```
/*
 HC-SR04 Ping distance sensor:
 VCC to arduino 5v
 GND to arduino GND
 Echo to Arduino pin 7
 Trig to Arduino pin 8
 */
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13 // Onboard LED

int maximumRange = 200; // Maximum range needed 2 m
int minimumRange = 0; // Minimum range needed 0 m
long duration, distance; // Duration used to calculate distance

//h bridge
const int motor1Pin = 4;    // H-bridge leg 1 (pin 4, 1A)
const int motor2Pin = 12;    // H-bridge leg 2 (pin 12, 2A)

//PID variables
unsigned long lastTime;
double pwm_output;
double dErr;
double D=0;
double lastErr;
double error;
int Setpoint = 10; /*working distance of 10 cm */
double kp=3, ki=0, kd=2000;

void setup() {
 Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
 pinMode(3, OUTPUT);
 pinMode(motor1Pin, OUTPUT);
 pinMode(motor2Pin, OUTPUT);
  }

void loop() {
/* The following trigPin/echoPin cycle is used to determine the
 distance of the nearest object by bouncing soundwaves off of it. */
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 distance = duration/58.2; //Calculate the distance (in cm) based on
the speed of sound.

 if (distance >= maximumRange || distance <= minimumRange){
 /* Send a negative number to computer and Turn LED ON
 to indicate "out of range" */
 Serial.println("-1");
 digitalWrite(LEDPin, HIGH);
 }
 else {
 /* Send the distance to the computer using Serial protocol, and
 turn LED OFF to indicate successful reading. */
 digitalWrite(LEDPin, LOW);
```

```
 Compute(); //calls PID function
 }
 // if the car is more distant than the setpoint, motor will turn
fowards:
 if (error >=2) {
     digitalWrite(motor1Pin, LOW);   // set leg 1 of the H-bridge low
     digitalWrite(motor2Pin, HIGH);  // set leg 2 of the H-bridge high
    }
 // if car is less distant than the setpoint , motor will turn
backwards:
 else if (error <=-2) {
     digitalWrite(motor1Pin, HIGH); // set leg 1 of the H-bridge high
     digitalWrite(motor2Pin, LOW);  // set leg 2 of the H-bridge low
    }
// if car is close to the setpoint , motor will stop:
 else {
      digitalWrite(motor1Pin, LOW); // set leg 1 of the H-bridge high
      digitalWrite(motor2Pin, LOW);  // set leg 2 of the H-bridge low
    }
 //Delay 100ms before next reading.
 delay(100);
}
void Compute()//PID function
{
   /*How long since we last calculated*/
   unsigned long now = millis();
   double timeChange = (double)(now - lastTime);
   error =  distance - Setpoint;
   dErr = (error - lastErr) / timeChange;
   double P = abs(kp*error);//proportional part
   if(error >=2){
      D = kd*dErr;//derivative part
   }else{
      D = abs(kd*dErr);//derivative part
    }
   /*Compute PID Output*/
   P = constrain(P, 0, 50); //constrains P for long distance values
   pwm_output = P + D + 160;
   pwm_output = constrain(pwm_output, 0, 255);//constrains output
values to a digital pwm arduino output
   analogWrite(3, pwm_output);

   //to monitorize the variables:
   Serial.print("distance: ");
   Serial.print(distance);
   Serial.print("\t");
   Serial.print("P: ");
   Serial.print(P);
   Serial.print("\t");
   Serial.print("D: ");
   Serial.print(D);
   Serial.print("\t");
   Serial.print("Output: ");
   Serial.println(pwm_output);
   /*Remember some variables for next time*/
   lastErr = error;
   lastTime = now;
}
```