

Day 12 CS570

Encoders and Gyros

Encoders

Encoders are a way to find out how much a motor has moved. There are many different kind of encoders, but we will focus on the kind of encoders that are on the ROMI's. These are **quadrature** encoders. They can tell you how far the wheels have moved by counting ticks of 2 sensors. Watch this youtube video to see more about how this works:

<https://www.youtube.com/watch?v=p4BCFhIuC88>

or more information can be found at:

<https://www.youtube.com/watch?v=oLBYHbLO8W0>

On the ROMI the encoders are visible. They are the little black pieces on the ends of the white motors that are attached to the main board.



The math of encoders The ROMI's encoders measure how much the motor has turned, but the motor turns the a gearbox which turns the wheel, and the size of the wheel contributes to how far the ROMI will move.

Challenge Work with other students near you to determine how much the ROMI moves in meters for one tick of the encoder.

You will need more information about the ROMI's motors, encoders, and gearing. I suggest that you look at this website: [Getting to know your ROMI](#)

Coding for Encoders There are Encoder objects in `wpiLib`:

```
from wpiLib import Spark, Encoder
...

self.leftEncoder = Encoder(4, 5)
self.rightEncoder = Encoder(6, 7)
```

The encoders are attached to pins 4 and 5 on the left and 6 and 7 on the right on the ROMI. The ROMI's take two pins because they are quadrature encoders that allow us to determine direction as well as distance.

You can let the encoders know how much each tick is worth in terms of distance using the `setDistancePerPulse` method.

You can reset the encoders back to zero using the `reset` method.

You can get the number of ticks the encoder has registered using the `get` method.

If you set the conversion for ticks to distance then you can use the encoder's `getDistance` method to find the distance travelled.

Project 3: Make ROMI go 2 meters in a straight line With the information above let's think about how we would make an autonomous routine where the ROMI moves straight for 2 meters.

Project Requirements The robot must be driveable in Teleop

- When you exit Autonomous you must make it so that the robot will perform the autonomous again when you reenter Autonomous.

Drivetrain class

- You must have a drivetrain class, that you use to drive the robot in Teleop and that you pass to your DriveStraight class.
- the drivetrain class must implement the motor encoders, and you must have methods that allow you to determine how far each wheel has rotated, and how much distance that would have covered in total.

- You can have other methods that you find useful as well, you may want to have a method that averages the distances the wheels have traveled for instance.

DriveStraight Class

This class will be in charge of the robot during autonomous.

- The DriveStraight class should take in a drivetrain and how far it is supposed to drive straight in meters.
- The DriveStraight class should have a run method that gets called in **autonomousPeriodic**. In the run method, the class should determine based on information from the motor encoders.

Warning

- The motors will not run straight exactly even if you tell them to. You will need to turn the robot to compensate.
- You may have to tune your constants a little to get the robot to drive the right distance.