**Practical no. 8**

package TorchFollower;

import ch.aplu.robotsim.Gear;

import ch.aplu.robotsim.LegoRobot;

import ch.aplu.robotsim.LightSensor;

import ch.aplu.robotsim.RobotContext;

import ch.aplu.robotsim.SensorPort;

import ch.aplu.robotsim.Tools;

/\*\*

\*

\*/

public class TorchFollower {

TorchFollower()

{

LegoRobot robot = new LegoRobot();

LightSensor lsFR = new LightSensor(SensorPort.S1, true);

LightSensor lsFL = new LightSensor(SensorPort.S2, true);

LightSensor lsRR = new LightSensor(SensorPort.S3, true);

LightSensor lsRL = new LightSensor(SensorPort.S4, true);

Gear gear = new Gear();

robot.addPart(gear);

robot.addPart(lsFR);

robot.addPart(lsFL);

robot.addPart(lsRL);

robot.addPart(lsRR);

gear.setSpeed(25);

gear.forward();

double s = 0.02;

while (!robot.isEscapeHit())

{

int vFR = lsFR.getValue();

int vFL = lsFL.getValue();

int vRR = lsRR.getValue();

int vRL = lsRL.getValue();

double d = 1.0 \* (vFL - vFR) / (vFL + vFR);

if (vRL + vRR > vFL + vFR) // torch behind robot

gear.left();

else if (d > -s && d < s)

gear.forward();

else

{

if (d >= s)

gear.leftArc(0.05);

else

gear.rightArc(0.05);

}

Tools.delay(100);

}

robot.exit();

}

public static void main(String[] args)

{

TorchFollower t =new TorchFollower();

}

// ------------------ Environment --------------------------

static

{

RobotContext.useTorch(1, 150, 250, 100);

}

}

**Output**

