ESLAB
Embedded System Lab

XM125 Test

Taehun Kim
bigteach0508@pusan.ac.kr
Embedded Systems Lab.
School of AI, PNU
2025. 03. 12

Contents

- Schematic
- Installation
- GUI results
- CLI installation
- CLI results
- Conclusion

Schematic

- Sparkfun XM125

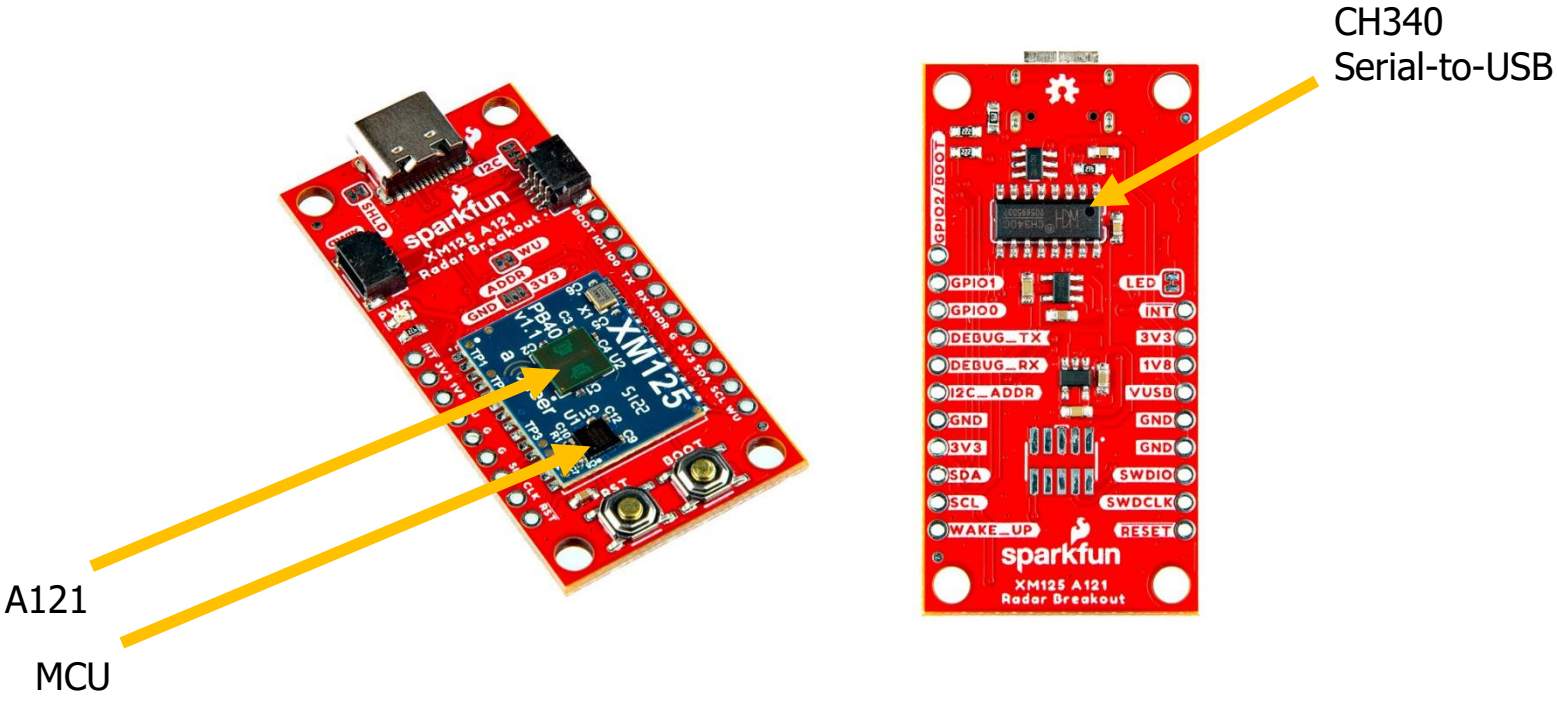


Figure 1. Sparkfun XM125

Schematic (cont'd)

- XM125 block diagram

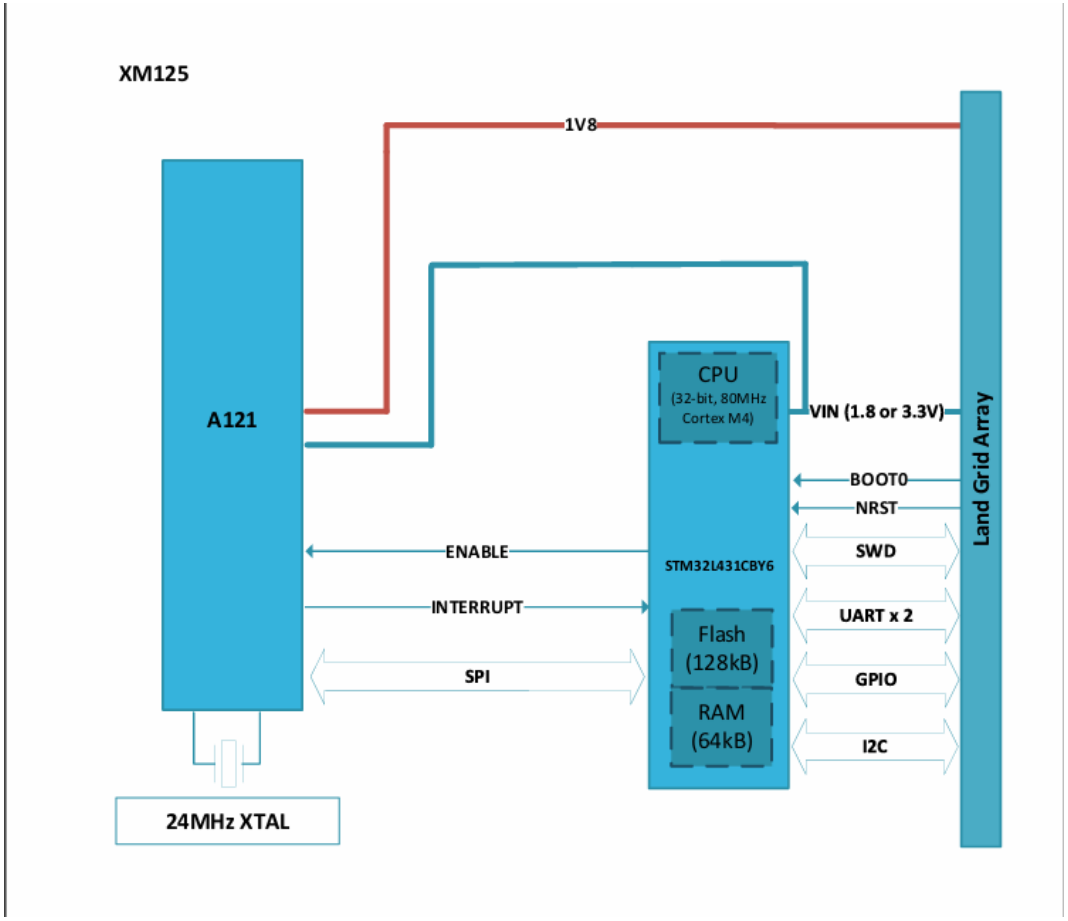


Figure 2. XM125 block diagram

Schematic (cont'd)

- A121
  - 1 Tx antenna and 1 Rx antenna
  - Serial Peripheral Interface(SPI) communication with MCU

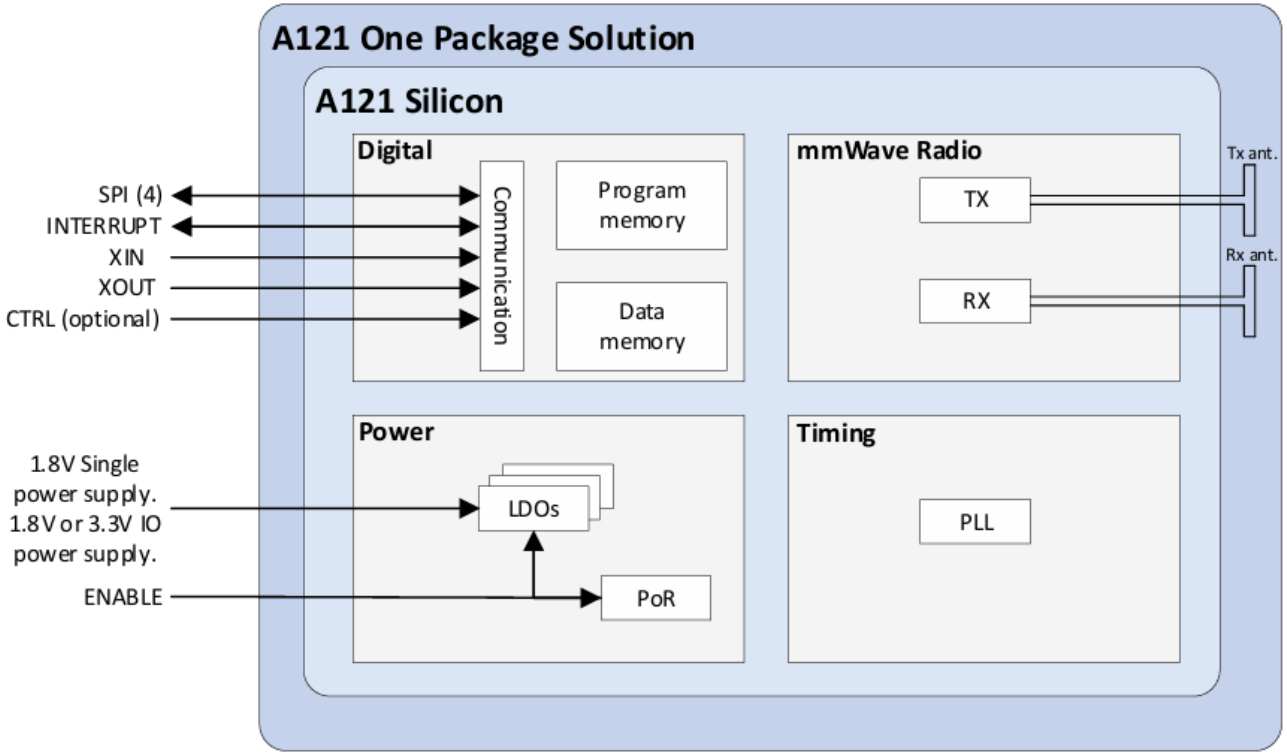


Figure 3. A121 block diagram

Installation

- Install CH340 driver
  - https://www.arduined.eu/ch340-windows-10-driver-download/
- Download Acconeer Exploration Tool
  - https://developer.acconeer.com/download/portable_exploration_tool/
- Double click the update.bat file and the run_app.bat

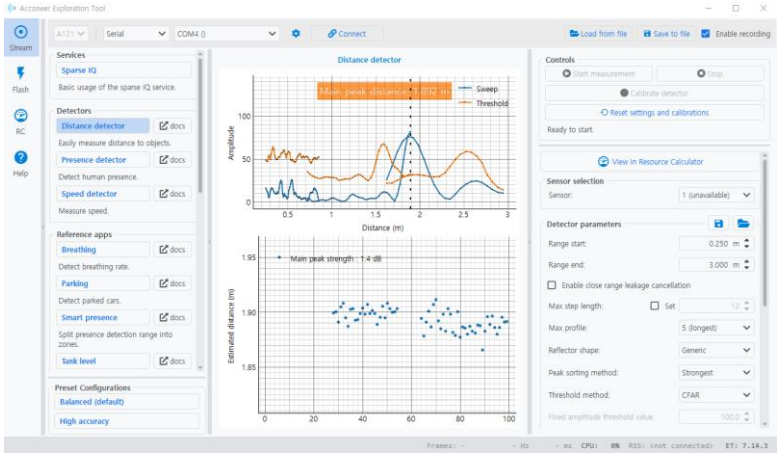


Figure 4. Acconeer Exploration Tool

Installation (cont'd)

- Flashing firmware
  - Click the flash menu
  - Click Get latest binary – Get latest bin file
  - Register at https://developer.acconeer.com/register/
  - Download the firmware
  - Click the flash button to flashing firmware

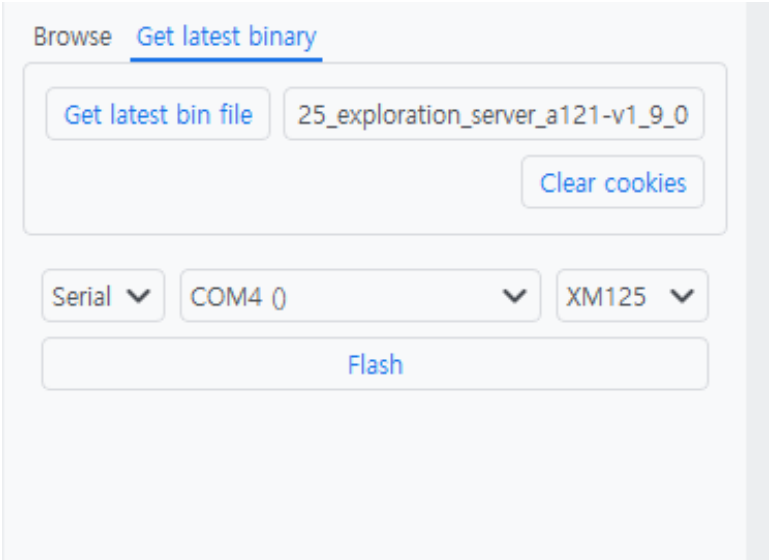


Figure 5. Download the firmware

Installation (cont'd)

- Baud rate setting
  - Click the setting(cogwheel) button
  - Setting the baudrate to 115200

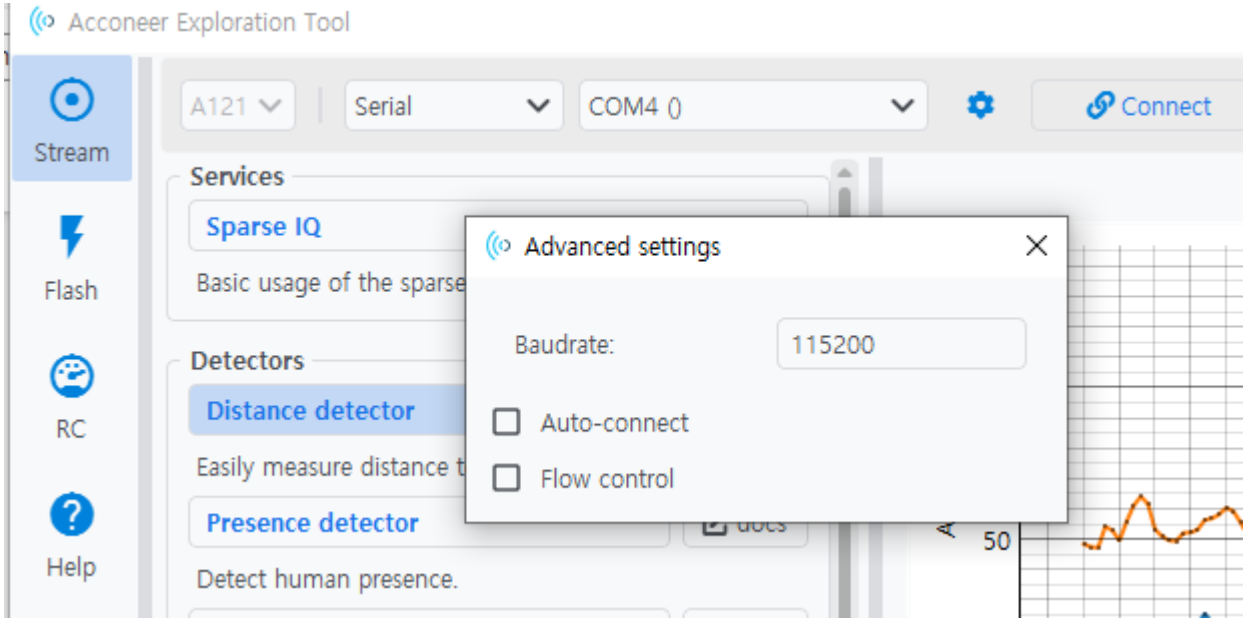


Figure 6. Setting the baudrate

GUI results

- Sparse IQ

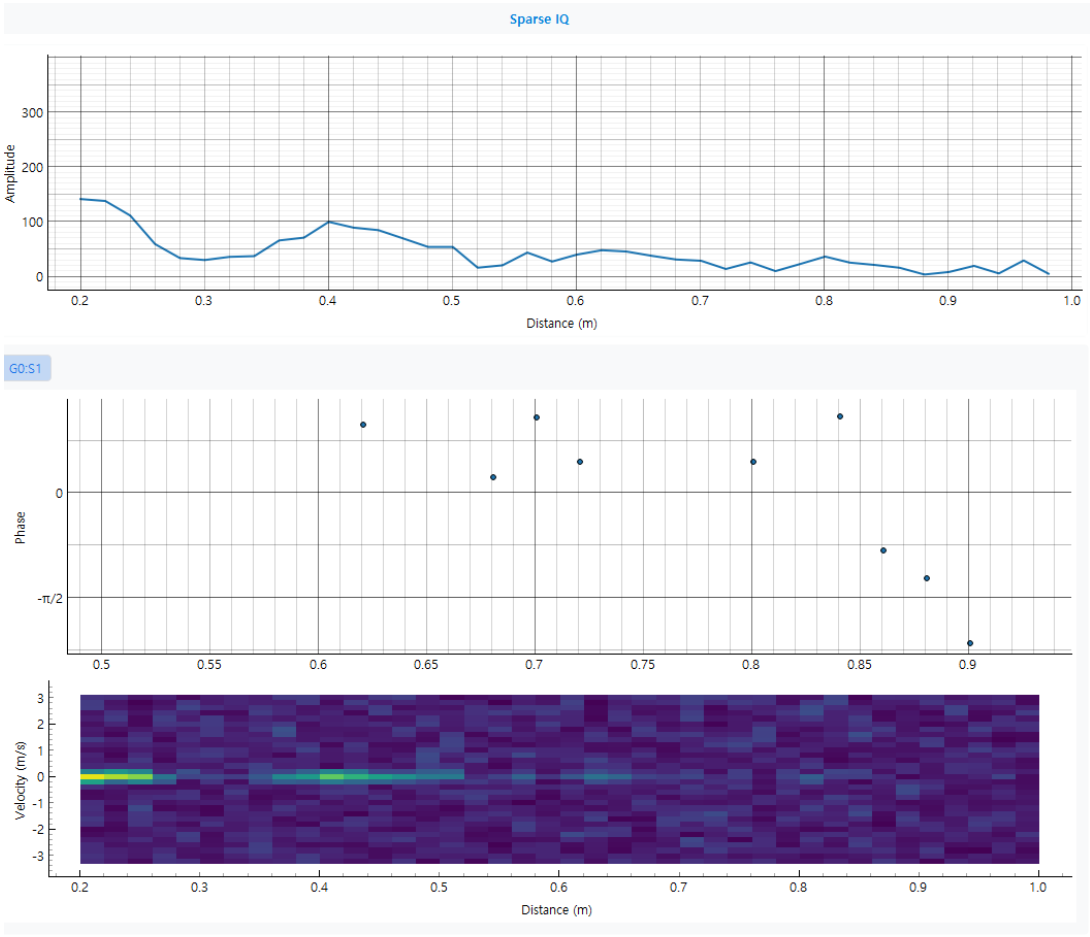


Figure 7. Sparse IQ

GUI results (cont'd)

- Distance detector

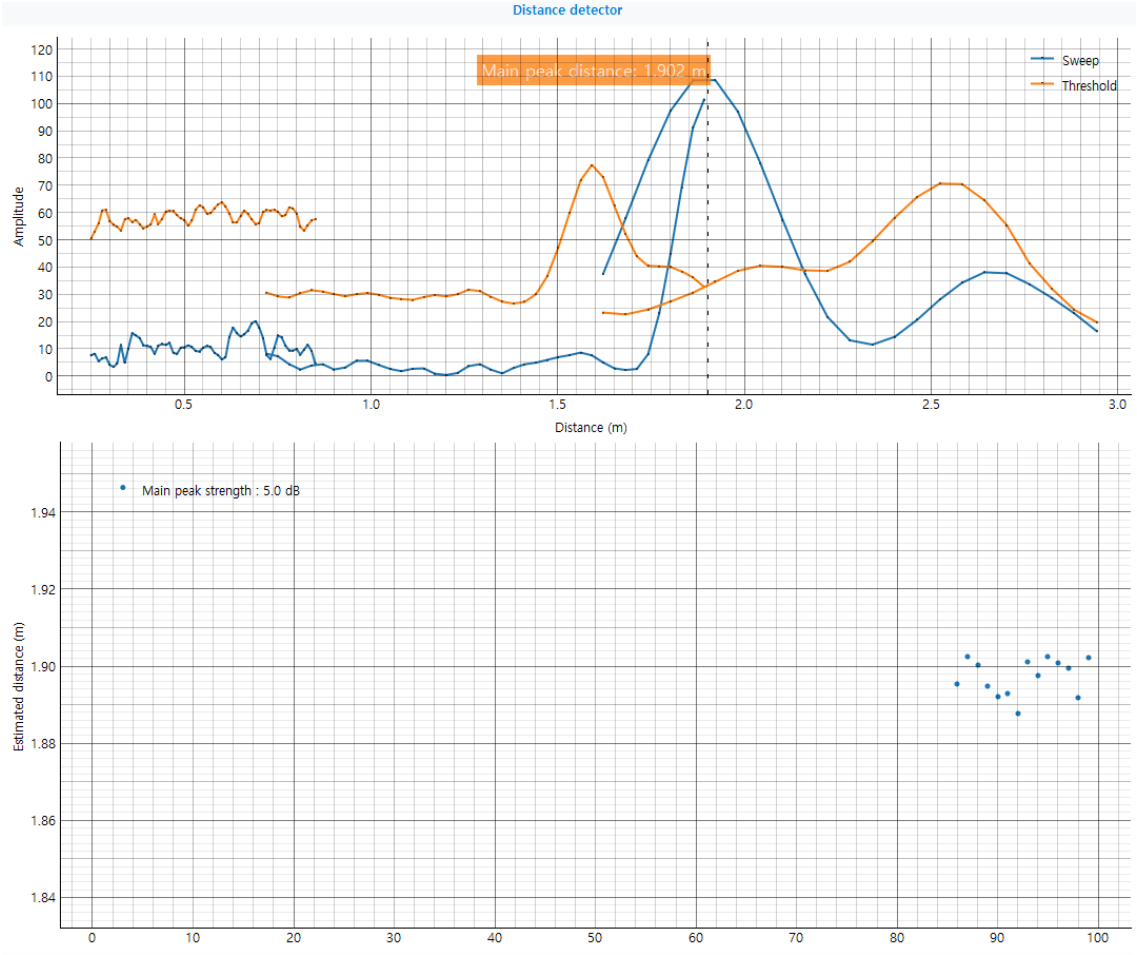


Figure 8. Distance detector

GUI results (cont'd)

- Presence detector

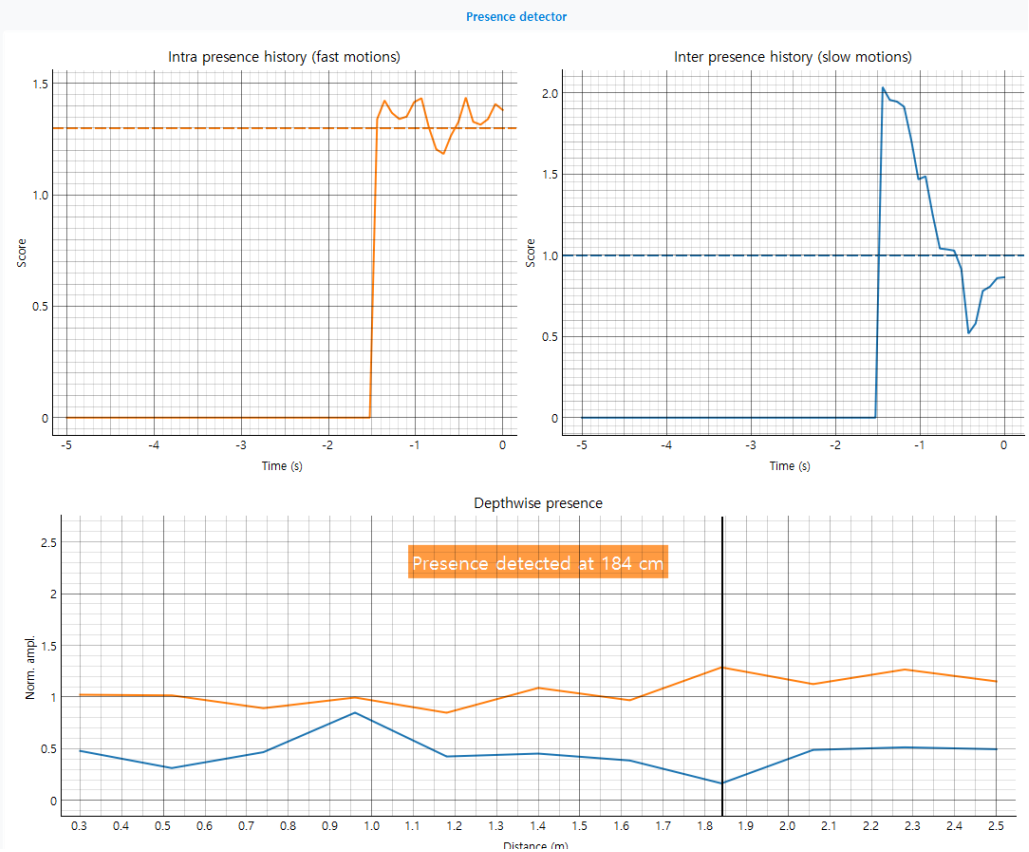


Figure 9. Presence detector

GUI results (cont'd)

- Speed detector

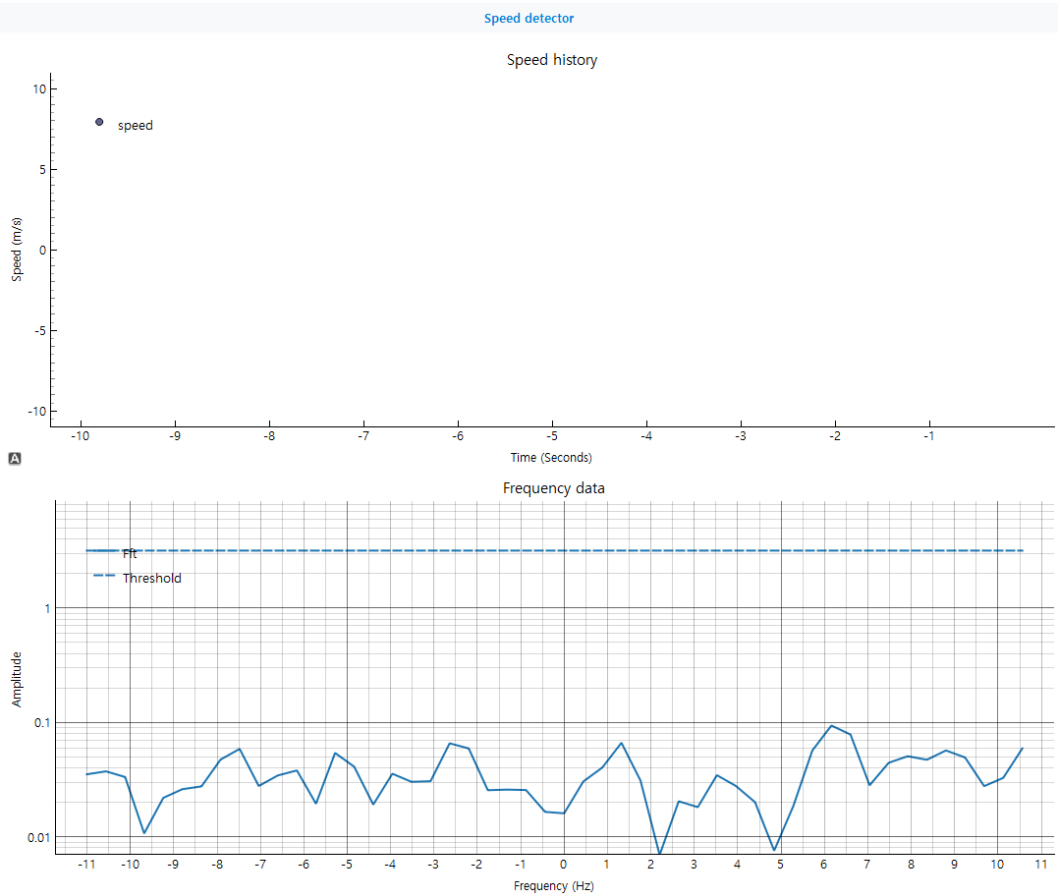


Figure 10. Speed detector

CLI installation

- Get example source code
  - https://github.com/acconeer/acconeer-python-exploration/tree/master/examples
- Double click the cmd_with_path.bat in portable_exploration_tool
- Set client information
  - Serial port and baudrate
- Run the source code
  - python examples/a121/basic.py

```
client = a121.Client.open(
    # ip_address="<ip address o
    # or
    # serial_port="<serial port
    # or
    # usb_device=True,
    # or
    # mock=True,
    serial_port='COM4',
    override_baudrate=115200
)
```

Figure 11. Serial Configuration

CLI results

- Basic.py
  - print real parts and imaginary parts of the receive signals
    - Sweep_per_frame x num_points

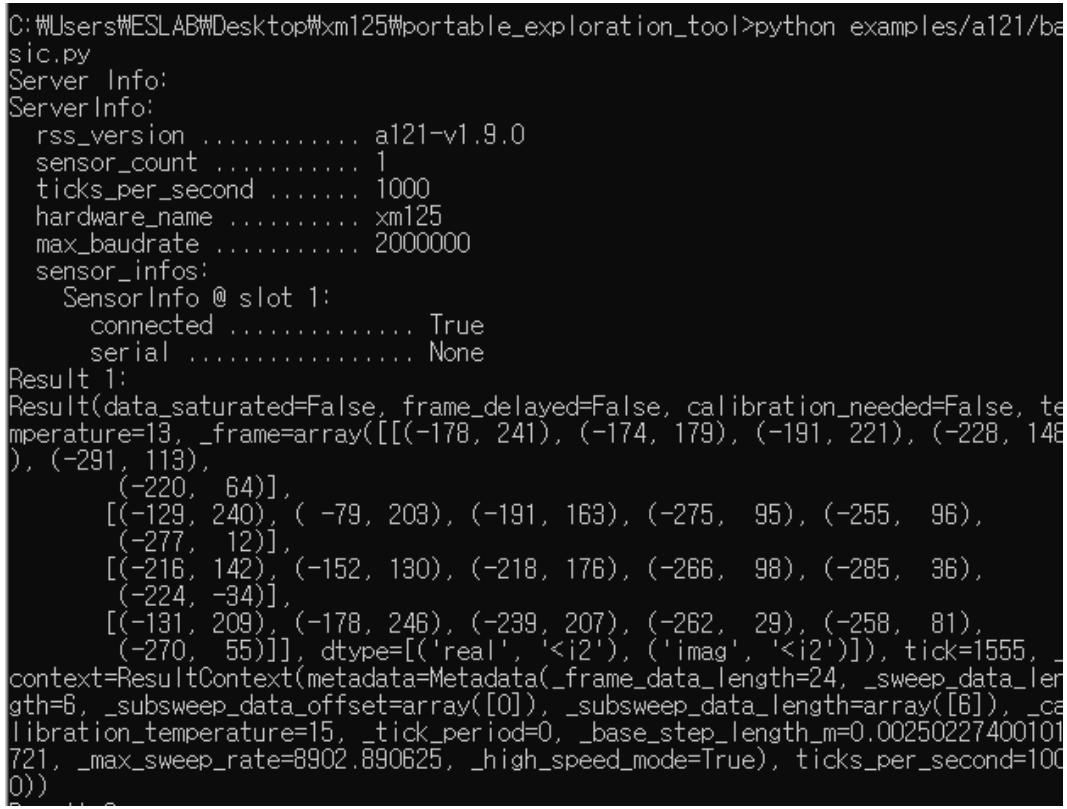


Figure 12. Real and imaginary part of signals

CLI results

- basic_plot.py

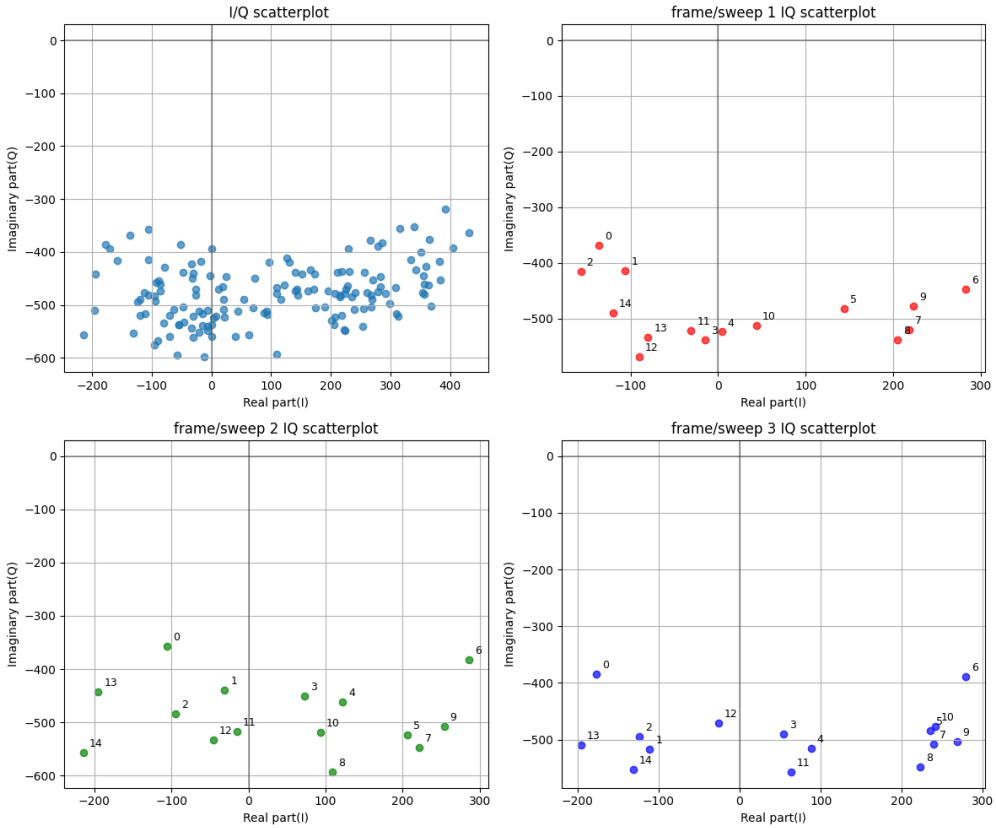


Figure 13. I/Q scatterplot

CLI results (cont'd)

- basic_plot.py

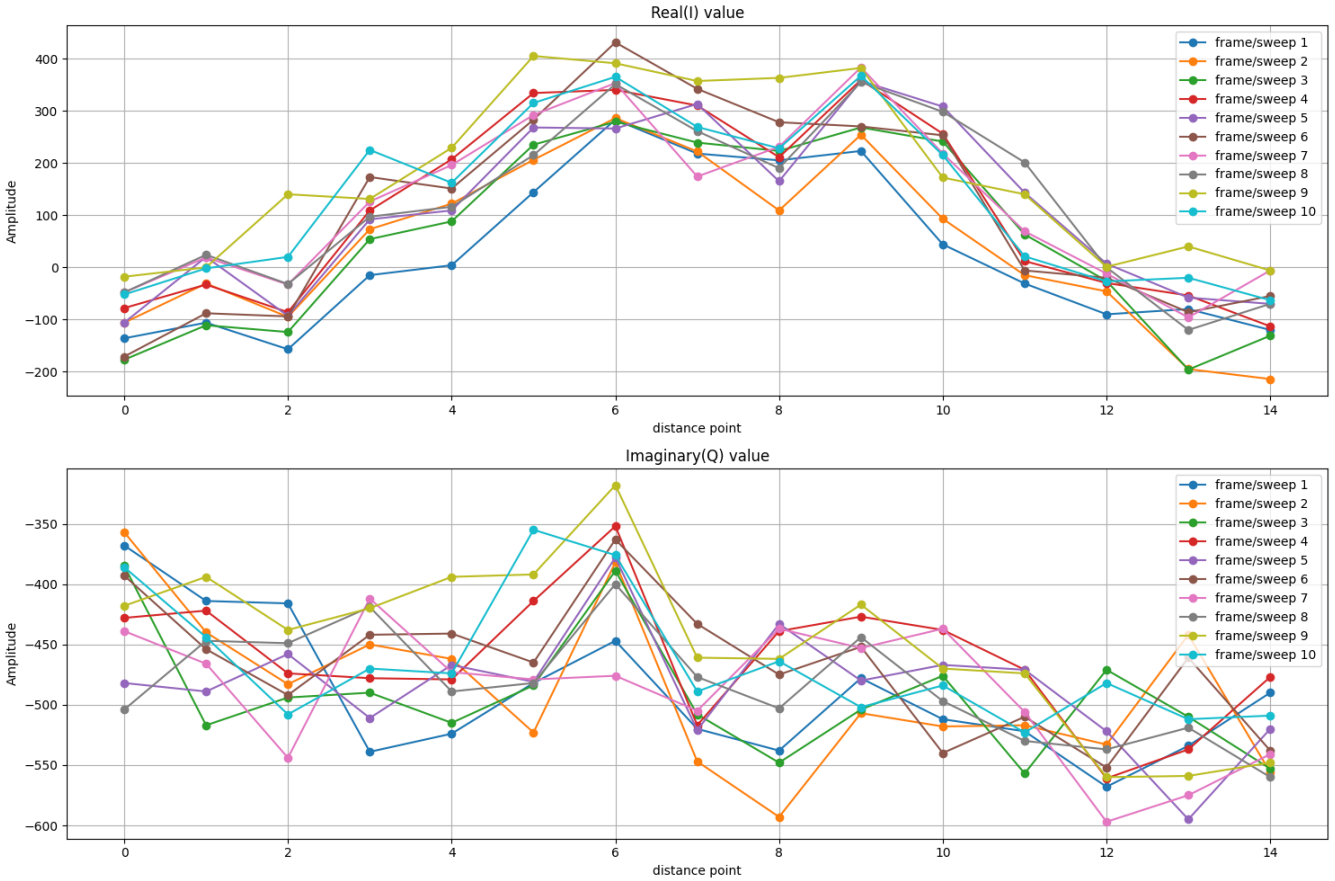


Figure 14. real and imaginary value by sweep and distance point

CLI results (cont'd)

- Basic_plot.py

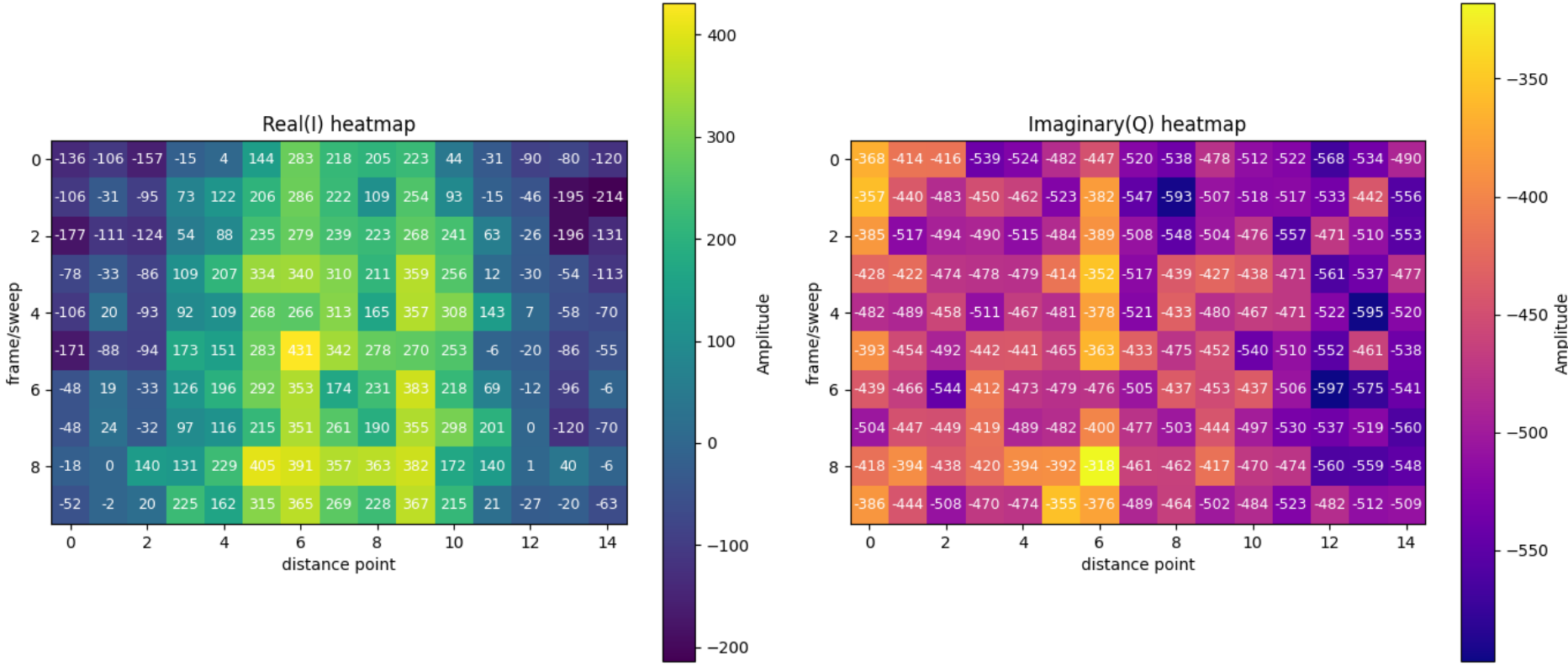


Figure 15. real and imaginary heatmap by sweep and distance point

CLI results (cont'd)

- Basic_plot.py
  - $Amp = \sqrt{I^2 + Q^2}, Phase = \arctan\frac{Q}{I}$

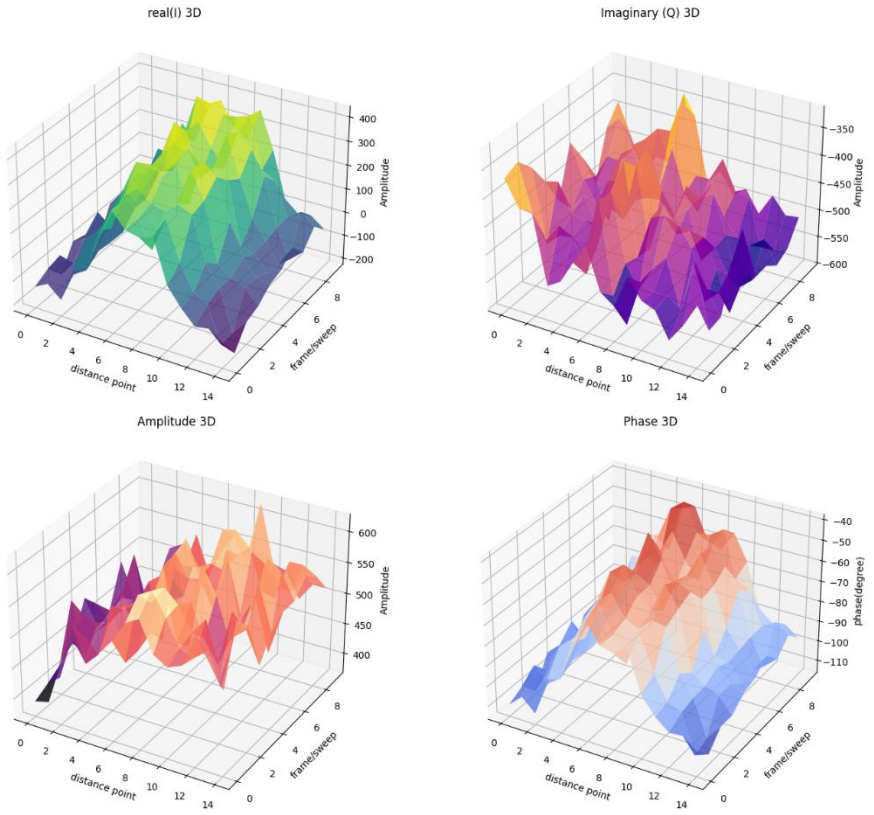


Figure 16. real, imaginary, amplitude, phase heatmap by sweep and distance point

CLI results (cont'd)

- basic_plot.py

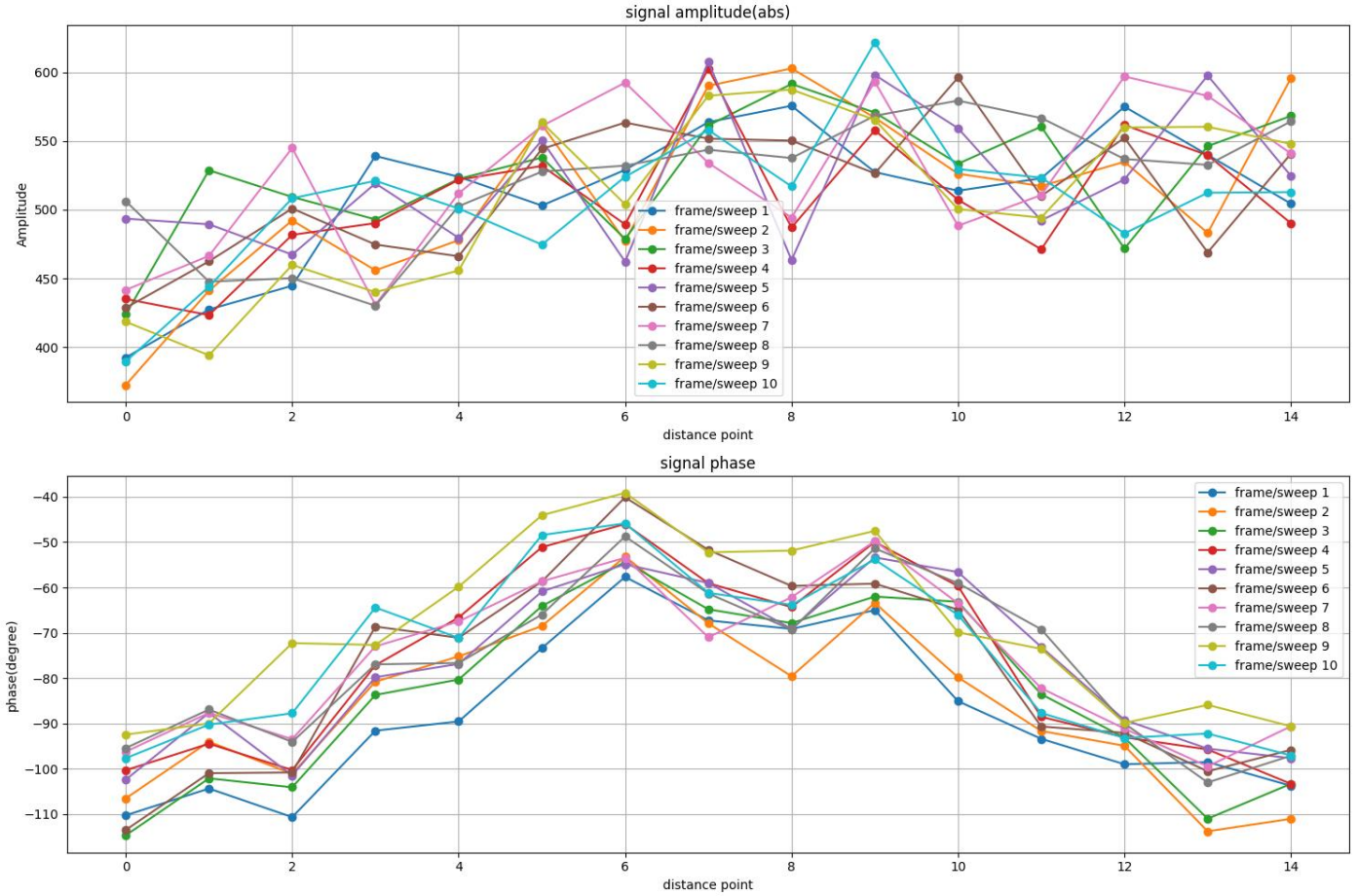


Figure 17. Amplitude and phase value by frame and distance point

CLI results (cont'd)

- Plot_analog.py

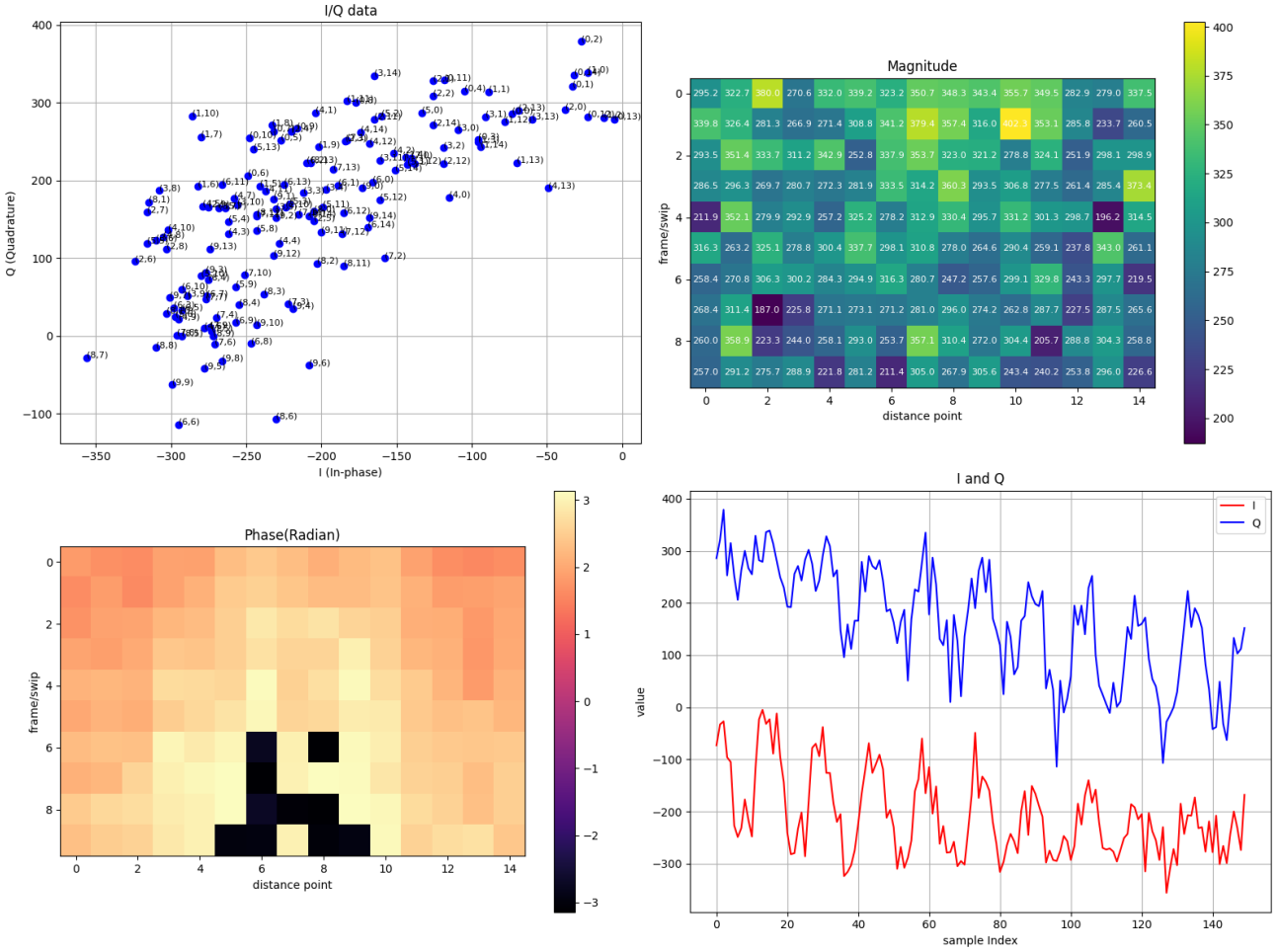


Figure 18. plot I/Q data, magnitude heatmap, phase heatmap, and I Q line plot

CLI results (cont'd)

- Plot_analog.py

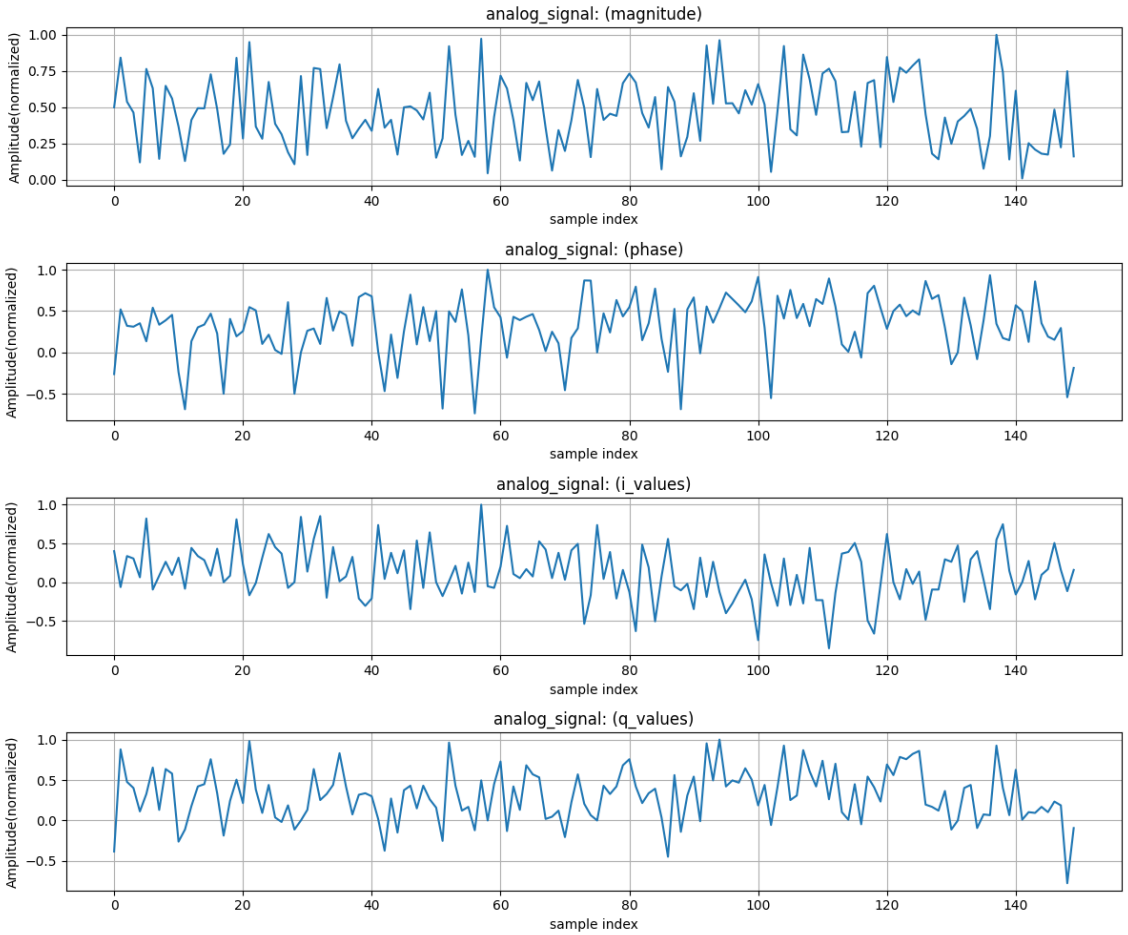


Figure 19. plot magnitude, phase, i_values, q_values after flattening signal

CLI results (cont'd)

- plot_analog2.py

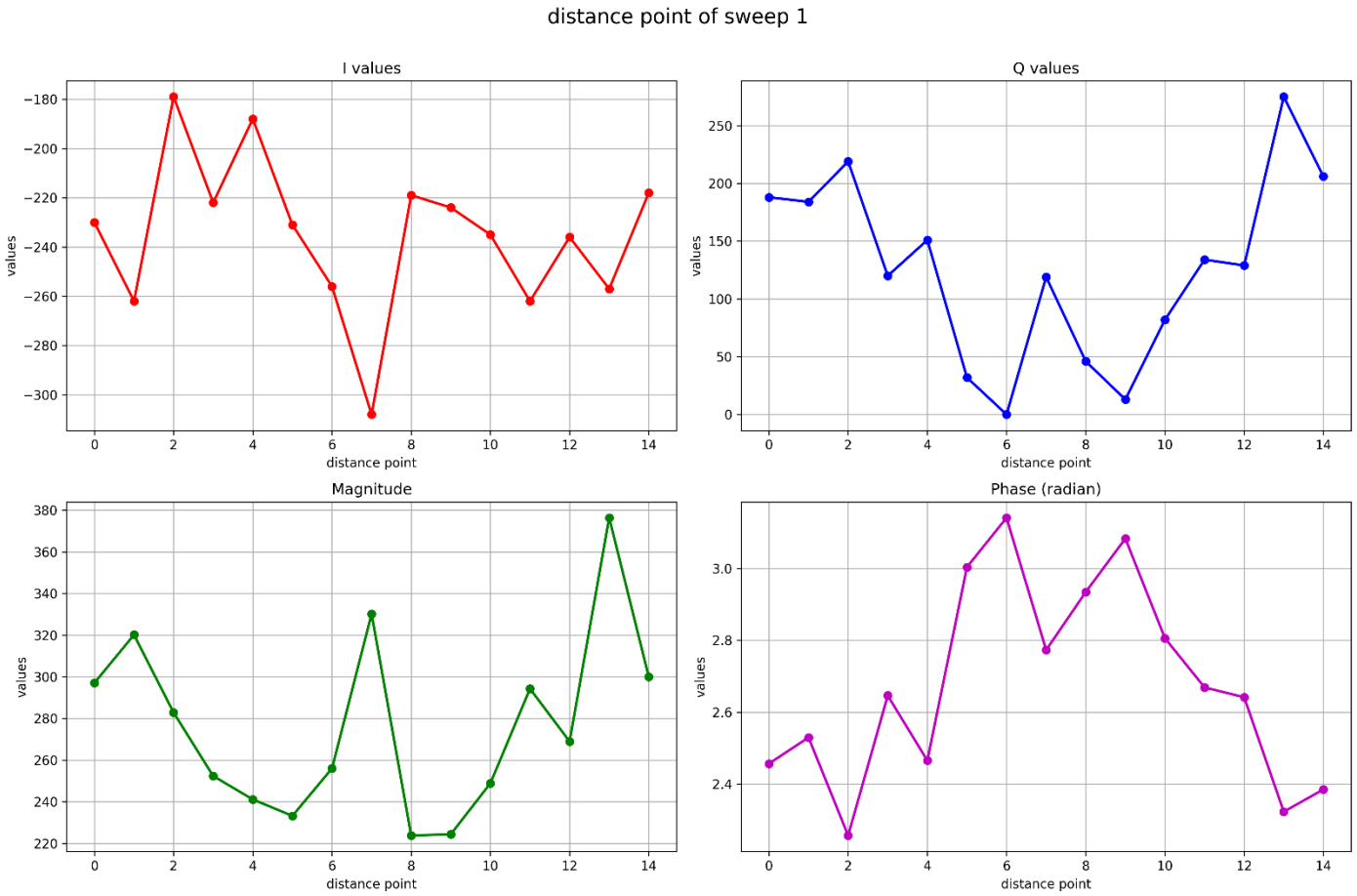


Figure 20. Distance point of sweep 1

CLI results (cont'd)

- plot_analog2.py

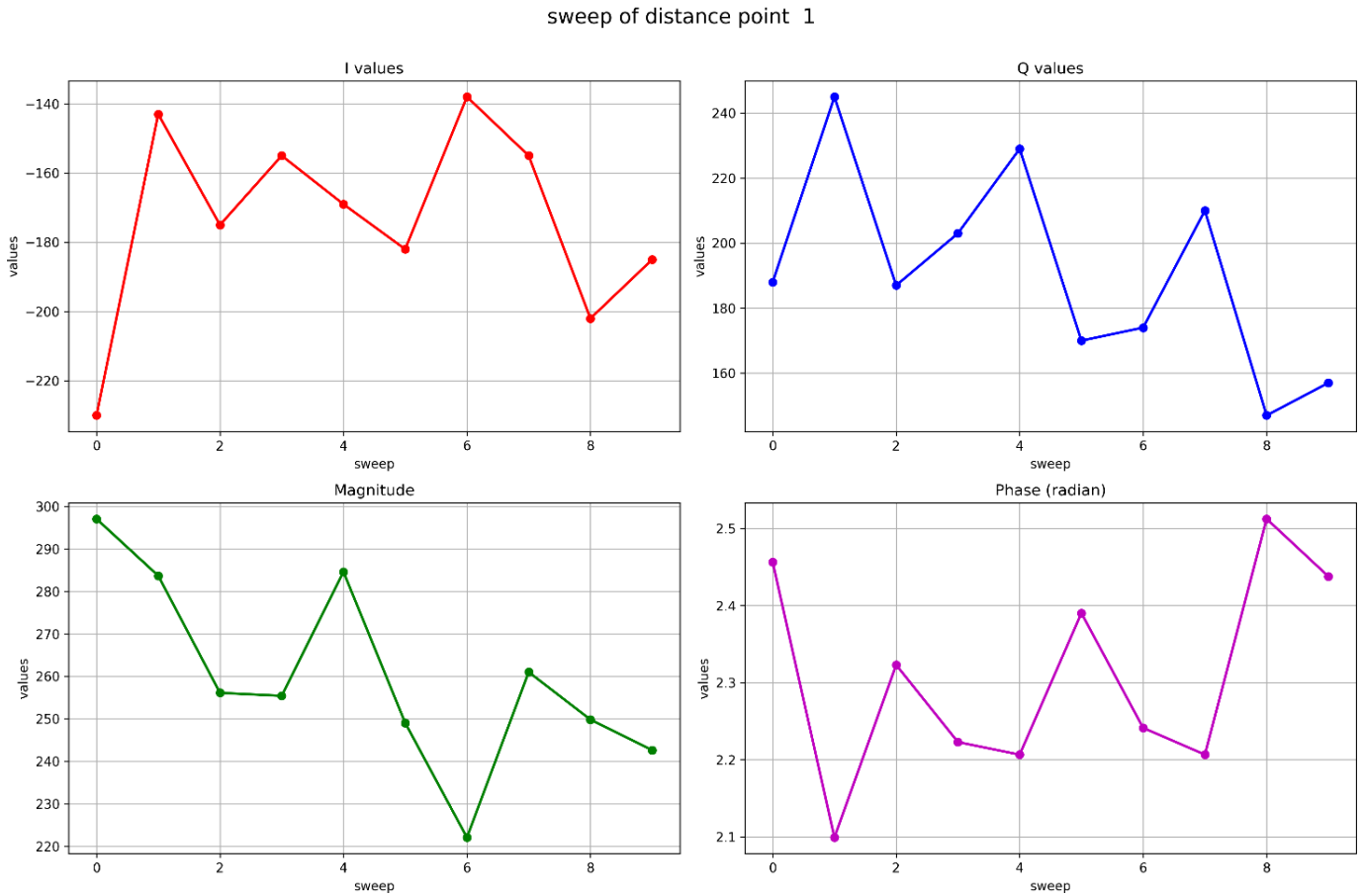


Figure 21. Sweep of distance 1

CLI results (cont'd)

- Distance measurement
  - Modify examples/a121/algo/distance/processor.py

```
C:₩Users₩ESLAB₩Desktop₩xm125₩portable_exploration_
mples₩a121₩algo₩distance₩processor.py
qt.core.qobject.connect: QObject::connect(QStyleH
tion of a QObject subclass
Press Ctrl-C to end session
distance: 0.307347947699052
distance: 0.31169640669684523
distance: 0.30788424648690377
distance: 0.3124415844453782
distance: 0.31049557093571317
distance: 0.3108571014046115
distance: 0.31328463547220003
distance: 0.3182344253138537
```

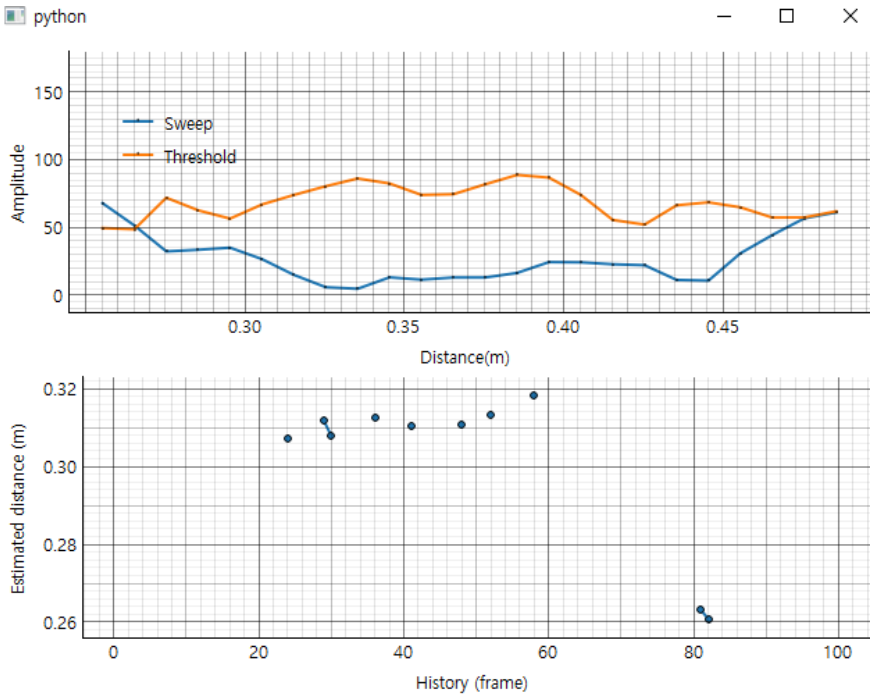
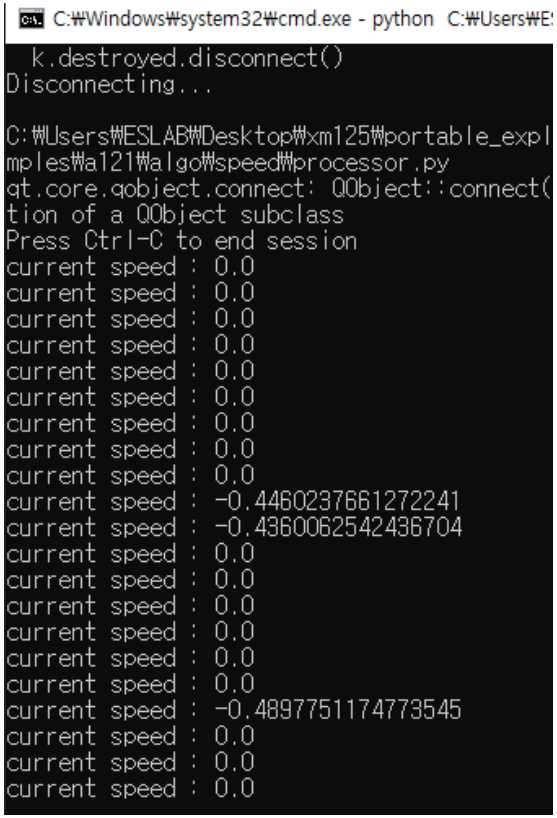


Figure 22. Distance measurement

CLI results (cont'd)

- Speed measurement
  - Modify examples/a121/algo/speed/processor.py



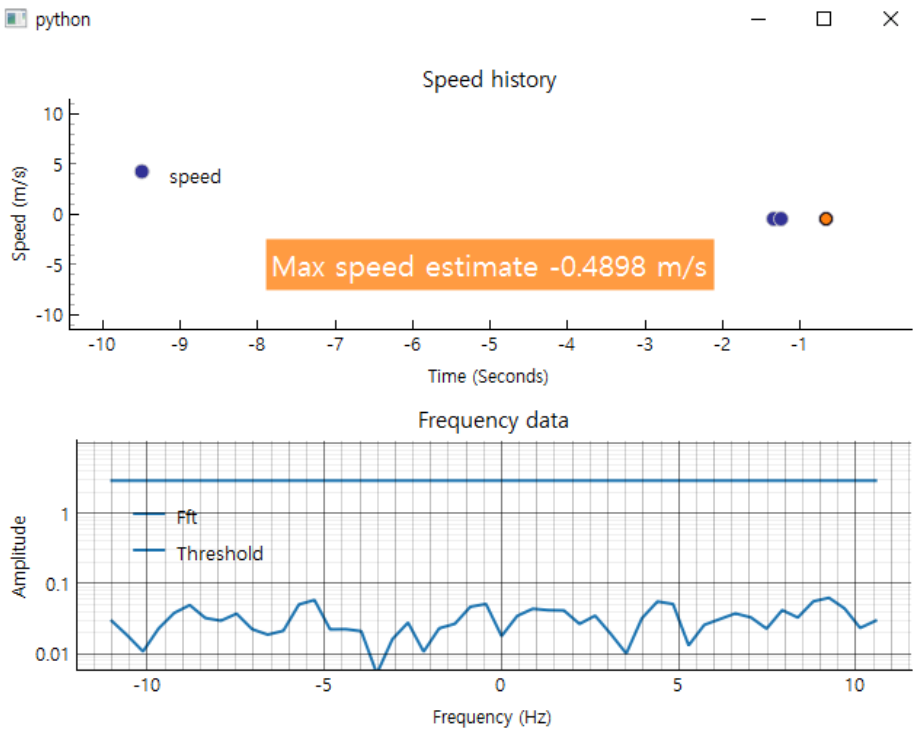


Figure 23. Speed measurement

CLI results (cont'd)

- Sparse IQ
  - Modify example/a121/algo/sparse_iq/sparse_iq.py
  - Range-doppler heatmap
    - Range: from start_point*0.0025 to start_point*step_size*num_points
      - Step_size = step_length*0.0025
    - Velocity: +- wavelength/(4*sweep_period)
      - Sweep_period=1.0/sweep_rate
      - Step_size=sweeps_per_frame

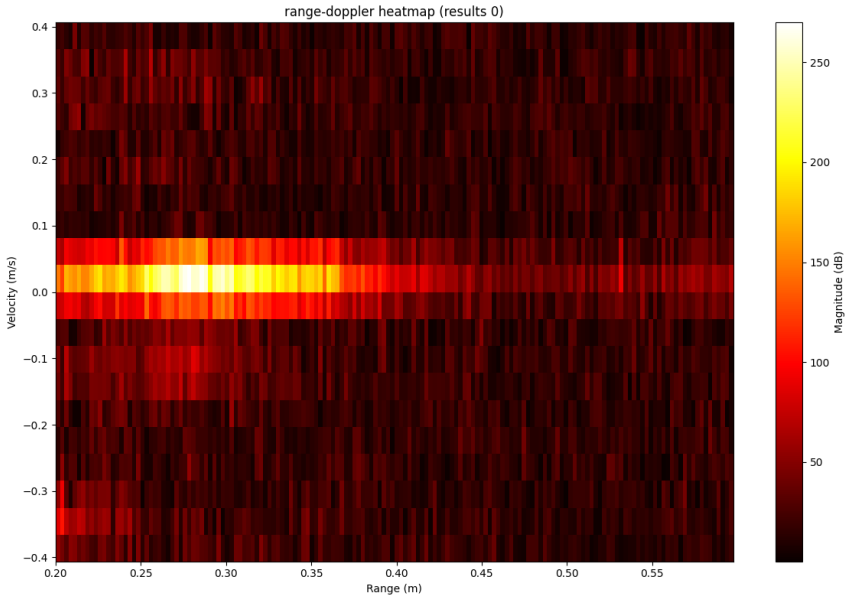


Figure 24. Range-doppler heatmap of result 0

Conclusion

- XM125 sensor cannot measure the Angle of Arrival(AoA) because it has only one Rx antenna
- Analog data may not be directly extracted from the sensor
- However, digital signal $I + Qj$ can be extracted from the sensor of sweep and distance point
- Using the complex number, amplitude and phase can be calculated
- Heatmap can be created from the digital signal
- Distance and speed data can be calculated

References

- Manual
  - XM125 datasheet
  - XM125 schematic

- Online
  - https://github.com/acconeer/acconeer-python-exploration
  - https://docs.acconeer.com/en/latest/exploration_tool/api/a121.html
  - https://matplotlib.org/stable/users/index.html
  - https://matplotlib.org/stable/gallery/mplot3d/index.html

Appendix

- Acconeer software offers
  - Service output is radar data with some pre-processing
  - Detector output is based on service output but uses further processing to create a result such as a distance or presence detection

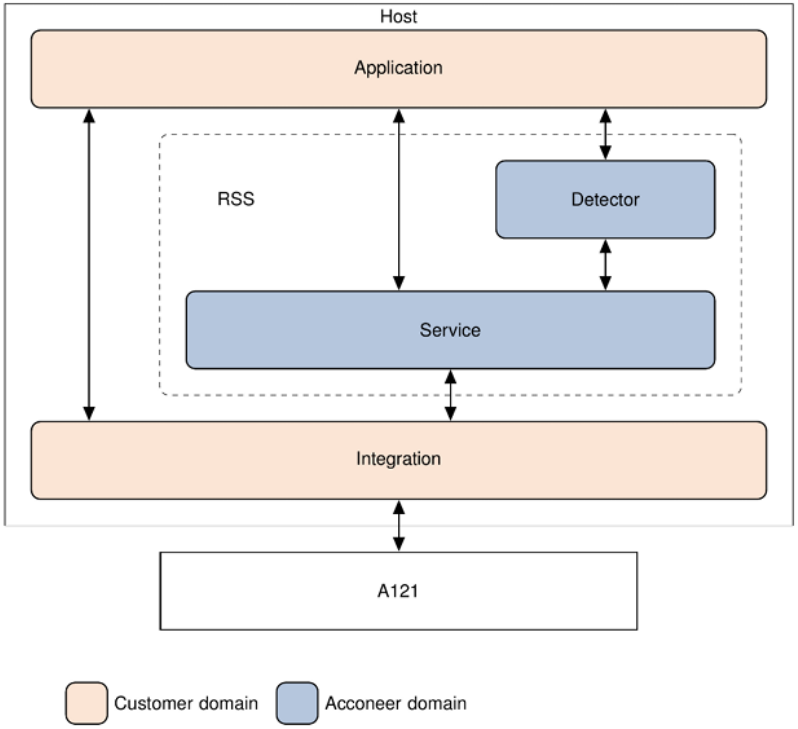


Figure 25. Acconeer software offers

Appendix (cont'd)

- Custom Range FFT

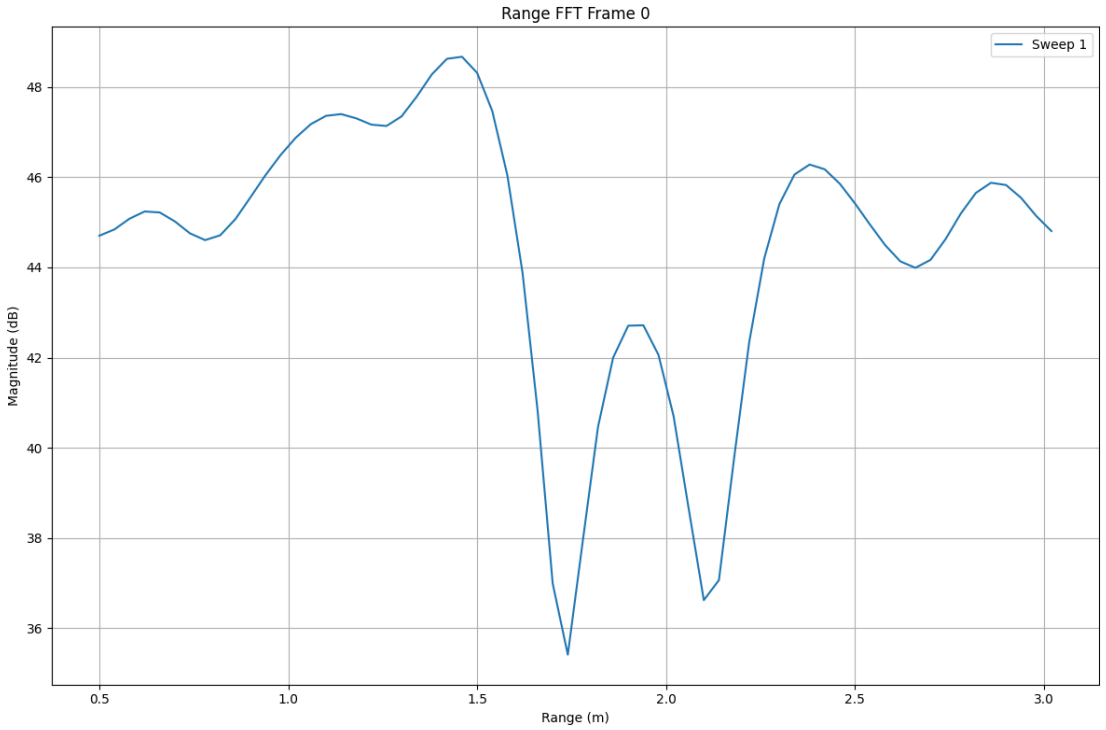


Figure 26. Range FFT

Appendix (cont'd)

- Custom range-doppler heatmap

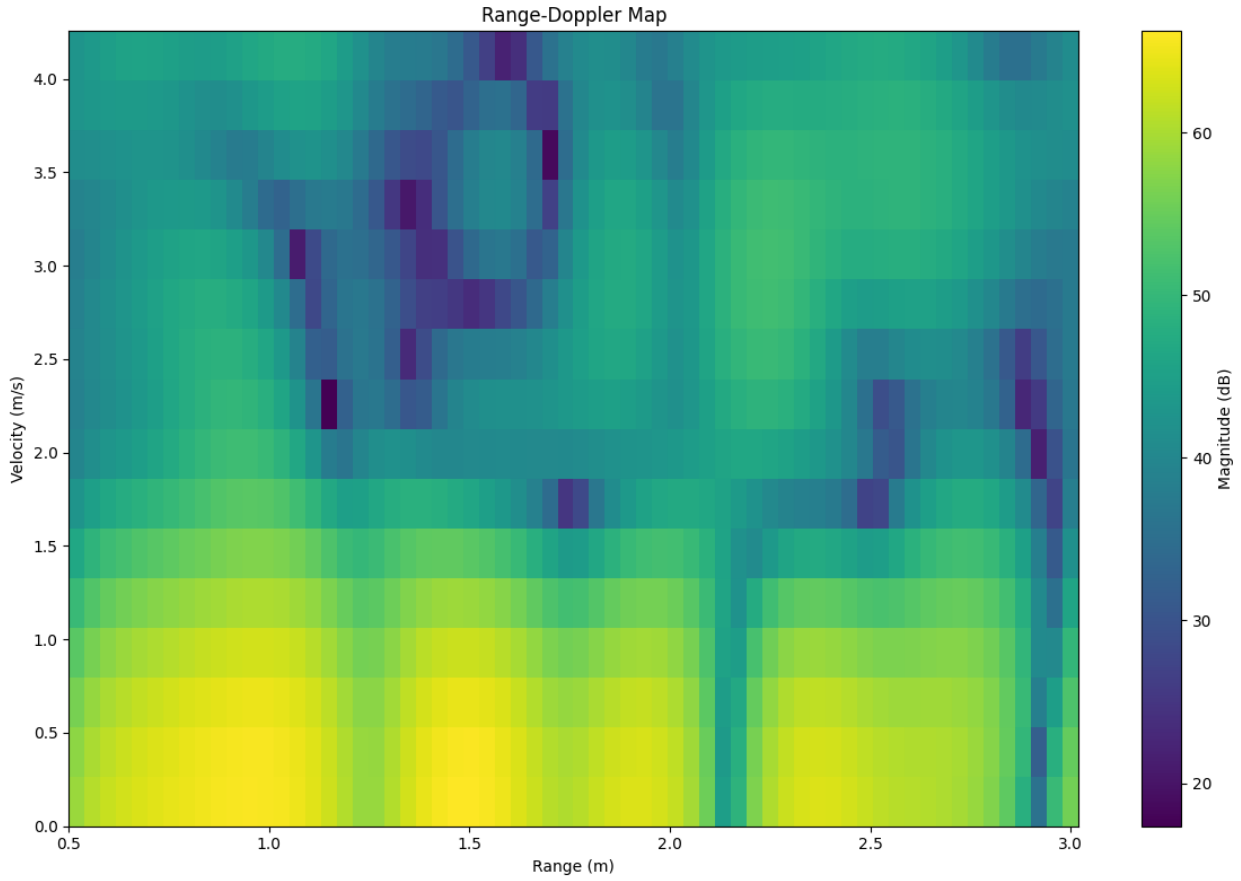


Figure 27. Range-Doppler FFT heatmap