Obstacle Avoiding Robot Kit (Arduino)

Circuit Connections:

Arduino to Motor Driver

Arduino	Motor Driver
9	EN A
3	IN1
4	IN2
5	IN3
6	IN4
10	EN B
Vin	Vin / Vcc
GND	GND

Arduino to HC SR-04 Ultrasonic Distance Sensor

Arduino	HC SR-04
A0	Trig
A1	Echo
5V	Vcc
GND	GND

* Connect battery power jack to Arduino DC socket.

* Connect motors to the MOTOR A and MOTOR B screw terminal. On each terminal, reverse the 2 wires of motor in case the corresponding motor starts rotating backwards after uploading code.

Arduino Code:

// (Please copy one page at a time)

```
#define trigPin A0
#define echoPin A1

#define MotorA_IN1 3
#define MotorA_IN2 4
#define MotorB_IN3 5
#define MotorB_IN4 6

#define MotorA_PWM 9
#define MotorB_PWM 10


void setup()
{
  pinMode(MotorA_IN1, OUTPUT);
  pinMode(MotorA_IN2, OUTPUT);

  pinMode(MotorB_IN3, OUTPUT);
  pinMode(MotorB_IN4, OUTPUT);

  pinMode(MotorA_PWM, OUTPUT);
  pinMode(MotorB_PWM, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}


int search(void)
{
  float duration = 0.00;
  float CM = 0.00;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  CM = (duration / 58.82);
  return CM;
}
```

```
void loop()
{
  float distance = 0.00;
  float RobotSpeed = 0.00;
  distance = search();

  if ((distance < 40))
  {
    RobotSpeed = 100;

    analogWrite(MotorA_PWM, RobotSpeed);
    analogWrite(MotorB_PWM, RobotSpeed);

    RobotStop();
    delay(10);

    RobotBackward();
    delay(400);

    RobotStop();
    delay(10);

    distance = search();

    int a = 250;
    int b = 250;

    if (distance < 30)
    {
      RobotRight();
      a = a + 50;
      delay(a);
      distance = search();
    }
    else
    {
      b = b + 50;
      RobotLeft();
      delay(b);

      distance = search();
    }
  }

  else if ((distance >= 40) && (distance <= 70))
  {
    RobotSpeed = 150;

    analogWrite(MotorA_PWM, RobotSpeed);
    analogWrite(MotorB_PWM, RobotSpeed);

    RobotForward();
  }
  else
  {
    RobotSpeed = 255;

    analogWrite(MotorA_PWM, RobotSpeed);
    analogWrite(MotorB_PWM, RobotSpeed);

    RobotForward();
  }
}
```

```
void RobotForward()
{
  digitalWrite(MotorA_IN1, HIGH);
  digitalWrite(MotorA_IN2, LOW);
  digitalWrite(MotorB_IN3, HIGH);
  digitalWrite(MotorB_IN4, LOW);
}

void RobotBackward()
{
  digitalWrite(MotorA_IN1, LOW);
  digitalWrite(MotorA_IN2, HIGH);
  digitalWrite(MotorB_IN3, LOW);
  digitalWrite(MotorB_IN4, HIGH);
}


void RobotLeft()
{
  digitalWrite(MotorA_IN1, LOW);
  digitalWrite(MotorA_IN2, HIGH);
  digitalWrite(MotorB_IN3, HIGH);
  digitalWrite(MotorB_IN4, LOW);
}

void RobotRight()
{
  digitalWrite(MotorA_IN1, HIGH);
  digitalWrite(MotorA_IN2, LOW);
  digitalWrite(MotorB_IN3, LOW);
  digitalWrite(MotorB_IN4, HIGH);
}

void RobotStop()
{
  digitalWrite(MotorA_IN1, LOW);
  digitalWrite(MotorA_IN2, LOW);
  digitalWrite(MotorB_IN3, LOW);
  digitalWrite(MotorB_IN4, LOW);
}
```