Installing and Using the Arduino IDE for chipKIT Development

Arduino IDE

The Arduino IDE is a software tool that is used to develop C and C++ programs for the Arduino and chipKIT development board. IDE is an acronym for Integrated Development Environment.

Windows

The Arduino IDE is compiled to run on a Windows PC, MAC OSX and Linux (as well as possible others). This section of the lab is written explicitly for Windows. This was written using the Windows 10 operating system.

For instructions on installing Arduino IDE on a Mac computer visit the following site:

https://www.arduino.cc/en/Guide/MacOSX

For instructions on installing Arduino IDE Linux computer visit the following site:

http://playground.arduino.cc/Learning/Linux

Where to find the Arduino IDE

These labs were tested using version 1.6.7 of the Arduino IDE. Download the zip file from this page by clicking on the link text “Windows ZIP file for non admin install” found on the following page:

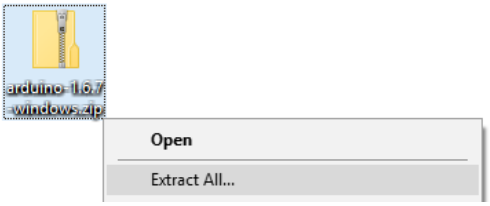
https://www.arduino.cc/en/Main/Software

The Arduino IDE is a fast changing open source tool and there are many versions to choose from. Any version after 1.6.9 should work with chipKIT-core. If version listed at the top of the page is not 1.6.7, then you should be able to find a list of previous released versions of the software here:

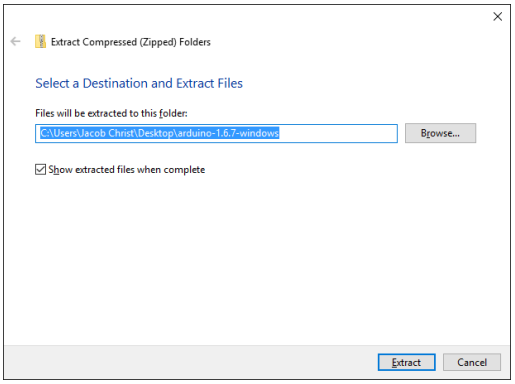
https://www.arduino.cc/en/Main/OldSoftwareReleases#previous

Unzipping the file in Windows

The file you have downloaded, regardless of the operating system, is a compressed file. In Windows this file can be extracted by right clicking on the downloaded file and picking the "Extract All..." item on the pop-up menu.



A new dialog will appear that asks for the location to extract the files too. This location should be your desktop.

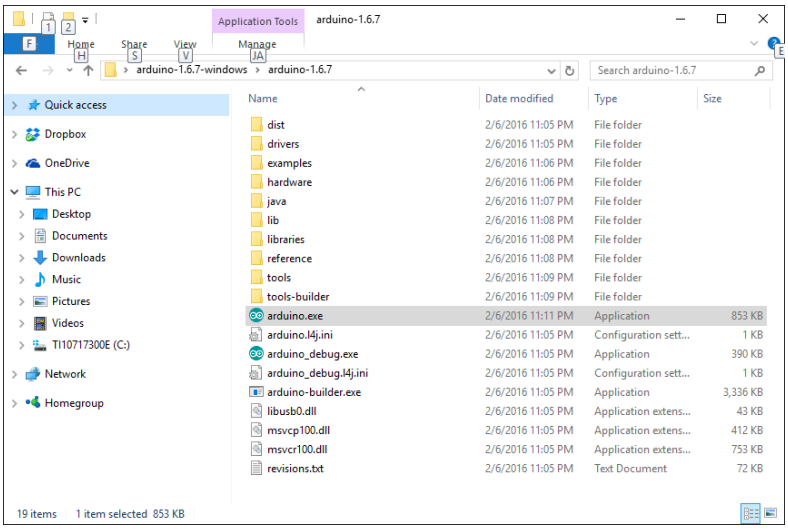


Click the "Extract" button to extract the Arduino IDE. This may take a few minutes due to the large size of the program. When complete this will result in a new folder on your desktop called "arduino-1.6.7-windows" which will contain the extracted development tools.

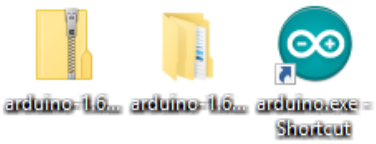
The Arduino IDE should work from any folder but for consistency in this manual it will be assumed that the Arduino IDE will be in a folder onto the Desktop.

Creating a Shortcut

You could easily run Arduino IDE by just double clicking on arduino.exe file in the unzipped folder, however for the class we will make a shortcut to the executable by right clicking on arduino.exe and selecting copy then right clicking on the desktop and pasting as a shortcut.

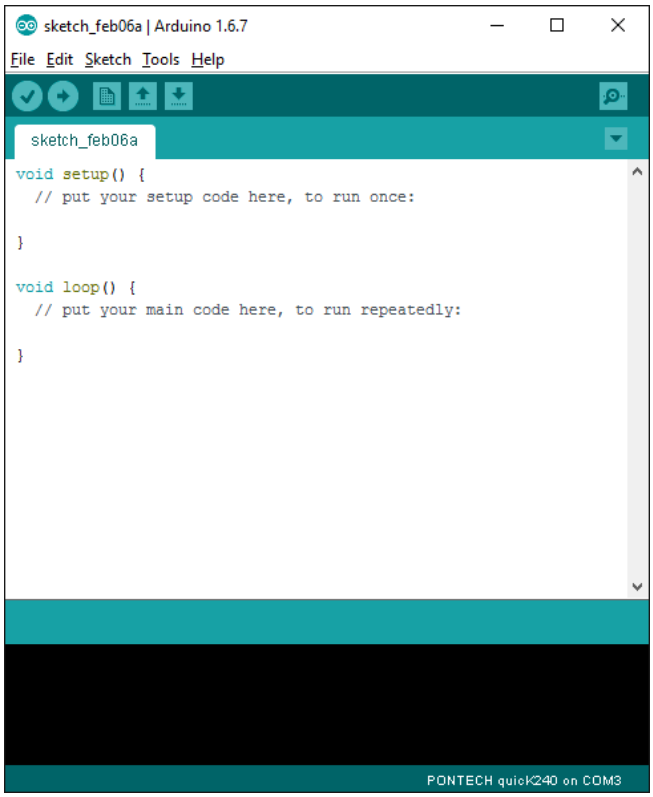


Your desktop will now look something like this:



How to run Arduino IDE

Double click on "arduino.exe - Shortcut" icon on your desktop and you should get a window that looks something like the one pictured below.



Part 4 Installing chipKIT-core into Arduino IDE and chipKIT drivers

Arduino IDE as a Multiplatform Tool

The Arduino IDE, as of version 1.6.x, is a multiplatform Arduino Compatible Integrated Development Environment. "Multiplatform" implies that it works with more than one platform. A platform in the embedded development world usually implies a family of microprocessors chips that have a common microprocessor architecture. You can think of different platforms kind of like different construction toys. For example, Lego bricks and Mega Bloks bricks. Both can be used to construct sculptures but they may not be interchangeable. That is to say they are different platforms for construction. In the case of embedded development, the tools used to convert your program source code to machine code are usually for a specific platform or architecture of microprocessor.

chipKIT-core

Platform tools for the Arduino IDE are implemented in what are called cores. The Arduino IDE comes with the 8-bit Atmel AVR core pre-installed. This core works with the chips on the original Arduino Uno and related boards. In other words, the 8-bit Atmel AVR platform. Besides the Arduino 8-bit Atmel AVR core there is also an Arduino 32-bit ARM core that does not come pre-installed as well as many others. The core we are going to be using is called the chipKIT-core and it targets the 32-bit Microchip PIC32 microcontrollers (which are based on the MIPS 4000 microprocessor). Besides the listed cores in the IDE and the chipKIT-core there are cores made by others and probably more on the way. The reason the cores for all these different platforms are not included by default is that they are quite large. If they had been included, it would mean you would need extra hard drive storage and more importantly extra time to download cores you are not planning on using. What a waste!

The Arduino IDE with the chipKIT-core installed utilizes the gcc open source C++ cross compilers. A cross compiler is a tool that lets you generate machine code for one computer architecture on another. The chipKIT-core cross compiler has similar functionality that you would find in a C++ compiler for writing applications for a PC but utilizes libraries specifically written for chipKIT embedded hardware. Programming is done utilizing the C++ language but automatically included libraries allow a beginner to ignore (at least in the beginning) the complexity associated with C++ so that they can dive in and get started creating fast.

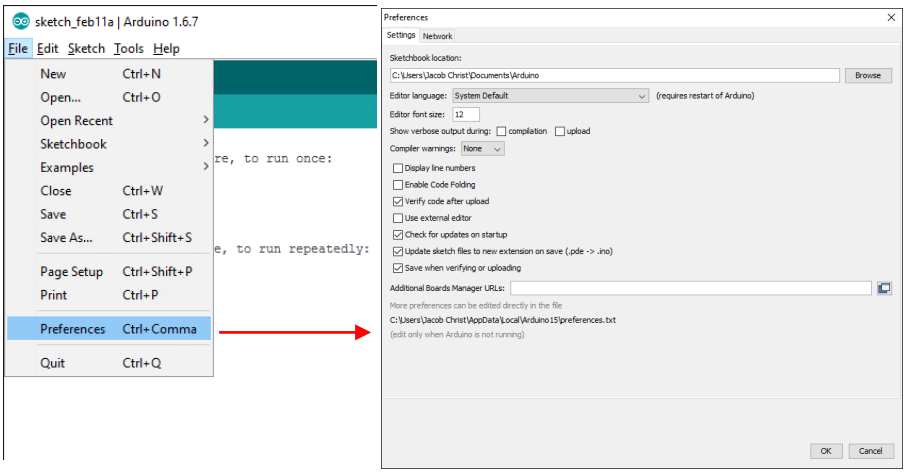
To use an additional core the Arduino team created a mechanism within the IDE to allow you to simply install a core by copying and pasting a URL in the IDE and then go to a “Board Manager” to download selected versions of additional cores.

Installing chipKIT-core

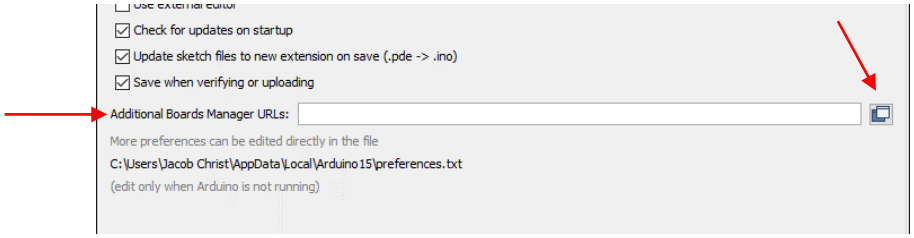
From within the Arduino IDE, click on the following menu items:

File->Preferences (for Windows) or Arduino->Preferences (for a MAC)

This will open the preferences dialog box.



Within the preferences dialog box look for the text entry field called "Additional Boards Manager URLs:". Click the icon to the right of the text field to open a new dialog box to allow you to edit all additional board manager URLs.

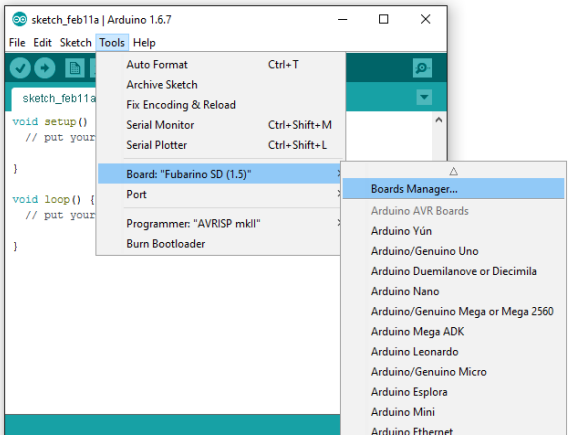


Each URL in the “Additional Boards Manager URLs” dialog must be on a line by itself. With this dialog box open copy and paste onto a blank line the following URL to enable downloading of the chipKIT-core:

https://github.com/chipKIT32/chipKIT-core/raw/master/package_chipkit_index.json

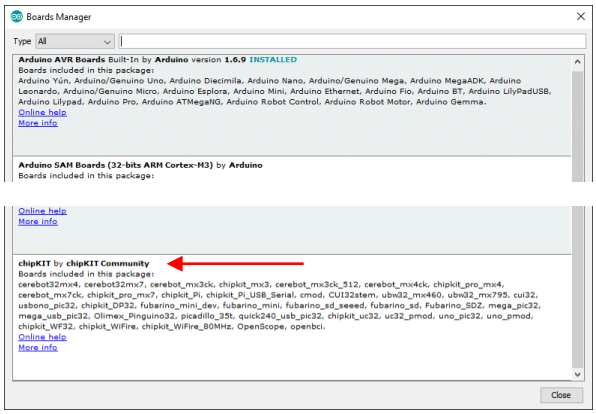
The Arduino IDE lets you have many different cores loaded into the IDE as long as each URL is on a separate line. Click OK to close the Additional Boards Manager URLs dialog box and then click OK again to close the Preferences dialog box.

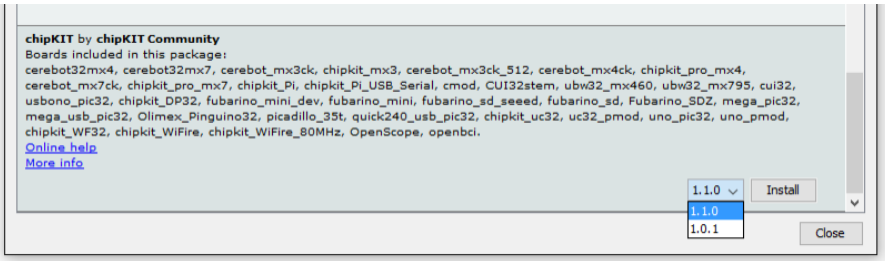
Now select the Tools->Board->Board Manager menu from the Arduino IDE, and it will open up the Boards Manager window. Once open you will see several other cores that are not installed by default. It may take a few seconds to retrieve the chipKIT information from the internet, but eventually the chipKIT-core will show up and be presented in the list of cores.



Scroll down until you see the “chipKIT by chipKIT Community” board listing. Click once on any of the text in this section. Once you click a dropdown menu and an Install button will appear.

Select **<u>version 1.3.1</u>** then press the Install button. It will take some time to download all of the chipKIT components and install them, but when it's done, you can click the Close button to close the Board Manager window.





Once complete verify the chipKIT-core is installed by looking in the boards menu for the chipKIT boards. Do this from the Tools->Board menu and scroll down until you see the chipKIT boards.

As new versions of the chipKIT-core files are released, you will be able to update your chipKIT-core files from inside the Arduino IDE. During this class, refrain from upgrading to be assure that all your labs work.

This section of the lab manual was largely based on documentation on the chipKIT wiki. In addition to this method there are also other ways to install a chipKIT core which are documented here:

http://chipkit.net/wiki/index.php?title=ChipKIT_core

FTDI USB Drivers for chipKIT Boards

FTDI VCP drivers are needed for boards that have a very popular USB to Serial converter chip made by FTDI. Some examples of these boards are the WiFire, Uno32, the Max32. The FTDI VCP driver can be downloaded from their web page:

http://www.ftdichip.com/Drivers/VCP.htm

Look for the operating system you are using to download the latest version of the driver. In windows you will download a zip file. Save this file to your desktop and right click on it to extract the files.

Run the executable and the driver will install within a few seconds.

Get Blink Sketch to run

The Arduino IDE

If not already running, double click on "arduino.exe - Shortcut" icon on your desktop and you should get a window that looks something like the one pictured to the right. Some notable items on the IDE window are pointed out below.

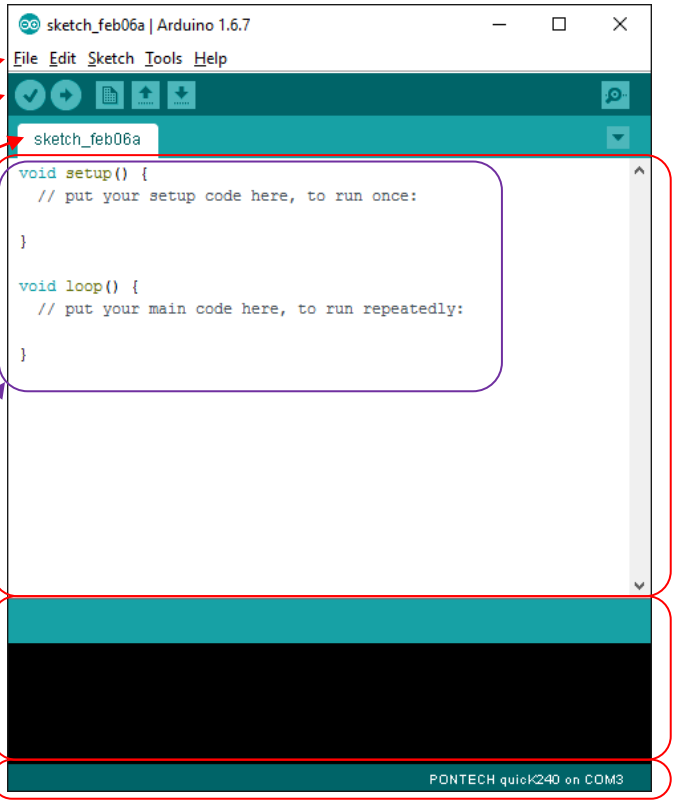
Menus

Icon Bar short cuts

Filename

Text editor where we edit our program

Code (and comments) rendered in colorful text



Compiler and Program Transfer Status

Selected target board and COM port Status

The Icon Bar

The most freqently used buttons when you are trying to get a program working are on the Icon Bar. A brief description of each button is given here:

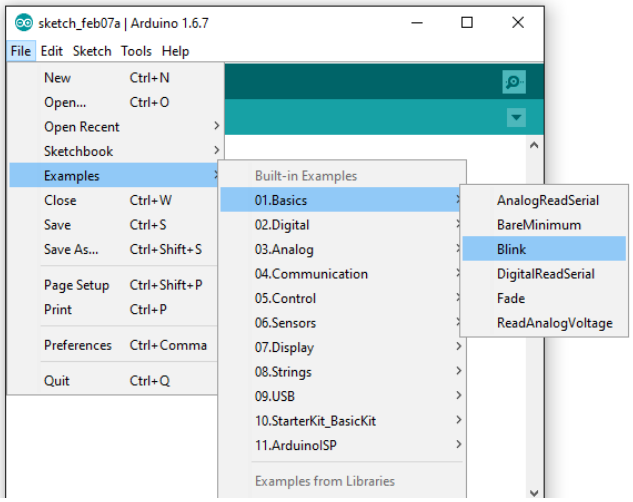
	Verify: compiles your code. Any errors are shown in the black status box at the bottom of the IDE.		Upload: Compiles your code then attempts to write the compiled machine code to a connected board.
	New: Creates a new sketch.		Open: Opens an existing sketch.
	Save: Saves your work.		Serial Monitor: Open a serial terminal to interact with your running program.

Sketches

All programs written using the Arduino IDE are called sketches. The idea behind the sketch is that Arduino IDE makes writing a programming as simple as an artist picking up a paper and pencil to draw a sketch and hence the name.

Example Sketches

One way to learn how to sketch is by looking at example sketches created by others. The Arduino IDE comes with many example sketches to help you understand how to write programs for Arduino and Wiring compatible boards such as chipKIT. These sketches are accessible from the File menu by moussing over Examples. As you can see from the image to the right in this installation there are many examples to choose from. We will start at the beginning and try a "01.Basic" sketch called "Blink". Mouse to and click on the "Blink" sketch to open the example file.



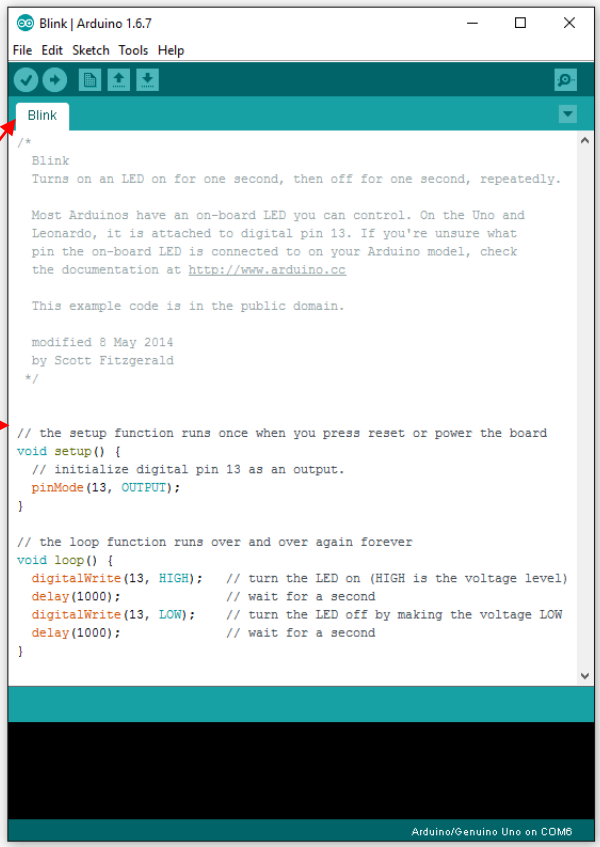
Blink Sketch

When you select the "Blink" example a new IDE window will open leaving your unused sketch behind the new example sketch. Notice that you can resize the window so that you can see the whole sketch on the screen in a single glance.

Two things should look different in this new window.

The tab will be named "Blink"

The text shown in the window is now filled with the code for this loaded sketch.



Notice also that the IDE editor has color coded the "source code" to make it easy to distinguish the parts of the code. The color code is listed roughly below.

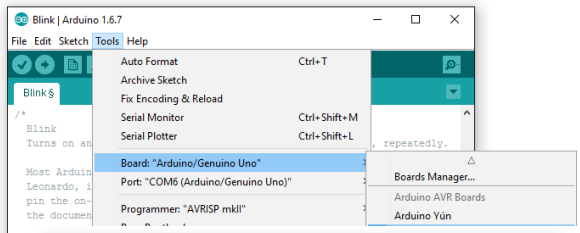
Gray: User comments, orange: function calls, olive: function definitions, blue: qualifiers and constants.

The “out of the box” functionality of the "Blink"

sketch is to blink a LED connected to the board at a rate of 1/2Hz (one second on then one second off).

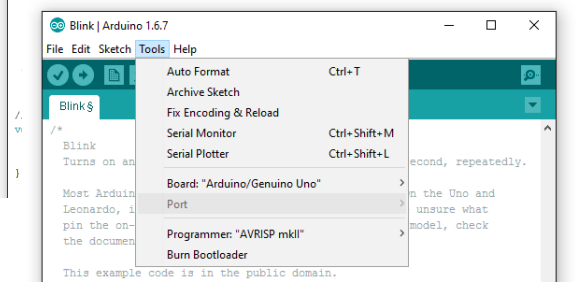
Selecting the Board

Now that we have a useful program loaded into the IDE we must next select the board we wish to run our program on. This is done from the Tools->Board menu select the “chipKIT WiFire” board.

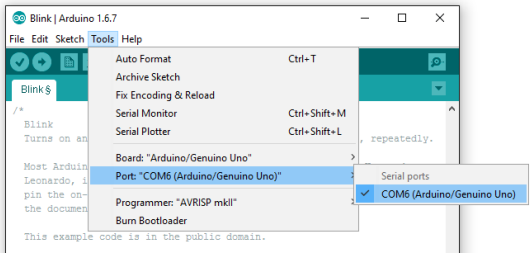


Selecting the COM Port

After selecting the target board, we must next select the COM Port the board is mapped to on our computer. This is done through the Tools->Port menu. Sometimes the board or COM port we are looking for is not present in the list of ports. When this happens the “Ports” menu item will be grayed out as shown to the right. This can happen when the driver is not installed correctly or the board is not plugged in to our computer.



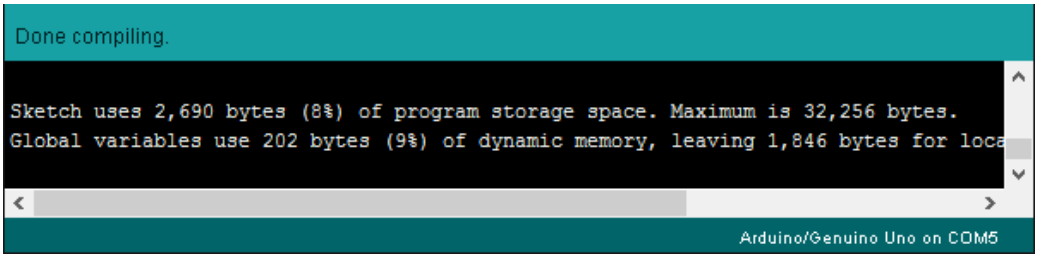
If the COM Port for our board is present, select it so that we can test our program.



Compiling the Sketch

Before we actually try to program our board, we need to verify that our program is syntactically correct. As mentioned previously the "Verify" button will compile the program. When you press it the bottom of the Arduino IDE interface will indicate that the program is compiling by displaying "Compiling sketch..." and showing a progress bar.

When finished compiling the stats will change to "Done compiling." and if everything was syntactically correct the size of the sketch will be displayed.



Upload and Test

Next press the upload icon button and observe the LED blinking on the board.