Real-Time Rendering of 3D LiDAR Data

Nathan McGillicuddy (Nathan.McGillicuddy@unh.edu)
Advisor: Dr. MD Shaad Mahmud
Department of Electrical and Computer Engineering, University of New Hampshire

University of New Hampshire

Introduction

Goal: Design and build a wireless LiDAR device that can measure, transfer, and plot live distance and angular measurements three-dimensionally to represent the surrounding environment.

Motivation: Current LiDAR scanning devices typically gather data and post-process after sample collection. The main goal of this project is to improve speed and reduce the processing power needed to compute a render in real time.

System Design

Embedded system control in ESP32 microcontroller

- Written in C++
- Controls A1M8 LiDAR and two TF Luna sensors via UART
- A1M8 LiDAR measures radial coordinates, one TF Luna measures offset and the other measure z-axis
  - UART configuration is 8N1
- Transmits data via UDP to PC
  - Serial transmission was used during testing

Graphical interface and processing software

- Written in Python
- Requests data from ESP32 via UDP
- Visualizes readings via PyQt

	SLAMTEC A1M8 RPLIDAR	TF Luna LiDAR
Rotation Frequency (Hz)	5	-
Measurement Frequency (Hz)	1,800	100
Distance Range (m)	0.15-12	0.2-8
Accuracy	1% if < 3m, 2% if 3-5m, 2.5% if > 5m	±6cm (0.2-3m), ±2% (3-8m)

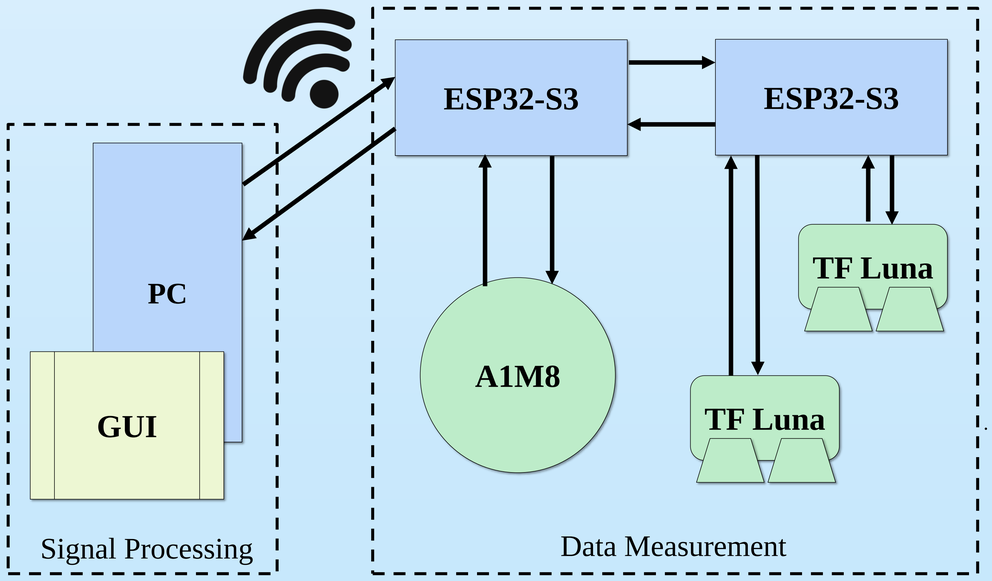
Methodology

Data measurement:

- One master ESP32-S3 for data collection from the A1M8 and a secondary microcontroller for polling from the two TF Luna sensors
- Gathered data values are delta encoded as integers
- A packet is prepared by encoding the data in binary and then sent to the workstation over the corresponding mobile port and IP in little endian format

Signal Processing:

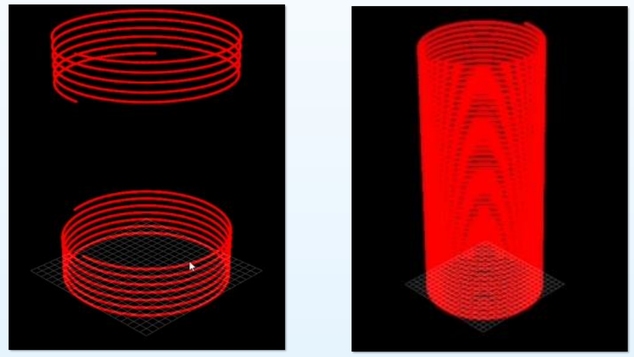
- A socket is bound to the matching mobile port and IP as the main ESP32-S3
- The socket is setup with a buffer to operate as a cache
- Data is decoded from binary and formatted as an integer
- Decoded data is stored in a buffer and updated in batches to reduce plot GUI updates



Results

- Plotting via Serial workstation-lidar system connection functions up to 1,800 data points per second on 115,200 baud with A1M8 pollable at 1,800Hz.
- Plotting via UDP workstation-lidar system connection functions up to 150 data points per second. Limited by mobile network speed

- 1,800 data point per second stress test via Serial connection to LiDAR system. Without a buffer (left) compared to with a buffer (right).



Obstacles/Challenges

- Originally used Processing and then MATLAB for GUI but proved to be too slow and resource intensive
- External library for RPLIDAR produced UART connection issues between the ESP32 and A1M8
- Low power devices do not have as much processing power

Future Advancements

- Construct a 3D printed chassis
- Mount the measuring unit to a drone within different environments and distances
- Increase main ESP32 baud rate to increase transmission rate
- Translate the system horizontally and observe differences in efficiency

References

RoboPeak RPLIDAR Driver for Arduino and Arduino-Compatible Devices
https://github.com/robopeak/rplidar_arduino