System Design for LiDart

Team 10
Jonathan Casella
Kareem Elmokattaf
Michaela Schnull
Neeraj Ahluwalia

January 7, 2023

1 Revision History

Date	Version	Authors	Notes
18/Jan/2023	1.0	Michaela Schnull Jonathan Casella Kareem Elmokattaf Neeraj Ahluwalia	Initial Release

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
BOM	Bill of Materials
CAD	Computer Aided Design
GUI	Graphical User Interface
LiDAR	Light Detection, Imaging, and Ranging
MG	Module Guide
MIS	Module Interface Specification
SRS	System Requirements Specification
USB	Universal Serial Bus

Contents

List of Tables

List of Figures

3 Introduction

LiDart is a low cost, simple to use, 3D scanning robot. The LiDart system uses of low cost LiDAR sensors, consumer grade web cams, and inexpensive location markers. The user interfaces with the robot through GUI that allows them to remotely drive the robot and perform 3D scans. This document provides detailed design specifications for LiDart

The rest of the document is organized as follows. Section 5 shows the boundaries between the system and it's environment. Section 6 describes the components and behaviour of the design implementation. Section 7 specifies the monitored variables, controlled variables, and constants. Sections 8, 9, 10, & 11 describe the design of the user interface, mechanical hardware, electrical components, and communication protocols respectively. Section 12 provides timeline that will be used to implement the design solution.

4 Purpose

This document describes the system functionality and design details of LiDart. Additional documents are used to specify the software architecture and design. The Module Interface Specification (MIS) [1] details the software architecture and the Module Guide (MG) [2] specifies the software detailed design. Requirements from the System Requirements Specification (SRS) [3] are traced to the design implementation to ensure that the solution meets all requirements.

5 Scope

Figure 1 depicts the boundary between the LiDart system and its environment. Any functionality not within the system boundary is out of the scope of this design documentation.

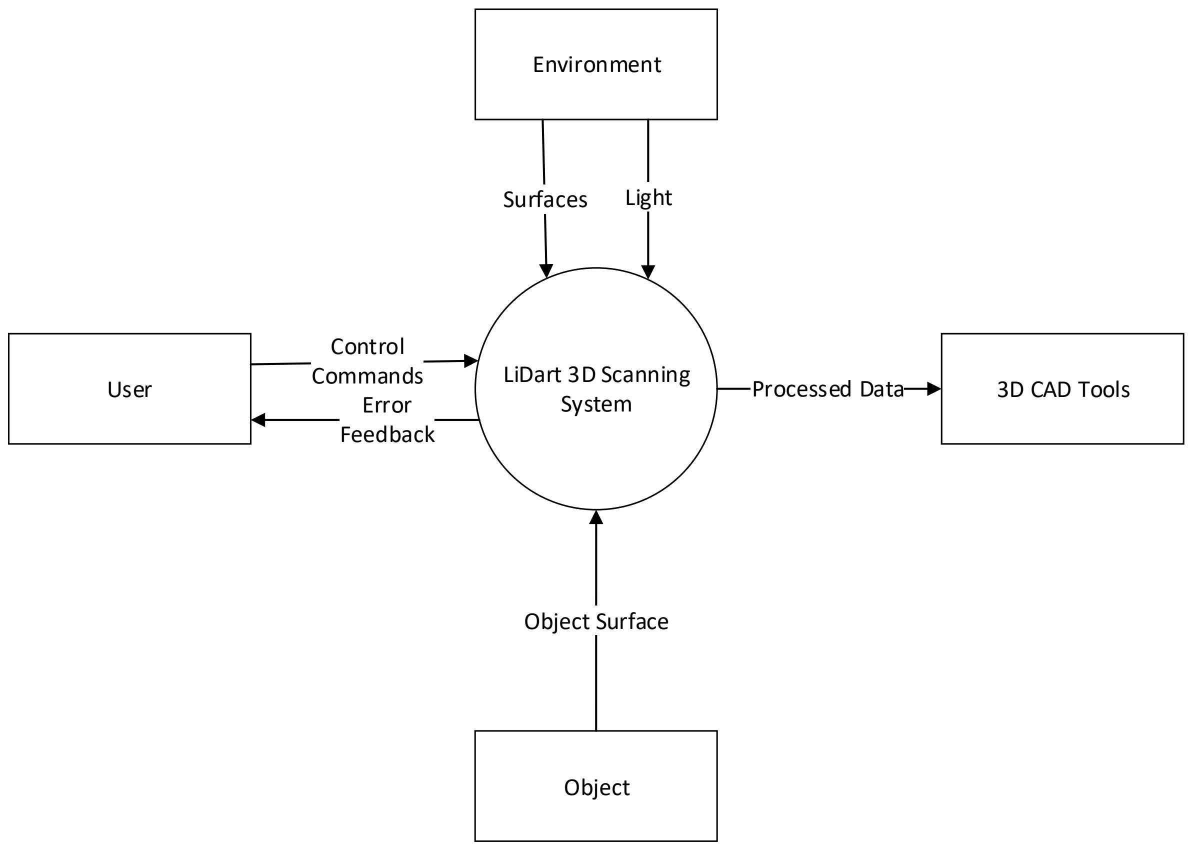

Figure 1: System Context Diagram

6 Project Overview

6.1 Normal Behaviour

LiDart is a remote-controlled robot that is operated by a user through a web-based user interface. In order to obtain scans, location markers must be set up in the environment surrounding the robot. The user will drive the robot to the desired scanning location by specifying the direction of movement on the user interface. Live camera feed will be displayed on the user interface to aid with navigation. Once the robot is at the desired location, the user can direct the robot to begin scanning through the user interface. The robot will continuously perform state estimation while taking scans through the LiDAR sensor. After scanning is complete, the point-cloud data will be stitched together to create the final 3D scan, which can then be saved as a .obj file.

Normal operation of the LiDart system will proceed as follows:

1. Robot is powered on
2. User connects to the robot via WiFi
3. User opens the GUI
4. User drives the robot to a desired location, the user can use the live video feed and LiDAR data preview to choose such a location
5. User initiates a scan
6. Steps 4 and 5 are repeated until the user has scanned all desired areas of the environment
7. User downloads the final stitched 3D scan

6.2 Undesired Event Handling

LiDart will detect error conditions due to undesired events and alert the user of any errors. Messages shall be displayed on the user interface to alert the user of errors and suggest steps to correct these errors. In a situation where the error cannot be resolved, the robot will need to be manually reset. Several events that fall outside of normal operations are described in the following sections.

6.2.1 Cancelling a Scan

If the user has initiated a scan but then decides that the location is not ideal, the user can cancel the scan. This discards the information from that specific scan and allows the user to resume driving the robot.

6.2.2 Robot Not Localized

If the robot is not localized, it will not allow the user to initiate a scan. To be able to initiate a scan the user must first drive the robot to a location where the robot can localize itself.

6.2.3 Emergency Stopping

If the robot begins misbehaving it can be powered off with an onboard emergency stop switch. This is required in the event that a hardware failure causes the robot to pose a safety risk or to stop responding to user commands.

6.3 Component Diagram

Figure 2 depicts the decomposition of LiDart into sub-systems. LiDartconsists of a user interface, an on-board computing module, robot inputs (sensors), and robot outputs (actuators). The embedded software running on the robot is further subdivided into modules which are described in detail in the Module Interface Specification and Module Guide documents.

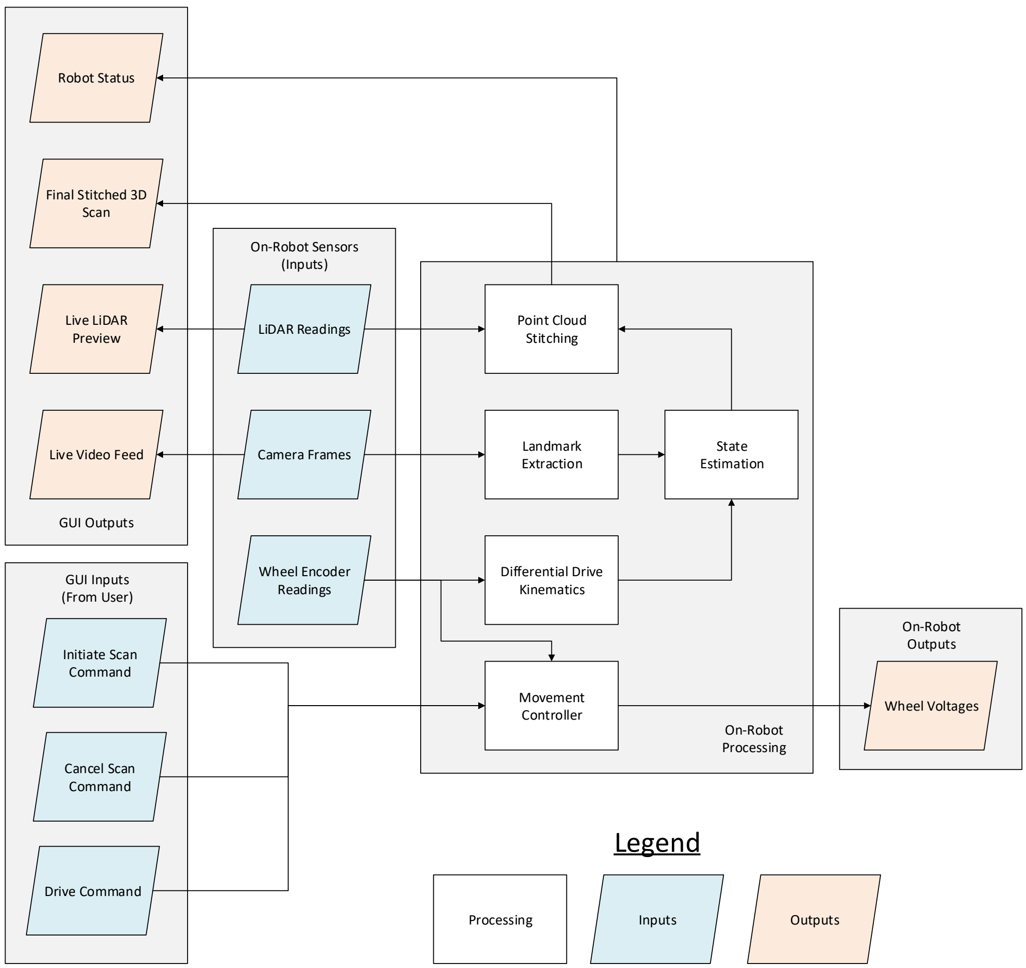

Figure 2: Functional Decomposition Diagram

6.4 Connection Between Requirements and Design

Table 1 shows the traceability between design components and requirements specified in the SRS.

Table 1: Traceability between system components and requirements

Component	**Requirement**
User Interface Inputs	
User Interface Outputs	
On-Board Robot Inputs	
On-Board Robot Actuators	

7 System Variables

This section defines the monitored, controlled, and constant system variables.

7.1 Monitored Variables

Table 2 specifies the variables that are measured by the system.

Table 2: Monitored Variables

Name	**Type**	**Range**	**Units**	**Description**
m_InitScan	boolean	[0,1]	-	Initiate scan command
m_CancelScan	boolean	[0,1]	-	Cancel scan command
m_Forward	boolean	[0,1]	-	Move robot forwards command
m_Backward	boolean	[0,1]	-	Move robot backwards command
m_Right	boolean	[0,1]	-	Move robot to the right command
m_Left	boolean	[0,1]	-	Move robot to the left command
m_ExportScan	boolean	[0,1]	-	Export scan command
m_Battery	float			Battery level reading
m_Distance	float	[200-12000]	mm	Distance between the rotating core of the LiDAR sensor and the sampling point
m_Heading	float	[0-360]	deg	Angle of the LiDAR sensor measurement
m_StartFlag	boolean	[0,1]	n/a	Flag to start a new LiDAR scan
m_Checksum	int		n/a	Checksum of the LiDAR data

7.2 Controlled Variables

Table 3 specifies the variables that are controlled by the system.

Table 3: Controlled Variables

Name	Type	Range	Units	Description
o_WheelRight	int	[0-5]	V	Right wheel output voltage
o_WheelLeft	int	[0-5]	V	Left wheel output voltage
o_Staus	int	[0-4]	n/a	Robot state (fault, idle, waiting for connection, ready to scan, scanning)
o_ErrorState	string	n/a	n/a	State estimation error message
o_ErrorObstacle	string	n/a	n/a	Obstacle error message
o_ErrorWifi	string	n/a	n/a	WiFi connection error message
o_ErrorBattery	string	n/a	n/a	Low battery level error message
o_ScanOut	.obj file	n/a	n/a	file containing 3D scan
o_Camera	binary	n/a	n/a	Live video feed

7.3 Constants Variables

Table 4 specifies the system constants.

Table 4: Constants

Name	Type	Range	Units	Description

- state estimation?, movement controller physical quantities i.e. wheel voltages, physical dimensions/properties of the robot, communication protocol constants i.e. baude rate

8 User Interfaces

The user interface is accessible using a smart device with WiFi capability. The user interface is used to control the movement of the robot, instruct the robot to start scanning, and save the data on the user's device. The main area of the user interface is used to display live video feed from web-cams attached to the robot. A horizontal bar across the top of the interface provides buttons to control the robot, start/stop scanning operations, and save the scanning data. The keyboard arrow keys may also be used to move the robot. Sketches of the user interface are provided in Appendix A.

9 Design of Hardware

- aluminum frame chassis
- omni-directional wheels (holonomic robot design)
- mounts for cameras, LiDAR sensor
- enclosure for electronics

CAD renderings of the robot are provided in Appendix B.

10 Design of Electrical Components

LiDart uses a micro-processor based embedded system to run the GUI and control the robot. A Raspberry Pi hosts a web-server for the GUI, interfaces with peripheral devices, and processes data obtained from the LiDAR sensor. The Raspberry Pi communicates with peripheral devices over USB), including cameras, the LiDAR sensor, and an Arduino micro-controller. The Arduino is used to drive a dual H-bridge motor driver which controls two DC motors. A circuit diagram and bill of materials is provided in Appendix C.

11 Design of Communication Protocols

USB (Universal Serial Bus) is used to communicate between the Raspberry Pi controller and peripheral devices. WiFi is used to communicate between the user's smart device, such as a cellphone or laptop, and the robot.

12 Timeline

Table 5 defines the design tasks and team member responsibilities.

Table 5: Design Timeline

Task	**Team Member**	**Date**
Rev 0 mechanical design, procurement/fabrication of components, and mechanical assembly	Neeraj Ahluwalia	09/Jan/2023 - 23/Jan/2022
Rev 0 electrical design, procurement of components, and electrical assembly Modules: Movement controller	Michaela Schnull	09/Jan/2023 - 23/Jan/2022
Rev 0 development of the frontend/user interface Modules: Frontend	Karim Elmokattaf	09/Jan/2023 - 23/Jan/2022
Rev 0 development of the state estimation/point-cloud stitching software Modules: State estimation, point-cloud stitching, landmark extraction, differential drive kinematics	Jonathan Casella	09/Jan/2023 - 23/Jan/2022
Rev 0 integration testing	All	24/Jan/2023 - 06/Feb/2022

References

[1] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Module interface specification for LiDart," 2022.

[2] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Module guide for LiDart."

[3] N. Ahluwalia, J. Casella, K. Elmokattaf, and M. Schnull, "Software requirements specification for LiDart."

[4] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, May 2011.

A Interface

B Mechanical Hardware

Insert CAD assembly diagram and BOM here.

C Electrical Components

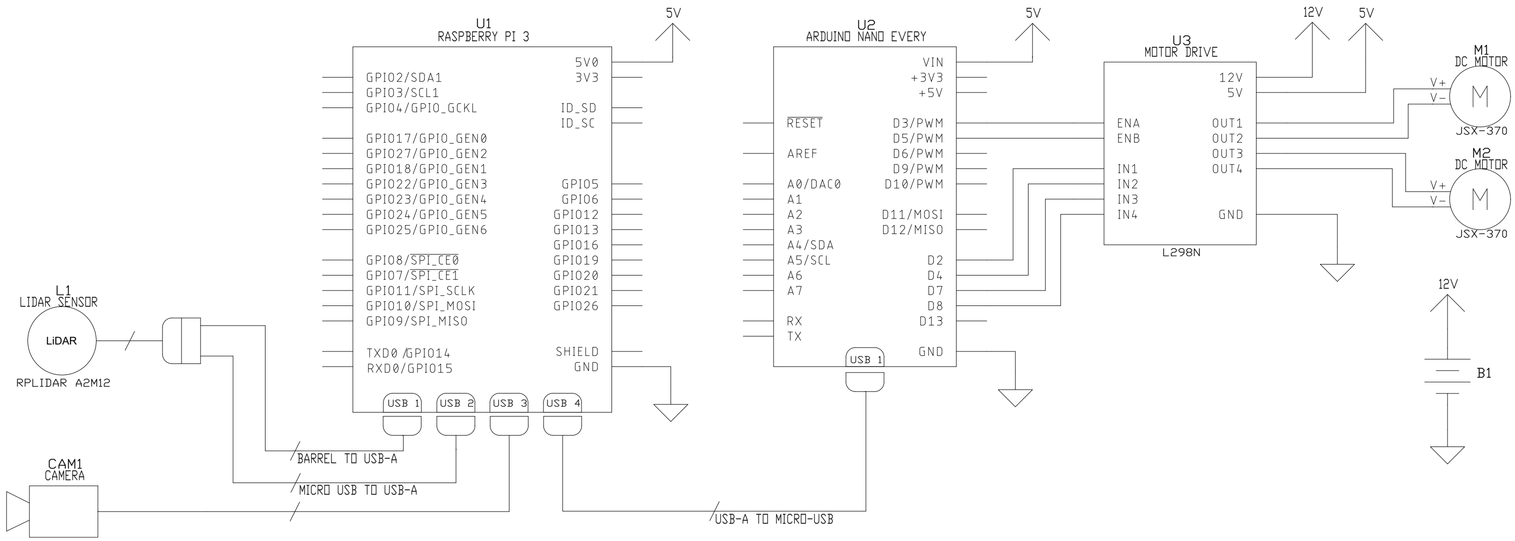

Figure 3: Circuit Diagram

Table 6: Bill of Materials - Electrical

Part Number	**Description**	**Quantity**	**Reference Designator**
Slamtec RPLIDAR A2M12	360-degree 2D laser scanner (LiDAR)	1	L1
Raspberry Pi Rev 3	Single board computer	1	U1
Arduino Nano Every	ATMega4809 AVR microcontroller board	1	U2
L298N	Dual H-Bridge Driver IC Board	1	U3
JSX-370	DC 12V Gear Reduction Motor	2	M1, M2
TBD	Mini-Webcam USB	1	CAM1
12V DC Battery	12V DC Battery	1	B1
TBD	Switch	1	SW1

D Communication Protocols

E Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

E.1 Cost Limitations

Cost constraints imposed limitations on the LiDart solution. An inexpensive LiDAR sensor with reduced accuracy and range compared to more expensive solutions was used due to the limited budget. The more expensive LiDAR sensor would improve the accuracy of data obtained and enable larger areas to be scanned. Limited funding also limited the mechanical design of the robot. The robot used primarily 3D printed components. Custom machined components are more expensive, however wold improve the durability, reliability, and tolerances of the robot. A custom PCB (printed circuit board) could also be designed instead of various connected boards This would reduce number of components and improve the reliability and maintainability of the system.

E.2 Localization Accuracy

Additional sensors could be added to improve the localization of the robot, such as wheel encoders and accelerometers. Sensor fusion could be implemented to improve the state estimation algorithms, resulting in and increase in the accuracy of scans.

E.3 Autonomous Scanning

The operation of the robot in the specified design requires the user to remotely pilot the robot. Since this could be time consuming for the user, an additional feature that LiDart could implement is autonomous scanning. While autonomously scanning, the robot would explore its environment without any user intervention. Implementing autonomous scanning would require the development of additional software.

2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)

E.4 User Interface

A desktop graphical user interface and web-based user interface were considered. The web based user interface was selected due to its flexibility, portability, and ease-of-use. A user can simply connect to the web-hosted interface instead of having to install a desktop application. A disadvantage of the web-hosted user interface is limited functionality compared to a desktop application.

E.5 Holonomic Robot Design

Holonomic and non-holonomic robot designs were considered. A holonomic robot design was chosen due to greater maneuverability and ability to navigate through small and narrow spaced. A drawback of using holonomic robots was that omni-directional wheels must be used which are not standard and have less traction.

E.6 Localization Markers

Round spherical markers and April Tags [4] were considered as markers used for the robot state estimation. April Tags provided many advantages compared to the round coloured markers. Unlike generic round markers, April Tags do not need to be calibrated, which is a time consuming and error-prone process. They are also easier to detect, and provide additional information such as the angle, resulting in more accurate state estimation.