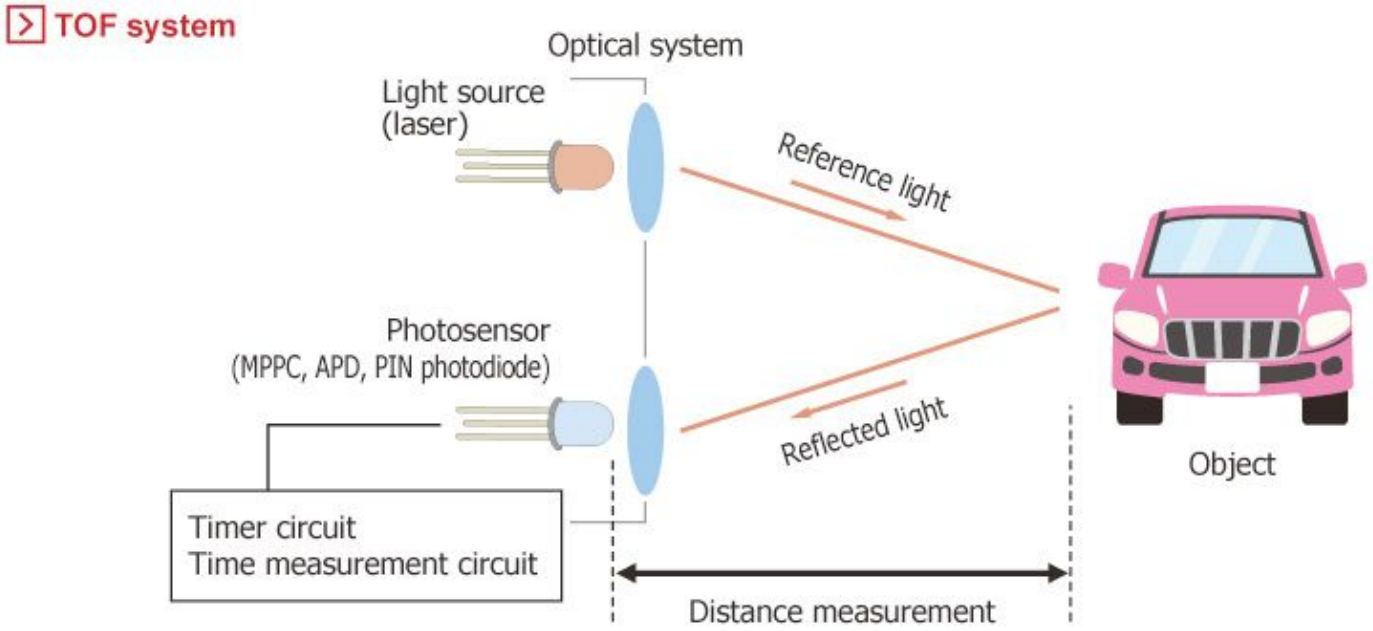
Open Source LIDAR

Light Seekers

Paul Roy, Gaurav Bhalla, Allen Chen, Aamhish Rao

LIDAR Background

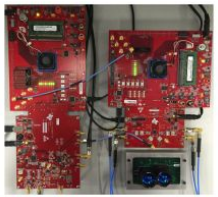


Evaluation of Alternative Solutions

Xaxxon’s OpenLIDAR

OSLRF-01

TIDA-01187

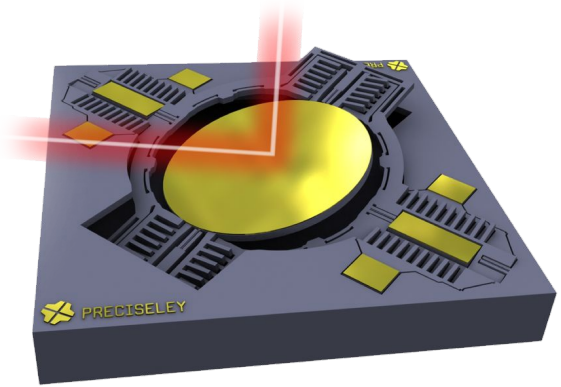


Open Source LIDAR Unruly

UNRULY

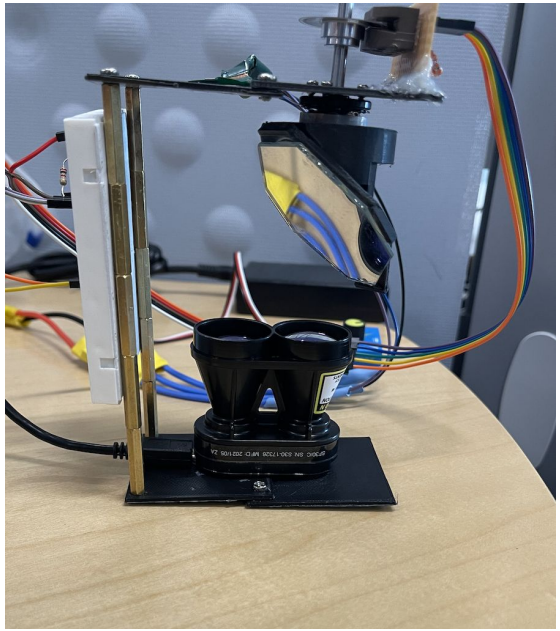
Problem Background

- Research group’s MEMS mirror is difficult to use.
- Commercial product do not have non-uniform sampling rate
- MEMS mirror create low and high speed areas of sampling
- We needed to copy this capability.



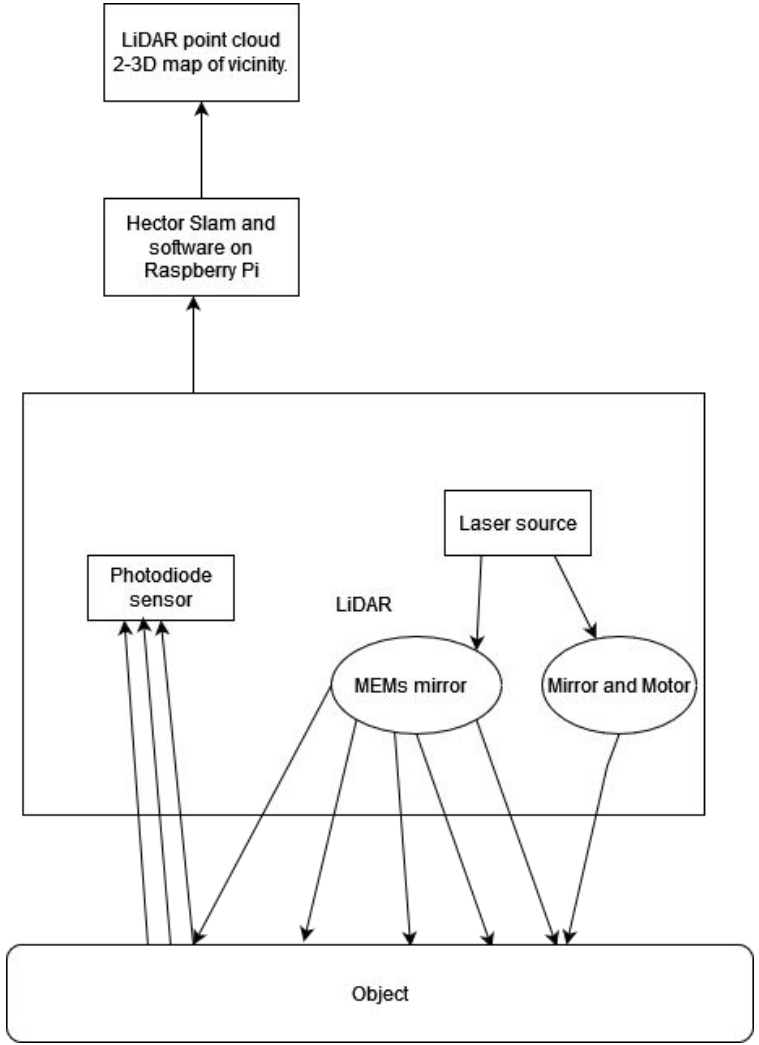
Project Goals

- Recreate a working LIDAR system. ✓
- Needed to sample space in a non-uniform manner.
- Needed to create re-configurable design.

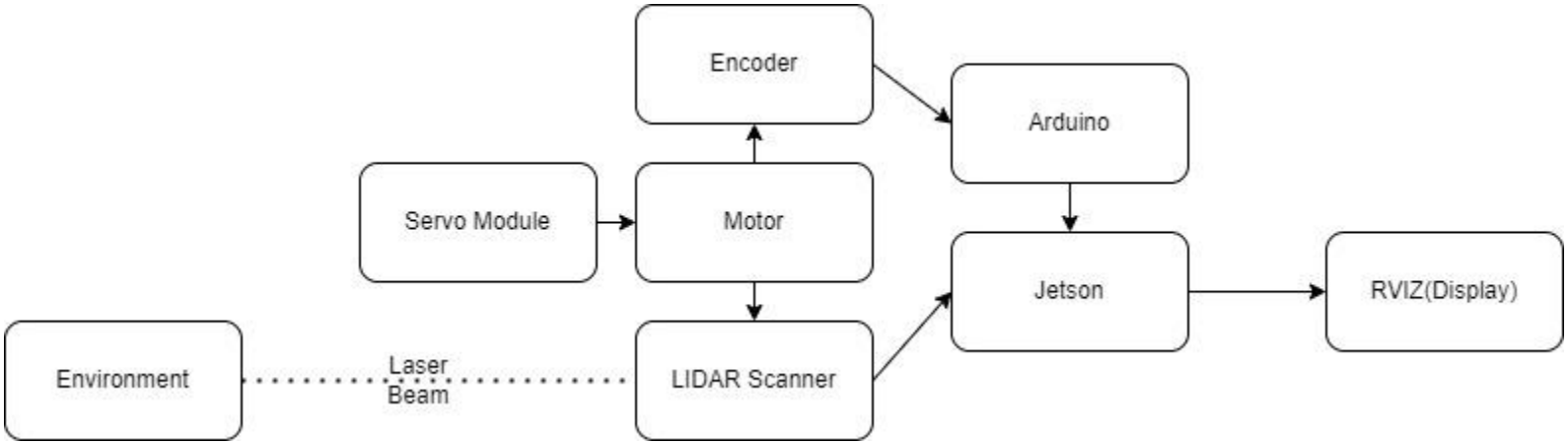


System Design

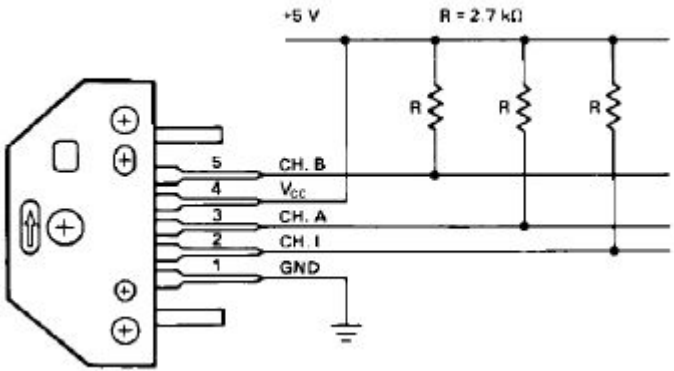
- Lidar
  - Laser source
  - Spinning mirror
  - Photodiode sensor
  - Electronic and optical components mounted on custom PCB board
- Post Processing
  - Hector SLAM/ROS to read and process lidar data
  - Generate 2d/3d point cloud

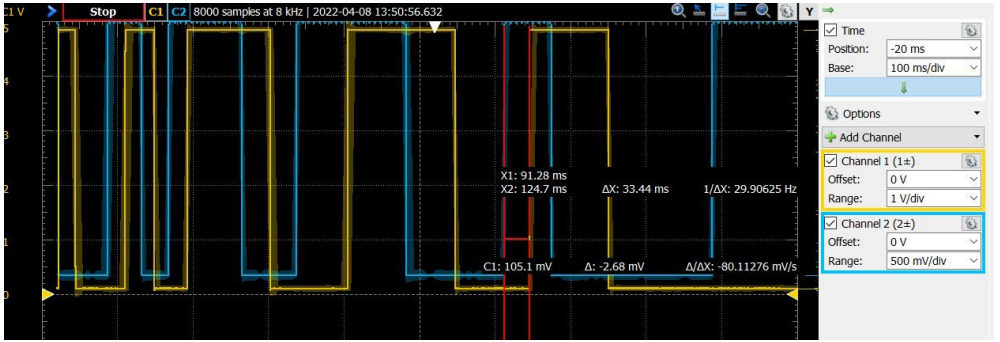


SF30C LIDAR Low Level Design



Encoder


+5 V
R = 2.7 kΩ
R
R
R
5
CH. B
4
Vcc
3
CH. A
2
CH. I
1
GND


Stop
C1
C2
8000 samples at 8 kHz | 2022-04-08 13:50:56.632
X1: 91.28 ms
X2: 124.7 ms
ΔX: 33.44 ms
1/ΔX: 29.90625 Hz
C1: 105.1 mV
Δ: -2.68 mV
Δ/ΔX: -80.11276 mV/s
Time
Position: -20 ms
Base: 100 ms/div
Options
Add Channel
Channel 1 (1±)
Offset: 0 V
Range: 1 V/div
Channel 2 (2±)
Offset: 0 V
Range: 500 mV/div

Motor

- 12V LiPo Battery and ESC.
- Used Servo module to interface with ESC and control motor with PWM.
- PID Loop with Encoder input and Motor output.

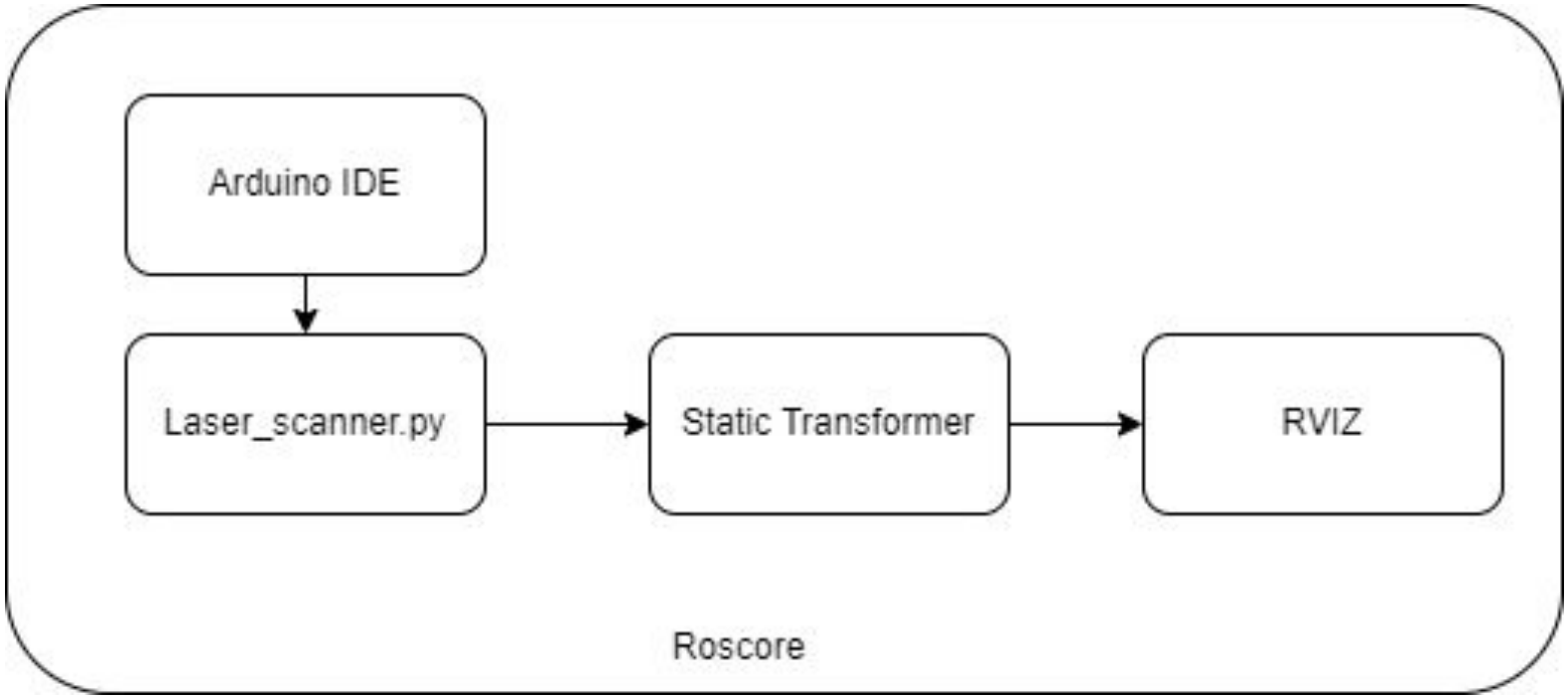
Arduino

```
  attachInterrupt(0, CounterA, RISING); // pin 2 on Arduino Mega
  attachInterrupt(1, CounterA2, FALLING); // pin 3 on Arduino Mega
  Serial.begin(9600);
  Serial.print("Setting up encoder program to read speed");
}

// Counter A interrupt service routine for rising edge
void CounterA() {
  if(!rising){
    rising_time = micros();
    rising = true;
  }
}
// Counter A interrupt service routine for falling edge
void CounterA2(){
   if(rising && !falling){
     falling_time = micros();
     falling = true;
   }
```

- attachinterrupt() for rising and falling edge.
- Measure time for 500 cycles.
- Every 500 cycles, output RPS and reset timer.

```
cycle_cnt += 1;
if (cycle_cnt == 500){
   Serial.println("Speed (RPS)");
   Serial.println(1000000.0 /(curr_time));
   curr_time = 0;
   curr_angle = 0;
   cycle_cnt = 0;
}
```

Python ROS Programs

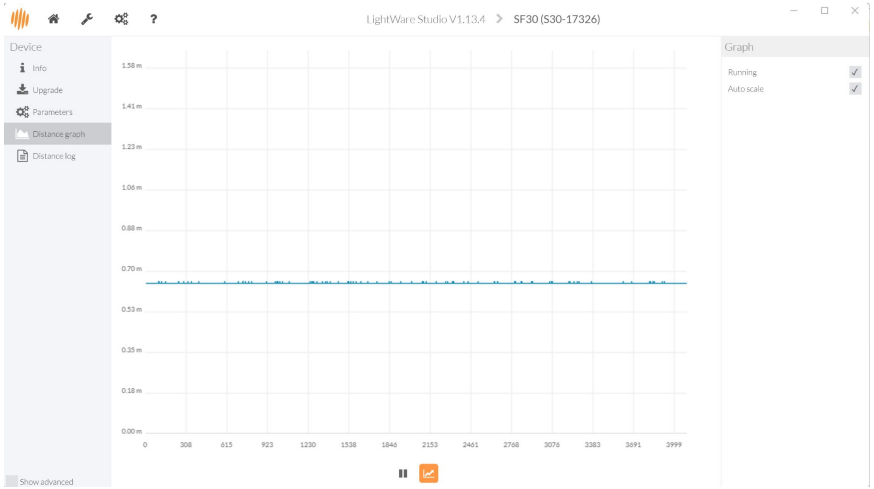


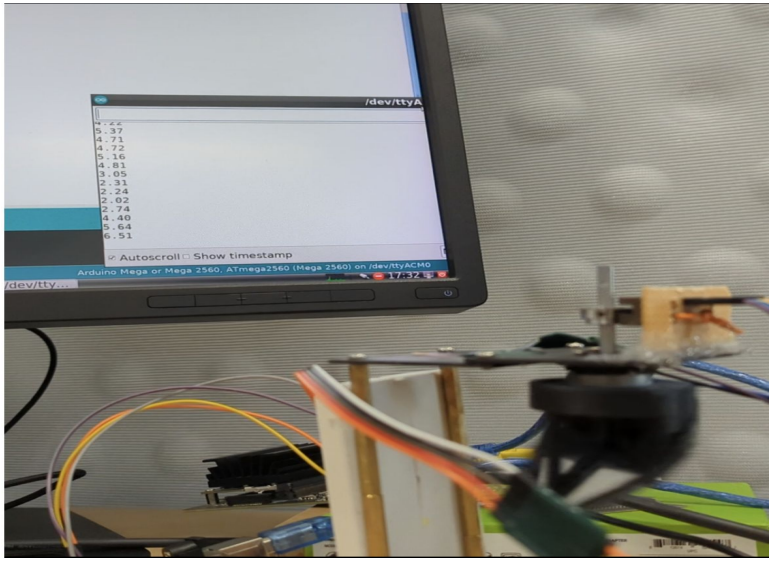
LaserScanner

```
laser_frequency =  625 # 20000 (5000 RPM motor = 0.012 seconds / revolution)

count = 0
r = rospy.Rate(1.0)

port = init_port()

while not rospy.is_shutdown():
    speed = arduino.readline()
    if speed.decode('utf-8'):
        print("speed.decode('utf-8') ", speed.decode('utf-8'))
        num_readings =int(laser_frequency/float(speed.decode('utf-8')))

    current_time = rospy.Time.now()

    scan = LaserScan()

    scan.header.stamp = current_time
    scan.header.frame_id = 'laser_frame'
    scan.angle_min = -3.14 #-1.57
    scan.angle_max = 3.14 #1.57
    scan.angle_increment = 2*3.14 / num_readings
    scan.time_increment = (1.0 / laser_frequency) / (num_readings)
    scan.range_min = 0.0
    scan.range_max = 100.0

    scan.ranges = []
    scan.intensities = []
    for i in range(0, num_readings):
        dist = get_dist(port)
        scan.ranges.append(1.0 * dist)  # fake data
        scan.intensities.append(1)  # fake data
    scan_pub.publish(scan)
    count += 1
```

- Initialize serial port
- Initialize ROS node
- Receive code from Arduino and decode data
- Create laser scan message, setup params
- In readings loop append the ranges in sets
- Fill with fake data for intensities.

Test/Evaluation Results



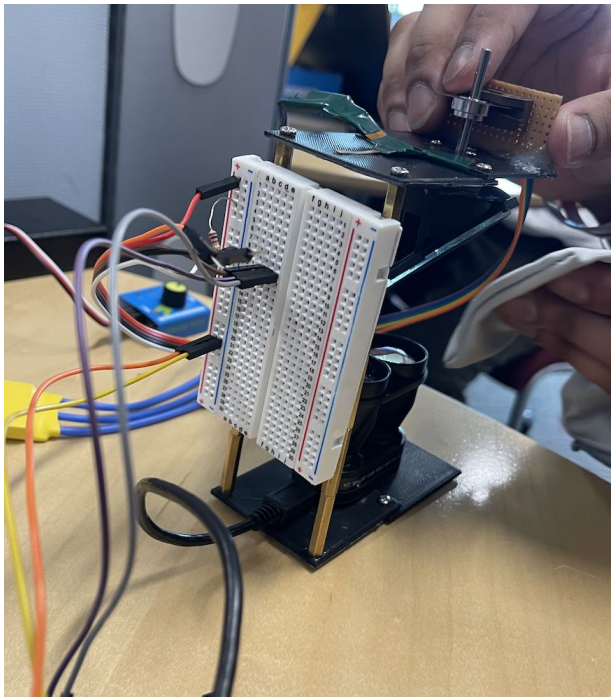


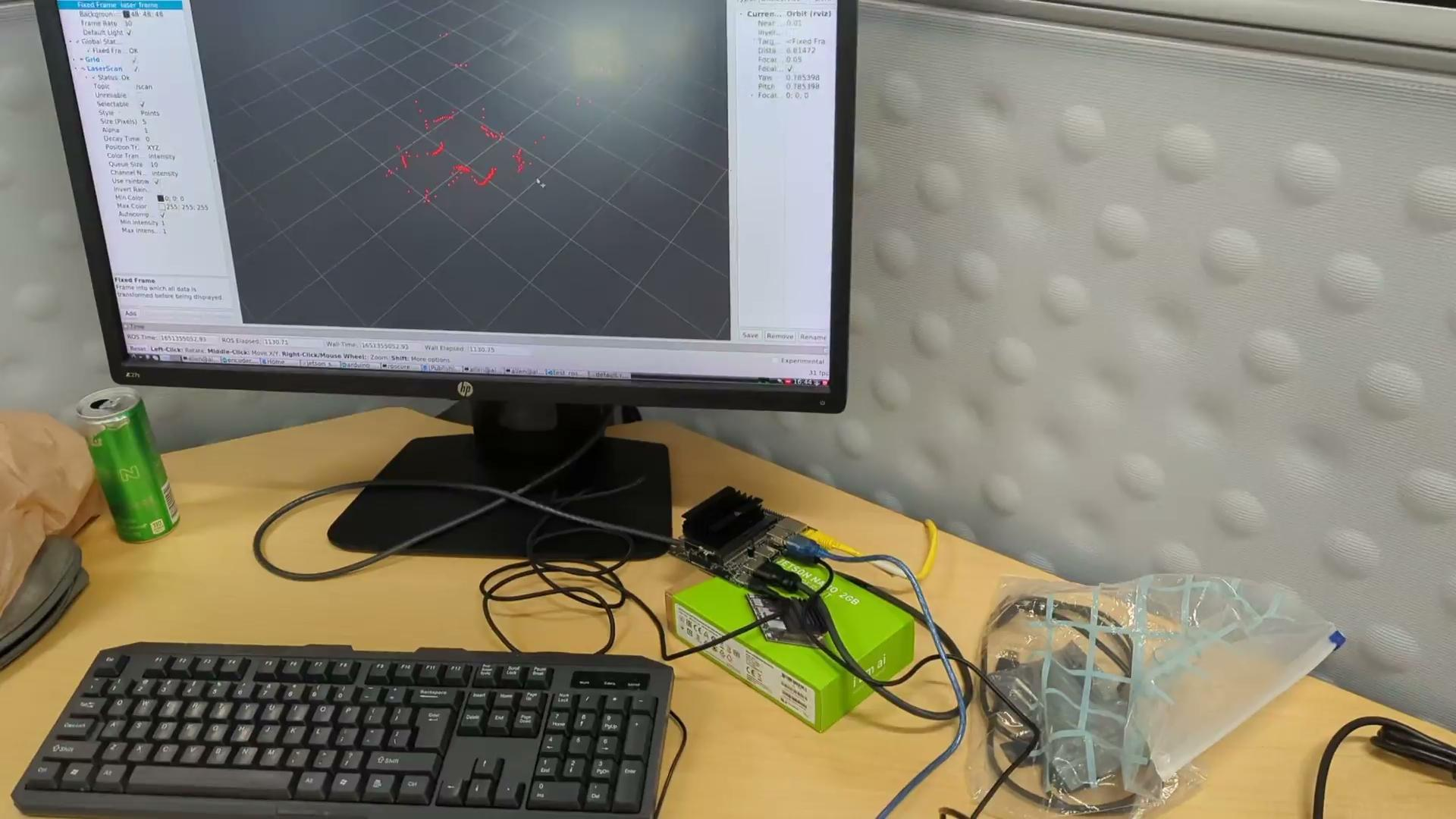
1)Validated physical measurements against SF30C.

2) Validated output from serial monitor after changing motor speed.

Demonstration plan

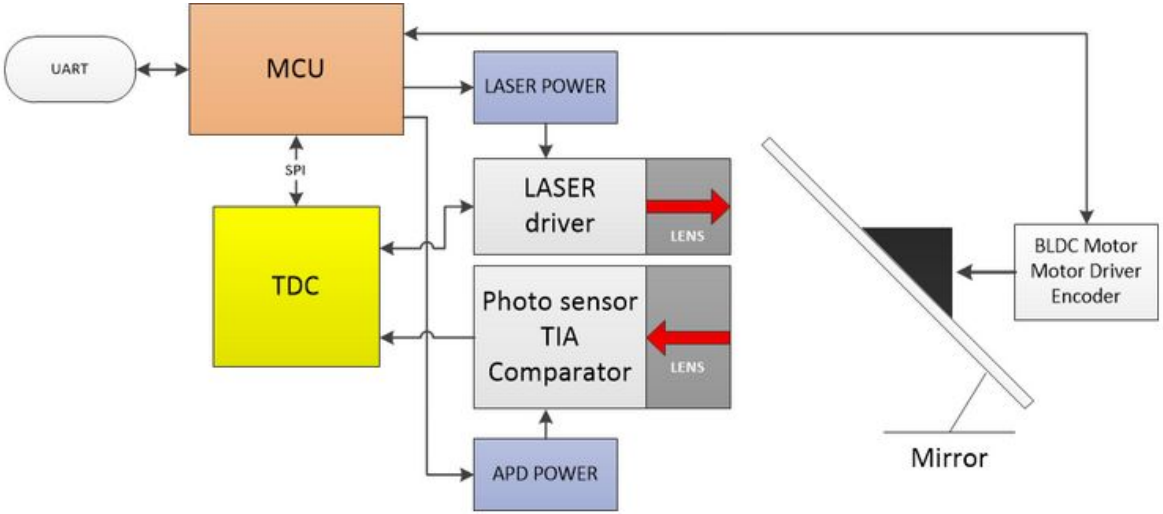
- Connect all of the components of the SF30 Lidar system.
- Power on battery
- Run python and arduino programs.
- Run RVIZ to see point cloud map of the local area.



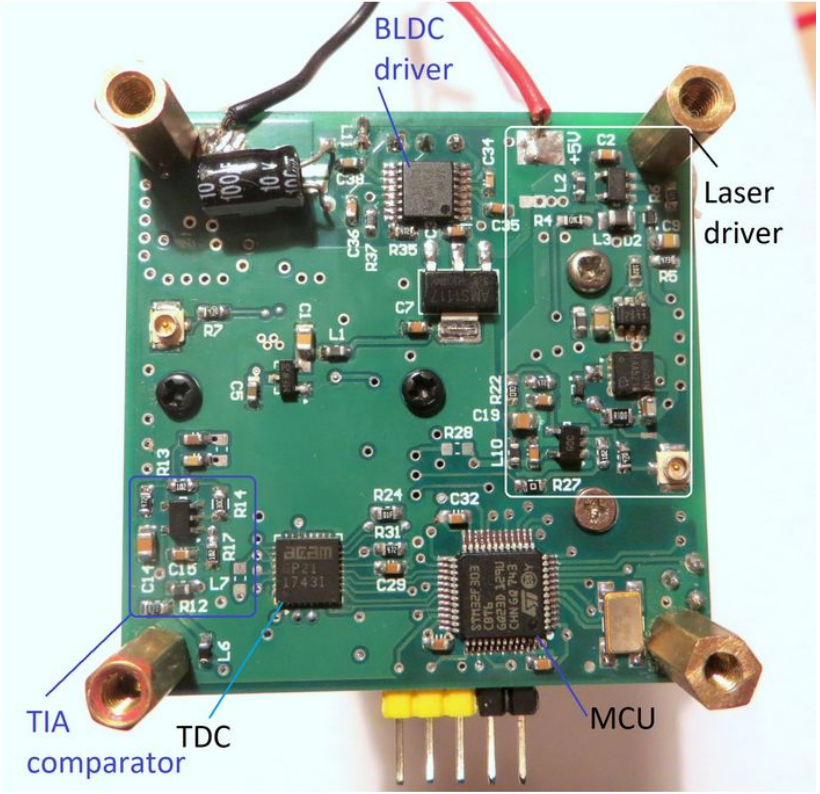


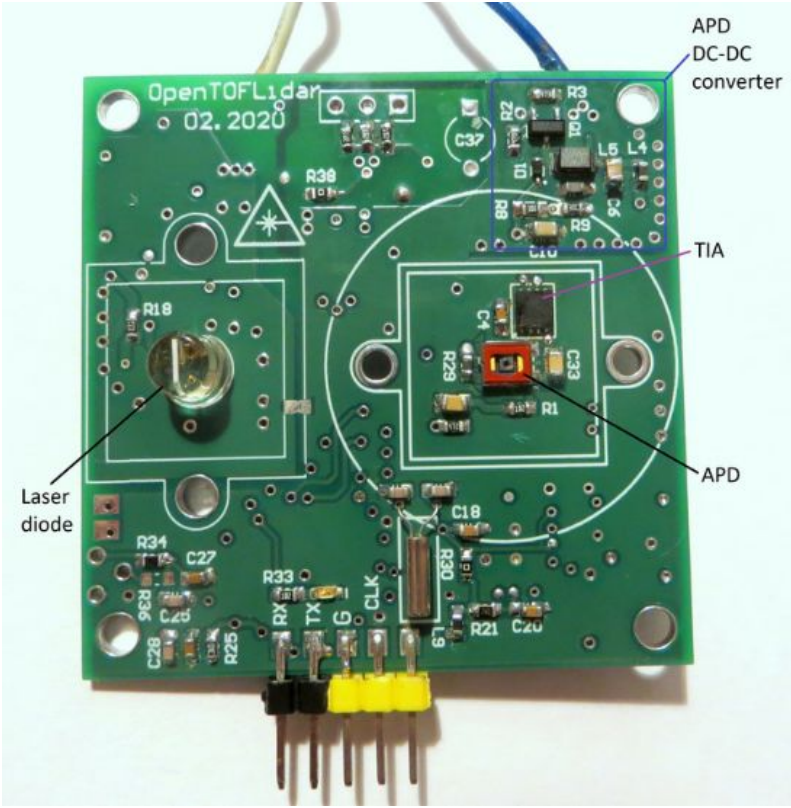
PCB Design Low Level Block Diagram

Author flashes firmware on the MCU.



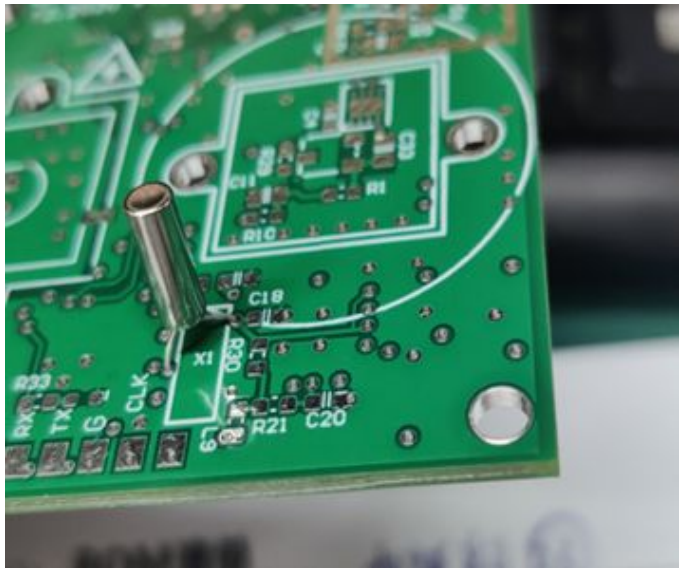
PCB Design Flow





Safety and Environmental Analysis

- Potential harm of LiDAR laser
  - Laser beam harmful to eye
- Environmental issues with PCB manufacturing
  - Contains Lead

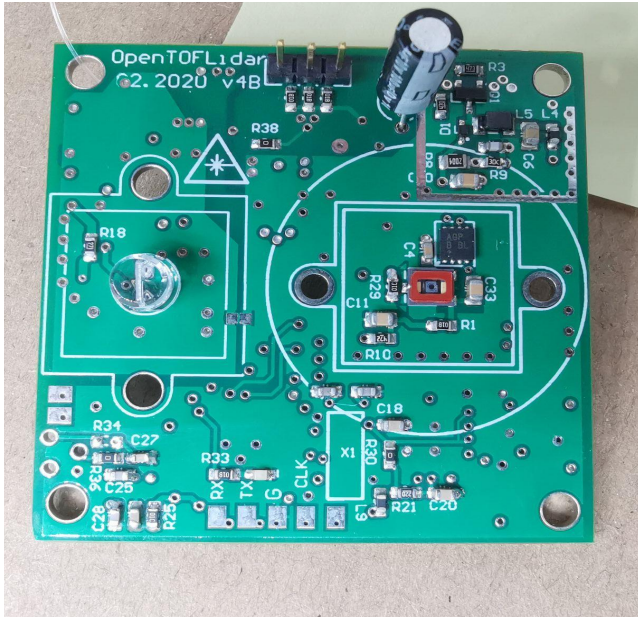


Social, Political, Ethical Concerns

- Used for self-driving cars too early.
- Ethical, social and political issues in Eastern Europe
- COVID outbreaks in China
- Lidar parts including the OpenTOF PCB heavily delayed

Manufacturability, Sustainability, and Economics

- PCB had a MOQ of 5.
- ~50 days to ship all PCB components.
- PCB uses some lead.
- SF30C PCB likely has lead.
- Total cost = $967.86 ($367.86 over budget)



Project Management

- Two sub-teams: hardware and software
- Weekly meetings and occasional in-person meetings.
- Key meeting notes and weekly meeting reports
- Github

Q&A