2018 Robot Concept of Operations / Interface Description

# Overview:

This purpose of this document is to keep the electrical, mechanical, & software teams aligned on the robot design maturation and concept of operation. This is a ***working document*** intended to stimulate discussion on how robot operates, what is the control strategy, what sensors are required, what is the logical drive team interface, and potential autonomous modes. ***If we take the time to document what the robot does & how it does it, we often flush out details that we would have been overlooked, avoiding robot integration problems***.

Appendix A: Robot Interface Control Document

Appendix B: User Interface Document

Appendix C: Autonomous Scenarios

# Team Vision & Strategy:

The team vision is to create a student experience that inspires every participant to turn pro - encouraging students to pursue exciting career opportunities in the science or engineering fields. Our robots are student inspired and student built.

***“WE EXCEL TO BE THE BEST & HAVE FUN!”***

The team’s game strategy: ***“Dominate the Scale”***

* Retrieve Cubes from Field Floor
* Deposit Cubes in the Exchange
* Score on the Scale & Switch
* Climb the Scale from the Side

# Robot Integration:

The FIRST robotics program scope is designed to be too large for any one team to accomplish by itself & requires teams to prioritize what is important to their strategy & fits within the time constraints. The SHS Robotics team will adopt an incremental capability mindset (especially with software). ***Ensure we have basic functionality first - add more complexity in stages.*** In the event we run out of time, we will always have some capability the students can use in the competitions.

* ***Driving the robot using the drive controller?***
* ***Demonstrate motor actuation & pneumatic actuation using the operator controller?***
* ***Display camera image on driver’s station?***
* ***Motion plan a navigation segment(s) & execute an autonomous task sequence?***

The team strategy to build two identical robots allows us additional integration & test time between ship day & the first competition to optimize robot performance and mature drive team proficiency. The extra robot gives us a platform to test & evaluate design modifications we make after the first competition to enhance performance for the team’s second competition. During the integration process, the team must be attentive to using configuration management discipline through software development & integration. This will help mitigate “merge” issues during development/integration and ensure we can retreat to a known working condition with known limitations but still play the game. ***Start debugging or verifying functional capability by constraining maximum motor speed (<50%)***.

# Robot Functions:

The major robot capabilities include drive, cube acquisition, cube manipulation, & climbing.

***Drive:***

In the confined field of play with two alliances (six robots), robot maneuverability is always at a premium. Robot maneuverability is a function of the robot physical configuration (wheelbase & track width, drive configuration, & wheel type), drive control algorithm, and driver prowess. The drive control strategy for the robot will operate in two distinct modes: differential steering for normal field maneuvers and spinning in-place for maneuvering in confined quarters. Adjusting the differential speed of the two wheels will change the robot's direction of travel.

The robot’s center wheels are Skyway treaded urethane tires. The robot’s front/rear wheels are omni wheels that will minimize wheel drag through a turn. Left & right motor assemblies are mounted on the robot chassis that require rotation in opposite directions to move the robot in the forward or reverse direction. DC motors have noticeable forward / reverse maximum speed bias. The robot will not move until the commanded torque overcomes the drive train efficiencies and load drag.

The SHS robotics design approach augments the WPI drive function with a universal lookup table using the left wheel velocities as a reference. The table is tailored to our particular robot physical configuration, balances the nonlinear the turning radius range to the available control range, and prevents the inner wheels from stop rotating in a tight turn. Steering & throttle joystick input commands will be used as the lookup table indices to determine the appropriate left & right normalized wheel velocities. Right wheel velocity is sign inverted to account for opposite rotation differences due to the chassis installation (WPILib function). The normalized velocities are then converted to the appropriate CAN/PWM commands for the speed controllers.

During autonomous mode, the robot will be required to dead reckon navigate to desired position and execute a set of commands to complete a scoring task. Dead reckoning usually relies only on wheel encoders and a navigation sensor. The encoder sensors will report robot distance & direction traveled. The ADXRS-450 gyro sensor will report a relative robot heading.

|  |  |  |
| --- | --- | --- |
| ***Motor Assembly***  ***Sprocket*** | ***Wheel***  ***Velocity*** | ***Encoder***  ***Distance Traveled***  ***Coefficient*** |
| ***15t*** | ***11.5 ft/sec*** | ***0.097 in / pulse*** |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Function*** | ***Actuation*** | ***Control*** | ***Sensor*** |
| ***Drive*** | ***4 –CIM’s*** | ***4 – Talon SRX***  ***(CAN Bus)*** | ***Encoder (direction & distance) – digital input***  ***ADXRS-450 Gyro (heading) – SPI interface*** |

***Cube Acquisition:***

The robot retrieves cubes from the field or the power cube zone (three different heights). Cubes retrieved from the exchange or portal need to be dropped on the floor by the human player as the robot approaches. The drive team should minimize the time the grasper is in floor acquisition position to avoid damage from unintended robot-to-robot or field element interaction (raise the elevator or retract the grasper wrist).

In starting position, the grasper wrist is retracted to keep the appendage within the frame envelope. As the robot approaches a cube, the grasper opens, the wheels are activated, & the grasper wrist is deployed. The grasper employs driven wheels at the end of the arms to increase the effective input area decreasing the need for precise driver alignment to the cube. The rotating wheel friction with the cube will move the cube into a position where pneumatic actuation can be engaged to capture the cube for lifting. After acquisition, the grasper wrist is retracted & elevation lift is raised to the switch position for transport. The camera image is displayed on the driver’s station to aid the drivers in positioning the robot to acquire cubes.

| ***Function*** | ***Actuation*** | ***Control*** | ***Sensor*** |
| --- | --- | --- | --- |
| ***Cube Acquisition*** | ***2 – RS550 w/***  ***4:1 P60 gearbox*** | ***Talon SRX***  ***(PWM)*** | ***Contact switch turns motor OFF or utilize Talon current limit???*** |
| ***Cube Grasper*** | ***2 – Pneumatic actuators***  ***0.75” bore x 2” throw*** | ***Mead Single Solenoid*** | ***Camera image - drive team aid?***  ***Contact switch triggers grasper closure???*** |
| ***Gripper***  ***Wrist*** | ***2 – Pneumatic actuators***  ***0.75” bore x 6” throw*** | ***Mead Single Solenoid*** |  |
| ***Cube Lift*** | ***1 – 775pro w/***  ***12:1 P60 gearbox*** | ***Jaguar***  ***(PWM)*** | ***Extension & retraction (home) contact limit switches***  ***Encoder – lift position indicator*** |

***Cube Manipulation / Scoring:***

Cube manipulation uses the grasper wrist, grasper, & elevation lift. Upper (forward) & lower (reverse) elevation lift limit switches provide hardware closed loop control using the Jaguar speed controller (no software intervention) to prevent the lift driving through the mechanical stops. The lower limit switch establishes the home position for the encoder. The elevator (lift) will have preset button heights to manipulate the cube to desired positions.

| ***Lift Height***  ***Preset Position*** | ***Height*** |
| --- | --- |
| ***1*** | ***Minimum Height (Zero)***  ***Floor - Exchange Level*** |
| ***2*** | ***Cube Zone – Second Level*** |
| ***3*** | ***Cube Zone – Third Level*** |
| ***Home*** | ***Switch Deposit Level*** |
| ***4*** | ***Scale Deposit Level – Low*** |
| ***5*** | ***Scale Deposit Level – High*** |
| ***5*** | ***Maximum Height***  ***Rung Level*** |

At the exchange, the switch, & the scale, the cube release is as follows: drop the wrist, reverse the acquisition motors, & open the gripper.

At the scale, a distance sensor (IR rangefinder, ultrasonic, camera) is required to prevent the robot from driving the lift into or under the scale or rung.

| ***Function*** | ***Actuation*** | ***Control*** | ***Sensor*** |
| --- | --- | --- | --- |
| ***Cube Acquisition*** | ***2 – RS550 w/***  ***4:1 P60 gearbox*** | ***Talon SR/SRX***  ***(PWM)*** | ***Contact switch turns motor OFF???*** |
| ***Cube Grasper*** | ***2 – Pneumatic actuators***  ***0.75” bore x 2” throw*** | ***Mead Single Solenoid*** | ***Camera drive team aid?***  ***Contact switch triggers grasper closure???*** |
| ***Gripper***  ***Wrist*** | ***2 – Pneumatic actuators***  ***0.75” bore x 6” throw*** | ***Mead Single Solenoid*** |  |
| ***Cube Lift*** | ***1 – 775pro w/***  ***12:1 P60 gearbox*** | ***Jaquar***  ***(PWM)*** | ***Extension & retraction (home) contact limit switches***  ***Encoder – lift position indicator*** |

***Climb the Scale:***

During the final 30 seconds of the match, appendages can extend past the 16” frame limit. The robot will approach the rung from the side. The elevator lift grabs the hook & raises it to rung level. The robot drives forward to engage the hook on the rung. The elevator lift lowers releasing the hook & the winch is engaged to climb. The hook design extends the winch point well outside of the rung.

A ratchet & pawl on the winch shaft will prevent the motor from back driving (necessary?). The operator will disable the winch when completing the climb.

| ***Function*** | ***Actuation*** | ***Control*** | ***Sensor*** |
| --- | --- | --- | --- |
| ***Elevator Lift*** | ***1 – 775pro w/***  ***12:1 P60 gearbox*** | ***Jaquar***  ***(PWM)*** | ***Extension & retraction (home) contact limit switches*** |
| ***Deploy Hook*** | ***Pneumatic actuator***  ***0.75” bore x 4” throw*** | ***Mead Single Solenoid*** |  |
| ***Winch*** | ***1 – 775pro w/***  ***64:1 P60 gearbox*** | ***Talon SRX***  ***(PWM)*** |  |

***Pneumatics:***

The compressor control to maintain the storage pressure (115-120 psi) is now a hardware closed loop control system using the pneumatic control module (no software intervention). A pressure switch provides feedback to determine when to turn the compressor power on or off, providing efficient use of both the compressor & managing the battery load. The compressor power is supplied by the PCM. When the storage pressure is <95psi, the PCM enables the compressor 12VDC power. When the storage pressure is >115psi, the PCM disables the compressor 12VDC power. Pneumatic actuator control remains under software control via the Mead solenoids.

| ***Function*** | ***Actuation*** | ***Control*** | ***Sensor*** |
| --- | --- | --- | --- |
| ***Storage Air (Pneumatics)*** | ***Compressor*** | ***PCM*** | ***Pressure switch – PCM input*** |

***Vision Target Acquisition & Tracking:***

A camera will be mounted to the top of the robot to help the drive team acquire game objects & align the robot to the switch or scale. The camera image will always be displayed on the driver’s station display as a driver aid to acquire gears from the field.

Camera feedback can be used to align to the scale & determine the distance to the scale.

# Control Matrix:

Historically, we have linked specific robot articulations to individual game controller buttons. As the FIRST challenges grow in complexity, long robot articulation sequences are required to accomplish a specific game task. During the heat of competition, multiple step control sequences become easily prone to student mistakes. The following table plots each game task versus the specific robot articulation with the intention of identifying command sequences that could be consolidated to single game controller buttons:

| ***Game***  ***Action*** | ***Cube Grasper*** | ***Cube***  ***Wrist*** | ***Acquisition Wheels*** | ***Elevator Lift*** | ***Hook Deployment*** | ***Climbing Winch*** |
| --- | --- | --- | --- | --- | --- | --- |
| ***Starting***  ***Position*** | ***Closed*** | ***Up*** | ***Off*** | ***Switch Height?*** | ***Stow*** | ***Off*** |
| ***Acquire Cube from Field Floor*** | ***Open⇨ Closed*** | ***Down*** | ***Acquire (Forward)*** | ***Home*** | ***Stow*** | ***Off*** |
| ***Transport***  ***Gear*** | ***Closed*** | ***Up*** | ***Off*** | ***Switch Height*** | ***Stow*** | ***Off*** |
| ***Place Cube in***  ***Exchange*** | ***Closed ⇨ Open*** | ***Down*** | ***Release (Reverse)*** | ***Home*** | ***Stow*** | ***Off*** |
| ***Place Cube on***  ***Switch*** | ***Closed ⇨ Open*** | ***Down*** | ***Release (Reverse)*** | ***Switch Height*** | ***Stow*** | ***Off*** |
| ***Place Cube on***  ***Scale*** | ***Closed ⇨ Open*** | ***Down*** | ***Release (Reverse)*** | ***Scale Low or High***  ***Height*** | ***Stow*** | ***Off*** |
| ***Climb*** | ***Closed*** | ***Up*** | ***Off*** | ***Rung Height*** | ***Deploy*** | ***On*** |

| ***Control***  ***Sequence*** | ***Command String*** |
| --- | --- |
| ***Acquire***  ***Cube*** | ***Open Grasper***  ***Acquisition Wheels – On to Acquire***  ***Lower Wrist*** |
| ***Transport***  ***Cube*** | ***Close Grasper***  ***Acquisition Wheels - Off***  ***Raise Wrist*** |
| ***Release***  ***Cube*** | ***Lower Wrist***  ***Acquisition Wheels – On to Release***  ***Open Grasper***  ***Raise Wrist*** |
| ***Deploy***  ***Hook*** | ***Raise Wrist***  ***Close Grasper***  ***Raise Lift – Rung*** |
| ***Climb*** | ***Drop Lift – Switch***  ***Backup up 12 inches***  ***Start winch***  ***(Operator manually stops winch)*** |

# APPENDIX A: Robot Interface Control Document

|  |  |  |
| --- | --- | --- |
| ***RoboRIO CAN Bus Assignments*** | | |
| ***Bus***  ***Assignment*** | ***Actuation*** | ***Signal Definition*** |
| ***1*** | ***N / A*** | ***Factory Default*** |
| ***2*** | ***Left Drive #1 (CIM)*** |  |
| ***3*** | ***Left Drive #2 (CIM)*** | ***Slave Mode to Left Drive #1*** |
| ***4*** | ***Right Drive #1 (CIM)*** |  |
| ***5*** | ***Right Drive #2 (CIM)*** | ***Slave Mode to Right Drive #1*** |
| ***6*** |  |  |
| ***7*** |  |  |
| ***8*** |  |  |

|  |  |  |
| --- | --- | --- |
| ***RoboRIO – Pulse Width Modulation (PWM)*** | | |
| ***Pin***  ***Assignment*** | ***Actuation*** | ***Signal Definition*** |
| ***0*** | ***Elevator Lift*** | ***Raising speed = 90%***  ***Lowering speed = 70%*** |
| ***1*** |  |  |
| ***2*** | ***Cube Acquisition*** | ***Acquisition speed = 90%***  ***Release speed = 70%*** |
| ***3*** |  |  |
| ***4*** | ***Climbing Winch*** | ***Climbing speed = 80%*** |
| ***5*** |  |  |
| ***6*** |  |  |
| ***7*** |  |  |
| ***8*** |  |  |
| ***9*** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***RoboRIO – GPIO*** | | | |
| ***Pin***  ***Assignment*** | ***Input/***  ***Output*** | ***Signal Function*** | ***Signal Definition*** |
| ***0*** | ***Input*** | ***Left Wheel***  ***Quadrature Encoder - A*** | ***Left Encoder A*** |
| ***1*** | ***Input*** | ***Left Wheel***  ***Quadrature Encoder - B*** | ***Left Encoder B*** |
| ***2*** | ***Input*** | ***Right Wheel***  ***Quadrature Encoder – B*** | ***Right Encoder B*** |
| ***3*** | ***Input*** | ***Right Wheel***  ***Quadrature Encoder - A*** | ***Right Encoder A*** |
| ***4*** |  |  |  |
| ***5*** | ***Input*** | ***Elevation Lift***  ***Quadrature Encoder - A*** | ***Lift Encoder A*** |
| ***6*** | ***Input*** | ***Elevation Lift***  ***Quadrature Encoder - B*** | ***Lift Encoder B*** |
| ***7*** |  |  |  |
| ***8*** | ***Input*** | ***Cube Acquisition***  ***Contact Switch*** |  |
| ***9*** |  |  |  |

|  |  |  |
| --- | --- | --- |
| ***RoboRIO Analog Input*** | | |
| ***Pin***  ***Assignment*** | ***Signal Function*** | ***Signal Definition*** |
| ***0*** | ***Rangefinder***  ***(distance from lift to***  ***switch, scale & rung)*** |  |
| ***1*** |  |  |
| ***2*** |  |  |
| ***3*** |  |  |

|  |  |  |
| --- | --- | --- |
| ***RoboRIO SPI Input*** | | |
| ***Pin***  ***Assignment*** | ***Signal Function*** | ***Signal Definition*** |
|  | ***ADXRS450 Gyro*** |  |

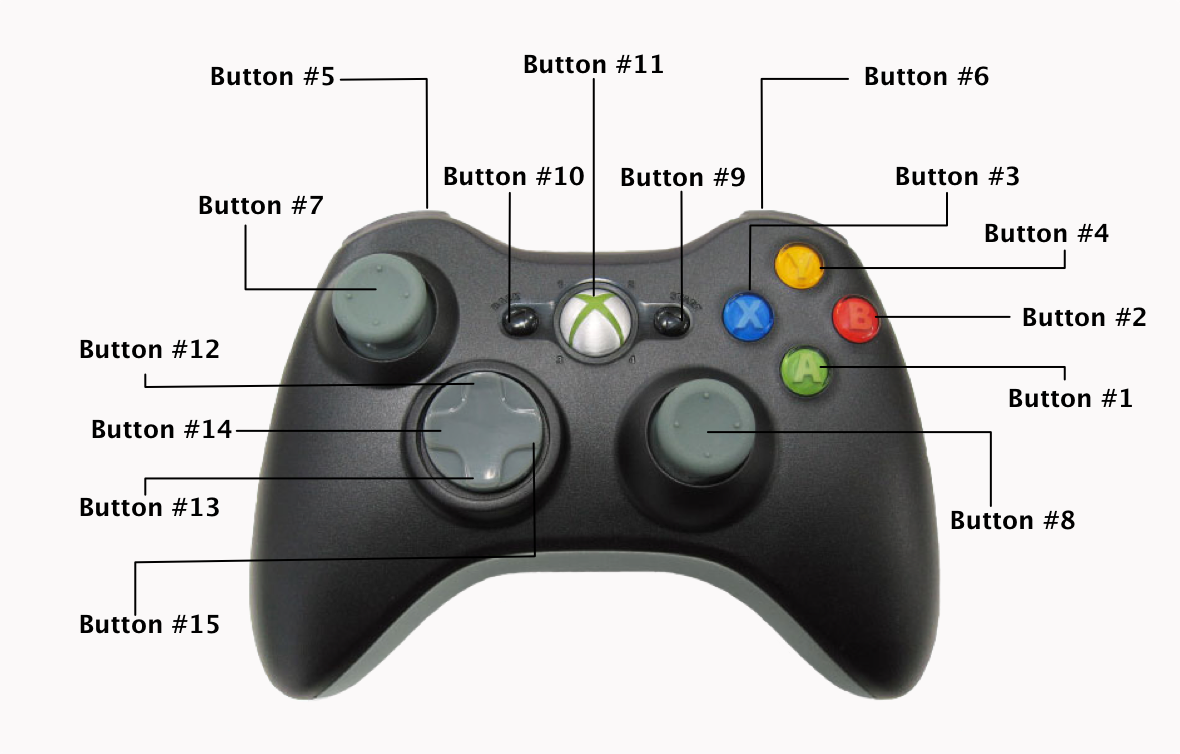
|  |  |  |
| --- | --- | --- |
| ***RoboRIO – Relay*** | | |
| ***Pin***  ***Assignment*** | ***Actuation*** | ***Signal Definition*** |
| ***0*** | ***Camera Light*** | ***ON (Fwd=1 & Rev=0)***  ***OFF (Fwd=0 & Rev=0)*** |
| ***1*** |  |  |
| ***2*** |  |  |
| ***3*** |  |  |

|  |  |  |
| --- | --- | --- |
| ***Pneumatic Control Module*** | | |
| ***Pin***  ***Assignment*** | ***Actuation*** | ***Signal Definition*** |
| ***0*** | ***Grasper*** | ***Single Solenoid***  ***Default OPEN***  ***Active CLOSED*** |
| ***1*** |  |  |
| ***2*** | ***Grasper***  ***Wrist*** | ***Single Solenoid***  ***Default UP***  ***Active DOWN*** |
| ***3*** |  |  |
| ***4*** | ***Hook***  ***Deployment*** | ***Single Solenoid***  ***Default UP***  ***Active DOWN*** |
| ***5*** |  |  |
| ***6*** |  |  |
| ***7*** |  |  |

**APPENDIX B: 2017 User Interface**

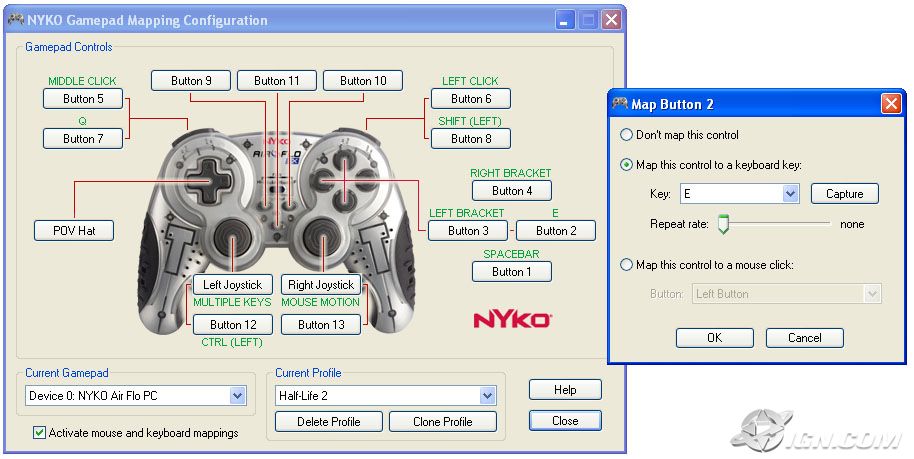
How do we partition functionality between the driver & operator? What game functions are drive dependent (depositing cube onto scale/switch or climbing rung)?

***Driver Controller***



| ***Robot***  ***Function*** | ***Game Pad Controller*** | ***Control Definition / Sequence*** | ***Control Constraint / Dependency*** |
| --- | --- | --- | --- |
| ***Throttle*** | ***Left Joystick – Button 7*** | ***Y-Axis = Throttle***  ***(Reverse < 0; Forward > 0)*** | ***N / A*** |
| ***Steering*** | ***Right Joystick – Button 8*** | ***X-Axis = Steering***  ***(Left < 0; Right > 0)*** | ***N / A*** |
| ***Release*** | ***Button 1*** |  |  |
| ***Deploy Hook*** | ***Button 4*** |  |  |

***Operator Controller***

****

| ***Robot***  ***Function*** | ***Game Pad Controller*** | ***Control Definition / Sequence*** | ***Constraint / Dependency*** |
| --- | --- | --- | --- |
| ***Cube***  ***Grasper*** | ***Operator game pad button 6***  ***When Pressed*** | ***Toggle cube grasper between OPEN & CLOSED*** |  |
| ***Grasper***  ***Wrist*** | ***Operator game pad button 5***  ***When Pressed*** | ***Toggle cube grasper wrist between UP & DOWN*** |  |
| ***Acquisition***  ***Wheels*** | ***Operator game pad POV hat***  ***When Pressed*** | ***Toggle between acquire (forward) & release (reverse)*** |  |
| ***Acquire***  ***Sequence*** | ***Operator game pad button 7***  ***When Pressed*** |  |  |
| ***Store***  ***Sequence*** | ***Operator game pad button 8***  ***When Pressed*** |  |  |
| ***Lift Position Floor Level*** | ***Operator game pad button 1***  ***When Pressed*** |  |  |
| ***Lift Position Cube Zone 2nd Level*** | ***Operator game pad button 2***  ***When Pressed*** |  |  |
| ***Lift Position Cube Zone 3nd Level*** | ***Operator game pad button 3***  ***When Pressed*** |  |  |
| ***Lift Position Switch Level*** | ***Operator game pad button 4***  ***When Pressed*** |  |  |
| ***Lift Position Scale Low Level*** | ***Operator game pad button 9***  ***When Pressed*** |  |  |
| ***Lift Position Scale Low Level*** | ***Operator game pad button 10***  ***When Pressed*** |  |  |
| ***Lift Position***  ***Rung Level*** | ***Operator game pad button 11***  ***When Pressed*** |  |  |
| ***Lift*** | ***Left Joystick*** | ***Lower lift (Y>0)***  ***Raise lift (Y<0)*** |  |
| ***Winch*** | ***Right Joystick*** | ***Unwind winch – lower robot (Y>0)***  ***Wind winch – raise robot (Y<0)*** |  |

# APPENDIX C: Autonomous Scenarios:

| ***Case*** | ***Starting Location*** | ***FMS String (X=Don’t Care)*** | ***Case*** | ***Options*** | ***Target*** | ***Mode*** |
| --- | --- | --- | --- | --- | --- | --- |
| ***1*** | ***Left*** | ***RLX*** | ***SCALE Strong*** | ***1*** | ***SCALE*** | ***1*** |
|  |  |  |  | ***2*** | ***SWITCH*** | ***2*** |
| ***2*** | ***Right*** | ***LRX*** |  |  |  |  |
|  |  |  |  | ***3*** | ***AUTO-LINE*** | ***3*** |
| ***3*** | ***Left*** | ***LRX*** | ***SWITCH Strong*** | ***1*** | ***SWITCH*** | ***4*** |
|  |  |  |  | ***2*** | ***SCALE*** | ***5*** |
| ***4*** | ***Right*** | ***RLX*** |  |  |  |  |
|  |  |  |  | ***3*** | ***AUTO-LINE*** | ***6*** |
| ***5*** | ***Left*** | ***LLX*** | ***SNAKE EYES*** | ***1*** | ***SCALE*** | ***7*** |
|  |  |  |  | ***2*** | ***SWITCH*** | ***8*** |
| ***6*** | ***Right*** | ***RRX*** |  |  |  |  |
|  |  |  |  | ***3*** | ***AUTO-LINE*** | ***9*** |
| ***7*** | ***Left*** | ***RRX*** | ***GOOSE EGGS*** | ***1*** | ***SCALE*** | ***10*** |
|  |  |  |  | ***2*** | ***SWITCH*** | ***11*** |
| ***8*** | ***Right*** | ***LLX*** |  |  |  |  |
|  |  |  |  | ***3*** | ***AUTO-LINE*** | ***12*** |
| ***9*** | ***Center*** | ***RLX*** | ***Right SWITCH Left SCALE*** | ***1*** | ***SWITCH*** | ***13*** |
|  |  |  |  | ***2*** | ***SCALE*** | ***14*** |
| ***10*** | ***Center*** | ***LRX*** | ***Left SWITCH Right SCALE*** | ***1*** | ***SWITCH*** | ***15*** |
|  |  |  |  | ***2*** | ***SCALE*** | ***16*** |
| ***11*** | ***Center*** | ***LLX*** | ***Both Left*** | ***1*** | ***SWITCH*** | ***15*** |
|  |  |  |  | 2 | ***SCALE*** | ***14*** |
| ***12*** | ***Center*** | ***RRX*** | ***Both Right*** | 1 | ***SWITCH*** | ***13*** |
|  |  |  |  | 2 | ***SCALE*** | ***16*** |

