Developing with Gumstix

This guide provides step-by-step instructions for using the Eclipse Integrated Development Environment (IDE) to develop applications for the Gumstix COMs.

We'll walk through these tasks:

- setting up Eclipse
- connection to your Gumstix system
- creating a simple "Hello, World!" Python program
- running a web server
- installing new packages on your Gumstix
- creating a small graphical Java application
- remote debugging of your Java application
- natively-compiling C/C++ code on your Gumstix
- remote debugging C/C++ using the GDB server interface

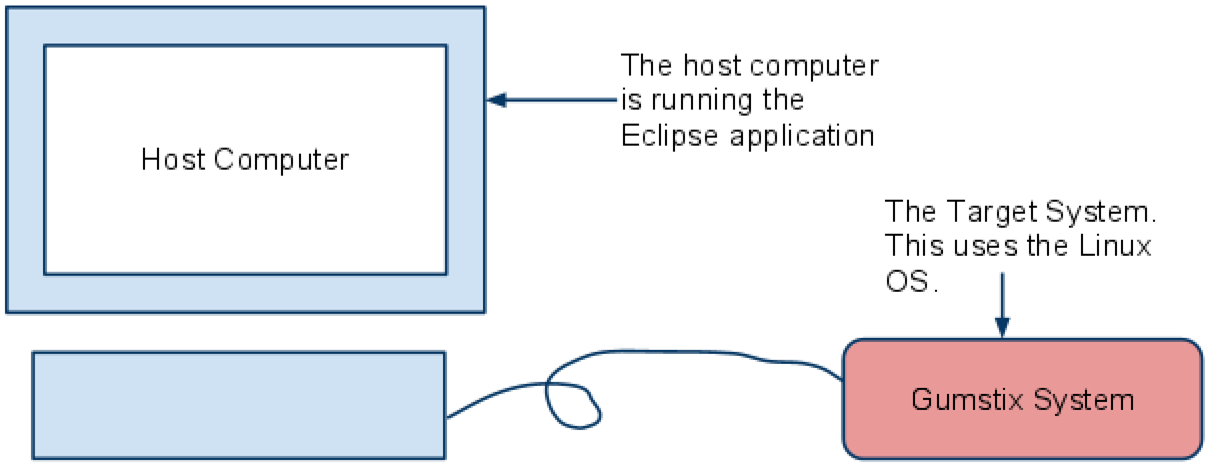
We will also explain how to set up a cross-compilation toolchain so you can develop C/C++ applications on your development computer and upload them to your Gumstix.

Remember, Gumstix COMs are fully-fledged, albeit tiny, Linux systems so there are many tools we don't cover here. For example, this document explains how to use the powerful OpenEmbedded build system within Eclipse.

Required Tools:

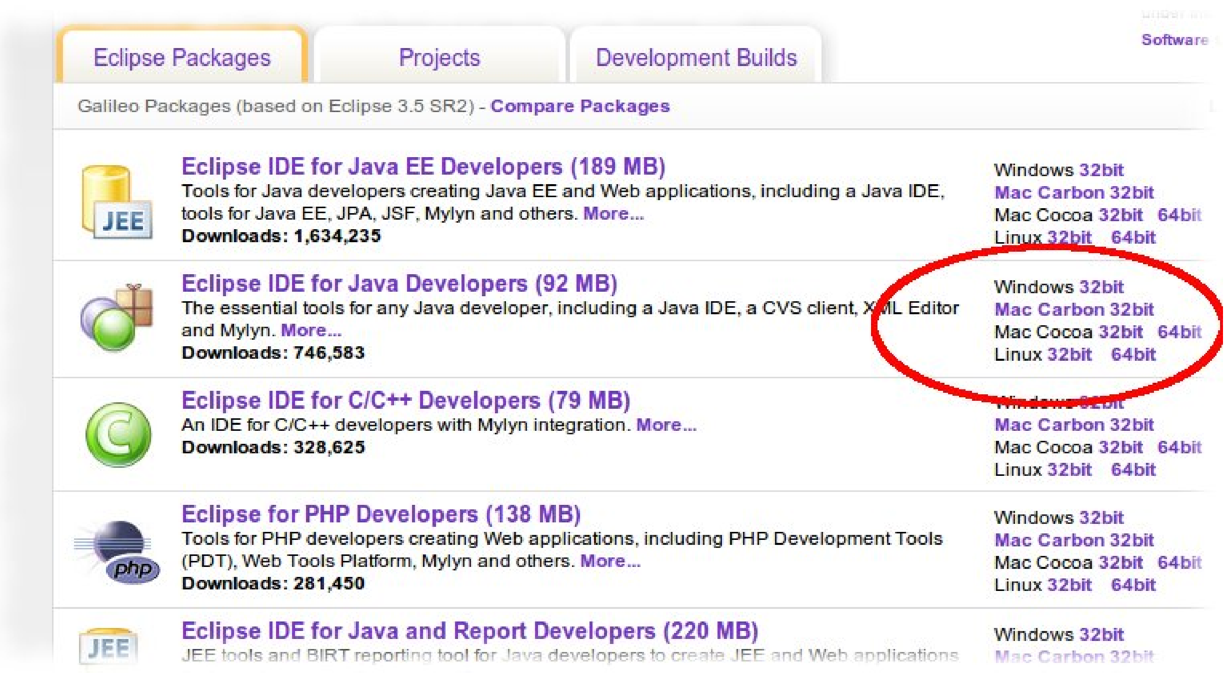
- any Windows, Apple or Linux computer with a USB port
- a recent version of the Java Runtime Environment (available here)
- a Gumstix COM running up-to-date software
- any expansion board of the Gumstix Overo series, such as Chestnut or Tobi, that has both a USB console port and a 10/100 Ethernet jack
- a 5V power supply
- an ethernet cable and internet connection
- a mini-A to Standard-A USB cable
- a bootable microSD card (only needed for native C/C++ compilation)

Key Idea:


Host Computer
The host computer is running the Eclipse application
The Target System. This uses the Linux OS.
Gumstix System

Step 1: Download Eclipse

Download the Eclipse IDE for Java Developers for your operating system.



Galileo Packages (based on Eclipse 3.5 SR2) - **Compare Packages**

JEE	**Eclipse IDE for Java EE Developers (189 MB)** Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn and others. More... **Downloads: 1,634,235**	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
	Eclipse IDE for Java Developers (92 MB) The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. More... **Downloads: 746,583**	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
	Eclipse IDE for C/C++ Developers (79 MB) An IDE for C/C++ developers with Mylyn integration. More... **Downloads: 328,625**	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
php	**Eclipse for PHP Developers (138 MB)** Tools for PHP developers creating Web applications, including PHP Development Tools (PDT), Web Tools Platform, Mylyn and others. More... **Downloads: 281,450**	Windows 32bit Mac Carbon 32bit Mac Cocoa 32bit 64bit Linux 32bit 64bit
JEE	**Eclipse IDE for Java and Report Developers (220 MB)** JEE tools and BIRT reporting tool for Java developers to create JEE and Web applications	Windows 32bit Mac Carbon 32bit

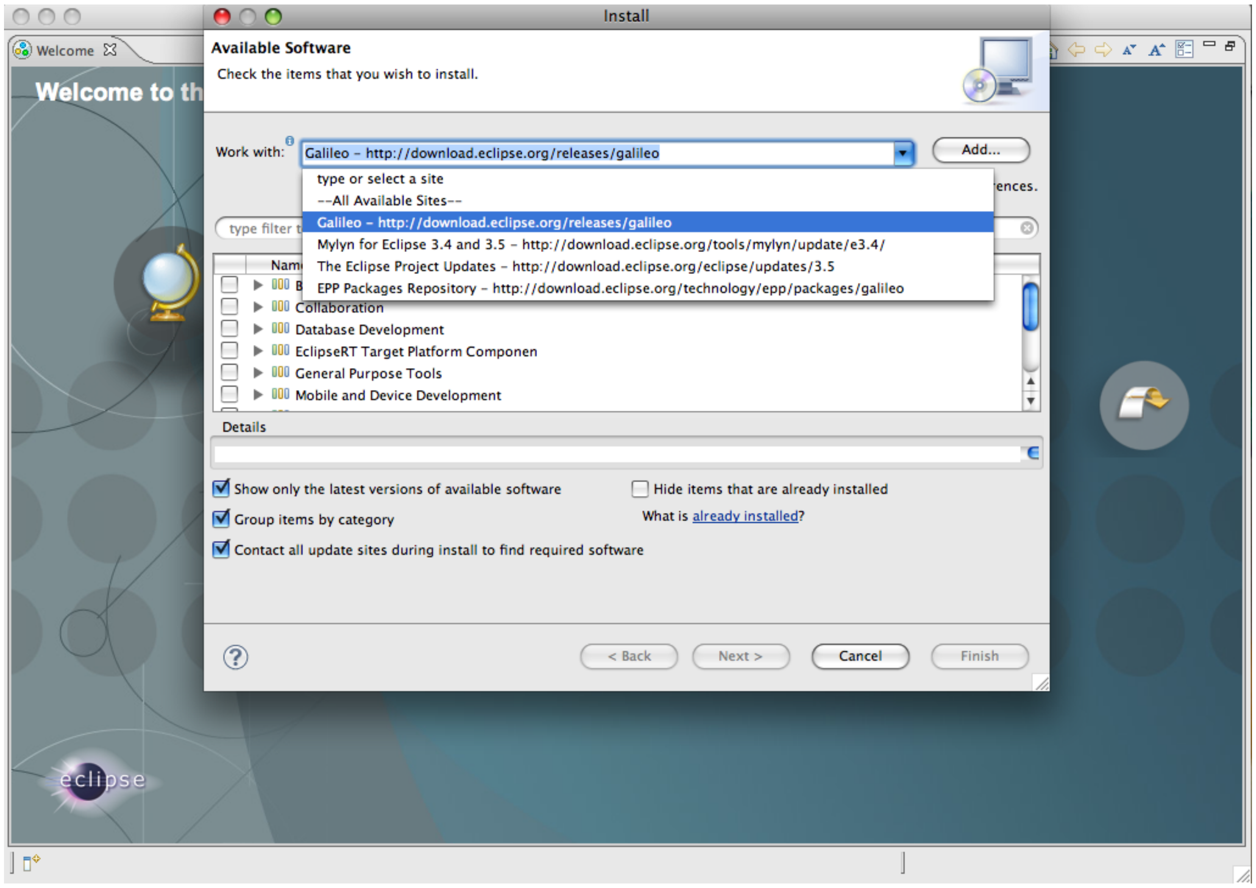
- it is *not* necessary to install the Eclipse software; just double-click on the application once it has finished downloading
- when prompted, choose a workspace folder in which you want to store your projects and your configuration information
- then click through to the Eclipse workbench itself

Step 2: Install Plugins

For developers, a great variety of additional features can be added to Eclipse from various repositories of plug-ins known as *Update Sites*.

To install a new plugin in Eclipse, first select the Update Site.

- navigate to Help->Install New Software...
- select the update site for your version of Eclipse from the "Work with" drop-down menu



Install

Available Software

Check the items that you wish to install.

Work with: Galileo - http://download.eclipse.org/releases/galileo

- type or select a site
- --All Available Sites--
- Galileo - http://download.eclipse.org/releases/galileo
- Mylyn for Eclipse 3.4 and 3.5 - http://download.eclipse.org/tools/mylyn/update/e3.4/
- The Eclipse Project Updates - http://download.eclipse.org/eclipse/updates/3.5
- EPP Packages Repository - http://download.eclipse.org/technology/epp/packages/galileo

Collaboration
Database Development
EclipseRT Target Platform Componen
General Purpose Tools
Mobile and Device Development

Details

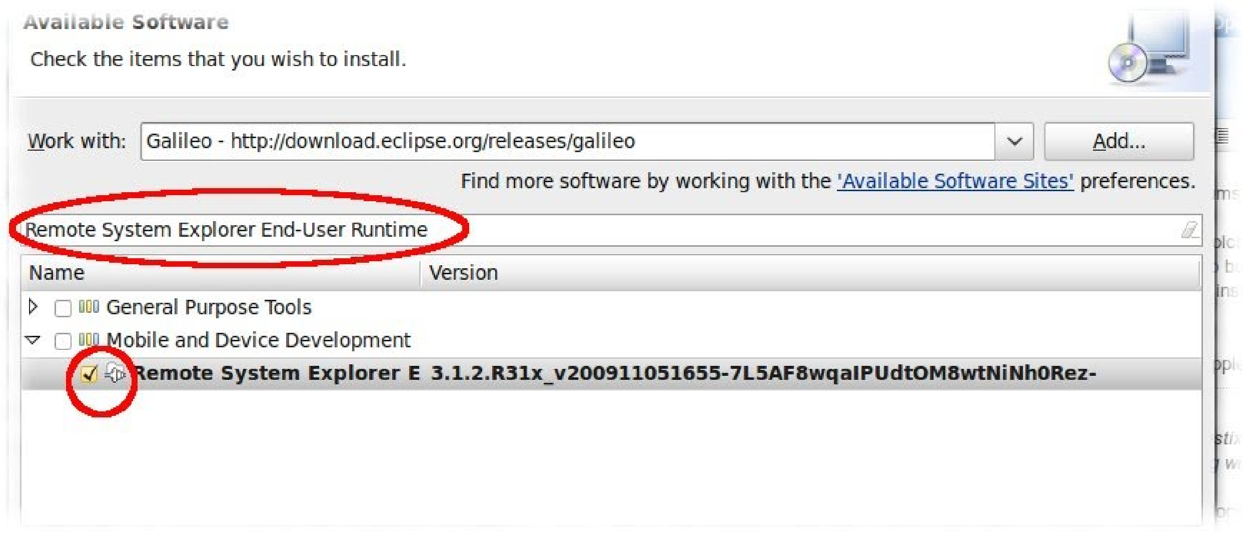
[✓] Show only the latest versions of available software
[ ] Hide items that are already installed
[✓] Group items by category
What is already installed?
[✓] Contact all update sites during install to find required software

Hint:

If the update site is not available, you can add it by clicking the 'Add...' button on the right. For the Galileo release, you can use the URL *http://dowload.eclipse.org/releases/galileo*.

As we are working with a remote system, we'll install the Remote System Explorer plugin.

- search for the *Remote System Explorer End-User Runtime* plugin
- select it for installation


Available Software

Check the items that you wish to install.

Work with: Galileo - http://download.eclipse.org/releases/galileo

Find more software by working with the 'Available Software Sites' preferences.

Remote System Explorer End-User Runtime

Name	Version
☐ General Purpose Tools	
☐ Mobile and Device Development	
✓ Remote System Explorer E	3.1.2.R31x_v200911051655-7L5AF8wqaIPUdtOM8wtNiNh0Rez-

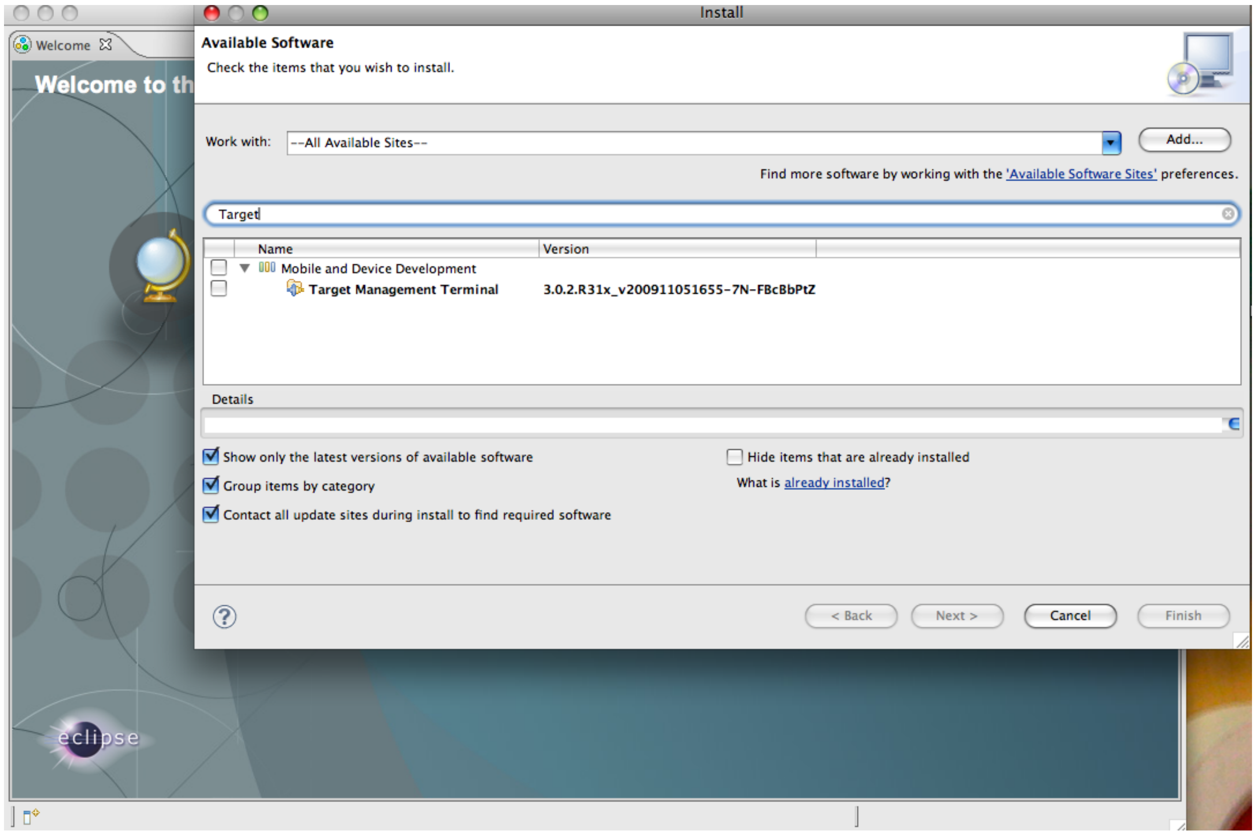
Still in Eclipse,

- select "Next".... twice
- select the radio button to "ACCEPT" the terms of the license agreement
- choose "Yes" to restart Eclipse for the changes to take effect

Once Eclipse has restarted, let's add two other useful plugins that allow connection to the console port of our Gumstix without needing a terminal emulator program such as kermit or TeraTerm.

From within Eclipse,

- navigate to Help->Install New Software
- search on "Target" to find the *Target Manager Terminal*



Available Software

Check the items that you wish to install.

Work with: --All Available Sites--

Find more software by working with the 'Available Software Sites' preferences.

Target

Name	Version
Mobile and Device Development	
Target Management Terminal	3.0.2.R31x_v200911051655-7N-FBcBbPtZ

[✓] Show only the latest versions of available software

[✓] Group items by category

[✓] Contact all update sites during install to find required software

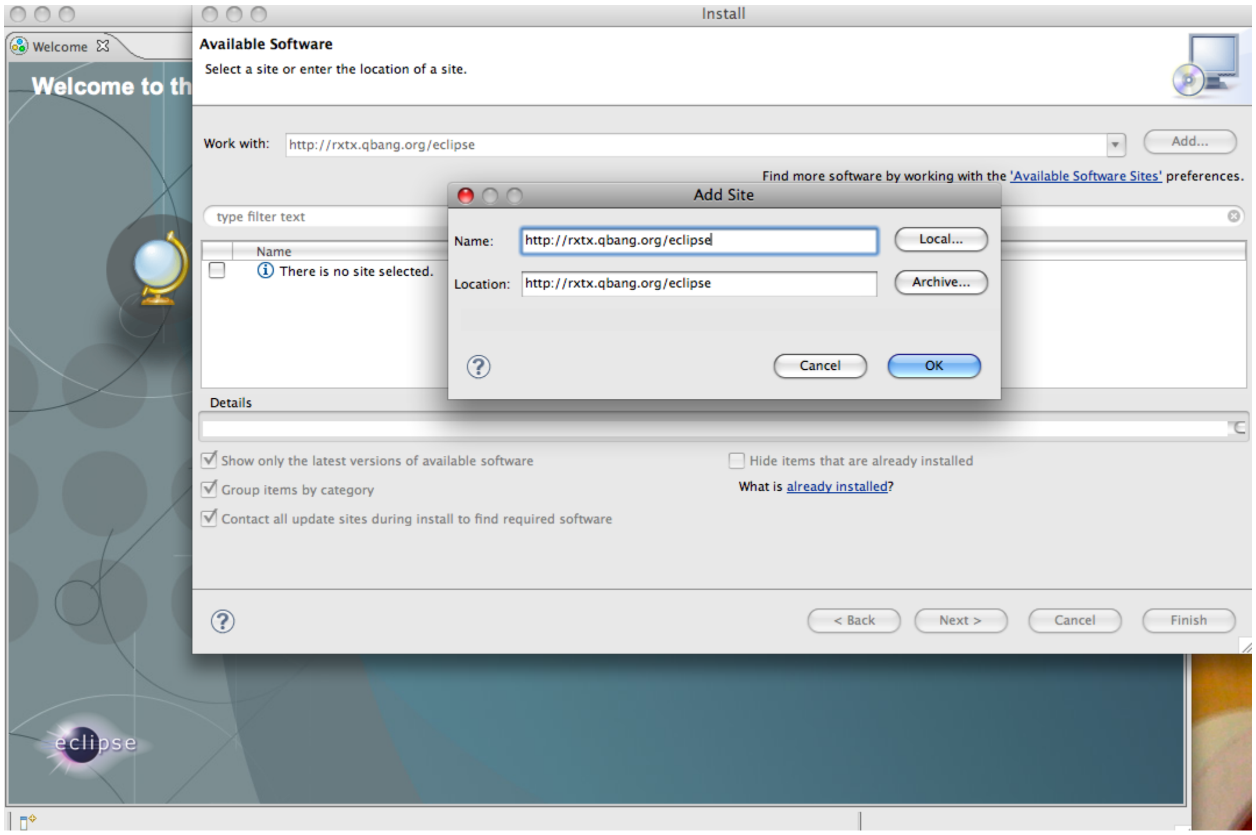
[ ] Hide items that are already installed

What is already installed?

- select the "Target Manager Terminal"
- click "Next"
- click "Finish" to complete installation
- choose "Yes" to restart Eclipse for the changes to take effect

We'll also need to add a new update site by doing these steps in Eclipse:

- navigate to Help->Install New Software...
- click the "Add..." button
- type "`http://rxtx.qbang.org/eclipse`" into the "Location" field and create a name


Install
Available Software
Select a site or enter the location of a site.
Work with:
Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text
Name
There is no site selected.
Add Site
Local...
Archive...
Cancel
OK
Details
Hide items that are already installed
What is already installed?
< Back
Next >
Cancel
Finish
Welcome
Welcome to th
eclipse

Name: http://rxtx.qbang.org/eclipse

Location: http://rxtx.qbang.org/eclipse

Work with: http://rxtx.qbang.org/eclipse

[✓] Show only the latest versions of available software

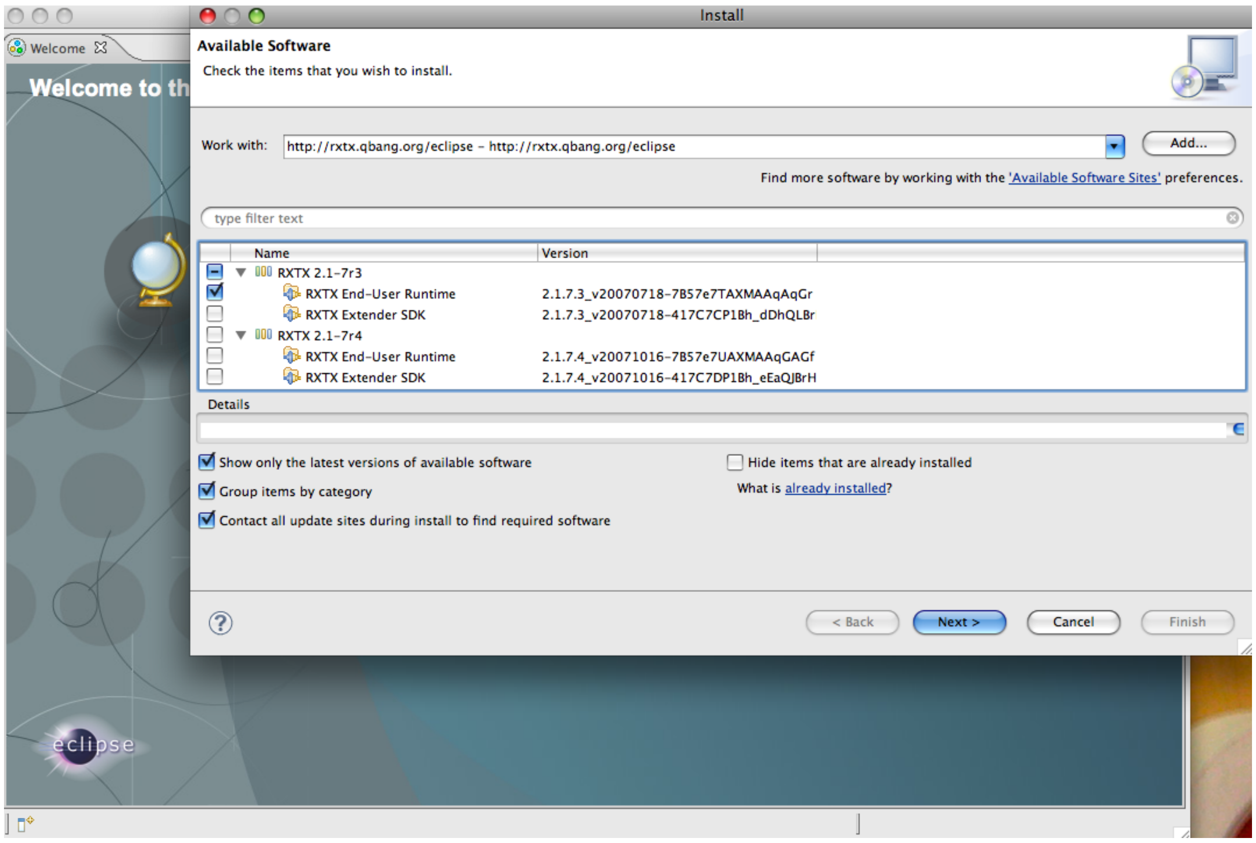
[✓] Group items by category

[✓] Contact all update sites during install to find required software

[ ] Hide items that are already installed

- Click the "OK" button

Go into the sub-directory as shown below to select the *RXTX End-User Runtime* plugin that this site provides.


Install
Welcome
Welcome to th
Available Software
Check the items that you wish to install.
Work with:
Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text
Details
What is already installed?
< Back
Next >
Cancel
Finish
eclipse

Work with: http://rxtx.qbang.org/eclipse - http://rxtx.qbang.org/eclipse

	Name	Version
▬	RXTX 2.1-7r3	
✓	RXTX End-User Runtime	2.1.7.3_v20070718-7B57e7TAXMAAqAqGr
☐	RXTX Extender SDK	2.1.7.3_v20070718-417C7CP1Bh_dDhQLBr
☐	RXTX 2.1-7r4	
☐	RXTX End-User Runtime	2.1.7.4_v20071016-7B57e7UAXMAAqGAGf
☐	RXTX Extender SDK	2.1.7.4_v20071016-417C7DP1Bh_eEaQJBrH

- [✓] Show only the latest versions of available software
- [✓] Group items by category
- [✓] Contact all update sites during install to find required software
- [ ] Hide items that are already installed

- click "Next" twice
- select the radio button to "ACCEPT" the terms of the license agreement
- click "OK" at the security warning window
- click "Finish" to complete installation
- choose the "Yes" button to restart Eclipse for the changes to take effect

Step 3: Connect Your Gumstix

Now it is time to connect your Gumstix---let's get started by establishing a console connection.

- mount your Gumstix COM down on to your expansion board
- plug the mini-A USB plug into the port marked CONSOLE and plug the other end of the cable into a standard USB port on your computer

Hint:

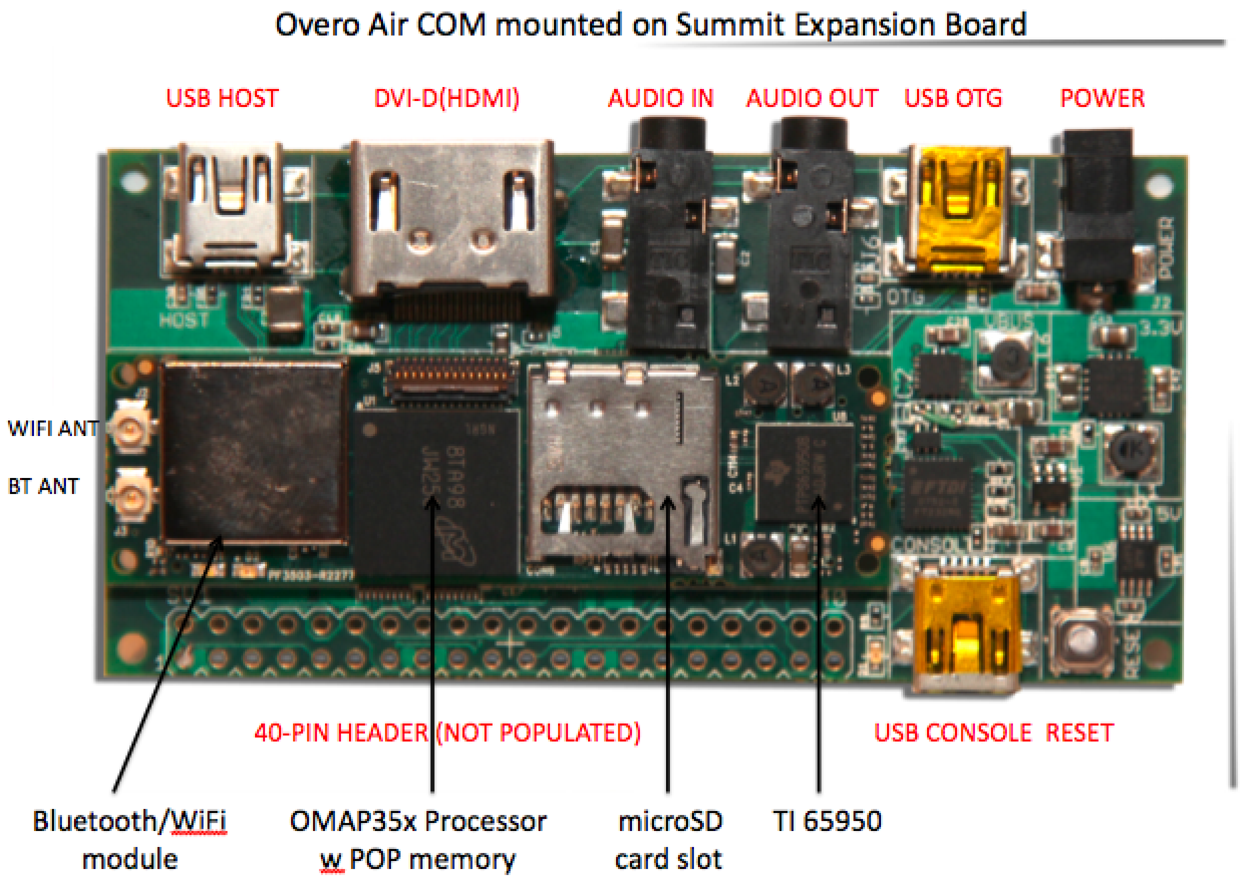
Your computer should detect your Gumstix system when you first connect the USB cable, even though the power is not yet turned on. On Macintosh and Windows systems, a pop-up window should appear instructing you to download a new driver (do so) or notifying you that a USB-serial device has been detected. If not, you can download drivers from this site. Choose the "VCP" driver.

Once you have downloaded this driver, make sure to unplug and then replug the USB cable running from your computer to your Gumstix expansion board so the Gumstix will be recognized by your computer.

- connect your Gumstix to the internet using an ethernet cable

NOTE: You still have NOT plugged the power cable in to the Gumstix expansion board!

This diagram shows the locations of the USB 'CONSOLE' port and the 'POWER' input port on a Summit board on which an Overo COM has been mounted.

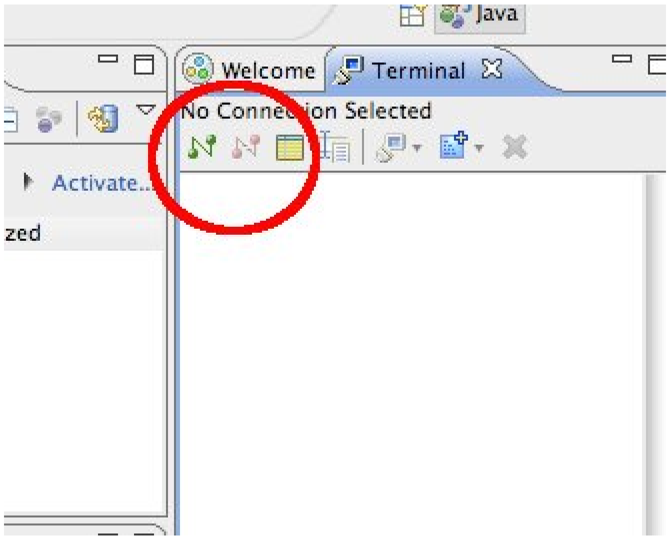


- if you are booting from a microSD card, insert this card now until you hear it click in place.

Before applying power to the board, let's open up a connection to the console port.

In Eclipse,

- navigate to Window->Show View->Other->Terminal->Terminal
- click on the 'N' connection icon in the window that appears



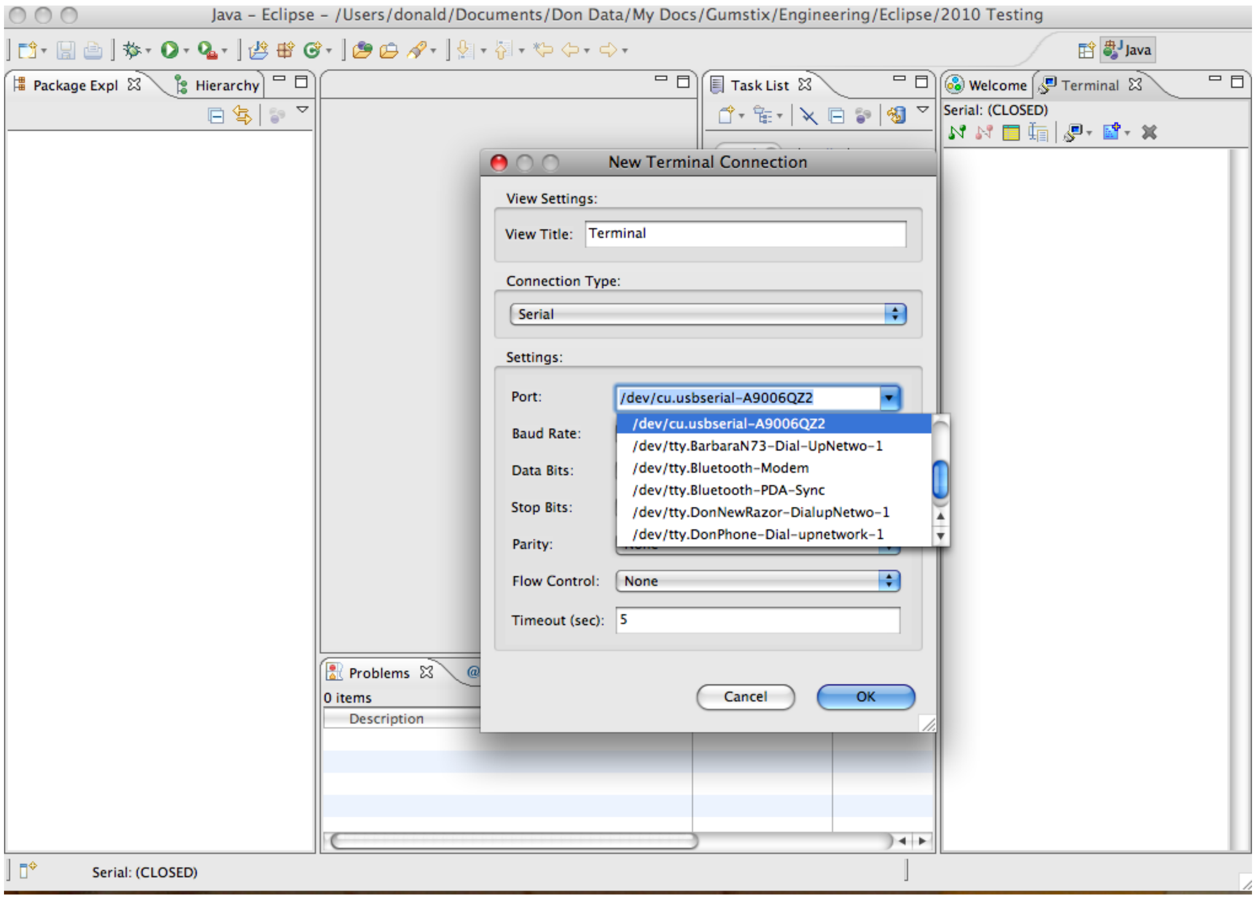
- make the connection using the following settings:

Connection Type:	`Serial`
Port:	`<port to which your Gumstix is connected>`
Baud Rate:	`115200`
Data Bits:	`8`
Stop Bits:	`1`
Parity:	`none`
Flow Control:	`none`
Timeout:	`5`

Hint:

The port to which your Gumstix is connected will have a different name depending on your Host operating system.

- On Windows, it should be '*USB-COM*'
- On Macintosh, it should be '*/dev/cu.usbserial-xxxxxxxx*' where '*xxxxxxxx*' is an arbitrary string of characters
- On Linux, it should be '*/dev/ttyUSBx*' where '*x*' is a number.


Java - Eclipse - /Users/donald/Documents/Don Data/My Docs/Gumstix/Engineering/Eclipse/2010 Testing
Package Expl
Hierarchy
Task List
Welcome
Terminal
Java
Serial: (CLOSED)
New Terminal Connection
/dev/cu.usbserial-A9006QZ2
/dev/tty.BarbaraN73-Dial-UpNetwo-1
/dev/tty.Bluetooth-Modem
/dev/tty.Bluetooth-PDA-Sync
/dev/tty.DonNewRazor-DialupNetwo-1
/dev/tty.DonPhone-Dial-upnetwork-1
Baud Rate:
Data Bits:
Stop Bits:
Parity:
Problems
0 items
Description
Cancel
OK

View Settings:

View Title: Terminal

Connection Type: Serial

Settings:

Port: /dev/cu.usbserial-A9006QZ2

Flow Control: None

Timeout (sec): 5

Now, it's time to apply power to the Gumstix expansion board so plug the 5V wall adapter into the power input jack on the Gumstix.

You should see text appear in this window; for now, just let the Gumstix boot. When prompted, you can log in to the root account. The default credentials are:

COM Board	**Username**	**Password**
Overo	root	<leave blank>
VerdexPro	root	gumstix

Step 4: Write 'Hello, World' à la Python

You are going to write a basic Python program for the Gumstix directly on the Gumstix; this is known as native development. Also, Python is an interpreted language which means you can run your programs directly without any compilation.

Once you have logged into your Gumstix, do these steps in the terminal window within Eclipse:

- open an editor on the Gumstix by typing: '`nano hello.py`'
- type out this simple Python program: '`print "Hello, Gumstix!" `'
- press `Ctrl+O`, `<Enter>`
- then press `Ctrl+X` to save and exit.
- run your program by typing: "python hello.py"
- check that you see the words "Hello, Gumstix" appear

If you see the words "Hello Gumstix", then you may have just written your first program on a Gumstix Overo computer.

Step 5: Create a Web Site

Each Gumstix comes with a built-in package manager called *opkg*. A package manager is a useful tool for installing software; you can install everything from media players to GPS mapping tools to the Firefox web browser. In this example, you'll use opkg tool to install a new application for your Gumstix: the Apache web server.

To make your own Gumstix web site, type the following in the terminal window:

```
opkg update           <Enter>
opkg install apache2  <Enter>
```

Before you start using Apache, which is now running, you'll need to find the IP address (internet address) of your Gumstix COM. To do this, type the following command into the terminal window:

```
ifconfig | grep 'inet addr'     <Enter>
```

You should see something like this appear on your screen.

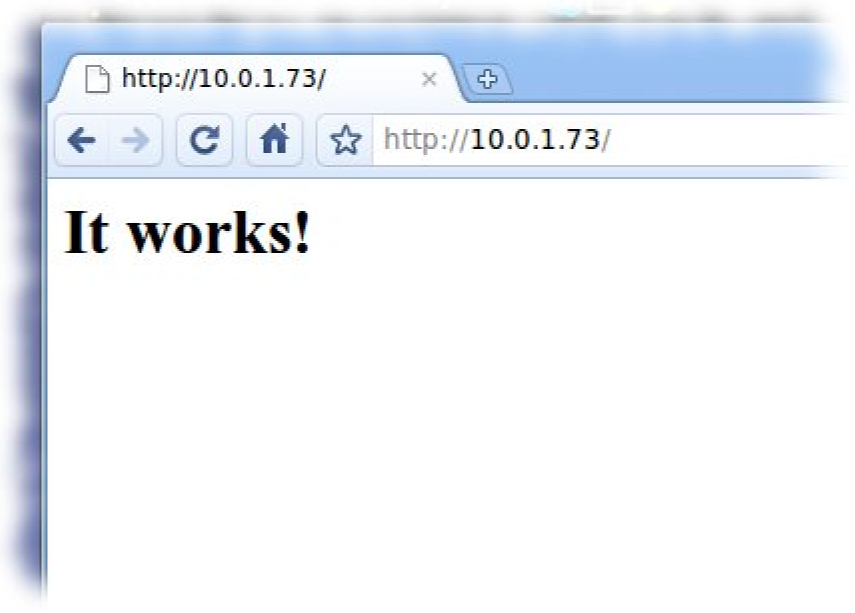
```
inet addr: 10.0.1.73 Bcast: 192.168.0.255 Mask:
255.255.255.0
```

The IP address of your Gumstix is the first address shown: *10.0.1.73* in this case.

Now,

- open up a web browser on your host computer
- type that IP address of your Gumstix into the address bar

You should see something like this which is the default web site for Apache running on your Gumstix.



It works!

To make your web site say something more interesting, do the follow in your terminal window:

- `type: 'nano /usr/share/apache2/htdocs/index.html'`
- replace the words at the top of the screen "It Works" with the words "Gumstix Rocks!"
- press `Ctrl+O`, `<Enter>`
- press `Ctrl+X` to save and exit
- refresh your web browser

Step 6: Write a Java Application

So far, you haven't actually used Eclipse other than as a console interface for your Gumstix. It is time to harness the power of Eclipse to write a small graphical Java application.

In Eclipse.

- Navigate to **File->New ->Java Project**,
- Name the project *gumstix-java*
- Select "Finish"

Once the project is created,

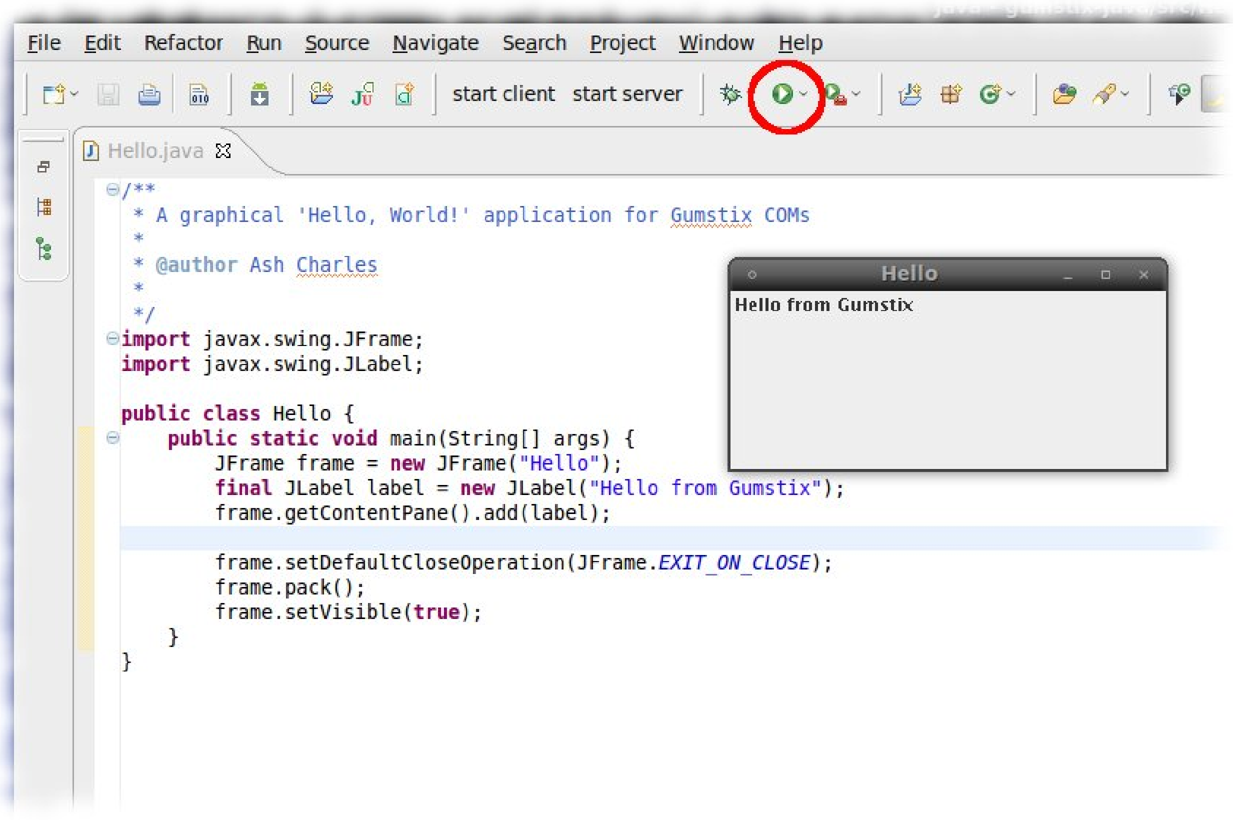
- Select File->New->Class
- Name your class *Hello*
- click "FINISH"

Add the following code to your class:

```
import javax.swing.JFrame;
import javax.swing.JLabel;
```

```
public class Hello {
        public static void main(String [] args) {

                JFrame frame = new JFrame("hello");

                final JLabel label = new JLabel("hello from Gumstix");

                frame.getContentPane().add(label);

                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                frame.pack();

                frame.setVisible(true);

        }
}
```

You should get something that looks like this:



```
/**
 * A graphical 'Hello, World!' application for Gumstix COMs
 *
 * @author Ash Charles
 *
 */
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Hello {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Hello");
        final JLabel label = new JLabel("Hello from Gumstix");
        frame.getContentPane().add(label);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

Press the green run button to test drive our application on our host machine.

Hint:

Eclipse has many auto-completion features. Simply typing `/**<Enter>` before a class creates a standard documentation block. Likewise, pressing `<Ctrl>+<Space>` provides context-specific auto-completion options. Many other hints for speedy development are explained in the tutorials here.

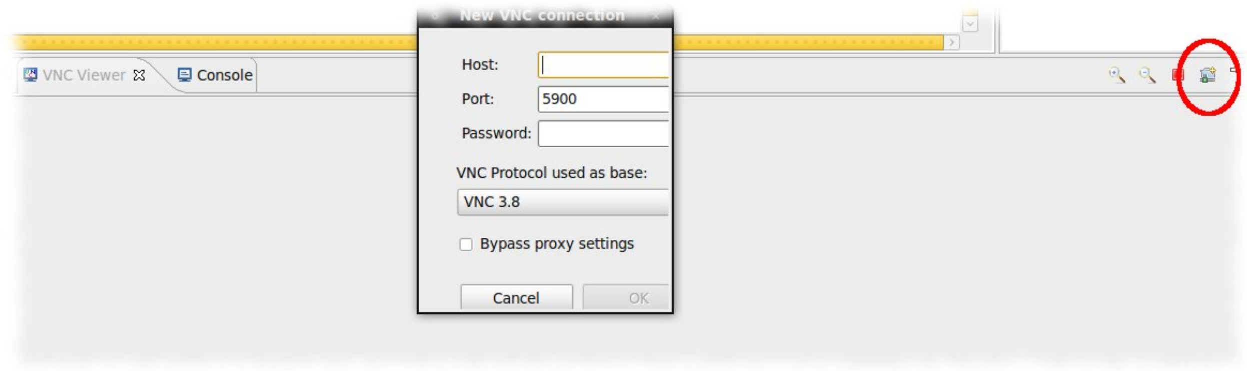
Step 7: Upload Code

In Step 5 we discovered the IP address of our Gumstix. With this, we can connect to our Gumstix using the SSH protocol which allows us to rapidly transfer files. For this step, navigate to **Window->Open Perspective->Other->Remote System Explorer** and click on the connection icon in the *Remote Systems* panel on the left. Select *SSH without shells* as the connection type and press *Next*. Put the internet address you found in Step 5 into the field marked *Host Name* and hit finish.
To test that everything is working correctly, right-click on the terminal item list under your new connection and select *Launch Terminal*. A console interface exactly like the USB console interface we set up in Step 3 should pop up.

Hint:

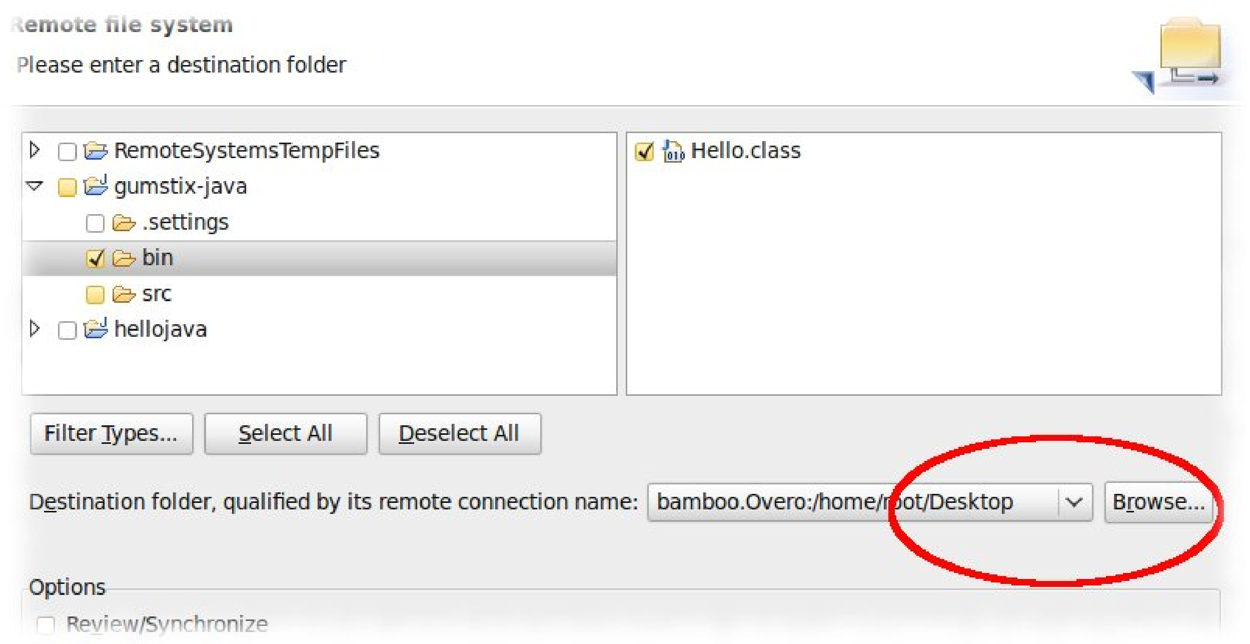
A 'Perspective' is an arrangement of windows and widgets, that is a specific view of the Eclipse workspace, designed for a particular task. If you wish to focus on any particular widget, double-click the tab in Eclipse; return to the previous view by double-clicking again.

By installing the VNC viewer plugin, we can get a graphical interface to our Gumstix system. Do this by adding the *Tools for mobile linux runtime* plugin **(Help->Install New Software**) from the main update site. To use, select the VNC viewer widget from **Window->Show View->Other->VNC Category->VNC** and click on the *New Connection* icon in the VNC Viewer panel. Again, the *Host* is the internet address of your Gumstix board.



Host:
Port: 5900
Password:
VNC Protocol used as base: VNC 3.8
☐ Bypass proxy settings

Finally, we need to upload our java code before we can run it. Switch back to the Java Perspective (**Window->Open Perspective->Java**) and right-click on our Java file and selecting Export. Choose the option to export file to a remote file system and choose to upload your *Hello.class* file.



Destination folder, qualified by its remote connection name: bamboo.Overo:/home/root/Desktop

You will need to *Browse* to find your Gumstix board; you should upload this code to your home directory on the Gumstix.

Step 8: Debug a Java Application

We can debug our java application remotely from Eclipse. To do this, we need a Java Virtual Machine capable of doing debugging. This time, the package we need isn't available in our package sources however we can fetch the package directly from a web site and install it using the same opkg package management tool.

```
wget http://bugcommunity.com/downloads/files/phoneme-
advanced-personal-debug_mr2-r1_armv6.ipk <Enter>
opkg install -force-depends phoneme-advanced-personal-
debug_mr2-r1_armv6.ipk <Enter>
```

Let's also update our Java libraries so we have access to the standard graphical libraries.

```
opkg install classpath-gtk <Enter>
```

Next, let's start our Java code in debugging mode on the Overo:

```
java-cdc -Xdebug -Xrunjwdp:transport=dHt_socket,
server=y,suspend=y,address=1234 Hello <Enter>
```

Finally, let's connect to our debugging session by navigating in Eclipse to **Run->Debug Configurations** and double-clicking on Remote Java Application to create a new configuration. Fill in the form as shown below using the internet address found in the previous step.

Step 9: Install a native C/C++ SDK

When we discussed Python and Java we noted that both are interpreted languages---neither needs to be compiled. C and C++ code requires compilation which can be done natively---on the Gumstix itself---or on a development machine using a cross-compiler. To do native compilation, you need to install a compiler as well as any required libraries. You'll need to be booting from a microSD card as this installation takes approximately 75MB of space.

```
$ echo 'src/gz angstrom-base http://www.angstrom-
distribution.org/feeds/unstable/ipk/glibc/armv7a/base' >
/etc/opkg/angstrom-base.conf
$ opkg update
$ opkg install task-native-sdk
```

Firstly, we are adding a new package repository to our package manager and updating our list of available packages before actually installing the complete package needed to do native development.

Let's create either of the following files in Eclipse, upload the code and then compile on the Gumstix using our newly installed compiler.

C++: hello.cpp	C: hello.c

`#include <iostream>`
 	
`using namespace std;`
 	
`int main ()`
`{`
`    cout << "Gumstix runs on C++" << endl;`
`    return 0;`
`}`
	`#include <stdio.h>`
	
	`int main ()`
	`{`
	`    printf("Gumstix runs on C\n");`
	`    return 0;`
	`}`

$ g++ -o hello hello.cpp Gumstix runs on C++	$ gcc -o hello hello.c Gumstix runs on C

<TODO: More exciting examples using on-board LEDs.>

Step 10: Use a C/C++ cross-compiler and debugger

In order to do C/C++ development on our host machine, we need to add in a plugin to Eclipse.
Also, we need to download an appropriate cross-compilation toolchain.
For now, I recommend the lovely instructions from

http://www.designarm.com/quickstart-guide/linux-software/eclipse.html & http://www.bugcommunity.com/wiki/index.php/BUG_Kernel_Development_with_Eclipse

Further Reading:

- Tutorials built into Eclipse
- Bitbake commander
- OpenEmebedded|Poky docs + Bitbake docs
- Eclipse Plugin Tutorial