Arduino IDE (Integrated Development Environment): Using It to Load / Modify uBitx Software

by Gordon L. Gibby MD
version 1.0 August 6 2018

The Arduino Integrated Development Environment (IDE) is a free software tool that allows you to edit, compile, and upload software into the Arduino family of microprocessors (Nano, Uno, etc). The hfsignals.com short-kit uBitx and earlier Bitx40 utilize the Arduino Nano microcontroller (microprocessor). If you wish to be able to update or improve your software loaded onto the "Raduino" of these transceivers, you'll need to learn how to utilize the IDE.

Software on the Nano appears to be a subset of the "C" programming language. This is a very powerful programming language that can take a while to learn, so just take tiny steps at first. There are many commercial kits for adults and school children to learn simple programming on the Arduino and these may become a great help to you as you developing your software skills. There are also many books, videos and other learning opportunities.

Diving into getting sofware upgraded on a uBitx

1. Downloading the IDE: https://www.arduino.cc/en/Main/Software Current download version at the time of this writing is version 1.8.5 (My current installed version is 1.8.4). Versions are avaialable for:

Windows, multiple flavors
Mac OS X
Linux 32/64 bits
Linux (ARM)

The IDE itself is open-source --- you can download the source for the development environment itself —but please don't bother with this unless you are a super-super-geek way way beyond me!

2. Install that system. On my Windows 10 computer, it installed itself into the following subdirectory:

"C:\Program Files (x86)\Arduino\arduino.exe"

3. Decide where you are going to keep your Arduino projects. You are going to want to keep each one of your possibly SEVERAL little arduino projects in separate subdirectories so they don't get confused with each other. Further, each project may possibly have multiple files, in some cases ending in ".ino" and in other cases (more rarely) ending in ".cpp".

An oddity of the development environment is that the primary development file of each project (they call a project as a "sketch") must have a name that corresponds to the directory in which it is kept.

For example, you might choose to keep your Arduino sketches in a directory c:\Projects\Arduino, and name them as follows:

c:\Projects\Arduino\ubitx_20
which includes ubitx_20.ino and several other files from the original factory release software of the distributed uBitX kits (boards known as "version 3")

c:\Projects\Arduino\ubitx_v4.3_code
which includes file ubitx_v4.3_code.ino and several others which all together make up the current factory software release of the currently distributed uBitX kits (boards known as "version 4") --- and it is this version that Alachua County North Florida Amateur Radio Club will be starting with

and you might wish to have a separate directory for your own personally developed code if you get to that point.

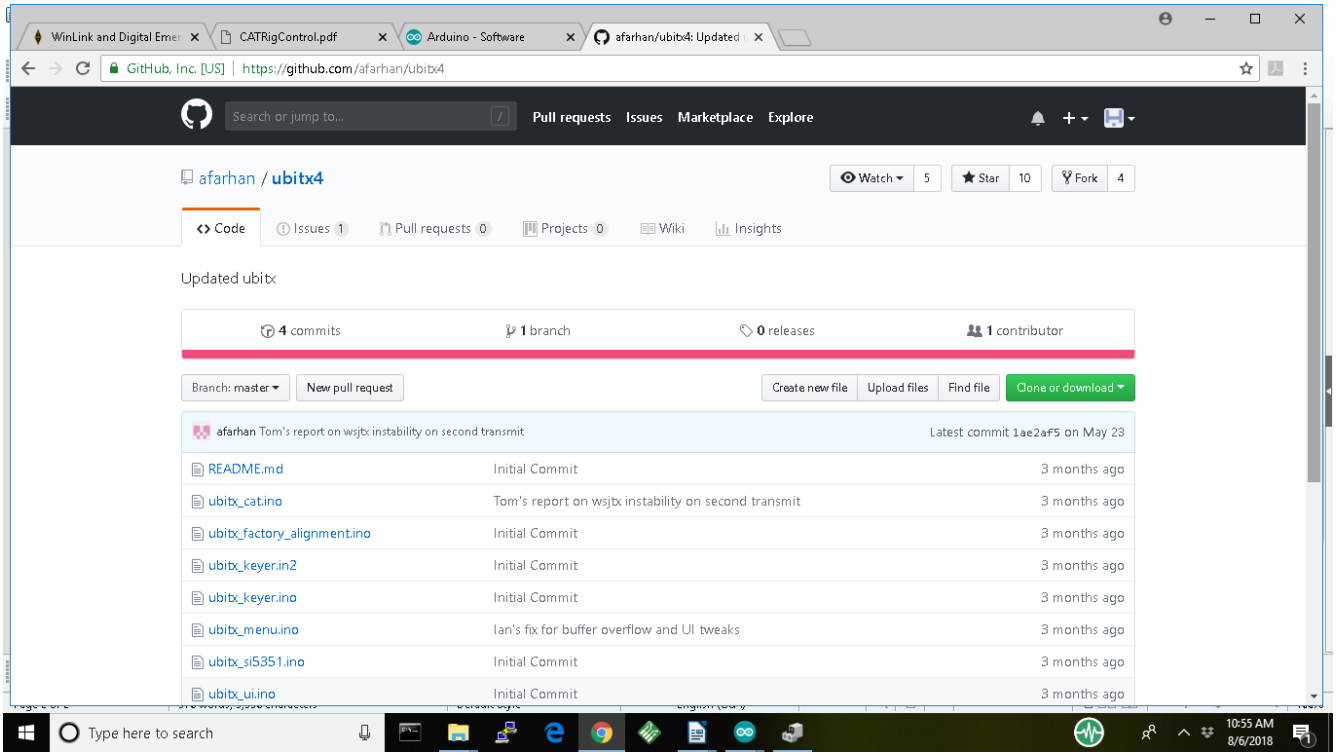
Even if you don't figure all that out, the IDE is pretty smart and if your files aren't in a properly named subdirectory, it will simply offer to create one for you and move them --- quite nice! The IDE also includes the ability to remember your "RECENT" projects, so you can easily go back to them even if you don't remember exactly how you placed them.

4. Download the current shipping version of uBitx code.

Using your internet browser, go to

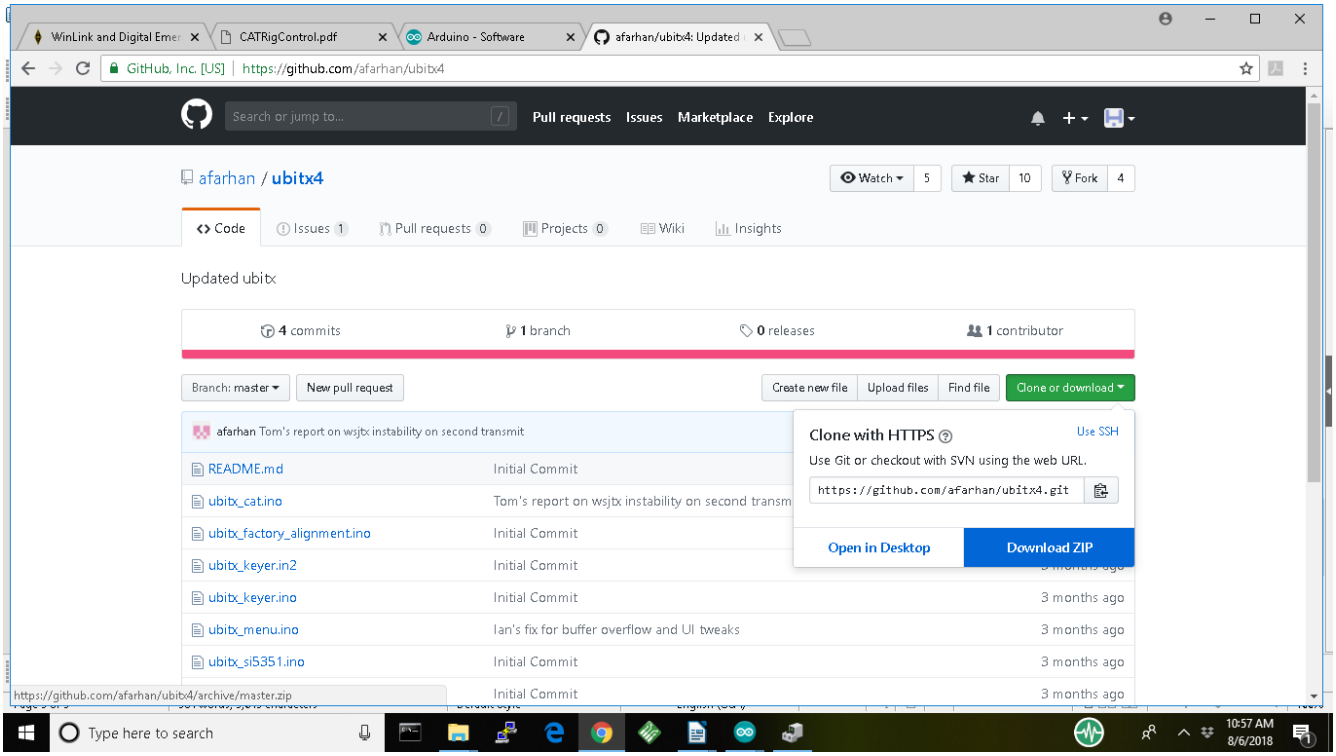
https://github.com/afarhan/ubitx4

(if the github site requires you to create an account --- I don't think it does for simple downloads – it is free and easy)



Click on that GREEN BUTTON “Clone or Download”

Then click on the DOWNLOAD ZIP option:

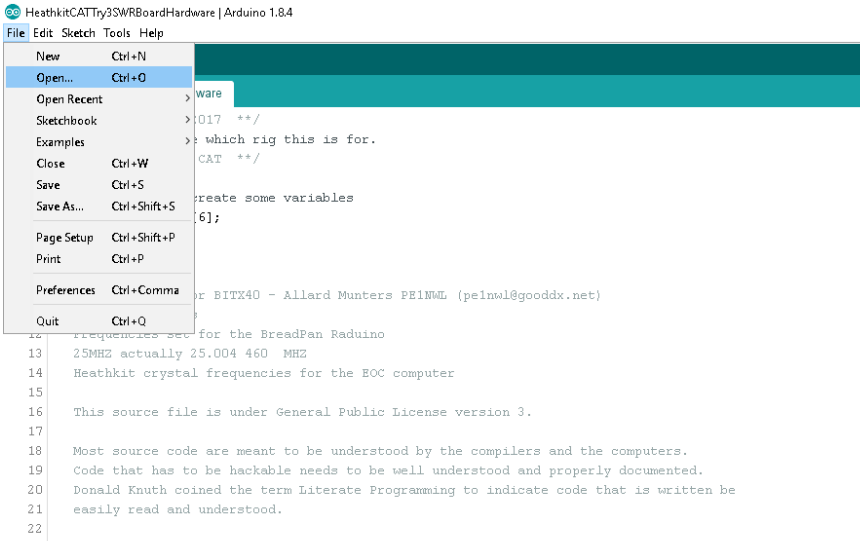


Once you have downloaded the zipped (combined version of several smaller files, also compressed) file, find it in the DOWNLOADS directory of your computer and EXTRACT it to the directory you chose above, possibly:

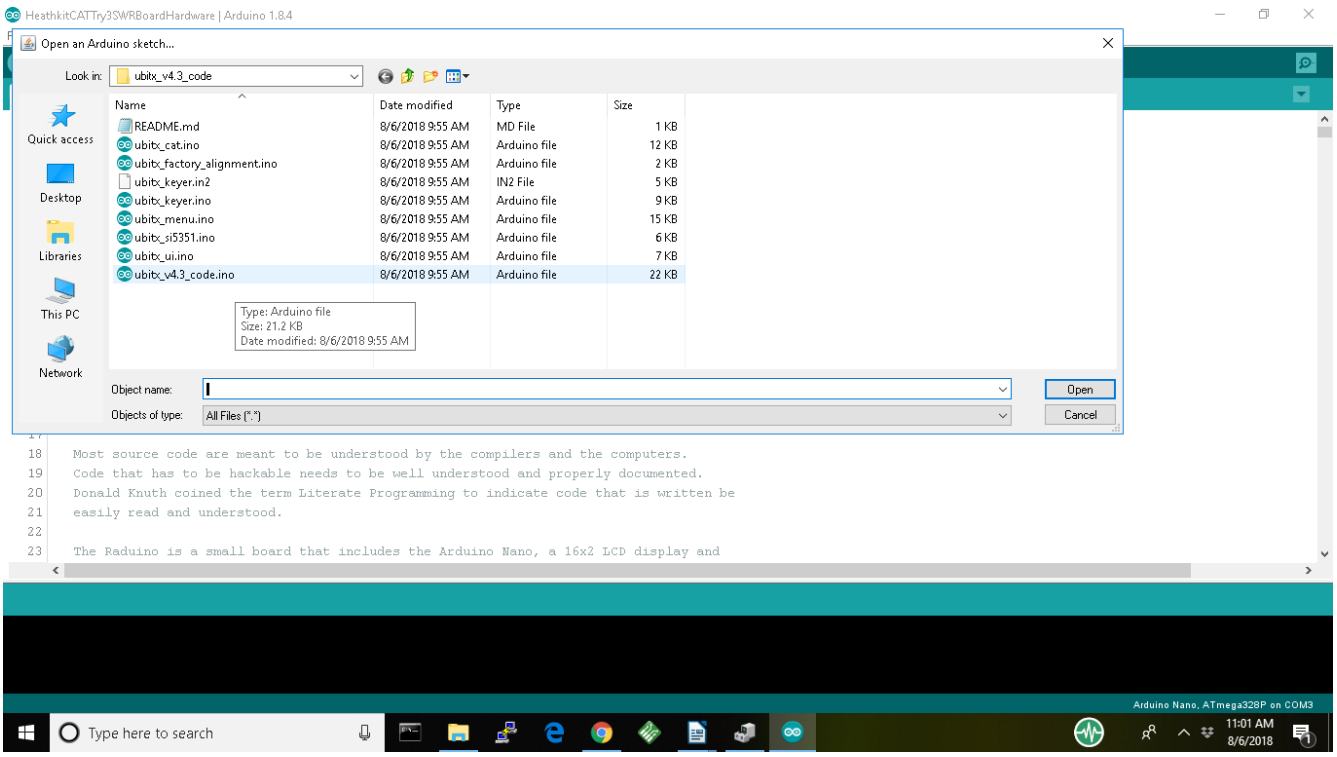
c:\Projects\Arduino\ubitx_v4.3_code

If it complains and wants it in a different directory, let it go wherever it wants to.

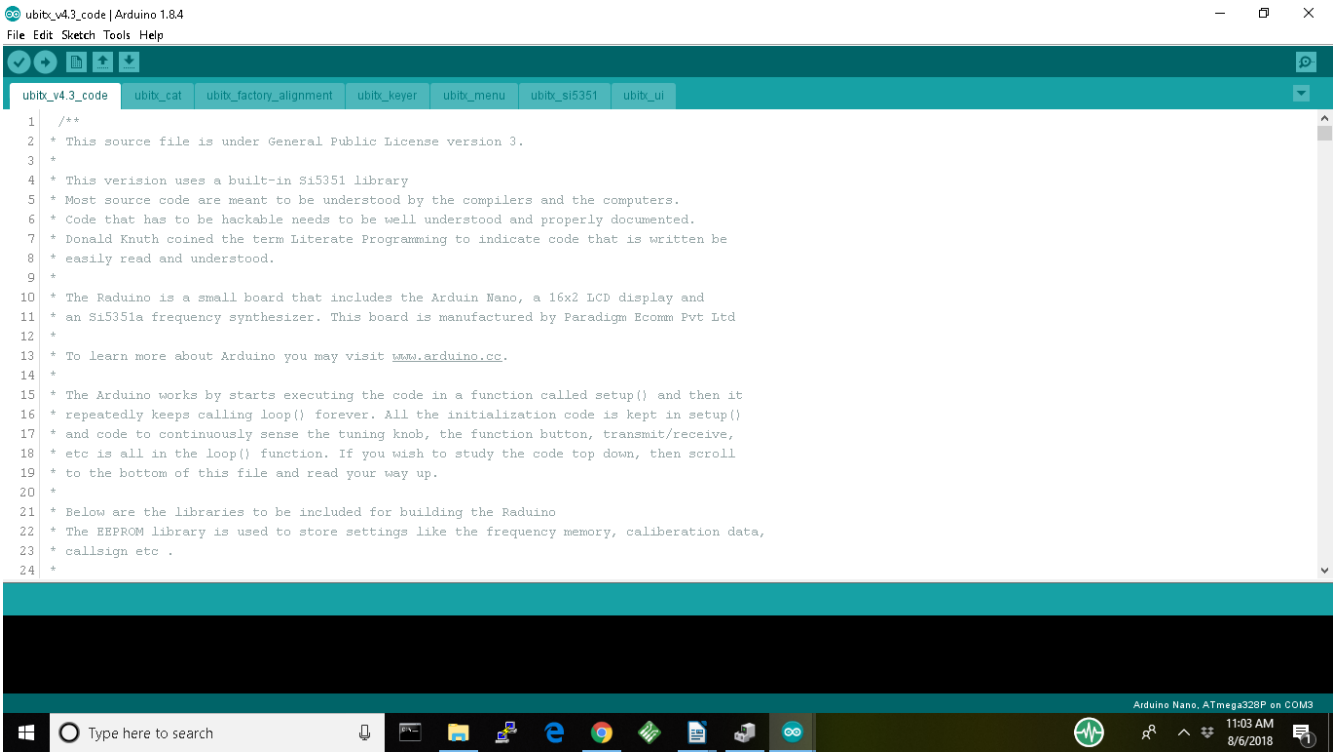
5. Open the IDE from the icon on your desktop or from your start menu



Your version will probably not have a project automatically opened in the background like mine, but using the FILE | OPEN menu choices, navigate to the software that you just downloaded and choose the file ubitx_v4.3_code.ino as below:



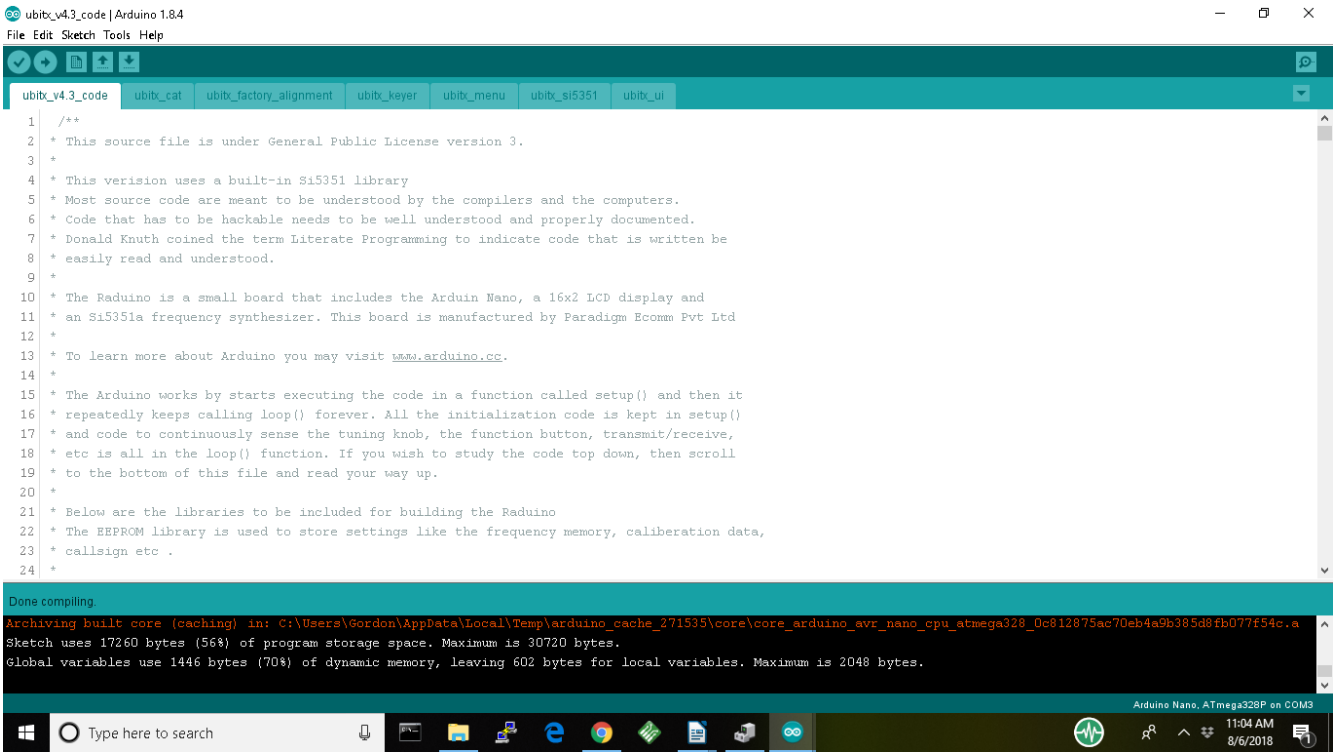
Once you do that, the following will display:



6. Give a try at compiling that factory code:

Sketch | Verify/Compile from the top menu:

A little green bar at the lower right will show your progress and if all goes well, you'll end up with a successfully compiled (turned into computer understandable 1's and 0's) software:



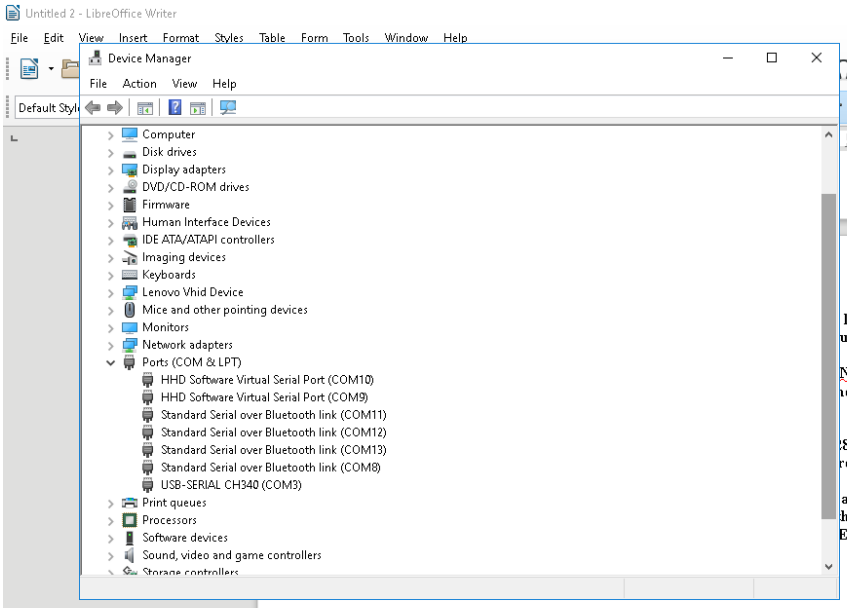
IF INSTEAD YOU GET ORANGE ERRORS in the bottom black horizontal box….it is time to find a friend who knows something about C-code and compiling and can help you out…. If you have no such friend, it is time to join the bitx20 group on groups.io and seek help!

7. Get your IDE setup for your card. You'll need a USB to mini USB cable to connect to the mini-USB port on the side of your Raduino

Make sure you have the right cable – don't try to force a rounded "micro-USB" connector into the funny little angled mini-USB jack on the Raduino.

Next you'll need to set the PORT, BOARD and PROCESSOR choices in the lower part of the TOOLS menu on your IDE --- two of these are easy, and the third isn't difficult:

> BOARD: for the Raduino (uBitx) the correct choice will be ARDUINO NANO. If you are programming some Elegoo kit or something else, you may need a different choice. It seems to often figure it out.
>
> PROCESSOR: for the Raduino (uBitx) the correct choice will be ATMega328P. Unless you are using some very inexpensive clone of something this will probably be the correct solution.
>
> PORT: This is more problematic because your WINDOWS computer will have assigned a PORT number to your cable and it may or may not be the last one listed by the IDE environment. My way of detective work to figure this out is to start the DEVICE MANAGER and then look into the PORTS section to find the one that matches the "USB Serial CH340" or some other give-away that it is the one on your Arduino….



In this case you can see that it is on COM3, so make that choice in your IDE.

8. UPLOAD new software: Once you have the software compiled successfully, and you have your board identified and properly connected, you can activate the upload by SKETCH| UPLOAD

It will usually do a complete re-compile and then UPLOAD to your Raduino --- when the upload is done the Raduino will reboot and you'll get a nice screen indicating you have

```
uBITX v4.3
```

CONGRATULATIONS! You are now capable of capturing all kinds of Raduino software (written correctly by others) and downloading them to upgrade your uBitX!!

Steps forward from here may included

– loading a special sketch that put your Raduino exactly on 25.000000000 MHz so that you can measure the “error” of your exact crystal and then later fix it in the version 4.3 code so your Raduino becomes “perfect” in its frequency measurements (if needed)

– adjusting various Beat Frequency Oscillators etc to make your received and transmitted signals sound the very best possible

and all kinds of other tweaks and improvemetns you can make. But for right now, you have a successfully installed piece of firmware and you can go about making your Radio work on the ham bands!!