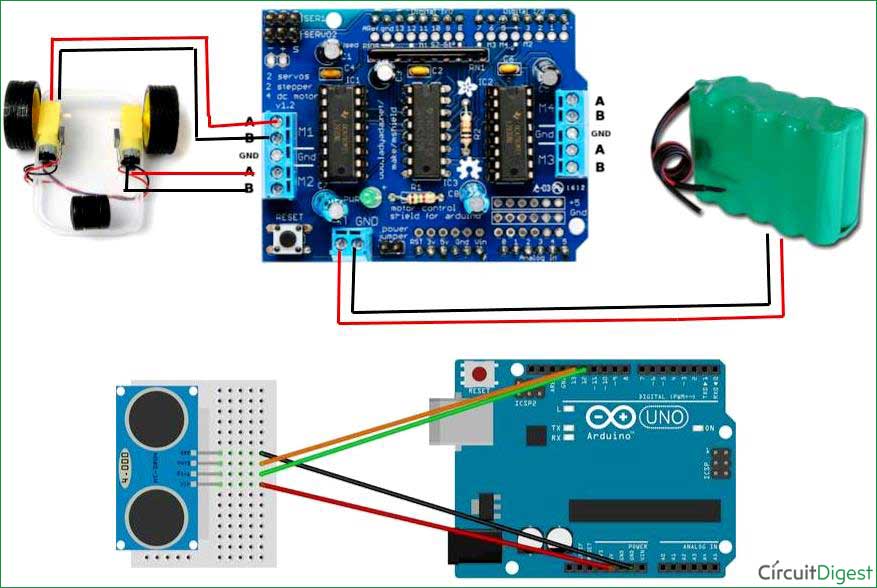
**Automatic Floor Cleaning Robot Using Arduino**

**INTRODUCTION:**

Automatic floor cleaners are nothing new, but they all share a common problem. They all are too expensive for what they do. Today, we will make a **Automatic Home cleaning Robot** that only costs a small fraction of the ones in the market. This Robot can detect the obstacles & objects in front of it and can continue moving, avoiding the obstacles, until the whole room is cleaned. It has a small brush attached to it to clean the floor.

**COMPONENTS REQUIRED**:

* Arduino UNO R3.
* Ultrasonic Sensor.
* Arduino Motor Drive.
* Wheel Drive Robot Chassis.
* Computer to Program the Arduino.
* Battery for the Motors.
* A Shoe Brush.

**CIRCUIT DIAGRAM**: 

**PROCEDURE:**

* Mount the Arduino on the chassis.. Secure the motors with the wheels and chassis using screws. Mount the shoe brush on the front of the chassis. Mount the Scotch Brite scrub pad behind the brush. Mount the batteries (or cables on the back of the chassis).
* Circuit for this **Automatic Home Cleaning Robot** is very simple. Connect the **Ultrasonic sensor** to the Arduino as mentioned below and place the **Motor Driver shield** on to the Arduino like any other shield.
* The Trig pin of Ultrasonic is connected to the A4 pin on the Arduino, the Echo pin is connected to the A5 pin, the voltage pin to the 5V pin and the Ground pin to the ground pin. The Echo pin and the Trig pin allow the Arduino to communicate with the sensor. Power is delivered to the sensor through the voltage and Ground pins, and the Trig and Echo pins allow it to send and receive data with the Arduino.
* The motor shield should have at least 2 outputs, and they should be connected to your 2 motors. Normally, these outputs are labelled “M1” and “M2” or “Motor 1” and “Motor 2”. Wire your batteries and power bank up to the motor shield and Arduino respectively. Do not cross connect them. Your motor shield should have an input channel. If you’re using wires, connect them to AC adapters.

**ArduinoCode:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  |   #include <NewPing.h>  int distance = 100;  const int motor1\_high = 13;  const int motor1\_low = 8;  const int motor2\_high = 12;  const int motor2\_low = 6 ;   #define TRIG\_PIN A4  #define MAX\_DISTANCE 200  #define ECHO\_PIN A5  NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE);  //motor1 is the left side motor and motor2 is on right side of the body  void setup() {      // put your setup code here, to run once:    pinMode(motor1\_high,OUTPUT);    pinMode(motor1\_low,OUTPUT);    pinMode(motor2\_high,OUTPUT);    pinMode(motor2\_low,OUTPUT);    distance = readPing();    delay(100);    distance = readPing();    delay(100);    distance = readPing();    delay(100);    distance = readPing();    delay(100);  }    void loop() {    // put your main code here, to run repeated  int distanceR = 0;   int distanceL =  0;   delay(40);  if(distance<=15)   {    moveStop();    delay(100);    backward();    delay(300);      distanceR = lookRight();    delay(200);    distanceL = lookLeft();    delay(200);    if(distanceR>=distanceL)    {      turn\_right();     moveStop();    }else    {      turn\_left();      moveStop();    }   }else   {    forward();   }  distance = readPing();  }  int lookRight()  {      digitalWrite(motor1\_high,1);    digitalWrite(motor1\_low,0);    digitalWrite(motor2\_high,1);    digitalWrite(motor2\_low,1);    delay(300);      int distance = readPing();      delay(100);        return distance;  }  int lookLeft()  {      digitalWrite(motor1\_high,1);    digitalWrite(motor1\_low,1);    digitalWrite(motor2\_high,0);    digitalWrite(motor2\_low,1);    delay(2000 );      int distance = readPing();      delay(100);      return distance;      delay(100);  }  int readPing() {    delay(70);    int cm = sonar.ping\_cm();    if(cm==0)    {      cm = 250;    }    return cm;  }  int turn\_left(){    digitalWrite(motor1\_high,1);    digitalWrite(motor1\_low,1);    digitalWrite(motor2\_high,0);    digitalWrite(motor2\_low,1);    delay(1000);   forward();  }  int turn\_right(){    digitalWrite(motor1\_high,1);    digitalWrite(motor1\_low,0);    digitalWrite(motor2\_high,1);    digitalWrite(motor2\_low,1);    delay(1000);    forward();    }  void forward(){    digitalWrite(motor1\_high,1);    digitalWrite(motor1\_low,0);    digitalWrite(motor2\_high,0);    digitalWrite(motor2\_low,1);  }  void backward(){    digitalWrite(motor1\_high,0);    digitalWrite(motor1\_low,1);    digitalWrite(motor2\_high,1);    digitalWrite(motor2\_low,0);    delay(1000);  }  void moveStop()  {    digitalWrite(motor1\_high,0);    digitalWrite(motor1\_low,0);    digitalWrite(motor2\_high,0);    digitalWrite(motor2\_low,0);  }  **ADVANTAGES:**   |  |  | | --- | --- | |  |  | |  |  |

* Low cost
* Saves time
* Ideal for People with Mobility Issues

**REFERENCES:**

1. [**https://circuitdigest.com**](https://circuitdigest.com)

**2.** [**https://nevonprojects.com/**](https://nevonprojects.com/)