AAE 490: TracSat Design Project

Fall 2020 Presentation

PURDUE UNIVERSITY

Introduction

TracSat Design Project

Mission Statement: To improve Purdue students engineering and collaboration skills through partnering with General Atomics, developing AAE's CubeSat testbed capabilities, and prototyping state-of-the-art CubeSat systems including Proximity Operations, Laser Communications, and Collision Avoidance.

General Atomics Sponsored Project

TracSat Team

Thomas Fu
HPL

Maya Havens
HPL

Kolin Palmer
LTS

Kaitlyn Ingalls
LTS

Pol Francesch
OSS

Derrek Gerrard
OSS

Cory Cooper
LLS

Sebastian Bell
Coding

Adam Frank
Coding

Kyler Kappes
Coding

Anderson Xu
Coding

PURDUE UNIVERSITY

Agenda

Project Overview

- Project Introduction
- Mission Objectives
- Major Milestones
- Concept of Operations

Subteam Capabilities

- Coding
- OSS: Object Surveillance Subsystem
- HPL: Hardware, Propulsion, Levitation
- LSS: Laser Link Subsystem
- LTS: Laser Tracking Subsystem

Project Overview

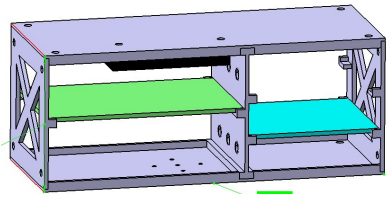
- Project Introduction
- Mission Objectives
- Major Milestones
- Concept of Operations

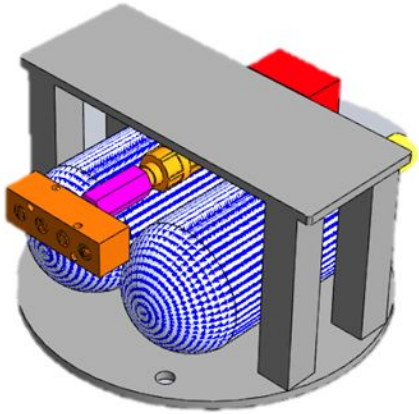
PURDUE UNIVERSITY

What Is TracSat?

PROJECT OVERVIEW

- “TracSat” is a portmanteau of Tracking Satellite
- **Goal:** Develop a testbed for 2D CubeSat operations and experiment with state-of-the-art CubeSat technologies based on inputs from General Atomics.
- Fall 2020 - Spring 2021
  - Develop single levitating CubeSat and Ground Station capable of sending video over laser communications.

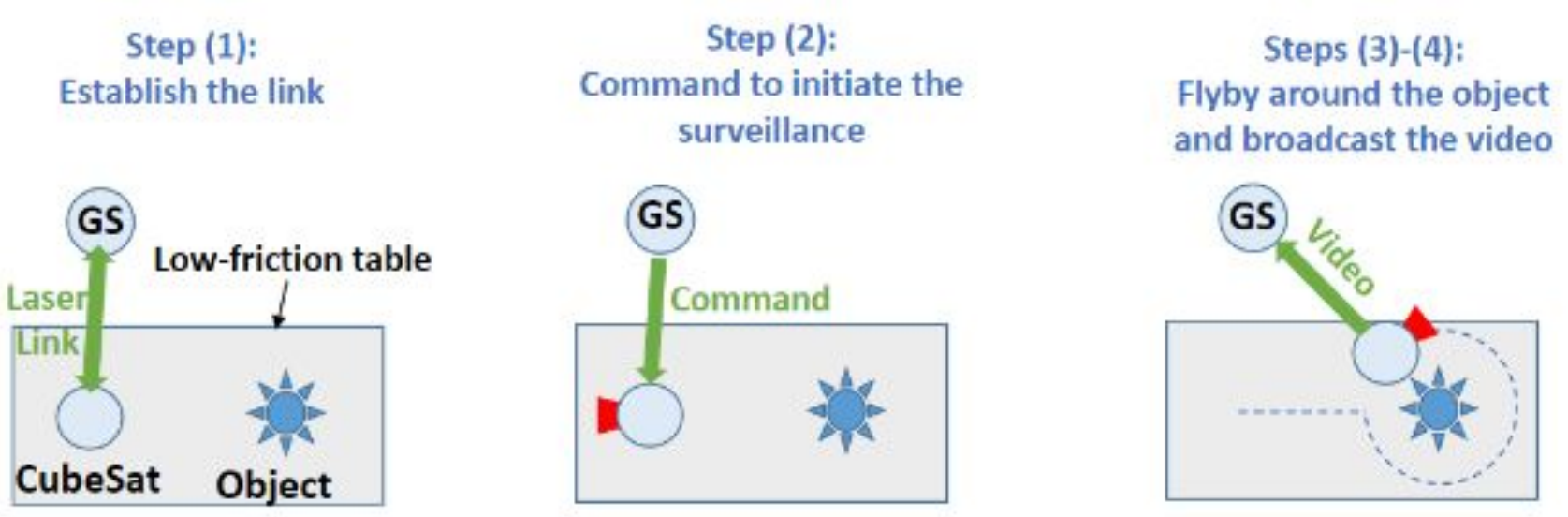




PURDUE UNIVERSITY

What is TracSat?

Project Overview



PURDUE UNIVERSITY

Mission Objectives

PROJECT OVERVIEW

Mission Objectives	
MO 1	Develop a “levitation base” for CubeSat testing and demonstration.
MO 2	Develop CubeSats capable of translational (2 dimension, xy-plane) and rotational (clockwise and counterclockwise) control.
MO 3	Demonstrate transmission of commands and data between a CubeSat(s) and Ground Station via tracking laser communication.
MO 4	Demonstrate autonomous collision avoidance methods to ensure CubeSat does not collide with object

Major Milestones

PROJECT OVERVIEW

Aug '20
- Team formation
- Development of mission concept

Sep '20
- Finalize COVID safety regulations
- Objective and requirement generation

Oct '20
- Optimized TracSat code and LTS Circuitry
- Familiarize ourselves with PYNQ board
- Conducted tests with LiDAR and Camera
- Test LLS Capabilities
- Order/test solenoids

Nov '20
- TracSat & GS LTS Test
- Central control software written for PYNQ board
- Finalize integration for camera and LIDAR
- Transfer LLS to PYNQ
- Create solenoids PCB and rework CAD

Dec '20
- Prepare for final review
- Organize team for next semester

PURDUE UNIVERSITY

Concept of Operations

PROJECT OVERVIEW

- Developing a Concept of Operations including:
  - Objectives and Goals
  - States and Modes
  - Nominal and Off-Nominal Mission Scenarios
  - System Architecture
- Goal is to ensure team cohesion and fluid system integration

PURDUE UNIVERSITY

Subteam Capabilities

- Coding
- OSS: Object Surveillance Subsystem
- HPL: Hardware, Propulsion, Levitation
- LSS: Laser Link Subsystem
- LTS: Laser Tracking Subsystem

PURDUE UNIVERSITY

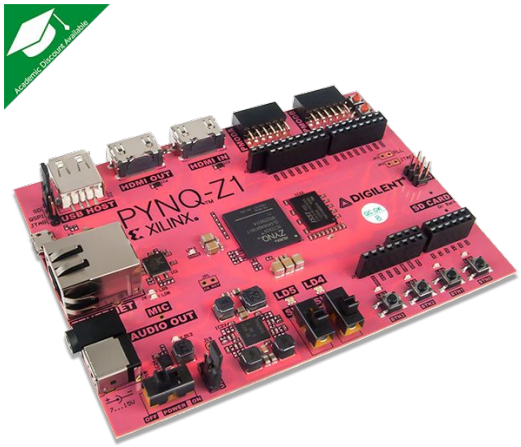
Coding Subteam

Capabilities In Development

Coding

PYNQ-Z1 Board

Used as the central hub for linking and driving each of the subsystems



Jupyter Notebook

Interactive environment used to write and run codes on PYNQ board.

jupyter

Vivado

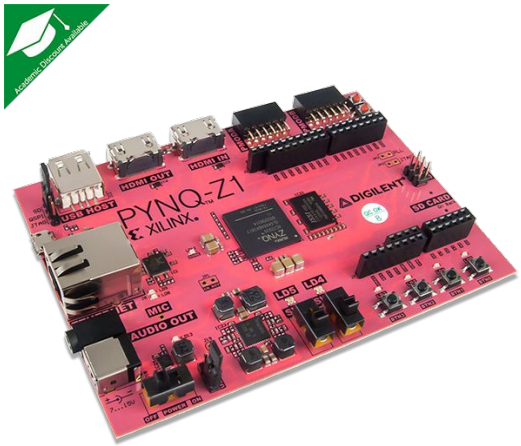
Coding environment used to code the hardware on the PYNQ board.

VIVADO™

PURDUE UNIVERSITY

Coding

- Successfully each set-up PYNQ boards
  - Flashed SD card images
- Each did a tutorial to get an LED flashing, some of the team used Vivado, some used Jupyter
- Got GPIO's working which allowed us to work on integration with other subteams
  - Web-cam and LIDAR so far



PYNQ - LED Demo

Button 0: Change LED color

Button 1: LED shifts from right to left

Button 2: LEDs shift from left to right

Button 3: End of the demo

Vivado Studio HLx

Coding

Pro's

- Is incredibly useful for coding the hardware on the board to be used without any connection to ethernet or a computer during use.

Con's

- The specific language the Vivado Design Suite uses with the PYNQ board is not familiar to any of the team so using Vivado is very slow compared to Jupyter
- Interface is not user friendly, takes some time getting used to

Jupyter Notebook

Coding

- Connected the board to internet and the Jupyter Notebook web page through ethernet cable using Static IP.

Pro's

- Easy application interface - write code, run code
- Allows us to use Python - many of us are already familiar with the language
- Makes preliminary testing much simpler

Con's

- The board needs to be connected to internet all the time, directly to computer or to wifi router

Narrowing Down

Coding

Jupyter Notebooks vs. Vivado Studio HL*x*

- Jupyter Notebooks is preferred by PYNQ boards as PYNQ is designed for Python, and so is Jupyter.
  - More of “just writing code to the board”
- Vivado Studio significantly more powerful in terms of low-level access, board configurations, etc.
  - More of “designing IP Cores that define external interfaces (i.e. motors, LEDs), constraints (min/max voltages, current, bit masking, etc.)

In terms of shortest time-to-results and learning curve, Jupyter Notebooks seems to be the obvious option.

- Python code is directly movable to Vivado, but will require custom IP Core designs which can take a substantial amount of time and have a steep learning curve.

Narrowing Down

Coding

Next Steps:

- Work with subteams as the PYNQ board experts and help integrate PYNQ board with their equipment
- Setup primary virtual environment for TracSat and create a repository that contains codes for all subteams
- Write System Controller
  - Single program that controls all subprograms (LTS, Propulsion, etc.)
  - Outline created, just needs to be implemented.
- Continue investigating GPIO write speed (currently 2kHz, advertised 200+ kHz)
- Continue transitioning non-PYNQ python code to PYNQ code.

OSS Subteam

Capabilities in Development

OSS

Camera

- Used for getting pseudo-live video stream for the ground station
- openCV in Python and a 1280x720p webcam

LIDAR

- Used for getting current information about the object and location
- Python library to obtain data through USB

PURDUE UNIVERSITY

Camera

OSS

- Requirement: Need to be able to capture video and send it to the ground station
- Camera Native Specification:
  - Logitech HD C525
  - 720p
  - 30fps
  - 24bit
  - FoV: 69°
  - Powered through USB
- Native specifications like resolution and FPS can be altered
- Using openCV python library to get single frames and convert to desired size and color parameters

PURDUE UNIVERSITY

Camera - Python Class

OSS

- Created a class in Python that handles everything about the camera
- LSS can simply call this class to get the most recent frame in bits
- Sample code:

```
video = Video(scale_percent=100)
frame = video.getFrameBits()
```

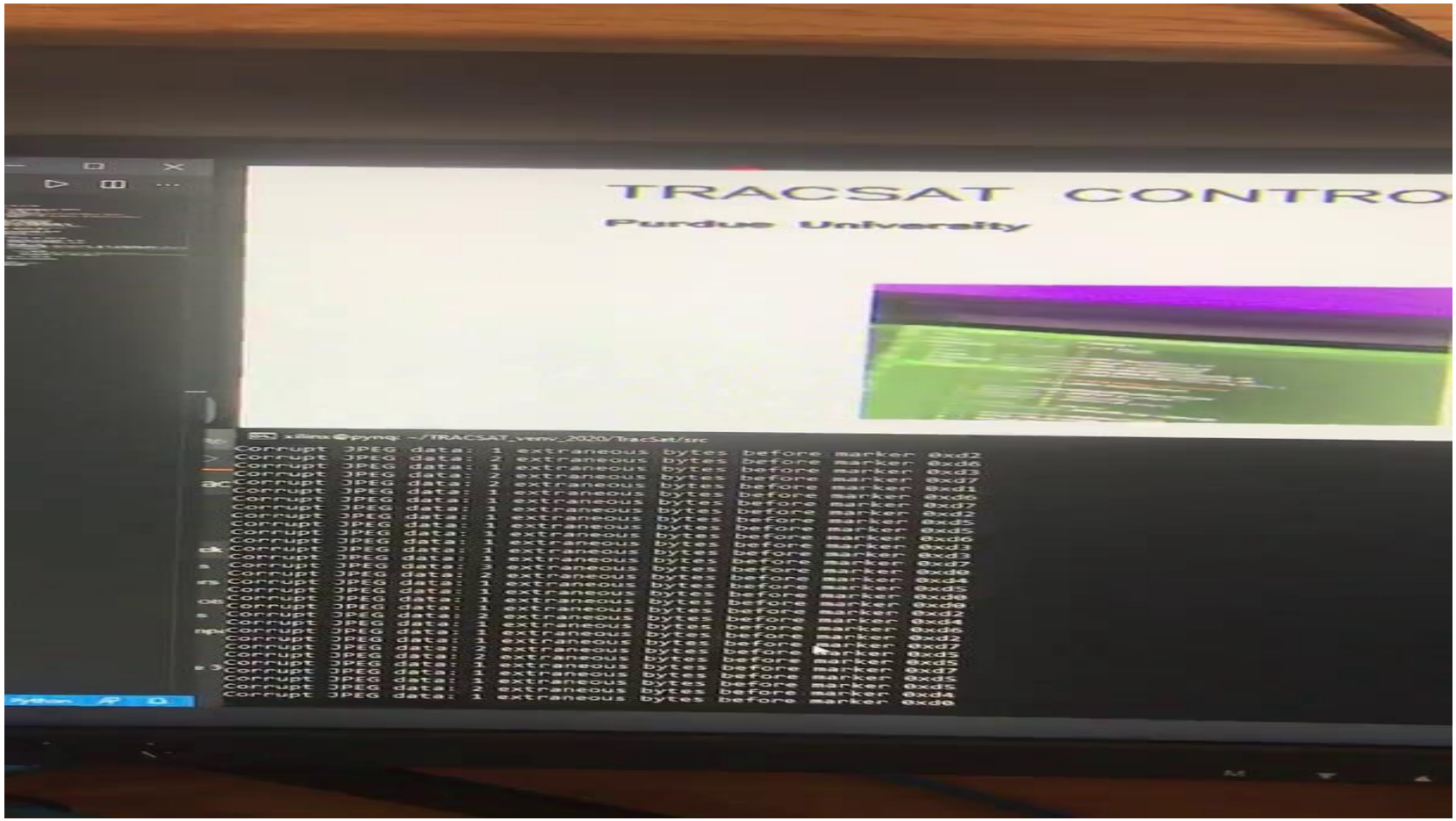
- If you run this code, it will return an array:

frame = `[01001010|11101010|10010101|00101010…]`

- This array can then be fed directly into the laser.
- Each set of 8 bits (1 byte) comprises 1 pixel.
- Note: scale_percent allows you to reduce the resolution of the camera.

Camera - PYNQ Integration

OSS



Video is slow due to rendering it in the screen. If we skip the rendering through the PYNQ board, performance is good

PURDUE UNIVERSITY

Bit rate vs Video values

OSS

To give a sense of what video quality we should expect for 50 kb/s (= 50 kHz)...

Resolution (px)		Frames/s	60	30	10	5	1	0.5	0.1
Width	Height	Color (bit)							
640	480	1	2304	1152	384	192	38.4	19.2	3.84
640	480	8	18432	9216	3072	1536	307.2	153.6	30.72
640	480	24	55296	27648	9216	4608	921.6	460.8	92.16
480	360	1	1296	648	216	108	21.6	10.8	2.16
480	360	8	10368	5184	1728	864	172.8	86.4	17.28
480	360	24	31104	15552	5184	2592	518.4	259.2	51.84
320	240	1	576	288	96	48	9.6	4.8	0.96
320	240	8	4608	2304	768	384	76.8	38.4	7.68
320	240	24	13824	6912	2304	1152	230.4	115.2	23.04
160	120	1	144	72	24	12	2.4	1.2	0.24
160	120	8	1152	576	192	96	19.2	9.6	1.92
160	120	24	3456	1728	576	288	57.6	28.8	5.76
			>500kb/s	500-50kb/s	<50Kb/s				

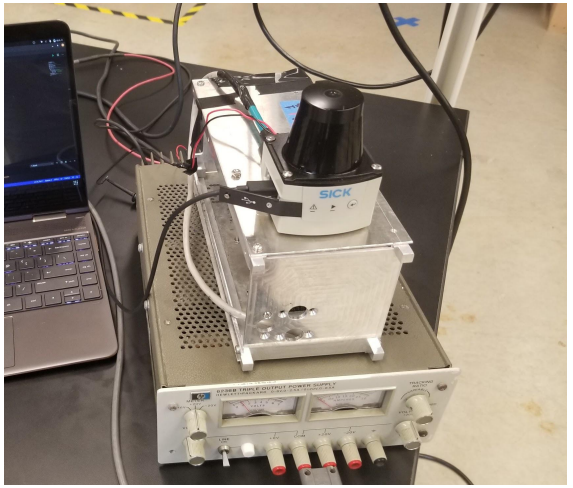
1 bit: ONLY black and white (good for initial testing)
8 bit: Greyscale
24 bit: Color (unlikely at this bitrate)

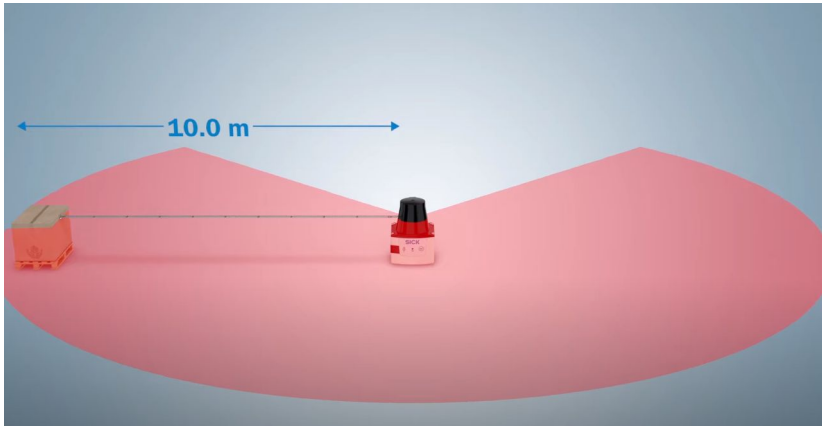
PURDUE UNIVERSITY

LIDAR

OSS

- Requirement: Retrieve distance data, size, and location of object and send that information to the ground station
- SICK TiM561 2-D LiDAR Specifications:
  - Aperture angle - 270°
  - Scanning Frequency - 15 Hz
  - Angular Resolution - 0.33°
  - Working Range - 0.05m to 10m
  - Powered using 8 - 30V DC Power Supply
  - Connected via USB, optional ethernet connection



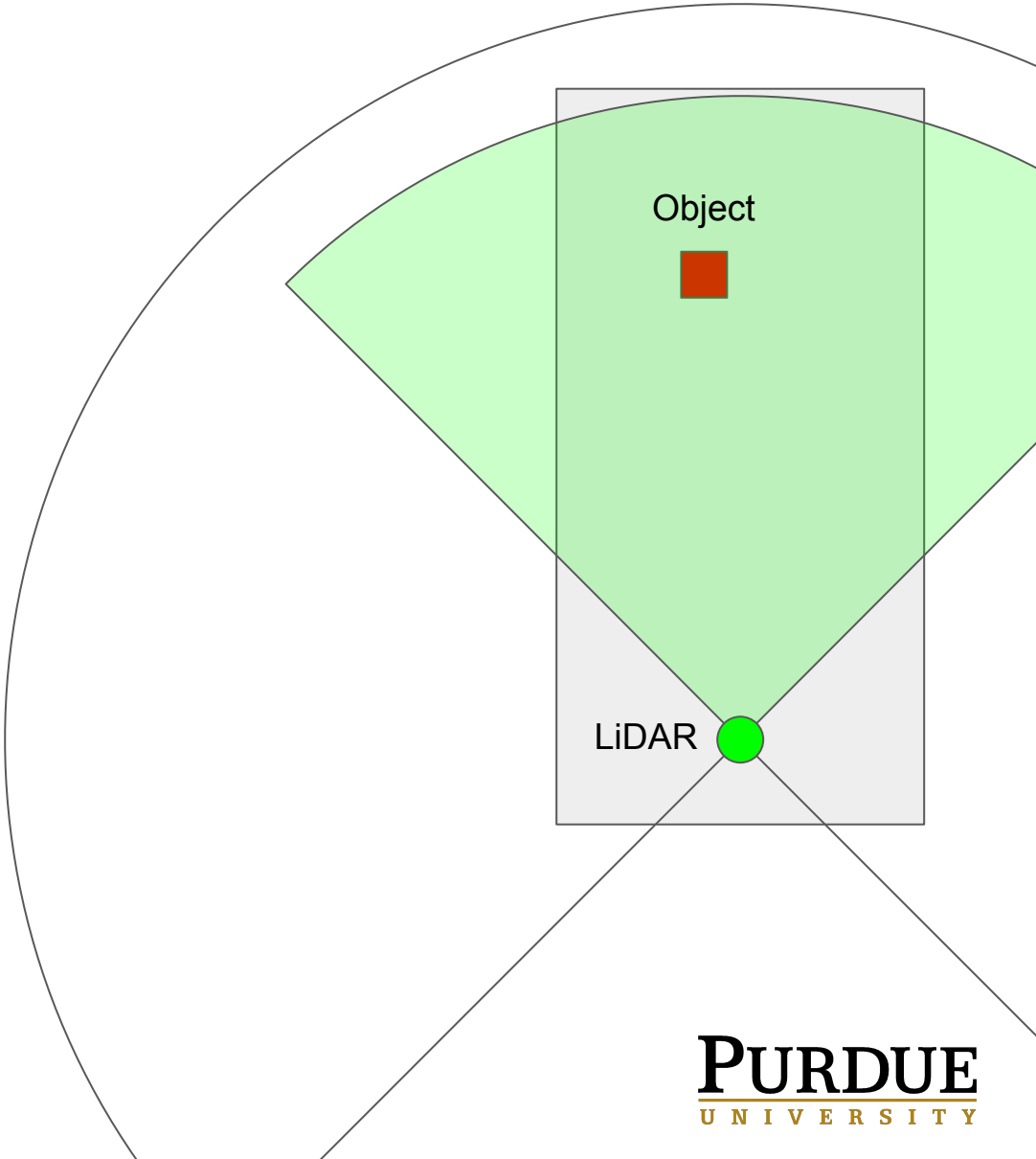


PURDUE UNIVERSITY

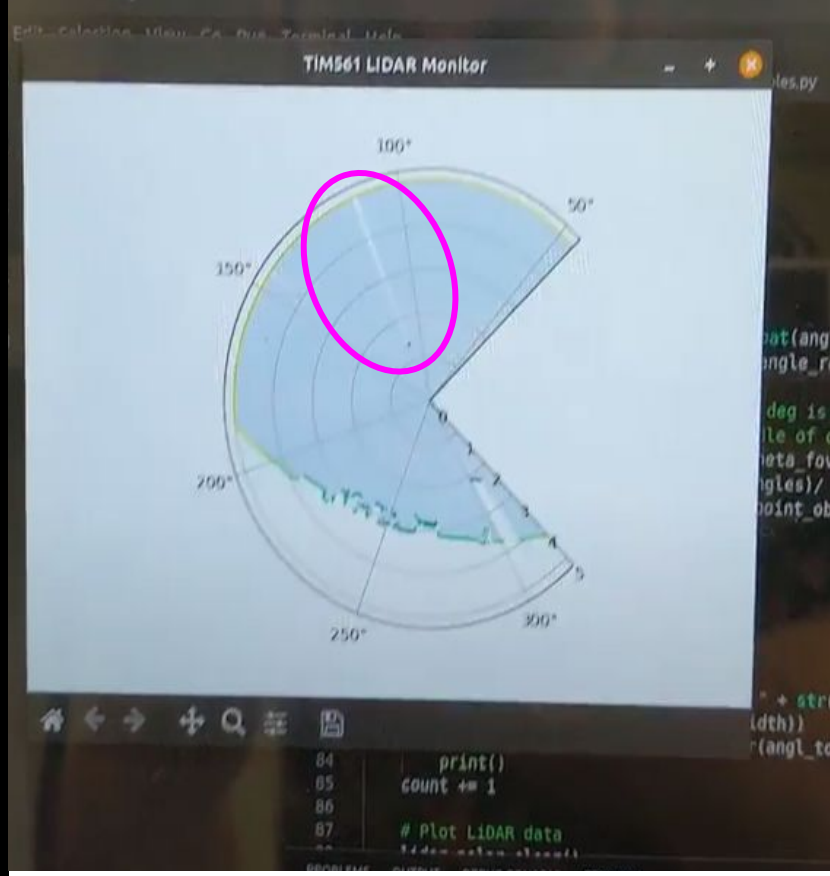
LIDAR - Integration

OSS

- Utilizes the pysicktim library to access and retrieve information from LiDAR
- Retrieves live data about object including:
  - Distance to Object
  - Width of Object
  - Angle Object is Located at (relative to LIDAR)
- Tested and works accurately up to about 1-2 cm for object size
- Maximum distance, field of view, and object detection can easily be changed



PURDUE UNIVERSITY



RMS Distance to Obstacle: 1.39168039913384333
Obstacle Width: 0.024287250117494805
Angle to Obstacle: 21.351484444444445

RMS Distance to Obstacle: 1.395688002384487
Obstacle Width: 0.024357189703657127
Angle to Obstacle: 21.351484444444445

PURDUE
UNIVERSITY


Terminal
Shell
Edit
View
Window
Help
MacBook Pro

LIDAR & Camera - PYNQ

OSS

- We have tested the LIDAR with the PYNQ board and the camera with the PYNQ board individually
- A problem arises
  - LIDAR and Camera both connect to computer using USB
  - PYNQ board only has one USB port
- Solution?
  - Use a USB hub
- We tested this solution with an old USB Hub using USB 1.0 standard
  - Error: Exceeded USB memory bandwidth
- Next tests:
  - Use a new USB Hub using USB 3.0 standard
    - Maybe we exceeded the bus of the old USB hub
  - Using the PYNQ ports, create a USB port and add it in using software
    - Common solution, lots of ways to do this
- Will test new USB hub first as it is less involved.

Narrowing Down

OSS

Next Steps:

- Finalize integration between LIDAR, camera, and PYNQ board

OSS is in an awkward spot…

- Initial team goals were to integrate LIDAR and camera to PYNQ
  - This is mostly completed…
    - Small tweaks may need to happen once we do table tests, but this will not occur for 3-4 months
  - What to do next?

Possible future projects:

- Scaffolding for floating obstacles above granite surface
- Start main code structure to handle all sub teams
  - Work closely with coding

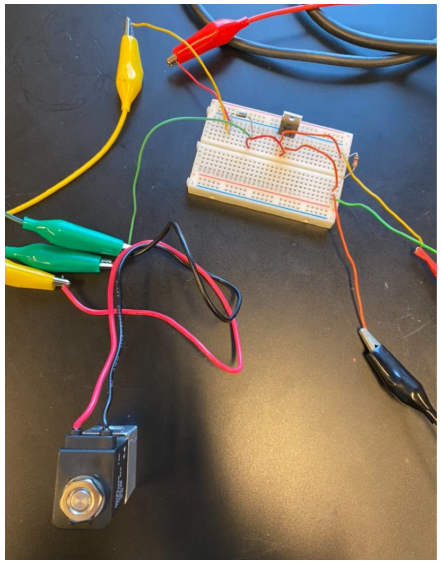
HPL Subteam

Capabilities In Development

HPL

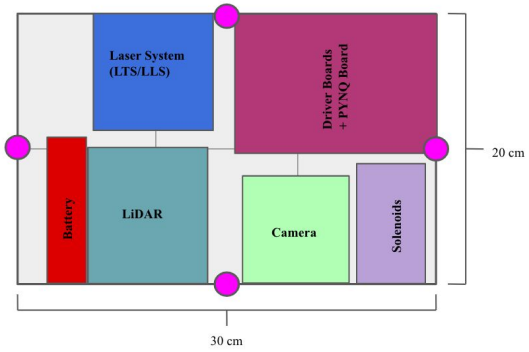
Solenoids

- Used to control the actuation of the thrusters



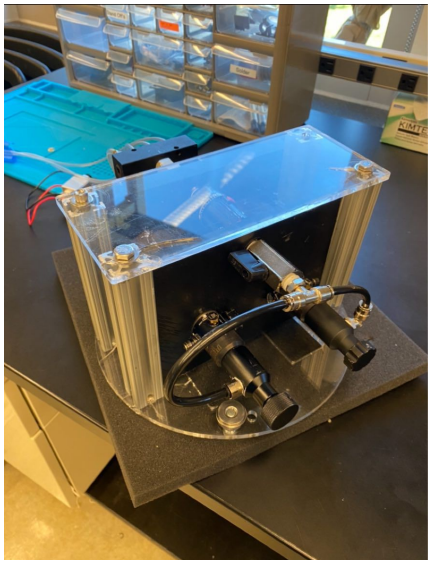
New Cubesat Design

- Used to organize and provide structure to the cubesat



Hardware

- Used to assist with translational motion



PURDUE UNIVERSITY

Solenoids - Specifications

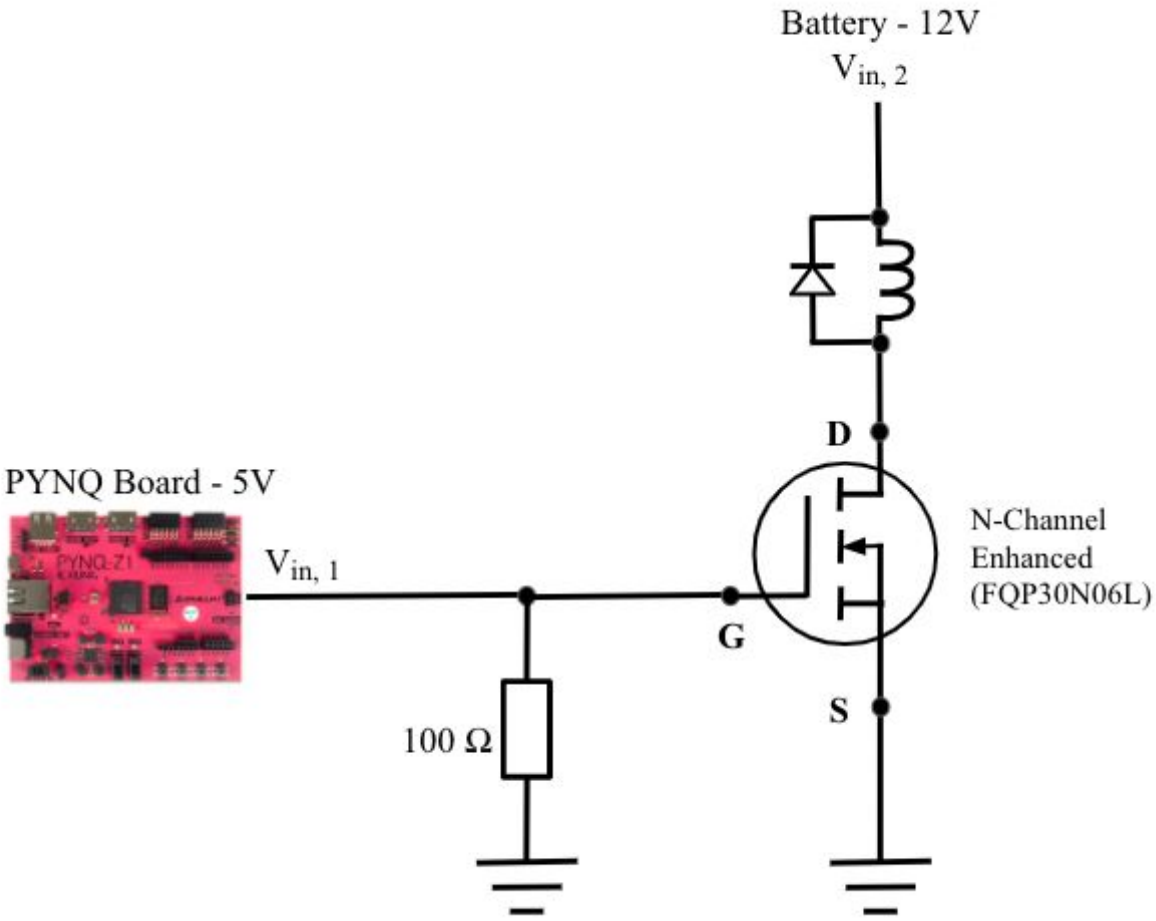
HPL

- Amazon Solenoid - “TAILONZ PNEUMATIC”
  - 12 V, 2-Way Normally Closed
  - Operating Pressure: 25 - 116 PSI
  - Fitting Size: ¼” NPT
  - Response rate: Up to 5 times per second
- Pneumadyne Solenoid
  - 12 V, 2-Way Normally Closed
  - Operating Pressure: 0 - 125 PSI
  - Fitting Size: ¼” NPT
  - Response time to energize: 8-10 ms
  - Response time to de-energize: 10-12 ms

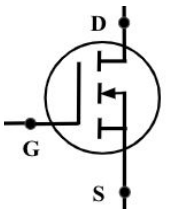
PURDUE UNIVERSITY

Solenolds - Circuit

HPL



Key

MOSFET	
Solenoid	
Diode	
Resistor	

PURDUE UNIVERSITY

Solenoids - Testing

HPL

- Demonstration: How the propulsion system will work using solenoids and circuit
- Tested using two power sources to actuate the thrusters and using the pneumatic system of the cubesat

New Cubesat Design - Allocation

HPL

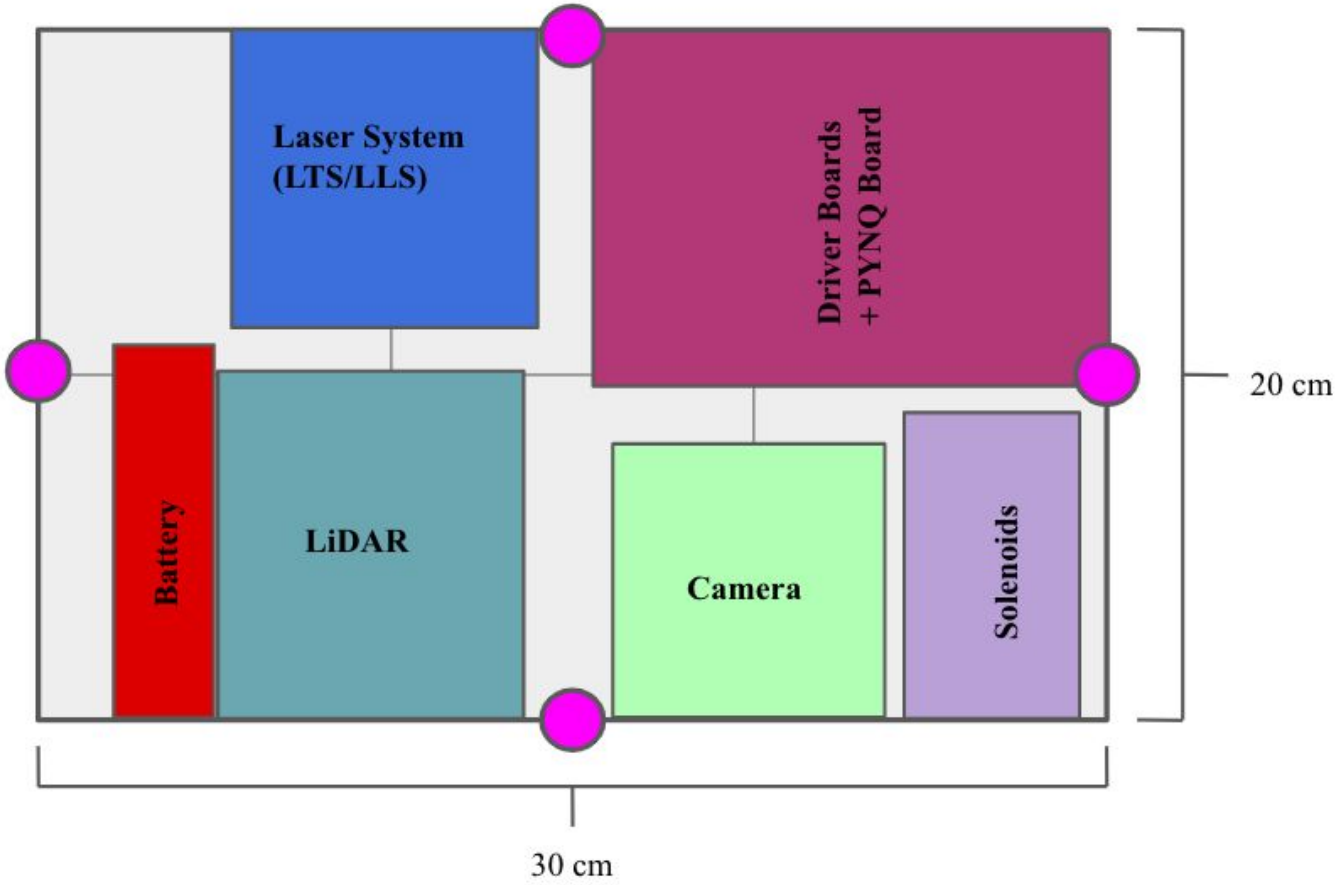
These are the results of the data obtained from each team on the part reporting form:

- Camera and LiDar need to be in the front
  - Camera: 7 x 7 x 5 cm
  - LiDar: 8 x 10 x 10 cm
- Driver boards + PYNQ board can be stacked vertically
  - Will allocate a 15 x 11 x 2 cm space for each one
- Battery
  - 10.5 x 3.5 x 3.5 cm
- With a vertical LTS system, we need to allocate an 8x8 cm square base for the part

New Cubesat Design - Organization

HPL

- Proposed model considering orientation and position requirements of each team



PURDUE UNIVERSITY

Hardware - Bearings, Drift Solution

HPL

- Tested last year's bearings and the new bearings: the new ones work well
- Tested last year's solution for the drift problem (adding weight) with new bearings
- Demonstration: Cubesat with minimized drift when weight is added

Future Work

HPL

Next Steps:

- Testing solenoid circuit with PYNQ board
  - To confirm that the circuit will function properly with the PYNQ board as a power source, since our tests were using two external power sources
- Testing all 4 solenoid circuits with PYNQ board
  - To verify that the PYNQ board can run all 4 circuits simultaneously
- Finalize, create & order PCB for solenoid valve circuits
- Create new CAD model

LLS Subteam

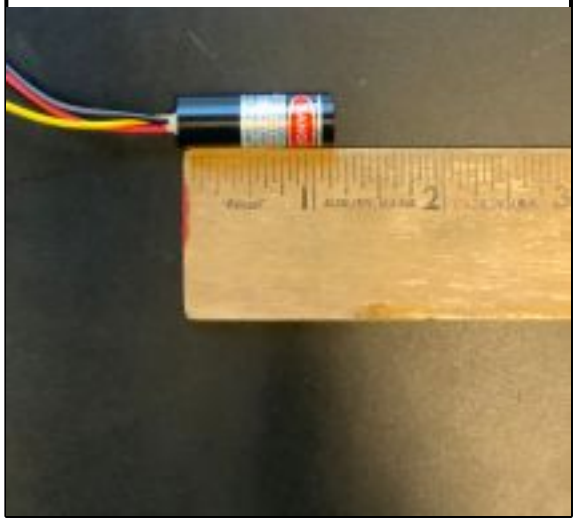
PURDUE UNIVERSITY

Capabilities In Development

LLS

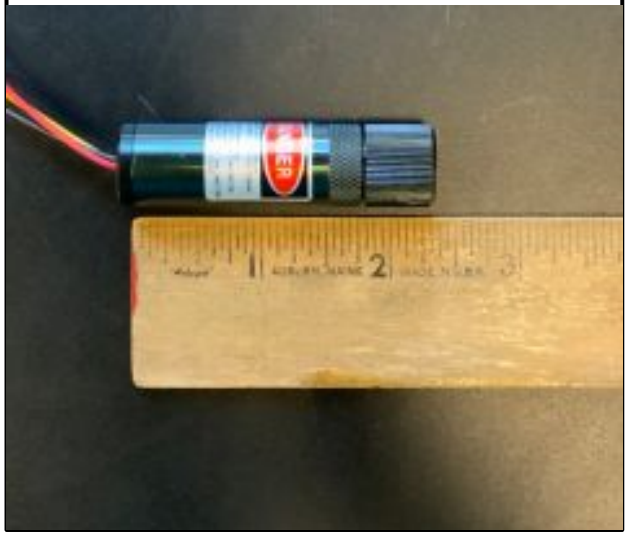
Adafruit Laser

- Advertised at 50KHz
- $18.95



Civil Laser

- Advertised at 1MHz
- $49.00



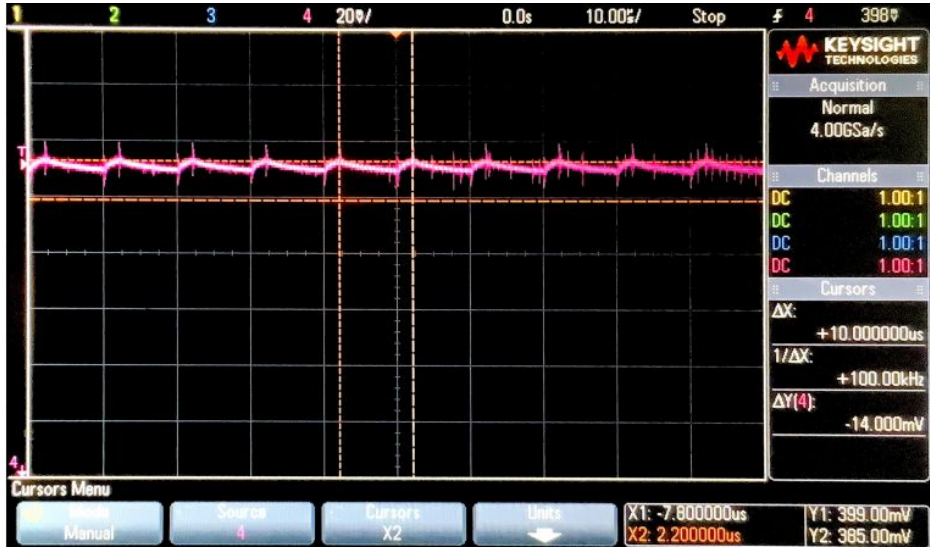
PURDUE UNIVERSITY

Adafruit Laser

LLS

- Tested at over 10 meters with no transmission issues
- Modulated with PYNQ board with no issues
- Capable of at least 100 KHz (more testing needed)

45 KHz, 20% Duty Cycle



100 KHz, 20% Duty Cycle

PURDUE UNIVERSITY

CMI Laser

LLS

- Capable of at least 100KHz (more testing needed)
  - Should be capable of 1MHz
- Much larger in size
- Wiring is exactly the same as Adafruit

100 KHz, 20% Duty Cycle (with photoresistor

PURDUE UNIVERSITY

Capabilities (Cont.)

LLS

- Both lasers can be ran directly from the PYNQ board (or buffer)
- Both lasers are able to be focused easily
- Both lasers are capable of meeting minimum frequency requirement of 50KHz

Narrowing Down

LLS

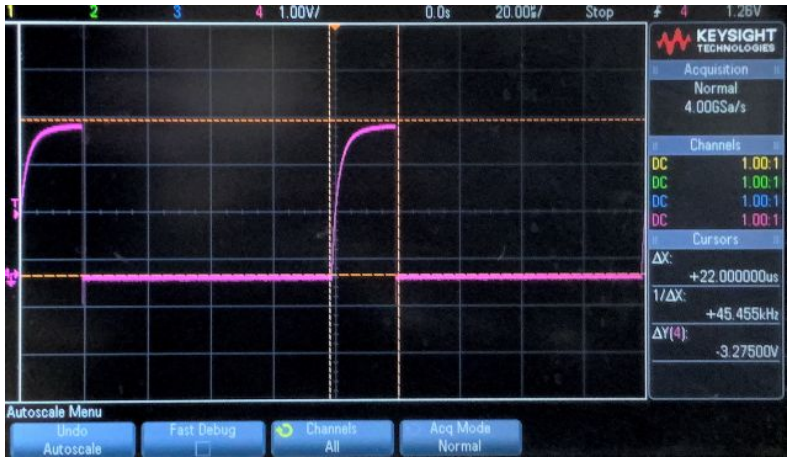
Major decision points:

- Speed (main factor)
- Size

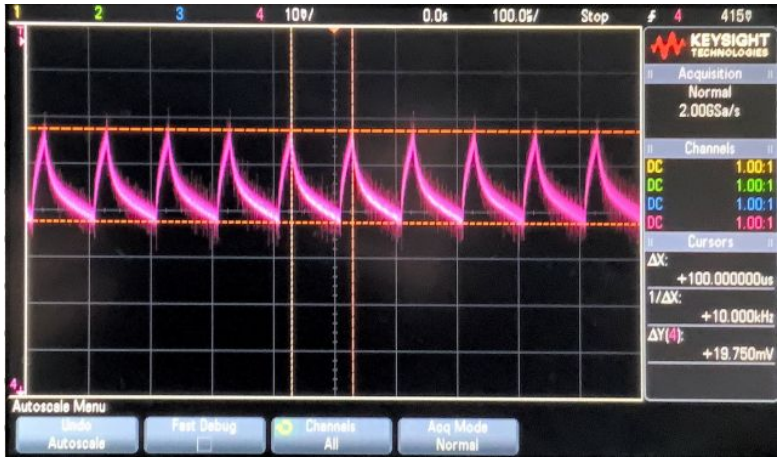
Next Steps:

- Work with LTS to use photodiodes to further test laser speed capabilities
  - (see next slide)
- Work closely with OSS and Coding teams to work on transferring laser communications code to PYNQ.

Narrowing Down (Cont.)

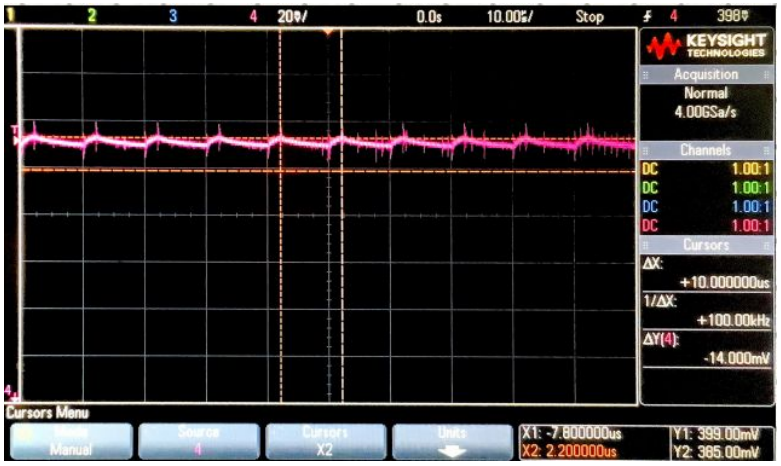


10 KHz, Photoresistor



10 KHz, Photodiode

100 KHz, Photoresistor



100 KHz, Photodiode

PURDUE UNIVERSITY

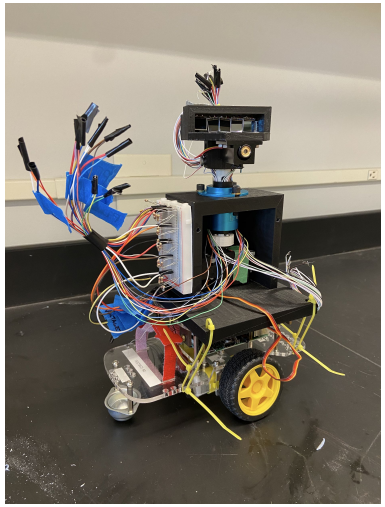
LTS Subteam

Capabilities In Development

LTS

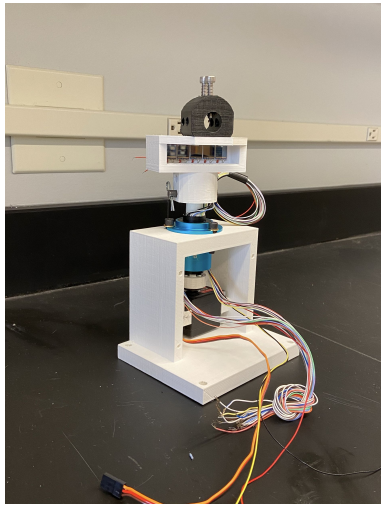
TracSat System

- Used to maintain laser link with Ground Station
- Utilizes 5 photodiodes and a 360 degree rotation motor



Ground Station System

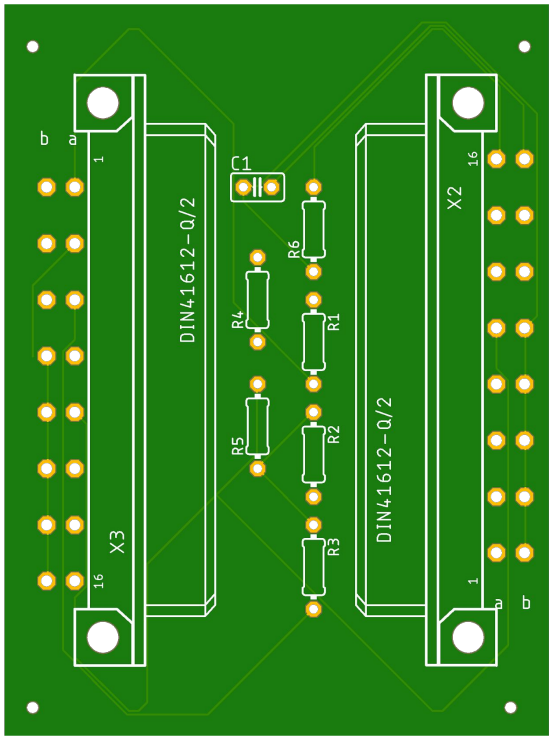
- Used to maintain laser link with TracSat
- Utilizes 5 photodiodes and a 360 degree rotation motor

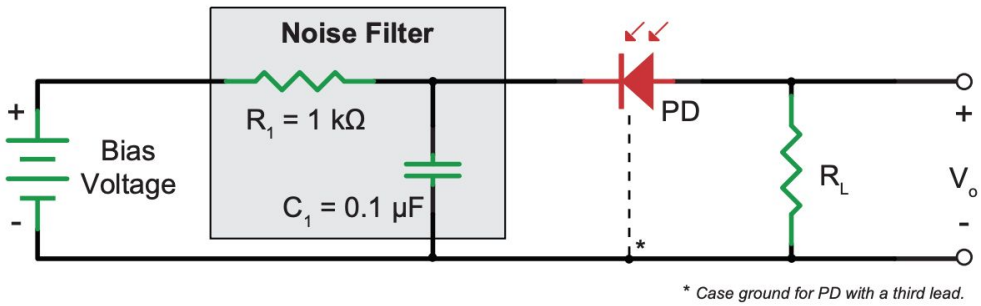


Optimization - PCB

LTS

- Problem Description: Ground station had a breadboard, but struggled to maintain a stable connection between Raspberry Pi and photodiodes
- Solution: Constructed PCB with optimized filtering system to replace current system
  - Reduced area by 31%
  - Extended lifetime
- Past filtering system filtered every signal separately, new system utilizes one common filter
- Most importantly, poor connection problems are resolved



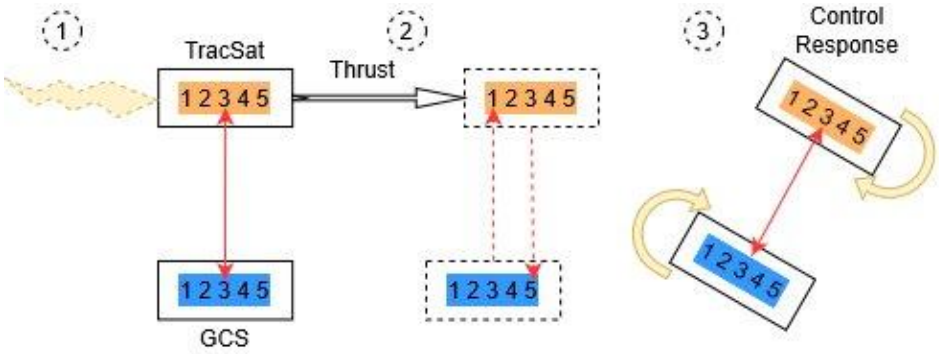


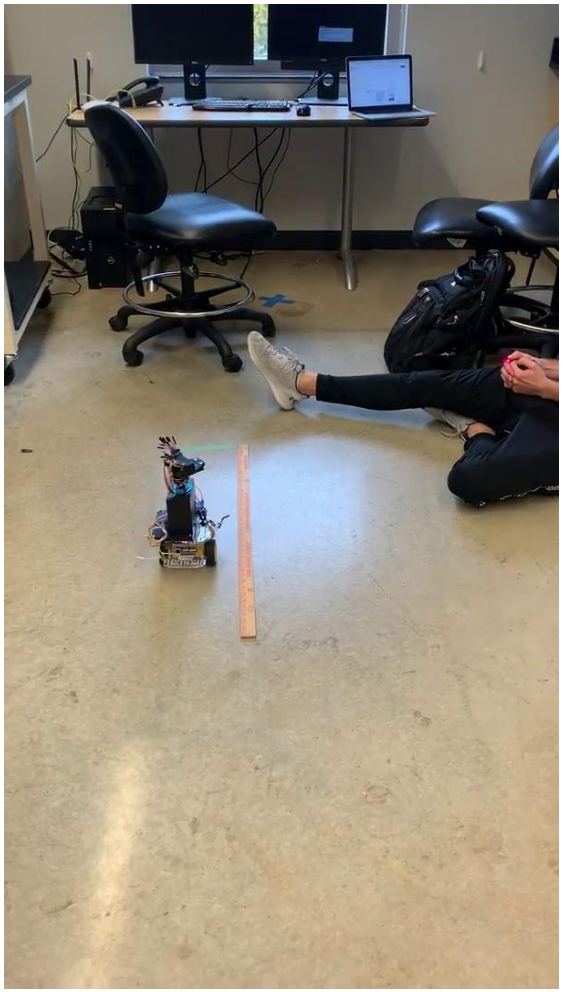
PURDUE UNIVERSITY

Optimization - Control Code

LTS

- Requirement: Able to maintain lock at a speed of 10 cm/s
- Servo spin rates for a given photodiode
  - Consistent across both systems
    - Same rates for 1&5, 2&4
- Delay for middle photodiode (#3)
  - Reduced to maintain lock at higher speeds
  - Video is moving at ~18 cm/s

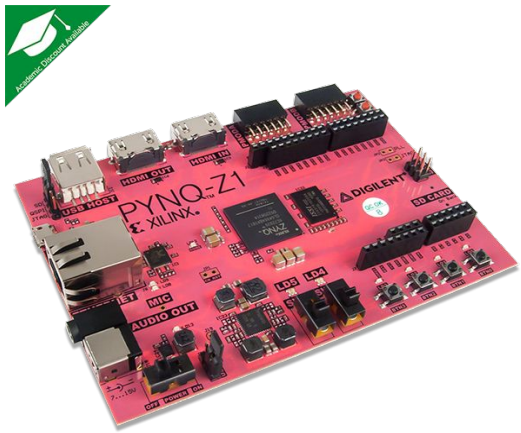




PURDUE UNIVERSITY

Integration

PYNQ



- Initially tried running the original code on Jupyter: ran into issues with libraries
- Created my own module on Jupyter using the the easygopigo3 source code: still need more libraries
- Productive conversation with Pol: scrap the gopigo
- Productive conversation with Adam: found information regarding PWM for servo control

Narrowing Down

LTS

Major decision points:

- Rotational rates for the system
  - Maximum optimal rate while taking into account the moment created

Next Steps:

- Implement PCB onto TracSat and Ground Station
- Dual system test (TracSat & GS)
- Further Optimization of Tracking Code
  - Rotation Rates
  - Delay times
- Final test with both systems and PYNQ board

Thank You

- Questions?

PURDUE UNIVERSITY