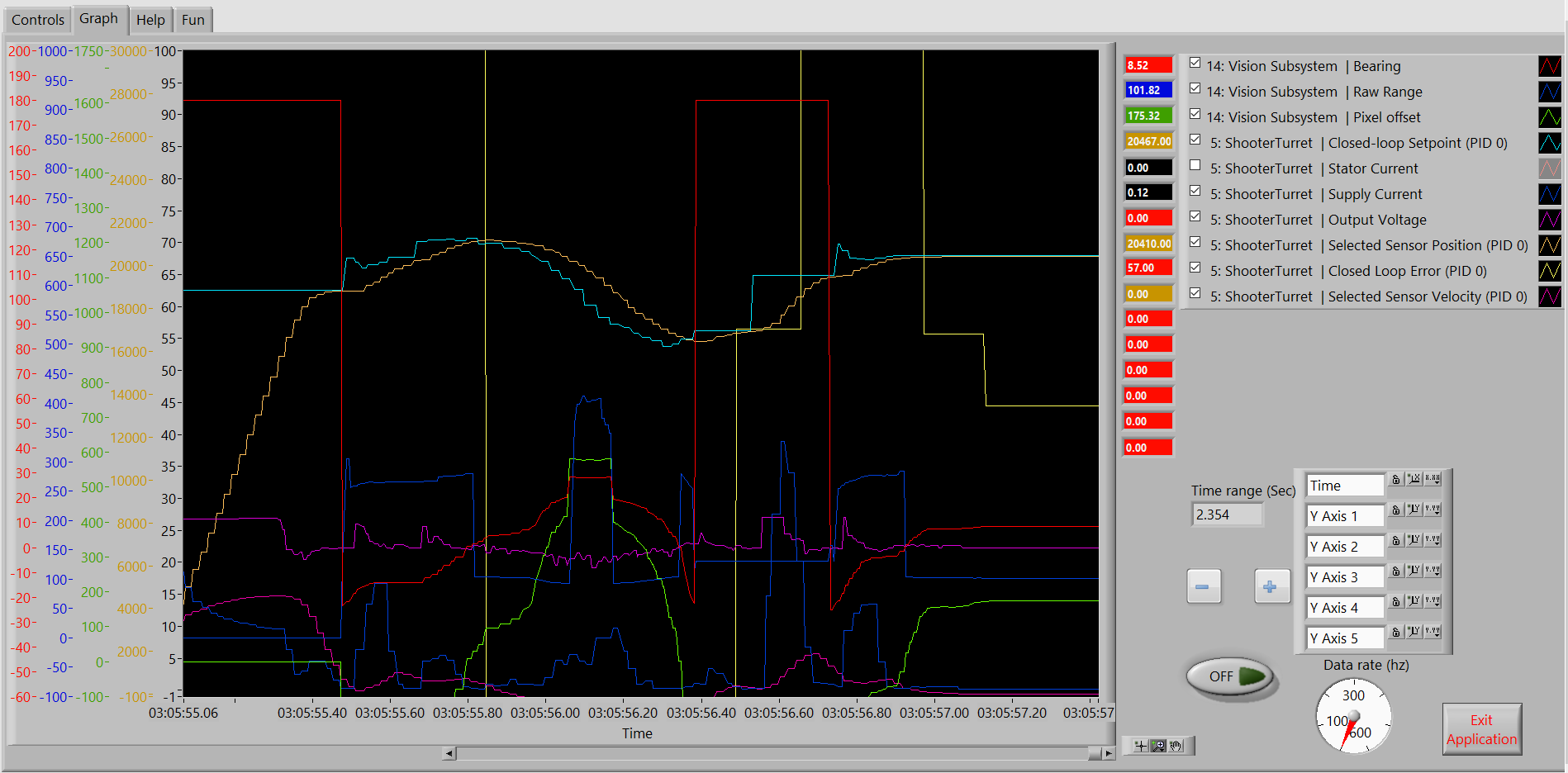
**Custom Development Tools**

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 plotting data received from the RoboRIO. Stryke Force is able to chart almost any data possible from the RoboRIO.

**Contact us**:

Facebook: 2767 Stryke Force-FIRST Robotics

Twitter: @2767StrykeForce

Website: http://strykeforce.org/

Instagram: @2767StrykeForce

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 https://www.strykeforce.org/resources

**Team 2767 Stryke Force**

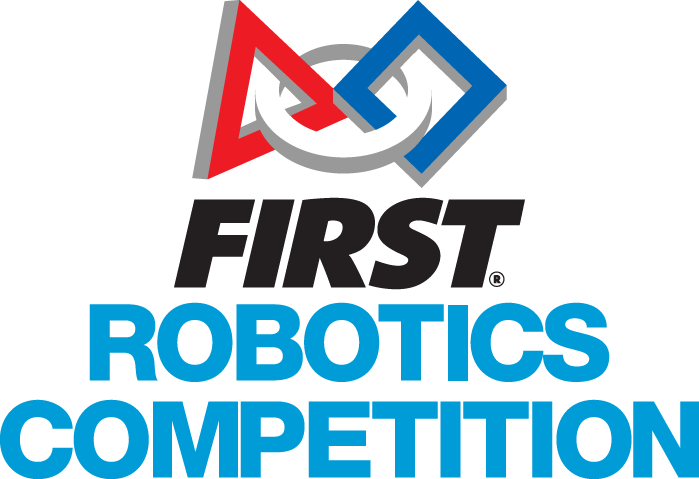
Est. 2009

Kalamazoo, MI

****

**Software**

**2023**



### Software Designed for Drivers

Stryke Force is proud to present the control system software for Snake Eyes — our robot competing in FIRST® Charged Up presented by Haas. 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. With the addition of April tags this season we are generating trajectories on-the-fly and automatically acquiring and delivering gamepieces.

### Precision Control Systems

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.

**Auton routines**

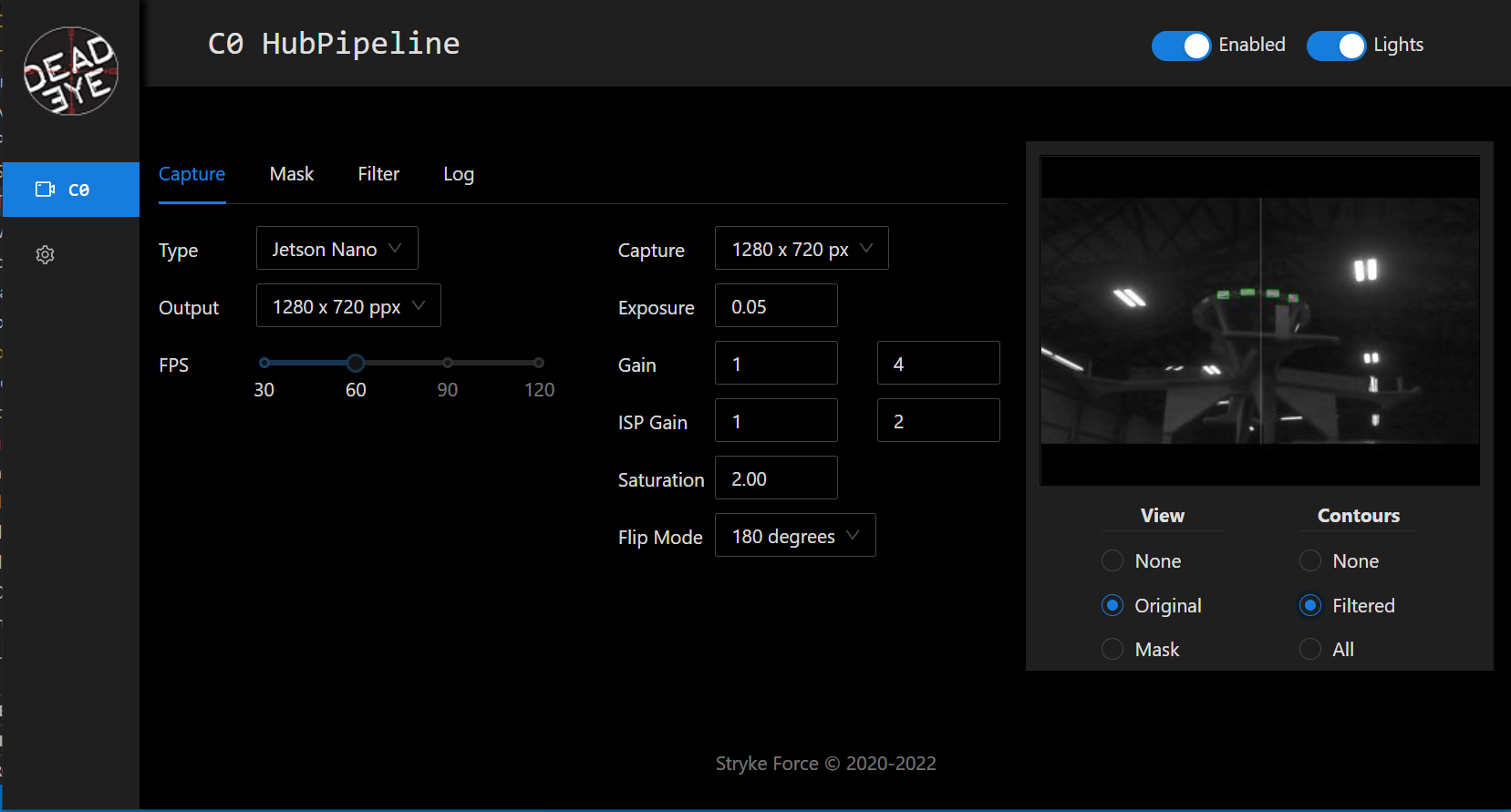
We also have a variety of autonomous routines, making for high cooperation with our teammates. We are striving to make our routines compatible with as many teams as possible.

A picture containing graphical user interface

Description automatically generated

#### Auto-place and pick-up

The software team placed a heavy focus on automatically acquiring and delivering our game pieces. To ensure maximum accuracy (because it’s embarrassing to drop pieces), we’re implementing computer vision (CV) to guide our system automatically. Since vision processing significantly affects CPU performance, we decided to use a raspberry pi 4 co-processor. The system first finds our location by having photon vision scan for April tags. Then a trajectory is generated on-the-fly and the robot drives this trajectory to the correct location. From there, all that’s necessary is to move the arm and drop the piece.



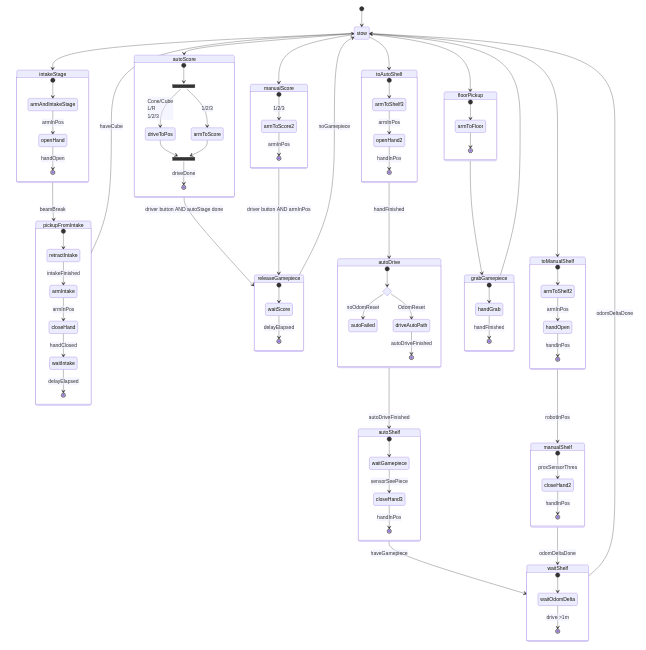
#### Indicator lights

This year, we decided to have RGB LEDs on top of the elevator for messaging. Our uses include:

* Signaling if our robot sees an April tag
* Indicating to the drive team that we successfully acquired a gamepiece from the loading station
* Telling the human player what piece we want at the substation
* General troubleshooting indicators

**State Machines**

Stryke Force has decided to implement hierarchical state machines as a way to further streamline robot code. This allows easier communication between subsystems about current actions. This also removes duplicated logic between commands and instead houses this state logic in the subsystem itself. Furthermore, this strategy allows for easier transition between the complicated code necessary for each state and its duties.



The two main subsystems that implemented this strategy are the robotState and arm 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.