

```

//#include<Ultrasonic.h>

// CARL CODE - SRF04 based

//#include <NewPing.h>

// Pin defines for motor - outputs

#define MPIN1B 5 // motor 1 pin B
#define MPIN1A 6 // motor 1 pin A
#define MPIN2A 9 // motor 2 pin A
#define MPIN2B 10 // motor 2 pin B

// SRF04 pin definitions

#define TRIGGER_PIN A2 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN A3 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters).
Maximum sensor distance is rated at 400-500cm.
#define REDPIN 1
#define GREENPIN 3
#define BLUEPIN 4

//NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //NewPing setup of pins and
maximum distance.

//Variables
int rSpeed = 195; // robot's max speed (used in Move() calculations)
//int TS1 = 75; // turn speed 1
//int FS = 75; // forward speed: the speed at which both motors turn at
// move function - we pass a speed for each wheel m1,m2 - as a percentage 0-100

void Move(int m1, int m2) {
// for each motor, one pin is held low, the other pin is toggled - this is called
sign-magnitude drive
// the other drive method uses two pwm signals, antiphase to each other and is called
locked antiphase

if (m1 < 0) { // for reverse - not used
analogWrite(MPIN1A, rSpeed * abs(m1) / 100);
analogWrite(MPIN1B, 0);
}
else { // normal operation
analogWrite(MPIN1A, 0);
analogWrite(MPIN1B, rSpeed * m1 / 100);
}

if (m2 < 0) { // for reverse - not used
analogWrite(MPIN2A, rSpeed * abs(m2) / 100);
analogWrite(MPIN2B, 0);
}
else { // normal operation
analogWrite(MPIN2A, 0);
analogWrite(MPIN2B, rSpeed * m2 / 100);
}
}

```

```

}
}

void Stop( int delaysms){
  RGBled(REDPIN, GREENPIN, BLUEPIN, 1, 0, 0); //RED
  Move(0,0);
  delay(delaysms);
}

void Forward(intdelaysms){
  RGBled(REDPIN, GREENPIN, BLUEPIN, 0, 1, 0); //GREEN
  Move(-100, -100);
  delay(delaysms);
  Move(0,0);
}

void Reverse(intdelaysms){
  RGBled(REDPIN, GREENPIN, BLUEPIN, 0, 0, 1); //BLUE
  Move(100, 100);
  delay(delaysms);
  Move(0,0);
}

void TurnRight(intdelaysms){
  RGBled(REDPIN, GREENPIN, BLUEPIN, 1, 1, 0); //YELLOW
  Move(-100, 100);
  delay(delaysms);
  Move(0,0);
}

void TurnLeft(intdelaysms){
  RGBled(REDPIN, GREENPIN, BLUEPIN, 0, 1, 1); //CYAN
  Move(100, -100);
  delay(delaysms);
  Move(0,0);
}

void RGBled(int redPin, int greenPin, int bluePin, int redValue, int greenValue, int
blueValue){
  //pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  //digitalWrite(redPin, redValue);
  digitalWrite(greenPin, greenValue);
  digitalWrite(bluePin, blueValue);
}

void activeBuzzer(int pin, int duration_ms){
  pinMode(pin, OUTPUT);
  digitalWrite(pin, HIGH);
  delay(duration_ms);
  digitalWrite(pin, LOW);
}

```

```

void passiveBuzzer(int pin, int duration_ms) {
  pinMode(pin, OUTPUT);
  digitalWrite(pin, HIGH);
  delay(duration_ms);
  digitalWrite(pin, LOW);
}

// Setup - runs once
void setup() {
  Serial.begin(115200); // serial for serial port
  pinMode(MPIN1B, OUTPUT); // set motor pin as output
  pinMode(MPIN1A, OUTPUT); // set motor pin as output
  pinMode(MPIN2A, OUTPUT); // set motor pin as output
  pinMode(MPIN2B, OUTPUT); // set motor pin as output
  Stop(2000);
  activeBuzzer(11, 500);
  //passiveBuzzer(11, 500);
}

// main loop - runs infinitely
void loop() {
  //Serial.print("Ping: ");
  //Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result
  (0 = outside set distance range)
  //Serial.println("cm");
  //Forward(500);

  for (int i = 0; i < 4; i++) {
    Forward(700);
    Stop(400);
    Reverse(700);
    Stop(400);
    TurnLeft(225);
    Stop(400);
  }

  TurnLeft(5000);

  /*Stop(150);
  Reverse(250);
  Stop(150);
  Reverse(250);
  Stop(250);
  TurnLeft(500);
  Stop(150);
  Forward(250);
  Stop(150);
  Reverse(150);
  Stop(150);
  TurnRight(1000);
  Stop(150);
  Reverse(150);
  Reverse(150);
  */
}

```

```

Stop(150);
TurnLeft(1500);*/

activeBuzzer(11, 500);
Stop(10000);
activeBuzzer(11, 500);
//passiveBuzzer(11,500);
}

//Move(75,0);delay(500); // back to the right 90deg
//Move(0,75);delay(500); // back to the left 90deg
//Move(-75,0);delay(500); // fwd to the right 90deg
//Move(0,-75);delay(500); // fwd to the left 90deg

/*
if(sonar.ping_cm()<= 1){
Move(0,0);delay(250); // stop
Move(0,75);delay(600); Move(0,0);delay(250); // back to the left 90deg
Move(-75,-75);delay(750); Move(0,0);delay(250); // fwd for 250 secs
Move(75,0);delay(500); Move(0,0);delay(250); // back to the right 90deg
Move(-75,-75);delay(1000); Move(0,0);delay(250); // fwd for 500 secs
Move(75,0);delay(600); Move(0,0);delay(250); // back to the right
Move(-75,-75);delay(700); Move(0,0);delay(250); // fwd for 250 secs
Move(0,75);delay(750); Move(0,0);delay(250); // back to the left 90deg
Move(-75,-75);delay(300); Move(0,0);delay(250); // fwd for 500 secs
Move(0,0);delay(10000);
}

Move(-75,-75);
*/

```