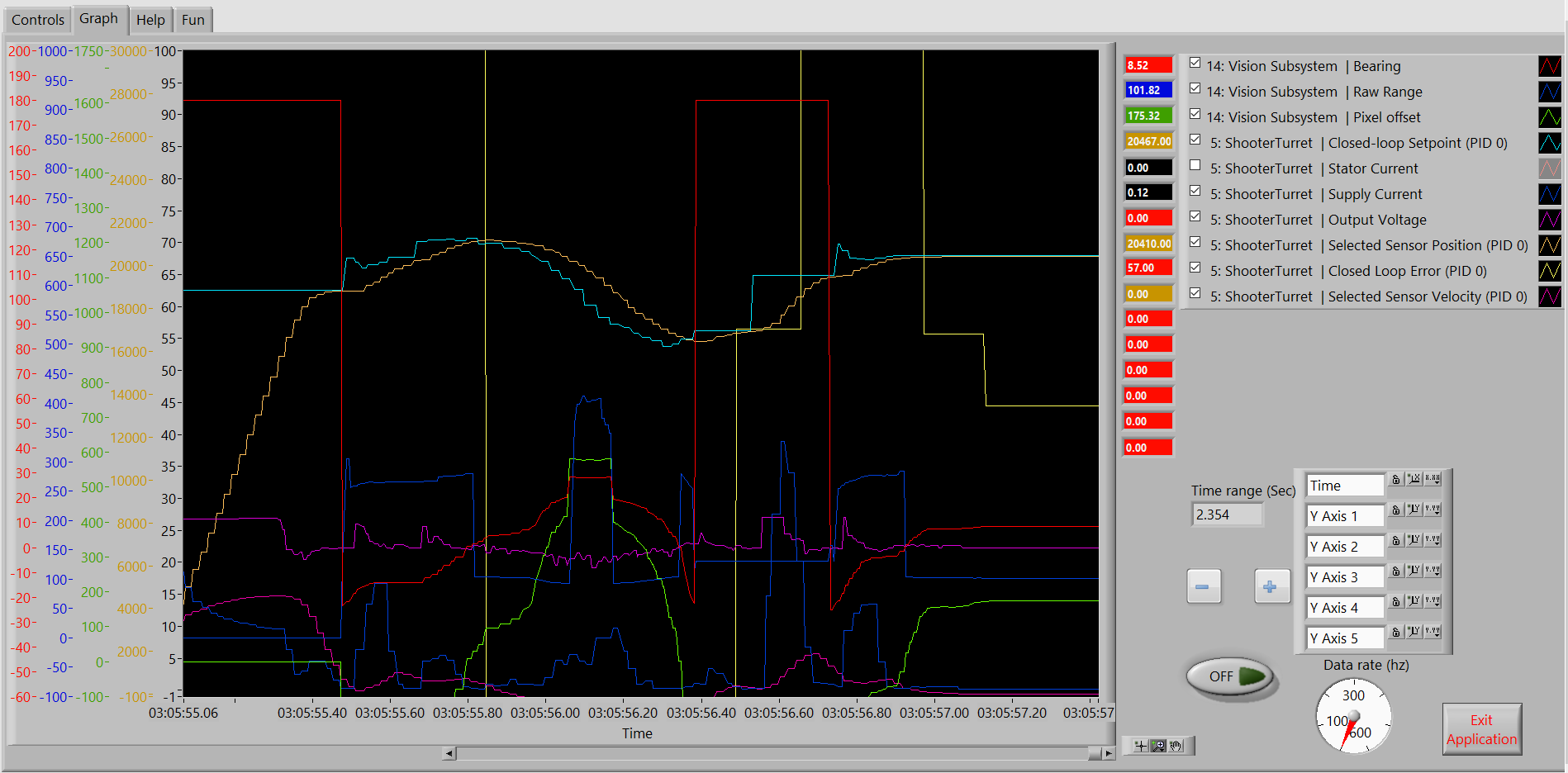
**Custom Development Tools**

The software team also builds and maintains custom development tools. These tools are used by the entire team to test and tune drive systems, actuators and sensors. The Stryke Force Grapher application allows the team to log and analyze the robot performance and perfect the tuning by ploing data received from the RoboRIO. Stryke Force is able to chart almost any data possible from the RoboRIO.

Third Coast Telemetry (TCT) and the Grapher applications provide Stryke Force with deep insight into robot performance and are invaluable to the development process.



We are pleased to make TCT and the Grapher available as open source at https://github.com/strykeforce. More resources are available at hps://www.strykeforce.org/resources



**Team 2767 Stryke Force**

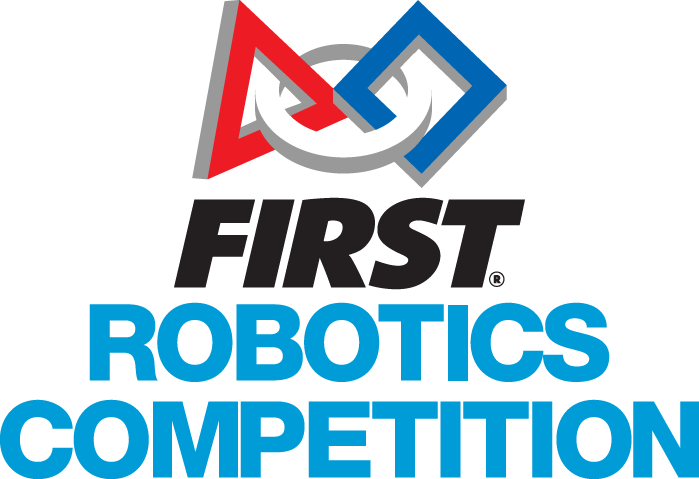
Est. 2009

Kalamazoo, MI

****

**Software**

**2022**



### Software Designed for Drivers

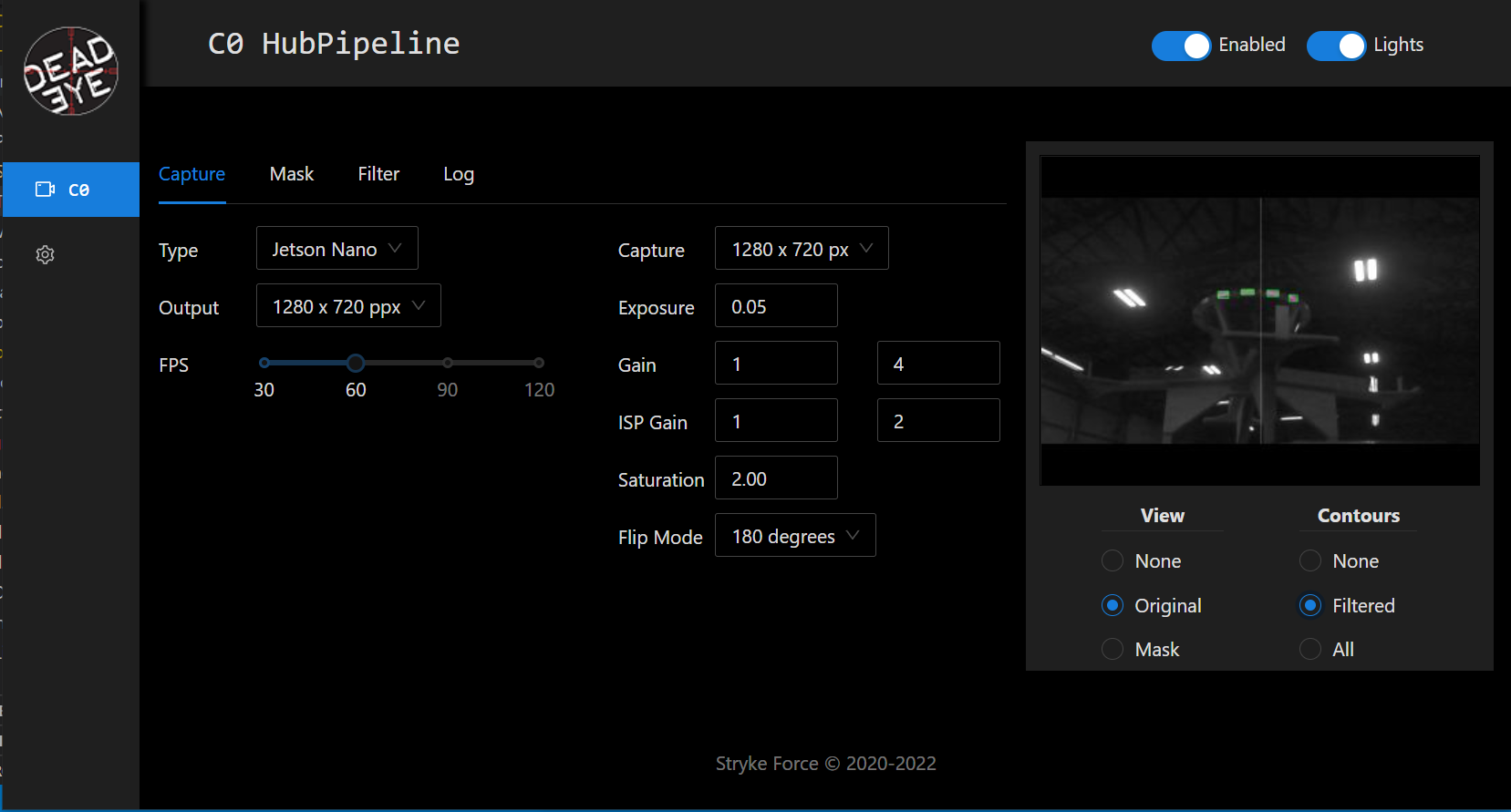
Stryke Force is proud to present the control system software for NAME IS NOT CHOOSEN—our robot competing in FIRST® RAPID REACT presented by The Boeing Company. We strive to meld high-performance hardware with custom software to provide our drive team with the best robot software possible. Our Third Coast Swerve Drive has historically provided unmatched maneuverability and response. Our highly optimized parallel movements enable us to deliver game pieces faster than ever before.

### Precision Control Systems

Stryke Force delved deep into motion profiling this year. Using the CTR-Electronics Talon SRX Motion Magic functionality, the programming team controls the motion profile using acceleration, velocity, current limit, and PID parameters. This feature in the software allows the robot to make precise and repeatable movements in all subsystems.

#### Vision Targeting

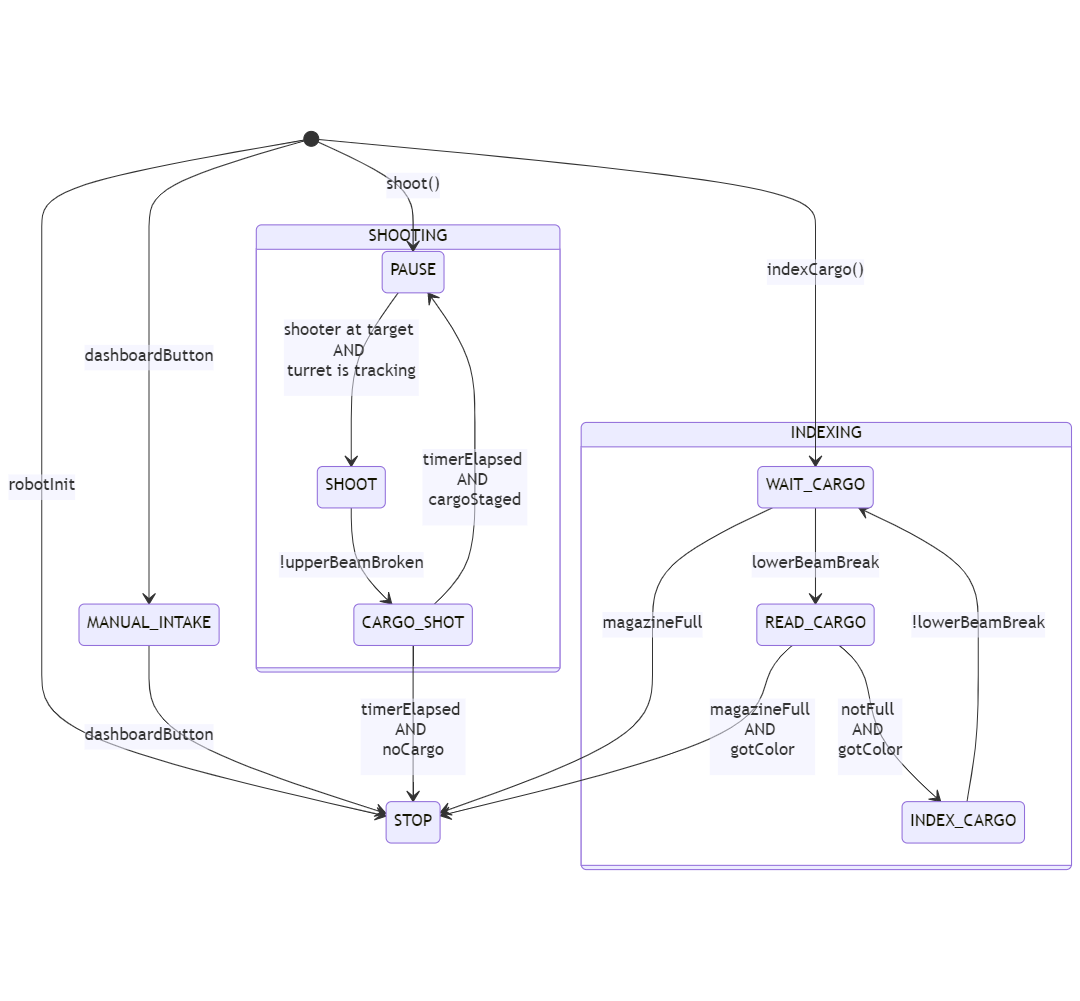
The software team placed a heavy focus on helping our drivers score cargo into the high goal. To guarantee we make the best shots possible, we’re implementing computer vision (CV) to guide our shooting system automatically. Since vision processing significantly affects CPU performance, we decided to use a Jetson co-processor. Our C++ CV application based on our custom Deadeye framework uses OpenCV to gather target data from the camera, which we then publish to NetworkTables. Deadeye allows us, as developers, to treat the vision system as a subsystem in the RoboRIO code.



When the driver presses the button to initiate the shooting process, the robot locks itself in position and aims the turret towards the target. It then uses the target data and its known gyro value to look-up the appropriate setpoints for shooter inclination and launcher speed. The programming team implemented a mathematical correction algorithm in order to compensate for the effects of fore-shortening on the target.

**State Machines**

Stryke Force has decided to implement state machines as a way to further streamline robot code. In doing this subsystems can talk about what they are doing in a much easier way. This communication allows for state specific commands that handle most if not all repeating logic. Furthermore, this strategy allows for easier transition between the complicated code necessary for each state and its duties.



The four subsystems that implemented this new strategy were the Shooter, Magazine, Climb and Turret subsystems. Before and throughout development we utilized easy to read flow charts as seen above as a structure to follow for our code. In short, this strategy of structuring code allowed us to make our code easier to read and more friendly to the human brain.