

```

#include <Arduino.h>
#include <analogWrite.h>
#include <cmath>
#include <Adafruit_BNO055.h>
Adafruit_BNO055 bno = Adafruit_BNO055(55, 0x28);

const int numberOfBallSensors = 8;
float ballVecx = 0.0;
float ballVecy = 0.0;
int ballSensorPorts[] = {25, 26, 36, 39, 34, 35, 32, 33};
int ballValues[8];
int motor1Power = 0;
int motor2Power = 0;
int motor3Power = 0;
float OriginalDirection = 0;

float GetDirection() {
    sensors_event_t ev;
    bno.getEvent(&ev);
    return ev.orientation.roll;
}

void moveMotor1(int power, bool Direction)
{
    power = max(min(power, 255), 0); //the max power for analogWrite() is
255 and minimum is 0

    analogWrite(0, power); //this pin determines the power that the motor
will spin

    //these two pins determine clockwise or counter-clockwise motion
    digitalWrite(2, Direction);
    digitalWrite(15, !Direction);
}

void moveMotor2(int power, bool Direction)
{
    power = max(min(power, 255), 0); //the max power for analogWrite() is
255 and minimum is 0

    analogWrite(4, power); //this pin determines the power that the motor
will spin

```

```

    //these two pins determine clockwise or counter-clockwise motion
    digitalWrite(16, Direction);
    digitalWrite(17, !Direction);
}

void moveMotor3(int power, bool Direction)
{
    power = max(min(power, 255), 0); //the max power for analogWrite() is
    255 and minimum is 0

    analogWrite(19, power); //this pin determines the power that the
    motor will spin

    //these two pins determine clockwise or counter-clockwise motion
    digitalWrite(18, Direction);
    digitalWrite(5, !Direction);
}

void setup()
{
    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(25, INPUT);
    pinMode(26, INPUT);
    pinMode(36, INPUT);
    pinMode(39, INPUT);
    pinMode(34, INPUT);
    pinMode(35, INPUT);
    pinMode(32, INPUT);
    pinMode(33, INPUT);

    pinMode(27, INPUT);
    pinMode(14, INPUT);
    pinMode(12, INPUT);
}

```

```

pinMode(5, OUTPUT);
pinMode(18, OUTPUT);
pinMode(19, OUTPUT);

pinMode(17, OUTPUT);
pinMode(16, OUTPUT);
pinMode(4, OUTPUT);

pinMode(0, OUTPUT);
pinMode(2, OUTPUT);
pinMode(15, OUTPUT);

bno = Adafruit_BNO055(55);
if (!bno.begin()) {
  Serial.println("COMPASS STUCK COMPASS STUCK AHH");
  while (1) {
  }
}
bno.setExtCrystalUse(true);
uint8_t system, gyro, accel, mag;
while (system != 3)
{
  Serial.printf("It's not fully calibrated : %d, %d, %d %d\r\n",
system, gyro, accel, mag);
  bno.getCalibration(&system, &gyro, &accel, &mag);
  delay(500);
}
Serial.println("It has calibrated!");
moveMotor1(50, 0);
delay(500);
moveMotor1(0,0);
delay(4000);
OriginalDirection = GetDirection();
}

void readBallValues()
{
  for (int i = 0; i < numberOfBallSensors; i++)
  {

```

```

        ballValues[i] = 4096 - analogRead(ballSensorPorts[i]);
    }
}

int readLine1()
{
    return (4096 - analogRead(27));
}

int readLine2()
{
    return (4096 - analogRead(14));
}

int readLine3()
{
    return (4096 - analogRead(12));
}

void printBallValues()
{
    for (int i = 0; i < numberOfBallSensors; i++)
    {
        Serial.print("Ball Sensor " + String(i) + " is " +
String(ballValues[i]) + " | ");
    }
    Serial.println("");
}

void loop()
{
    //prints hello to the terminal
    //Serial.println(); does the printing and the thing being printed is
    //placed inside. In this case the String "Hello" is placed inside.
    Serial.println("Hello");

    //a variable or function that returns an integer can also be
    //printed, but be sure to convert it to a String inside the print

```

```
//statement
Serial.println(String(readline1()));

//reads line sensors. Returns an integer value.
readLine1();
readLine2();
readLine3();

//you can store the values returned by these functions in a variable
int x = readline1();

//refreshes ball sensor values. The values are stored in an array
//called ballValues
readBallValues();
//to access individual ball sensor values, just use ballValues[i]
//where i is a number from 0 to 7 assigned clockwise from the left of
//the robot
//example: ballValues[2] would return the ball sensor value from the
//sensor pointing forward closest to the flat part of the robot.

//makes the robot wait for 100 milliseconds.
delay(100);

//moveMotor1 moves a motor. The first argument is the power from 0 to
//255. The second argument is the direction the motor should spin
//either 0 or 1.
//the example below moves motor1 at maximum speed in the direction 0
moveMotor1(255, 0);

//this moves motor 2 with half maximum speed in the direction 1
moveMotor2(127, 1);

//this makes motor 3 stop. The direction does not matter in this case
moveMotor2(0, 1);

}
```