

```
#include <MeMegaPi.h>
#include <Wire.h>
#include <SoftwareSerial.h>

MeLineFollower lineFinder(PORT_6);
MeGyro gyro(PORT_7);
MeUltrasonicSensor ultraSensor(PORT_8);

//motor_setup
MeMegaPiDCMotor motor1(PORT1B);
MeMegaPiDCMotor motor2(PORT2B);
MeMegaPiDCMotor motor3(PORT3B);
MeMegaPiDCMotor motor4(PORT4B);
uint8_t motorSpeed = 80;
uint8_t motorSpeed2 = 80;
uint8_t motorHiSpeed = 130;
uint8_t motorHiSpeed2 = 130;
uint8_t motorSpeed3 = 80;
uint8_t motorStop = 0;
int WallCount = 0;
//motor_setup_end

void setup(){
  Serial.begin(9600);
  gyro.begin();
  gyro.update();
  while (gyro.getAngleY() < 50){ // 50
    motor3.run(motorSpeed3);
    motor4.run(-motorSpeed3);
    Serial.print("Y: ");
    Serial.println(gyro.getAngleY());
    gyro.update();
  }
  motor3.run(motorStop);
  motor4.run(-motorSpeed3);
  delay(3000);
  motor4.run(motorStop);
}

void loop(){
  //UltraSoundSensor
  Serial.print("Distance : ");
  Serial.print(ultraSensor.distanceCm() );
  Serial.println(" cm");
  delay(100);

  if (ultraSensor.distanceCm() < 8){
    motor1.run(motorStop);
    motor2.run(motorStop);
    gyro.begin();
    gyro.update();
    while (gyro.getAngleY() > -7){
      motor3.run(-motorSpeed3);
      Serial.print("Y: ");
      Serial.println(gyro.getAngleY());
      gyro.update();
    }
    motor3.run(motorStop);
    if (WallCount == 0){
      motor4.run(motorSpeed3);
      delay(4000);
      WallCount++;
    }
    else{
      motor4.run(-motorSpeed3);
      delay(4000);
      WallCount = 0;
    }
    motor4.run(motorStop);
  }
```

```
gyro.begin();
gyro.update();
while (gyro.getAngleY() < 50){
    motor3.run(motorSpeed3);
    Serial.print("Y: ");
    Serial.println(gyro.getAngleY());
    gyro.update();
}
motor3.run(motorStop);
motor1.run(motorSpeed);
motor2.run(-motorSpeed2);
delay(500);
gyro.begin();
gyro.update();
while (gyro.getAngleZ() < 80){
    motor1.run(-motorHiSpeed);
    motor2.run(-motorHiSpeed2);
    Serial.println(gyro.getAngleZ());
    gyro.update();
}
motor1.run(motorStop);
motor2.run(motorStop);
}
else{
    //LineTrace
    int sensorState = lineFinder.readSensors();
    switch(sensorState){
        case S1_IN_S2_IN:
            //Serial.println("Sensor 1 and 2 are inside of black line");
            motor1.run(-motorSpeed);
            motor2.run(motorSpeed2);
            break;
        case S1_IN_S2_OUT:
            //Serial.println("Sensor 2 is outside of black line");
            motor1.run(-motorSpeed);
            motor2.run(-motorSpeed2);
            break;
        case S1_OUT_S2_IN:
            //Serial.println("Sensor 1 is outside of black line");
            motor1.run(motorSpeed);
            motor2.run(motorSpeed2);
            break;
        case S1_OUT_S2_OUT:
            //Serial.println("Sensor 1 and 2 are outside of black line");
            motor1.run(motorSpeed);
            motor2.run(-motorSpeed2);
            break;
    }
    delay(70);
}
}
```