
Introduction:
In this guide, the Ranger Bot will be programmed to move forward until it senses a black line. Upon sensing the black line, it will pause and then turn until it is off the line. It will then follow the line by continuously turning left and right. This guide is for use with the ROBOTC® programming language.

Getting Started:
1. To start, open the program from Line Follower Extension Part 2. It should look similar to the code given below. Beside the variables, there are two slashes (//) followed by what that variable is representing. These mark a comment. Notice that comments appear in green when code is typed into ROBOTC.
   
   **Note:** A comment is not part of the code that the NXT brick will execute, but it helps the programmer by allowing the addition of notes to describe what has been done.

Line Follower Extension Part 2 Code:

```c
#define LINE_FOLLOWER_THRESHOLD 50
#define MOTOR_POWER 15
#define MOTOR_POWER_REVERSE -15

int motorD = 0;
int motorE = 0;
int lightSensorValue = SensorValue(LightSensor);

void main()
{
    // Motor Power
    // light sensor threshold
    //
    while(lightSensorValue < LINE_FOLLOWER_THRESHOLD)
    {
        motor[motorD] = MOTOR_POWER;
        motor[motorE] = MOTOR_POWER;
        wait1Msec(500);
        lightSensorValue = SensorValue(LightSensor);
    }
}
```

2. At the end of the previous code, before the final closing brace, add a **While Loop** to make the containing code execute forever. Follow this with an **opening brace.**

```c
while (true)
{
    // Code will be executed forever
}
```

3. Add a **conditional statement** which compares the light sensor's value to the threshold value of 50. When the light sensor's value is less than 50, the robot is sensing the black line. Follow this statement with an **opening brace**.

```c
if(SensorValue(LightSensor) < threshold)
{
```

4. Type the **Motor Power** function and set `motorD` to a power of 30 using the `mValue` variable.

```c
motor[motorD] = mValue;
```

5. Type the **Motor Power** function and set `motorE` to a power of 0.

```c
motor[motorE] = 0;
```

6. Add a **closing brace** to end the conditional statement and add an “**else**” condition for when the robot is not sensing the black line. This code will only execute if the light sensor’s value is greater than or equal to the threshold value. Follow this with an **opening brace**.

```c
} else
{
```

7. Type the **Motor Power** function and set `motorD` to a power of 0.

```c
motor[motorD] = 0;
```

8. Type the **Motor Power** function and set `motorE` to a power of 30 using the `mValue` variable.

```c
motor[motorE] = mValue;
```

**Ending the Program:**

9. Add a **closing brace** to end the “else” condition.

```c
}
```

10. Add a **closing brace** to end the While Loop.

```c
}
```
Completed Code:

```c
task main()
{
    int mValue = 30;  // Motor power
    int dValue = 15;
    int threshold = 50;  // Light sensor threshold

    while(SensorValue(LightSensor) > threshold)  // while light sensor is seeing "white"
    {
        motor[motorD] = dValue;  // Motor D is run at a 15 power level
        motor[motorE] = dValue;  // Motor E is run at a 15 power level
    }

    motor[motorD] = 0;  // Motor D is run at a 0 power level
    motor[motorE] = 0;  // Motor E is run at a 0 power level
    wait1Msec(500);  // Wait 0.5 seconds

    while(SensorValue(LightSensor) < threshold)  // while light sensor is seeing "black"
    {
        motor[motorD] = mValue;  // Motor D is run at a 30 power level
        motor[motorE] = mValue*(-1);  // Motor E is run at a 30 power level in the reverse direction
    }

    motor[motorD] = 0;  // Motor D is run at a 0 power level
    motor[motorE] = 0;  // Motor E is run at a 0 power level

    while(true)
    {
        if(SensorValue[LightSensor] < threshold)  // If the Light Sensor reads a value less than 50
        {
            motor[motorD] = mValue;  // Motor D is run at a 30 power level
            motor[motorE] = 0;  // Motor E is run at a 0 power level
        }
        else  // If the Light Sensor reads a value greater than or equal to 50
        {
            motor[motorE] = mValue;  // Motor E is run at a 30 power level
            motor[motorD] = 0;  // Motor D is run at a 0 power level
        }
    }
}
```