Archive

- Phoenix 6 Migration
- Intake Logic

Phoenix 6 Migration

Pheonix 5 APi is officially deprecated

Phoenix 6 PID reference in converting PID gains: https://pro.docs.ctr-electronics.com/en/latest/docs/api-reference/device-specific/talonfx/basic-pid-control.html

- Take aways moving from native units (encoder tics) to canonical (rotations per second)
- PID constants must therefore be converted using this calculator (https://pro.docs.ctr-electronics.com/en/stable/docs/migration/migration-guide/closed-loop-guide.html)

Everything is now in rotations/sec therefore we can remove the conversion to encoder tics per 100 ms

Cancoder offsets Debugging

Possible Causes:

- Magnet Direction
  - Directionality is different so the offset might need to be inverted. Check within tuner X what the value is the **Counter Clockwise Positive** as the magnet direction
  - The value should be somewhere between -1 and 1. The sensor range is -0.5 to 0.5 so the offset could also be bounded between that.
  - HardwareMap.Java

  -
    ```
      public static final SwervePodHardwareID POD001 = new SwervePodHardwareID(10, 12, -172.135);
      public static final SwervePodHardwareID POD002 = new SwervePodHardwareID(20, 22, -225.186);
      public static final SwervePodHardwareID POD003 = new SwervePodHardwareID(30, 32, -40);
      public static final SwervePodHardwareID POD004 = new SwervePodHardwareID(40, 42, 140.463); // 120.5
      public static final SwervePodHardwareID POD005 = new SwervePodHardwareID(13, 14, -30.525);
      public static final SwervePodHardwareID POD006 = new SwervePodHardwareID(23, 24, 105);
      public static final SwervePodHardwareID POD007 = new SwervePodHardwareID(33, 34,
    ```

```
125.508);
    public static final SwervePodHardwareID POD008 = new SwervePodHardwareID(43, 44, -173);
    public static final SwervePodHardwareID POD009 = new SwervePodHardwareID(15, 16, -178);
```

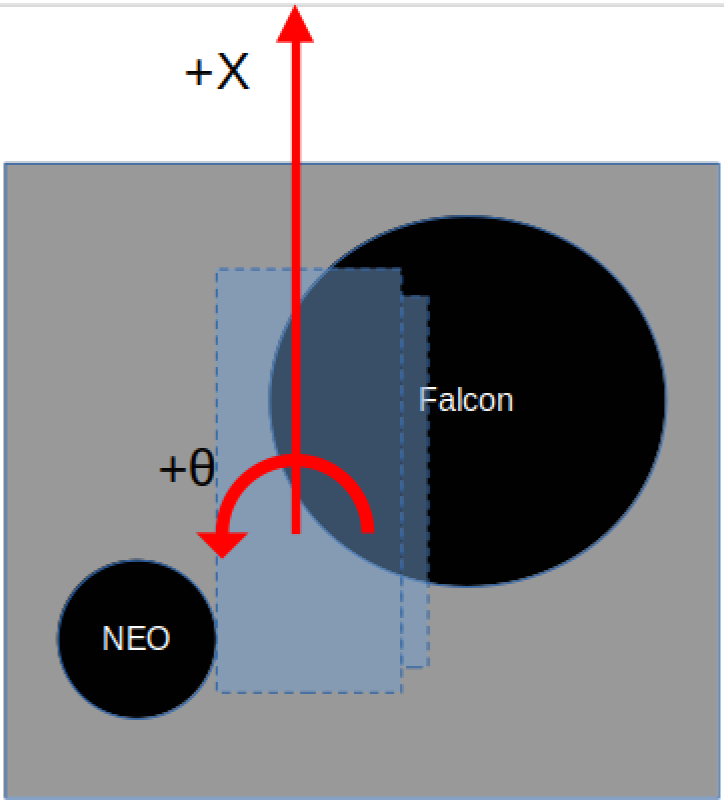
Getting encoder offsets

(this should be its own wiki page once we get it solidified with good screen shots)

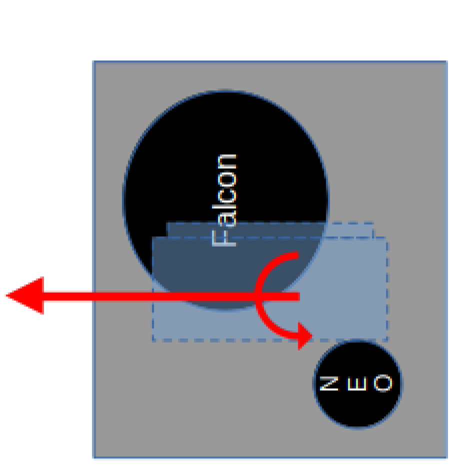
Requirements

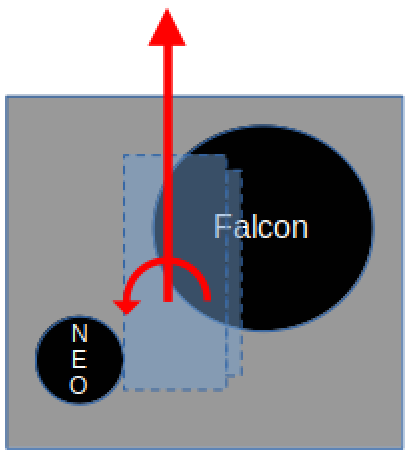
- USB B cable
- Windows laptop with Tuner X
- WPILIB VSCode

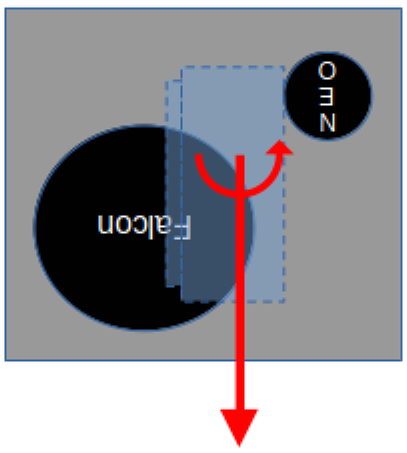
Steps

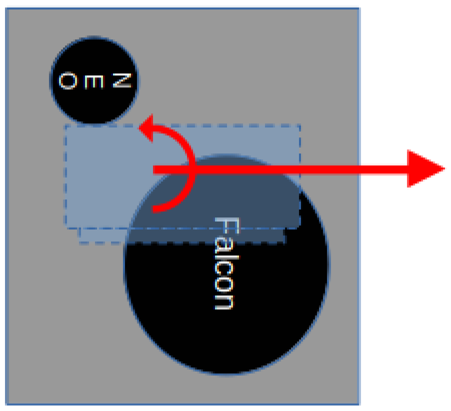
- Place pods into the tuning configuration
  - Because the pods orientation changes depending on which slot it is in (FR, FL, BR, BL) Pods are zeroed **AS IF** they were installed in the FR.
  - For the FR (front right) pod the wheel is turned so that it is facing forward. Looking from behind the driving gear should be facing to the right of the bot
  - 

  - Continue by placing the pods in the following configuration



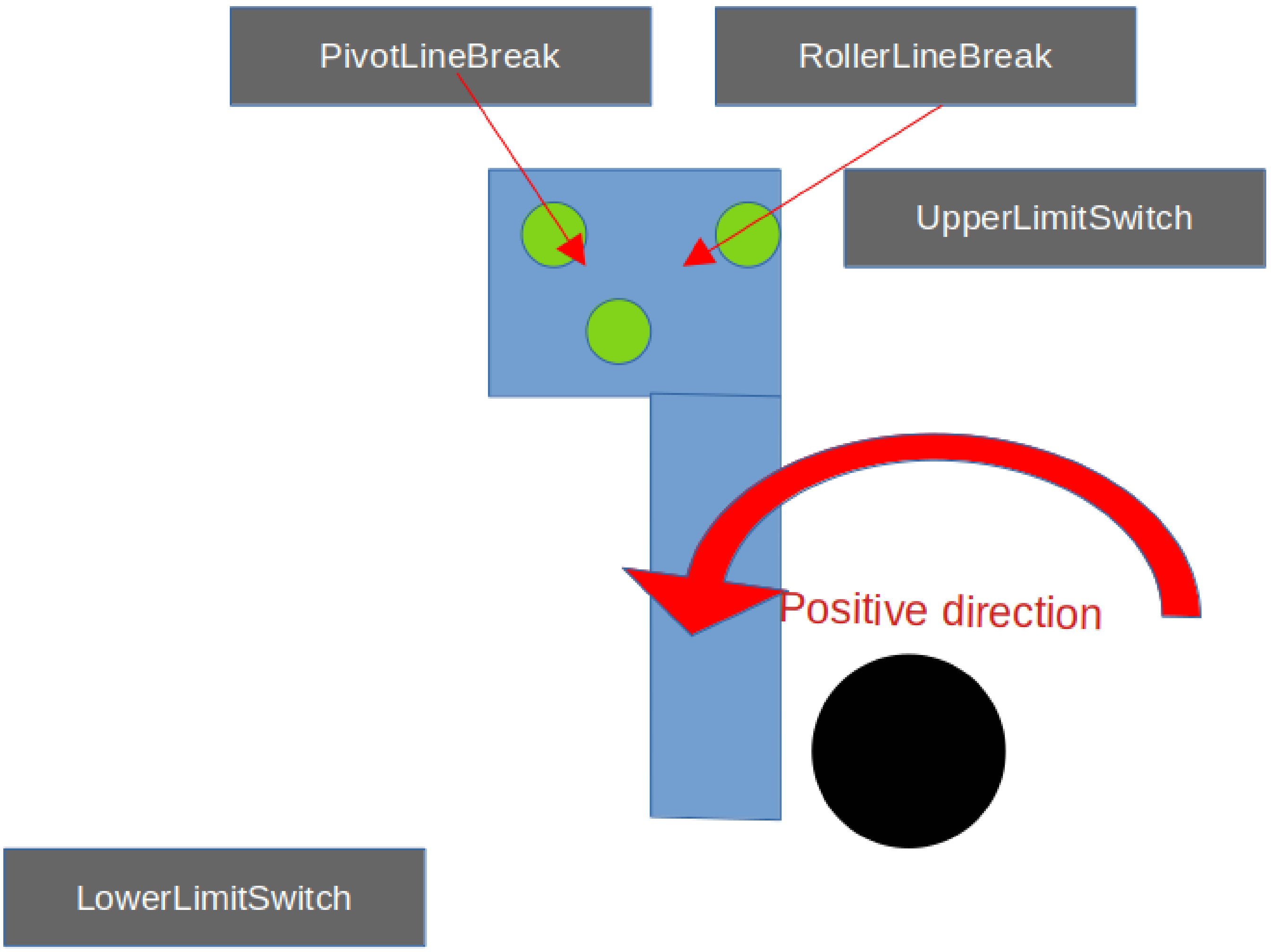






- Set Magnet offset to 0 in Tuner X
- Set Magnet direction to Counter Positive
- Run selftest snapshot
- Save Offsets to the `HardwareMap.java` the value should be between -0.5 and 0.5 for the pod number you are reading

Intake Logic



We need to ensure the direction of the motor is correct before we go.

Take a look at runPivot() within Intake() which uses the limit switches to protect the mechanism

State Machine

Review state machines with this video: https://www.youtube.com/watch?v=-Yicg2TTMPs&ab_channel=MATLAB

We have a state machine for our pivot. Check out the states defined in intake

Then look how the states are defined within periodic()

You will now need to implement the command PivotRetract by changing the state of the pivot;

Intake command

there are a few additional commands that need to be implements for our intake command to function

Take a look at the TODO and implement those commands.

LoggedTunableNumbers

test you work if it performs correctly change your hardcoded numbers into LoggedTunableNumbers so we can easily change the following

- pivot deploy and retract voltage
- intake roller voltage
- Wait time from the time linebreak is triggered and rollers stop