LFS-1.3-Changes

Ron Grant – October 10, 2020

I have not tested instructions for updating an existing sketch look for later release of this doc.

Recommended: Start up LFS_SimpleBot sketch located under File>Examples.
Verify Robot Name that appears is SimpleBot3. This indicates you are running example built for LineFollowerSim 1.3.0 or later.

Demo Viewports

The Tab key now toggles between a view with course and robot view and the sensor view.

The sensor view now has a variable position, and has been moved down away from that top status bar.

The variable courseTop has remained, but now really means course visible. If you right click on an instance of the variable you can "Show usage" which is handy in locating all of the conditional code realated to which view is visible.

This variable also selects the correct transformations to be applied for display of information on robot viewport or sensor viewport.

Sensor Enhancements

Sensors are now displayed on Robot and Sensor views.

As of now, Sensor areas are not scaled properly to appear in robot view, but do appear correctly scaled in Sensor view (the larger view);

Using mouse to hover over sensors will result in sensor being identified by name at the bottom of the viewport. LineSensors will have the current cell (spot) index # displayed as well. In addition the current intensity measured by the given spot is displayed 0.0 to 1.0.

Marker Support

New to LFS 1.3 is support of interactive start location markers which are displayed as magenta circles. Start locations are stored X,Y position coordinates with heading.

Markers may be placed whenever the robot is not running a contest. That is, after issuing R)un command you will not be able to place a marker.

To Place a Marker, move (if needed) the robot to a location, by clicking on course outside any existing marker circles and/or using mouse commands to position the robot.

Hold down Reft mouse button and move mouse to drag robot

Hold down Right mouse button to rotate robot to desired heading.

Press M (Marker command), a magenta circle appears, and the robot location (and heading) is recorded.

At any time left clicking on a marker will position the robot at the location (within 0.1 inches) with recorded heading (within 1 degree). Also, at this time pressing M will erase the marker.

Also, dragging robot to location near the center of a marker then pressing M will erase the marker.

Markers are automatically loaded/saved in data folder in file with same name as course file, but with .mrk extension.

Changes to Existing User Program

First,I would recommend saving your sketch to a new name.

Make sure your sketches are not in examples folder. If so, make a copy of that folder.

Download new version of library.

Make sure your sketch runs OK.

Load from Examples LFS_SimpleBot. Make sure header mentions LFS_SimpleBot3

Assuming you have not modified main tab, replace your main tab with LFS_SimpleBot3's code.

Keeping both sketches open you can copy lines from SimpleBot3 into your sketch.

Below lines are show from SimpleBot3 with comments removed.

When you copy lines, copy the comments too.

From SimpleBot3 UserDraw userDraw method copy to lines and paste into the beginning your userDraw method .

```
 if (courseTop) lfs.setupUserDraw();
 else lfs.setupUserDrawSensorViewport();
```

Remove original lfs.setupUserDraw();

In UserDrawPanel tab, copy and paste the following two lines to start of userDrawPanel method

```
   lfs.showSensors((courseTop)?'R':'S');
```

```
    if (courseTop) lfs.markerDraw();
```

Copy and paste the following two lines in UserInit tab, userInit method
to the end of your userInit method

```
  nameSensorsUsingVariableNames();
  lfs.markerSetup();
```

In UserKey copy and paste the mouseClicked() method into your code

Copy and paste the following line to your code just before the if(key=='C') statement.

```
 if (key == 'M') lfs.markerAddRemove();
```

In UserReset copy and paste the following line into your UserReset just after call to userInit.

```
lfs.moveToStartLocationAndHeading();
```

remove the line(s)

```
 if (courseNum == 2)
  lfs.setPositionAndHeading (52,12,0);
```

Adding Code To Change Appearance of Sensors

When you analyze sensor data, you now have the ability to set the color of the on-screen sensor display. For example if a spot sensor detects a brightnes above 0.2 you might want to change the color of the sensor on-screen. This helps provide some feedback that your code is doing what it should be doing...

See the LFS_SimpleBot sketch for examples of both SpotSensor coloring and LineSensor coloring.

SpotSensor new method setColor
LineSensor new method getColorArray

getColorArray requires you declare a reference to the LineSensor color array

In both cases colors are represented as integer values and the color method is handy for setting values. For example filling all sensor elements of a line sensor with bright blue.

```
int[] colorTable = sensor1.getColorArray();
for (int i=0; i<sensor1.getSensorCellCount(); i++)
   colorTable[i] = color(0,0,255);   // set all  r,g,b  to bright blue    colors 0..255 range
```