Control Award Content Sheet

Team #6022	Team Name: TBD - To Be Determined	Tournament: World Championship

Autonomous Objectives:

Route 1: Shoot Two Particles, Press Two Beacons, then Either Knock Cap Ball or Park on Corner Vortex
Route 2: Shoot Two Particles, then Either Knock Cap Ball or Park on Corner Vortex, then Either Knock Cap Ball or Park on Corner Vortex
Route 3: Shoot Two Particles, Spin Center Vortex by Driving into Cap Ball, Press Two Beacons,
Route 4: Start Sideways and Sweep a Third Particle from Partner, Shoot Three Particles, then Either Knock Cap Ball or Park on Corner Vortex

Sensors Used

Sensor	Purpose
Encoders	Measures distance driven in short-length applications, especially when there aren't fixed reference points nearby.
Gyroscope	Measures the robot heading for accurate turning and maintaining heading while driving straight.
Color Sensors	Returns the color of each side of the beacons.
ODS (Field-Facing)	Detects the field surface, especially white lines and field tiles.
ODS (Sweeper-Facing)	Determines if the sweeper is in its correct position. See "driver controlled enhancements" for details.
Range Sensor	Measures the distance from the front of the robot to the field perimeter wall or other robots.

Key Algorithms:

Algorithm	Purpose
Drop-Down Autonomous Customization Menu	To be compatible with partners, our robot controller prompts the user for settings during initialization. This object-oriented menu allows for customization of alliance color, delays, routes, and starting positions.
DC Motor Acceleration Control	Due to our high-speed drivetrain, instantaneously changing motor power would lead to voltage drop and jerky motion. To remedy this, we have a closed-loop where motors are set to a target power and then is gradually ramped up or down to its target power over time. See "drive controlled enhancements" for more.
Custom PID Loop	Our custom PID loop is implementable in multiple contexts, such as turning, driving straight, and shooter speed. We use tuned presets that we created using the Zeigler-Nichols method.
Encoder Wrapping	After noticing limitations in the native FTC SDK, we implemented our own encoder handling into our drivetrain wrapper. Functions include eliminating sporadic values through verifying three consecutive readings, reducing reset latency through virtual zeroing, and averaging magnitudes to compensate for PID correction.
Gyroscope Wrapping	To add further functionality to the gyroscope, this wrapper allows for virtual zeroing, automatic calibration upon initialization, and switching which direction is positive and negative.
Dual Beacon Color Sensors	Two color sensors are wrapped in a single beacon actuator bar object that can use comparative RGB values to determine the color of each side and therefore determine if the beacon is solid red, solid blue, or randomized.
ODS Adaptive Threshold	To adapt to different lighting environment, thresholds for detecting field surfaces are computed through adding constant values to the initial measurement on the foam tile.

Driver Controlled Enhancements:

Enhancement	Purpose
DC Motor Acceleration Control	Gives the user more control in the teleoperated period by remedying slipping wheels and jerky motion through a trapezoidal acceleration curve. See "key algorithms" section for technical details.
Shooter PID with RPM Telemetry Display	To prevent curveballs, the joystick sets a target speed that is achieved through a PID loop. The current RPM is also calculated from encoders and displayed via telemetry for the drive coach to analyze. The telemetry display also includes a counter that increments every time we attempt a shot to make match analysis easier.
Shooting Only at Target RPM	To eliminate the risk of shooting a dud particle due to motors that haven't yet reached full speed, the above-mentioned RPM calculation is used to allow the driver to shoot only when the target RPM is reached.
Sweeper Self-Alignment	Given that our sweeper can interfere with our beacon bar if angled outward, there is a bristle with reflective tape that is read via an ODS so that our sweeper can be aligned to a vertical position when not being used to sweep. Can be toggled on and off to prevent accidentally sweeping opponent particles or pushing our particles away.

Engineering Notebook References:

Software Section – Pages 9, 16, 25, 30, 32, 34, 36, 37, 48, and 56

Autonomous Program Diagrams:

For reference, the yellow arrows represent a base driving route.

Note that particle shooting route can start in a variety of locations.

Also note that there are two options for the end of the autonomous run. Our most frequently used is knocking the cap ball and parking (green route). We can also partially park on the corner vortex (orange route).

In addition, when running the particle shooting route, we can start sideways and sweep up a third particle placed to the side of our alliance partner, allowing us to shoot three particles (magenta route).

These routes may vary slightly due to other parameters available on our robot controller phone. Other customization options include setting a delay and choosing the appropriate alliance color.