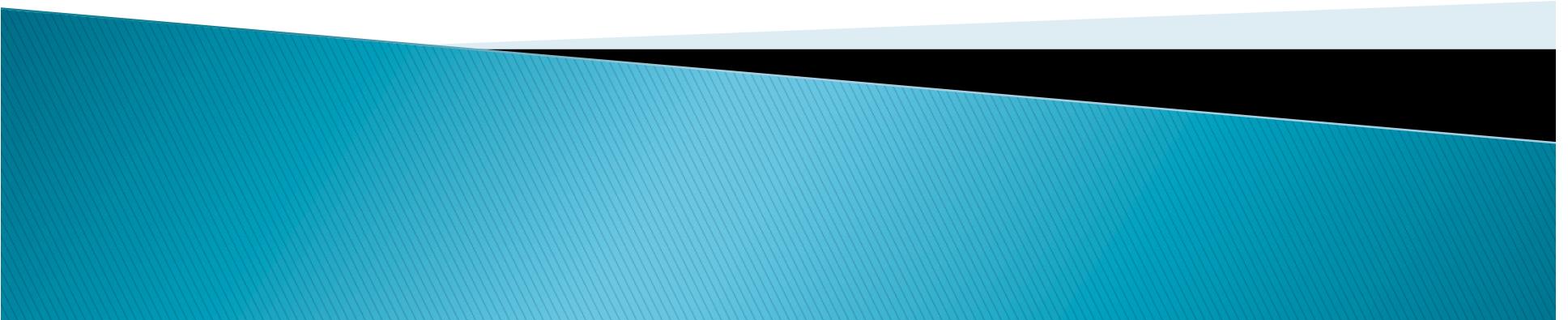


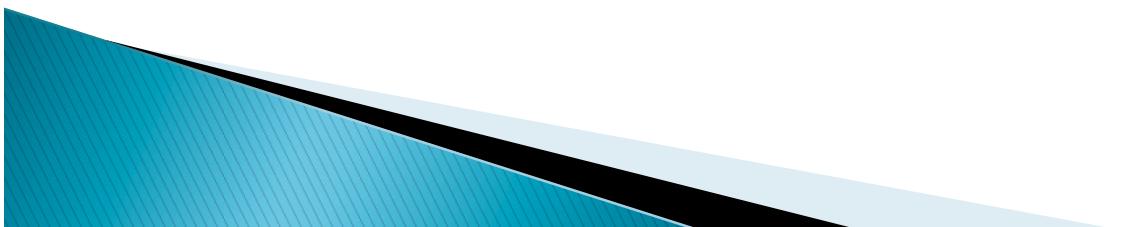
EGN 4060: Introduction to Robotics Lab Instructions

Instructor: Gita Sukthankar
Lab Supervisor: Astrid Jackson



Outline

- ▶ Announcements
- ▶ Assembling iRobot
- ▶ Connecting to network
- ▶ Running code on robot
- ▶ Brief Java tutorial
- ▶ Useful functions
- ▶ Troubleshooting



Announcements

- ▶ Lab hours:
 - Tuesday 6:00pm – 9:00pm
 - Thursday 6:00pm – 9:00pm
- ▶ Room: HEC 302

Notes:

- ▶ Bring your laptops to the lab
- ▶ Bring an ethernet connector for your laptop if it doesn't have one built-in
- ▶ If TA is running late, gather in front of the lab area



TAs

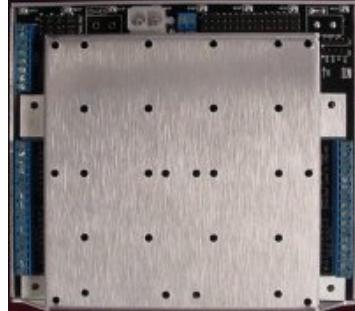
- ▶ Astrid Jackson
 - a.jackson@knights.ucf.edu
 - Office hours: by appointment (send email)
 - Office: HEC 201



Assembling iRobot

- ▶ Where do we start?
 - We will give you the parts for the robot
- ▶ What are the robot parts?
 - 2 main parts and connection cables

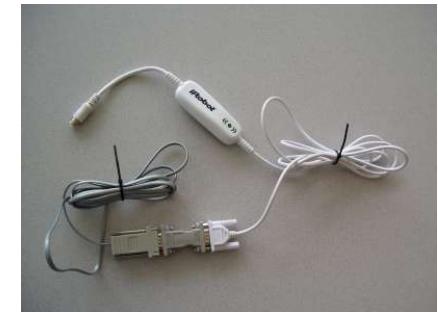
Qwerk Board
(middleware)



iRobot Create

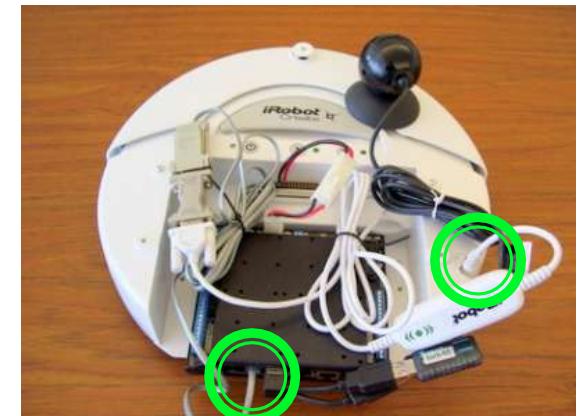
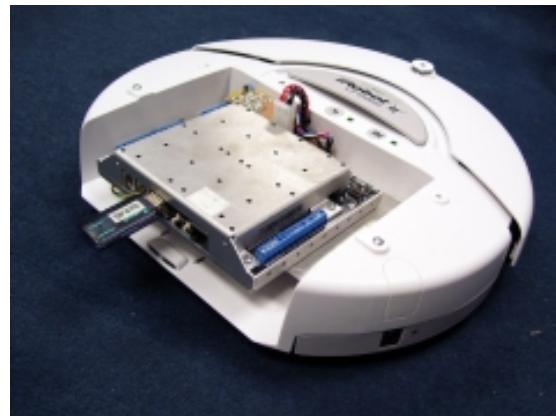
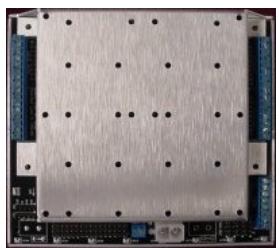


Serial Cable



iRobot

Qwerk Board

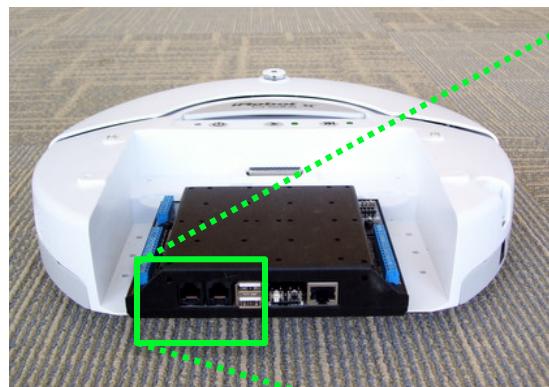


iRobot Create



Important !

- ▶ Cable should be plugged into UART 2



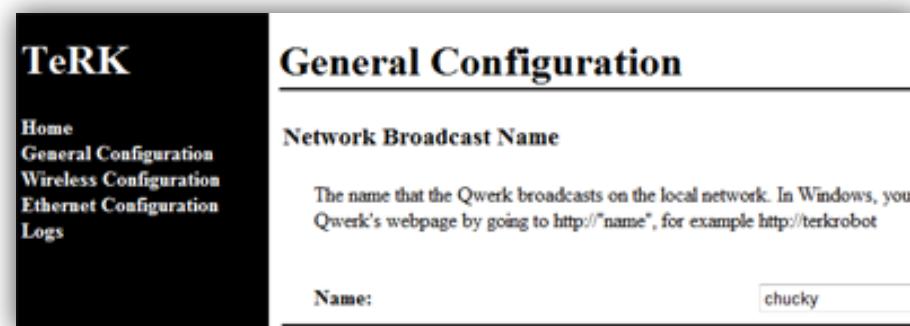
Connecting to Network



- ▶ Your laptop performs direct connection
- ▶ How? Security?
 - Configure laptop's Ethernet connection with static IP address (IP: 10.0.0.1, Subnet mask: 255.0.0.0)
 - Connect robot to laptop via Ethernet cable
 - LED 1 and LED 2 should light steadily and LED 6 should be blinking when connection is established

Robot Setup

- ▶ Assemble your robot
- ▶ Establish cable (Ethernet) connection to laptop
- ▶ Go to: **http://10.0.0.10**
- ▶ Select General Configuration
 - Confirm Direct–Connection is opted
 - Name your robot
(ie: chucky)



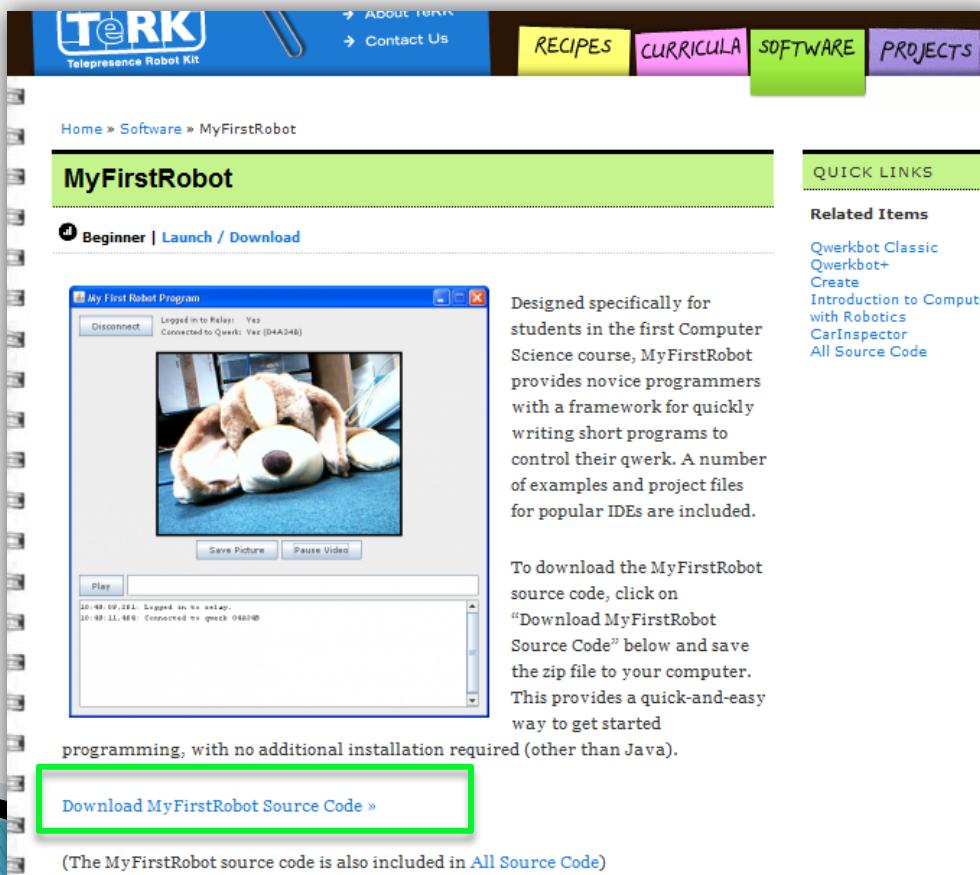
Setting up Software Tools

- ▶ We're going to use Java for coding
- ▶ Install Java platform SE 8 Update 20
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- ▶ We will use Eclipse Standard 4.4 as IDE:
<http://www.eclipse.org/downloads/>



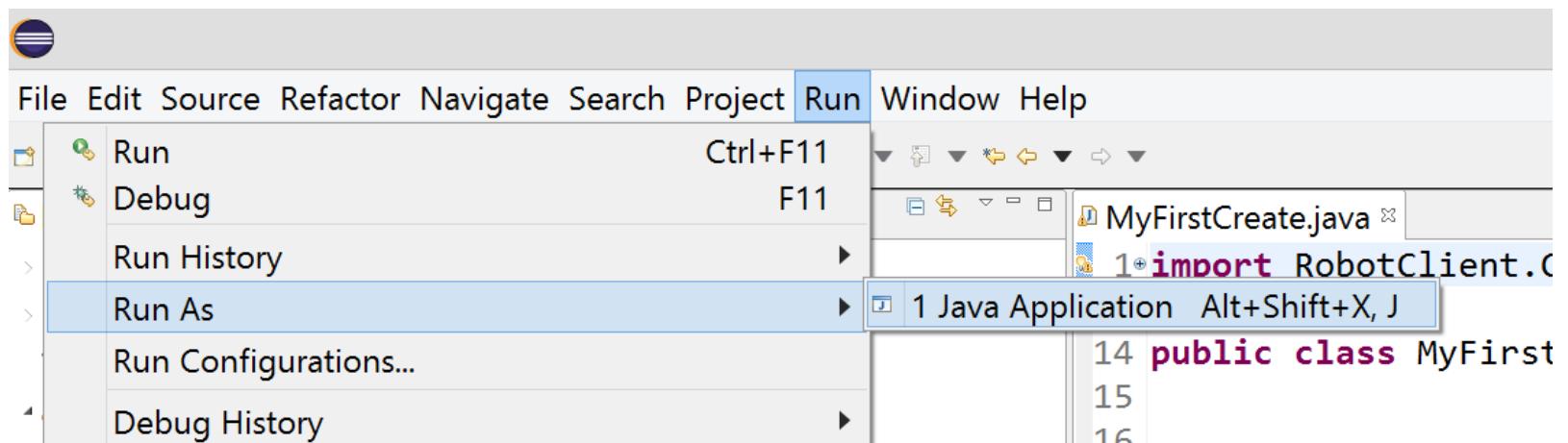
Running Code on Robot

- ▶ Download the robot source code from webcourses



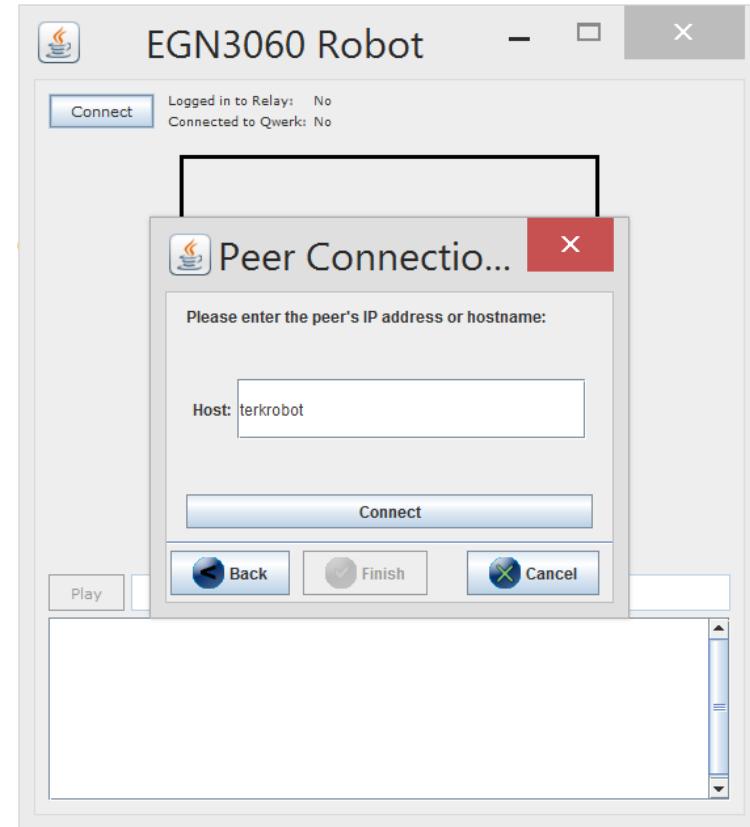
Running Code on Robot

- ▶ Import project into Eclipse
- ▶ Open “MyFirstCreate.java” code
- ▶ Run As > Java Application



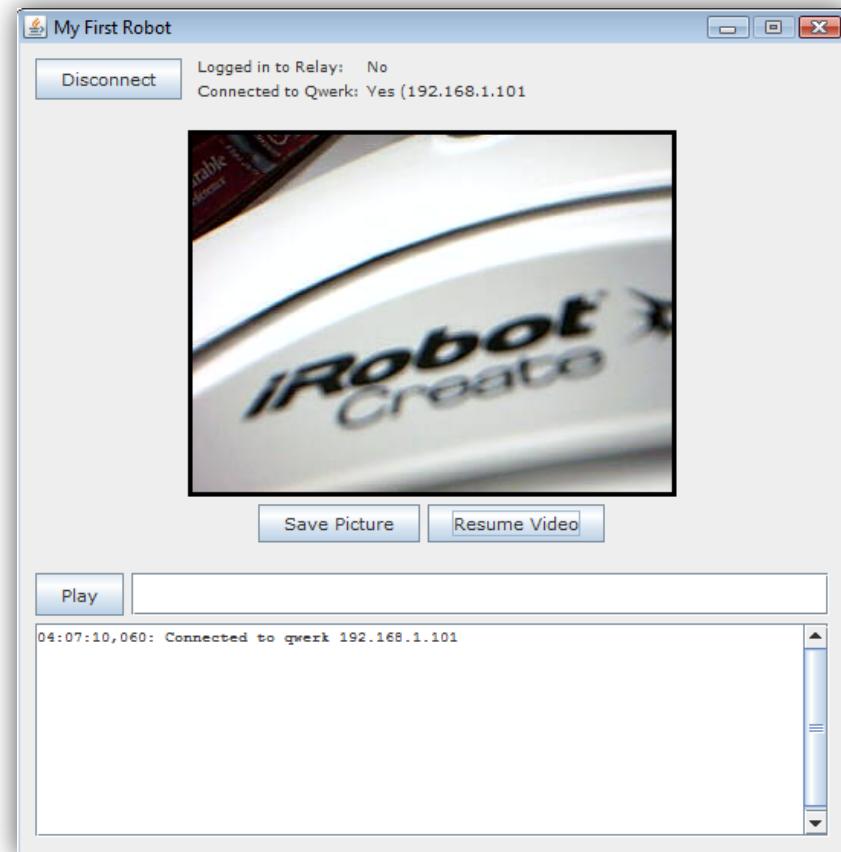
Running Code on Robot

- ▶ Hit: “Connect” on top left corner
- ▶ Enter your robot’s name or IP address into popup window
- ▶ Click “Connect”
- ▶ Wait for connection
- ▶ Click “Finish”



Running Code on Robot

- ▶ Click “Play Video” and check if you get video frames from the camera



Running Code on Robot

- ▶ Also run “CreateMove.java”
 - You should observe your robot move
- ▶ Familiarize yourselves with the “RobotClient” and “CreateClient” java files



Important !

- ▶ If your robot doesn't move at this point
 - Check if your robot is connected
 - Check if your robot has enough battery



Brief Java

```
public class MyFirstRobot
{
    // comments
    public static void main (String[] args)
    {
        // write your code here
    }
}
```

- You write your code in the “main” method



Printing out information

- ▶ Calling the `System.out.println` method prints a character string

```
total = 6;  
System.out.println("The Total is: " + total);
```

object method name Input to the method
 (parameters)

> The Total is: 6

Problems with Coding Math

- ▶ If both operands to the division operator (/) are integers, the result is an integer (the fractional part is discarded)
 - $14/3 \rightarrow 4$, whereas `(double)14/3` $\rightarrow 4.666667$
 - $8/12 \rightarrow 0$, whereas `(double)8/12` $\rightarrow 0.666667$
- ▶ The modulo operator (%) returns the remainder after dividing the second operand into the first
 - $14 \% 3 \rightarrow 2$
 - $8 \% 12 \rightarrow 8$
 - $-2 \% 5 \rightarrow -2$ (not 3 in java)



If – else Statements

Conditional statements

```
if (ballAngle > 20) {  
    myRobot.moveAngle(20);  
} else if(ballAngle < -20){  
    myRobot.moveAngle(-20);  
} else {  
    myRobot.move(100,100);  
}
```



Loops

► Continuous statements

```
for(int i=0; i<numSteps; i++){  
    myRobot.moveDistance(400);  
}
```

```
while(true){  
    myRobot.move(100,100);  
}
```



Java and Eclipse Tutorial

- ▶ <http://sourceforge.net/projects/ecliptutorial/files/>



Useful Robot Functions

Purpose	Functions
User input	waitForPlay() isPlaying() getTextFieldValueAsInt()
Sensors	bumpRight() bumpLeft()
Movement	moveMotors() moveDistance()
Turning	moveAngle()
Stopping	stopMoving()
Program Output	writeToTextField()
Other	dockRobot() unDockRobot()

Using Functions

- ▶ Initialize the robot before you call functions
 - `myRobot.initialize()`
- ▶ Use myRobot object to call the functions
 - `myRobot.moveDistance(400); //will make the
//robot move 40cm`
- ▶ Check examples to familiarize yourself with the robot functions



Troubleshooting

- ▶ If camera gives gray video, charge your robot
- ▶ If you get “Flushed data from serial buffer” warning, charge and restart your robot
- ▶ If you grab your robot and wheels come down, restart your robot
- ▶ Instead of holding your robot up, roll it on front wheel
- ▶ Check division errors
 - ie: $400 * (\text{pos} / \text{side}) \rightarrow 0$ if $\text{side} > \text{pos}$
 - Use casting instead: $400 * ((\text{double})\text{pos} / \text{side})$



Questions ?

