Future Vehicle Education Workshop

Subject : LiDAR

Automation Lab.

SUNGKYUNKWAN UNIVERSITY

1398

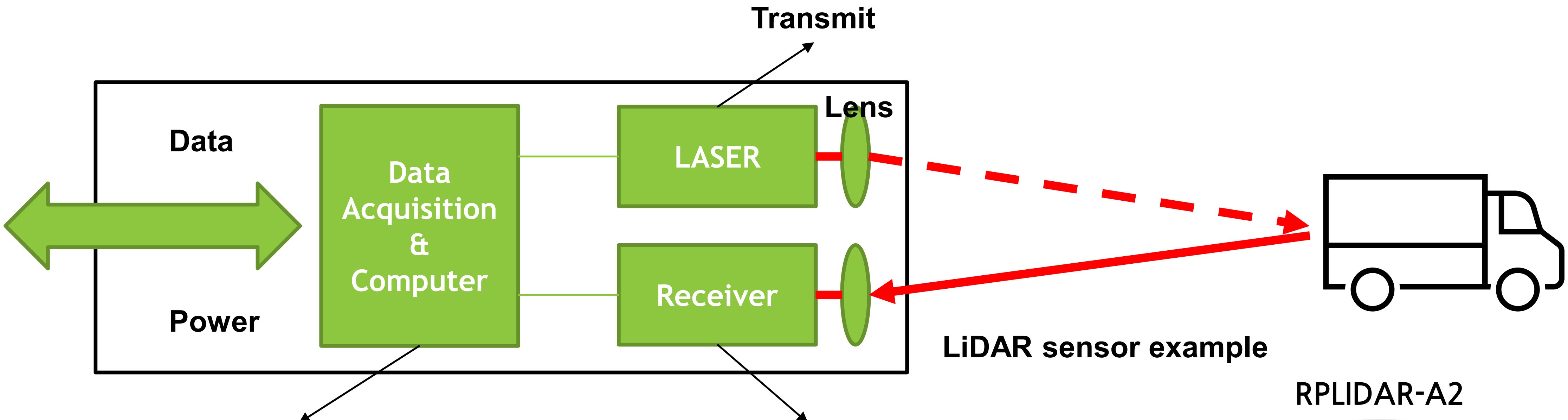
성균관대학교

Contents

- INTRODUCTION
- EXERCISE

Introduction

■ LiDAR(Light Detection And Ranging)



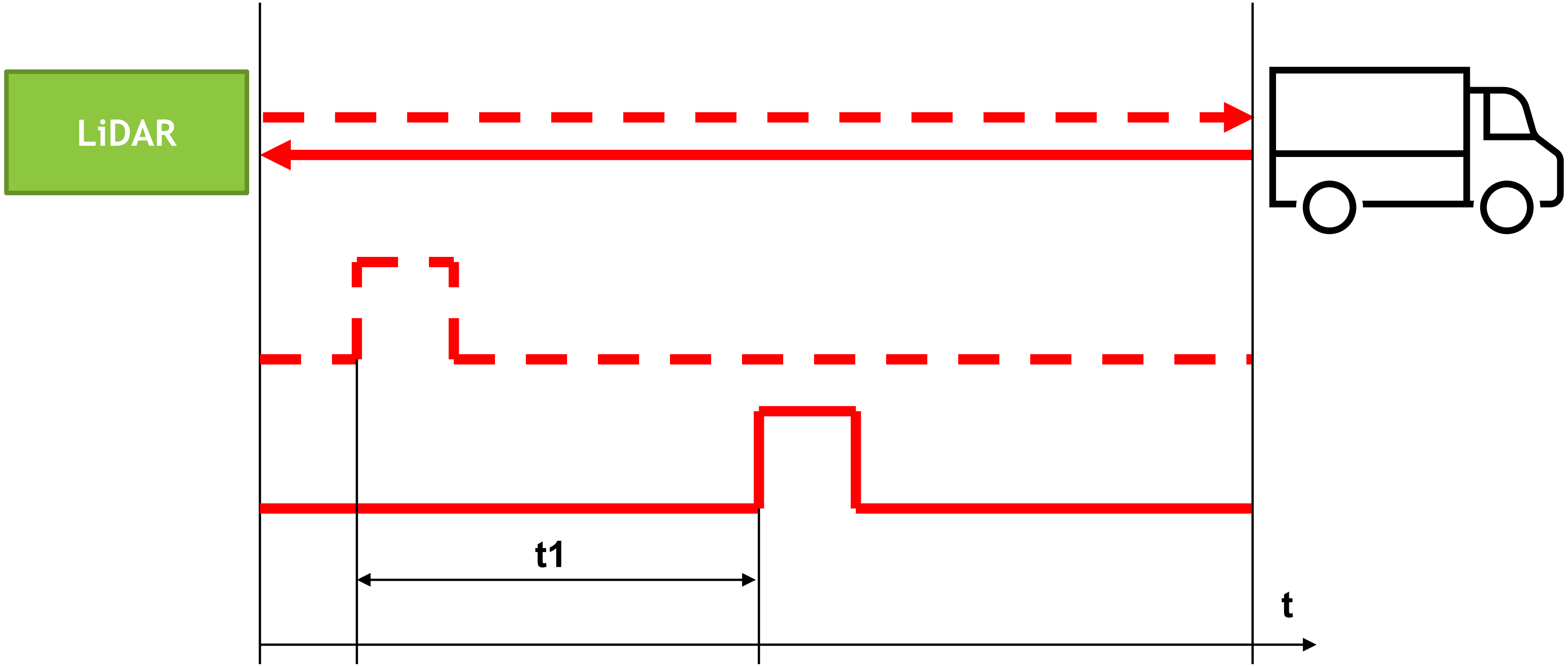
LiDAR sensor example

RPLIDAR-A2

Introduction

Distance Measurement

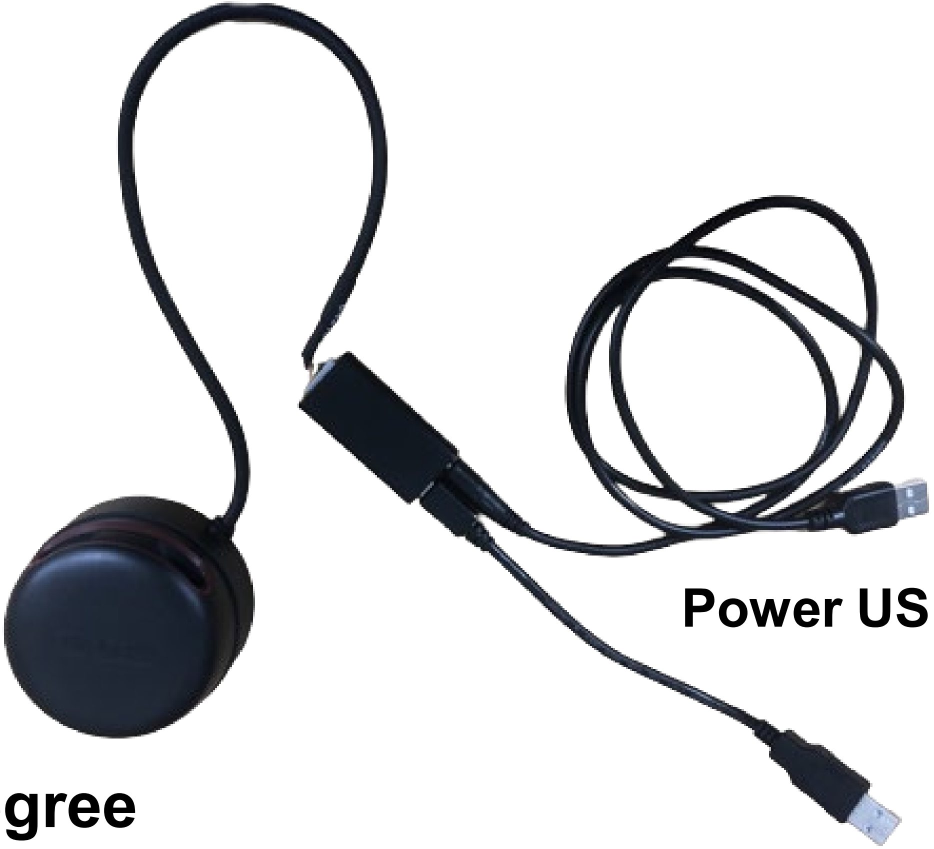
- Time-Of-Flight(TOF)



Introduction

■ RPLIDAR-A2(LiDAR)

- **Product Specification**
  - · Distance range: 0.15m ~ 6m
  - · Angular range: 0 ~ 360 degree
  - · Angular resolution: 0.45 ~ 0.9 degree

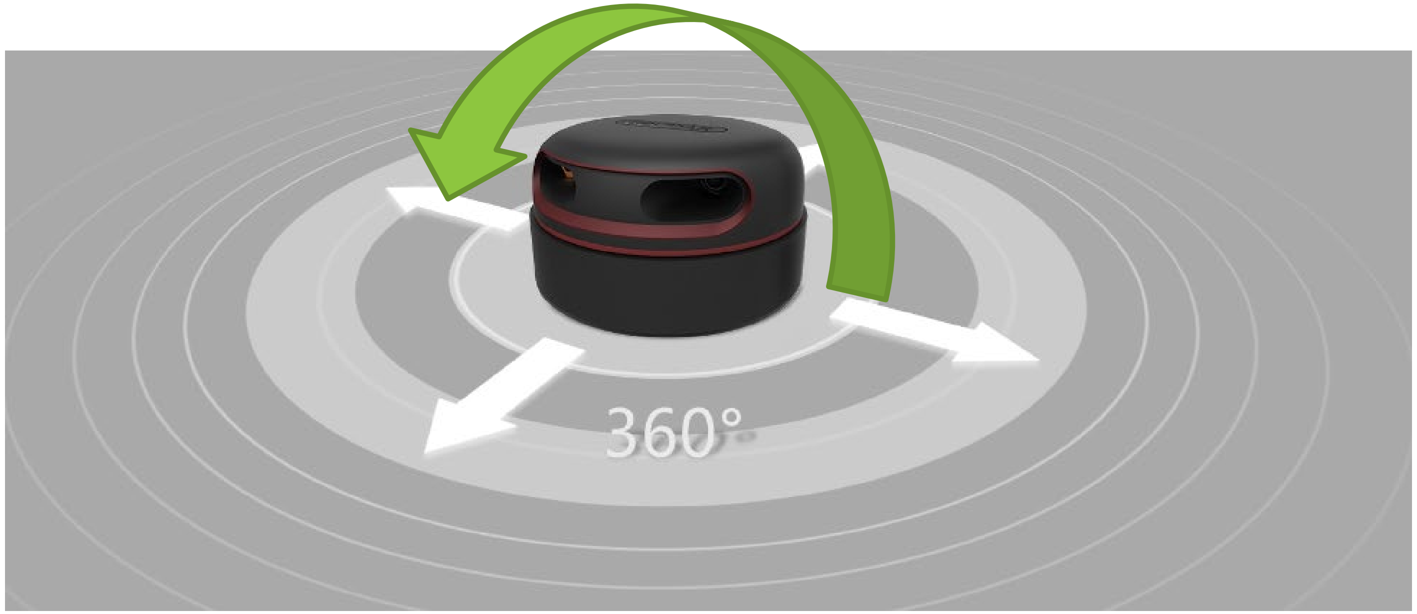


Power USB Cable

Data USB Cable

Introduction

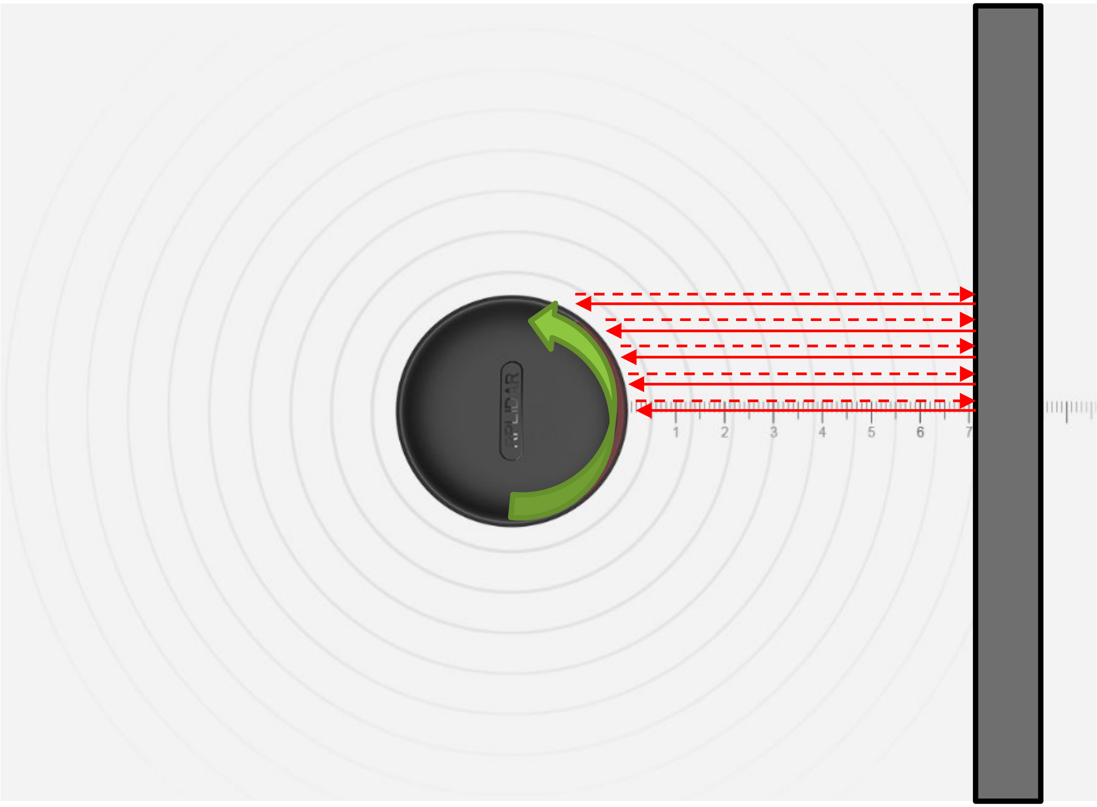
■ LiDAR Operating

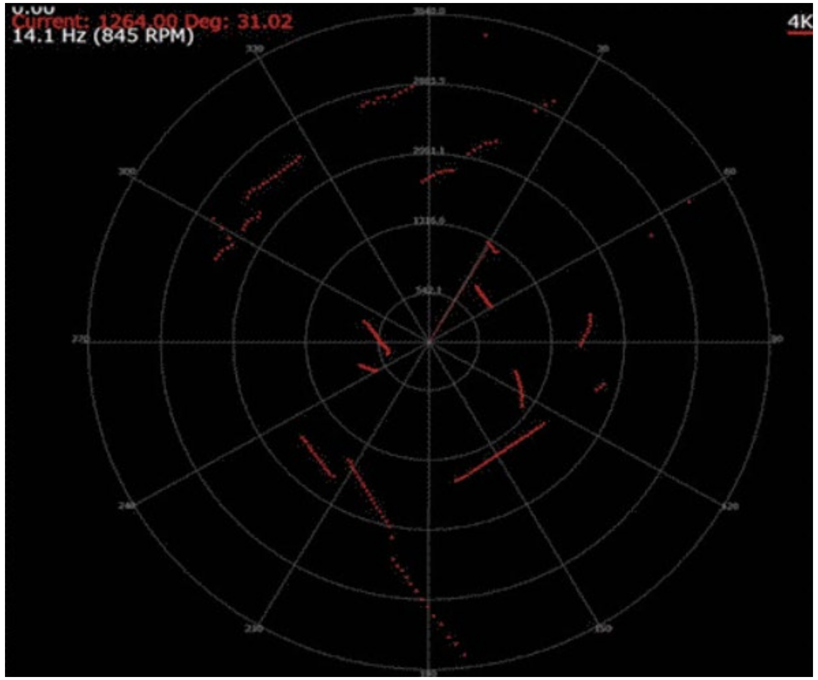


360 degree rotation

Introduction

■ LiDAR Operating

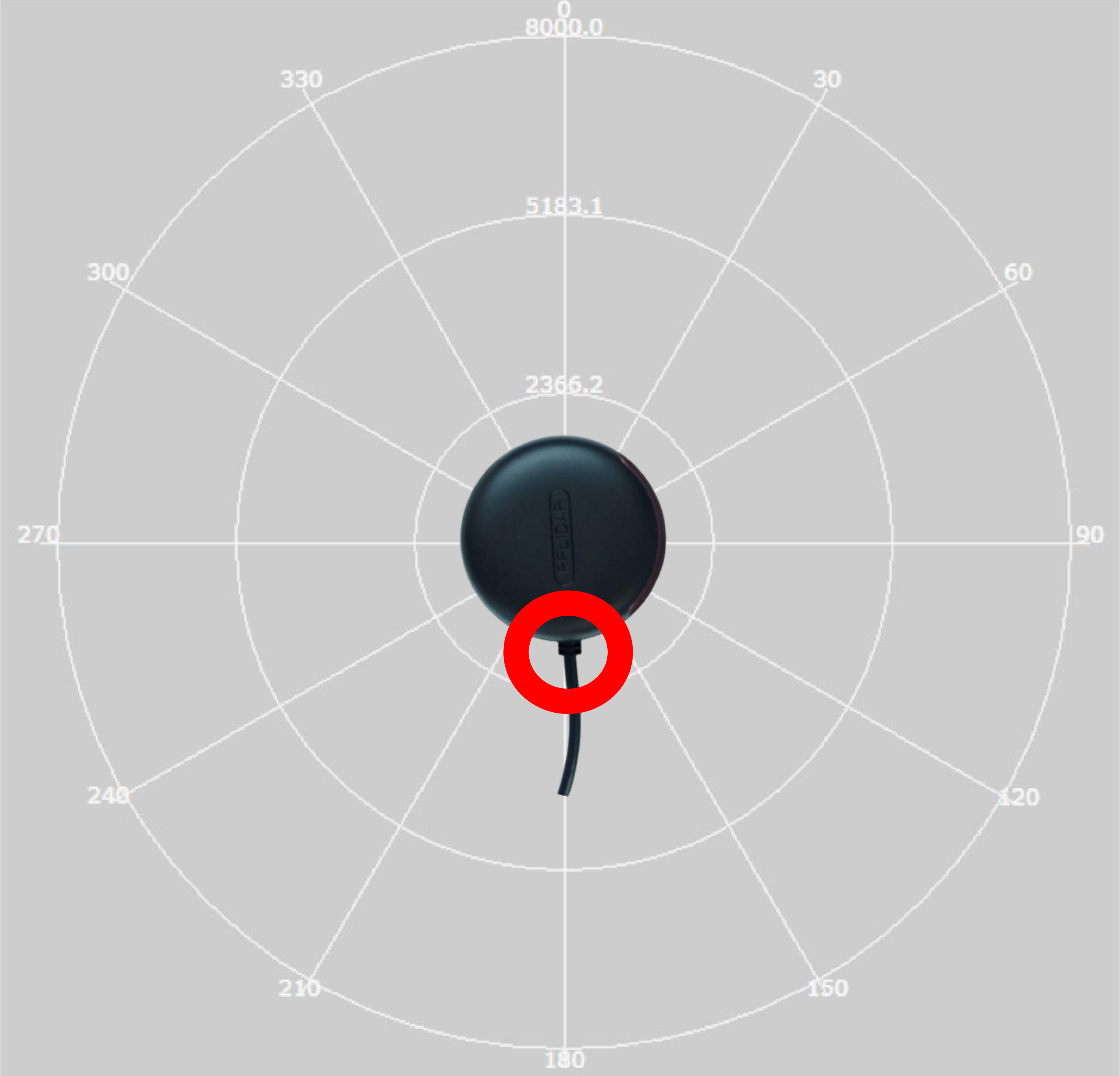




```
Current: 1264.00 Deg: 31.02
14.1 Hz (845 RPM)
```

Introduction

- LiDAR Operating
  - Measurement Angle



Contents

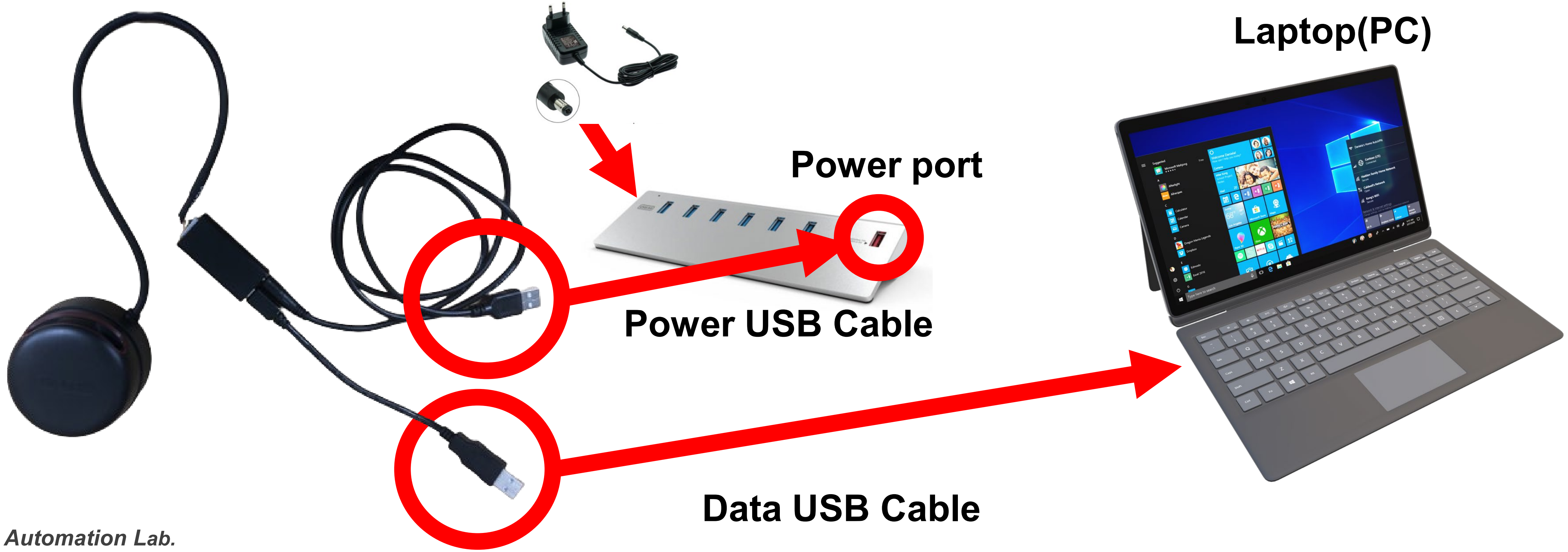
- INTRODUCTION
- EXERCISE

Exercise 1

■ LiDAR Application

- Step 1 : Hardware connection

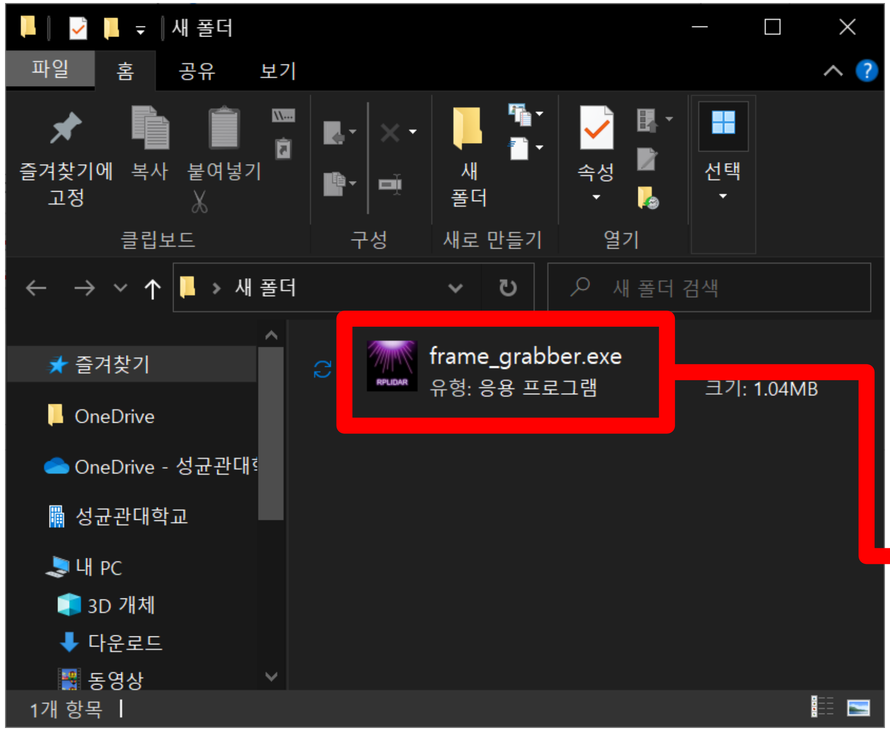
→ Connect LiDAR's power USB cable and data USB cable to the hub and PC



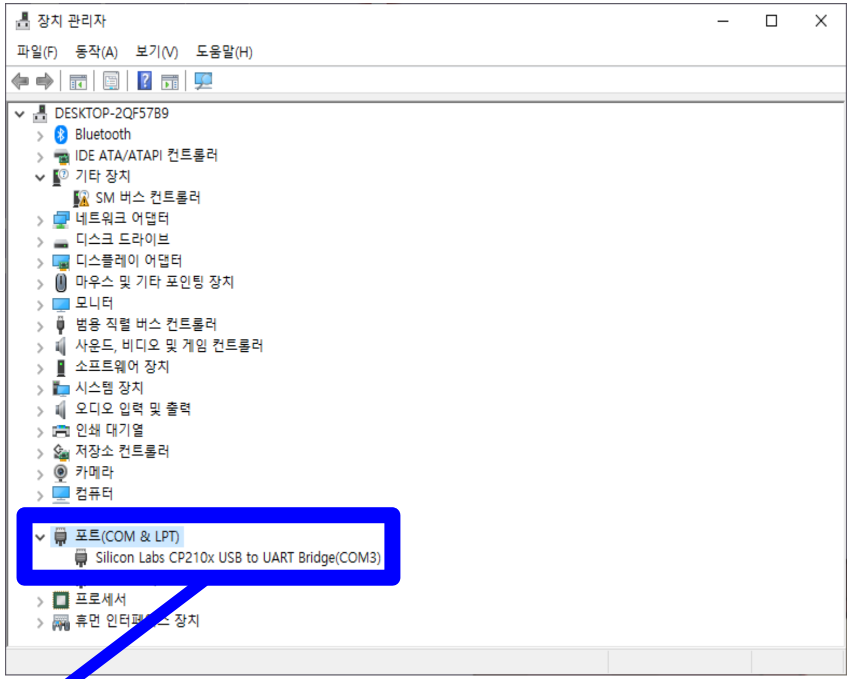
Exercise 1

LiDAR Application

- Step 2 : Running program and Connection



frame_grabber.exe
유형: 응용 프로그램
크기: 1.04MB

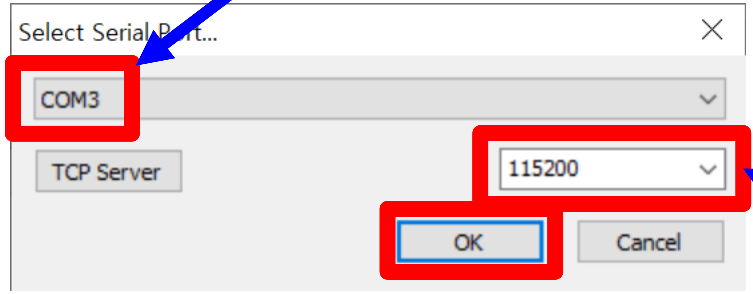


장치 관리자

- DESKTOP-2QF57B9
  - Bluetooth
  - IDE ATA/ATAPI 컨트롤러
  - 기타 장치
    - SM 버스 컨트롤러
  - 네트워크 어댑터
  - 디스크 드라이브
  - 디스플레이 어댑터
  - 마우스 및 기타 포인팅 장치
  - 모니터
  - 범용 직렬 버스 컨트롤러
  - 사운드, 비디오 및 게임 컨트롤러
  - 소프트웨어 장치
  - 시스템 장치
  - 오디오 입력 및 출력
  - 인쇄 대기열
  - 저장소 컨트롤러
  - 카메라
  - 컴퓨터
  - 포트(COM & LPT)
    - Silicon Labs CP210x USB to UART Bridge(COM3)
  - 프로세서
  - 휴먼 인터페이스 장치

Device Manager– port(COM & LPT)
Check CP210x USB Port number

Port setting

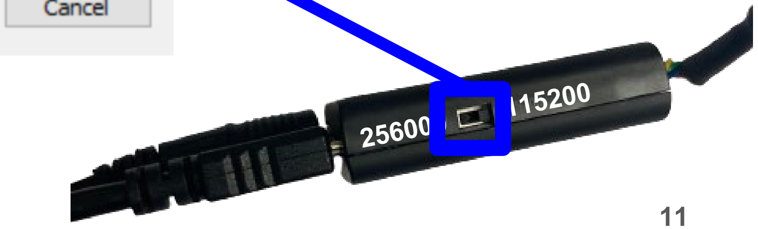


Select Serial Port...

COM3

115200

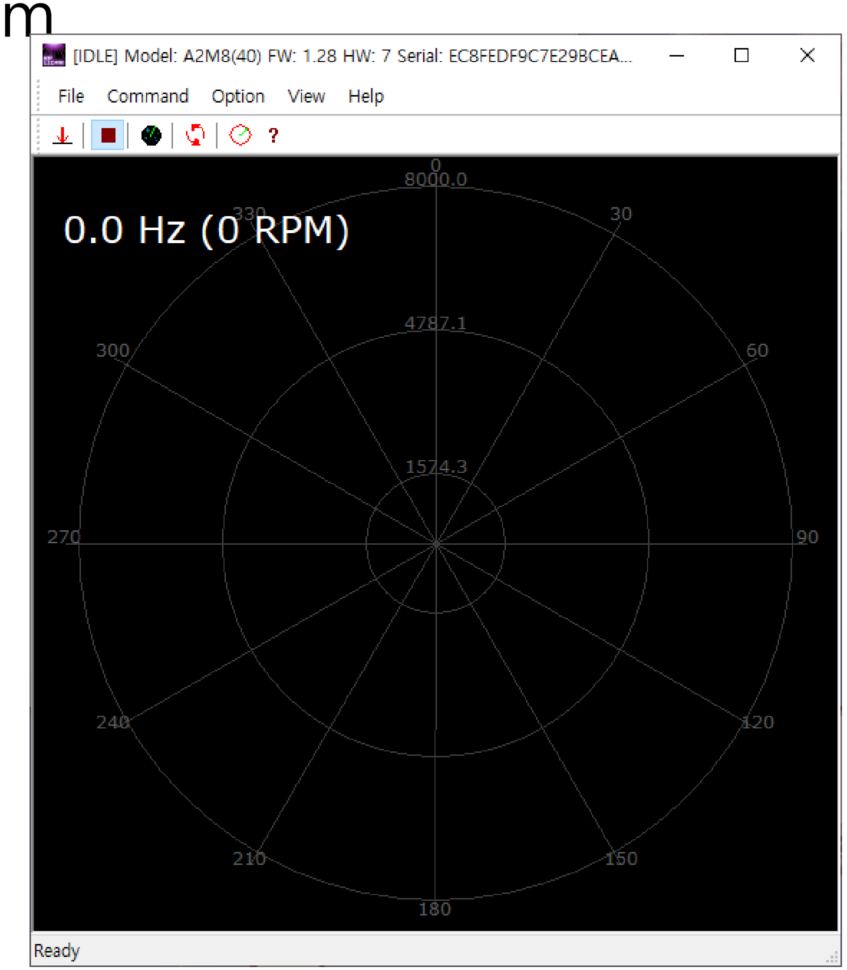
Communication speed



Exercise 1

LiDAR Application

- Step 3 : Running Program

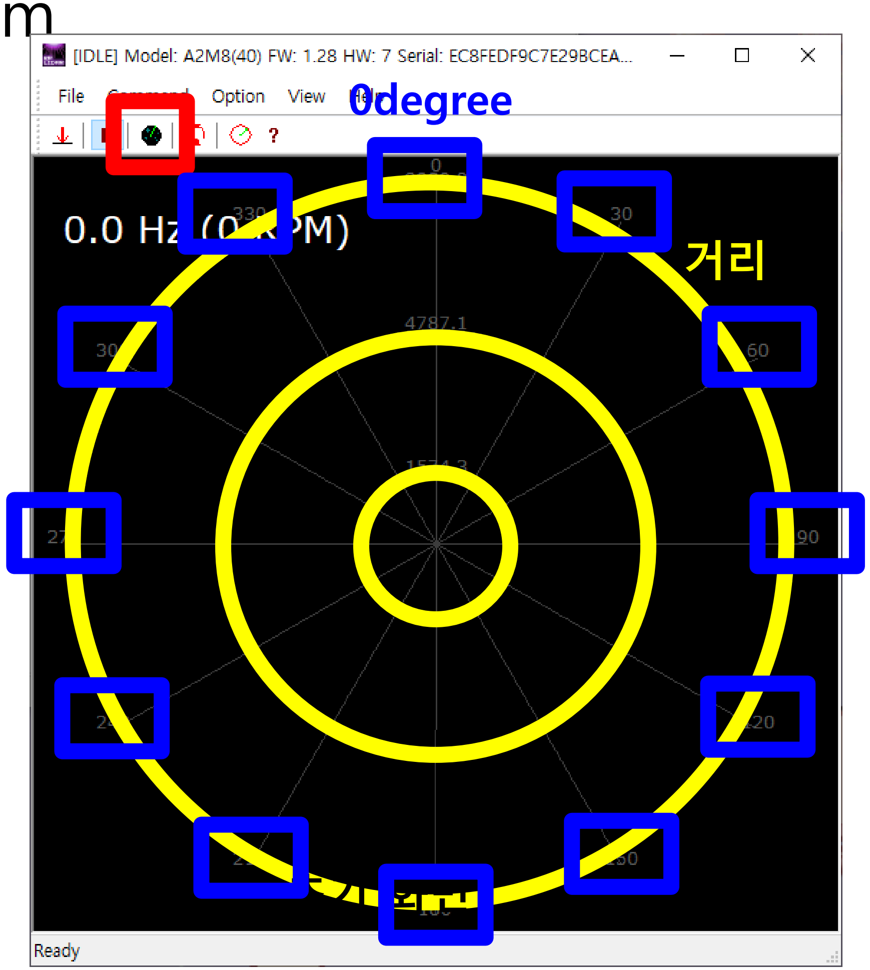


Exercise 1

■ LiDAR Application

- Step 3 : Running Program

When clicked, LiDAR operates.



[IDLE] Model: A2M8(40) FW: 1.28 HW: 7 Serial: EC8FEDF9C7E29BCEA...

0.0 Hz (0 RPM)

Ready

0degree

90degree

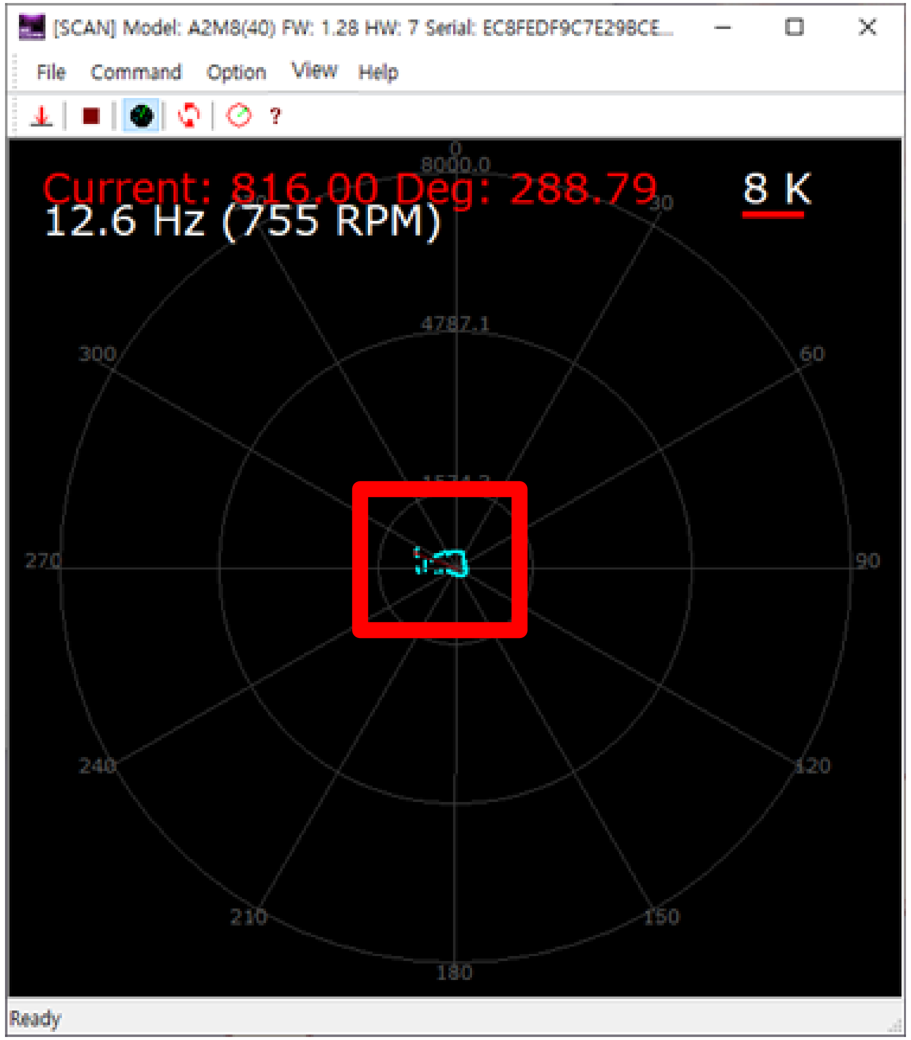
180degree

270degree

Exercise 1

LiDAR Application

- Step 3 : Program Running



[SCAN] Model: A2M8(40) FW: 1.28 HW: 7 Serial: EC8FEDF9C7E298CE...

Current: 816.00 Deg: 288.79

12.6 Hz (755 RPM)

8 K

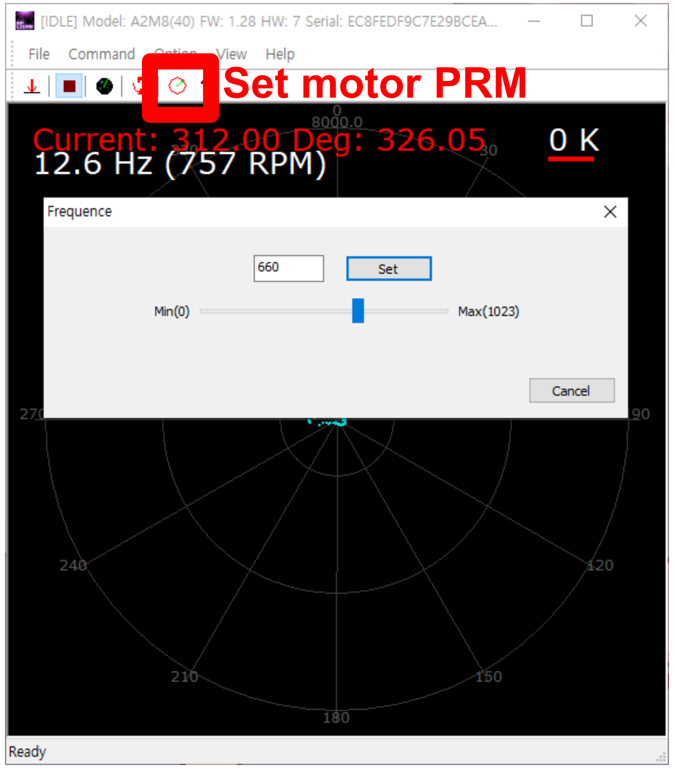
Ready

When the rider is in operation, detect objects as follows

Exercise 1

■ LiDAR Application

- Step 3 : Running Program



Set motor PRM

[IDLE] Model: A2M8(40) FW: 1.28 HW: 7 Serial: EC8FEDF9C7E29BCEA...

Current: 312.00 Deg: 326.05

12.6 Hz (757 RPM)

0 K

Frequence

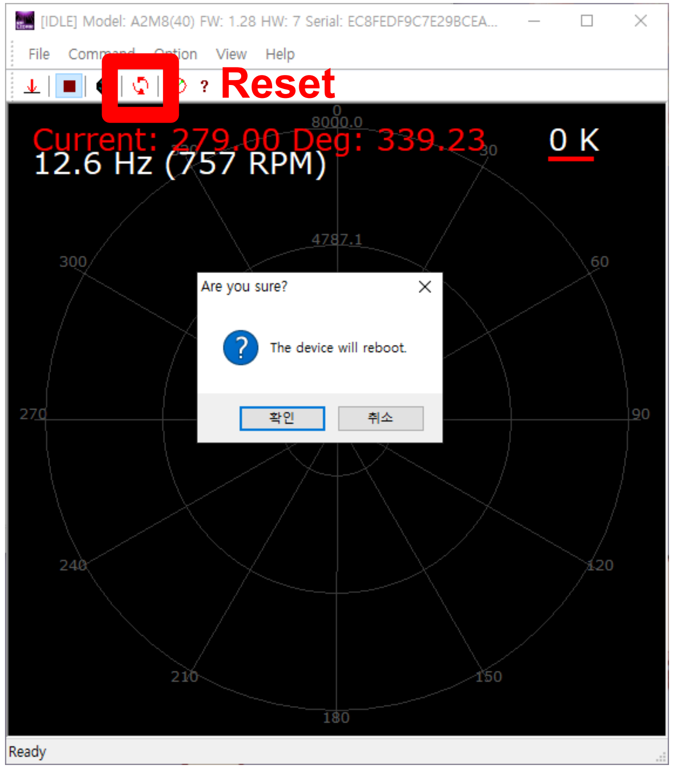
660 Set

Min(0) Max(1023)

Cancel

Ready

Motor Motion Speed Settings



Reset

[IDLE] Model: A2M8(40) FW: 1.28 HW: 7 Serial: EC8FEDF9C7E29BCEA...

Current: 279.00 Deg: 339.23

12.6 Hz (757 RPM)

0 K

Are you sure?

The device will reboot.

확인 취소

Ready

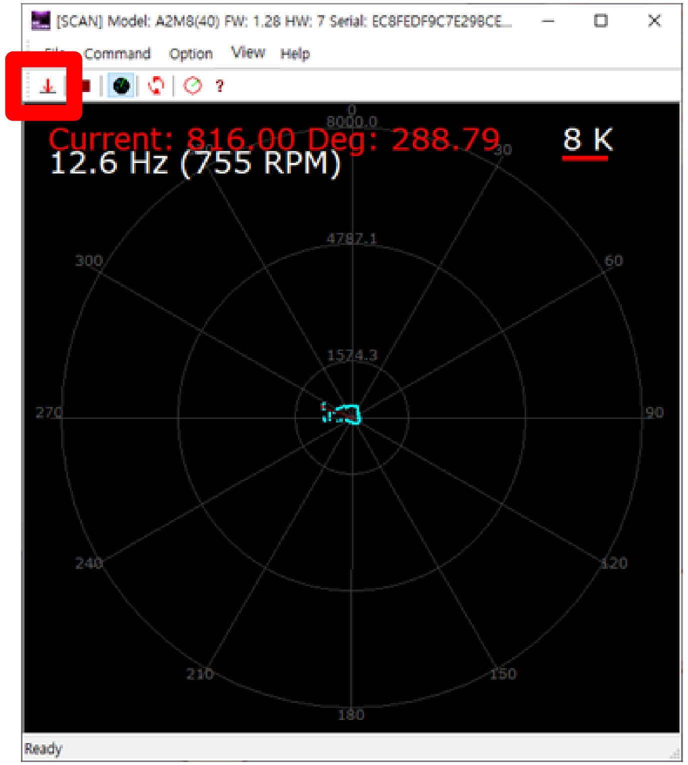
Reset

Exercise 1

■ LiDAR Application

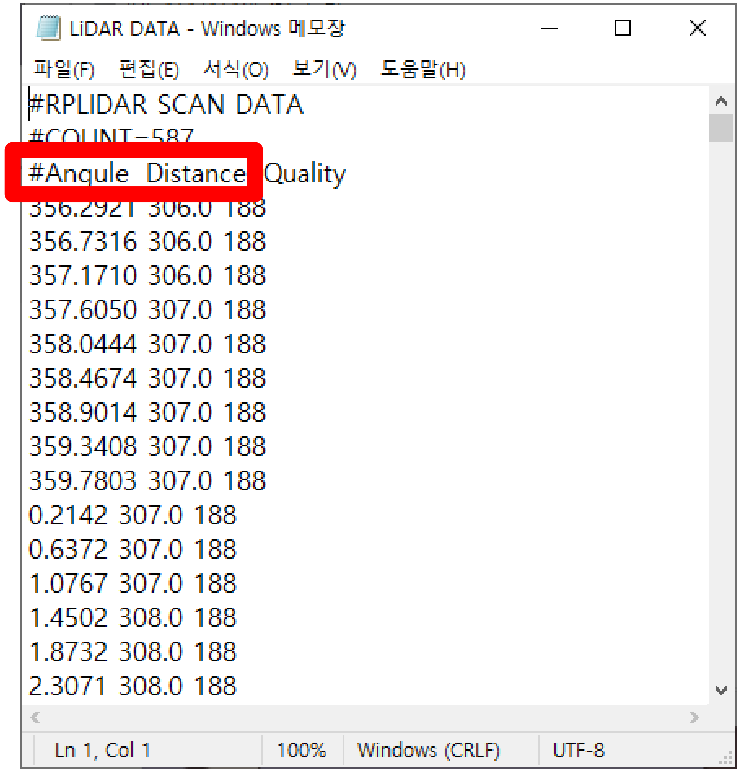
- Step 3 : Running Program

Dump data



[SCAN] Model: A2M8(40) FW: 1.28 HW: 7 Serial: EC8FEDF9C7E298CE...
File Command Option View Help
Current: 816.00 Deg: 288.79
12.6 Hz (755 RPM)
8 K
8000.0
4787.1
1574.3
30
60
90
120
150
180
210
240
270
300
Ready

Storing LiDAR detection information



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

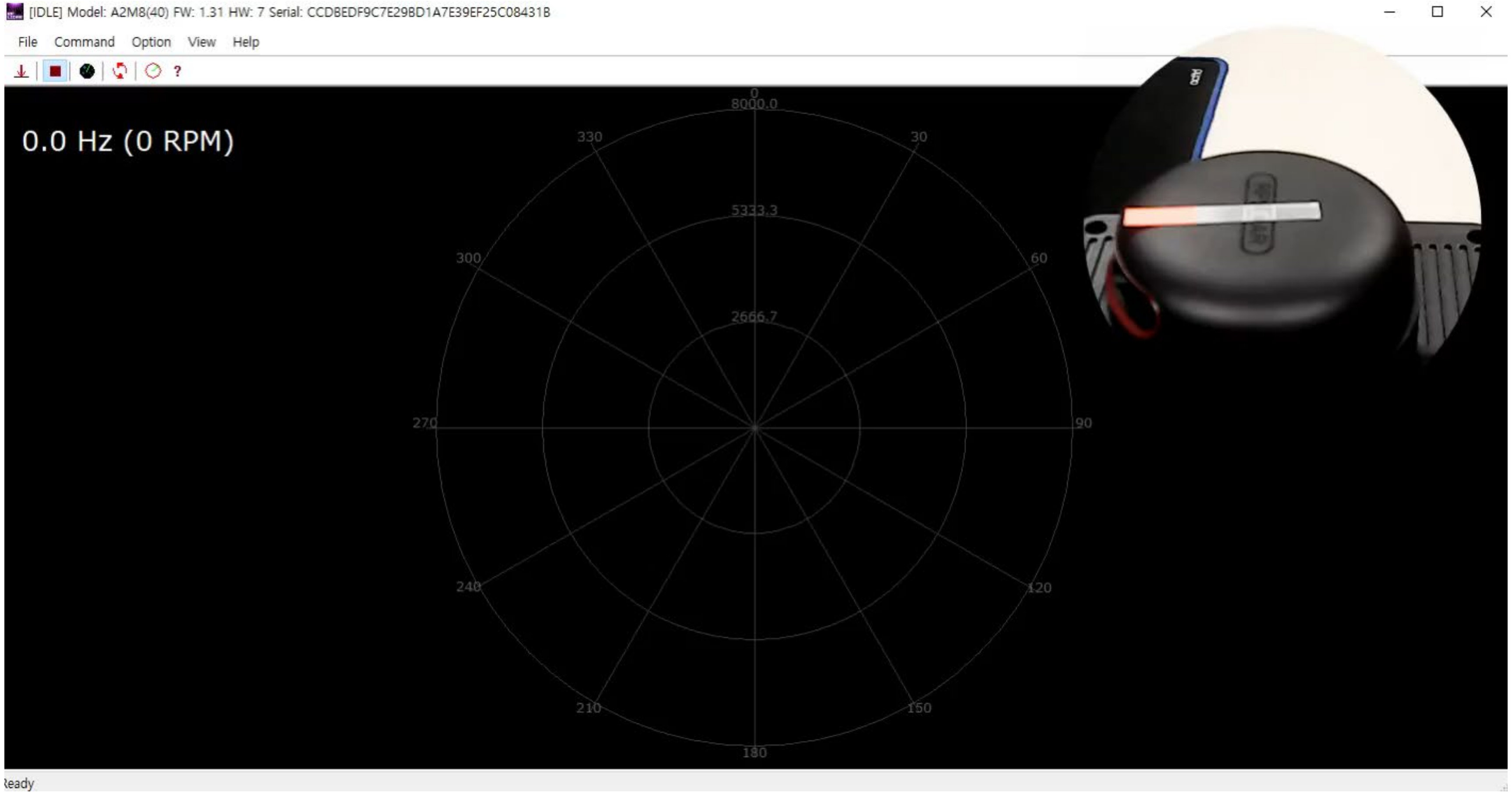
```
#RPLIDAR SCAN DATA
#COUNT=587
#Angule  Distance  Quality
356.2921 306.0 188
356.7316 306.0 188
357.1710 306.0 188
357.6050 307.0 188
358.0444 307.0 188
358.4674 307.0 188
358.9014 307.0 188
359.3408 307.0 188
359.7803 307.0 188
0.2142 307.0 188
0.6372 307.0 188
1.0767 307.0 188
1.4502 308.0 188
1.8732 308.0 188
2.3071 308.0 188
```

Distance information available based on degrees from 0 to 365

Exercise 1

■ LiDAR Application

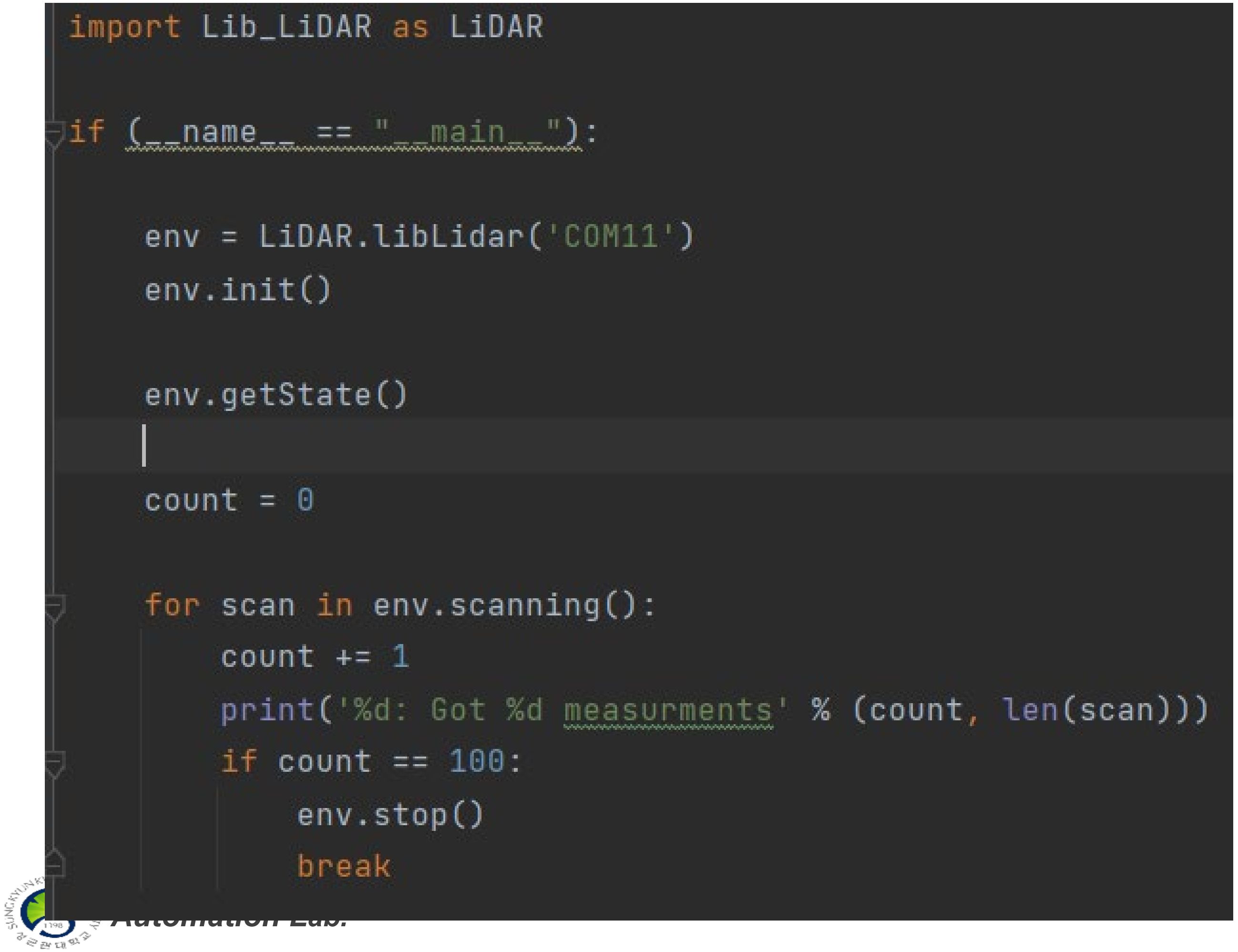
- ● Result Video



[IDLE] Model: A2M8(40) FW: 1.31 HW: 7 Serial: CCDBEDF9C7E29BD1A7E39EF25C08431B

Exercise 2

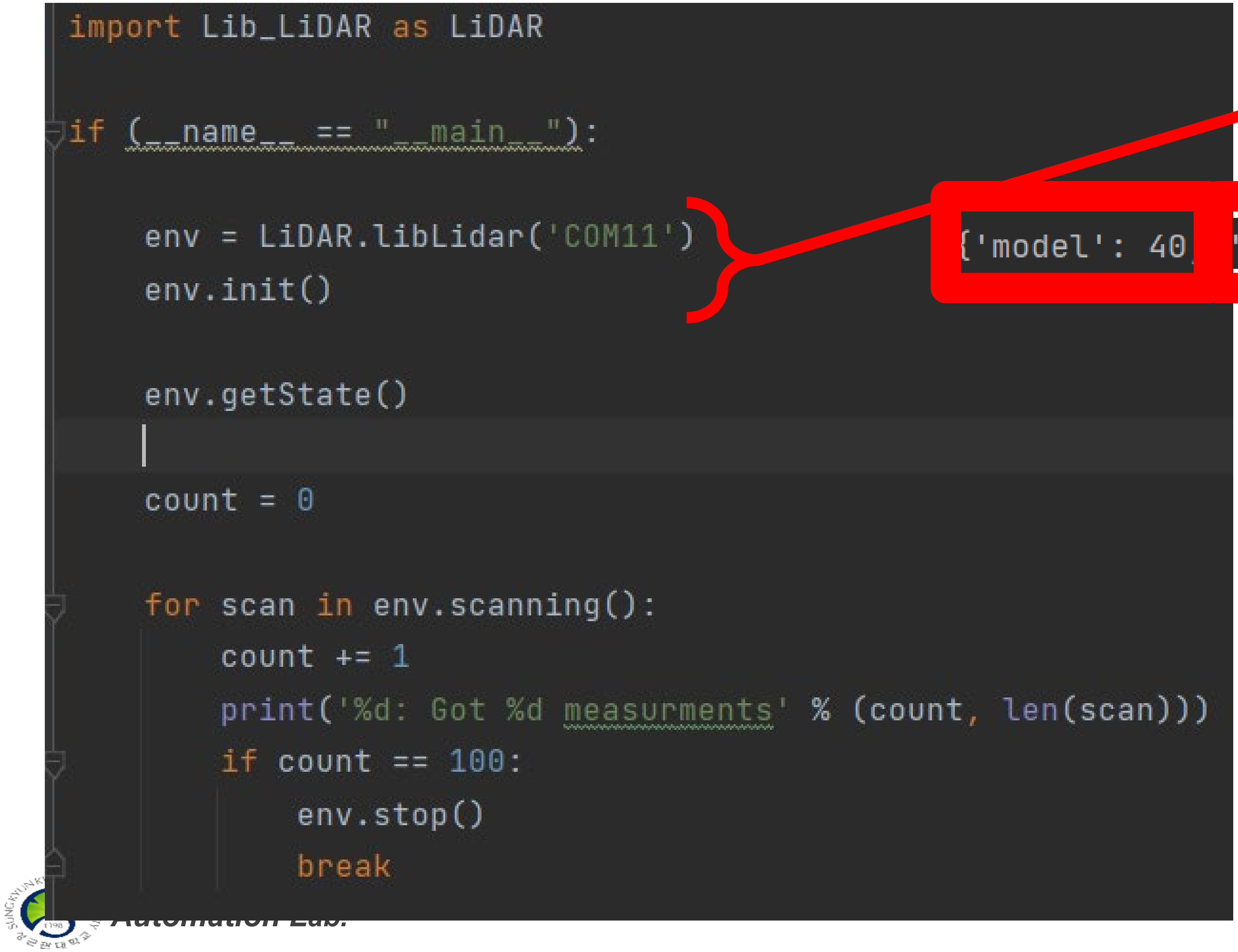
■ LiDAR Basic Functions



```
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
```

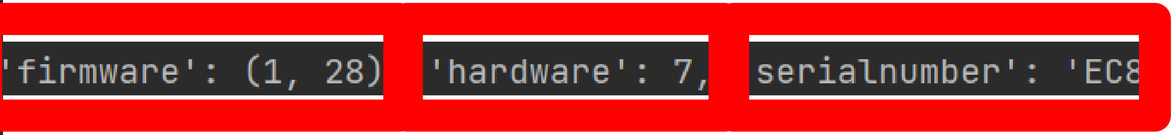
Exercise 2

■ LiDAR Basic Functions

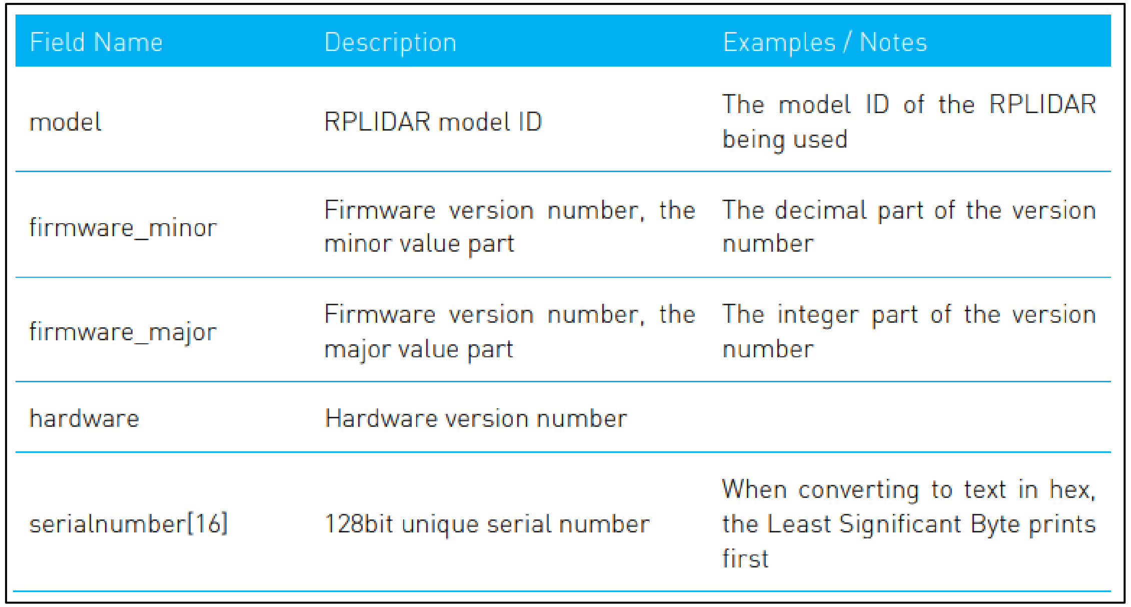


```
import Lib_LiDAR as LiDAR

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 Initialize



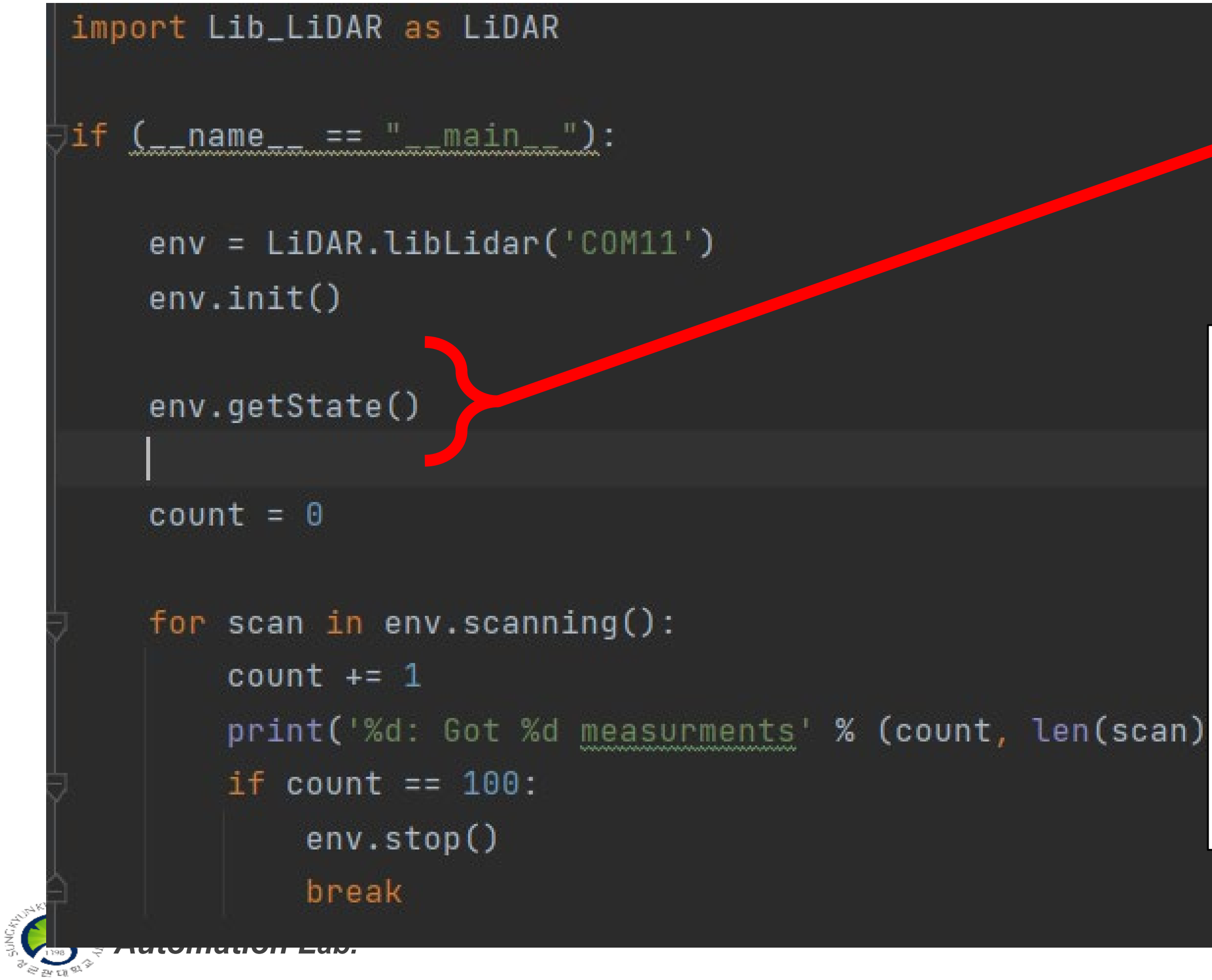
```
{'model': 40, 'firmware': (1, 28), 'hardware': 7, serialnumber': 'EC8
```



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

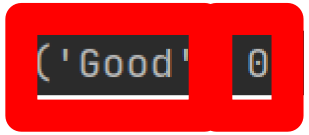
Exercise 2

■ LiDAR Basic Functions

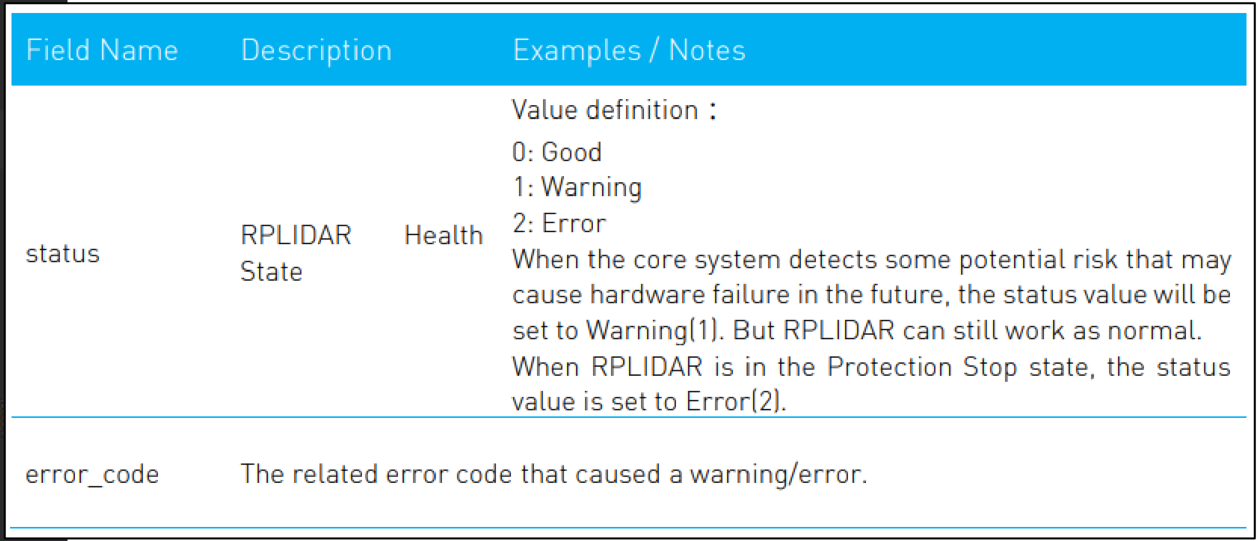


```
import Lib_LiDAR as LiDAR

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
```

Get LiDAR status



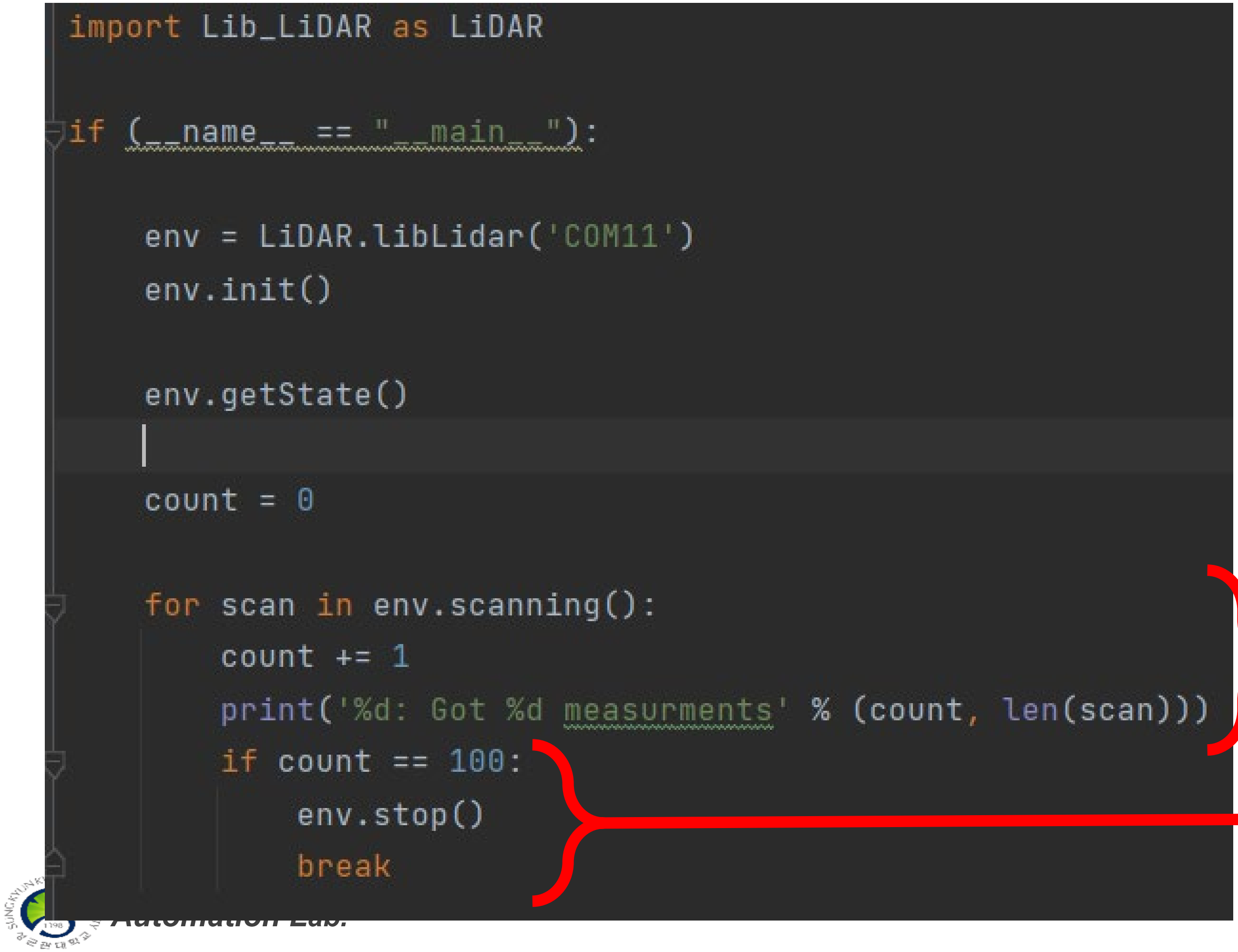
```
('Good'  0
```



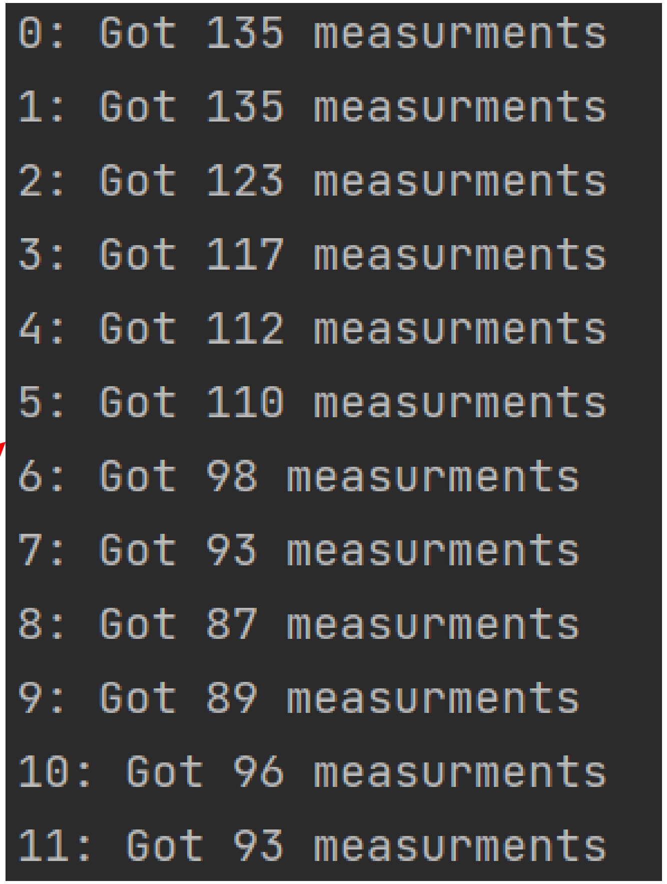
Field Name	Description	Examples / Notes
status	RPLIDAR Health State	Value definition : 0: Good 1: Warning 2: Error When the core system detects some potential risk that may cause hardware failure in the future, the status value will be set to Warning(1). But RPLIDAR can still work as normal. When RPLIDAR 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 Basic Functions



```
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
```



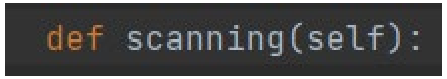
```
0:  Got 135 measurments
1:  Got 135 measurments
2:  Got 123 measurments
3:  Got 117 measurments
4:  Got 112 measurments
5:  Got 110 measurments
6:  Got 98 measurments
7:  Got 93 measurments
8:  Got 87 measurments
9:  Got 89 measurments
10:  Got 96 measurments
11:  Got 93 measurments
```

LiDAR stop

Exercise 2

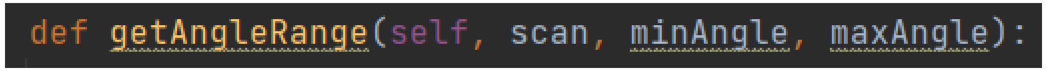
LiDAR Basic Functions

1) LiDAR data scan



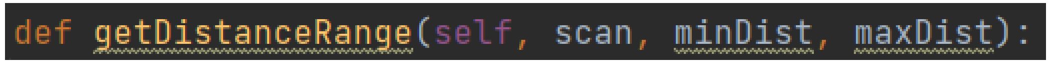
```
def scanning(self):
```

2) Output only data within a specific angle range



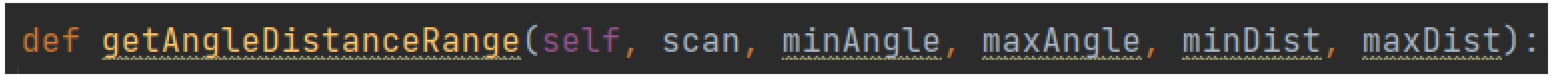
```
def getAngleRange(self, scan, minAngle, maxAngle):
```

3) Only output data within a specific distance range



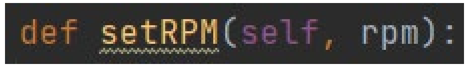
```
def getDistanceRange(self, scan, minDist, maxDist):
```

4) Output only data within a specific angle and distance range

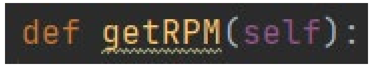


```
def getAngleDistanceRange(self, scan, minAngle, maxAngle, minDist, maxDist):
```

5) Change and check the RPM of the LiDAR Motor

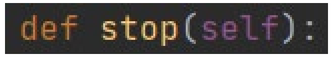


```
def setRPM(self, rpm):
```



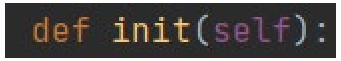
```
def getRPM(self):
```

6) LiDAR stop



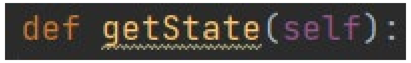
```
def stop(self):
```

7) LiDAR initialization



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

8) Check LiDAR status

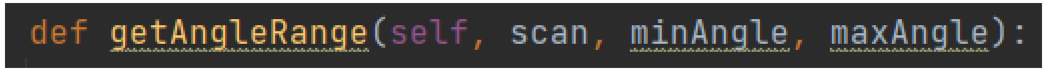


```
def getState(self):
```

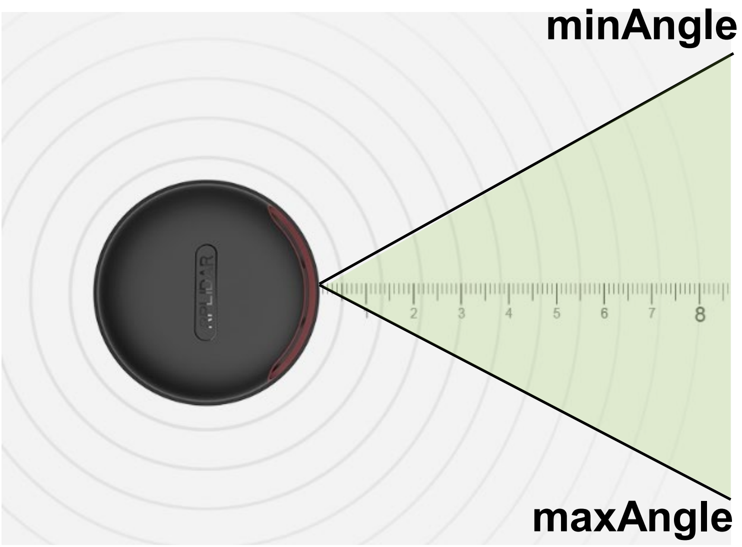
Exercise 2

■ LiDAR Basic Function

1) Only output data within a specific angle range



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



- Scan

· Result data obtained through the scanning() function

- minAngle

· Minimum value of the angle to scan (0 or higher)

- maxAngle

· Maximum value of the angle to scan(360 이하)

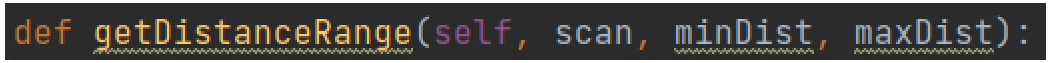
- Return: Search results in the form of a list

· Output only data that satisfies the set conditions from the input data.

Exercise 2

■ LiDAR Basic Function

2) Output only data within a specific distance range



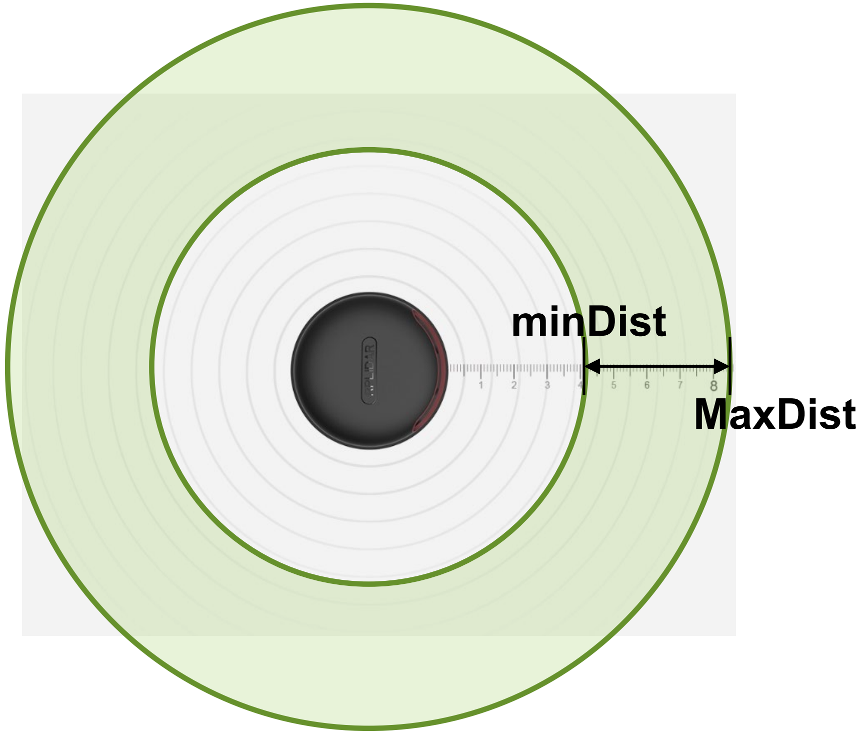
```
def getDistanceRange(self, scan, minDist, maxDist):
```

- Scan
Result data obtained through the scanning() function

- minDist
· Minimum distance to san (150 이상)

- maxDist
· Maximum distance to scan (600 이하)

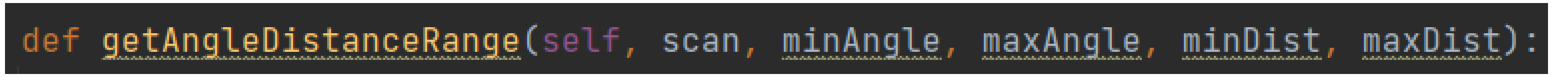
- Return: Search results in the form of a list
· Output only data that satisfies the set conditions from the input data.



Exercise 2

■ LiDAR Basic Function

3) Output only data within a specific angle and distance range



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

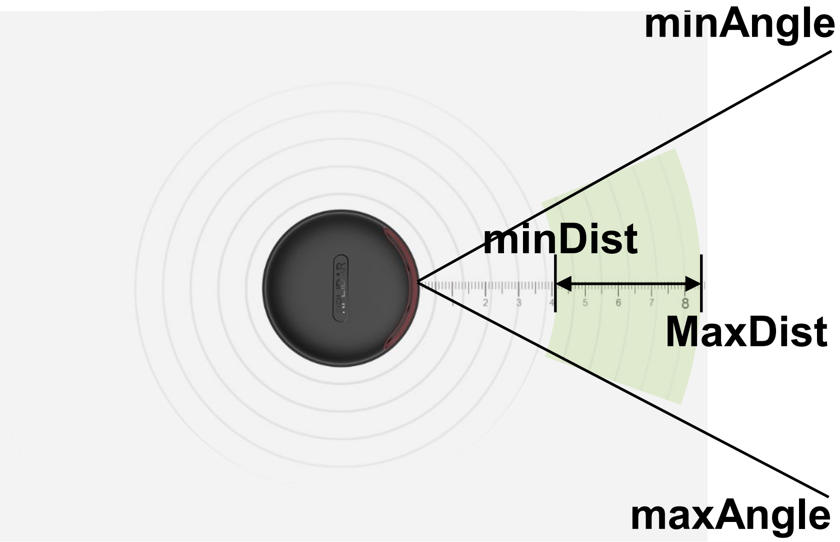
- Scan
· Result data obtained through the scanning() function

- minAngle
· Minimum value of the angle to search (0 or higher)

- maxAngle
· Maximum angle to search (360 or less)

- minDist
· Minimum distance to search (150 or higher)

- maxDist
· Maximum distance to search (600 or less)

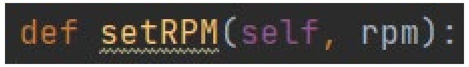


- Return: Search results in the form of a list
· Output only data that satisfies the set conditions from the input data.

Exercise 2

■ LiDAR Basic Function

4) Change and check the RPM of the LiDAR Motor

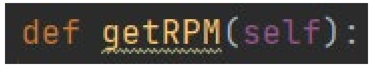


```
def setRPM(self, rpm):
```

- rpm

· Rotational speed of lidar motor
· Minimum 0, maximum 1023
· Default : 660

- Return: None



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

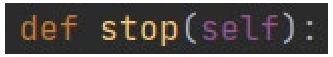
- Return: rpm(Int)

· Returns the currently set rpm value

Exercise 2

■ LiDAR Basic Function

5) LiDAR Stop



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

- When the function is executed, stop LiDAR operation and disconnect.

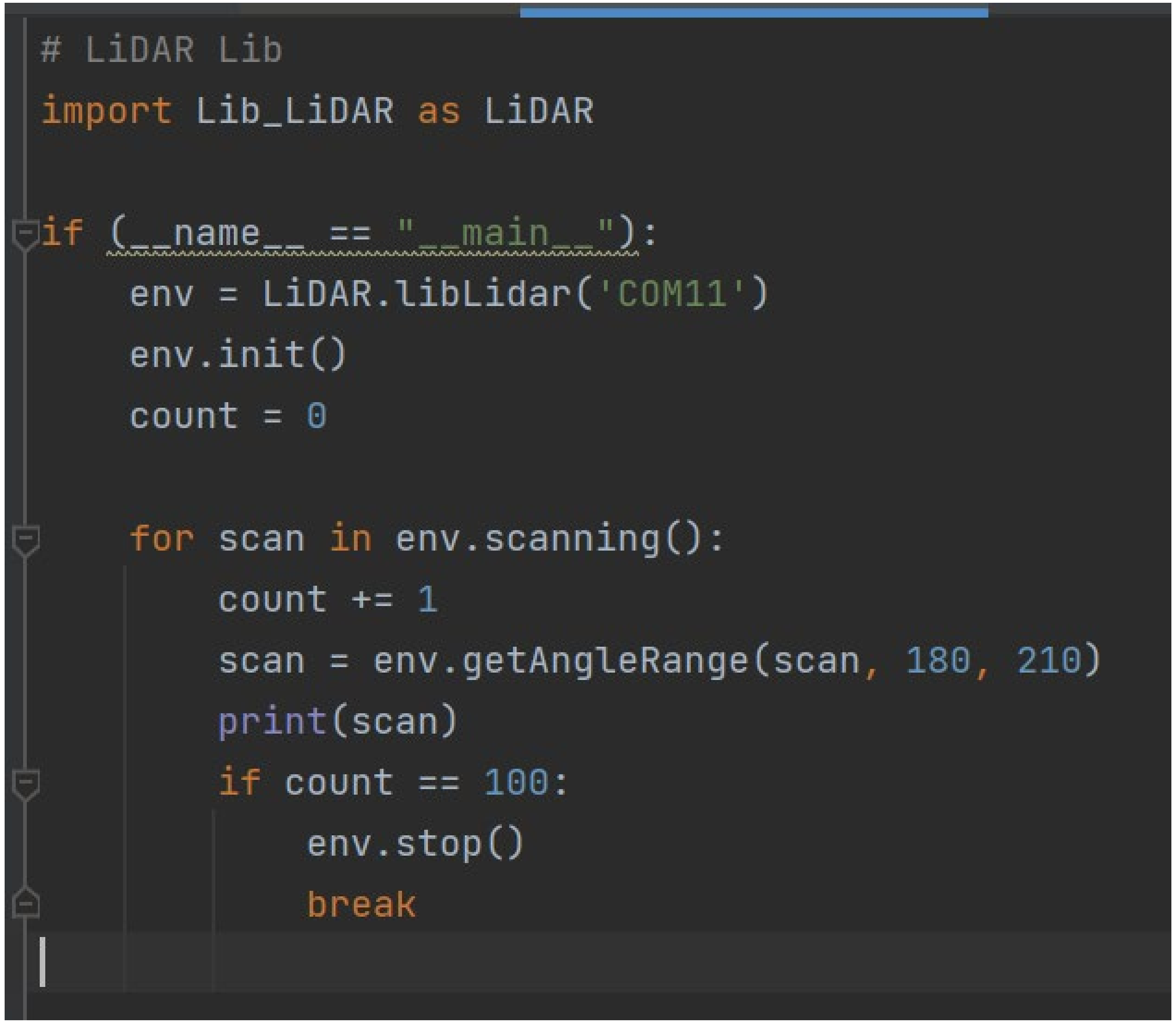
- Return: None

Exercise 2-1

LiDAR Basic Function

- 2-1 : Output only LiDAR information within 180° to 210°

Exercise 2-1



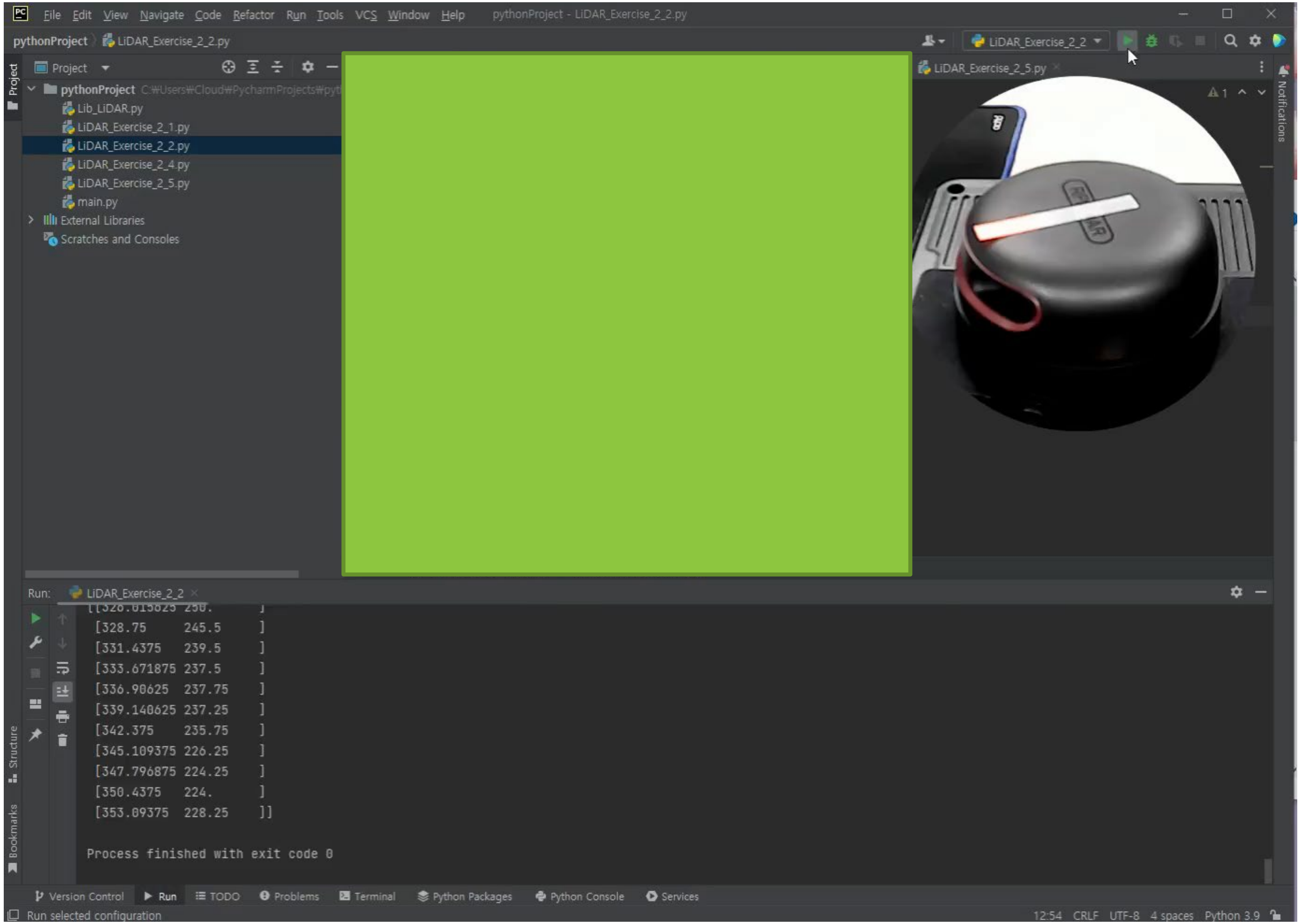
```python
# LiDAR Lib
import Lib_LiDAR as LiDAR

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 Basic Function

- 2-2 : Output only LiDAR information within 150mm ~ 300mm

Exercise 2-2



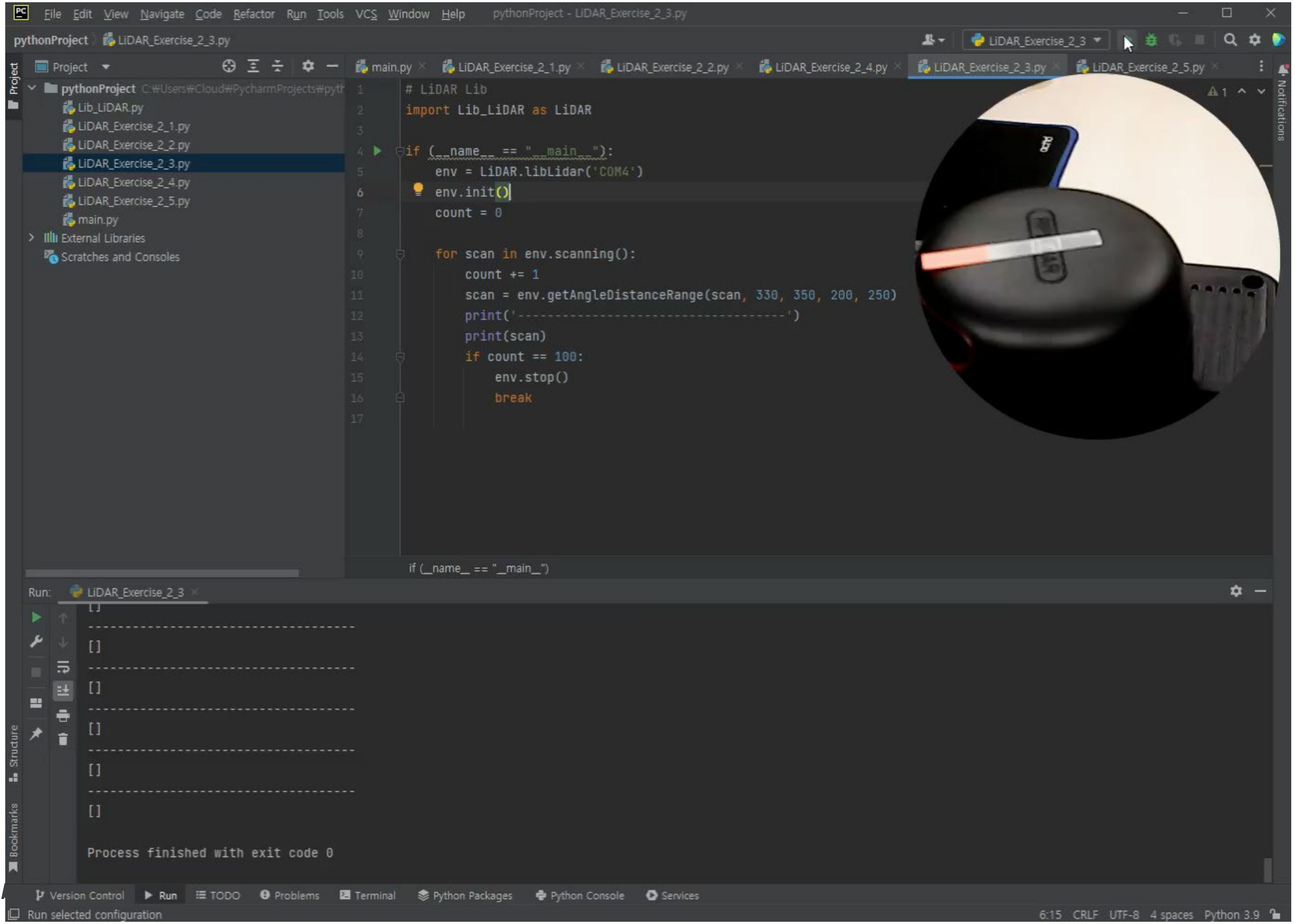
```
[[326.015625 250.        ]
 [328.75     245.5      ]
 [331.4375   239.5      ]
 [333.671875 237.5      ]
 [336.90625  237.75     ]
 [339.140625 237.25     ]
 [342.375    235.75     ]
 [345.109375 226.25     ]
 [347.796875 224.25     ]
 [350.4375   224.       ]
 [353.09375  228.25     ]]

Process finished with exit code 0
```

Exercise 2-3

LiDAR Basic Function

- 2-3 : Output only information contained within 200 mm to 250 mm of LiDAR information within 330° to 350°.

Exercise 2-3



```python
# LiDAR Lib
import Lib_LiDAR as LiDAR

if (__name__ == "__main__"):
    env = LiDAR.libLidar('COM4')
    env.init()
    count = 0

    for scan in env.scanning():
        count += 1
        scan = env.getAngleDistanceRange(scan, 330, 350, 200, 250)
        print('-----------------------------------')
        print(scan)
        if count == 100:
            env.stop()
            break
```

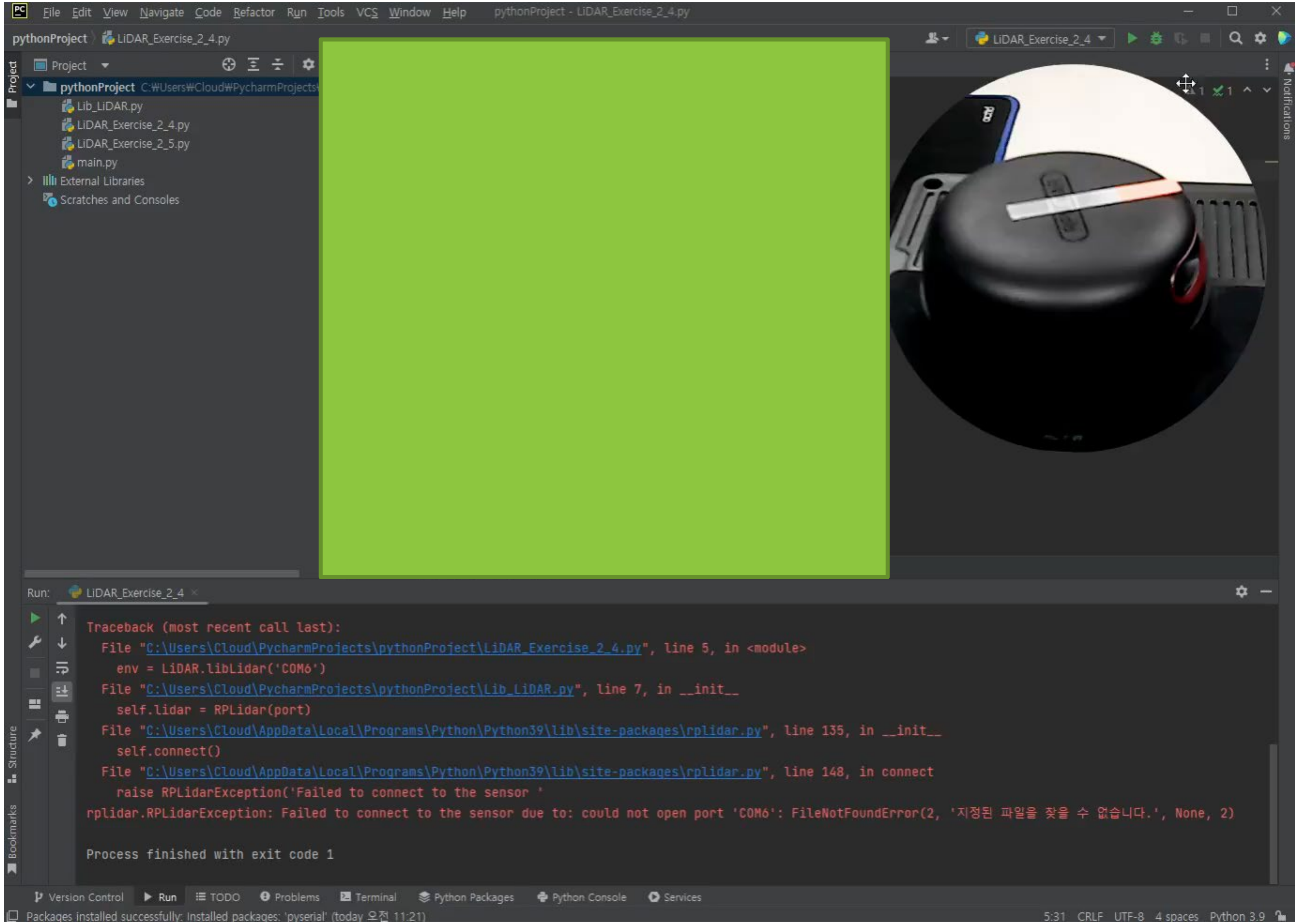
```
[]
-----------------------------------
[]
-----------------------------------
[]
-----------------------------------
[]
-----------------------------------
[]
-----------------------------------
[]

Process finished with exit code 0
```

Exercise 2-4

LiDAR Basic Function

- 2-4 : First, start the LiDAR at 660 rpm, and after 30 operations, change the motor RPM to 1000 rpm.

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
```

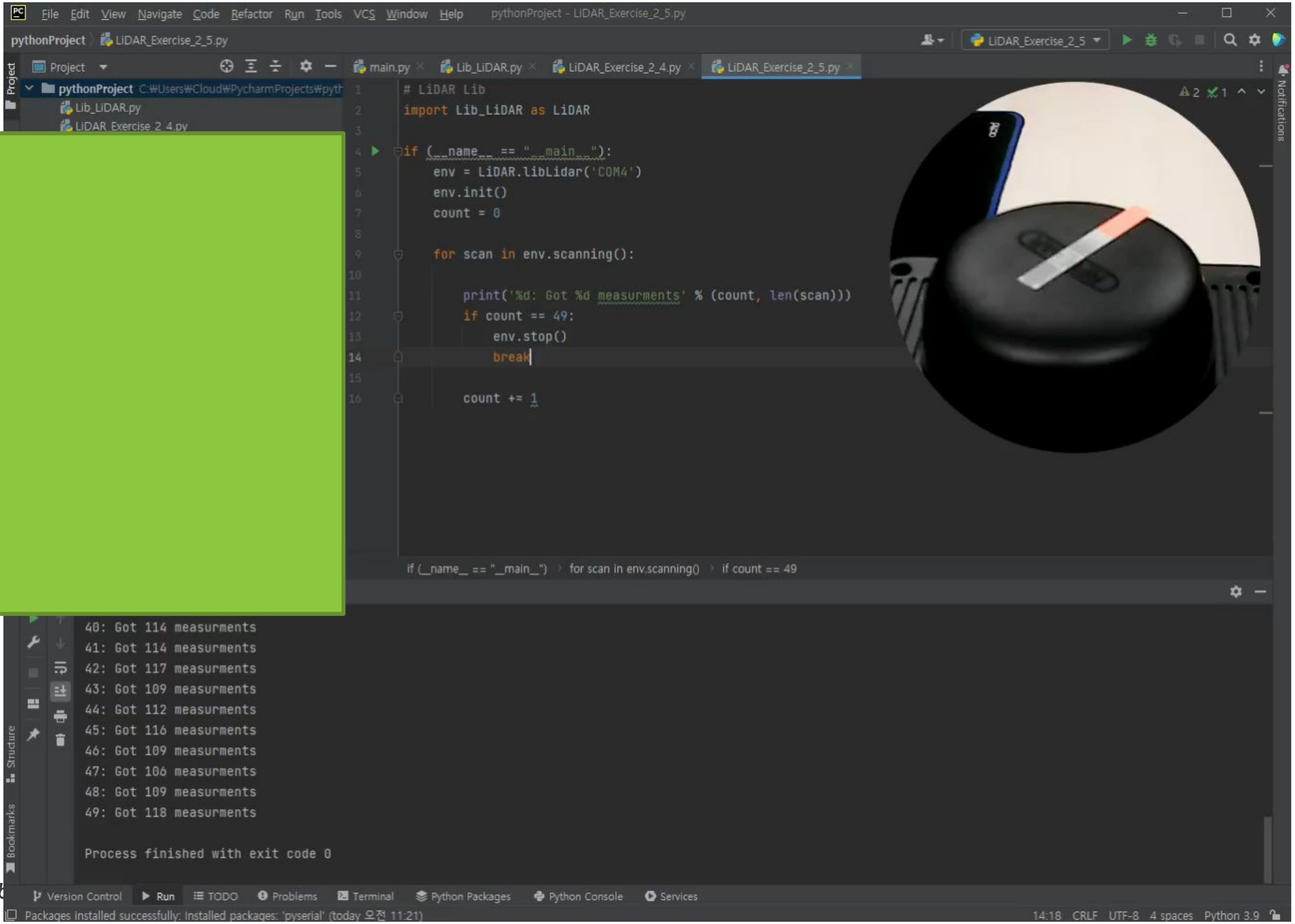
Packages installed successfully: Installed packages: 'pyserial' (today 오전 11:21)

Exercise 2-5

LiDAR Basic Function

- 2-5 : After 50 LiDAR scanning operations, Stop the LiDAR

Exercise 2-5


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - LiDAR_Exercise_2_5.py
pythonProject
LiDAR_Exercise_2_5.py
Project
pythonProject C:₩Users₩Cloud₩PycharmProjects₩pyth
Lib_LiDAR.py
LiDAR_Exercise_2_4.py
main.py
Lib_LiDAR.py
LiDAR_Exercise_2_4.py
LiDAR_Exercise_2_5.py
LiDAR_Exercise_2_5
if (__name__ == "__main__") › for scan in env.scanning() › if count == 49
Version Control
Run
TODO
Problems
Terminal
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

```python
# LiDAR Lib
import Lib_LiDAR as LiDAR

if (__name__ == "__main__"):
    env = LiDAR.libLidar('COM4')
    env.init()
    count = 0

    for scan in env.scanning():

        print('%d: Got %d measurments' % (count, len(scan)))
        if count == 49:
            env.stop()
            break

        count += 1
```

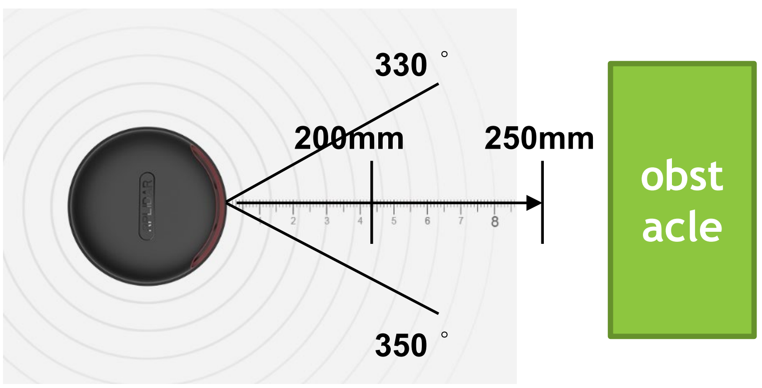
```
40: Got 114 measurments
41: Got 114 measurments
42: Got 117 measurments
43: Got 109 measurments
44: Got 112 measurments
45: Got 116 measurments
46: Got 109 measurments
47: Got 106 measurments
48: Got 109 measurments
49: Got 118 measurments

Process finished with exit code 0
```

Exercise 3

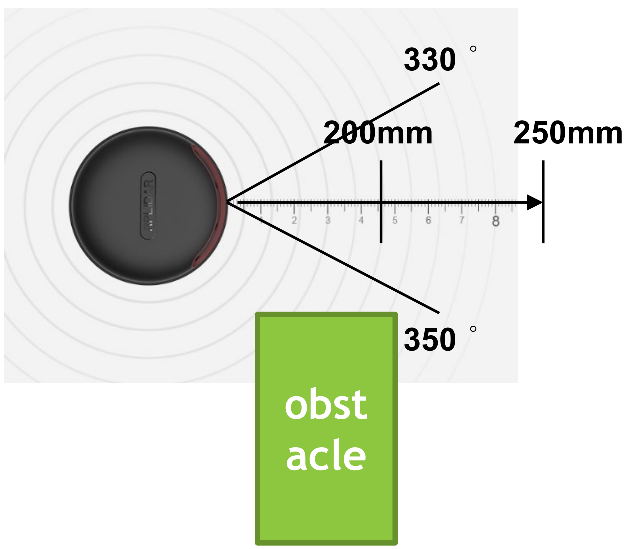
LiDAR Operation after object detection

- If no object is detected at a specific angle and distance, output “Go,” and if an object is detected, output “Stop” and stop the LiDAR.
  - Angle (330° to 350°), distance (200 to 250 mm)

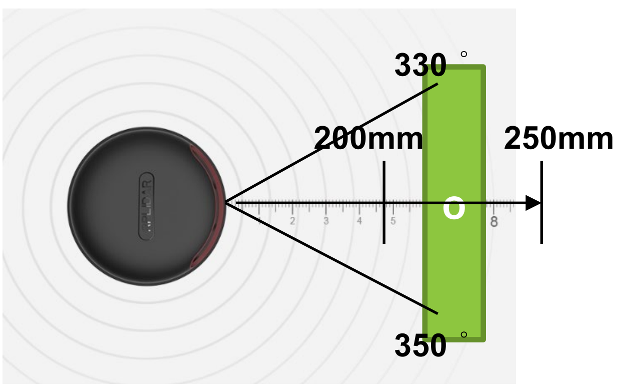
“Go”



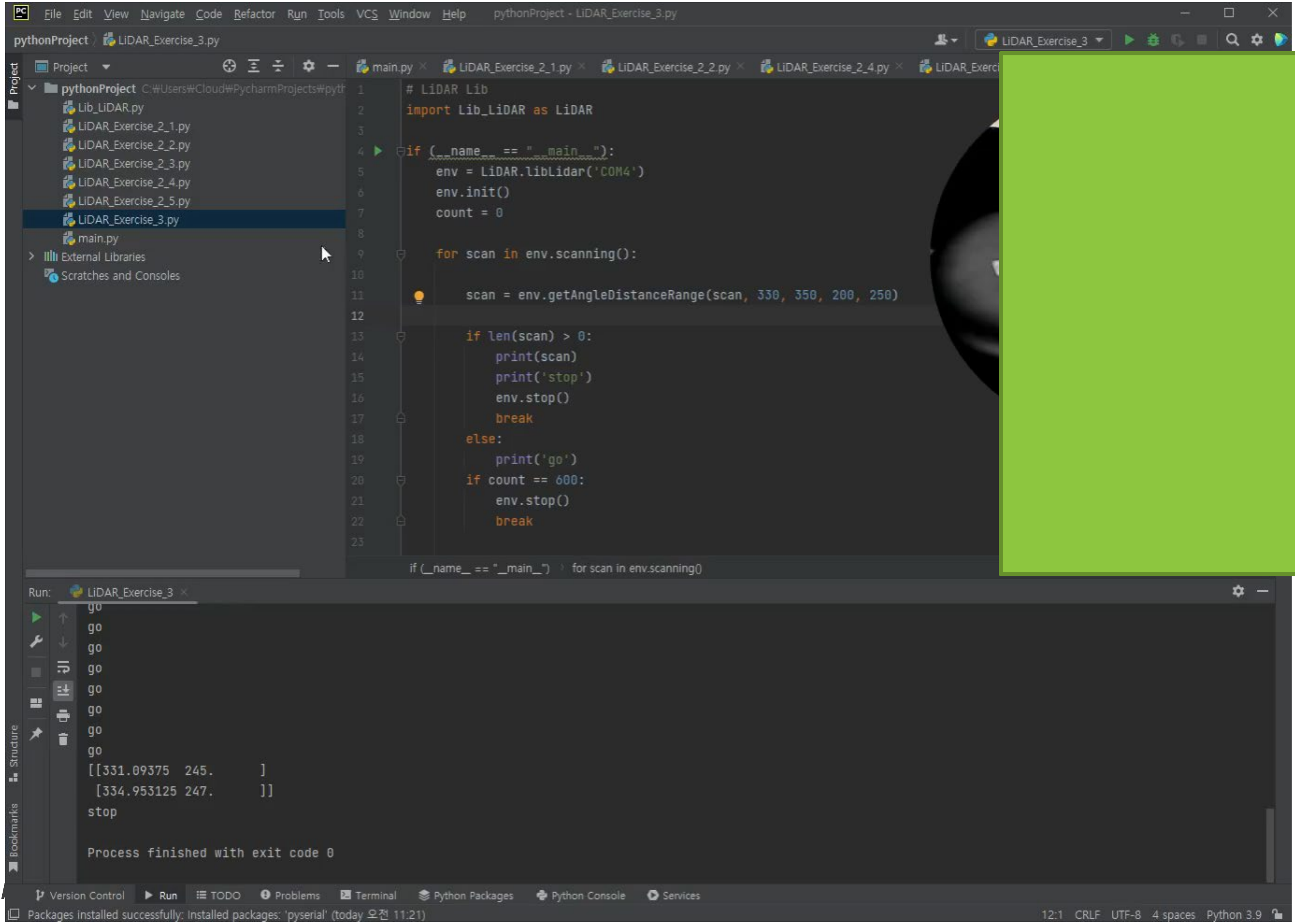
“Go”



“Stop”-> stop LiDAR



Exercise 3



```python
# LiDAR Lib
import Lib_LiDAR as LiDAR

if (__name__ == "__main__"):
    env = LiDAR.libLidar('COM4')
    env.init()
    count = 0

    for scan in env.scanning():

        scan = env.getAngleDistanceRange(scan, 330, 350, 200, 250)

        if len(scan) > 0:
            print(scan)
            print('stop')
            env.stop()
            break
        else:
            print('go')
        if count == 600:
            env.stop()
            break

```

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

Process finished with exit code 0
```

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

Thank You!

Automation Lab.

SUNGKYUNKWAN UNIVERSITY

1398

성균관대학교