6.205 Project Block Diagram Report

Juan Antonio Luera — Jluera
Manuel Valencia — manuelv
Juan Angel Luera — j_luera

Oct 25, 2023

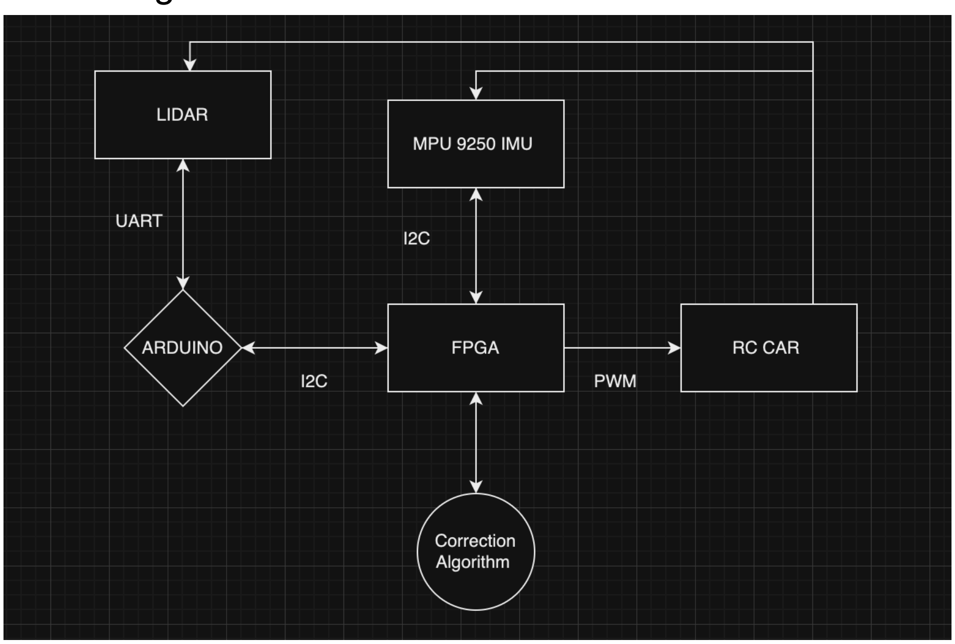
Abstract

The integration of LIDAR sensors and Inertial Measurement Units (IMUs) in robotic navigation can provide enhanced data sensing processing and energy efficiency. Our project aims to explore the robustness and responsiveness of FPGA in handling the demands of these sensors. Leveraging the parallel processing capabilities of FPGAs, our system is designed to simultaneously interpret continuous LIDAR data and calculate the car's real-time orientation and position using IMUs. This approach ensures an efficient and real-time reaction to surrounding obstacles.

The minimally viable project consists of implementing the I2C protocols to interface with an IMU and a LIDAR camera connected to an arduino and using this data navigate an RC car. The FPGA will process information from the IMU and LIDAR camera and update the trajectory of our RC car if an object is detected. An RC car has already been assembled, and the motors can be controlled through PWM signals produced by the FPGA. The goal is to recognize obstacles and maneuver around them quickly. Our stretch goal is to implement a path solving algorithm like A* or SLAM algorithm for room mapping.

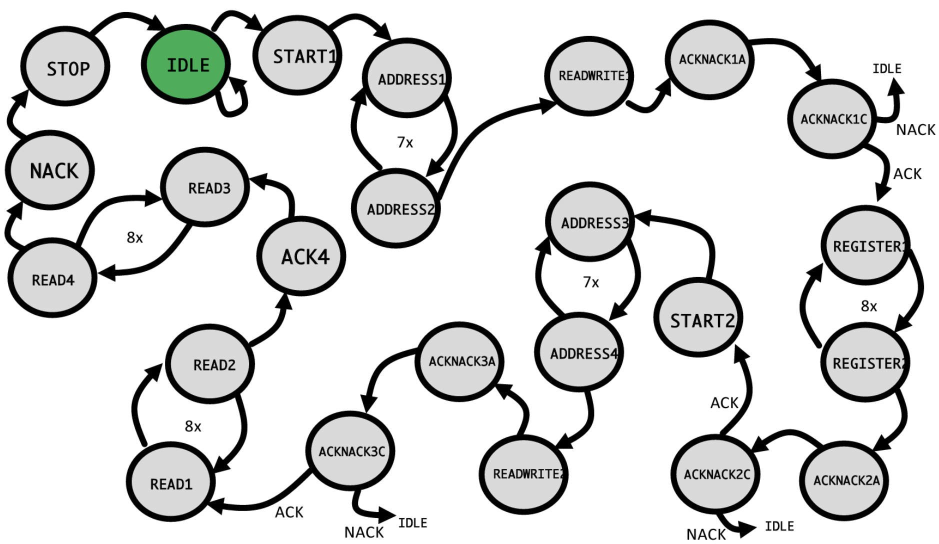
Our aim is to demonstrate how FPGA's timing and parallel processing can optimize real-time navigational tasks.

General Block Diagram



Interfacing

I2C



(copied from Lecture)
We plan to communicate with the MPU 9250 and the Arduino using the I2C protocol. We will implement this communication protocall in the LAB. The FPGA will be the Main and the Arduino and MPU 9250 will be the secondaries. We will make sure both are addressable separately, and we will collect data from each as fast as possible. This will be done using two modules from which we can retrieve data from both in parallel.

LiDAR data Processing on Arduino

We will be using this LiDAR, which uses UART communication. We plan to control the speed of the LiDAR as well. We will configure the Arduino with several commands to change the speed of the lidar rotation and to retrieve data using the I2C protocol. We will transmit the data through a processed packet of bytes where each byte will be one depth measurement, and we will send N amount depending on the configuration of the LiDAR.

PWM & Motor Control

We have LM293 H-Drives on our RC car that we can directly wire to pins on the FPGA to drive the motors using PWM signals. If possible we will make use of the servo pins on the FPGA board. Will use the same PWM from class

Algorithms

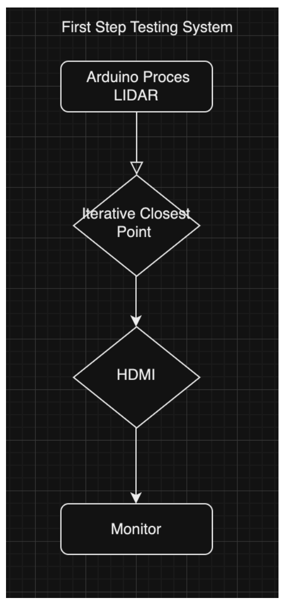
This will be the bulk of our project, and we will need to implement several algorithms to make this successful. In this section, we go over the several modules we will need to implement.

Iterative Closest Point Algorithm (ICP)

We will need two BRAMS for this algorithm.

The LiDAR sends point cloud data, which, after it is processed, will be received by the FPGA and stored in a BRAM module. We will have a module that will sample this data using a Iterative Closest Point Algorithm which we will implement, will make a 2-D map of the surroundings. Obstacles will be shown as edges and we will store this in a different BRAM for ease of access by the visualizer and our control algorithm for the RC car.

Point Cloud Data Visualizer



The first step in our project will be to create a simple visualizer for the point cloud data sent by the Lidar. We plan to display our visualizer as a 2D sketch where lines indicate obstacles. We will want to update this every so often, but we will need to consider this more in-depth, depending on the timing constraints of interfacing with the arduino. Ideally, we will want to display this data at a rate of 24 frames per second. To visualize the data, we will re-use the HDMI modules developed in LAB. We will also make a mapping algorithm that will map the data from the BRAM into pixel values to display on the screen.

Maze-Solver and Possible Stretch Goals

For our project MVP we plan to use the data we processed and drive the car in straight lines and sharp ninety degree turns. The data from the lidar will determine if we are able to move straight or need to turn. If an obstacle is found a sharp right turn will occur and the RC car will move straight until no obstacle on the left side is detected before turning back to it's original heading and moving straight. This module might also need to implement some position control for the RC car using the IMU and LiDAR data. If time allows, possible stretch goals include implementing an A* or RRT path solving algorithm or improved path control system for traversing more complex environments.

IMU for Error Correction

The IMU will be utilized to correct the heading of the RC car. The MPU 9250 has a three axis gyroscope that will be utilized for this task. We hope to also be able to and maintain some position control after using the IMU and LiDAR data in conjunction. We will communicate with the IMU through I2C communication and our control system will update the PWM signals for the RC car.

Deliverables

Data	Antonio	Angel	Manny
Week 1	LiDar Arduino implemented with I2C communication with FPGA	PWM and Control System for RC car from FPGA. Start implementing IMU.	I2C on FPGA, HDMI module, IMU
Week 2	Iterative Closest Point Algorithm	Straight Lines and Turns using IMU	Visualize lidar data. (No ICP and with ICP)
Week 3	Pipelining/system integration	Implement Maze Solver Algorithm	Visualizer for maze solving (points updating correctly with moving car)
Week 4	Final Integration	Final Integration	Final Integration / A*
Week 5	Report Write Up	Report Write Up	Report Write Up