```
#include <PS2X_lib.h>
#include <Servo.h>

#define MAX 133
#define STA 56
//lifting values
#define LFT1 62
#define LFT2 63
#define LFT3 64
//propulsion values
#define PUL1 75
#define PUL2 80
#define PUL3 85
#define PUL4 95

PS2X ps2x;
Servo rudder;
Servo motor_l;
Servo motor_p;

int error = 0;
byte type = 0;
byte vibrate = 0;
int angle;
int prop = STA;
int turn = 0;
int cst = STA;
int cst0 = STA;
int counter = 0;
int time = 0;
int test = 0;

void setup(){
  Serial.begin(57600);
  rudder.attach(4);
  motor_l.attach(5);
  motor_p.attach(6);
  motor_l.write(MAX);
  motor_p.write(MAX);
  delay(1000);
  error = ps2x.config_gamepad(13,11,10,12, true, true);

  if(error == 0)
    Serial.println("Found Controller, configured successful");
  else if(error == 1)
    Serial.println("No controller found, check wiring.");
  else if(error == 2)
    Serial.println("Controller found but not accepting commands. see readme.txt to
enable debug.");
  else if(error == 3)
    Serial.println("Controller refusing to enter Pressures mode, may not support
it.");

  type = ps2x.readType();
  switch(type){
    case 0:
      Serial.println("Unknown Controller type");
      break;
    case 1:
      Serial.println("DualShock Controller Found");
      break;
    case 2:
      Serial.println("GuitarHero Controller Found");
      break;
  }
}

void loop(){
  if(error == 1) //skip loop if no controller found
    return;

//DualShock Controller
  ps2x.read_gamepad(false, vibrate);          //read controller and set large motor
to spin at 'vibrate' speed

//turning part
  angle = 95;
  if(ps2x.Analog(PSS_LY)&&ps2x.Analog(PSS_RY)){
    angle = 11 * ps2x.Analog(PSS_LX) / 47 + 66;
    angle = angle + 10 * (ps2x.Analog(PSS_RX) - 128) / 127;
  }
  rudder.write(angle);

//auto-braking at turning
  if(0<=counter&&counter<30){
    test = 0;
  }
  else if(30<=counter&&counter<45){
    test = 1;
  }
  else if(45<=counter&&counter<70){
    test = 2;
  }
  else{
    test = 3;
  }

  switch(test){
    case 0:
      time = 0;
      break;
    case 1:
      time = 25;
      break;
    case 2:
      time = 75;
      break;
    case 3:
      time = 125;
      break;
  }

  if(ps2x.Analog(PSS_LX)==255||ps2x.Analog(PSS_LX)==0){
    turn = 1;
    if(ps2x.Button(PSB_R2)){
      motor_l.write(LFT2);//turning on the friction
    }
  }
  else{
    if(turn==1){
      turn = 0;
      counter = 0;
      motor_l.write(STA);
      delay(time);
      if(ps2x.Button(PSB_R2)){
        motor_l.write(LFT3);
      }
      else if(ps2x.Button(PSB_L2)){
        motor_l.write(LFT2);
      }
      else{
        motor_l.write(cst);
      }
    }
  }

  if(ps2x.Button(PSB_SELECT)==0){
  //initializing
    if(ps2x.ButtonPressed(PSB_START)){
      motor_l.write(STA);
      motor_p.write(STA);
      delay(1000);
    }

  //general lifting
    if(ps2x.Button(PSB_PAD_DOWN)){
      cst = STA;
      motor_l.write(cst);
    }
    else if(ps2x.Button(PSB_PAD_UP)){
      cst = LFT1;
      motor_l.write(cst);
    }

  //stalls
    if(ps2x.ButtonPressed(PSB_BLUE)){
      prop = PUL1;
    }
    else if(ps2x.ButtonPressed(PSB_PINK)){
      prop = PUL2;
    }
    else if(ps2x.ButtonPressed(PSB_GREEN)){
      prop = PUL3;
    }
    else if(ps2x.ButtonPressed(PSB_RED)){
      prop = PUL4;
    }

  //motion
    if(ps2x.Button(PSB_L2)){
      counter = counter + 1;
    }
    else if(ps2x.Button(PSB_R2)){
      counter = counter + 1;
    }

    if(ps2x.ButtonPressed(PSB_R2)){
      motor_p.write(prop);
      cst0 = cst;
      cst = LFT3;
      motor_l.write(cst);
      vibrate = ps2x.Analog(PSAB_R2);
    }
    else if(ps2x.ButtonPressed(PSB_L2)){
      motor_p.write(prop);
      cst0 = cst;
      cst = LFT2;
      motor_l.write(cst);
      vibrate = ps2x.Analog(PSAB_L2);
    }
    else if(ps2x.ButtonReleased(PSB_R2)){
      motor_p.write(STA);
      motor_l.write(STA);
      delay(100);
      cst = cst0;
      motor_l.write(cst);
      counter = 0;
      vibrate = 0;
    }
    else if(ps2x.ButtonReleased(PSB_L2)){
      motor_p.write(STA);
      motor_l.write(STA);
      delay(100);
      cst = cst0;
      motor_l.write(cst);
      counter = 0;
      vibrate = 0;
    }

  //stop
    if(ps2x.ButtonPressed(PSB_L1)){
      if(ps2x.ButtonPressed(PSB_R1)){
        cst = STA;
        prop = STA;
        motor_p.write(prop);
        motor_l.write(cst);
      }
      else{
        motor_l.write(STA);
      }
    }
    else if(ps2x.ButtonReleased(PSB_L1)){
      motor_l.write(cst);
    }
  }
  delay(50);
}
```