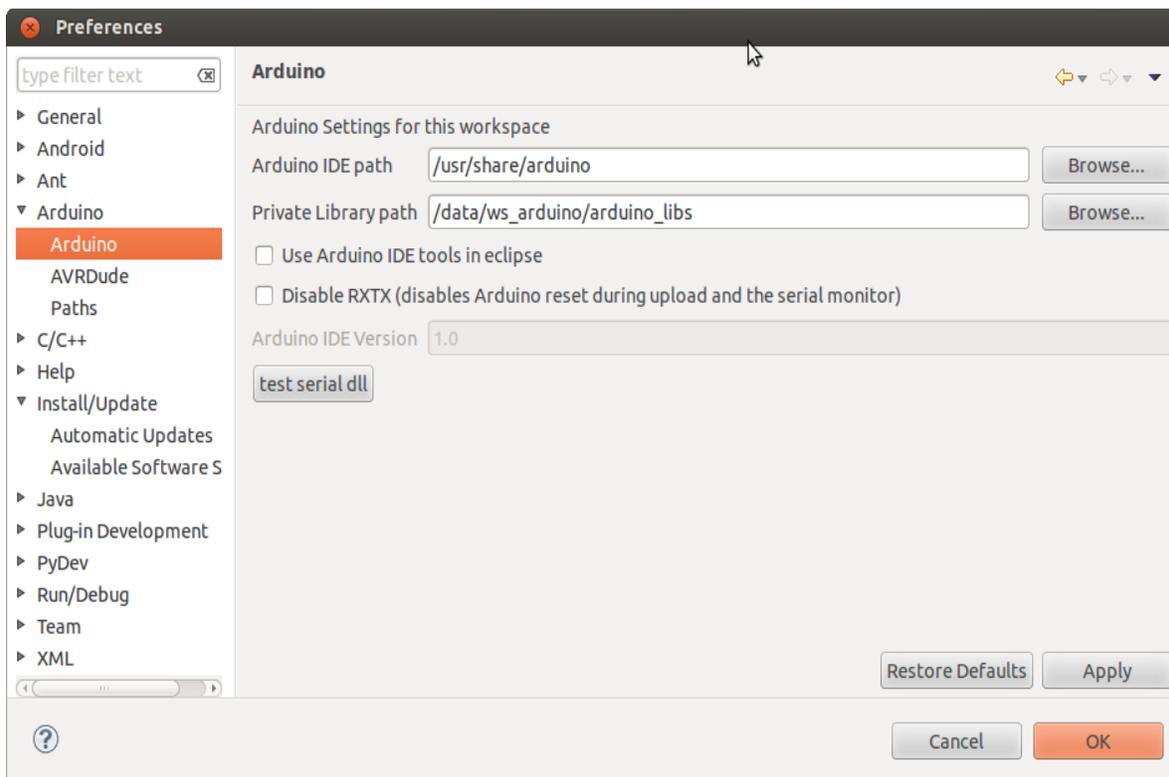


How To - Set Up Eclipse to compile Arduino sketches

1. Get the eclipse C/C++ Version from the website or update your eclipse from Help > Install New Software... and set <http://download.eclipse.org/tools/cdt/releases/indigo> as the software site. (Note: replace indigo by your eclipse distribution name)
2. Install the arduino eclipse plugin from the software site http://www.baeyens.it/eclipse/update_site/. If you encounter an error during installation try using the uncategorized arduino plugin (currently version 1.2.4) (for more help refer to <http://arduino.cc/forum/index.php?topic=79595.0> or <http://www.baeyens.it/eclipse/>)
3. Once the plugin is installed and eclipse restarted, go to Window > Preferences > Arduino.

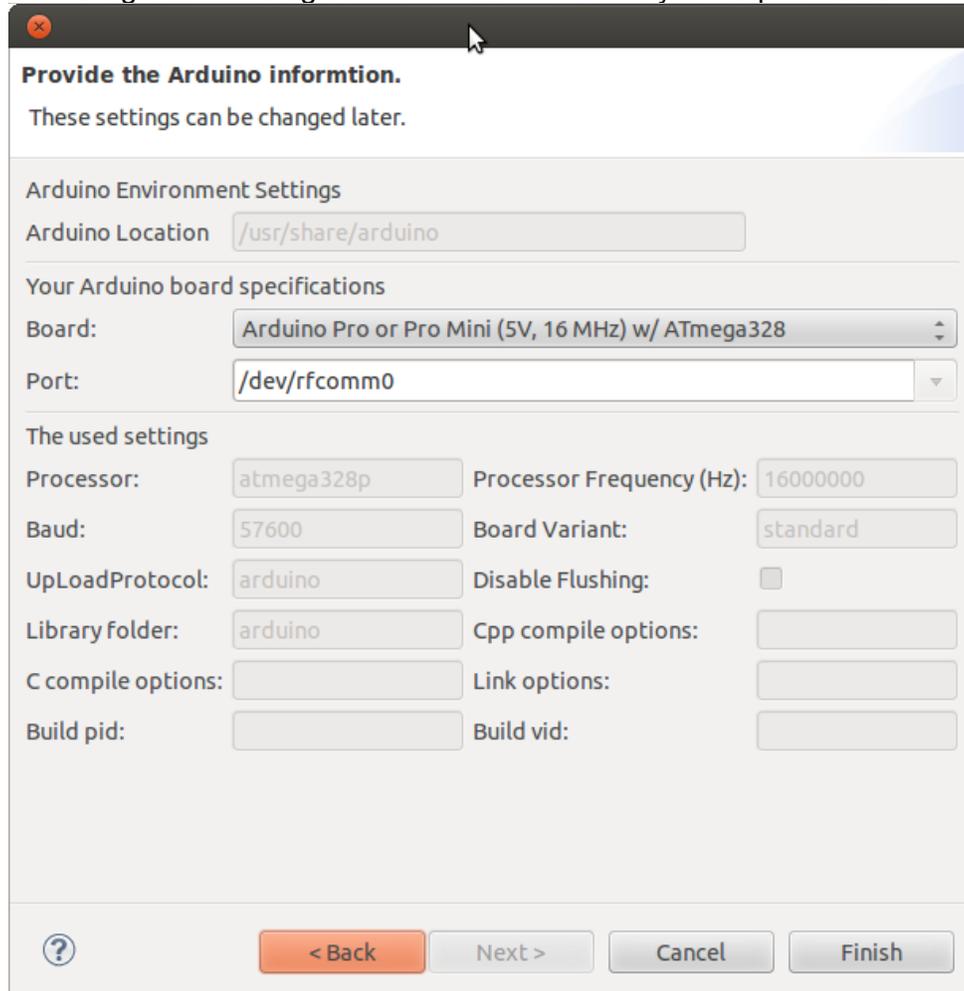


4. Assign the Arduino IDE path and the Private Library path. Under Linux, the Arduino files are by default stored in /usr/share/arduino. For the Private Library path you can choose any location where you want to store your globally used arduino libraries, but it is recommended to use the arduino library folder (/usr/share/arduino/libraries).
5. In order for eclipse to recognize *.ino and *.pde files we need to tell eclipse to process them as C++ files. Go to Window > Preferences > C/C++ > File Types, then press New... , add as Pattern *.ino and choose as Type C++ Source File. Repeat the same for *.pde

Note: If you encounter problems, or need more details, refer to <http://arduino.cc/forum/index.php?topic=79595.0>. I only included here only the basic information to set up Eclipse and Arduino.

How To - Create a new Sketch with Eclipse

1. Go to File > New > Project and select Arduino > New Arduino sketch.
2. At the Configuration Dialog set the Board and Port to your requirements.

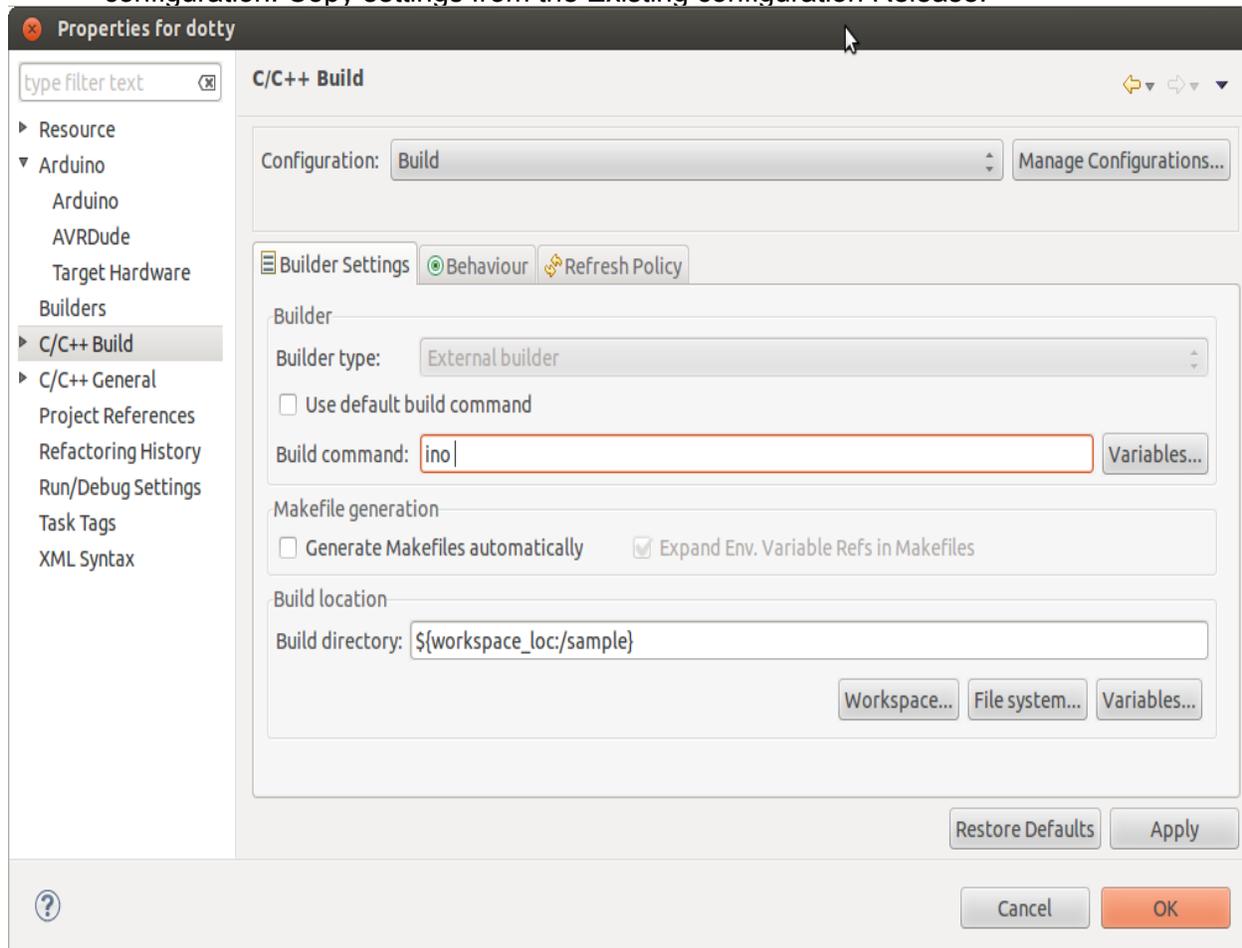


3. Once you press Finish, Eclipse will create two project folders. Once the folder for you project you just defined, and once the folder for the arduino board including default arduino libraries.
4. If you want to change the board or port go to Project > Properties > Arduino > Arduino. Note: if you change the board then eclipse will create a new project folder for the new board (if it didn't exist already)
5. There seems to be a problem Indexer concerning the Serial. If Serial is underlined as red although the code compiles, see item 'Eclipse tells me there are bugs in my code but it compiles ok' at http://www.baeyens.it/eclipse/Arduino_eclipse_plugin_FAQ.htm#bugs

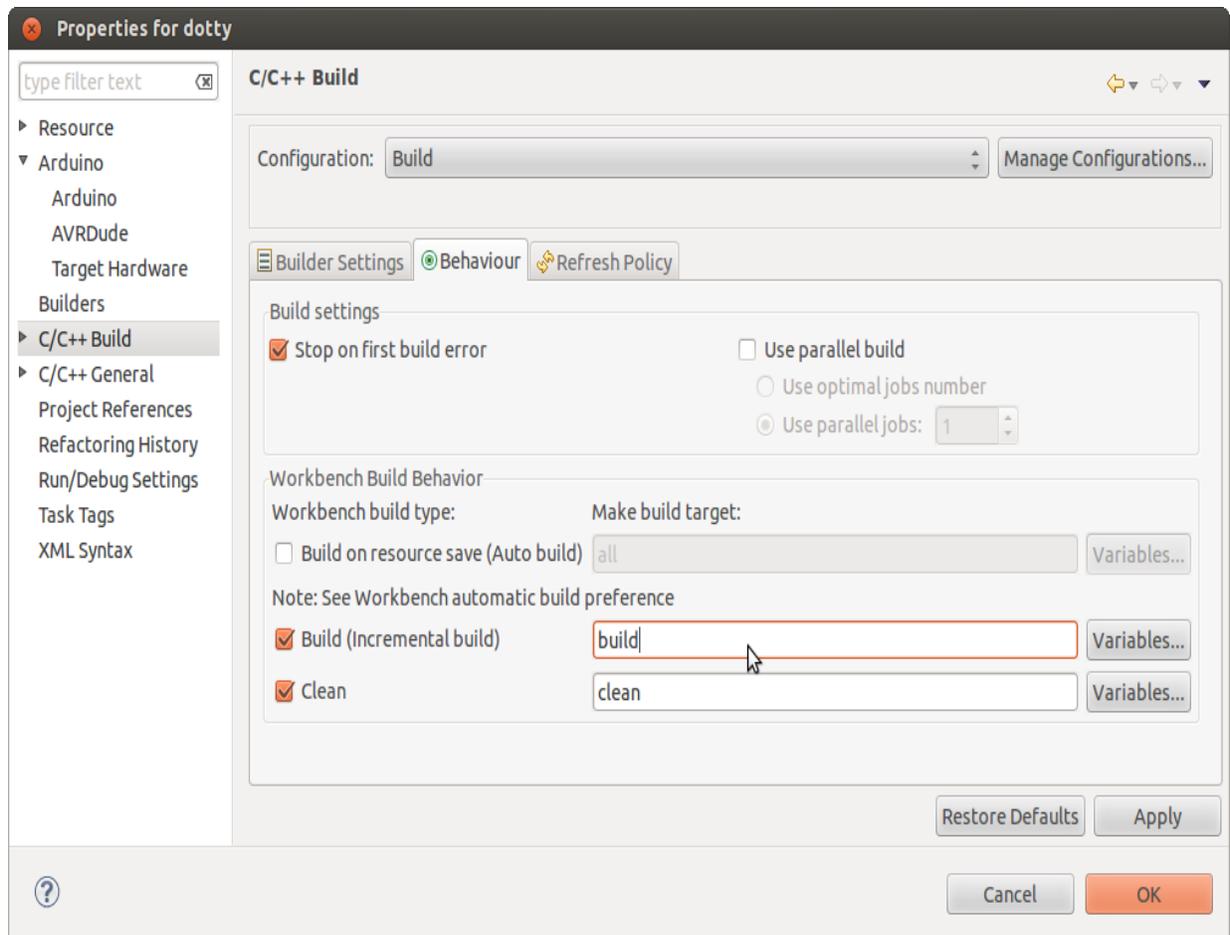
How To - Set up Eclipse to compile with ino

Ino is a command line script tool to compile and upload arduino sketches. Obtain the tool from <http://inotool.org> and follow the instructions to install. Make sure you initialised your project with the command `ino init` and copy your source files to the `src` folder before trying to build with ino.

1. To build the sketch with ino we need to define a new build configuration. Go to Project > Properties > C/C++ Build. Press on Manage Configurations and then New to add a new configuration. Copy settings from the Existing configuration Release.



2. Disable the Checkbox for 'Use default build command' and set the Build command to 'ino'
3. Disable the Checkbox for 'Generate makefiles automatically' and set the Build location to your project folder
4. Go to the tab Behaviour and replace the target for Build (Incremental build) with build. This is the parameter that will be passed to ino to build the sketch. Leave the Clean target as it is. (The clean target will be executed if you choose Project > Clean)



5. Press on Manage Configuration, choose the new configuration and press Set Active so that the sketch is built with ino.

How To - Use the Scripts for Wireless Programming in Eclipse

In order to establish a serial connection to the arduino, the scripts need root access, but calling sudo in a script doesn't work when the script is called from eclipse. Additionally, the serial monitor cannot be called in a console inside eclipse and colour is not displayed correctly. For these reasons, instead of executing the scripts in a console of eclipse, we instead start a console outside eclipse and let that console call the script.

1. If a console is called with a command as parameter, the console exits by default once the command is executed, so we wouldn't see the result of the command. To avoid this we need to change add a new profile that keeps the console open ones the command exits. (alternatively you can also change the default profile)
Open a Terminal Window, go to Edit > Profiles, then press New and give it a name like 'KeepOpen', based on the Default profile. (if you want to change the default profile instead, select Default and press Edit). In the new Window go to Title and Command

- and select 'Hold the terminal open' for 'When command exits:'
2. In Eclipse, go to Run > External Tools > External Tools Configuration, then select Program and press New.
 3. In the Main tab select for the Location a terminal program such as '/usr/bin/gnome-terminal'
 4. For the Working Directory set the project root folder
 5. In the Arguments we need to specify the following:
 - a. If you made a new Profile in step 1, we first have to tell the terminal to use that profile to execute the command with `--window-with-profile==XXX`, where XXX is the Profile Name. If you changed the default profile you can skip this parameter
 - b. Next we specify the command that should be executed. This starts with `-e` to tell the terminal that we want to execute a command. Next comes the script name we want to execute, e.g. `./upload.sh`
If you need to pass parameter to the script, for example if you want to specify the target name, the script and parameter need to be surrounded by `"`, e.g. `"./upload.sh -t RN42-CC33"`

Example:

```
--window-with-profile=KeepOpen -e ./scan.sh
```

or

```
--window-with-profile=KeepOpen -e "./upload.sh -t RN42-CC33"
```

6. In the Build tab, disable the 'Build before launch' as the upload script already includes a build step and for the rest of the scripts it is not necessary to build the sketch first.
7. In the Common Tab, disable the 'Allocate console' as we are executing our own console.
8. Save the Configuration under a new name. Now if you run this configuration a terminal console will be opened, then the sketch is built and uploaded to the arduino

External Tools Configurations

Create, manage, and run configurations

Run a program



- type filter text
 - Ant Build
 - API Use Report
 - Program
 - Connect
 - Scan
 - Serial
 - Test
 - Upload
- Filter matched 8 of 8 items

Name: Upload

Main Refresh Build Environment Common

Location:
/usr/bin/gnome-terminal
Browse Workspace... Browse File System... Variables...

Working Directory:
\${workspace_loc:/sample}
Browse Workspace... Browse File System... Variables...

Arguments:
-window-with-profile=KeepOpen -e ./upload.sh
Variables...

Note: Enclose an argument containing spaces using double-quotes (").

Apply Revert



Close Run