

ROD 308 and ROD 508

Demo script 2D

Content

1. Introduction 3

2. Requirements and preparation 4

 2.1. Launch script 4

 2.2. Direct execution 4

 2.3. Firewall settings 4

1. High-Level Workflow 6

 1.1. Connection setup 6

 1.2. Command handling 6

 1.3. Data reception and parsing 7

 1.4. Data integrity 7

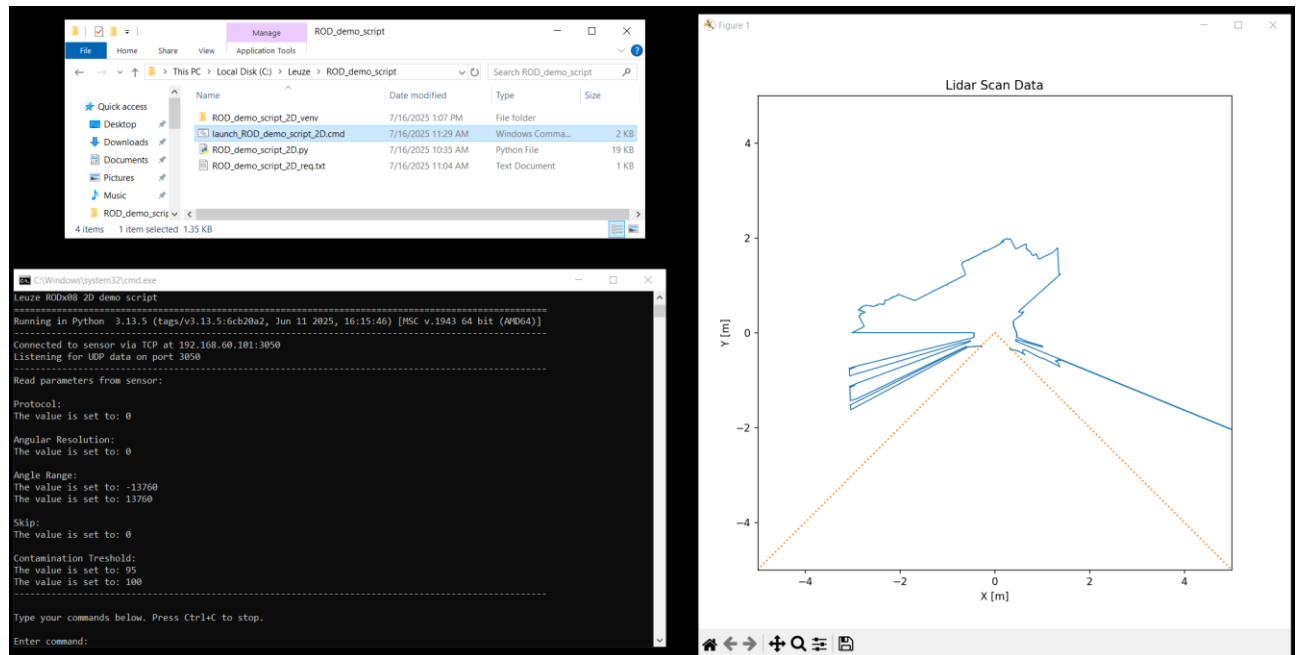
 1.5. Visualization 7

 1.6. Code structure 7

2. Changelog 9

1. Introduction

The ROD demo script 2D is a Python script designed to connect ROD 308 and ROD 508 LIDAR sensors using TCP and UDP communication protocols. The primary objective is to receive, process, and visualize LIDAR scan data in real time. This system involves setting up network connections, handling data packets, calculating CRCs for data integrity, checking if the scan is complete, and visualizing the resulting scan data.



Content of the script folder:

- ROD_demo_script_2D_venv = Python runtime environment folder
- launch_ROD_demo_script_2D.cmd = launch script
- ROD_demo_script_2D.py = demo python script
- ROD_demo_script_2D_req.txt = list of the Python required packages

Note that the folder ROD_demo_script_2D_venv is missing until the script is not executed for the first time.

2. Requirements and preparation

The latest released version of the demo script has been developed and tested with Python version 3.13, so it is recommended to have it installed to execute the demo. This version of Python can be downloaded from official [Python web page](#).

2.1. Launch script

It is recommended to execute the demo using the launch script *launch_ROD_demo_script_2D.cmd*. The launch script creates a virtual environment for Python 3.13 and installs only required packages in specified versions (for more details, see *ROD_demo_script_2D_req.txt*). Please keep in mind that the first execution of the demo will take a little bit longer, because all the required packages are downloaded and installed.

In case you need to reinstall the virtual environment, it is necessary to delete folder *ROD_demo_script_2D_venv* from the script parent folder and execute the launch script again.

To run the demo just double click on the *launch_ROD_demo_script_2D.cmd* file, but if the demo does not run properly, it can be useful to execute it via command line to see an error log.

2.2. Direct execution

In case it is not possible to install Python 3.13., there is still a possibility to run the demo directly by executing Python script *ROD_demo_script_2D.py*. To do that it is necessary to ensure that all required packages mentioned in the file *ROD_demo_script_2D_req.txt* have been installed or you must install them manually.

Since the script does not use any non-standard packages besides matplotlib, it should be sufficient to run the following command in the command line to install it and all its dependencies.

➤ `pip install matplotlib`

To run the demo just double click on the *ROD_demo_script_2D.py* file, but if the demo does not run properly, it can be useful to execute it via command line to see an error log.

2.3. Firewall settings

There is no known problem when running the demo script on a Linux computer, but when running on Windows, it may happen that the graph does not appear after running the script. In this case, UDP data does not go through because of the Windows firewall. It is necessary to allow the application in the firewall.

Click the Start button, then type Windows Firewall in the Search box. Click Windows Firewall, and then click “Allow an app through Firewall”. Click the Change settings button. Find python and enable the public network. See pictures and the following page for more information.

The second option is to simply turn off the firewall on public networks.

It is also possible that a Windows firewall warning appears when you try to run it. Then you just need to enable python on public networks.

Windows Security

←

☰

Home

Virus & threat protection

Account protection

Firewall & network protection

Device security

Device performance & health

Firewall & network protection

Who and what can access your networks.

Domain network (active)

Firewall is on.

Private network

Firewall is on.

Public network (active)

Firewall is on.

Allow an app through firewall

Network and Internet troubleshooter

Firewall notification settings

Advanced settings

Restore firewalls to default

Windows Community videos

Learn more about Firewall & network protection

Have a question?

Get help

Who's protecting me?

Manage providers

Help improve Windows Security

Give us feedback

Change your privacy settings

View and change privacy settings for your Windows 10 device.

Privacy settings

Privacy dashboard

Privacy Statement

Allowed apps

← → ↕

System and Security > Windows Defender Firewall > Allowed apps

Search Control Panel

Allow apps to communicate through Windows Defender Firewall

To add, change, or remove allowed apps and ports, click Change settings.

What are the risks of allowing an app to communicate?

For your security, some settings are managed by your system administrator.

Allowed apps and features:

Name	Domain	Private	Public	Group Policy
<input checked="" type="checkbox"/> Python	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	No
<input checked="" type="checkbox"/> QsConfig.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> QsMainConfigWizard.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> QsTestShellRunner.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> QsTestShellRunner.x86.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> QsTestShellStudio.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> QsTestShellStudio.x86.exe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> Recommended Troubleshooting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
<input checked="" type="checkbox"/> Remote Assistance	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> Remote Assistance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
<input checked="" type="checkbox"/> Remote Desktop	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Yes
<input checked="" type="checkbox"/> Remote Desktop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No

Details...

Remove

Allow another app...

OK

Cancel

1. High-Level Workflow

1.1. Connection setup

The driver connects to the ROD 308 / ROD 508 LIDAR sensor using a TCP interface for sending commands and reading of the sensor configuration and a UDP interface for receiving scan data.

The ROD 308 / 508 default IP address (192.168.60.101) and port (3050) are used, but both can be changed by modifying the Python script.

```
# ROD x08 scanner ethernet configuration
CONST_ROD_IP_ADDR = '192.168.60.101'
CONST_ROD_IP_PORT = 3050
```

1.2. Command handling

User commands are entered through a terminal and can be:

- SendMDI
- StopMDI
- Set commands
 - e.g., resolution – SetResol 2
 - output data port – SetPort 3050
 - filter – SetFilter 1
 - angle range – SetRange -4500 22500
- Get commands
 - e.g., resolution – GetResol
 - output data port – GetPort
 - filter – GetFilter
 - angle range – GetRange

The commands are sent via TCP, and responses are received to confirm the action. A complete list of possible commands is available in the ROD-300-500 Communication Protocol document available at [ROD 308](#) and [ROD 508](#) Leuze web page.

1.3. Data reception and parsing

Data packets are received over UDP. These packets contain information about the LIDAR scan, including distance measurements and angles. The packet is parsed, and the distances are stored for visualization.

The format of the MDI data packet is shown as below:

Header :

SYNC	Packet type	Packet size	Re-served	Re-served	Re-served	Packet NO.	Total NO.	Sub NO.	Scan freq	Scan spots	First angle	Delta angle	Time-stamp
4 Bytes	1Bytes	2Bytes	2Bytes	2Bytes	2Bytes	2Bytes	1Byte	1Byte	2Bytes	2Bytes	4Bytes	4Bytes	2 Bytes

Msg :

Distance	Intensity
Max 700*2Bytes	Max 700*2Bytes

Footer :

CRC
2Bytes

1.4. Data integrity

Each data packet includes a CRC16 checksum. The calculate_crc16 function ensures that the received data is valid and not corrupted by comparing the received CRC value to a calculated one.

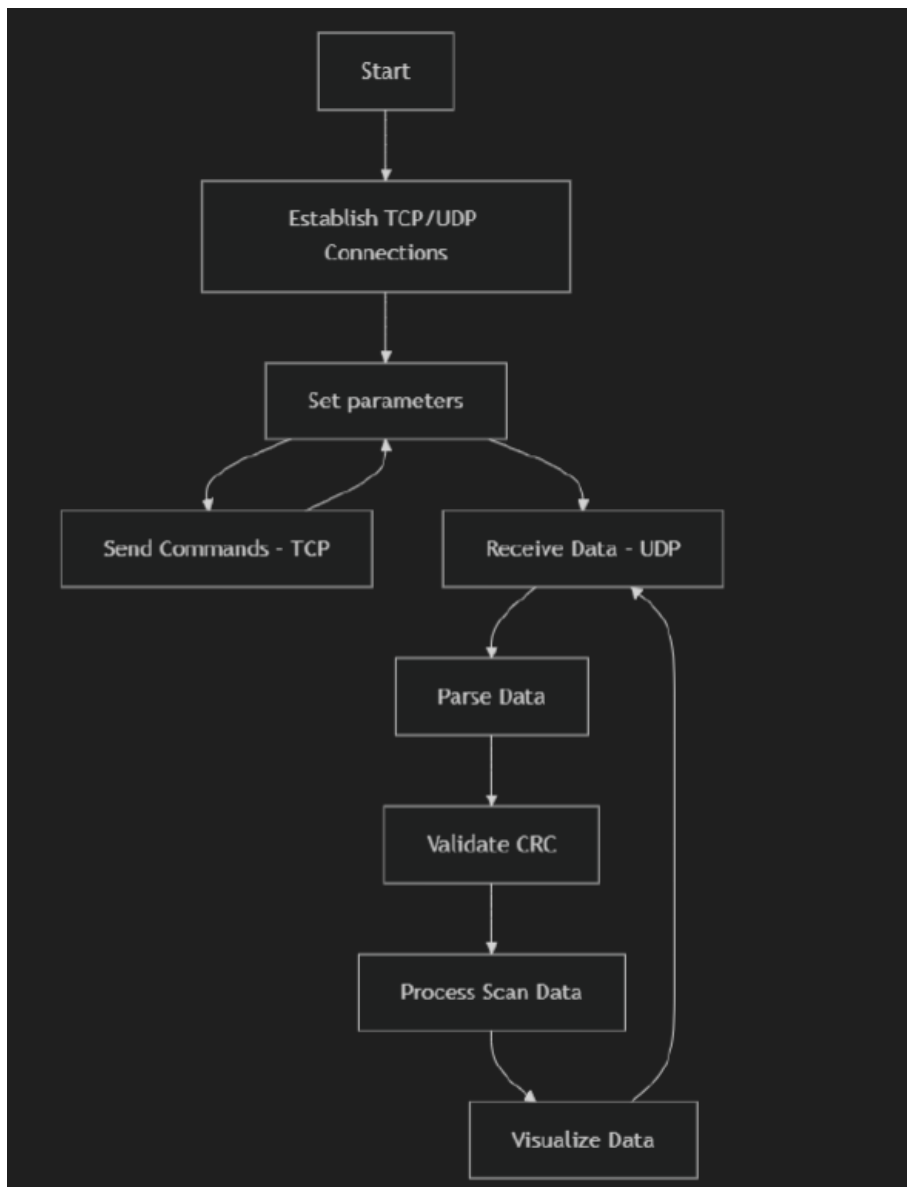
1.5. Visualization

The scan data is visualized using matplotlib. The scan is transformed from polar coordinates (range and angle) to Cartesian coordinates (x, y), and plotted in real-time as a scatter plot.

1.6. Code structure

- **Start:** The program starts and establishes the TCP and UDP connections to the LIDAR sensor.
- **Send Commands:** Commands are sent to the sensor via TCP, such as setting resolution or retrieving configuration.
- **Receive Data:** The program listens for UDP data packets containing the LIDAR scan data.
- **Parse Data:** Each received packet is parsed, and the data is extracted.
- **Validate CRC:** The packet's CRC is validated to ensure the data integrity.
- **Process Scan Data:** The scan data (distance and angle) is processed.
- **Visualize Data:** The scan data is visualized in real-time using matplotlib.

- **Loop:** The process repeats, continuously receiving and visualizing data. In parallel, it listens to see if the user has entered a command.



2. Changelog

- **07-03-2025**
 - First release version
- **16-07-2025**
 - Support for commands “SendMDI” and “StopMDI” added
 - TCP command interface response parsing improved
 - Console logs improved
 - Visualization improved
 - Scan orientation for the latest ROD 308 / ROD 508 scanner fixed
 - Visualization window is not locked on the top anymore
 - Visualization window and plot ranges are defined by constants and can be modified easily
 - Matplotlib warnings have been suppressed
 - Launch script created
- **18-07-2025**
 - Simple help added
 - Given parameter values print bugfix