Many times we need to work with custom arduino boards which are cool. For say we all know arduino Uno

But you know you can make core library for any microcontroller in such a way that it will support the arduino programming environment .

For say Serial.begin(9600) will be same but the underlying code codes will be microcontroller specific . In this way many smart embedded software developers have made different microcontrollers compatible with arduino . And nowadays trend is like this if you want to sell your chip you better make this core . See this link this has the unofficial microcontroller list which are compatible with arduino

<https://github.com/arduino/Arduino/wiki/Unofficial-list-of-3rd-party-boards-support-urls>

Now usually you use Arduino IDE . Arduino IDE has by default only the official boards listed but you can other boards via board manager but that does not load the unofficial list (mentioned above) which has this cool set of boards

[PlatformIO](http://platformio.org/) is a console tool with rich commands interface it can work with different IDEs as the “builder, uploader & debugger”.

So better Arduino IDE options are:

1. Using Atom IDE with PlatformIO as plugin
2. Using Atmel Studio with vMicro as plugin
3. Using Visual Studio with vMicro plugin
4. Using Eclipse & platform IO as plugin
5. Using Visual Studio with PlatformIO as plugin

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Using Atom IDE with platformIO as plugin\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Right now ATOM ide is the official IDE of platformIO and by default installs the unofficial list and gives you the opportunity to play with these 3rd party boards . Platform IO is cool and its under development . there idea is to make an IDE which will be common for all arduino compatible hardwares i.e. microcontrollers

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Using Atmel Studio with vMicro Plugin\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Now lets get professional . you have built an application now you want to debug it with JTAG like interface I mean in depth debug . both platformIO and Arduino Ide do not have this kind of feature . So what is a better IDE ? Answer is for AVR mcu it is AVR studio 7. You can connect Atmel ICE like powerful debugging hardware to your target AVR hardware and perform in depth debugging using atmel studio 7 .

So download AVR studio 7 .

For say We Our application is using the mightyCore board <https://maniacbug.wordpress.com/2011/11/27/arduino-on-atmega1284p-4/> and following the tutorial you can add the board to your aduino IDE by the following steps:

1. Download the [ZIP File](https://github.com/maniacbug/mighty-1284p/zipball/master)
2. Unzip it a folder called ‘hardware’ off your sketches directory, e.g. /Users/maniacbug/Source/Arduino/hardware/mighty-1284p

But we want to use it in atmel studio 7

Oh You can create Arduino sketches now in avr studio 7 but with the official boards only .

So what to do ? so how can I add this mightcore board support to atmel studio 7: stay with me follow the steps:

1. Download and install the latest Arduino IDE from official website
2. Go to tools>extension and updates
3. Install “Arduino IDE for Atmel Studio 7” :does not work now may be went to full paid
4. <https://github.com/MCUdude/MightyCore> download the zip
5. Restart you will see vMicro
6. Go to vMicro>board>find or install new Board
7. Click Install new hardware from zip and point the zip downloaded

Tada !!! now you have installed an unofficial board at your powerful Atmel Studio 7 IDE

1. Actually the downloaded new board folder goes to

“C:\Users\Hassin\Documents\Arduino\hardware”

And you will see that your vMicro at atmel studio 7 configuration manager has

“C:\Users\Hassin\Documents\Arduino” directory

1. vMicro is actually always working with arduino IDE that you have installed so new library or hardware adding will always be got at vMicro automatically
2. Now goto vMicro > new Arduino Project and start your creative works ;)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* using Visual Studio with vMicro plugin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

As vMicro is paid now and is not reachable from atmel studio 7 so step 3 wont work . upon googling I found <http://www.visualmicro.com/page/Arduino-Visual-Studio-Downloads.aspx> but this installer will work with visual studio .

Right now visual studio has got free so you can use it so download ans install it then install the visual micro extension

Give it the arduino ide installation location

Include existing files or add new class by right clicking the project not the solution

<https://www.hackster.io/shakram02/arduino-visual-studio-fast-dev-683414>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* eclipse with platformIO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A better option may be to switch to Eclipse:

Eclipse

1. Download eclipse ide for c/c++ developers (eclipse mars)
2. Intro <https://www.youtube.com/watch?v=6t7UbX812Yw> to platformio [this step is optional you may see it]

Now following the video

PlatformIO is written in python

[PlatformIO](http://platformio.org/) is a console tool with rich commands interface it can work with different IDEs as the “builder, uploader & debugger”.

* Install platformIO core

1. Install python 2 and add to user variable path
2. Download <https://raw.githubusercontent.com/platformio/platformio/master/scripts/get-platformio.py>
3. Go the download directory > open command window > Then run this script file

python get-platformio.py

1. Close the command prompt and restart
2. platformio gets installed and platformio.exe is at “python27/scripts” dir and as this directory is got at user variable path
3. so now manually add C:\Python27\Scripts to system variables path also .
4. so now call platformio and you will see that it is installed

* Now you can follow this tutorial as reference

<http://thomasweldon.com/tpw/courses/embeddsp/p00pcFrdmK64_eclipsePlatformioSetup.html>

i will be following this

* Go to the directory where you want to create the project for example “blinkOnUno”
* Then run

platformio init --ide eclipse –b uno

* Then you will see project has been successfully initialized so its ok now
* Go to the project directory and you will see the folders & other files have got automatically created
* Now you can follow this as reference <http://docs.platformio.org/en/latest/ide/eclipse.html>
* so now open eclipse and import the project
* file>import>general>existing project into workspace then click finish
* create main.cpp in src folder
* write the code
* right click on the project : properties>C/C++ build>environment>check the path variable whether it has “C:\Python27\Scripts” > apply it
* then go to the right panel of the IDE and select Make target and click PlatformIO:build
* then click PlatformIO:upload

1. You can link platform IO to eclipse <http://www.ikravets.com/computer-life/programming/2014/06/20/building-and-debugging-atmel-avr-arduino-based-project-using-eclipse-ideplatformio>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Using Visual Studio with PlatformIO as plugin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*