NITRObot line following

A Line Follower Robot (LFR) is an automated guided vehicle (AGV), which follow a visual line embedded on the floor or ceiling. The LFR follows a line until that line exists. The line can be normal visible color or invisible magnetic field (magnetic tape) or electric field. Usually, black line on a white surface is used as a visual line, but white line on a black surface can be used as well.

Large line follower robots are usually used in industries for assisting the automated production process. They are also used in military applications, human assistance purpose, delivery services etc. They perform tasks such as transportation, manipulation of different objects and as a replacement for conveyer belts.





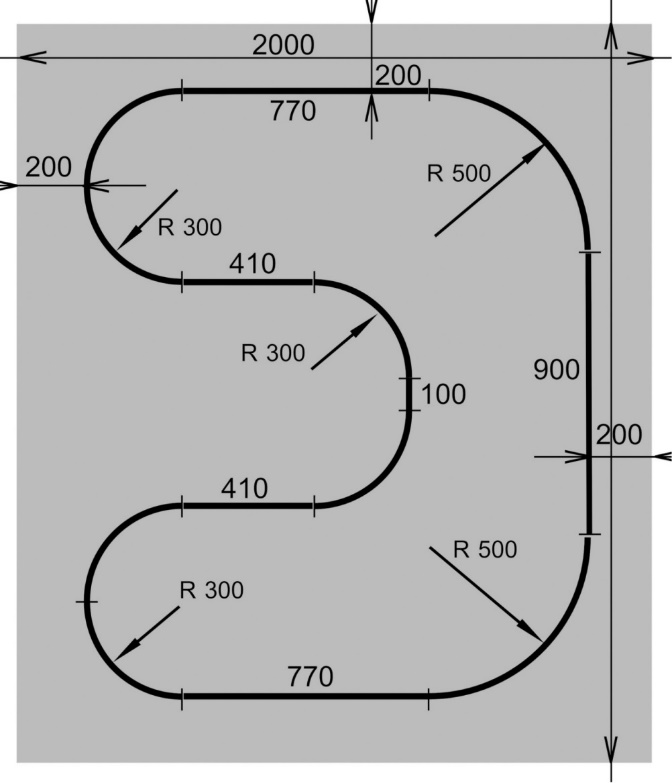
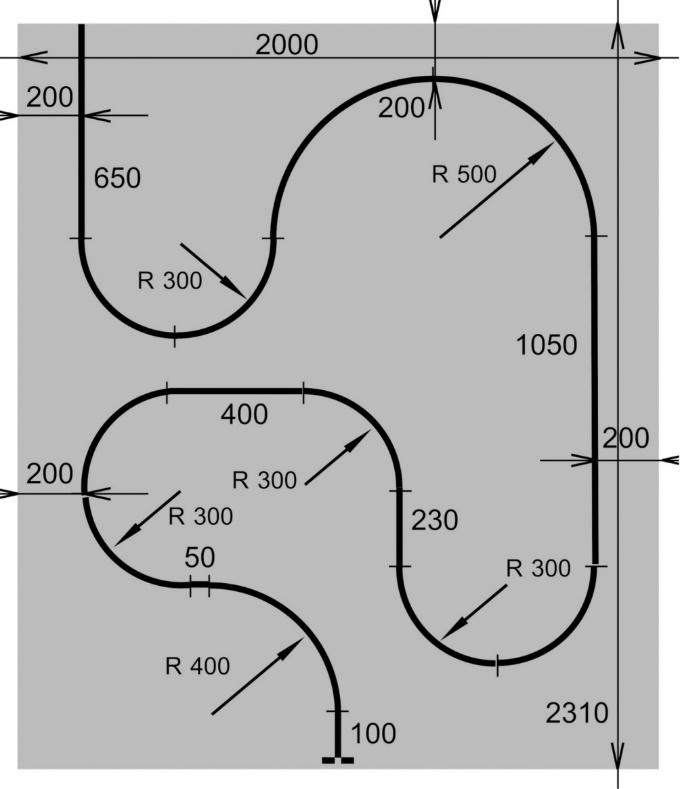
Line follower Robot is one of the first robots that beginners and students get their first robotic experience with. NITRObot is equipped with different sensors and designed to let you simulate tasks from the industrial robotics world.

In this program, we will use NITRObot's line detecting sensors to perform the line following task similarly to the way it is done in the big industrial robots.

1. **The First step Is to design our line following track.**

Use black electrical tape to create a line on a white non slippery surface. It can be cardboard, plastic or other material. The first track you create should be a closed loop type.

For best results, because of the size, speed and turn rate of NITRObot you should follow a simple rule – Turn radius should not be less than 300 mm (30 centimetres). Here are two example designs of a closed loop and open track:

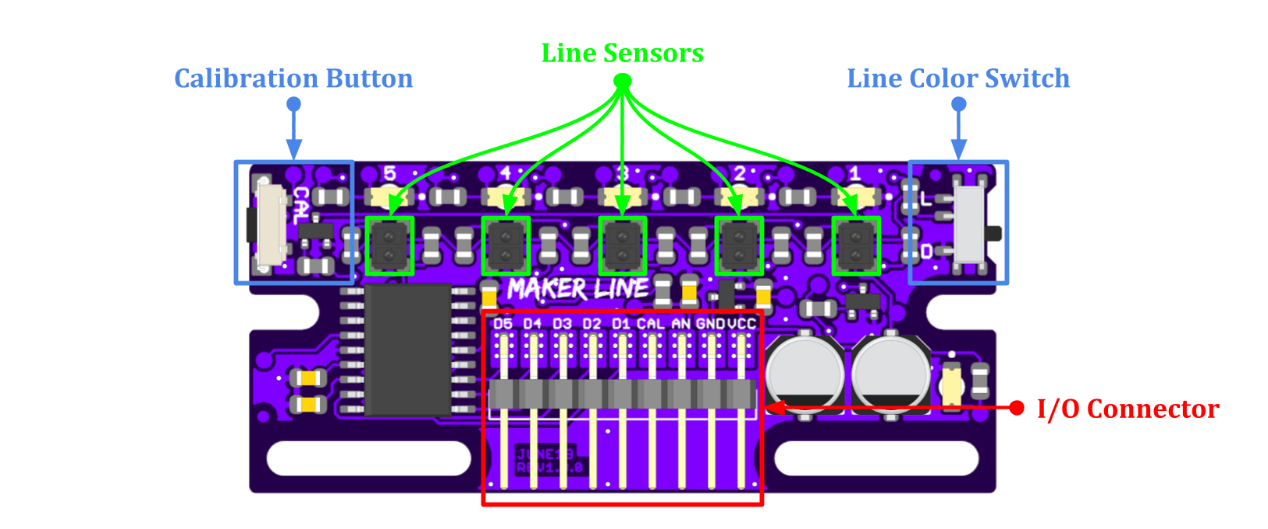


1. **The next step is to get familiar with the** **CYTRON MAKER-LINE Datasheet**

You can find the datasheet at the following link:

<https://docs.google.com/document/d/1wZwWl72CKkajTDIyiMxSQ3bwXbWbqW0eq7rdg2MUPeU/edit>

This is where the controls and the line sensors are located on the Maker-line when you look bottom-up at the NITRObot:

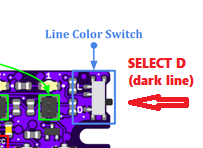


1. **NITRObot LINE SENSOR CALIBRATION**

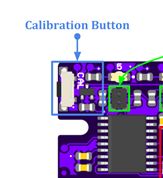
Before starting to use the MAKER-LINE, the sensors need to be calibrated so that it can differentiate between the line and the background. Calibration data will be saved in the EEPROM and it will be retained even after the MAKER-LINE is powered off.

Calibration only needs to be carried out once unless the sensor height, line or background color has changed. To perform sensor calibration, follow the steps below:

1. Upload the BareMinimum sketch from Examples->01.Basics in the Arduino IDE to the NITRObot
2. Power up the NITRObot
3. Select the line color using the slide switch.



1. Press and hold the calibrate button for more than 2 seconds. The LEDs will show the progress while waiting.



1. All 5 LEDs will start blinking after entering calibration mode. The push button can be released now.
2. Swap the MAKER-LINE across the line so that all sensors have been exposed to the line.
3. Press the calibrate button again to exit calibration mode. The LEDs will show the calibration result.
   * Running Light : Calibration is successful.
   * Fast Blink : Calibration failed. Contrast between line and background is too

low or the sensor is too high from the surface.

Now the MAKER-LINE is calibrated and is ready to use.

|  |  |
| --- | --- |
| ***Warning:*** | *Please make sure the distance of MAKER-LINE from the surface does not change during the calibration process.* |

**You can watch the CYTRON MAKER-LINE Calibration how-to video at:**

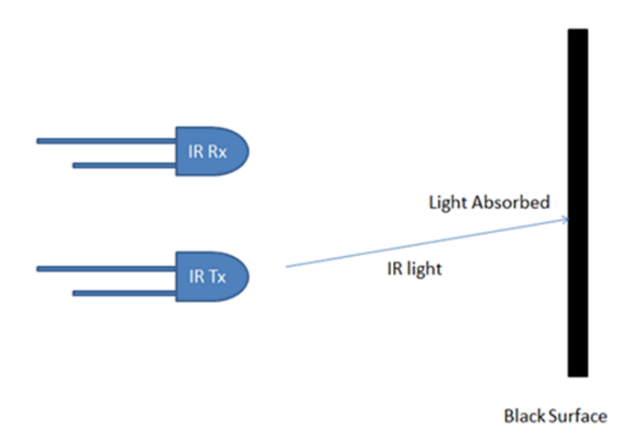
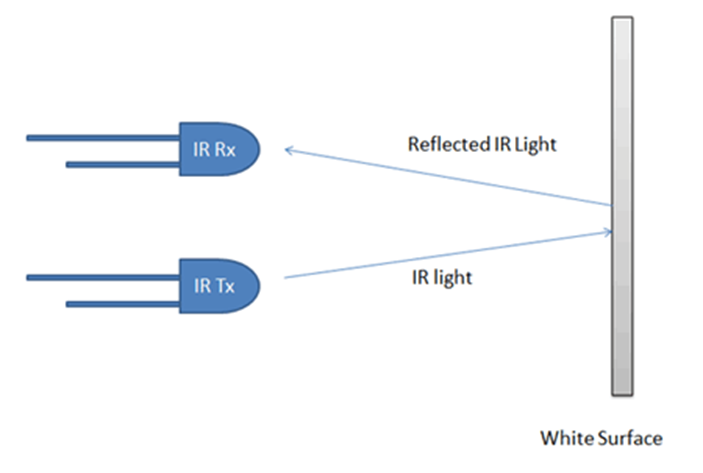
<https://youtu.be/QD0_McI0vU0> on the **NITROclubs YouTube channel**

1. **How the line following sensor works**

The concept of the line follower robot is related to light. Here, we use the behaviour of light on the black and white surface. The white colour reflects all the light that falls on it, whereas the black colour absorbs the light.

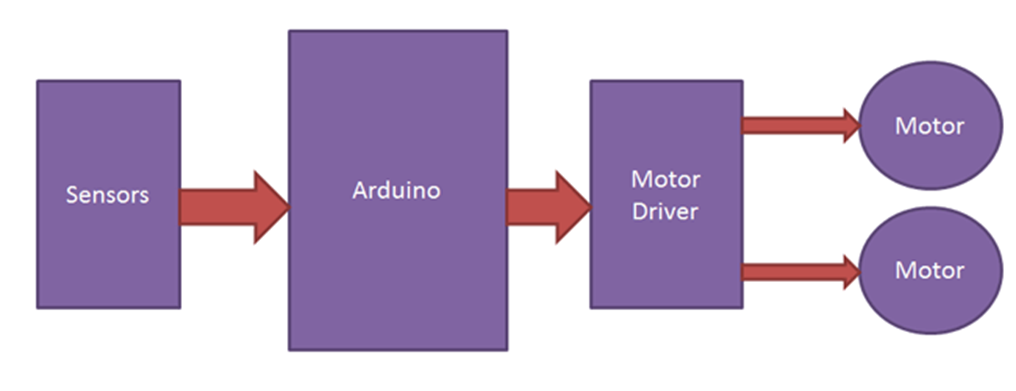
Each sensor on the CYTRON MAKER-LINE is an electronics chip which has integrated IR transmitter (LED) and receiver (photodiode). They are used to send and receive the light. When IR rays fall on a white surface, it is reflected towards IR receiver, generating some voltage changes.

When IR rays fall on a black surface, it is absorbed by the black surface, and no rays are reflected; thus, the IR receiver doesn’t receive any rays.



With the MAKER-LINE setup we did in the previous step, when a sensor senses a white surface, the Arduino Mega will get 0 (LOW) as an input on the pin, connected to the sensor, and when it senses a black line, the Arduino will get 1 (HIGH) as input.

Based on these inputs, the program should provide the proper output to the motors in order to control the robot movement.



1. **The algorithm for line following**

Let’s put down all the possible combinations of the sensor readings and what should be the required action in order for the NITRObot to follow the line.

We will do this looking at the NITRObot top-down and in the direction of movement.

The main cases are

|  |  |  |
| --- | --- | --- |
| Sensor readings combination | Robot position relative to the line | Required action |
|  |  |  |
| 0 0 0 0 0 | No line detected | Go forward to look for the line |
| 0 0 1 0 0 | Centred on the line | Go straight |
| 0 1 1 0 0 | Slight to the right | Slight turn to the left |
| 0 1 0 0 0 | More to the right | Turn mode to the left |
| 1 1 0 0 0 | Even more to the right | Turn even more to the left |
| 1 0 0 0 0 | Almost out of the line (to the right) | Aggressive turn to the left |
| 0 0 1 1 0 | Slight to the left | Slight turn to the right |
| 0 0 0 1 0 | More to the left | Turn mode to the right |
| 0 0 0 1 1 | Even more to the left | Turn even more to the right |
| 0 0 0 0 1 | Almost out of the line (to the left) | Aggressive turn to the right |
| 0 0 0 0 0 | No line detected | Go forward and look for a line to follow |

***Arduino C++ Language REFRESHER:***

*if [Control Structure]:* [*https://www.arduino.cc/reference/tr/language/structure/control-structure/if/*](https://www.arduino.cc/reference/tr/language/structure/control-structure/if/)

*else [Control Structure]:* [*https://www.arduino.cc/reference/tr/language/structure/control-structure/else/*](https://www.arduino.cc/reference/tr/language/structure/control-structure/else/)

*switch…case [Control Structure]:* [*https://www.arduino.cc/reference/tr/language/structure/control-structure/switchcase/*](https://www.arduino.cc/reference/tr/language/structure/control-structure/switchcase/)

You can see how this is realized in the code – first with a cascade of if…else statements we check which condition (combination) is met and set **robotPosition** variable to a particular number representing the case.

***REFRESHER:*** NITRObot is a differential drive steering type of vehicle – you slow down one side (and eventually accelerate the other side) in order to make a turn. The wheels on both sides are slipping while making the turn.

Then using switch...case based on the **robotPosition** value, we apply different speed reducing coefficients to the motors/wheels closer to the line (**currentLeftSpeed**) and increase the speed of the motors/wheels on the side away from the line (**currentRightSpeed**) in order to make a turn towards the line. The bigger the difference between the two speeds, the sharper the turn towards the line is.

For example:



***NOTE:***

*DefaultLeftSpeed \* .2;* in the code is a shorthand for: *DefaultLeftSpeed \* 0.2;*

There are more cases not listed in the table above, we will use them to detect two possible finish line types:

1. Line perpendicular to the line the robot follows

CASE:

1 1 1 1 1

1. Two black markers (pieces of electrical tape) attached on the left and right side of the line

CASE:

0 1 1 1 0

1. Two black markers (pieces of electrical tape) located on the left and right side of the line with one tape width white space between them and the line. Here we check for more than one condition(case) as the robot might detect the markers while turning to the line from left or right direction at different angle.

CASES:

1 1 0 1 1

1 0 0 1 1

1 1 0 0 1

1 1 1 0 1

1. **Try NITRObot\_Line\_following\_5\_sensors.ino line following program.**

* **Replace the default left and right speed values (of 100) with the ones you obtained with the one obtained from the calibration of your robot using** **NITRObot\_motor\_calibration.ino:**

****

* **Upload NITRObot\_Line\_following\_5\_sensors.ino to the robot**
* **Place the NITRObot on the line**

***NOTE:***

*There is a delay set to 1 second inside void setup() {….}, which will delay the start of the robot in order for the robot not to jump out of your hands when you turn on the switch. You can change the delay time or remove the delay if you wish.*

* **Switch on the robot**
* **ENJOY!**
* **If there is a finish marker on the track (line) the robot should stop after reaching the marker, if no marker is presented, NITRObot will follow the line until switched off.**

1. **Tunning**

In case your NITRObot is too fast and it loses the line by overturning towards the line, make a new calibration with lower robot speed using **NITRObot\_motor\_calibration.ino,** while ensuring the robot is moving straight. Then use the values you’ve obtained from the new calibration as the new default speed values in **NITRObot\_Line\_following\_5\_sensors.ino.**

***Do this until NITRObot starts to successfully flow the line.***

1. **Advanced tunning**

**In order to achieve the maximum speed of line following, experiment with a making the default speeds higher and try changing the motor speed and reduction (slow down) coefficients in the switch...case cases.**