



# arduino for visual studio (<https://www.visualmicro.com/>)

Edit and debug 100's of Arduino (or compatible) boards and 1000's of libraries. Uses the same configuration as the arduino ide (+ advanced mode). Hardware and software debugging (usb, jtag, ice, gdb, jlink etc.), serial monitors, shared projects, editable libraries and much more

## Arduino IDE and GDB for Visual Studio

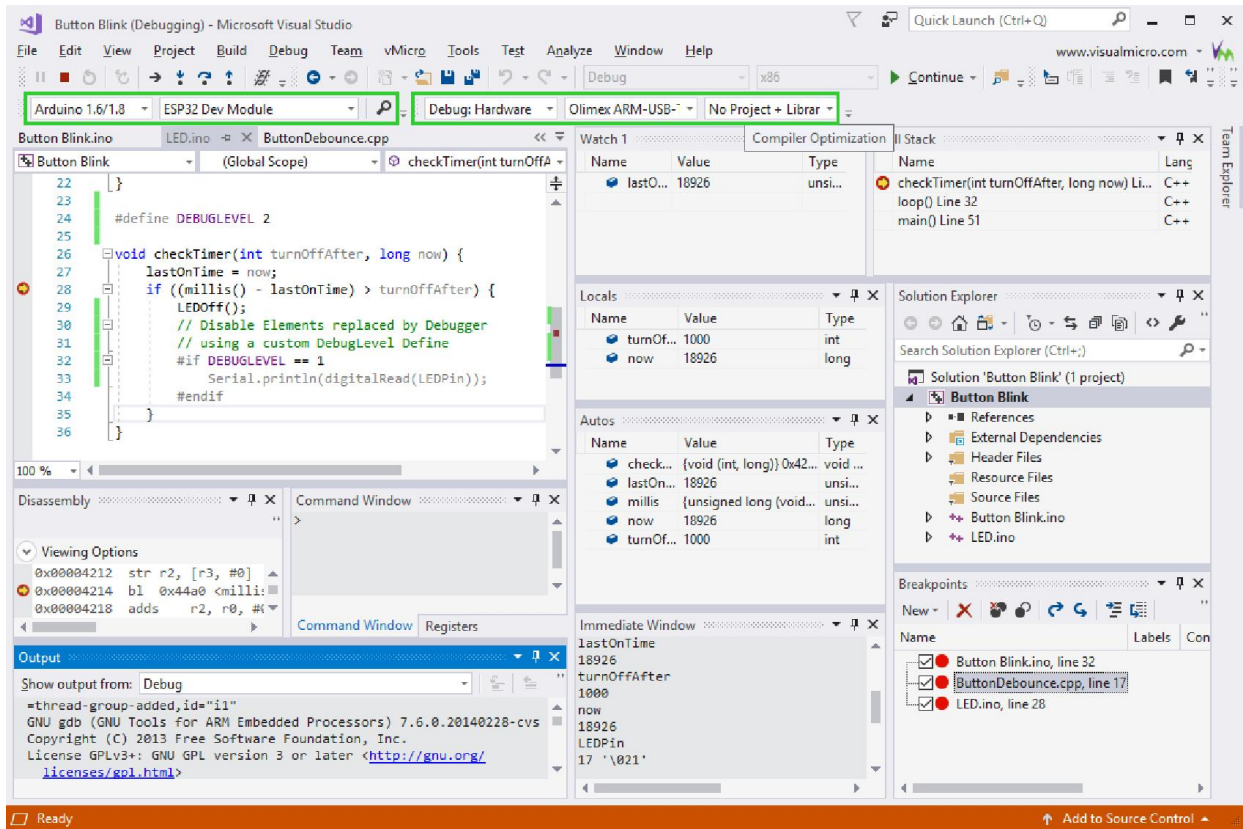


*common links:* [release notes](https://www.visualmicro.com/page/Visual-Micro-Product-Version-History-Fixes-and-Additions.aspx) (<https://www.visualmicro.com/page/Visual-Micro-Product-Version-History-Fixes-and-Additions.aspx>) | [next release, alternative download, previous releases and email notifications](https://www.visualmicro.com/forums/YaBB.pl?board=VS_ARDUINO_EXT_RELEASES) ([https://www.visualmicro.com/forums/YaBB.pl?board=VS\\_ARDUINO\\_EXT\\_RELEASES](https://www.visualmicro.com/forums/YaBB.pl?board=VS_ARDUINO_EXT_RELEASES)) | [Setup](https://www.visualmicro.com/page/User-Guide.aspx?doc=First-steps.html) (<https://www.visualmicro.com/page/User-Guide.aspx?doc=First-steps.html>) | [Correct Way to Open Projects](https://www.visualmicro.com/post/2018/04/19/The-best-way-to-open-Arduino-projects-in-Visual-Studio.aspx) (<https://www.visualmicro.com/post/2018/04/19/The-best-way-to-open-Arduino-projects-in-Visual-Studio.aspx>) | [Plotting and Charting](https://www.visualmicro.com/page/User-Guide.aspx?doc=Multiplot.html) (<https://www.visualmicro.com/page/User-Guide.aspx?doc=Multiplot.html>) | [Atmel Studio](https://www.visualmicro.com/page/Arduino-for-Atmel-Studio-7.aspx) (<https://www.visualmicro.com/page/Arduino-for-Atmel-Studio-7.aspx>) | [Create Project](https://www.visualmicro.com/page/User-Guide.aspx?doc=Your-First-Project.html) (<https://www.visualmicro.com/page/User-Guide.aspx?doc=Your-First-Project.html>) | [Create Arduino Library](https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx) (<https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx>) | [Share Code Between Projects](https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx) (<https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx>) | [Faster Builds](https://www.visualmicro.com/post/2016/01/14/How-to-improve-compile-times-dramatically.aspx) (<https://www.visualmicro.com/post/2016/01/14/How-to-improve-compile-times-dramatically.aspx>) | [Extend Intellisense](https://www.visualmicro.com/page/Extending-Visual-Studio-Intellisense.aspx) (<https://www.visualmicro.com/page/Extending-Visual-Studio-Intellisense.aspx>) | [ESP32](https://www.visualmicro.com/page/How-to-install-Esp32-for-Arduino-and-Visual-Micro.aspx) (<https://www.visualmicro.com/page/How-to-install-Esp32-for-Arduino-and-Visual-Micro.aspx>) | [ESP8266](https://www.visualmicro.com/page/esp8266.aspx) (<https://www.visualmicro.com/page/esp8266.aspx>) | [ESP32 Debugging](https://www.visualmicro.com/page/ESP32-Debugging.aspx) (<https://www.visualmicro.com/page/ESP32-Debugging.aspx>) | [Difference between .ino and .cpp](https://www.visualmicro.com/page/User-Guide.aspx?doc=INOs-and-CPPs.html) (<https://www.visualmicro.com/page/User-Guide.aspx?doc=INOs-and-CPPs.html>) | [Custom Build Events](https://www.visualmicro.com/post/2017/03/20/Arduino-Custom-Build-Events.aspx) (<https://www.visualmicro.com/post/2017/03/20/Arduino-Custom-Build-Events.aspx>) | [Command Line Interface](https://www.visualmicro.com/page/Visual-Micro-Command-Line-Interface.aspx) (<https://www.visualmicro.com/page/Visual-Micro-Command-Line-Interface.aspx>) | [Custom Debug Output](https://www.visualmicro.com/post/2018/04/14/Custom-Arduino-Breakpoint-Debug-Output.aspx) (<https://www.visualmicro.com/post/2018/04/14/Custom-Arduino-Breakpoint-Debug-Output.aspx>) | [CLI](https://www.visualmicro.com/page/Visual-Micro-Command-Line-Interface.aspx) (<https://www.visualmicro.com/page/Visual-Micro-Command-Line-Interface.aspx>) | [Teensy](https://www.visualmicro.com/page/Teensy-for-Microsoft-Visual-Studio.aspx) (<https://www.visualmicro.com/page/Teensy-for-Microsoft-Visual-Studio.aspx>) | [Arduino Build Hooks](https://www.visualmicro.com/page/Arduino-Build-Events-and-Hooks.aspx) (<https://www.visualmicro.com/page/Arduino-Build-Events-and-Hooks.aspx>) | [Atmel Studio 7](https://www.visualmicro.com/page/Arduino-for-Atmel-Studio-7.aspx) (<https://www.visualmicro.com/page/Arduino-for-Atmel-Studio-7.aspx>) | [Override Build Properties](https://www.visualmicro.com/page/Local-Project-Boardtxt-Examples.aspx) (<https://www.visualmicro.com/page/Local-Project-Boardtxt-Examples.aspx>) | [Debug](https://www.visualmicro.com/page/User-Guide.aspx?doc=Debugging-Walkthrough-Start.html) (<https://www.visualmicro.com/page/User-Guide.aspx?doc=Debugging-Walkthrough-Start.html>) | [Esp FreeRTOS](https://esp32.com/viewtopic.php?f=13&t=5859) (<https://esp32.com/viewtopic.php?f=13&t=5859>) | [Custom Cores](https://www.visualmicro.com/forums/YaBB.pl?num=1526481480) (<https://www.visualmicro.com/forums/YaBB.pl?num=1526481480>) | [freeRTOS esp-idf on ESP32](https://esp32.com/viewtopic.php?t=5859) (<https://esp32.com/viewtopic.php?t=5859>) | [Auto Increment Version \(build events\)](https://www.visualmicro.com/forums/YaBB.pl?num=1545471409) (<https://www.visualmicro.com/forums/YaBB.pl?num=1545471409>)

### What is Visual Micro?

Visual Micro is a plugin for Microsoft Visual Studio (*with C++ installed*). Visual Micro helps you create Arduino compatible cross-platform programs for hundreds of different Arduino compatible micro-controllers. The Visual Studio and Atmel Studio IDEs are available for free for full time students. Visual Micro also includes multiple serial monitors, simple pre-configured hardware (GDB) and alternative unique full speed software debugger.

When writing code with Visual Micro, if you adhere to the Arduino.cc rules then the code you create will remain Arduino IDE compatible. Use the Arduino compatible library and board managers to discover and download hundreds of Arduino compatible boards and libraries. Visual Micro can be installed from within an IDE using the "Extension Manager" or from the Microsoft Gallery. Full control over tool bar visibility (auto/manual) enables uncluttered or easy access modes.



([https://www.visualmicro.com/pics/Debug-Break-ESP32-OlimexARMUSBTiny-VSOnly\\_Light.png](https://www.visualmicro.com/pics/Debug-Break-ESP32-OlimexARMUSBTiny-VSOnly_Light.png))

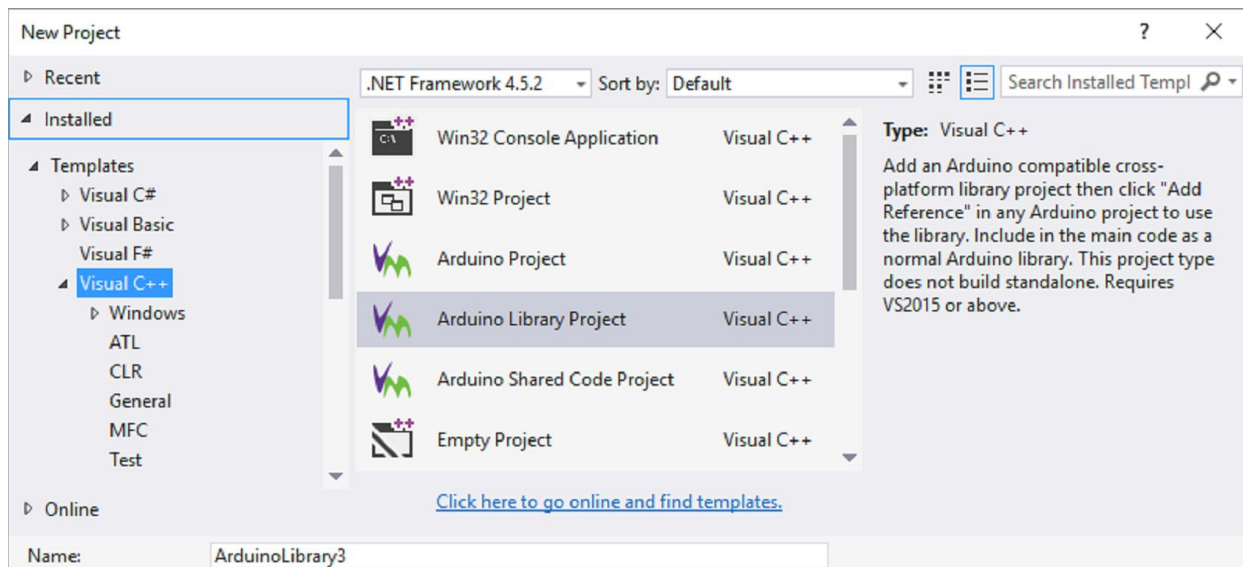
note: VS2019 tool window issues? Switch off "Tools>Environment>General>Optimize rendering for screens with different pixel densities". tip: Intellisense and GoTo code definition can be improved without affecting the build by clicking "vMicro>Show Hidden Files".

The free Microsoft Visual Studio Community Edition IDE is fully supported. The Microsoft IDE is free for independant commercial and non-commercial developers, open source projects and students. Visual Micro is free for students (alongside daily nag screen) otherwise starts at \$30. A 90 day evaluation of the software is allowed prior to purchase.

Common Build Error Note 1 - Currently Visual Micro applies weak logic to the auto generation of prototypes for functions directly inside .ino code. For example, functions defined inside negative #if conditions will be generated when the should not be. A workaround is to switch off "Prototype Generation" in the right click "Project Properties" tool window. Function prototypes can be added manually if/as required. This issue will be resolved shortly.

Common Build Error Note 2 - Some Arduino projects are designed to override the concept of using .ino code and instead use a .cpp file. The Marlin project is an example of this. For Visual Micro , the .cpp file name must be ProjectName.cpp where as in the Arduino IDE the file name will be ProjectName.ino.cpp. Visual Micro will shortly provide more flexibility and auto detection of the name of the "override" file. In the meantime please keep this in mind.

File - New Project - Wizard



Team capabilities of Visual Micro have been extended in recent versions along with better compatibility with the Arduino 3rd party hardware specification. Using automatic intellisense path management alongside cross-platform shared c++ projects, teams can build Arduino compatible programs without treading on other feet!

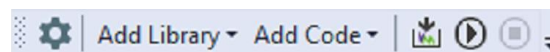
Arduino code created with Visual Micro can be shared with the windows, linux and mac Arduino community. Visual Micro adheres to the standards created by the various hardware manufacturers, such as arduino.cc. Put simply, this means that newly released hardware can be programmed in Visual Micro without need for a software update from Visual Micro. Visual Micro even uses the same configuration as the Arduino IDE!

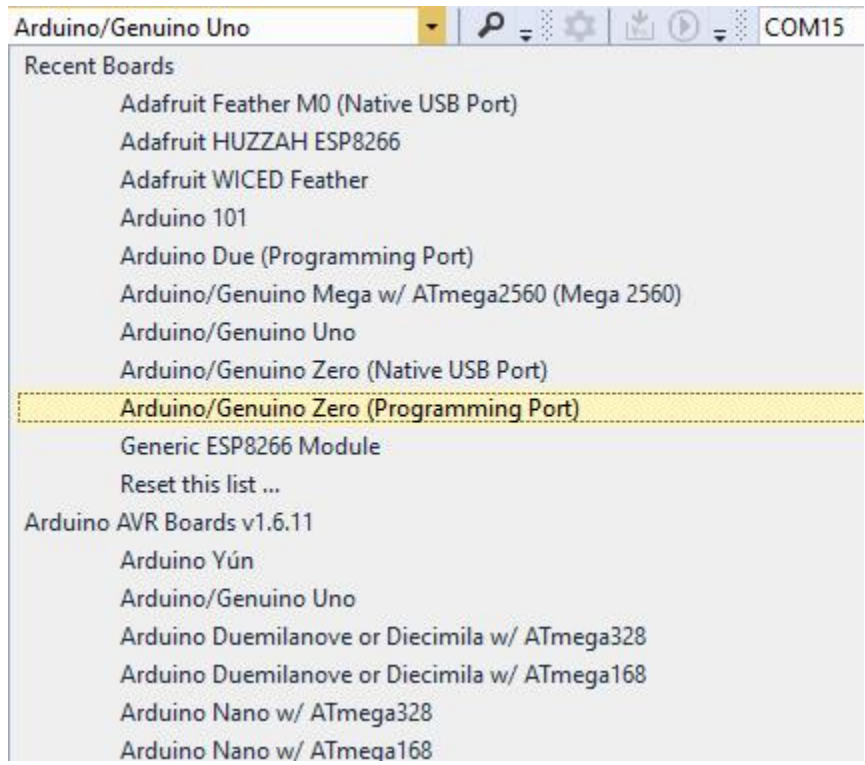
Arduino compatible hardware from a wide range of manufacturers is supported. Arduino sketch/project code does not need to change to work in Visual Studio, it remains fully compatible with the Arduino IDE (whichever version you are using). All Arduino examples for installed hardware and libraries can be used without modification.

## Easy To Use

New Arduino users are guided to work within the normal Arduino framework, in a similar manner to the Arduino IDE. Advanced Arduino users have a range of options that allow the simple (but restricting) Arduino rules to be broken. The following example demonstrates the IDE in it's simplest form.

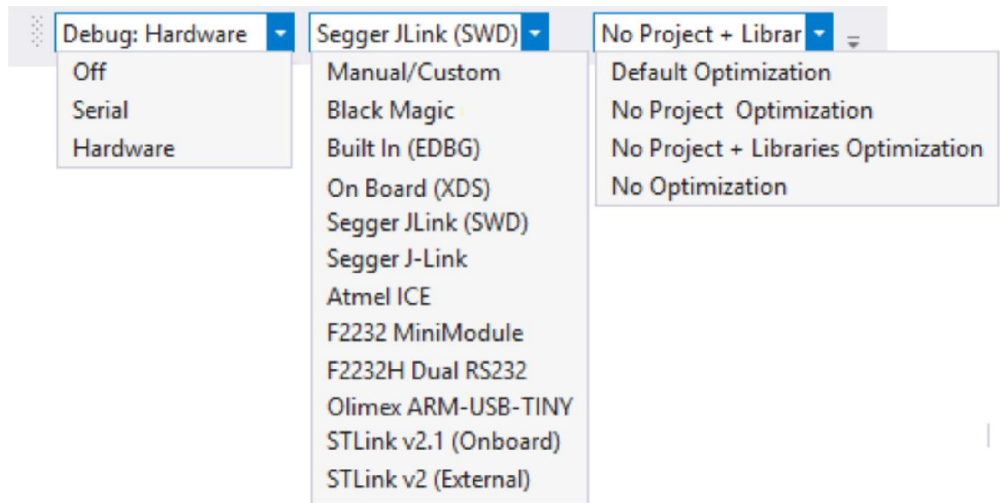
Selecting a Board, adding Libraries and Cpp or Ino Code To An Arduino Project





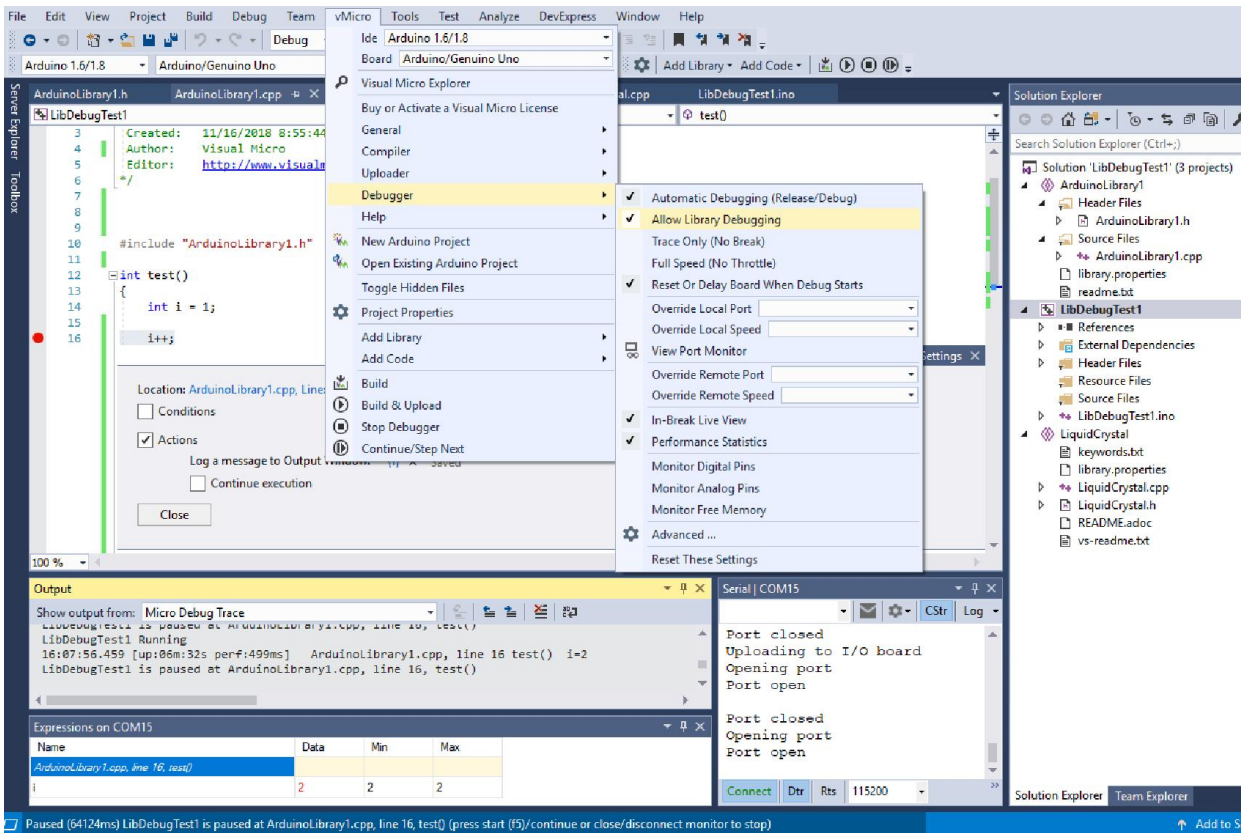
### GDB and Other Debugging

The Debug Options Bar or Menu Provides One-Click Setting Of Debug and Optimization

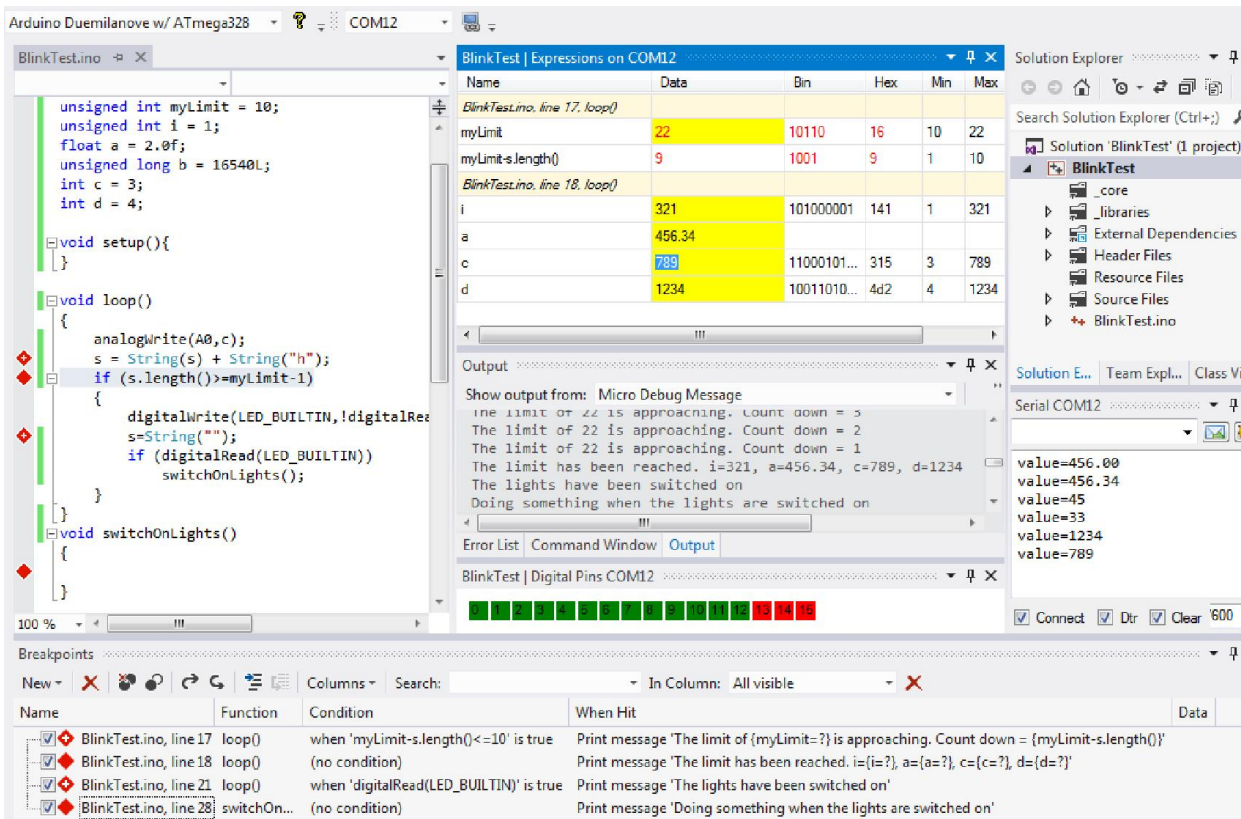


### Serial/Bluetooth/XBee/WiFi Debugging

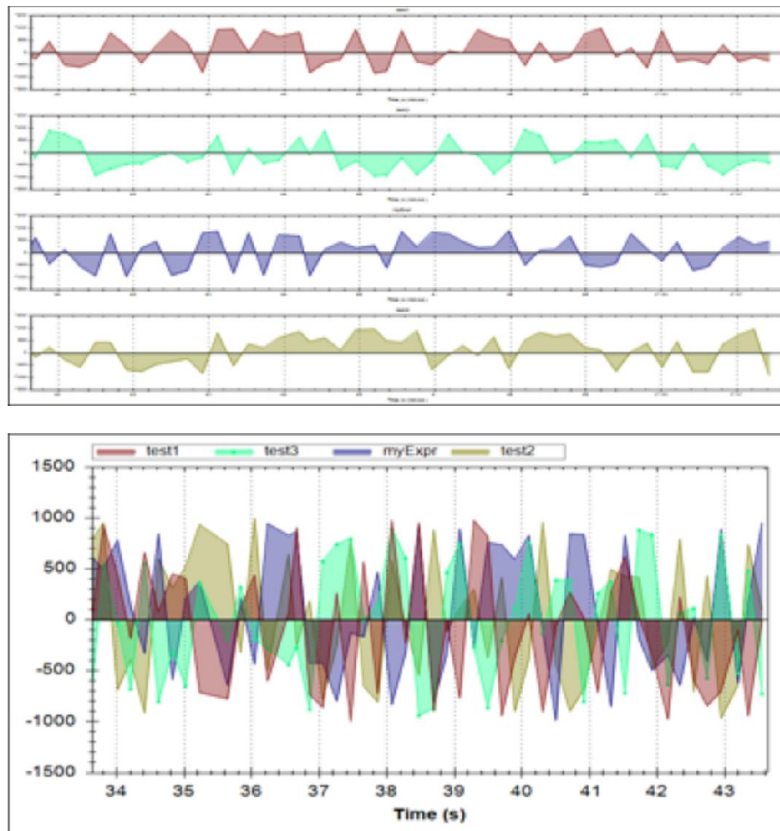
Library debugging (new Nov 2018)



Visual Micro includes a unique usb/wifi debugger that allows code to be more easily visualized.



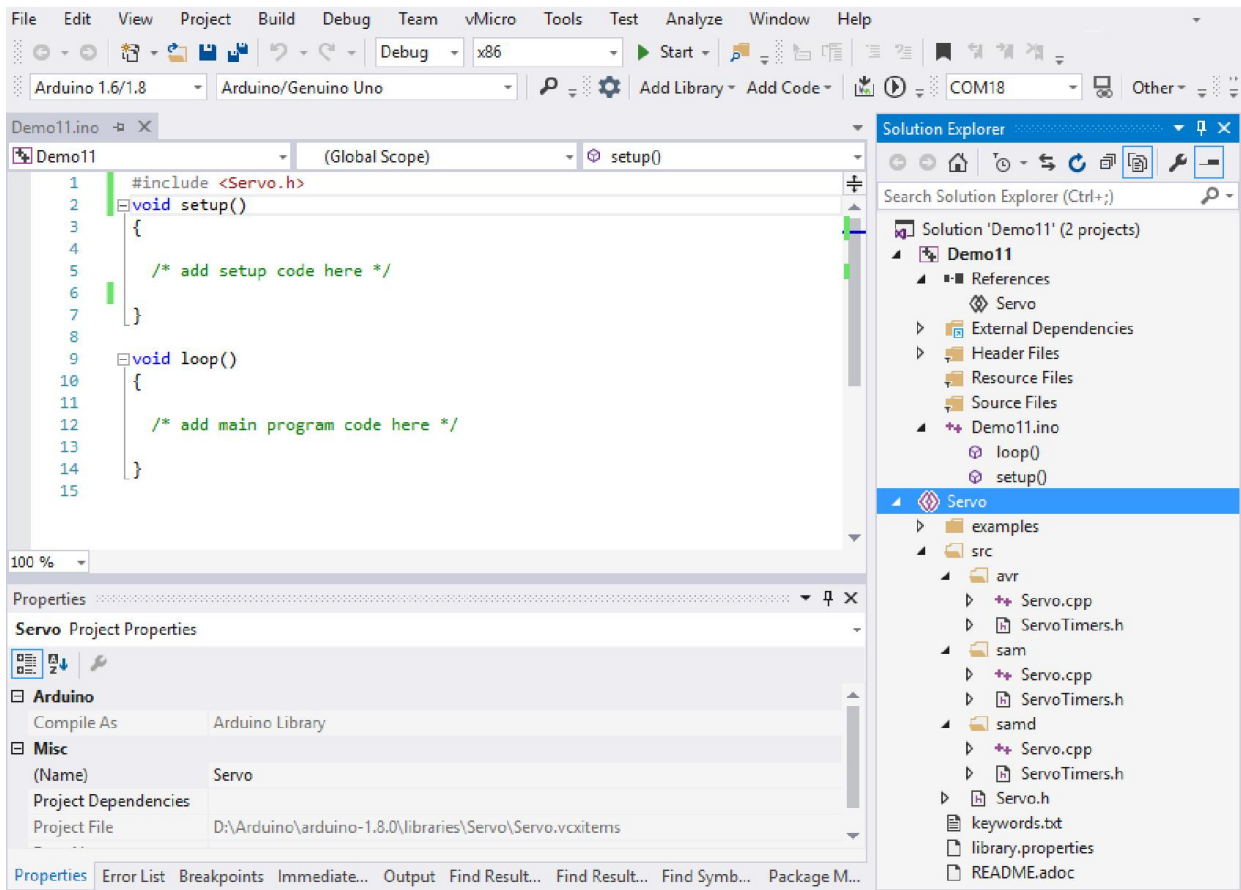
**Create Charts and Graphs Without Code Modification** [read more](https://www.visualmicro.com/page/User-Guide.aspx?doc=Multiplot.html)  
 (https://www.visualmicro.com/page/User-Guide.aspx?doc=Multiplot.html)



## Cross Platform Shared Code and Library Editing

Development teams can take advantage of [Visual Studio Shared Cross-Platform Projects](https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx) (<https://www.visualmicro.com/post/2017/01/16/Arduino-Cross-Platform-Library-Development.aspx>) which allow cross-platform code to be created in smaller projects that are automatically combined with the program code during build. Alternatively teams might elect to develop Arduino libraries which also benefit from Visual Studio shared projects (see image below)

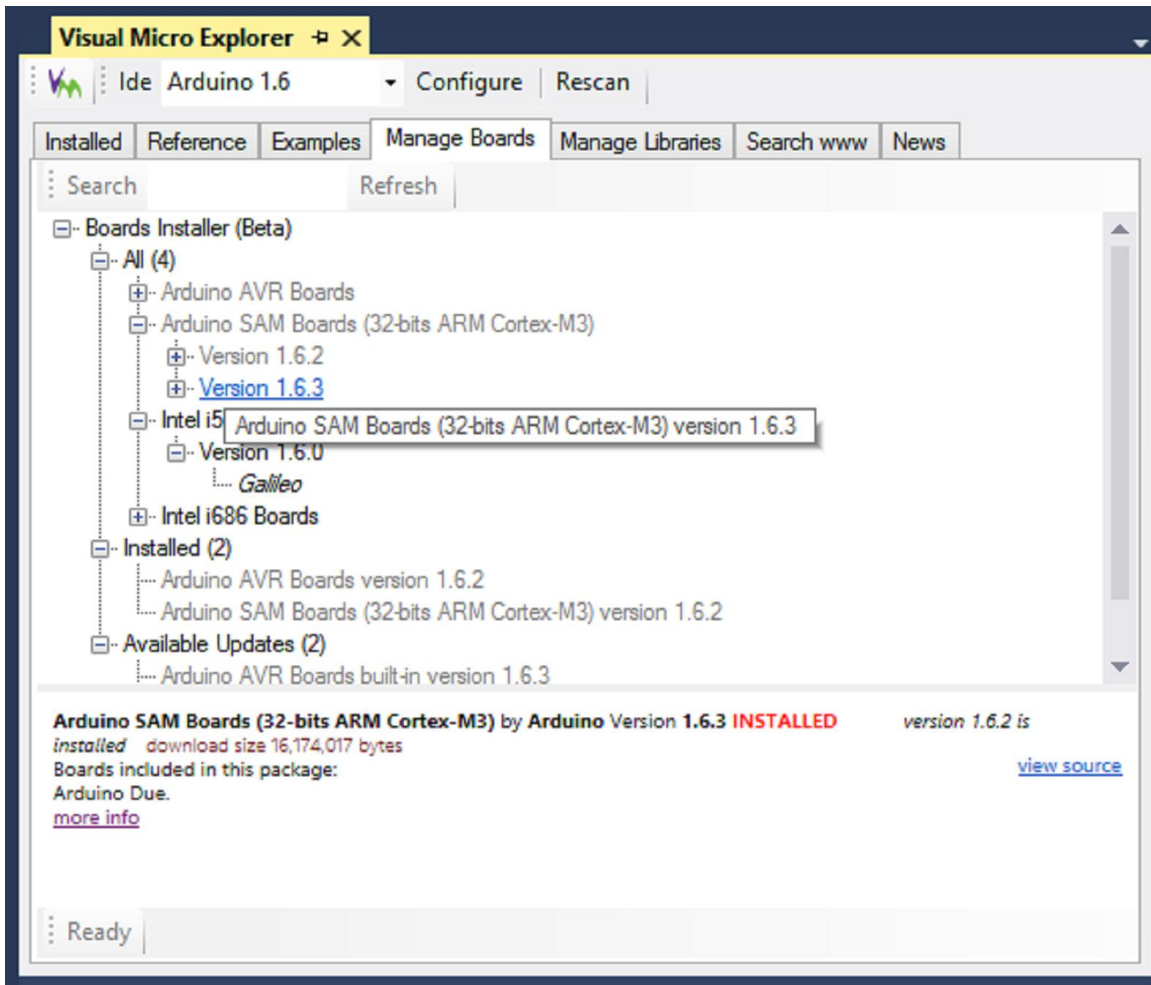
Shared projects and shared libraries can be located in any location which makes source control and version control very easy. Another benefit is that Visual Micro provides varying cross-platform intellisense and build/upload when editing shared code. This is useful because shared cross-platform code does not have a specific architecture, therefore would not normally have platform specific intellisense or build/upload capability.



## Arduino Compatible Board and Library Manager

*The plugin also supports all Arduino library formats, all third party Arduino compatible hardware and, for the more advanced user, all programmer combinations.*

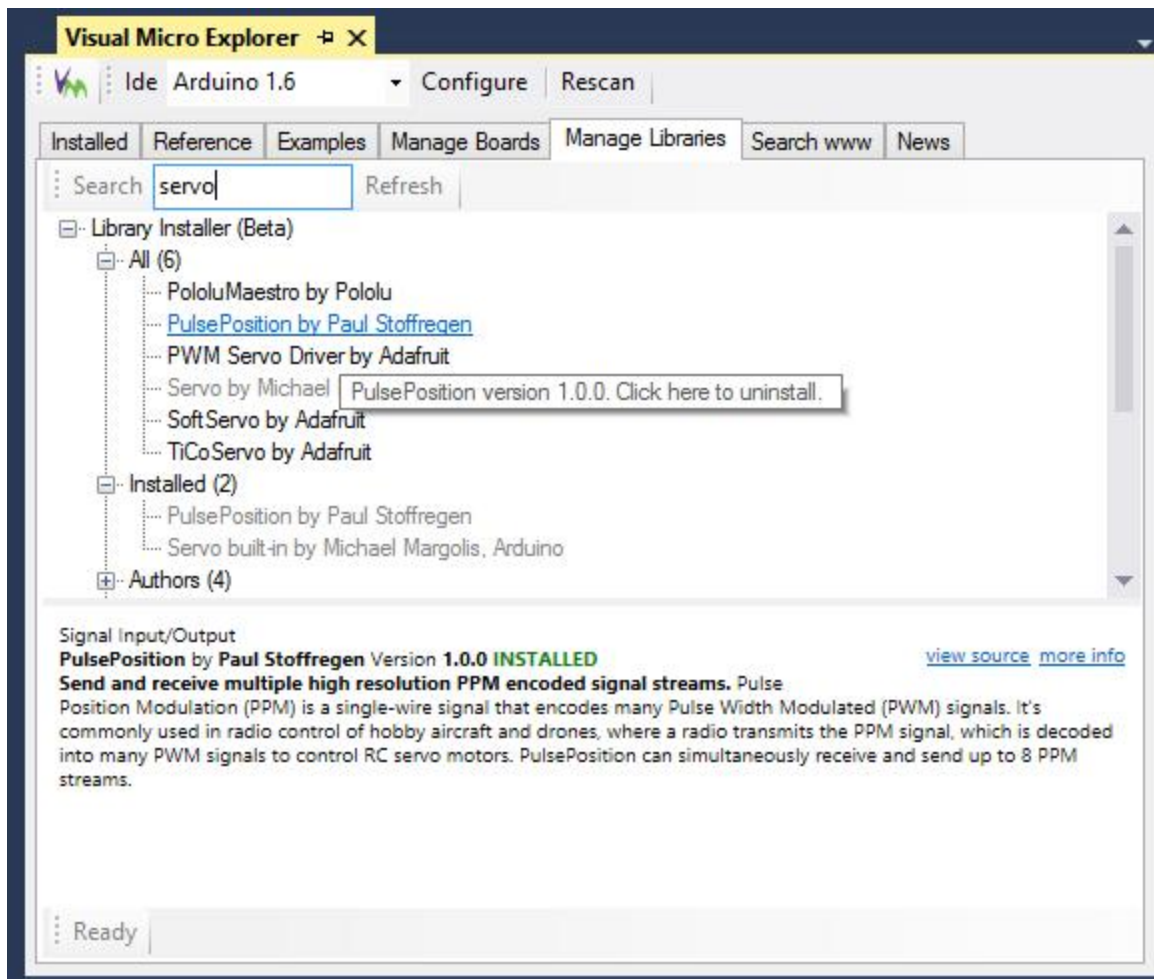
*April 2015 now includes an automatic Board Downloader and Manager*



(<https://www.visualmicro.com/page/User-Guide.aspx?doc=Board-Manager.html>)

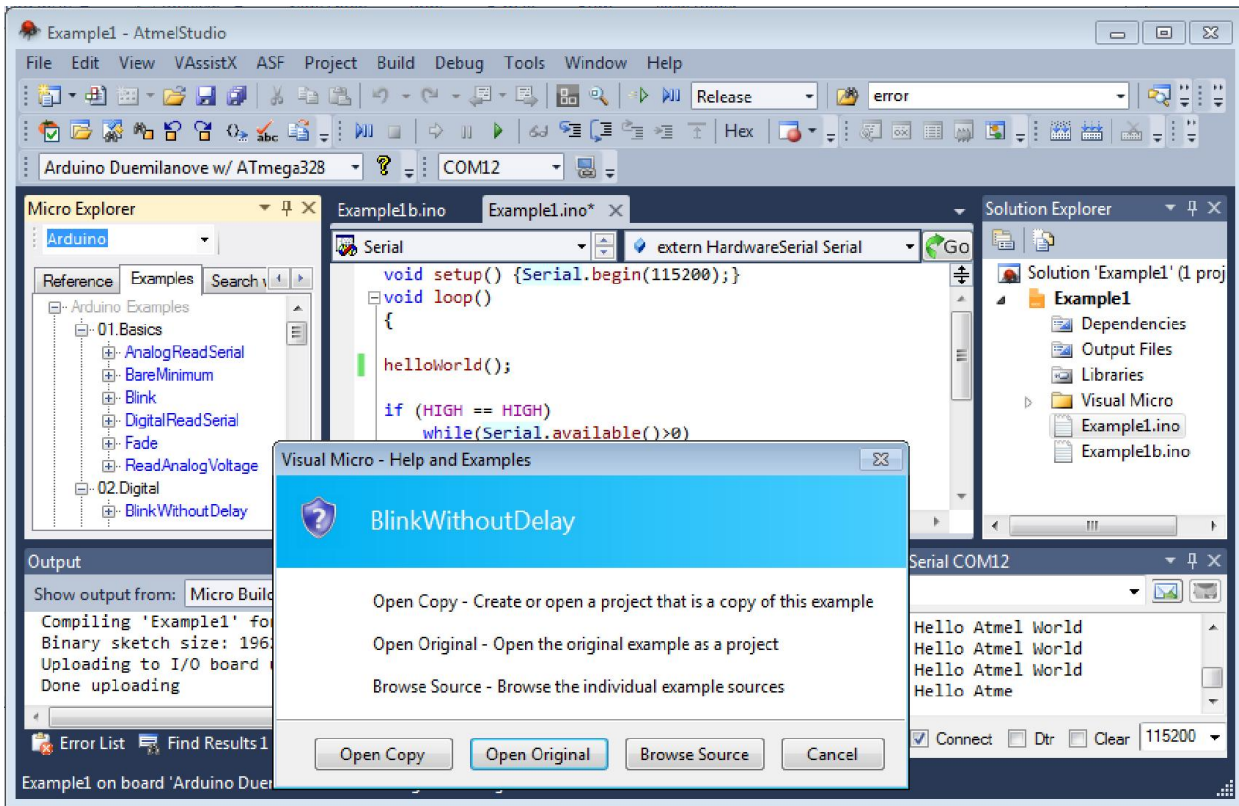
and a Library Downloader and Manager





### Arduino Reference Help and Examples

Search Arduino IDE references and browse examples. Automatically discovers examples for newly installed libraries and hardware. Create a new project from an example.



**Please see the [documentation \(https://www.visualmicro.com/page/User-Guide.aspx?doc=index\)](https://www.visualmicro.com/page/User-Guide.aspx?doc=index) for more information. For help, assistance and new release notifications please join the [forum \(https://www.visualmicro.com/forums\)](https://www.visualmicro.com/forums)**

**Intellisense Note:** Intellisense and build/upload are two separate functions of Visual Micro. You might see a few invalid intellisense errors (which you should report to the forum). After building a project with substantial code changes intellisense will become more accurate. However for some boards it is only 99% but can easily be extended.... Set the "Errors List" window to "Build Only" and/or "Open Documents Only" if false intellisense errors are annoying you, +report your code + board name to the forum thanks!

Tip: Slow compile times? are caused by some Virus Checkers. The two build folders that Visual Micro uses do not create files windows can run so you might consider them safe (your choice). The build folders can be seen when Verbose build is enabled (see vMicro menu), the folders are usually "c:\users\[name]\appdata\" (vmbuilds and vmbcore sub folders). These folders can be overridden in the options "tools>options>visual micro>compiler"

tip: Project library versioning: Copy/use shared library projects. Shared library projects are normal Arduino libraries that can exist in any location, giving control to each project over which library version will be used.

Visual Micro Limited - Registered in England No. 8093282



(<https://smarticon.geotrust.com>)

[/sm...onprofile?&erer=https://www.visualmicro.com](https://www.visualmicro.com)