Robotics 2023 Programming Plan

Version J1

Controller diagram labels:

- Close Claw (Automatic Detection)
- OpenClaw
- Rotate To Face Away from You
- Forward
- Get Claw / Rotate to Face Towards You
- Close Claw on Cube
- Get Cube
- Close Claw on Cone
- BACK
- START
- Left Sliding
- Right Sliding
- Backward
- Y
- X
- B — Auto Top Row Score
- A — Autoalign
- Rotate Left (Not Needed)
- Rotate Right (Not Needed)
- Press: Actuate balancing tool (only works in final mode)
- Instantly point in any direction
- Press: Move arm slightly down and grab object (Useful for moving game piece from one row to another)

Scan QR code to view most recent version online or visit tinyurl.com/5535-2023-plan

Field diagram labels:

- (596.51)
- (570.32)
- April Tag ID#5
- April Tag ID#4
- (26.19)
- (91.55)
- April Tag ID#6
- April Tag ID#3
- (66.00)
- (66.00)
- April Tag ID#2
- April Tag ID#7
- April Tag ID#8
- April Tag ID#1
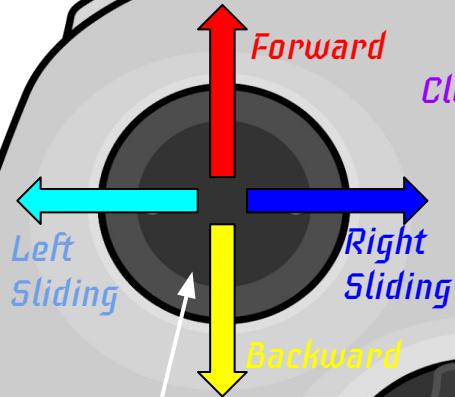- CHARGED UP
- CHARGED UP
- 55 35
- +Y
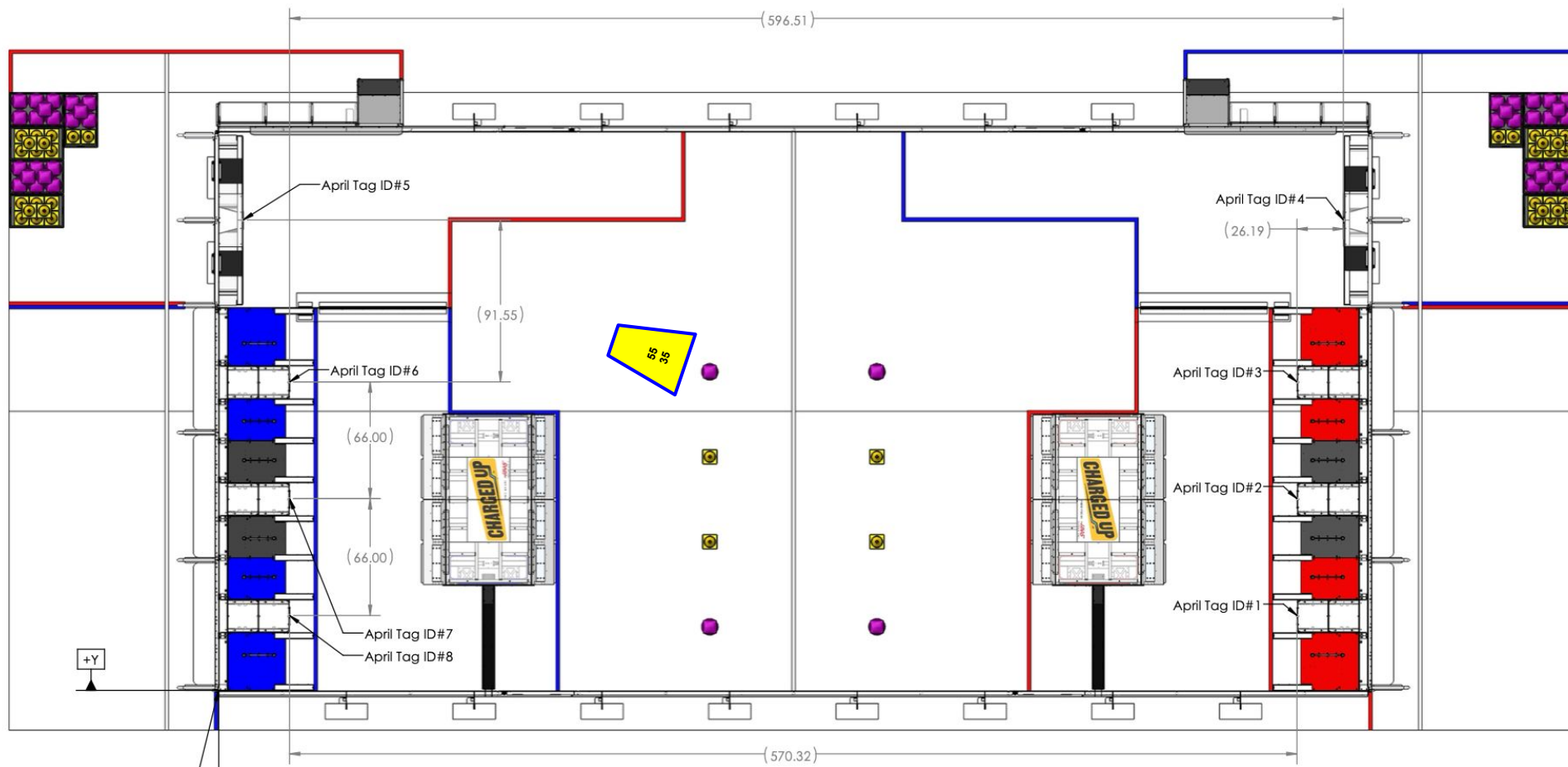
Close Claw (Automatic Detection)

OpenClaw

Rotate To Face Away from You

Rotate to Face Towards You

Get Cone

Forward

Close Claw on Cube

Close Claw on Cone
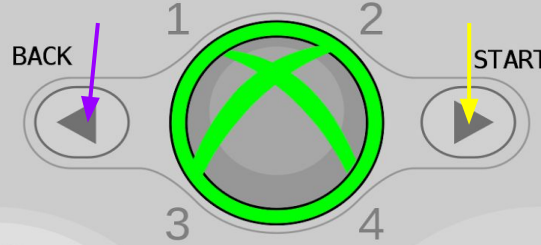
Get Cube

1   2

BACK

START

Y

Left Sliding

Right Sliding

X

B

3   4

Auto Top Row Score

Backward

Autoalign

Rotate Left (Not Needed)

Rotate Right (Not Needed)

Press: Actuate balancing tool (only works in final mode)

Instantly point in any direction

Press: Move arm slightly down and grab object (Useful for moving game piece from one row to another)

PRIMARY CONTROLLER

(Smart Mode)

Rotate robot back on track left

Rotate robot back on track right

Enable Final Mode

Disable Final Mode

Forward Movement Minor Adjustment

Disable Arm

Enable Arm

Cancel Smart Action

Reset NavX

BACK

1   2

START

Y

X

B

Left Sliding Minor Adjustment

Right Sliding Minor Adjustment

3   4

Disable NavX Features

Backward Movement Minor Adjustment

Arm to Top Row

Signal Cone

Arm to Middle Row

Peg In

Peg Out

Arm to Normal

A

Enable NavX Features

Signal Cube

Arm to Bottom Row

SECONDARY CONTROLLER

THINGS THAT COULD GO WRONG AND HOW TO SOLVE THEM

| Problem	Solution
Wheel on robot is not pointed in the right direction.	The person with the secondary controller can also use the SmartDashboard on the computer. They will see values that say "A offset", "B offset", and so on. These values can be edited to adjust each of the wheels' angles on the robot. "A" is the front-left wheel, and the rest of the wheels continue clockwise around the robot from "A". Later on, replace the CANCoder and/or the rotation magnet that pairs with it, or you will see this issue again. Set the new CANCoder to absolute position with the Phoenix Tuner, restart the robot, find the new rotational offset for this wheel, and then enter it into the actual robot programming where Weswerve is added to robot.java.
Wheel on the robot is spinning in reverse.	Add 180 to the reversed wheel's offset value. See the solution above for more information on adjusting a wheel's offset value.
Robot goes crazy, runs away, or does something similar.	Whether intentionally or unintentionally, a smart feature may have been activated. Cancel a smart action by pressing Y on the secondary controller.
Robot is spinning very fast in one direction.	The NavX came unplugged or stopped working. You will have to work without the NavX. Press B on the secondary controller to disable all navx features. Notify the driver that they are now in car mode, not headless mode.
Robot gets off course or points slowly in the wrong direction.	This is the most likely issue to occur. The NavX yaw value is slowly drifting. The person with the secondary controller should use the triggers on their controller to get the robot back on course. (Left = Rotate counterclockwise; Right = Rotate Clockwise)
Robot is skippy because battery is very low.	Drive the robot much slower. Do not accelerate or decelerate quickly. Press BACK on the secondary controller to disable any arm movement or lifting power. Also try disabling all navx features by pressing B on the secondary controller. Use your remaining power to get to the charging station.

SIMPLIFIED AUTONOMI SYSTEM

Please note that this is a very basic autonomi system that we will not be using very long.

Consistent order of actions:
- Robot places preloaded cube on cube node
- Robot backs up while rotating 180 degrees
- Robot collects game piece closest to it
- Robot backs up while rotating 180 degrees
- Robot places game piece on top row
- If option selected in SmartDashboard, robot assumes that it started in front of the cooperetition grid, backs up onto the grid, and automatically balances.

Total Time ≈ 14 seconds
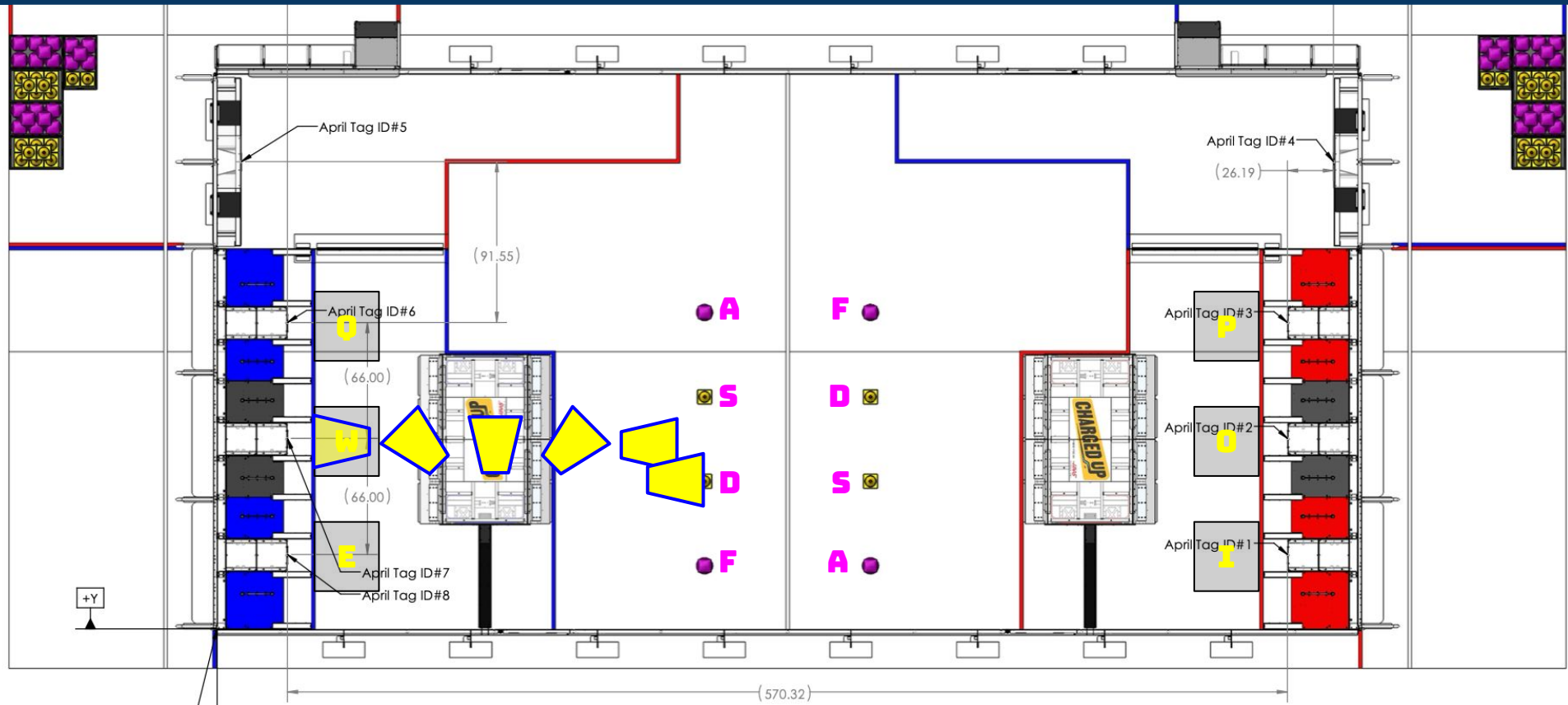
GENERAL STRATEGY

Award	Awarded for...	AUTO	TELEOP
MOBILITY	each ROBOT whose BUMPERS have completely left its COMMUNITY at any point during AUTO	3	
GAME PIECES	scored on a bottom ROW	3	2
	scored on a middle ROW	4	3
	scored on a top ROW	6	5
LINK	3 adjacent NODES in a ROW contain scored GAME PIECES.		5
DOCKED and not ENGAGED	Each ROBOT (1 ROBOT max in AUTO)	8	6
DOCKED and ENGAGED	Each ROBOT (1 ROBOT max in AUTO)	12	10
PARK	Each ROBOT whose BUMPERS are completely contained within its COMMUNITY but does not meet the criteria for DOCKED.		2

ROBOT SETUP

Because the robot needs to know some things about its position when it is first turned on, the robot has been programmed to think that it is starting with the following criteria met:

- Robot is facing the grid and pressed against it.
- Robot is perfectly horizontally aligned with a cube node.
- Robot is starting at point W or O if "CHARGE UP!" is selected as "YES" in the Shuffleboard, and robot is starting at point Q, E, P, or I if "NO" is selected under "CHARGE UP!".
- Robot arm is down in starting config, with alpha at –2 degrees and beta at 14 and the cube sitting on the plexiglass base.
- Robot is squeezing cube very tightly.

TELEOP

Always go for the nearest game piece. Picking up off the ground is best for our robot. When you are in front of the game piece, simply press a button, drive to the grid, and score the game piece. It's that simple until endgame or until something goes wrong.

AUTONOMOUS

When discussing the plan with other teams, make sure that other robots will not be in our robot's path. Additionally, make sure that we don't start at point W or point O if we are not balancing at the end – Our robot will use the charging station as a pathway when starting at these points.