ECD304 – ML Autonomous Car

Final Design Presentation

November 18th, 2022

Aslam Aliyu, Andrew Deoki, Patrick Nilan & Daniel Nissim

Sponsored by LOCKHEED MARTIN

Our Team

Daniel Nissim

CoE Major
Project Lead

Aslam Aliyu

CoE Major
Software/Finance Lead

Andrew Deoki

EE Major
Electrical/Mechanical Co-Lead

Patrick Nilan

CoE Major
Electrical/Mechanical Co-Lead

Project Mentors

Project Advisor

Professor Vladimir Nikulin

Industry Mentors

Thomas Gaska

Thomas Norton

LOCKHEED MARTIN

Agenda

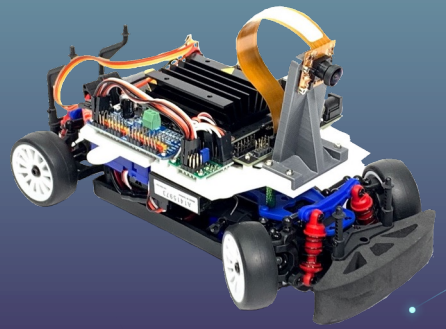
Executive Summary

Objective: Develop an autonomous vehicle built on the NVIDIA JetRacer open-source Artificial Intelligence/Machine Learning (AI/ML) platform

Hardware Used	Software Used
• LaTrax Rally RC Car Kit • Jetson Nano Developer Kit • CSI Camera • Inertial Measurement Unit (IMU)	• Jupyter Notebook • Python • JetCam, OpenCV, PyTorch

Key Capabilities:

- Line detection
- Variable throttle
- Data collection via sensors



Use Cases

Our Use Case

- RC Car will run autonomously on designed track with commands sent via WiFi connection from computer
    - RC Remote available as backup to control car from a major crash
- All code and libraries stored on Jetson Nano
- Computer manages image data collection to be stored on Jetson Nano
- Jetson Nano will process images to develop an autonomous driving model via machine learning algorithm
- Computer used to correct autonomous driving model in live driving session via GUI with image data from camera

Real World Applications

- Full-scale version can be developed for various deployments
    - Robotaxis
    - Autonomous personal vehicles
- Convert platform to serve various industries
    - Personal home robots
    - Reconnaissance with computer vision

Requirements

Derived Requirements

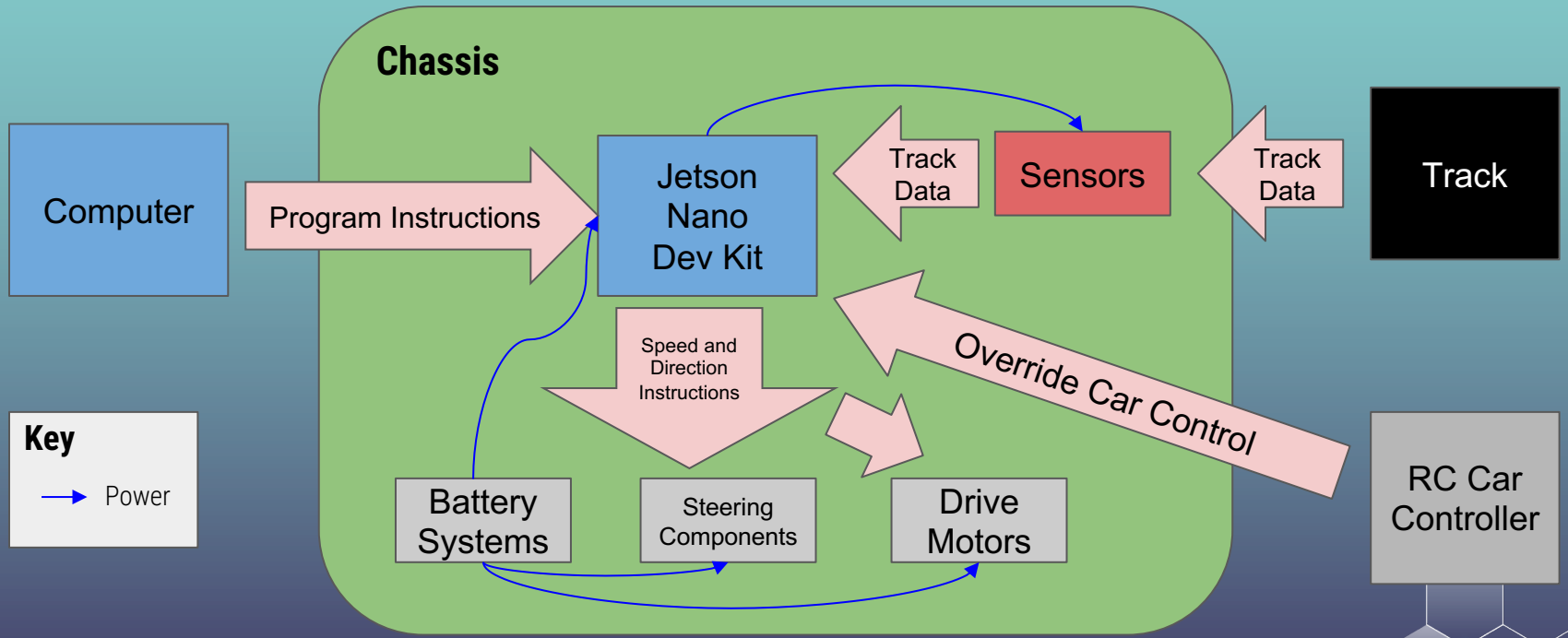
- The car shall use the NVIDIA JetRacer open-source platform utilizing the LaTrax Rally RC car kit
- The car shall use the Jetson Nano Developer Kit
- The car shall operate around the track autonomously
- The car shall be capable of line following
- The model shall use (x, y) positional data for optimal direction inputted with a user mouse click
- The car shall be capable of variable throttle control for different driving scenarios

Stretch Goals

- The car should be capable of object detection i.e. stop signs, people and red or green stop-and-go strips
- The user-interface (UI) should show a bounding box around detected objects
- The car should stop when it sees a stop sign, a person or a red strip on the track
- The car should avoid an obstacle, such as a cone, when detected on the track

High-Level System Design

High-Level Operational Context



Development Considerations

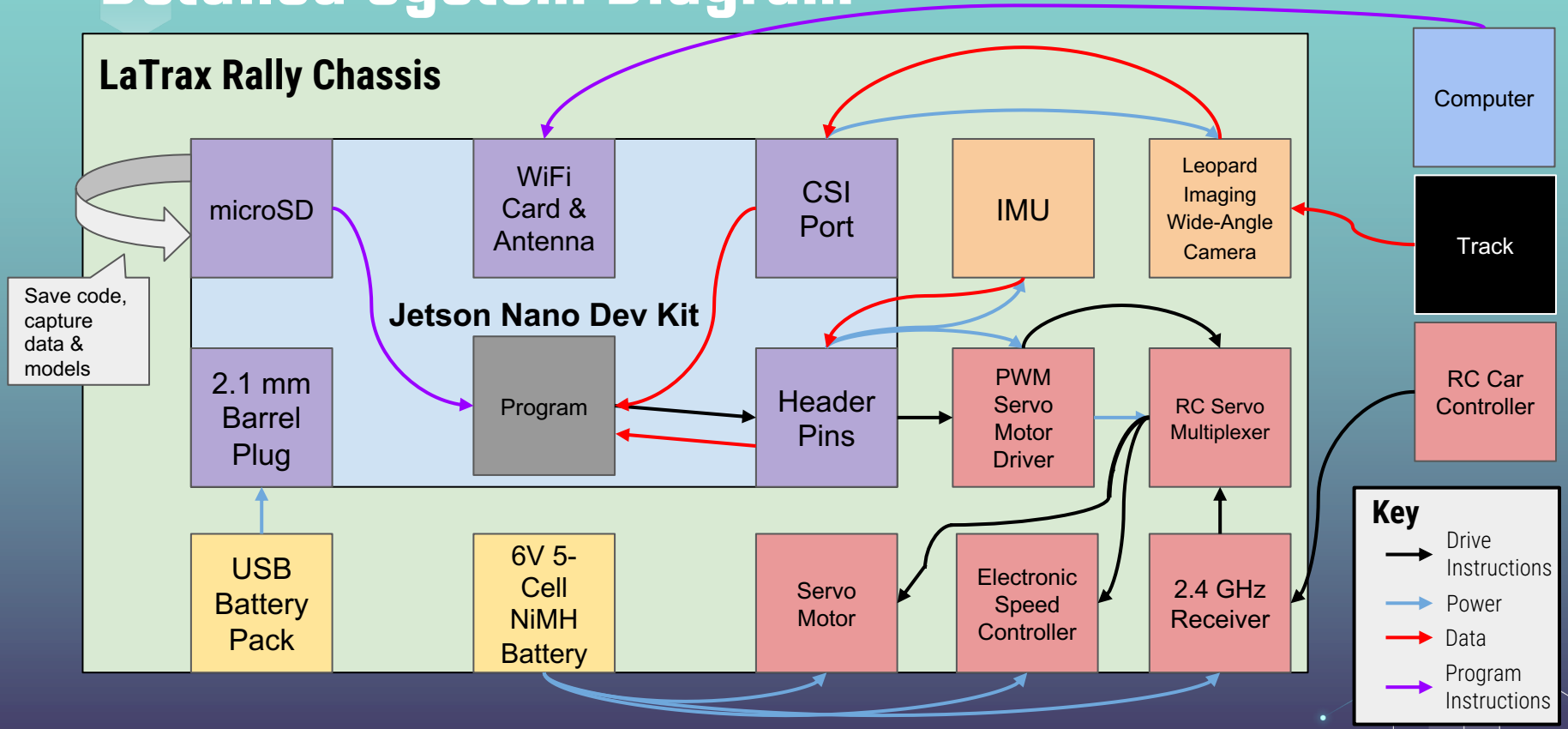
Three open source platforms were evaluated to build the ML Autonomous Car

Platform	Trade Study Score	Pros	Cons
NVIDIA JetRacer	87.25%	• On-device training • Well documented code that can be easily read and altered	• Expensive • Jetson Nano may be hard to procure
DeepPiCar	84%	• Easy to assemble kit • Built-in support for sign and object detection	• No simulation support • Small frame limits expansion
Donkey Car	89.25%	• Can be used with Donkey Car Simulator • Ideal frame to add additional sensors	• Training GUI is a “black box” → limit adjustments to code

- Several team members noted they would have scored the JetRacer higher if we could procure the Jetson Nano in time
- JetRacer was selected due to use of last year’s Jetson Nano and the “black box” nature of the Donkey Car’s training GUI

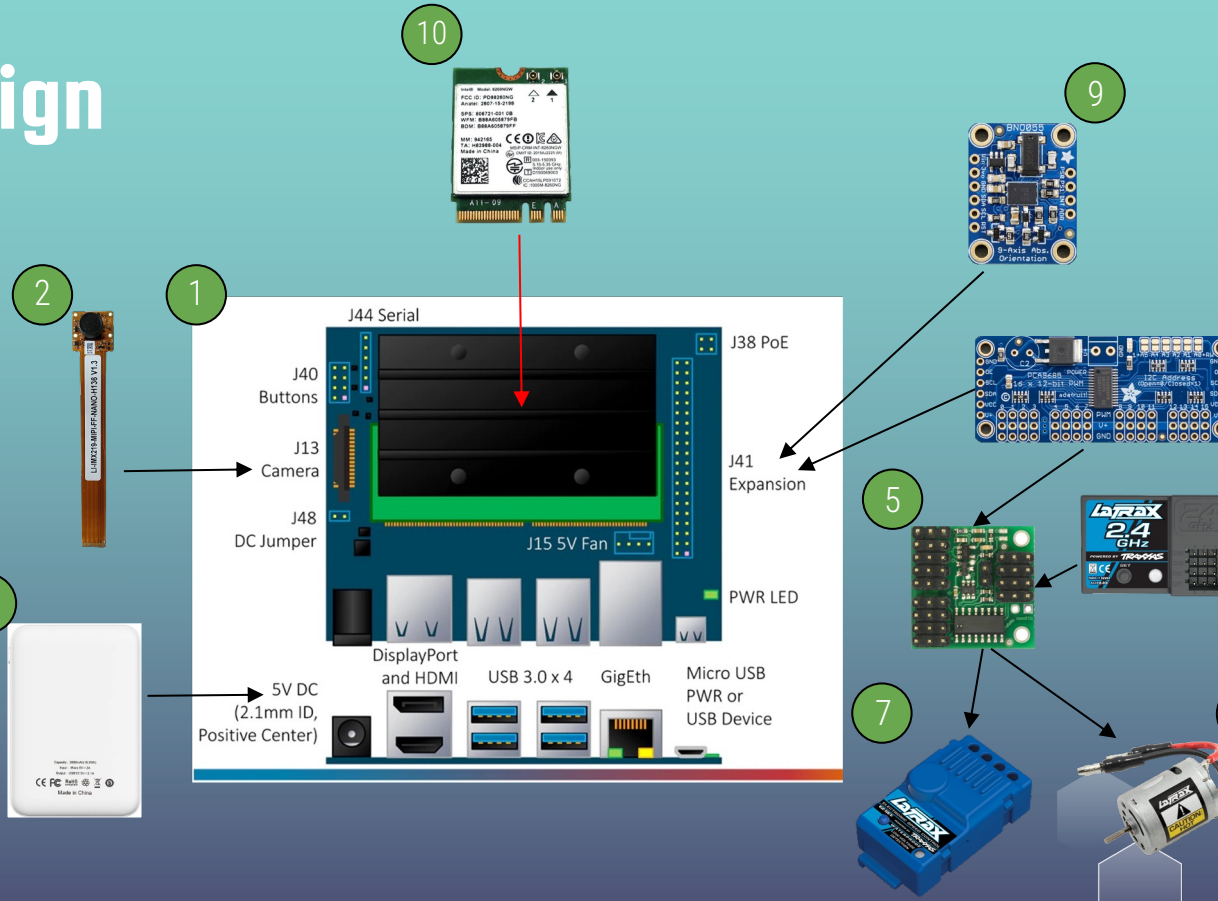
Detailed Design Overview

Detailed System Diagram



Hardware Design

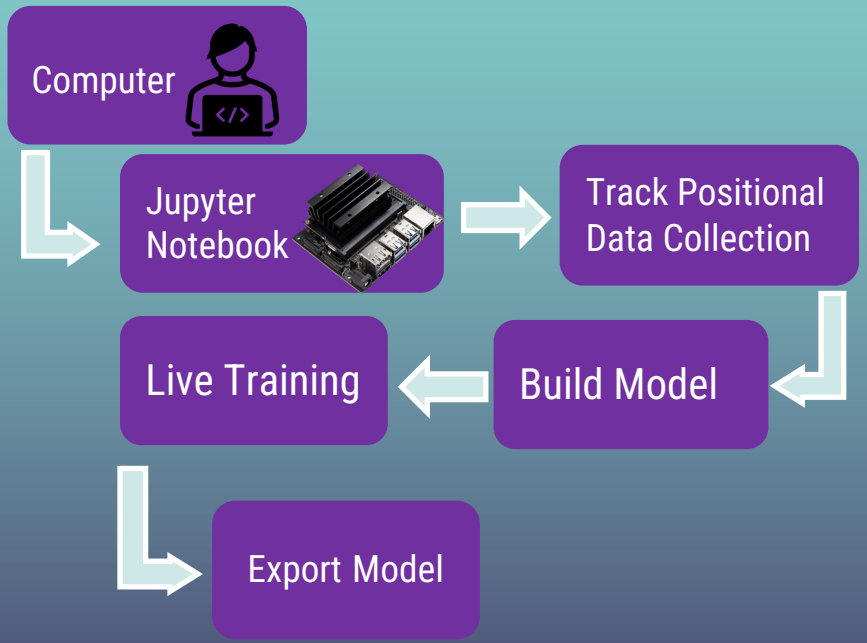
1. Jetson Nano Dev Kit
2. Leopard Imaging Wide-Angle Lens Camera
3. USB Battery Pack
4. PWM Servo Motor Driver
5. RC Servo Multiplexer
6. 2.4 GHz Receiver
7. Electronic Speed Controller
8. Servo Motor
9. Inertial Measurement Unit
10. WiFi Card



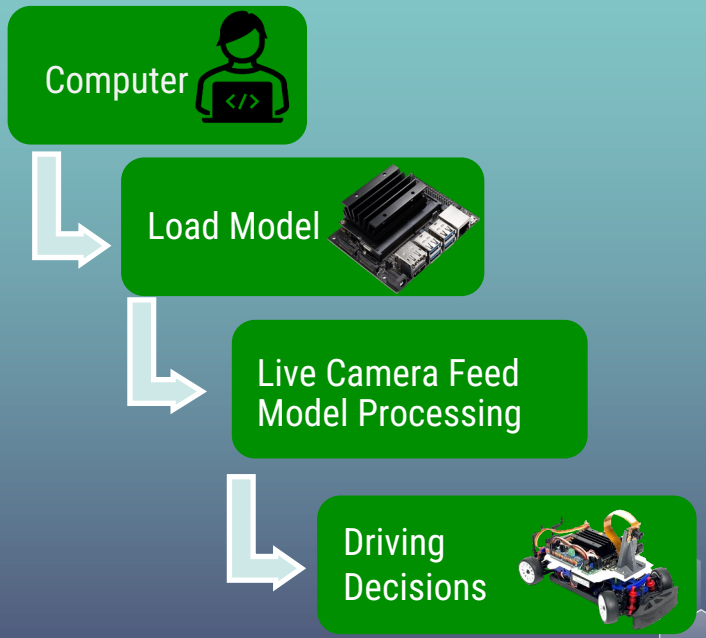
Finished Build

Software Design

ML Model Training



Autonomous Driving

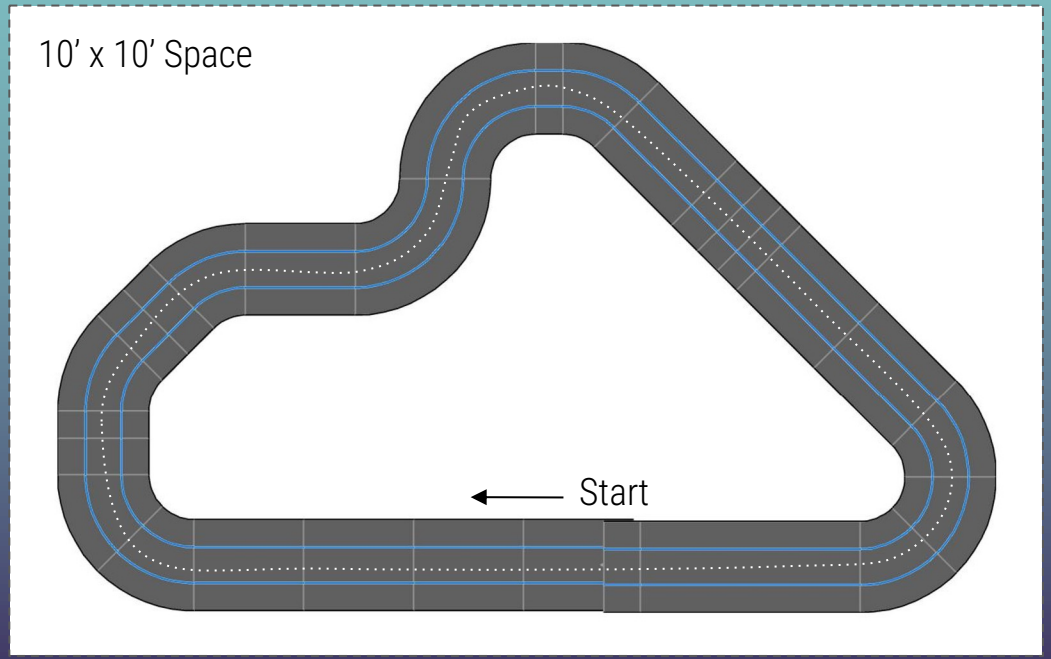


Live Training View

Track Design

Key Features:

- Straightaways for high speeds
- Curves
- Chicane to challenge car with two consecutive low-speed turns

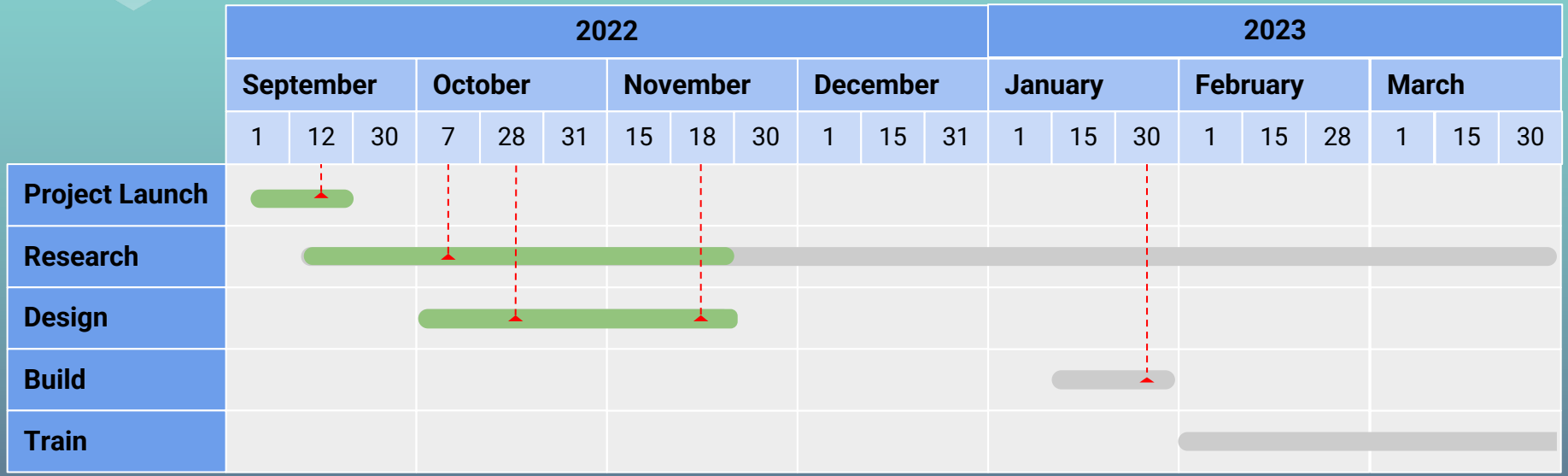


Materials:

- Blue painter’s tape for boundaries
- White dashed painter’s tape for optimal path to follow

Plan of Action

Project Plan



Sept 12 — Project Launch

Oct 7 — Trade Study

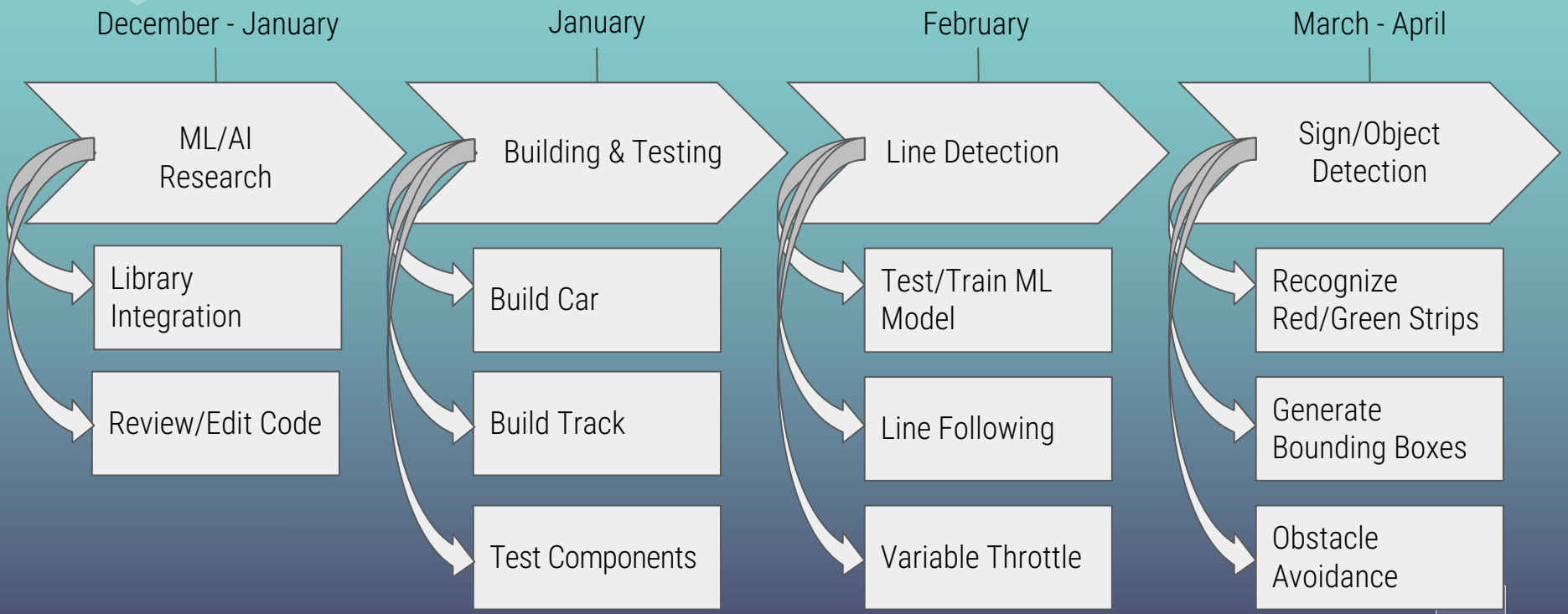
Oct 7 — System Concept

Oct 28 — Preliminary Design

Nov 18 — Final Design

Jan 30 — Finish Building

Implementation Phase Plans



Financial Summary

RC Car: Traxxas LaTrax Rally	$120
Camera: Leopard Imaging LI-IMX219	$29
USB Battery: Attom Tech Ultra Compact 5000-mAh	$15
PWM Servo Motor Driver: Adafruit 16-Channel 12-Bit PWM/Servo Driver	$15
IMU: Adafriut 9-DOF Absolute Orientation IMU	$35
RC Servo Multiplexer: Pololu 4-Channel Multiplexer	$12
SD Card: SanDisk 64GB	$11
WiFi Card: Intel Dual Band Wireless	$12
Miscellaneous	$110
Shipping	$111
Total	**$470**

Risk Analysis

- **Risk:** Component integration issues
  - **Mitigation:** Reference last year's design and other successful builds
- **Risk:** Library flexibility/complexity
  - **Mitigation:** Conduct extensive research on libraries and other platforms to develop alternative code as needed
- **Risk:** Ability to train model with necessary variables
  - **Mitigation:** Research methods for adding variable throttle to develop multiple strategies for deployment
- **Risk:** Time to complete project
  - **Mitigation:** Complete necessary preparation over winter break and plan agile sprints for flexibility during spring semester

Questions?

Appendix A: Management & Engineering Tools

Management Tools

- Google Drive
- GitHub
- Latex
- Jira
- Zoom

Hardware/Software Tools

- JetCard image
- Jupyter Notebook
- Python
- Software packages/libraries
  - JetCam
  - OpenCV
  - PyTorch
  - Torch2trt
- Fusion 360
- 3D Printer
- Soldering Iron

Appendix B: Trade Study

			NVIDIA JetRacer				DeepPiCar				Donkey Car			
	Weight	Score	Dan	Aslam	Andrew	Patrick	Dan	Aslam	Andrew	Patrick	Dan	Aslam	Andrew	Patrick
Meets Requirements	35%	1-5	5	5	5	5	4	4	4	5	4	4	4	4
HW/SW Environment & Support	25%	1-5	5	4	4	4	4	5	4	5	4	5	5	4
Cost	10%	1-5	3	2	3	2	5	5	5	5	4	5	4	4
Availability	10%	1-5	3	3	3	3	4	4	4	4	4	4	5	5
Room for Expansion	20%	1-5	5	3	3	5	2	3	4	2	2	5	5	5
Final Score (Weighted)	100%		97	80	82	90	76	86	86	88	74	96	96	91
Final Score (Average)				**87.25**				**84**				**89.25**		