

# Edge of the World detection for Dummies

By: Vinh Trinh

“Edge of the world” detection is a method that will allow a robot to “sense” the edge of a table or any raised surface. It is very easy to build, cheap, easy to code, and quite reliable if done correctly. It is also a crucial feature if you plan to operate your robot on a raised surface, such as a bar or a table.

The main idea behind edge detection is light variation. The heart of the sensors which allow for edge detection are CDS (Cadmium Sulfide cells). These components vary with resistance depending on the amount of light that is present on the surface. Using this simple idea, one can setup a basic voltage divider circuit and use the voltage drop across the CDS Cell to track the intensity of light.

So having said all that, lets build some edge detectors!

Things you'll need:

- CDS Cells
- 47K Ohm Resistor
- 220 Ohm Resistor
- Ultra Bright LED (2000+ MCD, Pref. White)
- Analog to Digital Port (To take readings)
- Output Port (To control LED)
- Electrical tape (Heat shrink tubing works well also)
- Some cloth

## Step 1: The voltage divider

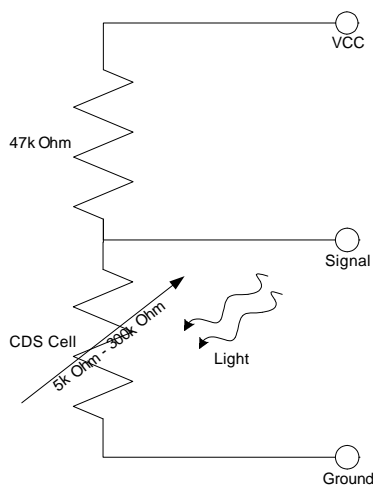


Figure 1

## Step 2: Setup a light source

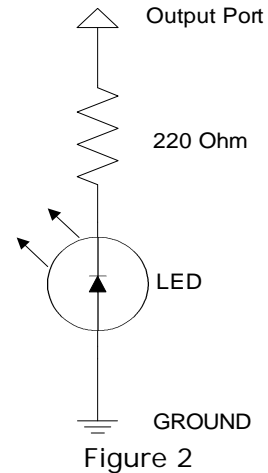


Figure 2

It is important that VCC is connected to an output port so that one can control the when the LED is on or off.

## Step 3: Assembly

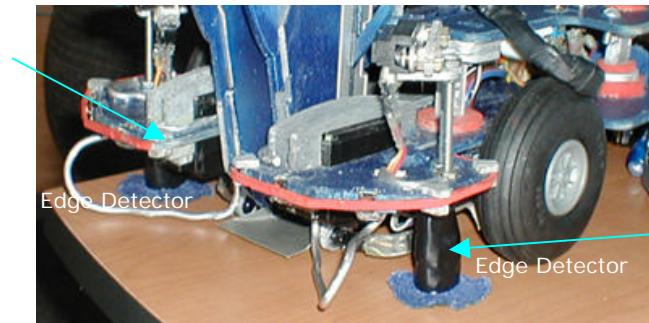


Figure 3

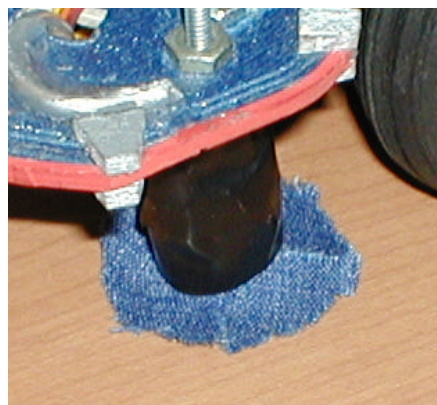


Figure 4

## Edge of the World detection for Dummies

By: Vinh Trinh

As you can see from figure 3, I have my Detectors completely columnated with electrical tape to prevent ambient light from affecting the readings. Also, I have a cloth skirt fanned out at the bottom of the columnation to prevent any ambient light to be reflected off the table into the CDS cell. I found that using a cloth skirt as opposed to making the columniation all the way to the table works much better since it does not prohibit or tamper with the robots movement. The skirt also blocks out ambient light much more effectively.

It is also a good idea to place these in front of your wheels for obvious reasons. Here's a picture of how the inside should look.

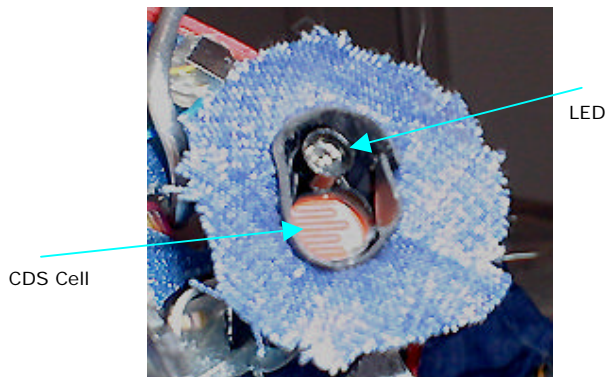


Figure 5

The LED is columnated to prevent direct illumination onto the CDS cell. As you can see from figure 5 it is also offset a little in front of the CDS cell so the beam of light will not be blocked by the cell.

### Step 5: Making them work

There are 2 main cases which one must consider when edge detecting

- 1.) Robot is in a very bright room
- 2.) Robot is in a very dark room

The sole reason why we implement edge detection with an internal light source is for the 2<sup>nd</sup> case. Without it, the robot could only reliable edge detect if the room is well lit. However, if a room IS well lit (much like the MIL lab), we want the option of turning off the LED so that we can detect for the ambient room light once our CDS cell runs over an edge.

And so having said that, we want to setup 2 scenarios.

- 1.) For dark rooms:
  - LED is ON
  - CDS normally sees LIGHT
  - CDS detects by seeing DARK
- 2.) For Bright Rooms:
  - LED is OFF
  - CDS normally sees DARK
  - CDS detects by seeing LIGHT

With this in mind, when writing software you should initiate your robot's actions accordingly.

This is some pseudo code on how I did it, it should be pretty self explanatory.

```
initialize_robot();

DARK_THRESHOLD = X;
BRIGHT_THRESHOLD = Y;

while (Bumper is not pressed) {
  if (Left Bumper)
    Robot is in a dark room
    turn CDS leds ON;
    room = 0;
  else if (Right Bumper)
    Robot is in a bright room
    room = 1;
    turn CDS leds OFF;
}

Initial_LEFT = LEFT_CDS;
Initial_Right = RIGHT_CDS;

Infinite loop {
  go forwards();
  L_reading = absoluteV(Initial_LEFT - LEFT_CDS);
  R_reading = absoluteV(Initial_RIGHT - RIGHT_CDS);

  if (
    (L_reading > DARK_THRESHOLD (and) room == 0)
    or
    (L_reading > BRIGHT_THRESHOLD (and) room == 1))
    backup and turn right();

  else if (
    (R_reading > DARK_THRESHOLD (and) room == 0)
    or
    (R_reading > BRIGHT_THRESHOLD (and) room == 1))
    backup and turn left();
}
```

Change X and Y to suit your robots needs, and be sure to press the correct bumper for the different operation conditions.

**That's all there is too it!**