Bringing ROS to the FIRST Robotics Competition

ZEBRACORNS
Team900.org

Niall Mullane NCSSM
Olivia Fugikawa NCSSM
Ryan Greenblatt Brown University
Anja Sheppard UT Dallas
Kevin Jaget NVIDIA
Marshall Massengill VMware
Meg Massengill IBM

What we're doing:

- 40 high schoolers
- 10 mentors
- Yearly competition
- Integrating latest tech
- Inspiring others

What we strive to do:

- Win competitions
- Push boundaries
- Fully automate robot

ZEBRACORNS

TEAM900.ORG

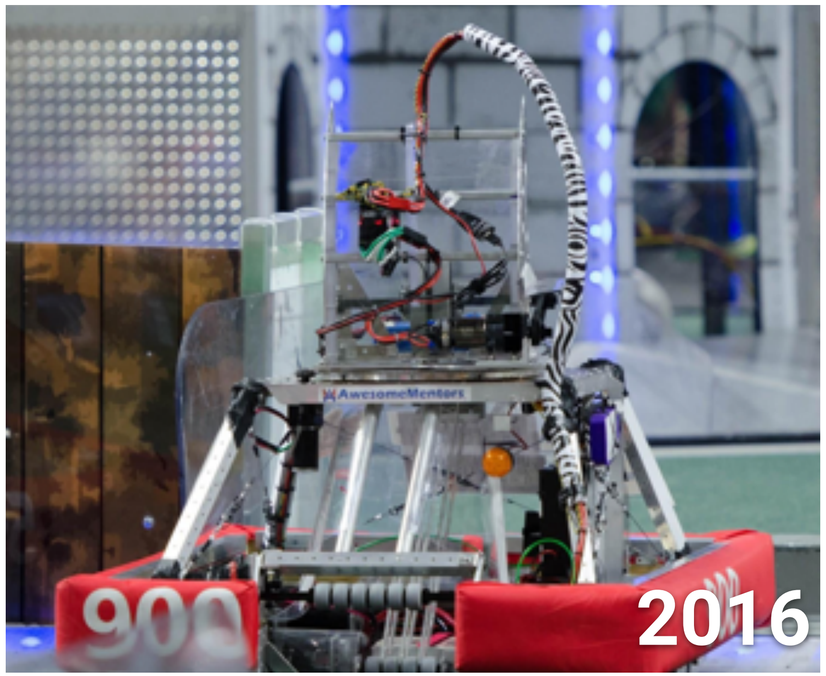
FIRST®

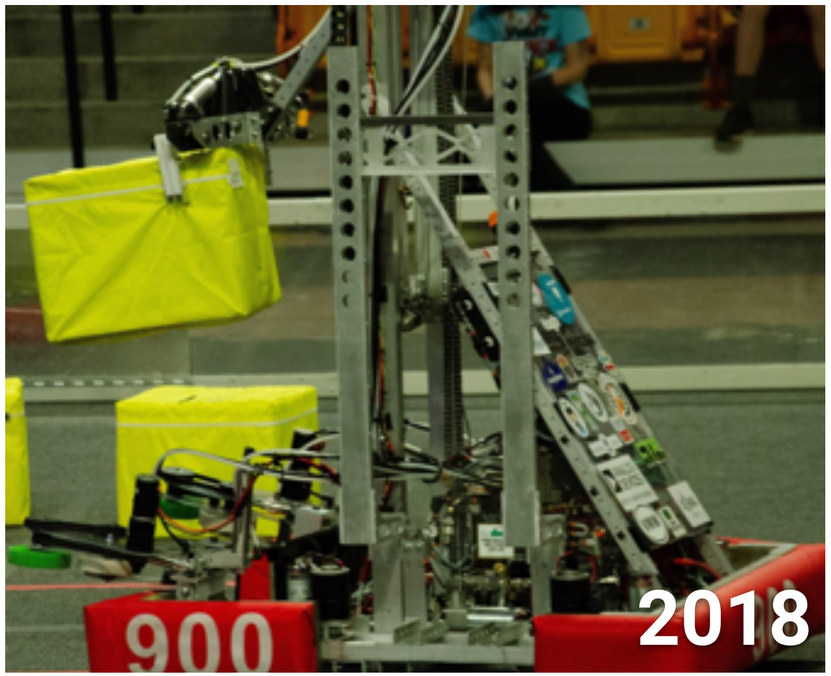
For Inspiration and Recognition of Science & Technology

FIRST® LEGO® LEAGUE JR.	FIRST® LEGO® LEAGUE	FIRST® TECH CHALLENGE	FIRST® ROBOTICS COMPETITION

ZEBRACORNS
TEAM900.ORG

FIRST Robotics Competition (FRC)





FRC Robots

Hardware:

- ~150 lbs
- Size limited
- Parts limited by cost and rules

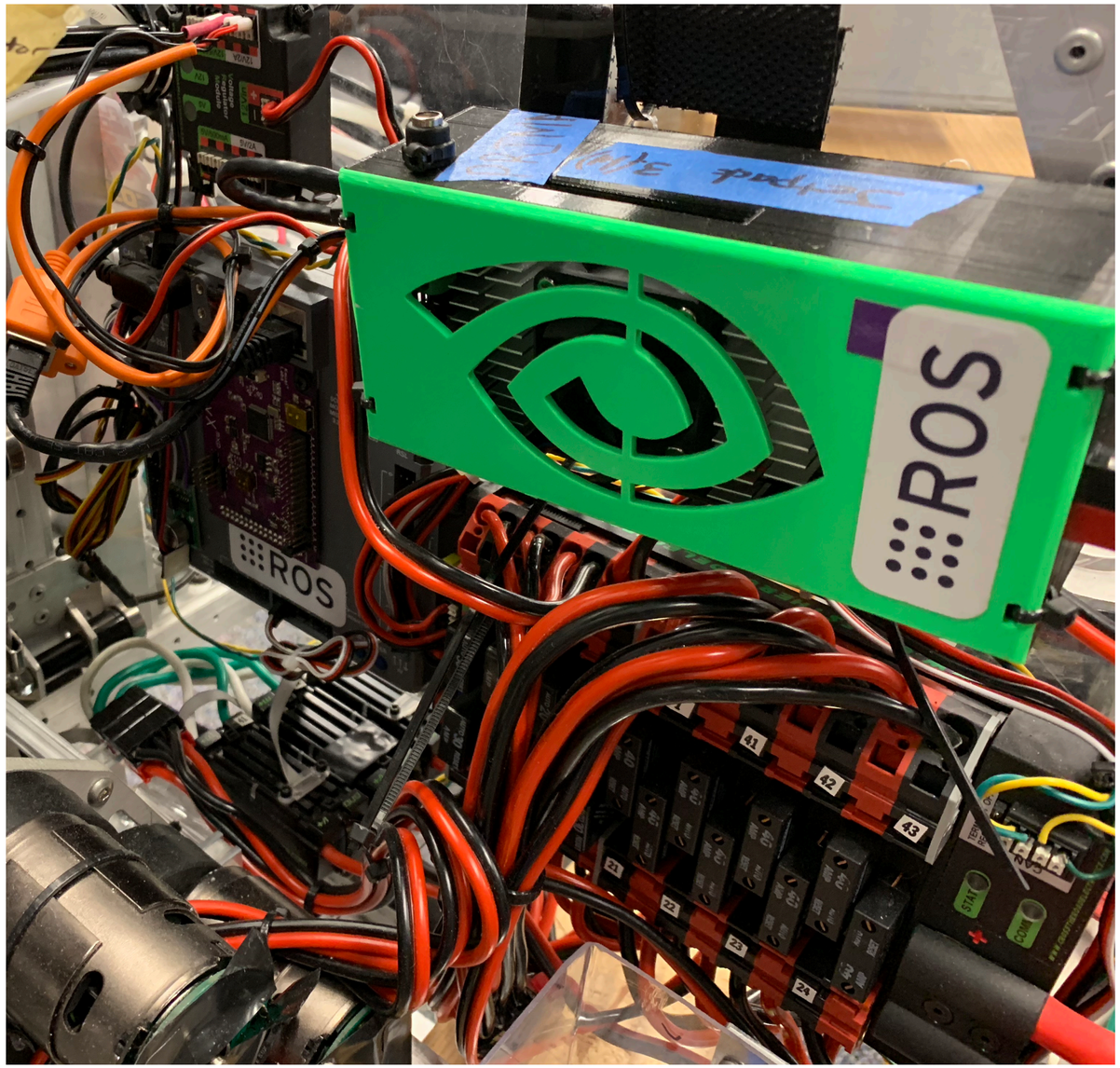
Software:

- NI roboRIO CPU/IO module
- C++, Java, & LabVIEW
- WPIlib HAL

Why ROS?

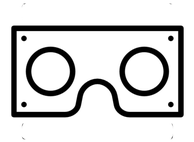
Zebracorn Robots:

- Multiple processors (Jetsons)
- Sensors:
  - ZED stereoscopic camera
  - LIDAR
  - mmWave RADAR
  - Lots of others…
- Data acquisition and analysis
- Dreams of full automation



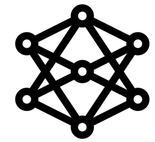
How did we get here?

2012 — Stereoscopic vision



Jetson TK1 and ZeroMQ — 2015

2016 — TX1 running neural networks



Team discovers ROS — Summer 2016

2017 — ROS for LabVIEW

Tufts UNIVERSITY

Initial working ROS build — Summer 2017

2018 — First FRC robot to run full stack

ROS

roboRIO: Hardware & Software

Hardware:

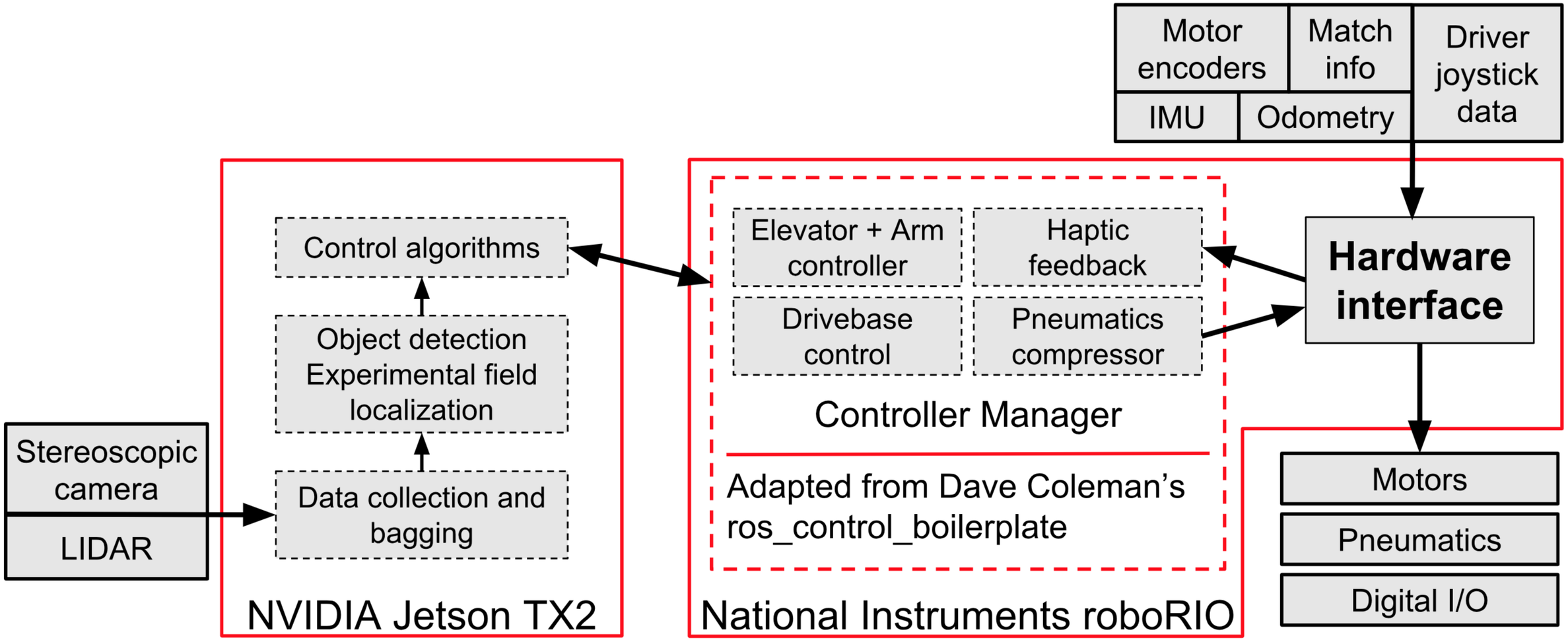
- Dual ARM Cortex-A9 @667Mhz
- 256MB RAM
- 512MB File System
- Analog, Digital I/O, SPI, CAN

Software:

- National Instruments RT Linux
  - No pre-built ROS binaries
- ROS source code
  - Hacked os_detect.py
- Cross-built on x86
  - FRC Specific GCC ARM compiler
- WPIlib provides hardware Interface
  - ROS now a recognized framework!

NATIONAL INSTRUMENTS

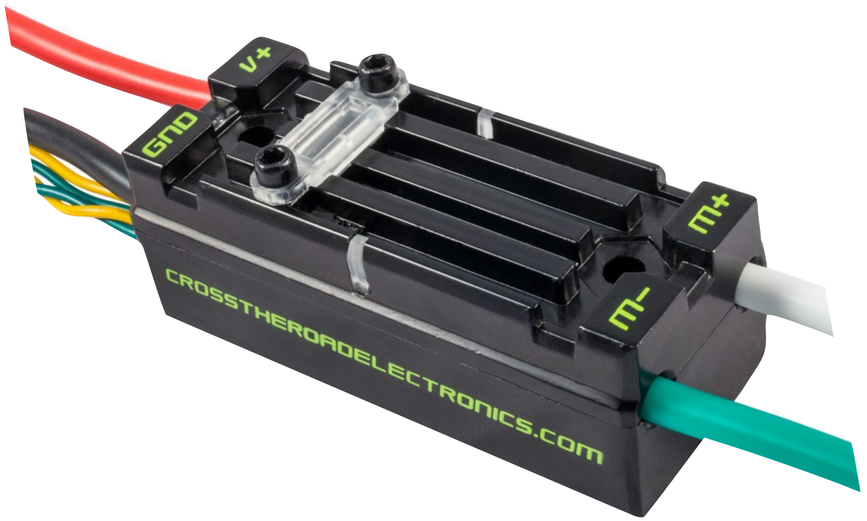
ROS Configuration



Motor Controllers

Cross the Road Electronics - Talon SRX:

- CAN bus motor controller
- 60A, 6-28V
- Highly configurable software

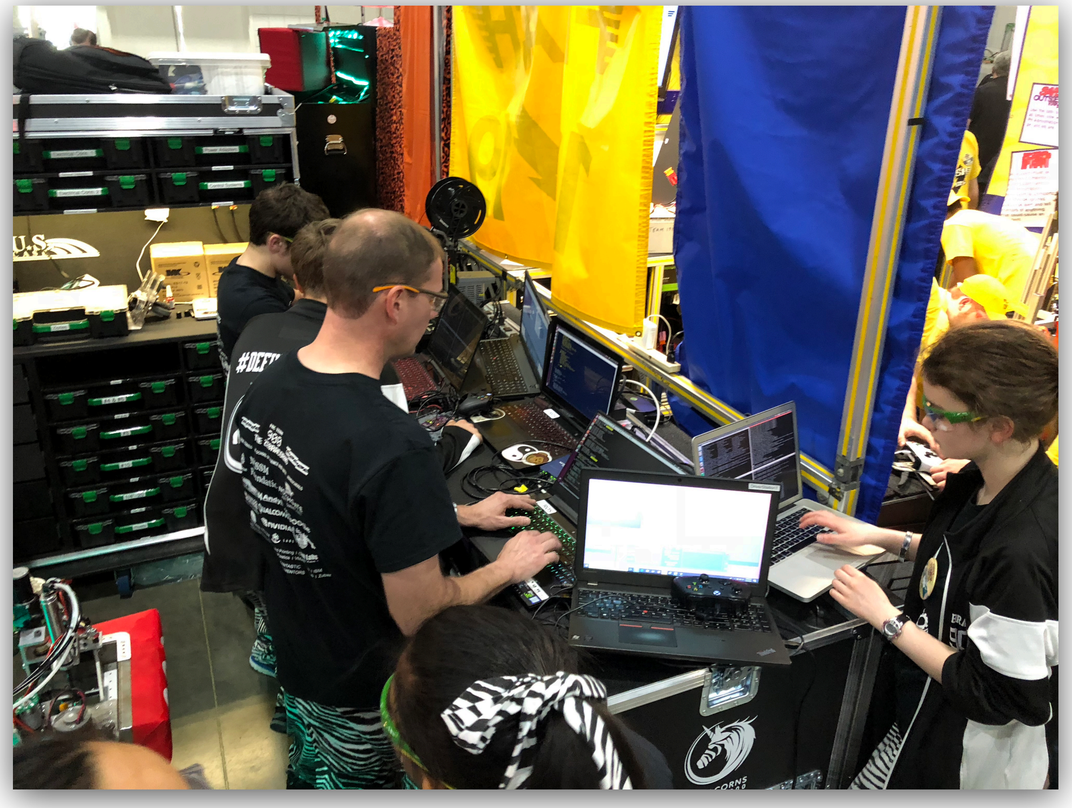


Features:

- PIDF 1KHz rate
  - Position, velocity, & current
- Motion profiling modes
- Encoder/Limit Switch Inputs
- Ramp rates
- Offload control from roboRIO

CTR ELECTRONICS

Our Custom Controller Packages



`talon_interface`

- Array of objects for talon state
- Array of objects for talon command

`talon_state_controller`

- Publishes state data
- Talon hardware → software controllers

`talon_controllers`

- One base talon command controller
- Child classes provide more functionality for different modes
- Designed as a drop-in replacement for standard ROS controllers

Other Controllers

Mechanism controllers:

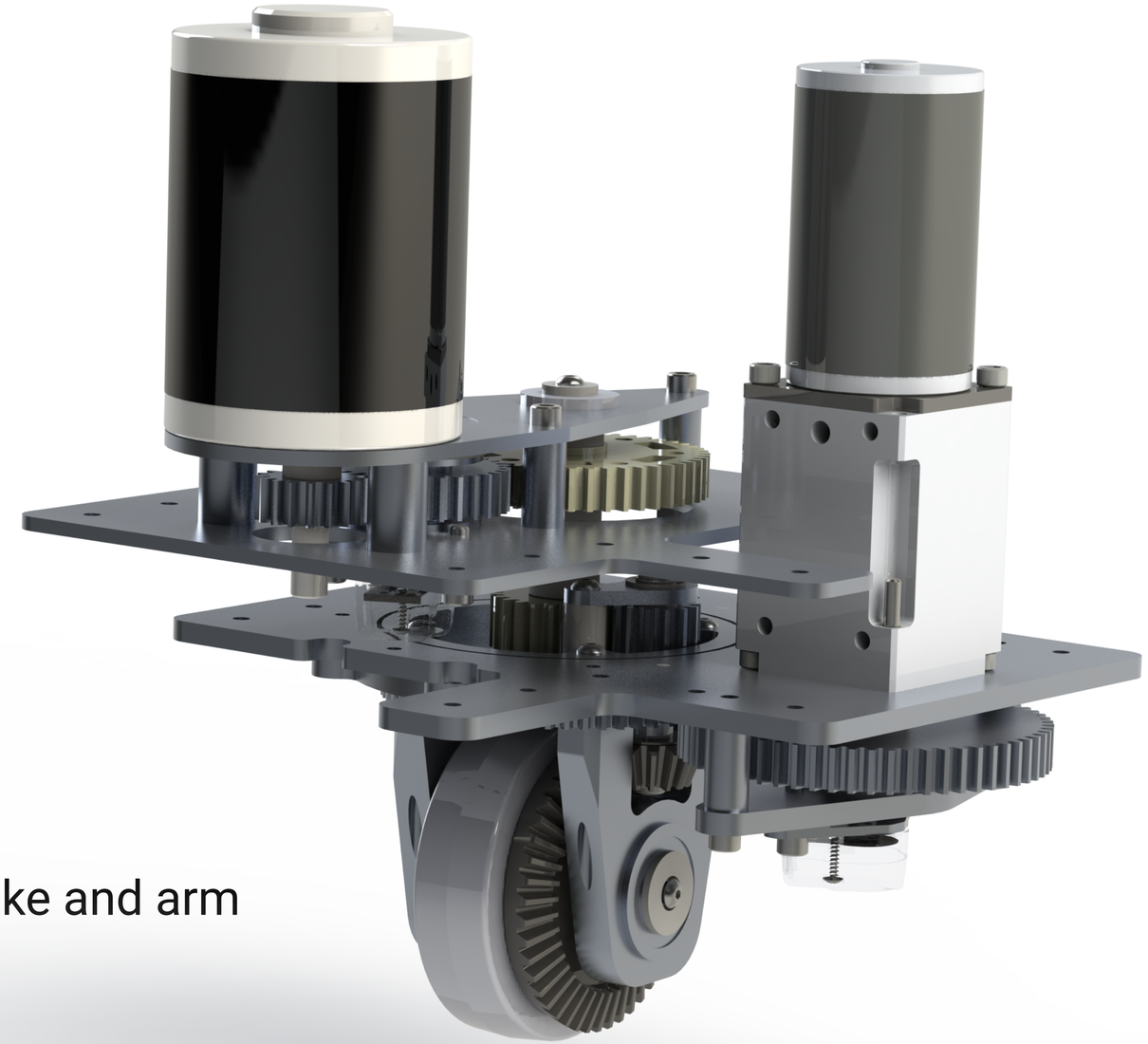
- Integrates multiple talon controllers

Swerve drive controller:

- Motion profiling
- High maneuverability (3 DOF)

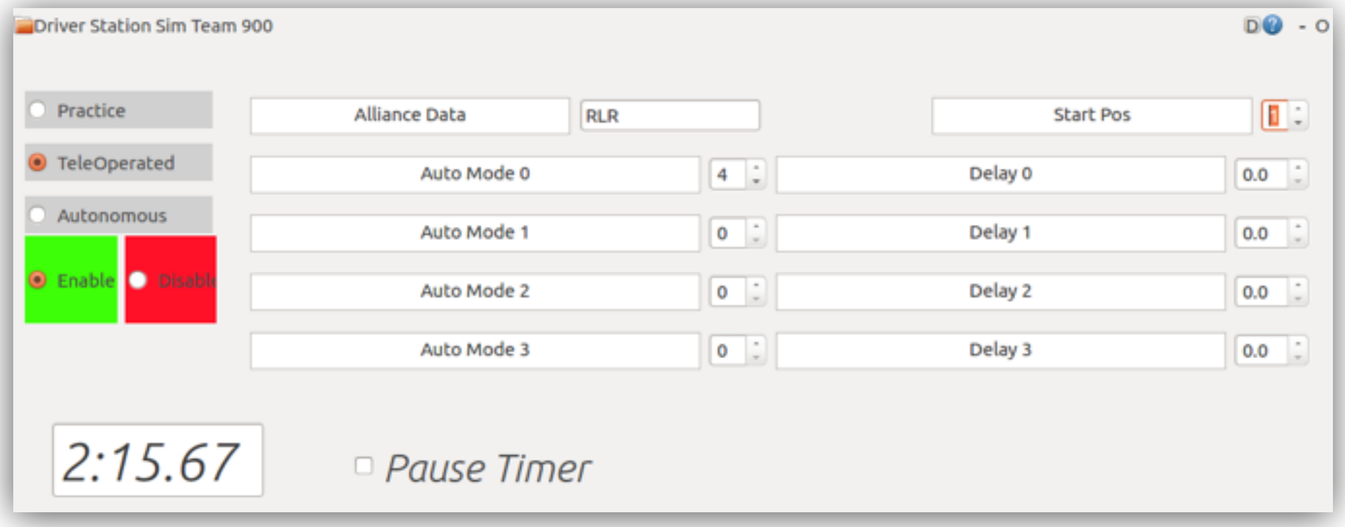
Elevator controller:

- Arm limiting
- Managed intaking and lifting cubes
- Automated transfer of cube between intake and arm

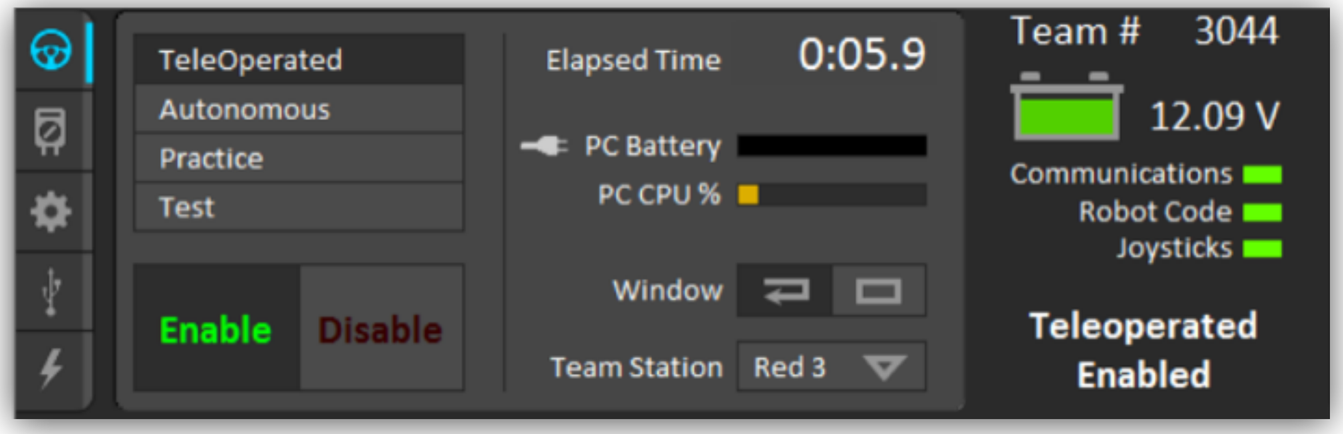


Custom Simulation Interface

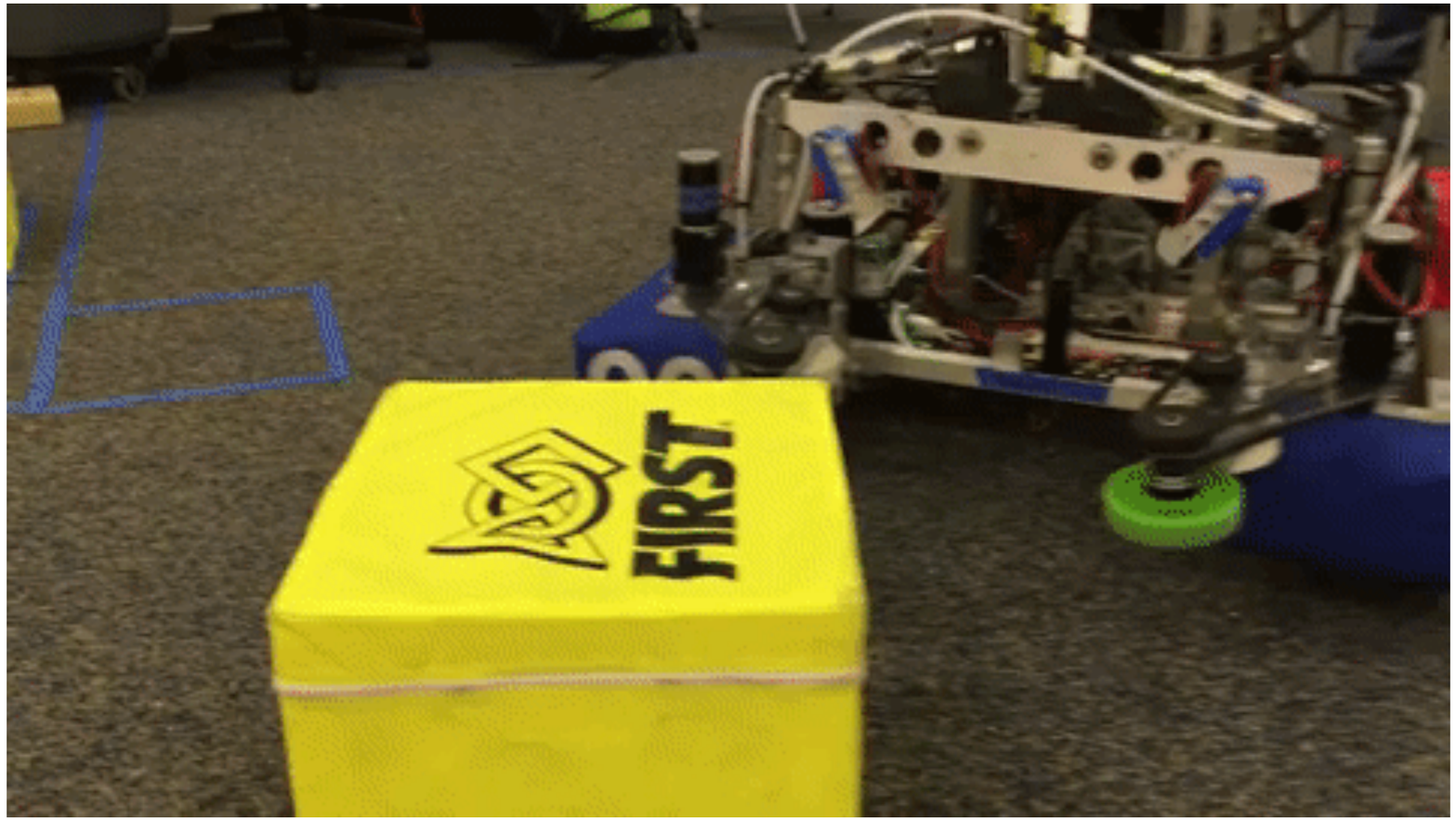
- Run code without a physical robot
- Custom plugin for FRC driver station
- Testing autonomous modes
  - Elevator and arm path verification



Custom Driver Station, Powered by ROS



FRC Driver Station, Powered by NI LabVIEW



Success!

Automated handoff with elevator controller
http://zebracorn.link/rosgif

Plans for the Future

ROS for Mainstream FRC

Robot Localization

Improve Simulation

Fully Autonomous Robot

Get Involved!

Involve your company!
Find a local team or event:
http://firstinspires.org

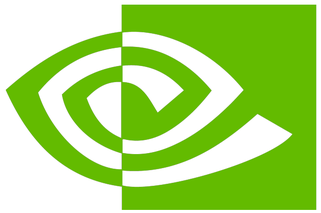
Help us!
Get in touch with us:
mentors@team900.org
or contribute:
http://github.com/frc900

Involve your University!
Tell us about
your programs

Become a sponsor!
We're a non-profit

Thanks to our sponsors!

NCSSM Foundation



NVIDIA

AndyMark®

SOLIDWORKS

Google

QUALCOMM®

YDLIDAR

CTR ELECTRONICS

vmware®

US CASE US MADE

Kauai Labs
Build Better Robots ®

ZEBRACORNS
TEAM900.ORG