K9 914 control Project

-Billy G Moon

There are 5 basic Panels that the GUI provides as follows:

1) Manual Control Panel--> allows you to specify power, max speed and left/right distances +/- mm for the robot to move. Click a button and the robot does the movement.
2) A Sensor Panel--> allows you to turn on monitoring of individual sensors on the robot. Distance readings are shown in cm. >From this, you will see that the sensor reading routines can supply "reasonable" distance measurements without any calibration.
3) A Light Control Panel --> allows you to turn on/off and blink the head and eye lights at various rates.
4) An M3 Command Panel --> allows you to construct and send any M3 command. Results of sending the command are shown. This is very good for playing around with low level commands on the robot.
5) An M3 Trace Panel --> allows you to turn on/off 'command' tracking of the actual M3 commands and responses. Log can be cleared and/or saved to a file.

The serial interface for the Java code is provided by the open source package supplied at www.rxtx.org. RXTX and Java are supported on over 100 different Operating System/Hardware platforms and so this code should run fine on any of those (of course, YMMV.)

We had to make one small change to the base RXTX package in order to support the 416667 baud rate. We will provide a custom build of that code with the software for others to use.

The Java code is divided into two packages: k9 and k9.device. K9 provides the 'main' application and simple user interface to illustrate how to use the k9.device code. All code is pretty well documented and contains full JavaDoc documentation as well.

The k9.device package contains the following classes:

k9.device.constants --> basically M3 opcodes that can be included in any other class
k9.device.DeviceFactory --> used to get access to various k9 devices
k9.device.Lights --> used to control the lights on the robot
k9.device.M3 --> provides low level M3 interfaces and communications
k9.device.M3CommandTracer --> interface that allows applications to register for M3 command traces from the M3 class
k9.device.Motors --> used to control the motors on the robot
k9.device.MotionListener --> interface that allows applications to register for motion events with the Motors class
k9.device.Sensors --> used to get access to sensor data
k9.device.SensorListner --> interface that allows applications to register for sensor events with the Sensors class

Here are some keys that can be used with the K9 control application. To use them, you add commands to the command line when starting the K9 application. Some examples:

java -Dk9.device.M3.serialPortName="\dev\tty03" -jar K9.jar

java -Dk9.device.debug=true -Dk9.device.lowLevelDebug=true -jar K9.jar

======Configuration KEYS===============================

k9.device.M3.serialPortName = "text name that makes sense for the OS" ==> Sets the name of the serial port to use to talk to the M3 controller. Default is "COM3".

k9.device.debug=<true or false> ==> Turns on/off high level debug statements. Basically, when on, K9 will tell you about each sub-system as it is enabled and configured. Output is on standard out. Default is "false".

k9.device.lowLevelDebug=<true or false> ==> Turns on/off low level debug statements. Basically, when on, K9 will tell you about each packet that it transmits and receives with the M3 controller. A lot like the "tracer" window, but starts at power up. Again, output is on standard out. Default is "false".

k9.device.motors.acceleration=<floating point value in mm/sec^2> ==> Sets the default acceleration rate for the motors. Default is 100.

k9.device.motors.deceleration=<floating point value in mm/sec^2> ==> Sets the default deceleration rate for the motors. Default is 100.

k9.device.motors.power=<floating point value from 0 to 100> ==> Sets the default power level for the motors. Default is 75 for 75%. Range should be between 20% and 100%. Don't include the % in the definition.

k9.device.motors.velocity=<float point value in mm/sec> ==> Sets the default maximum velocity to be used during movement. Default is 100.

k9.device.motors.pollInterval=<long value in millliseconds> ==> Sets the polling interval that the "motors" class uses to check on the status of the M3 controller. Default is 500 or about 1/2 second. Don't go much below 50 or greater than 1500.

k9.device.sensors.pollInterval=<long value in milliseconds> ==> Sets the polling interval that the "sensors" class uses to read the sensors at. Default value is 200. Don't go much below 100 or more than 3000. The sensor hardware's max response time is ~50 ms.