//(Arduino) Driving DC motors straight using PWM

//Set parameters, pins, interrupts

//Encoder is 16 counts per revolution (CPR)

void drivestraight(){

set initial motor power

calculate/set encoder target count (based on robot deployment range)

while (enc\_l < target\_count && enc\_r < target\_count) {

//set temporary variable

enc\_l = encoder(L);

enc\_r = encoder(R);

temp\_l = enc\_l;

temp\_r = enc\_r;

//compute difference

diff\_l = temp\_l - enc\_l\_prev;

diff\_r = temp\_r - enc\_r\_prev;

//Set the next values for enc\_prev

enc\_l\_prev = temp\_l;

enc\_r\_prev =temp\_r ;

if ( diff\_l > diff\_r ) {

decrease left motor speed

increase right motor speed

}

if ( diff\_r > diff\_l ) {

decrease right motor speed

increase left motor speed

}

}

}

//Using the encoder

void encoder(L or R){

reset encoder counts (left and right encoders)

while(1){ //loop and count encoder ticks

if (encoder changes from low to high){

increment encoder count

}

}

}

//Driving the DC motors

void drive(){

//Set DC motor direction

digitalWrite()

//Set DC motor speed

analogWrite() //Writes analog values (PWM wave) to a pin

}

//distance function can be implemented to determine pole-to-pole distance by setting variables for each START to POLE distance (dist1, dist2...)

void distance(){

ticks = number of times sensor is triggerred

if (digitalRead(HallEffectSensorpin) == HIGH){

ticks++

}

calculate distance using wheel circumference, # of ticks per revolution, # motor rev/wheel (calibrate distance to measure pole center)

average l and r encoder distances

if (dist\_total >= employment range){

stop powering motors (brake)

}

}

void brake(){

analogWrite(pins, 0) //no power

}