

Migrating your sketches to Arduino 1.0

Although it should not be difficult to get your existing sketches working with Arduino 1.0, that release has important changes you need to be aware of. The first thing you will notice when launching the software is the look of the IDE. Some icons have altered from previous versions of the software and there are changes and additions in the menus. The error messages when dealing with selecting boards have been improved and the new ADK and Ethernet board and the Leonardo boards have been added.

More significant are changes in the underlying core software and libraries. The stated purpose of 1.0 is to introduce disruptive changes that will smooth the way for future enhancements but break some code written for older software. New header files mean that older contributed libraries will need updating. Methods in Ethernet and Wire have been changed and there are subtle differences in the print functionality.

New functionality has been added to Streams (the underlying class that anything that uses `.print()` statements), Ethernet, Wire (I2C) and low level input/output.

Improvements have been made to the way libraries handle dependencies and to simplify the support for new boards. Because of these changes, third party libraries will need updating, although many popular ones may already have been updated.

The file extension used for sketches has been changed from `.pde` to `.ino` to differentiate Processing files from Arduino and to remove the inconvenience of accidentally opening a file in the wrong IDE.

Sketches opened in the 1.0 IDE will be renamed from `.pde` to `.ino` when the file is saved. Once renamed, you will not be able to open them in older versions without changing the extension back.

If you want to use Arduino 1.0 alongside an earlier release on the same computer then you may want to make copies of all your sketches into separate folders so you can have one branch for 1.0 and another for the earlier release.

Here is a summary of changes for older code to compile in 1.0:

print (`byte`) now prints the integer value of the byte as ASCII characters, previous releases sent the actual character. This affects Serial, Ethernet, Wire or any other library that has a class derived from the Print class:

Change

```
Serial.print(byteVal);
```

To

```
Serial.write(val); //send as char
```

The `BYTE` keyword is no longer supported:

Change

```
Serial.print(val, BYTE)
```

To

```
Serial.write(val); //sends as char
```

The write methods now return a value (`size_t`) so if you have created a class derived from `print` you need to change the write method from

```
void write
to
size_t write
```

Wire (I2C) methods have been renamed::

```
Change
Wire.send()
To
Wire.write()
```

```
Change
Wire.receive()
To
Wire.read()
```

You now need to specify the type for literal constant arguments to write, for example:

```
Change write(0x10) to write((byte)0x10)
```

Ethernet classes and methods have been renamed

```
Change
client client(server, 80)
To
EthernetClient client;
```

```
Change
if(client.connect())
to
if(client.connect(serverName, 80)>0)
```

Note that `client.connect` should test for values `> 0` to ensure that errors returned as negative values are detected.

```
Change
Server server(80)
To
EthernetServer server(80)
```

```
Change
UDP
To
EthernetUDP
```

Arduino 1.0 introduced a simple parsing capability to enable finding and extracting strings and numbers from any of the objects derived from Stream, such as: Serial, Wire and Ethernet. These functions include:

- **find(char *target);**
- **findUntil(char *target,char *term)**
- **readBytesUntil(term,buffer,length);**
- **parseInt();**
- **parseFloat();**

See the online documentation for more about these useful functions.

Libraries: If your sketch includes any libraries that have not been designed for 1.0 then you will need to change the library if it uses any of the old header files that have been replaced with the new Arduino.h file.

If any of the following header files are included:

Change any of these includes:

```
#include "wiring.h"  
#include "WProgram.h"  
#include "WConstants.h"  
#include "pins_arduino.h"
```

to:

```
#include "Arduino.h"
```

You can use a conditional include (see recipe 17.6) to enable libraries to also compile in earlier versions. For example:

Change

```
#include "WProgram.h"
```

to

```
#if ARDUINO >= 100  
#include "Arduino.h"  
#else  
#include "WProgram.h"  
#endif
```

Copyright Michael Margolis 19 September 2011

These note are based on Arduino 1.0-rc1