#define trigPin A0 //Assin PIN A0 as trigPin (Connect ARDUINO UNO "A0" pin with Ultrasonic Sonar Sensor "TRIG" Pin)

#define echoPin A1 //Assin PIN A1 as echoPin (Connect ARDUINO UNO "A1" pin with Ultrasonic Sonar Sensor "ECHO" Pin)

#define MotorA\_IN1 3 //Assin PIN 3 as MotorA\_IN1 (Connect ARDUINO UNO "3" pin with L293D Motor Driver "IN1" Pin)

#define MotorA\_IN2 4 //Assin PIN 4 as MotorA\_IN2 (Connect ARDUINO UNO "4" pin with L293D Motor Driver "IN2" Pin)

#define MotorB\_IN3 5 //Assin PIN 5 as MotorB\_IN3 (Connect ARDUINO UNO "5" pin with L293D Motor Driver "IN3" Pin)

#define MotorB\_IN4 6 //Assin PIN 6 as MotorB\_IN4 (Connect ARDUINO UNO "6" pin with L293D Motor Driver "IN4" Pin)

#define MotorA\_PWM 9 //Assin PIN 9 as MotorA\_PWM (Connect ARDUINO UNO "9" pin with L293D Motor Driver "ENA" Pin)

#define MotorB\_PWM 10 //Assin PIN 10 as MotorB\_PWM (Connect ARDUINO UNO "10" pin with L293D Motor Driver "ENB" Pin)

void setup() {

pinMode(MotorA\_IN1, OUTPUT); //Declare "MotorA\_IN1" as "Output Pin".

pinMode(MotorA\_IN2, OUTPUT); //Declare "MotorA\_IN2" as "Output Pin".

pinMode(MotorB\_IN3, OUTPUT); //Declare "MotorB\_IN3" as "Output Pin".

pinMode(MotorB\_IN4, OUTPUT); //Declare "MotorB\_IN4" as "Output Pin".

pinMode(MotorA\_PWM, OUTPUT); //Declare "MotorA\_PWM" as "Output Pin".

pinMode(MotorB\_PWM, OUTPUT); //Declare "MotorA\_PWM" as "Output Pin".

pinMode(trigPin, OUTPUT); //Declare "trigPin" as "Output Pin".

pinMode(echoPin, INPUT); //Declare "echoPin" as "Input Pin".

}

int search(void)

{

float duration = 0.00; //Float type variable declaration

float CM = 0.00;

digitalWrite(trigPin, LOW); //Trig\_pin uotput as 0V (Logic Low-Level)

delayMicroseconds(2); //Delay for 2 us

//Send 10us High Pulse to Ultra-sonic Sonar Sensor "trigpin"

digitalWrite(trigPin, HIGH); //Trig-pin output as 5V (Logic High-Level)

delayMicroseconds(10); //Delay for 10 us

digitalWrite(trigPin, LOW); //Trig\_pin uotput as 0V (Logic Low-Level)

duration = pulseIn(echoPin, HIGH); //Start counting time, upto again "echoPin" back to Logical "High-Level" and puting the "time" into a variable called "duration"

CM = (duration / 58.82); //Convert distance into CM.

return CM;

}

void loop() {

float distance = 0.00;

float RobotSpeed = 0.00;

//Measuring the distance in CM

distance = search();

if ((distnace < 40)) //If obstaclw found in 40 CM.

{

RobotSpeed = 100; //Speed Down

analogWrite(MotorA\_PWM, RobotSpeed); //Update speed in MOTORA Output Terminal

analogWrite(MotorB\_PWM, RobotSpeed); //Update speed in MOTORB Output Terminal

RobotStop(); //Robot Stop

delay(10);

RobotBackward(); //Robot Run Backward Direction

delay(400);

RobotStop(); //Robot Stop

delay(10);

distance = serach(); //Check obstacle again

int a = 250;

int b = 250;

if (distance < 30) //30cm

{

RobotRight(); //Robot Turn into Right Direction

a = a + 50;

delay(a);

distance = search(); //Check obstacle again

}

else

{

b = b + 50;

RobotLeft(); //Robot Turn into Left Direction

delay(b);

distance = search(); //Check obstacle again

}

}

else if ((distance >= 40) && (distance <= 70))

{

RobotSpeed = 150; //Speed Increase Slightly

analogWrite(MotorA\_PWM, RobotSpeed); //Update speed in MOTORA Output Terminal

analogWrite(MotorB\_PWM, RobotSpeed); //Update speed in MOTORB Output Terminal

RobotBackward();

}

else

{

RobotSpeed = 255; //Speed increase to full speed

analogWrite(MotorA\_PWM, RobotSpeed); //Update speed in MOTORA Output Terminal

analogWrite(MotorB\_PWM, RobotSpeed); //Update speed in MOTORB Output Terminal

RobotForward(); //Robot Move to Forward Direction

}

}

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);

}