Control Award Sponsored by Arm Submission Form

Team # 8696	**Team Name: Trobotix**

Autonomous objectives:

Score a pre-loaded cone on the nearest high junction past the signal sleeve.
Obtain a second cone from the cone stack and score on the nearest low junction.
Park in the correct zone based upon the signal sleeve.

Sensors used:

Drive Encoders x4 (built-in)	Accurate driving and navigation.
Magnetic Limits Switches x2	Consistent linear slide positioning (Autonomous).
Slide Encoder (built-in)	Specific linear slide positions inaccessible by magnetic limit switches.
REV Color Sensor V3	Read the signal cone.
Inertial Measurement Unit (IMU)	Keeps the robot at a specific orientation (Autonomous).

Key algorithms:

Finite-State Machine	Control linear slide positioning via. magnetic limit switches without compromising the availability of other mechanisms.
Controller Input ROC	Squared direct controller input to create a linear, instead of constant, rate of change for driving. Provides a more “natural” feel.
PID Controller	Combines both motor encoder counts and the orientation measured by the IMU to nsure the robot moves as consistently as possible. This PID controller is custom made and tuned.

Attempted algorithms:

Motion Profiling	Smoothens velocity as it changes. Incompatible with current set up.

Driver controlled enhancements:

Automatic linear slide positioning using encoders.
Convenient switching between automatic and manual slide positioning.
Improved controller sensitivity.

Driver interfaces:

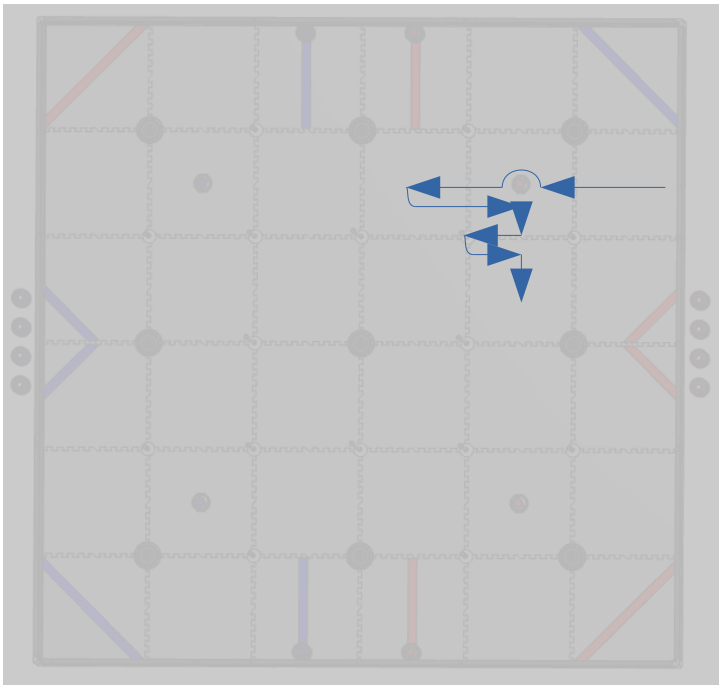
Driver Configurable Autonomous	Text-based user interface displayed in telemetry. Allows drivers to configure autonomous behavior, such as which side of the field the robot is on.
Autonomous Scripting	Custom scripting language designed to be user-friendly. Turns the autonomous structure from code into a readable form. Executed by an “interpreter” written in Java. Especially useful for members outside of programming to calibrate the autonomous.

Engineering portfolio references:

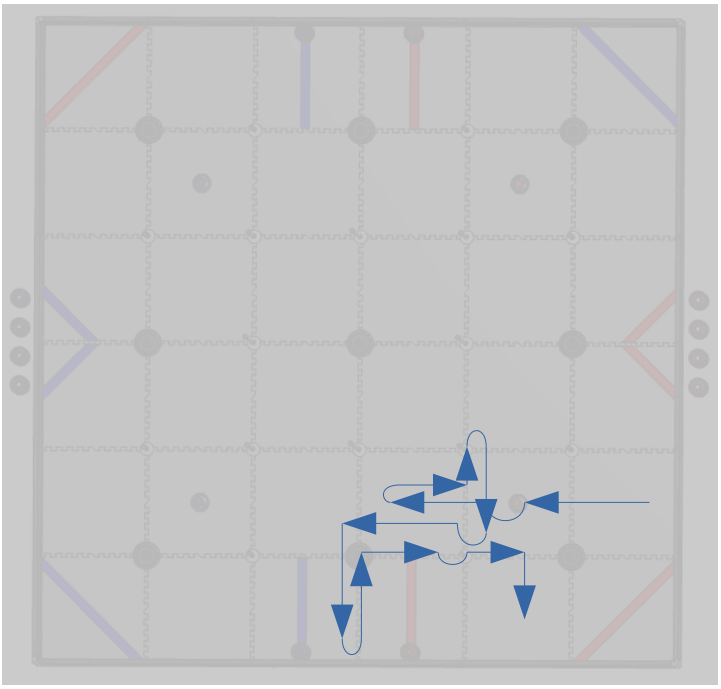
Design § 3.5.a	Sensors
Programming § 4.1	TeleOp
Programming § 4.2	Autonomous

Autonomous program diagrams:

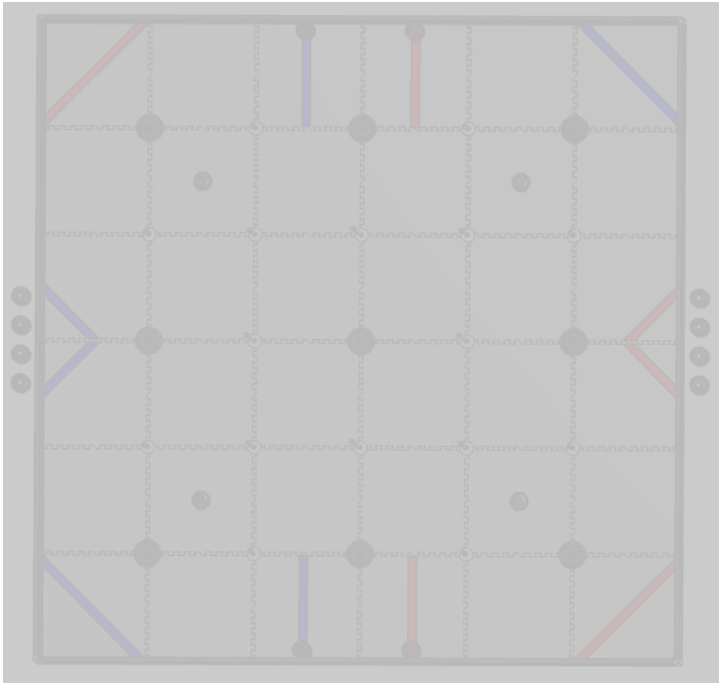
League Meet Sample Autonomous



Super Qualifier Sample Autonomous



State Sample Autonomous



In Progress