

Traversing the Maze, Plan C: Triple Ultrasonic Sensor

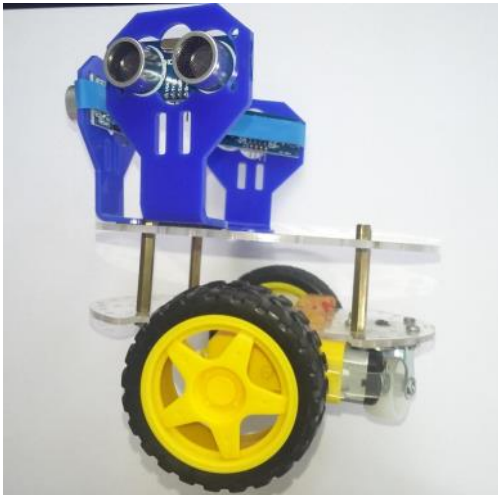
Friday, 6 March 2020 12:35 PM

EXPLORE

Source 1

I found this arduino solution with three Ultrasonic sensors:

<https://maker.pro/arduino/tutorial/how-to-build-an-arduino-based-maze-solving-robot>



Build Idea

I will use this build as a starting point:

<https://www.flcasts.com/materials/354-freedom-folk-race-tank-robot-with-treads-and-three-ultrasonic-sensors>



DEVELOP

I have decoded the arduino code from source 1 into the following algorithm:

Loop Forever

```
// get readings
distFront = front ultrasonic distance
distLeft = left ultrasonic distance
distRight = right ultrasonic distance
```

```
//clear in front, move forward
if distFront > 20
  forward 1 sec
```

```

// if wall is between 10 and 20 on the left, move forward
if distLeft > 10 and distLeft < 20
  forward 1 sec
//if move away from the wall on the left, steer gradually back
if distLeft >= 20
  spin left a bit 30ms
  forward 1 sec
// if too close to wall on left, drive right a bit
if distLeft < 10 and distLeft > 0
  spin right a bit 30ms
  forward 1 sec
//wall in front but clear to the right
if distFront <= 10 and distRight > 20
  stop 300ms
  spin right 90 degrees
//both front and right have a wall spin right 180 degrees
if distFront <= 20 and distRight < 20
  stop 1 sec
  spin right 180 degrees

```



test 1 - forward and stop at wall in front

```

forever
  set distFront to ultrasonic 2 distance
  set distLeft to ultrasonic 3 distance
  set distRight to ultrasonic 1 distance
  if distFront > 7 then
    tank motors B+C -25 % -25 % +

```

test 2

- forward and stop at wall in front
- stick to left hand wall
- turn left if clear

```

forever
  set distFront to ultrasonic 2 distance
  set distLeft to ultrasonic 3 distance
  set distRight to ultrasonic 1 distance
  if distFront > 7 then
    tank motors B+C -25 % -25 % +
  if distLeft > 5 and distLeft < 10 then
    tank motors B+C -25 % -25 % +
  if distLeft >= 6 then

```

```

set distLeft to ultrasonic 3 distance
set distRight to ultrasonic 1 distance
if distFront > 7 then
  tank motors B+C -25 % -25 % +

```

```

forever
  set distFront to ultrasonic 2 distance
  set distLeft to ultrasonic 3 distance
  set distRight to ultrasonic 1 distance
  if distFront > 7 then
    tank motors B+C -25 % -25 % +
  if distLeft > 5 and distLeft < 10 then
    tank motors B+C -25 % -25 % +
  if distLeft ≥ 6 then
    tank motors B+C 25 % -25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -
  if distLeft > 0 and distLeft ≤ 4 then
    tank motors B+C -25 % 25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -

```



lego-TripleB
uild (5)

This works OK, but is a bit jerky. A PID control would be good

- test 3 -
- forward and stop at wall in front
 - stick to left hand wall
 - turn left if clear
 - wall in front but clear to the right

```

forever
  set distFront to ultrasonic 2 distance
  set distLeft to ultrasonic 3 distance
  set distRight to ultrasonic 1 distance
  if distFront > 7 then
    tank motors B+C -25 % -25 % +
  if distLeft > 5 and distLeft < 10 then
    tank motors B+C -25 % -25 % +
  if distLeft ≥ 6 then
    tank motors B+C 25 % -25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -
  if distLeft > 0 and distLeft ≤ 4 then
    tank motors B+C -25 % 25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -
  if distFront < 8 and distRight > 10 then
    stop all motors
    tank motors B+C -25 % 25 % for 1.3 seconds -

```

Test 4 -

- forward and stop at wall in front
- stick to left hand wall
- turn left if clear
- wall in front but clear to the right
- dead end - both front and right have a wall spin right 180 degrees

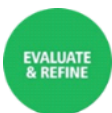
```

forever
  set distFront to ultrasonic 2 distance
  set distLeft to ultrasonic 3 distance
  set distRight to ultrasonic 1 distance
  if distFront > 9 then
    tank motors B+C -25 % -25 % +
  if distLeft > 5 and distLeft < 10 then
    tank motors B+C -25 % -25 % +
  if distLeft ≥ 6 then
    tank motors B+C 25 % -25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -
  if distLeft > 0 and distLeft ≤ 4 then
    tank motors B+C -25 % 25 % for 100 milliseconds -
    tank motors B+C -25 % -25 % for 100 milliseconds -
  if distFront < 8 and distRight > 10 then
    stop all motors
    tank motors B+C -25 % 25 % for 1.3 seconds -
  if distFront < 8 and distRight < 7 then
    stop all motors
    tank motors B+C 25 % 25 % for 300 milliseconds -
    tank motors B+C -25 % 25 % for 2.3 seconds -

```



lego-TripleB
uuld (11)



All went well when there were dead ends to the left, as this is a left hand rule algorithm. However, dead-ends were skipped when running the maze the other way and dead-ends were on the right. This actually would not be a problem if the maze is closed and the entry and exit is the same.

Dead-ends to the right

Dead-ends to the left



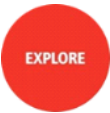
<https://youtu.be/hdD2BivU1Vc>



<https://youtu.be/V3ta7CWXeE8>

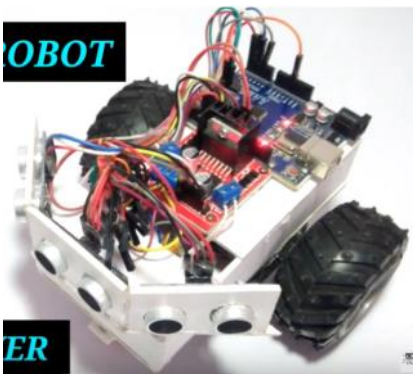
For mazes where the entry and exit are in different places, the solution would be to have a left or right program loaded up and make sure that you are traversing the maze in the correct direction.

Of all the plans, so far, this one has been the most effective.



Source 2

Here is another idea. I haven't coded or tested this out. Give it a go!



<https://www.youtube.com/watch?v=AjkUfTUXxEE&feature=youtu.be>



1st refinement

Set dis to 25

Loop forever

 Check left, front, right distances

 If wall on all 3 sides, dead end

 Turn right until 180 degrees the other way

 If wall on right and front sides

 Turn left

 If wall on left and front sides

 Turn right

 If wall on front side

 Turn right

 If wall on left side

 Turn right

 Go forward

 If wall on right side

 Turn left

 Go forward

 Else

 Go forward

2nd Refinement

```
dis = 25
Loop forever
  Check left, front, right distances

  If frontSensor < dis AND rightSensor < dis AND leftSensor < dis // wall in front of all 3 sides, dead end
    spin right 180 degrees

  Else if frontSensor < dis AND rightSensor < dis AND leftSensor > dis // wall on right and front sides
    spin left 90 degrees

  Else if frontSensor < dis AND rightSensor > dis AND leftSensor < dis // wall on left and front sides
    spin right 90 degrees

  Else if frontSensor < dis AND rightSensor > dis AND leftSensor > dis // wall on front side
    spin right 90 degrees

  Else if frontSensor > dis AND rightSensor > dis AND leftSensor < dis // wall on left side; follow wall and correct
    spin right 180 ms
    forward

  Else if frontSensor > dis AND rightSensor < dis AND leftSensor > dis //wall on right side; steer away
    spin left 180 ms
    forward

  Else
    forward
```

3rd refinement

