Simple Line Following (What’s My Line?) Arduino

Time required: 60 minutes

Please read all the directions carefully before beginning the assignment.

1. Comment your code as shown in the tutorials and other code examples.
2. Follow all directions carefully and accurately.
3. Think of the directions as minimum requirements.

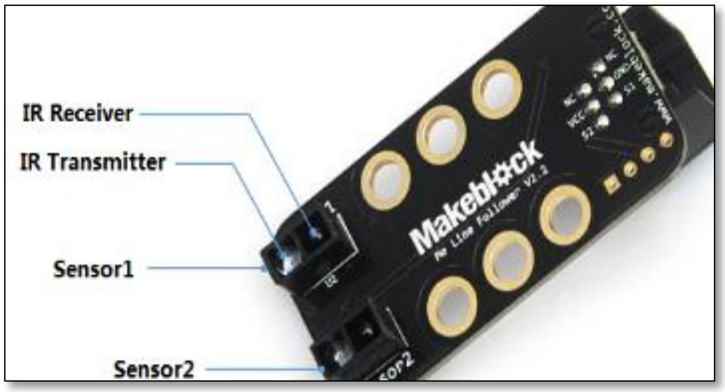
Understanding

Demonstrate understanding of:

line-follower sensor, if then else

Principles of the Line-follower Sensor

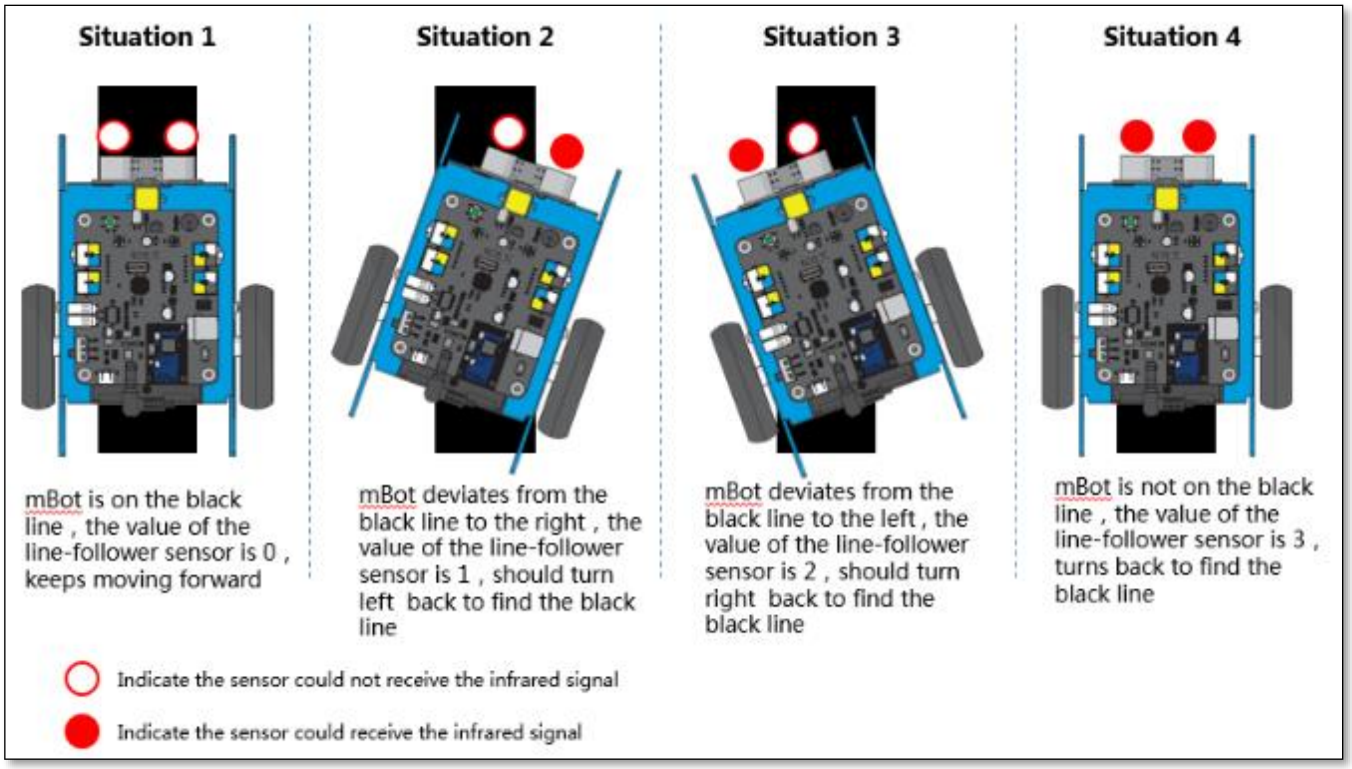
The line-follower sensor is below the robot (see the attached diagram), which consists of two sensors, Sensor 1 and 2, each consisting of an infrared emitter and an infrared receiver (see the attached diagram). As it is often used to keep the robot moving straight, it is called a line-follower sensor. Its detection range is 1 to 2 cm.



The infrared emitter continually emits infrared light during the mBot moving:

- If the infrared light is reflected (encountering white or other light color surfaces), the receiver receives the infrared signal and output the value 1 (now you can see the blue LED on the back of the line-follower sensor is lighted);
- If the infrared light is absorbed or cannot be reflected, the receiver will not receive the infrared signal but output the value 0.

The mBot line-follower sensors can detect a white line on a black surface, or a black line on a white surface.


Situation 1
Situation 2
Situation 3
Situation 4
mBot is on the black line , the value of the line-follower sensor is 0 , keeps moving forward
mBot deviates from the black line to the right , the value of the line-follower sensor is 1 , should turn left back to find the black line
mBot deviates from the black line to the left , the value of the line-follower sensor is 2 , should turn right back to find the black line
mBot is not on the black line , the value of the line-follower sensor is 3 , turns back to find the black line
Indicate the sensor could not receive the infrared signal
Indicate the sensor could receive the infrared signal

- **Situation 1:** Line follower = 0. Both sensors detect a line indicated by both blue lights shutting off.
- **Situation 2:** Line follower = 1. The right sensor no longer detects a line indicated by the right blue light turning on. In order to get the mBot back on the line, therefore, we turn the mBot left until both sensors are activated and the mBot continues moving forward.
- **Situation 3:** Line follower = 2. The left sensor no longer detects a line indicated by the left blue light turning on. So we turn the mBot right until both sensors are activated and the mBot continues moving forward again.
- **Situation 4:** Line follower = 3. Both sensors no longer detect a line. Run backward until the robot detects a line.

Sensor Position	**Value**
Both sensors over the line	0
Right sensor off line	1
Left sensor off line	2

Both sensors off line	3

Line Follower Track: A line follower track can be made with foam board and black tape. Automotive cloth wiring harness, electrical tape duct tape works well.

Relational Operators

Relational operators test for true or false by comparing one value to another. In this program we will compare the distance the sensor detects to the distance that we have set.

Operator	Interpretation	Examples	Result
>	Greater than	10 > 9 10 > 10	true false
>=	Greater than or equal to	10 >= 10 10 >= 11	true false
<	Less than	9 < 10 10 < 10	true false
<=	Less than or equal to	10 <= 10 10 <=-9	true true
==	Equal to	9 == 9	true
!=	Not equal to	9 != 9	false

Simple Line Following

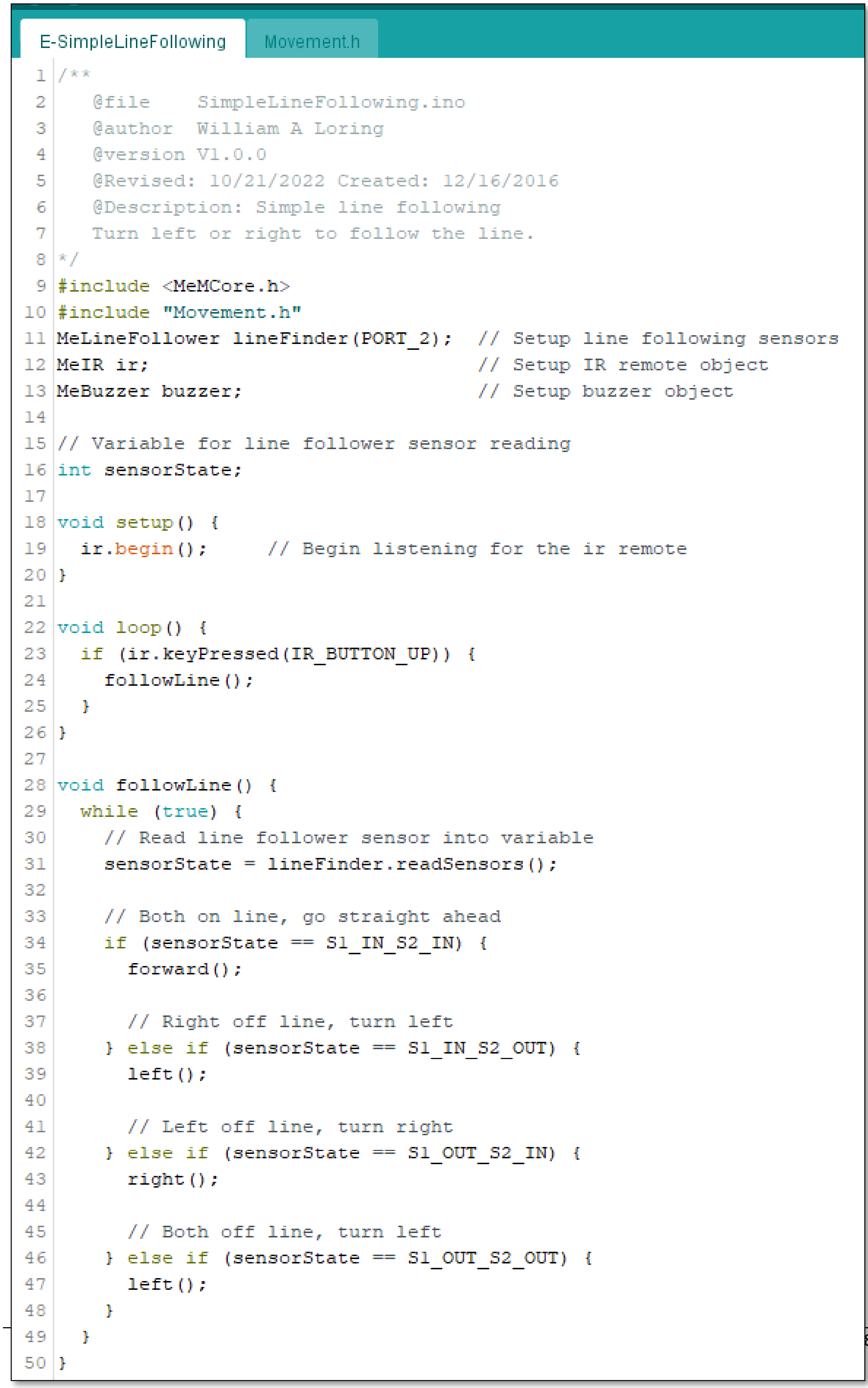
This is a sketch shows how the line following sensor works with the mBot in Arduino C. Four possible states of the line sensor provides five different motor responses. You will modify and use **Movement.h** to control the robot.

Left Sensor	**Right Sensor**	**Sensor Reading**	**Motor Response**
In	In	S1_IN_S2_IN Both on line	Go Straight
In	Out	S1_IN_S2_OUT Right off line	Left turn
Out	In	S1_OUT_S2_IN Left off line	Right turn

Out	Out	S1_OUT_S2_OUT Both off line	(If previously left turn) Left Turn
			(If previously right turn) Right Turn

Tutorial Assignment

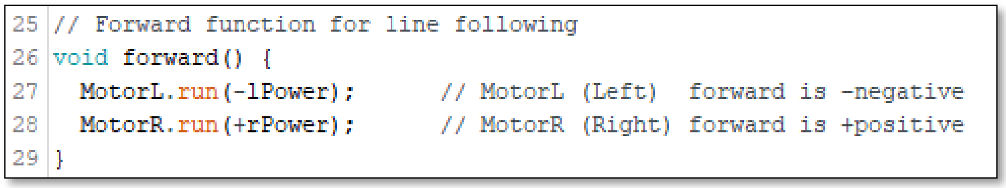
1. Start the Arduino IDE. Save the sketch as **SimpleLineFollowing**.
2. Copy the file **Movement.h** into the sketch folder. **Movement.h** will need to be modified as shown below.
3. Complete and test the program as pictured.



```
/**
   @file    SimpleLineFollowing.ino
   @author  William A Loring
   @version V1.0.0
   @Revised: 10/21/2022 Created: 12/16/2016
   @Description: Simple line following
   Turn left or right to follow the line.
*/
#include <MeMCore.h>
#include "Movement.h"
MeLineFollower lineFinder(PORT_2);  // Setup line following sensors
MeIR ir;                            // Setup IR remote object
MeBuzzer buzzer;                    // Setup buzzer object

// Variable for line follower sensor reading
int sensorState;

void setup() {
  ir.begin();      // Begin listening for the ir remote
}

void loop() {
  if (ir.keyPressed(IR_BUTTON_UP)) {
    followLine();
  }
}

void followLine() {
  while (true) {
    // Read line follower sensor into variable
    sensorState = lineFinder.readSensors();

    // Both on line, go straight ahead
    if (sensorState == S1_IN_S2_IN) {
      forward();

      // Right off line, turn left
    } else if (sensorState == S1_IN_S2_OUT) {
      left();

      // Left off line, turn right
    } else if (sensorState == S1_OUT_S2_IN) {
      right();

      // Both off line, turn left
    } else if (sensorState == S1_OUT_S2_OUT) {
      left();
    }
  }
}
```

Movement.h Modifications

Add forward, reverse, left and right functions to your **Movement.h** file. A forward function is given to start with. Add reverse, left and right functions.



```
// Forward function for line following
void forward() {
  MotorL.run(-lPower);      // MotorL (Left)  forward is -negative
  MotorR.run(+rPower);      // MotorR (Right) forward is +positive
}
```

Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.
- **In class assignment submission** → Demonstrate in person.
- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.