***Introduction to Robotic Systems Course***

**LAB 4**

**Line Following**

Contents

[1 Introduction 1](#_Toc30055509)

[1.1 Lab Overview 1](#_Toc30055510)

[2 Requirements 1](#_Toc30055511)

[3 TCRT5000 IR Reflector Sensor 1](#_Toc30055512)

[4 Task: Mount and Connect IR Sensors 2](#_Toc30055513)

[5 Task: Set Up Dynamixel Motors 2](#_Toc30055514)

[6 Task: Write the Program for Line Following 3](#_Toc30055515)

[6.1 Create Line Following Program 3](#_Toc30055516)

# Introduction

## Lab Overview

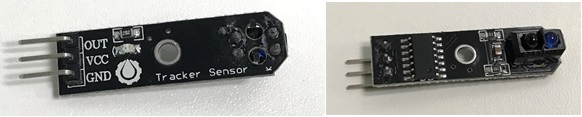
In this lab, we will program the robot to do line following. To do this, the robot will sense the line and follow the pattern of the line by applying suitable speed controls to its wheels. Four low-cost infrared (IR) reflector sensors each with digital outputs will sense the line and provide inputs to the micro-controller.

Four modes of control will be applied to the motors using the data received from the sensors. These modes are: straight, low, mid, and hard. We will apply these controls as follows: low-left, mid-right, hard-left, etc., with low, mid, and hard referring to the degree of turn.

# Requirements

The following hardware and software are required to complete this lab:

* **Hardware:** 
  + TurtleBot 3 Burger (robot).
  + Four TCRT5000 IR reflector sensors (shown in Figure 1).
* **Software:** Arduino IDE.

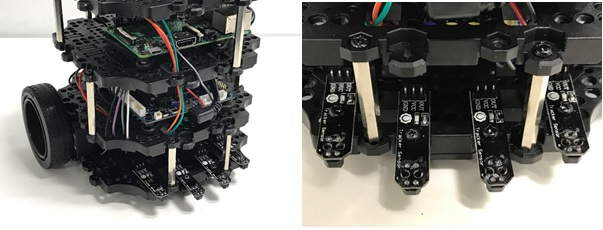
*Figure 1: TCRT5000 IR reflector sensor.*

# TCRT5000 IR Reflector Sensor

The IR sensor works by emitting and detecting IR signals reflected from a surface. The surface color determines the strength of the reflected IR. For this lab, it is best to use a white or black line of about 2–3 cm in width on a black or white surface, respectively.

# Task: Mount and Connect IR Sensors

Figure 2 shows the four IR sensors mounted at the back of the robot. Each IR sensor has three pins (GND, VCC, and OUT).



A2

A0

A1

A3

*Figure 2: IR sensors mounted at the back of the TurtleBot.*

Connect the power pins of the IR sensors to the 3.3 V, 5 V, and GND outputs on both the OpenCR1.0 Arduino connector and power outputs in the TurtleBot PC (Raspberry Pi).

Connect the “OUT” pins of the IR sensors to the analog pins, A0, A1, A2, and A3 on the Arduino connector on the OpenCR1.0 board. Looking from behind TurtleBot3 as depicted in Figure 2, the connections we have chosen for the IR sensors from left to right are: A2, A0, A1, A3.

# Task: Set Up Dynamixel Motors

If this is the first time you use the robot, you may need to set up both dynamixel motors by running the **turtlebot3\_setup\_motor** program and following the instructions on the serial monitor. To do this, follow the steps shown below.

* Open Arduino IDE and go to **File** -> **Examples** -> **TurtleBot 3** -> **turtlebot3\_setup** -> **turtlebot3\_setup\_motor** to open the turtlebot setup motor program.
* Next, put the OpenCR1.0 board in bootloader mode (steps 1 to 7 in the section “Enter Bootloader mode” in the Getting Started Guide) and burn the bootloader. Next, upload the turtlebot3 setup motor program to the board. The terminal at the bottom of the Arduino window will display **Done uploading** and **jump\_to\_fw** when upload is successful.
* Disconnect one of the Dynamixel motors from the board
* Next, open a serial monitor by clicking on the Serial Monitor icon at the top right of your Arduino IDE window.
  + On the Serial Monitor window, type “1” in the input textbox and press **Enter**.
  + When asked “Do you really want to setup? y/n:” type “y” in the input textbox and press **Enter**.
  + Ensure that the message “Setup Motor Left….. ok” is displayed.
  + Type 3 in the input textbox and press **Enter**. You will see the connected motor rotating.
  + Type “exit” in the input textbox and then repeat the last four steps, but this time, the inputs should be 2 and 4, respectively. You will see the connected motor rotating in the opposite direction.
* Reconnect the first dynamixel motor disconnected and then disconnect the one not initially disconnected and repeat the step above.

# Task: Write the Program for Line Following

In this task, you will write a program to follow a black line on a white background. We have provided the code for you. Copy and paste the code on your Arduino sketch and click **verify** after which you can upload it to the board.

## Create Line Following Program

The following steps will guide you through creating the line following program.

* Start a new Arduino project
* Ensure that the board settings are correct:
  + **File** -> **Preferences** -> **Additional Boards Manager URLs**.
  + **Tools** -> **(Boards, Ports, and Programmer)**.
* Enter the code shown below on the sketch, save, verify, and upload to the board

1. #include <DynamixelWorkbench.h> //link the header file of the required library
2. #define DXL\_BUS\_SERIAL4 "/dev/ttyACM0" //give the name of the connected port and
3. #define BAUDRATE 1000000 //identification of baudrate and dynamixels
4. #define DXL\_ID 1
5. #define DXL\_ID 2
6. DynamixelWorkbench dxl\_wb;
7. void setup() // the variables are initialized by their definition
8. {
9. **Serial**.begin(57600);
10. dxl\_wb.begin(DXL\_BUS\_SERIAL4, 1000000);
11. dxl\_wb.ping(1);
12. dxl\_wb.ping(2);
13. dxl\_wb.wheelMode(1); //starts dynamixels
14. dxl\_wb.wheelMode(2);
15. }
16. void loop()
17. {
18. **Serial**.print("\tA0 = "); //for analyzing, serial monitor outputs the reads
19. **Serial**.print(analogRead(A0));
20. **Serial**.print(" ");
21. **Serial**.print("\tA1 = ");
22. **Serial**.print(analogRead(A1));
23. **Serial**.print(" ");
24. **Serial**.print("\tA2 = ");
25. **Serial**.print(analogRead(A2));
26. **Serial**.print(" ");
27. **Serial**.print("\tA3 = ");
28. **Serial**.print(analogRead(A3));
29. **Serial**.println(" ");
30. dxl\_wb.goalSpeed(1, -50); //Sets the constant speed for robot when not on line
31. dxl\_wb.goalSpeed(2, -50); //You can set it to 0 if you want it to stop
32. lowleft(); //The loops for the required velocity modes
33. lowright(); //All of them run virtually at the same time
34. midleft();
35. midright();
36. hardleft();
37. hardright();
38. straight();
39. }
40. void straight() //compares the 10-bit unsigned integer from the analog ports, 700 is used for assurance as
41. //sometimes the decimal value drops from 1023 to 850, but it should vary
42. { // between 0 and 10 when sensor is not on a line to around 1000 when sensor is on a line
43. if (analogRead(A0) < 700 && analogRead(A1) < 700 && analogRead(A2) > 700 && analogRead(A3) > 700){
44. **Serial**.println("\tStraight");
45. **Serial**.println(" ");
46. dxl\_wb.goalSpeed(1, -50); //if true, sets the speed of left and right dynamixels
47. dxl\_wb.goalSpeed(2, -50);
48. delay(50); //delay should be 50 ms to read meaningful data
49. }
50. else {
51. } //if false, does nothing, the other loops redirect to
52. } //this loop for straight movement
53. void lowleft()
54. {
55. if (analogRead(A0) > 700 && analogRead(A1) < 700 && analogRead(A2) > 700 && analogRead(A3) > 700){
56. **Serial**.println("\tLow Left");
57. **Serial**.println(" ");
58. dxl\_wb.goalSpeed(1, -70);
59. dxl\_wb.goalSpeed(2, -40);
60. delay(50);
61. }
62. else {
63. straight();
64. }
65. }
66. void lowright()
67. {
68. if (analogRead(A0) < 700 && analogRead(A1) > 700 && analogRead(A2) > 700 && analogRead(A3) > 700){
69. **Serial**.println("\tLow Right");
70. **Serial**.println(" ");
71. dxl\_wb.goalSpeed(1, -40);
72. dxl\_wb.goalSpeed(2, -70);
73. delay(50);
74. }
75. else {
76. straight();
77. }
78. }
79. void midleft()
80. {
81. if (analogRead(A1) < 700 && analogRead(A3) < 700 && analogRead(A0) > 700 && analogRead(A2) > 700){
82. **Serial**.println("\tMid Left");
83. **Serial**.println(" ");
84. dxl\_wb.goalSpeed(1, -90);
85. dxl\_wb.goalSpeed(2, -30);
86. delay(50);
87. }
88. else {
89. straight();
90. }
91. }
92. ­void midright()
93. {
94. if (analogRead(A0) < 700 && analogRead(A2) < 700 && analogRead(A1) > 700 && analogRead(A3) > 700){
95. **Serial**.println("\tMid Right");
96. **Serial**.println(" ");
97. dxl\_wb.goalSpeed(1, -30);
98. dxl\_wb.goalSpeed(2, -90);
99. delay(50);
100. }
101. else {
102. straight();
103. }
104. }
105. void hardleft()
106. {
107. if (analogRead(A3) < 700 && analogRead(A0) > 700 && analogRead(A1) > 700 && analogRead(A2) > 700){
108. **Serial**.println("\tHard Left");
109. **Serial**.println(" ");
110. dxl\_wb.goalSpeed(1, -100);
111. dxl\_wb.goalSpeed(2, 0);
112. delay(50);
113. }
114. else {
115. straight();
116. }
117. }
118. void hardright()
119. {
120. if (analogRead(A2) < 700 && analogRead(A3) > 700 && analogRead(A0) > 700 && analogRead(A1) > 700){
121. **Serial**.println("\tHard Right");
122. **Serial**.println(" ");
123. dxl\_wb.goalSpeed(1, 0);
124. dxl\_wb.goalSpeed(2, -100);
125. delay(50);
126. }
127. else {
128. straight();
129. }
130. }