//motion part

ledcSetup(0, freq, res);

ledcSetup(1, freq, res);

ledcAttachPin(ENA, 0);

ledcAttachPin(ENB, 1);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

milliNew = millis();

}

void loop() {

if (tryNumber == 1) { //check for the try number, then run the appropriate code

firstTry();

}

else {

secondTry();

}

}

bool cycle(float x, float y)

{

x += 360, y += 360;

if (abs(x - y) > (360 - max(x, y)) + min(x, y))

return true;

return false;

}

void turnRight()

{

set\_point += 90;

angle = imu\_6050();

if (set\_point > 180)

{

set\_point -= 360;

}

while (cycle(angle, set\_point) ? angle > set\_point - 15 : angle < set\_point - 15)

{

right();

angle = imu\_6050();

}

milliNew = millis();

stopp();

}

void turnLeft()

{

angle = imu\_6050();

set\_point -= 90;

if (set\_point < -180)

{

set\_point += 360;

}

while (cycle(angle, set\_point) ? angle < set\_point + 15 : angle > set\_point + 15)

{

left();

angle = imu\_6050();

}

milliNew = millis();

stopp();

}

void right()

{

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

ledcWrite(0, spd);

ledcWrite(1, spd);

}

void left()

{

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

ledcWrite(0, spd);

ledcWrite(1, spd);

}

void forward()

{

angle = imu\_6050();

float rcorrection = min(abs(correction / rscale), spd) \* (correction / abs(correction ? correction : 1));

float lcorrection = min(abs(correction / lscale), spd) \* (correction / abs(correction ? correction : 1));

Serial.print("Set Point: "); Serial.print(set\_point);

Serial.print(", Right Correction: "); Serial.print(rcorrection);

Serial.print(", Left Correction: "); Serial.print(lcorrection);

Serial.print(", Angle: "); Serial.print(angle);

Serial.print(", Right Speed: "); Serial.print(spd - rcorrection);

Serial.print(", Left Speed: "); Serial.print(spd + lcorrection);

Serial.println();

ledcWrite(0, spd - rcorrection); //right

ledcWrite(1, spd + lcorrection); //left

}

void stopp()

{

ledcWrite(0, 0);

ledcWrite(1, 0);

}