Table of Contents

List of Figures

1 Subsystem Name

- Rename this file to be "`SubsystemName.md`"
- Briefly (3 or fewer sentences) sumarize the strategic goal of this subsystem as it relates to the current game.

1.1 Coding

- (if re-using external code) credit the original source
- link to the source if available
- Include example definition of the subsystem's class:

```
// Example instantiation within RobotContainer.java
private final MySubsystem mySubsystem = new MySubsystem();
```

1.1.1 Tele-op control

- How will the subsystem be controlled in tele-op?

1.1.2 Autonomous control

- How will the subsystem be controlled in autonomous (if different than tele-op)

1.2 Electrical

1.2.1 Feedback

- Sensor Types and their use-cases

1.2.2 Interface types

- CAN/I2C/SPI/PWM/Ethernet?
- Address/ID?
- Wiring Diagram showing both data *and* power connections

1.3 Mechanical

- Initial source material source
- link to the source if available

1.3.1 CAD Models

- Picture(s) of the CAD Model, **NOT** from a phone camera

- Label parts of the model.

2 Elevator Subsystem

The Elevator Subsystem's purpose is to lift the Algae and Coral intakes and hold at steady heights so that

2.1 Coding

```
// Example instantiation within RobotContainer.java
private final ElevatorSubsystem elevatorSubsystem = new ElevatorSubsystem(
```

2.1.1 Tele-op control

- In Tele-Op mode, the Elevator will be controlled by setting a target position
- Code snippit of how go-to-position commands are called in `RobotContainer.java`:

```
controller.button(0).onTrue(new ParallelCommandGroup(
        new SequentialCommandGroup(
            new MoveElevatorToPositionAuto(
                elevatorSubsystem,
                20
            ),
            new ParallelCommandGroup(
                Commands.run(
                    () -> elevatorSubsystem.MoveElevatorToPosition(20)
                )
            )
        )
    )
);
```

2.1.2 Autonomous control

- Autonomous will re-use the functions designed for Tele-Op to
- set target positions corresponding to the motor's encoder value
- use embedded PID controller with encoder's value used as feedback

2.2 Electrical

2.2.1 Feedback

- Using Motor's encoder, this should provide an accurate estimation of the elevator's height based on how many rotations the motor has made.

2.2.2 Interface types

- Controlled via CAN bus through a REV Robotics SPARK MAX
- CAN ID: **15**

2.3 Mechanical

- Initial source material source
- link to the source if available

2.3.1 CAD Models

- Picture(s) of the CAD Model, **NOT** from a phone camera
- Label parts of the model.