Finch Commands

`http://www.finchrobot.com/javadoc/index.html?edu/cmu/ri/createlab/terk/robot/finch/Finch.html`

Please be sure to send quit() command to the finch at end of program.

void	buzz(int frequency, int duration)	Plays a tone at the specified frequency for the specified duration on the Finch's internal buzzer.
void	closeAccelerometerGraph()	Closes the opened Accelerometer Graph
void	closeLightSensorGraph()	Closes the opened Light sensor Graph
void	closeTemperatureGraph()	Closes the opened temperature Graph
double[]	getAccelerations()	Use this method to simultaneously return the current X, Y, and Z accelerations experienced by the robot.
int	getLeftLightSensor()	Returns the value of the left light sensor.
int[]	getLightSensors()	Returns a 2 integer array containing the current values of both light sensors.
boolean[]	getObstacleSensors()	Returns the value of both obstacle sensors as 2 element boolean array.
int	getRightLightSensor()	Returns the value of the right light sensor.
double	getTemperature()	The current temperature reading at the temperature probe.
double	getXAcceleration()	This method returns the current X-axis acceleration value experienced by the robot.
double	getYAcceleration()	This method returns the current Y-axis acceleration value experienced by the robot.
double	getZAcceleration()	This method returns the current Z-axis acceleration value experienced by the robot.
boolean	isBeakDown()	This method returns true if the beak is pointed at the floor, false otherwise
boolean	isBeakUp()	This method returns true if the beak is up (Finch sitting on its tail), false otherwise
boolean	isFinchLevel()	This method returns true if the Finch is on a flat surface
boolean	isFinchUpsideDown()	This method returns true if the Finch is upside down, false otherwise
boolean	isLeftLightSensor(int limit)	Returns true if the left light sensor is less than the value specified by limit, false otherwise.
boolean	isLeftWingDown()	This method returns true if the Finch's left wing is pointed at the ground
boolean	isObstacle()	Returns true if either left or right obstacle sensor detect an obstacle.
boolean	isObstacleLeftSide()	Returns true if there is an obstruction in front of the left side of the robot.
boolean	isObstacleRightSide()	Returns true if there is an obstruction in front of the right side of the robot.

boolean	isRightLightSensor(int limit)	Returns true if the right light sensor is less than the value specified by limit, false otherwise.
boolean	isRightWingDown()	This method returns true if the Finch’s right wing is pointed at the ground
boolean	isShaken()	Returns true if the Finch has been shaken since the last accelerometer read
boolean	isTapped()	Returns true if the Finch has been tapped since the last accelerometer read
boolean	isTemperature(double limit)	Returns true if the temperature is less than the value specified by limit, false otherwise.
void	playClip(String fileLocation)	Plays a wav file over computer speakers at the specificied fileLocation path.
void	playTone(int frequency, int duration)	Plays a tone over the computer speakers or headphones at a given frequency (in Hertz) for a specified duration in milliseconds.
void	playTone(int frequency, int volume, int duration)	Plays a tone over the computer speakers or headphones at a given frequency (in Hertz) for a specified duration in milliseconds at a specified volume.
void	quit()	This method properly closes the connection with the Finch and resets the Finch so that it is immediately ready to be controlled by subsequent programs.
void	saySomething(String sayThis)	Takes the text of ’sayThis’ and synthesizes it into a sound file and plays the sound file over computer speakers.
void	saySomething(String sayThis, int duration)	Takes the text of ’sayThis’ and synthesizes it into a sound file and plays the sound file over computer speakers.
void	setLED(Color color)	Sets the color of the LED in the Finch’s beak using a Color object.
void	setLED(Color color, int duration)	Sets the color of the LED in the Finch’s beak using a Color object for the length of time specified by duration.
void	setLED(int red, int green, int blue)	Sets the color of the LED in the Finch’s beak.
void	setLED(int red, int green, int blue, int duration)	Sets the color of the LED in the Finch’s beak for the length of time specified by duration.
void	setWheelVelocities(int leftVelocity, int rightVelocity)	This method simultaneously sets the velocities of both wheels.
void	setWheelVelocities(int leftVelocity, int rightVelocity, int timeToHold)	This method simultaneously sets the velocities of both wheels.

void	showAccelerometerGraph()	Displays a graph of the X, Y, and Z accelerometer values.
void	showLightSensorGraph()	Displays a graph of the left and right light sensor values.
void	showTemperatureGraph()	Displays a graph of the temperature value.
void	sleep(int ms)	This method uses Thread.sleep to cause the currently running program to sleep for the specified number of seconds.
void	stopWheels()	Stops both wheels.
void	updateAccelerometerGraph (double xVal, double yVal, double zVal)	updates the accelerometer graph with accelerometer data specified by xVal, yVal, and zVal.
void	updateLightSensorGraph (int leftSensor, int rightSensor)	Updates the light sensor graph with the left and right light sensor data.
void	updateTemperatureGraph (double temp)	Updates the temperature graph with the most recent temperature data.

Sample Program

```
import acm.program.*;
import java.awt.*;
import edu.cmu.ri.createlab.terk.robot.finch.Finch;

public class FinchWorks extends FinchProgram {

    public void run() {

        Finch myFinch = getFinch();
        myFinch.buzz(350, 200);
        myFinch.sleep(220);
        myFinch.saySomething("I am a talking finch.  Listen to me!");
        myFinch.setLED(Color.cyan);

        myFinch.quit();
    }

    public static void main(String[] args) {
        new FinchWorks().start(args);
    }
}
```