**\*\*Please turn in this sheet during your Judge Interview along with your Engineering Notebook\*\***

|  |  |
| --- | --- |
| **Team # 9042** | **Team Name: The MidKnight Magic** |

**Autonomous objectives:**

* **Drop Climbers in Shelter**
* **Park in Same Side Mountain Midzones**
* **Park in Rescue Beacon Repair Zone**

**Sensors used:**

* **Encoders – Driving straight (keeping robot on course) to a certain distance, Making accurate turns, Stopping propeller parallel to drive base, Alternating the propeller to clear debris.**
* **Touch Sensor – Prevent climbers from being dumped if the robot is not in right position.**

**Key algorithms:**

* **Autonomous State Control – Using Java States to make programming the autonomous programs chronological so all functions do not occur at the same moment, and easier to read and troubleshoot.**
* **Parallel Propeller – Using the encoder on the propeller motor to align the propeller parallel to our drive base in order to facilitate climbing up the mountain. The algorithm finds the nearest half rotation of the propeller, how much farther the propeller has to travel to reach that state, and then sets that value as the target.**
* **Encoder Drive Straight – Using the encoders on all 4 wheels, we designed a logarithm that would allow us to type in the number of inches the robot wanted to travel straight, and the robot would go to that distance. We did this using the circumference of each wheel by the encoder ticks per rotations of the motor to find the ticks:inches ratio.**

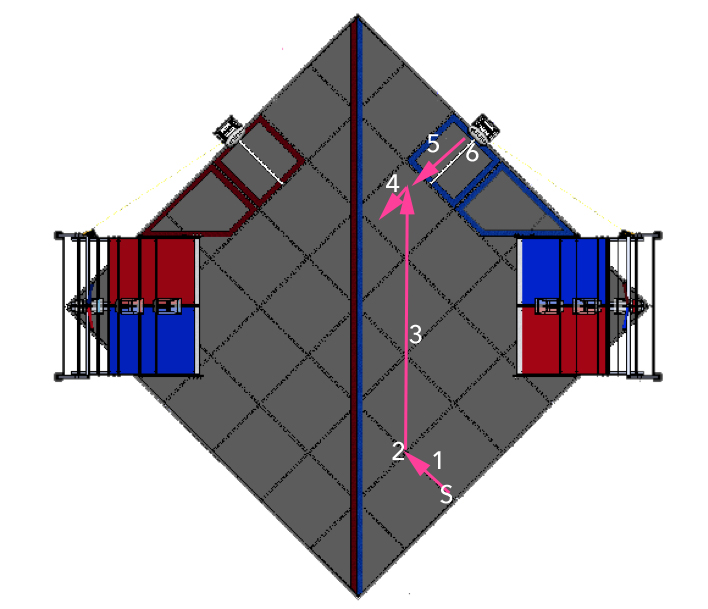
**Driver controlled enhancements:**

* **Turtle Drive – Slows down all driving and turning so driver can make small adjustments in order to prepare for mountain ascension, and robot does not flip over while trying to overcome bars.**
* **Back Drive – If climbers do not dump in autonomous, the driver has a reverse driving option because the climber dumping mechanism is on the back of our robot. This allows the driver to operate the drive train with drive controls reversed to allow for simple alignment without having to drive the robot backwards.**
* **Continuous Servo Stop – For our continuous servo rack and pinion assembly that knocks down the zipline levers, we added a constant (9/17) that is the exact middle of the servo input values that allows our servo to stop keeping it from burning out.**

**Engineering notebook references:**

* **1. Finding the Servo Goldilocks value – pg. 52**
* **2. Encoder Drive Code – Programming pg. 28**
* **3. Autonomous Encoder Propeller Code – Programming pg. 29-31**

**Autonomous program diagram for Blue Side (Flip for Red Side):**



**S – Starting point between the 2nd and 3rd mats from bottom with robot facing forwards.**

**1 – Go forward 15 inches with encoders to provide a clear path across the field.**

**2 – Turn 45°** **with encoders to be semi-parallel to the middle line so the robot does not incur the 50 point penalty for crossing the line before 10 seconds.**

**3 – Drive straight 96 inches across the field to get in position to back up into the shelter.**

**4 – Turn around 135° to align robot to back up into the shelter zone while propeller knocks debris into the floor goal.**

**5 – Back up until the back end of robot is touching the rescue beacon.**

**6 – Activate servo to its 0.5 position to drop climbers into the shelter.**

**Variation for climbing mountain Autonomous’ is to stop midway on step 3, turn right 90 degrees, and drive slowly up the mountain over two churros to park in midzone.**