As a solution, we changed our approach in the programming. Instead of relying on the rotation angle of the catapult from the rotation sensor, we used the encoder in the motor to measure the rotation angle. By monitoring the torque values returned by the motor encoder, we were able to determined the position of the catapult arm, achieving the same effect as the original rotation sensor.

```
int rcThrow()
{
  waitUntil(autoFinished || Controller1.Axis3.position()>20);
  motorThrow.setBrake(hold);
  motorThrow.spin(fwd,50,pct);
  waitUntil(motorThrow.torque()>0.3);
  waitUntil(motorThrow.torque()<0.12);
  motorThrow.stop();
  wait(200, msec);
  motorThrow.setBrake(coast);
  motorThrow.resetPosition();
  throwDown(true);
  double tmp;
  while(1)
  {
    if(Controller1.ButtonUp.pressing())
    {
      throwDown(false);
      std::cout<<"123"<<std::endl;
      tmp = motorThrow.position(deg);
      motorThrow.spin(fwd,80,pct);
      waitUntil(motorThrow.position(deg)>tmp+100);
      motorThrow.stop();
      wait(100, msec);
      throwDown(true);
    }
    wait(5, msec);
  }
}
```

project program IZY witnessed by: Joker

date: 12.2