

# 미래형자동차 교육 워크숍

Subject : LiDAR

***Embedded System Lab.***

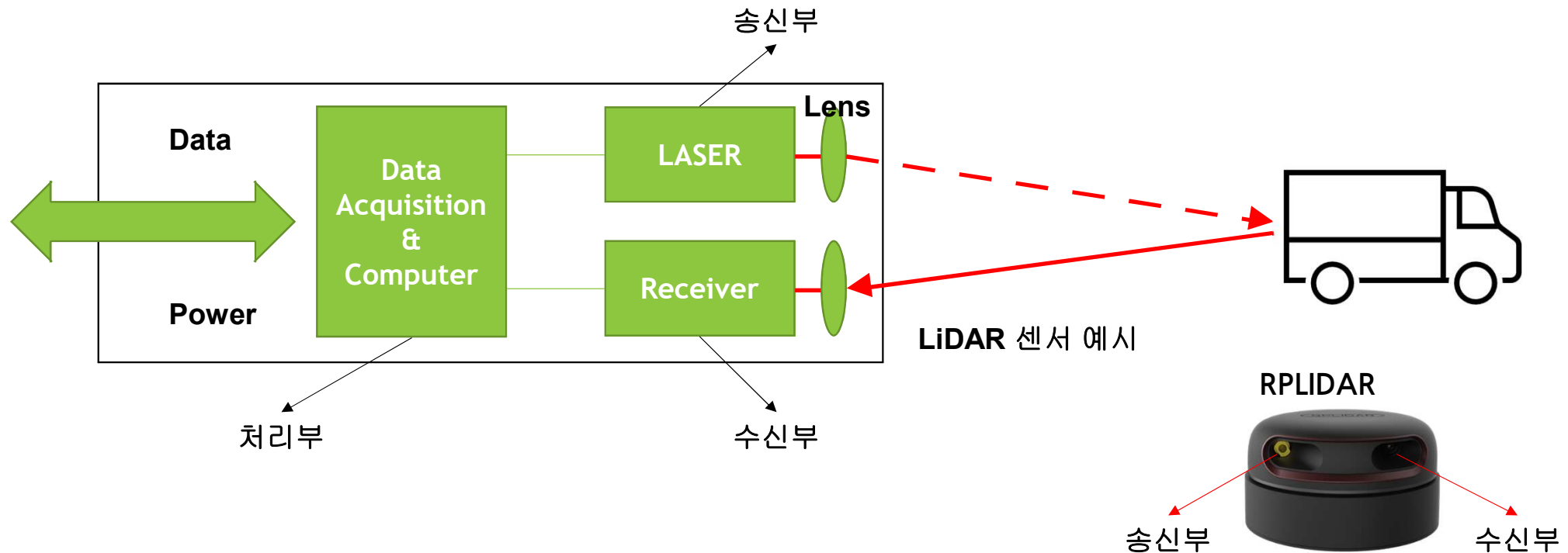


# 목차

- **INTRODUCTION**
- **EXERCISE**

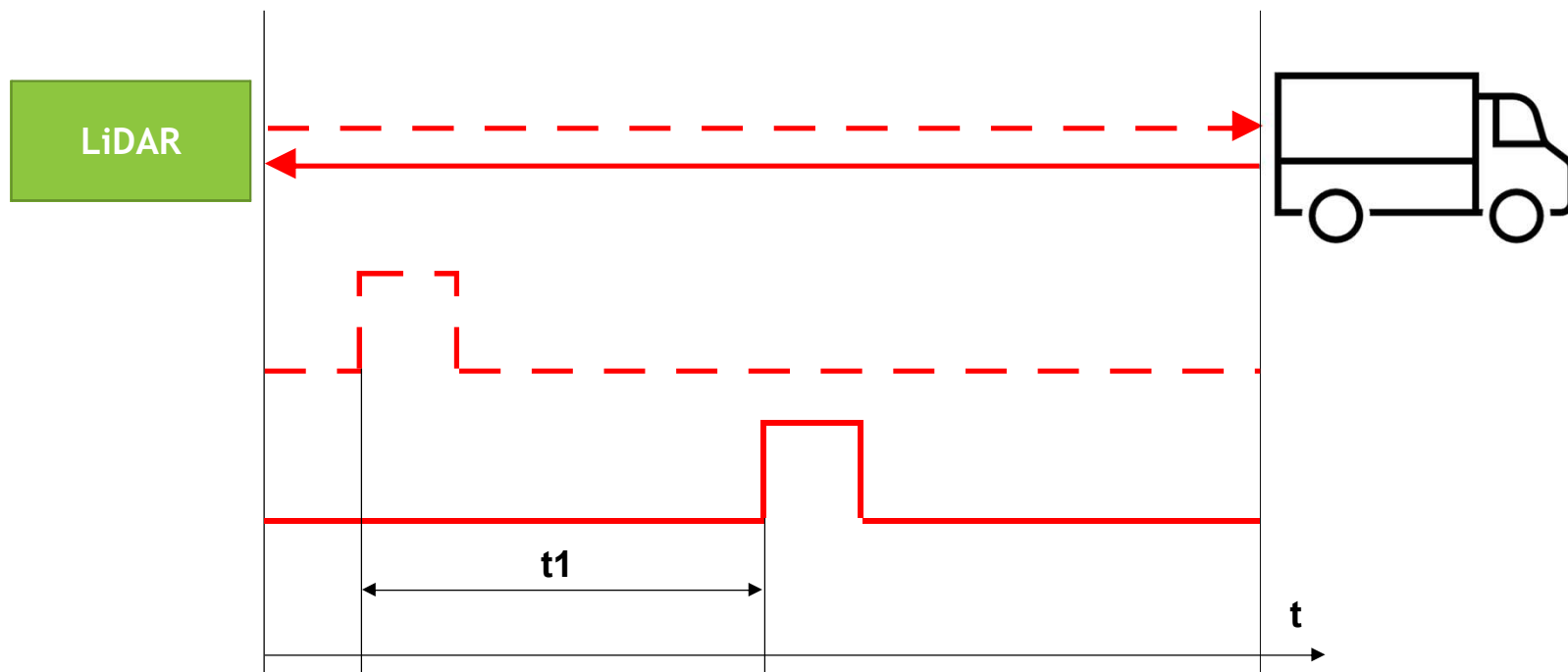
# Introduction

## ■ LiDAR(Light Detection And Ranging)



# Introduction

- Distance Measurement
  - Time-Of-Flight(TOF)



# Introduction

## ■ RPLIDAR-A1 (LiDAR)



- 제품 사양
  - 거리 범위 : ~12m
  - 각도 범위 : 0 ~ 360 degree
  - 각도 해상도 : 1 degree

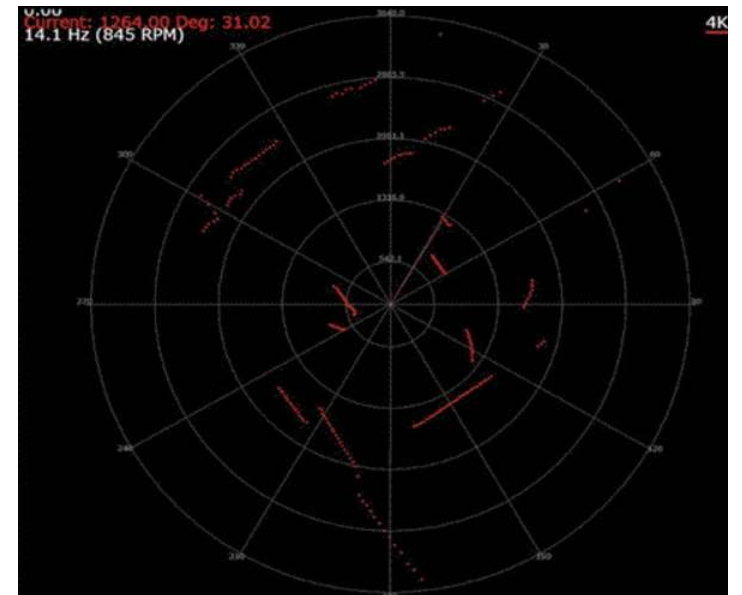
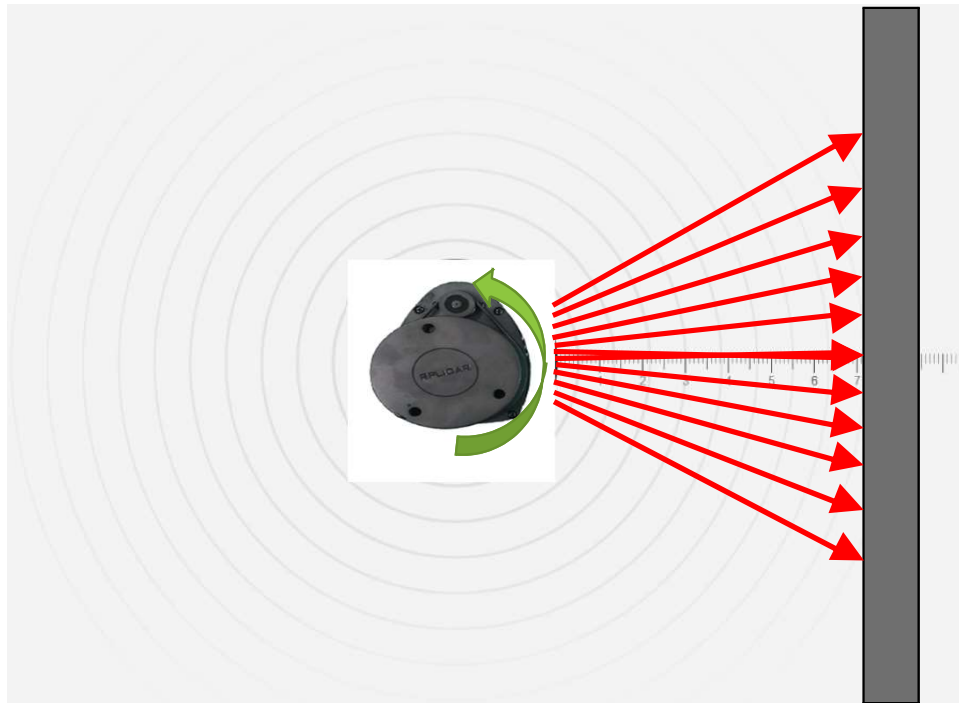
# Introduction

## ■ LiDAR Operating



# Introduction

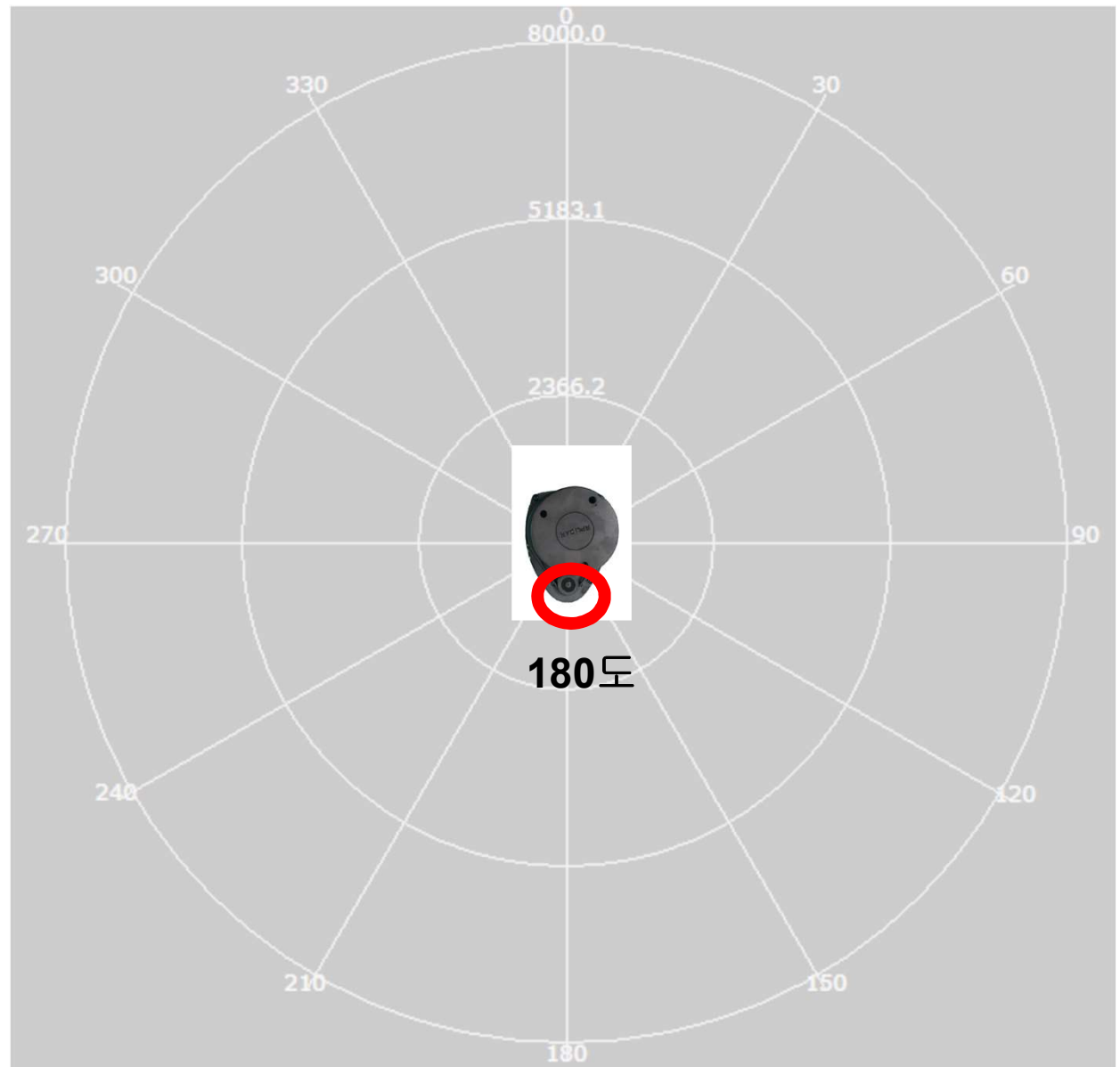
## ■ LiDAR Operating



# Introduction

## ■ LiDAR Operating

- 측정 각도





# 목차

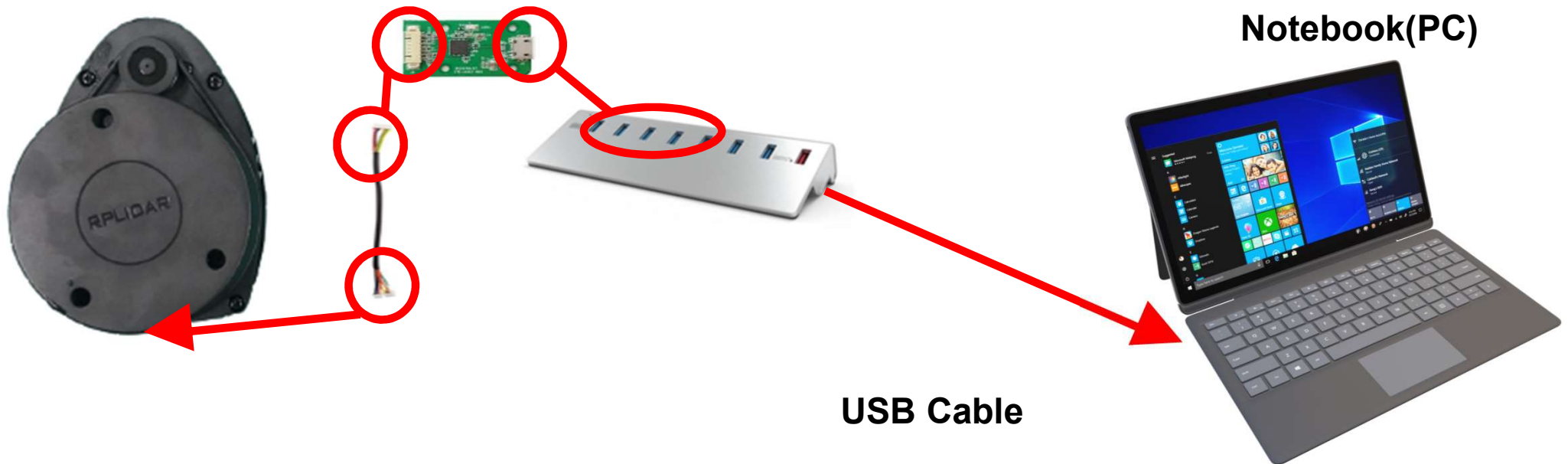
- INTRODUCTION
- **EXERCISE**

# Exercise 1

## ■ LiDAR 응용프로그램 연동하기

- Step 1 : Hardware 연결

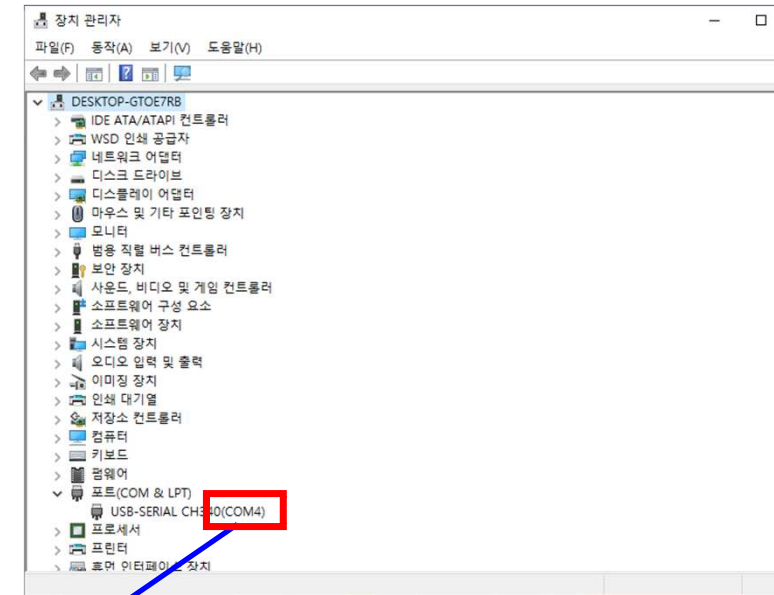
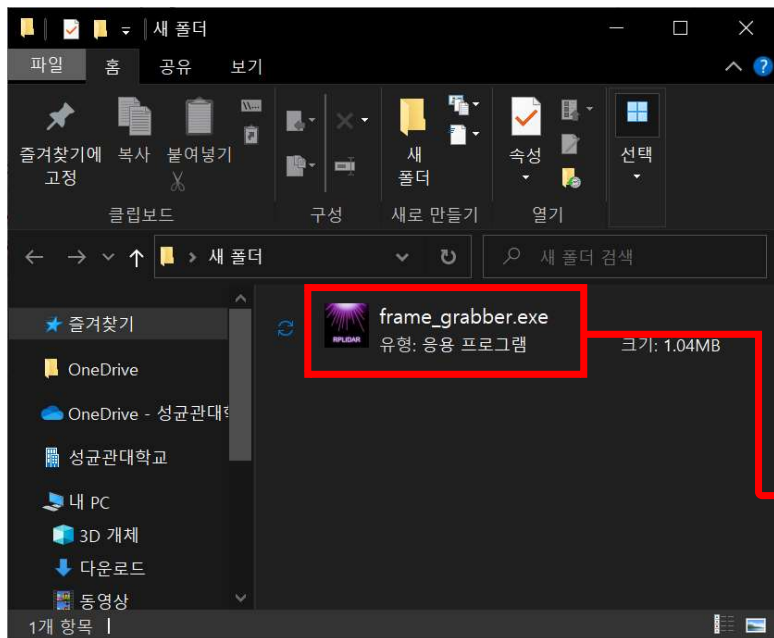
→ LiDAR의 전원 USB 케이블과 데이터 USB 케이블을 허브와 PC에 연결



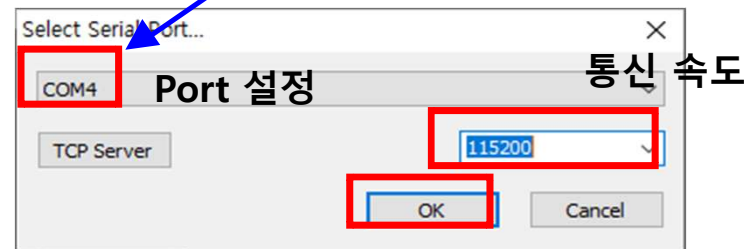
# Exercise 1

## ■ LiDAR 응용프로그램 연동하기

- Step 2 : 프로그램 실행 및 연결



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

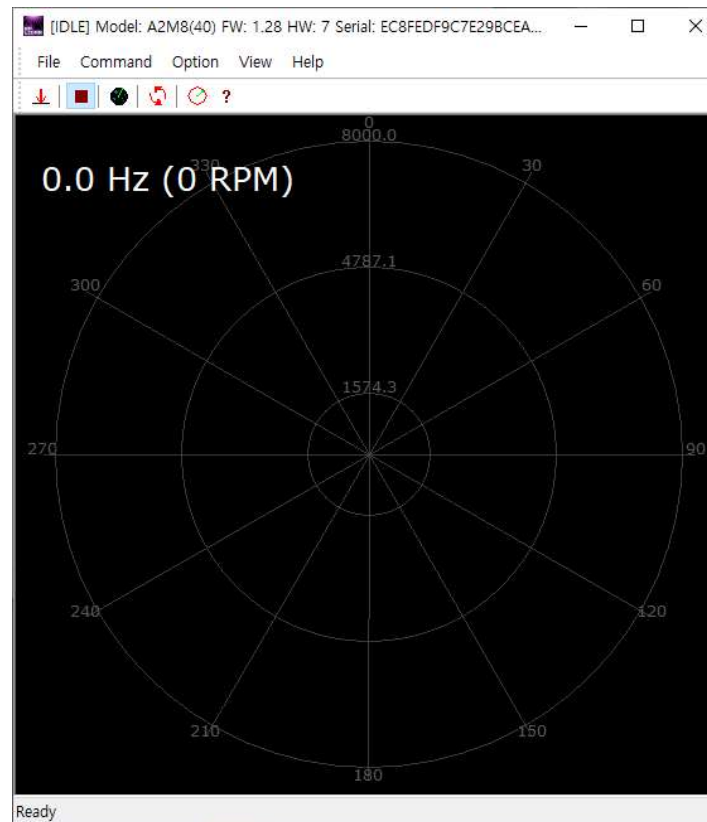


통신 속도

# Exercise 1

## ■ LiDAR 응용 프로그램 연동하기

- Step 3 : 프로그램 실행

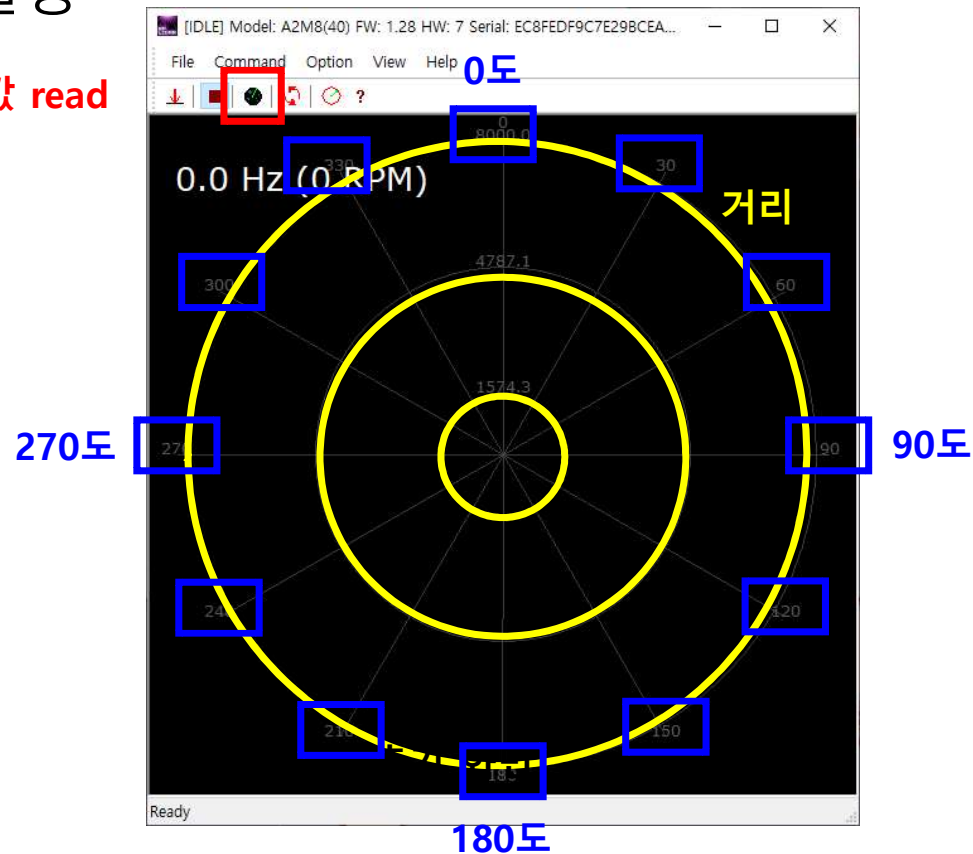


# Exercise 1

## ■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행

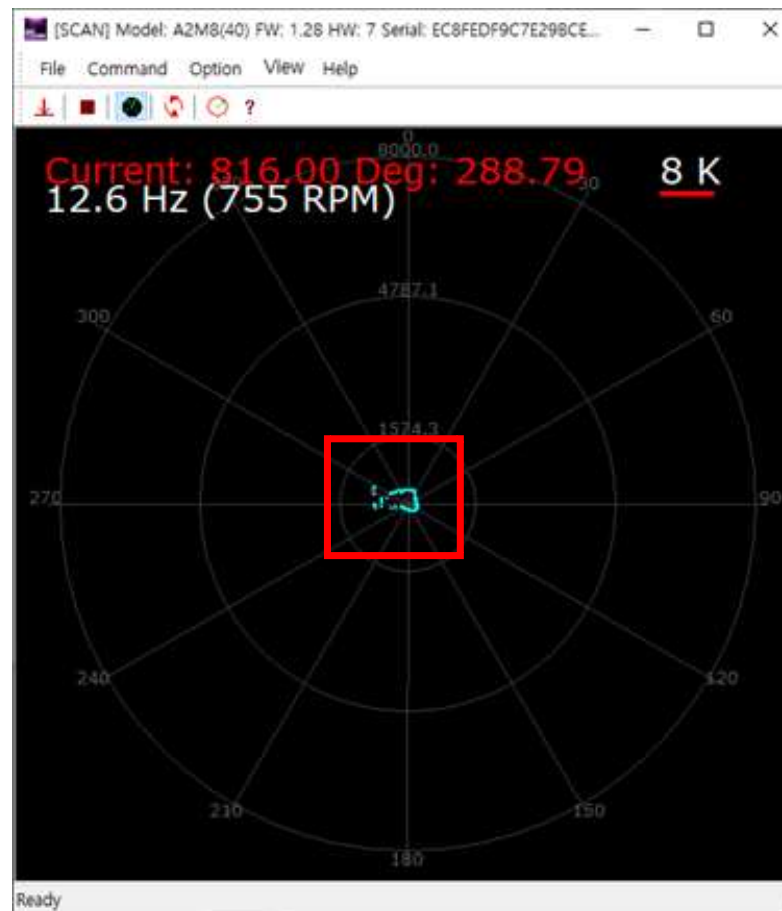
클릭 시, 라이다 값 read



# Exercise 1

## ■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행



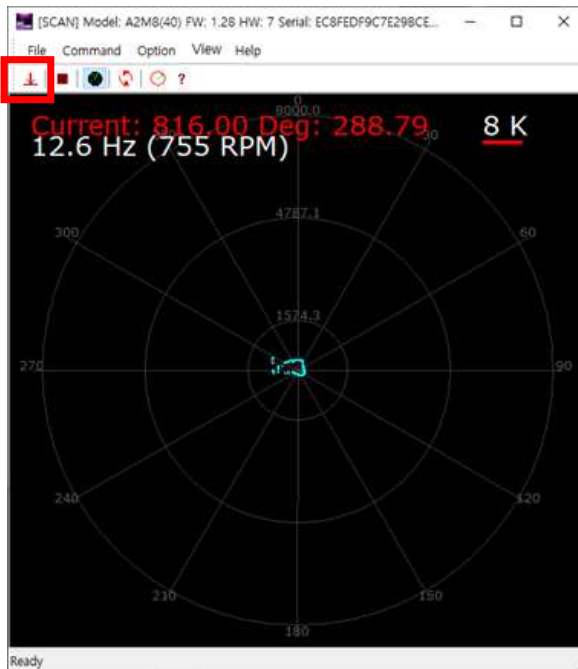
라이더 동작 시,  
다음과 같이 물체 감지

# Exercise 1

## ■ LiDAR 응용프로그램 연동하기

- Step 3 : 프로그램 실행

Dump data



LiDAR 감지 정보 저장

LiDAR DATA - Windows 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
#RPLIDAR SCAN DATA  
#COUNT=587  
#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  
Ln 1, Col 1 100% Windows (CRLF) UTF-8

0도 ~ 365도에 따른 거리 정보 확인 가능

# 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
```



# 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
```

LiDAR 초기화

`['model': 40, 'firmware': (1, 28), 'hardware': 7, 'serialnumber': 'EC8`

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 measurments' % (count, len(scan)))
        if count == 100:
            env.stop()
            break
```

LiDAR 상태 확인

('Good', 0)

Field Name	Description	Examples / Notes
status	RPLIDAR State	Health 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):
```

### 6) LiDAR 초기화

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

### 7) LiDAR 상태 확인

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

### 8) LiDAR 종료

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

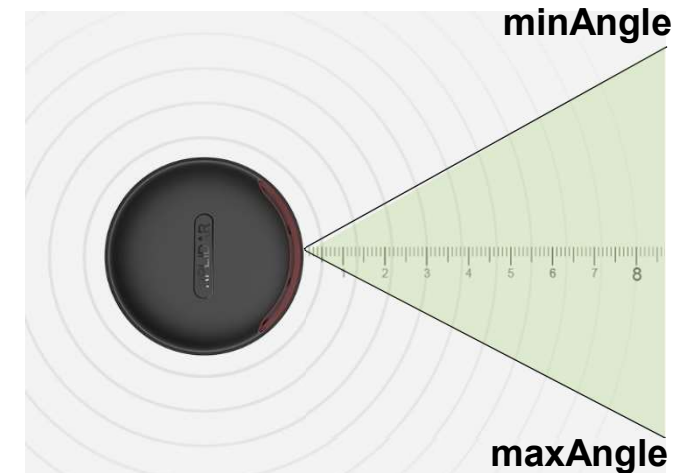
# Exercise 2

## ■ 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

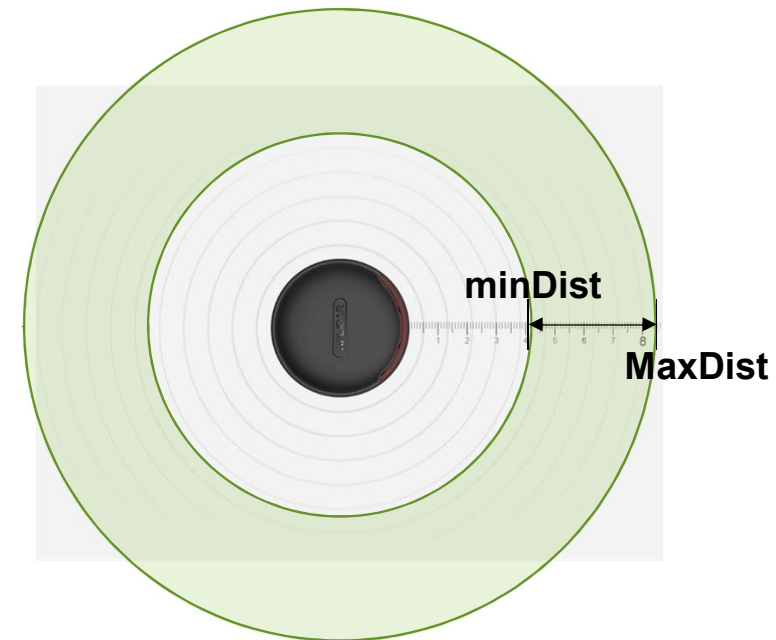
· 검색할 거리의 최소 값(150 이상)

#### - maxDist

· 검색할 거리의 최대 값(6000 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

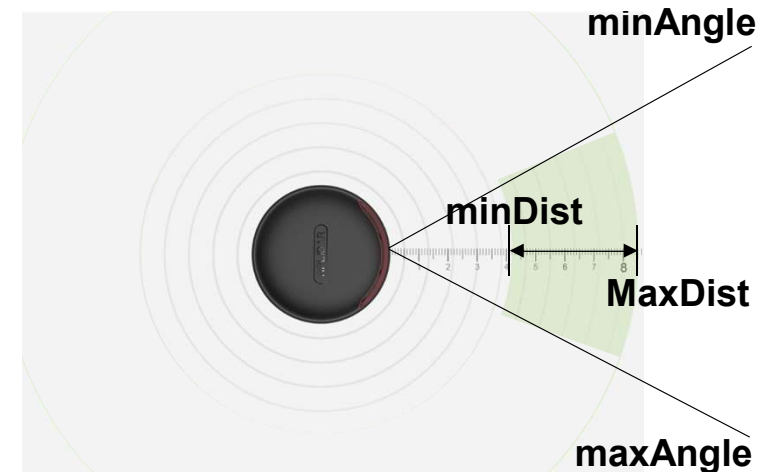
- 검색할 거리의 최소 값(150 이상)

#### - maxDist

- 검색할 거리의 최대 값(6000 이하)

#### - Return: List 형태의 검색 결과

- 입력 받은 데이터 중, 설정한 조건에 만족하는 데이터만 출력



# Exercise 2

## ■ LiDAR 기본 함수 사용

### 4) LiDAR 종료

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

- 함수 실행 시, 연결 해제.  
라이다가 회전하는 것을 멈추는 것이 아니라,  
프로그램 내부적으로 연결 해제하여 값을 읽어오는 것을 중지함. 라이다는 계속 회전함.

- **Return:** 없음

- 이 함수를 실행하지 않고 프로그램이 종료 된 후, 다시 프로그램을 실행하면 아래와 같은  
오류 발생하므로, 프로그램이 끝나기 이전에 이 함수를 실행해주세요.

```
raise RPLidarException('Incorrect descriptor starting bytes')  
rplidar.RPLidarException: Incorrect descriptor starting bytes
```



# Exercise 2-1

## ■ LiDAR 기본 함수 사용

- 2-1 : 180도 ~ 210도 내의 LiDAR 정보만 출력

# Exercise 2-1

```
# LiDAR Lib
import Function_Library as LiDAR

if (__name__ == "__main__"):
    env = LiDAR.libLiDAR('COM6')
    env.init()
    count = 0

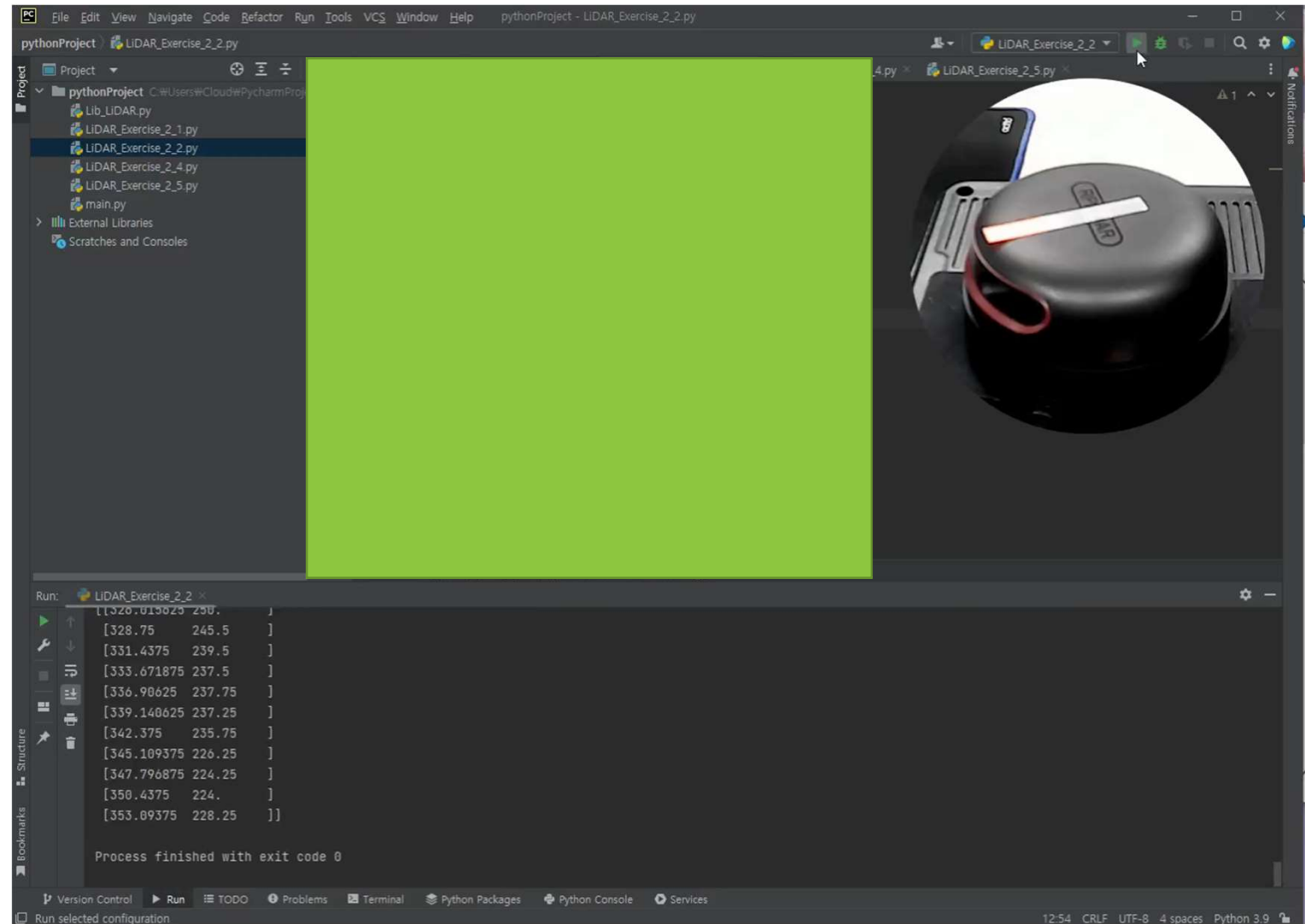
    for scan in env.scanning():
        count += 1
        scan = env.getAngleRange(scan, 180, 210)
        print(scan)
        if count == 100:
            env.stop()
            break
```

# Exercise 2-2

## ■ LiDAR 기본 함수 사용

- 2-2 : 150mm ~ 300mm 내의 LiDAR 정보만 출력

# Exercise 2-2

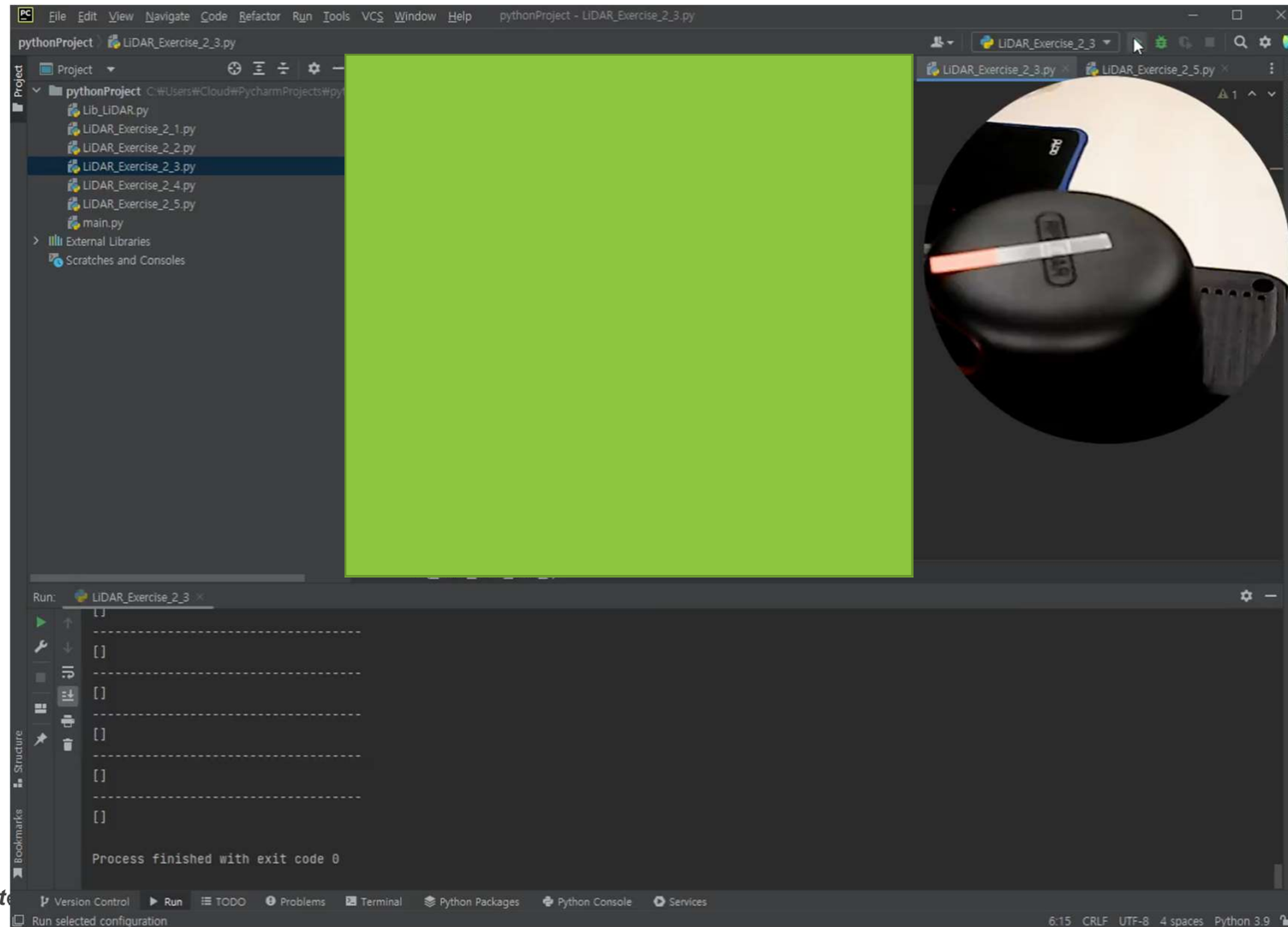


## Exercise 2-3

### ■ LiDAR 기본 함수 사용

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

# Exercise 2-3

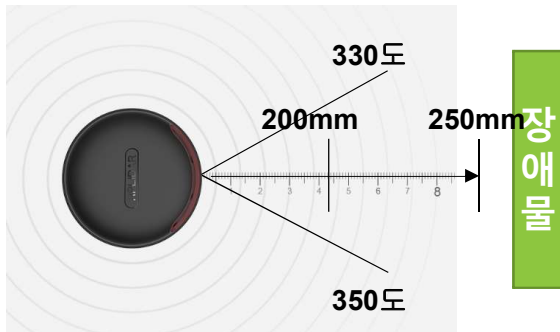


# Exercise 3

## ■ 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)

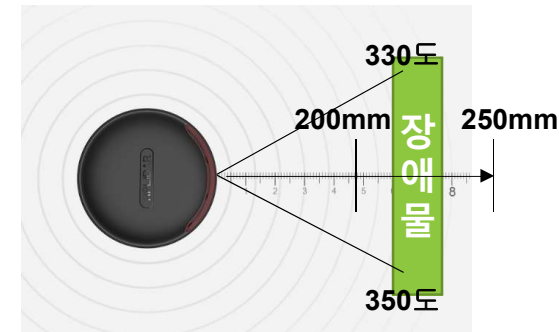
Go 출력



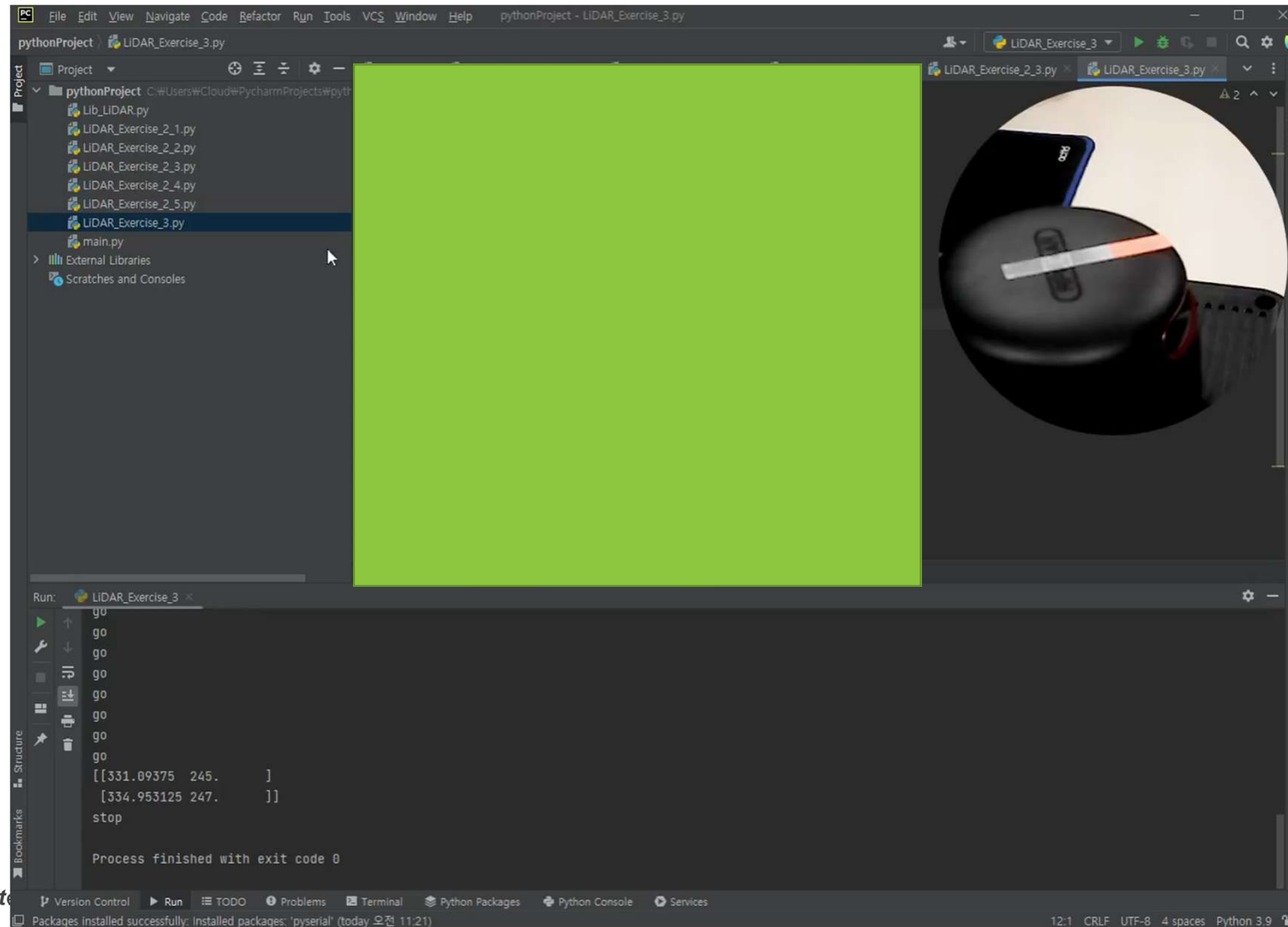
Go 출력



Stop 출력 -> LiDAR 종료



# Exercise 3





# Thank You!

***Embedded System Lab.***

