Control Award Submission Form

****Please turn in this sheet during your judge interview along with your engineering portfolio****

Team # 3543	Team Name: Titan Robotics Club

Autonomous objectives:

Our Autonomous program scores up to **50 pts**: (scoring purple and yellow pixels according to Team Prop position and parking at the backstage) or **55 pts**: (in addition to the 50-pts auto, we pick up an extra white pixel from the stack and score it at the backdrop).

Before the match has even started, we track the Team Prop (Color Blob) so that we can determine which Spike Mark to score our purple pixel and to score the yellow pixel at the appropriate slot on the backdrop. To ensure a reliable and accurate autonomous, we detect the AprilTags on the backdrop to re-localize our robot's absolute location on the field. We also use 3 odometry wheels to accurately track our location and Pure Pursuit driving algorithms for smooth motion control – making it **resistant to almost all disturbances**. We also utilize one distance sensor on our pixel dropper to avoid hitting the backboard when scoring and a second sensor to detect a second pixel in our dropper. To avoid writing multiple OpModes for all autonomous permutations (red or blue alliances, starting position at audience or backstage sides, etc.), we wrote just one autonomous OpMode. With our menu system that prompted the driver to make choices on alliance color, starting position, etc.), a single autonomous OpMode can handle all combinations by examining the auto choices. In addition, all our code is written from the blue alliance view, and we call *adjustPoseByAlliance* for translating path points to the red alliance view drastically reducing the code complexity.

Sensors used:

*We use **11 sensors** to maximize robot efficiency and reach 100% reliability.*

- **Webcam (2)** – front for Team Prop/Pixels and back for AprilTag detection
- **Pixel Tray Distance Sensor (2)** – one for detecting the proper distance to the backdrop and the other detects whether we have picked up a pixel
- **Odometry Wheels (3)** – keep track of robot's absolute field position (x, y) during auto and TeleOp
- **Motor Encoders (1)** – detects position of elevator
- **Absolute Encoder (1)** – detects the absolute position of the arm
- **IMU/Gyro (1)** - measures robot heading
- **Limit Switches (1)** – set lower limit positions for elevator which is necessary since the elevator encoder is not absolute

Key Algorithms:

AprilTag – using OpenCV to detect AprilTags on the backdrop as well as on the Audience wall for re-localization of the robot's field location.
Color Blob detection – using OpenCV to detect red or blue Team Props.
Pure Pursuit – autonomous path following.
adjustPoseByAlliance - converts blue alliance path points for red alliance.
Field Oriented Driving – allows the driver to use the field as the reference frame, the robot drives forward no matter what heading it is pointing at.
AutoChoice Menus – prompts driver for choosing autonomous options so that one auto OpMode can handle all permutations of the choices.
Localization – integrating odometry to keep track of the robot's absolute field location (x, y, heading).
Stall Detection – algorithm to detect motor stalls (i.e. power has been applied to the motor but it's not moving).
Priority LED indicator – allow different subsystems to show their status using different LED color patterns without overriding each other.
Gyro-Assist Driving – due to mechanical imperfection, robot may curve when driving straight, this maintains the robot's heading using the gyro to redistribute power to the wheels.

Driver Controlled Enhancements (Auto-Assist):

Our TeleOp provides Auto-Assist features that automate picking up pixels and scoring them on the backdrop. This eliminates human errors in driving the robot into the backdrop and de-scoring pixels or auto picking pixels where the driver may not have clear view of the pixels.

Auto-Assist Pickup: *(Picking up a pixel in front)*

1. Use vision to find the closest pixel in front. Use Homography to calculate the pixel's real-world coordinate.
2. Navigate the robot to the pixel using PurePursuit Drive.
3. Turn on the intake to pick up the pixel and stop once the pixel is sensed in the pixel tray (dropper).

Auto-Assist Scoring: *(Scoring a pixel onto the selected slot at the backdrop for both autonomous and TeleOp)*

4. Use vision to find the AprilTags on the backdrop.
5. Use the AprilTag's absolute field location to calculate the robot's absolute field location and re-localize it.
6. Raise the arm and elevator to the selected scoring height.
7. Use PurePursuit Drive to approach the selected slot on the backdrop.
8. Score the pixel.

Smart Intake

The operator can put the intake into auto assist mode by pressing a button: intake will turn on and auto stops when detected a pixel in the pixel tray.

References: Pg 15 (Teleop Auto Assists), Pg 14 (Pure Pursuit + Odometry), Pg 12 (Vision)

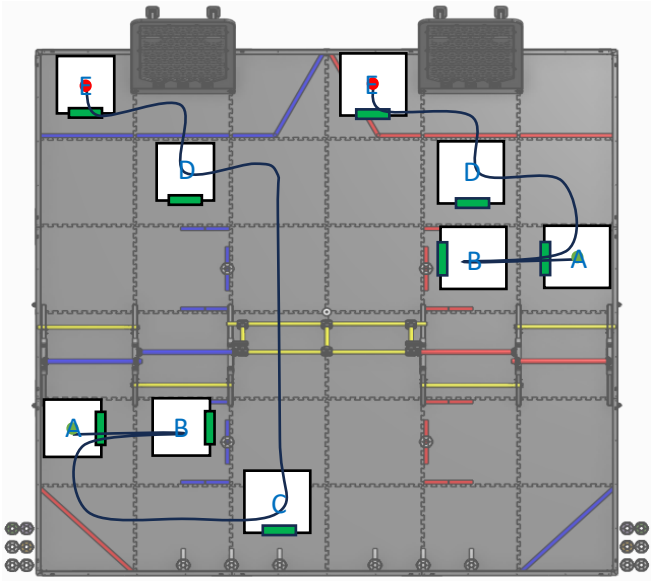
Priority LEDs Displaying Subsystem Status

During autonomous initialization, our LEDs change color depending on the Team Prop's location detected by vision (**1-violet**, **2-green**, **3-yellow**). During autonomous, the LED flashes cyan if vision detected AprilTag on the backdrop giving the driver confirmation that vision detected AprilTag. During TeleOp, different subsystems display their status with different color patterns. To avoid subsystems fighting for the LEDs, each status color pattern is assigned a priority. For example, our LEDs flash lime if Field Oriented Driving is enabled but if the camera detects an Apriltag, the LEDs will cyan instead of lime. Once we have finished signaling, the LEDs will return to lime.

Autonomous program diagrams:

Autonomous:

1. Front webcam detects randomized position based on team prop location
2. Drive to place **purple pixel** on correct spike mark
3. If we are running 2+1 (automatically audience side) stop at pixel stack to get another pixel
4. Drive to lookout position to see AprilTags
5. Score a **yellow pixel** using April tags on correct spot
    a. If we are doing 2+1 score the white first away from the yellow pixel to avoid interference
6. Park in the backdrop



Revision 1: 9/9/2023