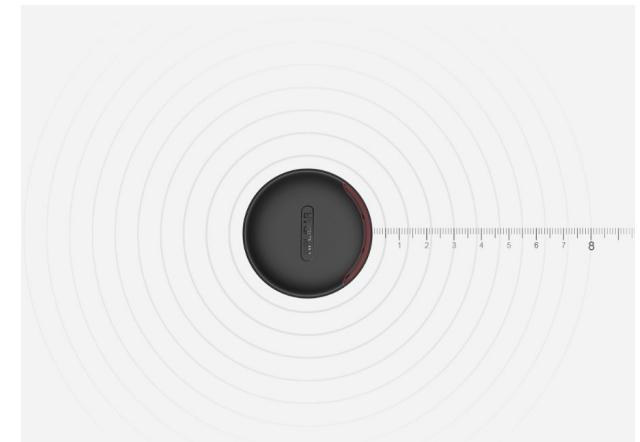
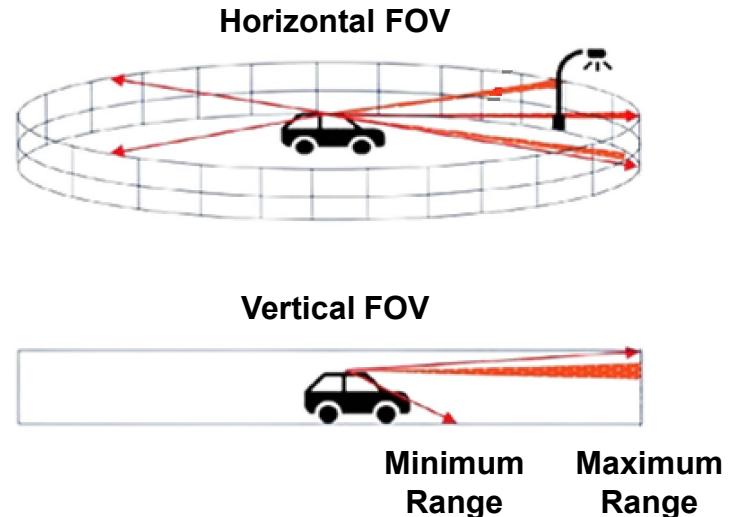
Process finished with exit code -1
```

USB 포트는 단독으로 연결

Introduction

■ LiDAR Main Spec

- FOV(Field of View)
 - 수신기의 측정 가능 각도
- 공간 분해능(Spatial Resolution)
 - 측정되는 각도 단위
- 거리
 - LiDAR가 측정할 수 있는 거리 값



Exercise 9

■ 라이다 스캔값 극좌표 형식으로 변환해서 그리기

