

# HPSDR Firmware Programming

by Dave KV0S

openhpsdr.org



Loading HPSDR firmware in to the radio board FPGA seems to confuse many users. The tools that we have developed for the HPSDR project are design to make the process easier. To use the software you have to understand the interaction of the various part for the process to work correctly. This document will describe the options in order from the easiest to the most difficult.

## HPSDR Programmers

This groups of programs talk to a suitable firmware already load to upgrade the current firmware. The process in accomplished using the UDP network protocol.



*Figure 1. A diagram illustrating the connections used to load firmware into the Radio board.*

In the programmer you specify which network you are using, which is required as many computer have multiple network interfaces.

1. Then you initiate a board discovery on that network interface. This will return a list of radio boards that the programmer recognizes.
2. You choose the radio board you wish to program.
3. Next you select the firmware file that you wish to load into the radio board. These files end with \*.rbf.
4. First the flash memory is erased.
5. Next the file is broken into 256 byte packets which are sent to the programming part of the previous firmware still running in the FPGA. These packets are stored in the flash memory.
6. The firmware in the flash memory is read into the FPGA on the power up of the board.

7. You should have the new firmware running. If any mistakes happens in this process the firmware will not be in the FPGA or the flash memory.

Those of us that work of the software have very few mistakes and we know what to do when a mistake happens. Most mistakes I believe come from doing things out of order or being unsure what the feed back means. One of the purposes of this document is to help you be more familiar with the normal process and what should happen.

There are several programs that implement this type of programing protocol. They include:

1. HPSDRProgrammer\_web (New Protocol)– This a program that is started on the command line and produces a local web server that you point you web browser to and follow the same process to complete the firmware programming process for the new protocol.
2. HPSDRProgrammer\_cmd (New Protocol)– This is a command line program that implements this process with the new protocol.
3. HPSDRProgrammer\_V2-nopcap (Original Protocol) – This a GUI program that performs This process for the original protocol.

## Protocol Conversion Note

the HPSDRProgrammers to convert from Original Protocol to New Protocol But the HPSDRProgrammers are specific to their Protocol. All programmers can properly write any of the rbf files. But only original Protocol programmers will detect original Protocol firmware. And only New Protocol HPSDRProgrammers can detect New Protocol firmware. Many of the development group have done this several time but it is easy to get confused.

Because of the tremendous advantages of the New Protocol that fixes several short coming in the Original Protocol it was deem not necessary to handle both protocols in one program.

The alternative is to use the HPSDRBootloader that does not depend on either protocol to program the radio but does need a jump set to complete the task.

## HPSDRBootloader

This is a backup program that works with all protocols. It works in a different way to the HPSDRProgrammer. The HPSDRBootloader talks to a bootloader firmware is loaded in the first 1 Mb of flash memory. This firmware is loaded during the manufacture in all HPSDR derived radio boards. HPSDRBootloader requires a pcap library either libpcap on linus or Winpcap on Windows.

For the HPSDRBootloader to talk to the the Bootloader firmware you need to put a jumper on the J1 (Metis), J12 (Hermes) or J17 (Angelia) pins.

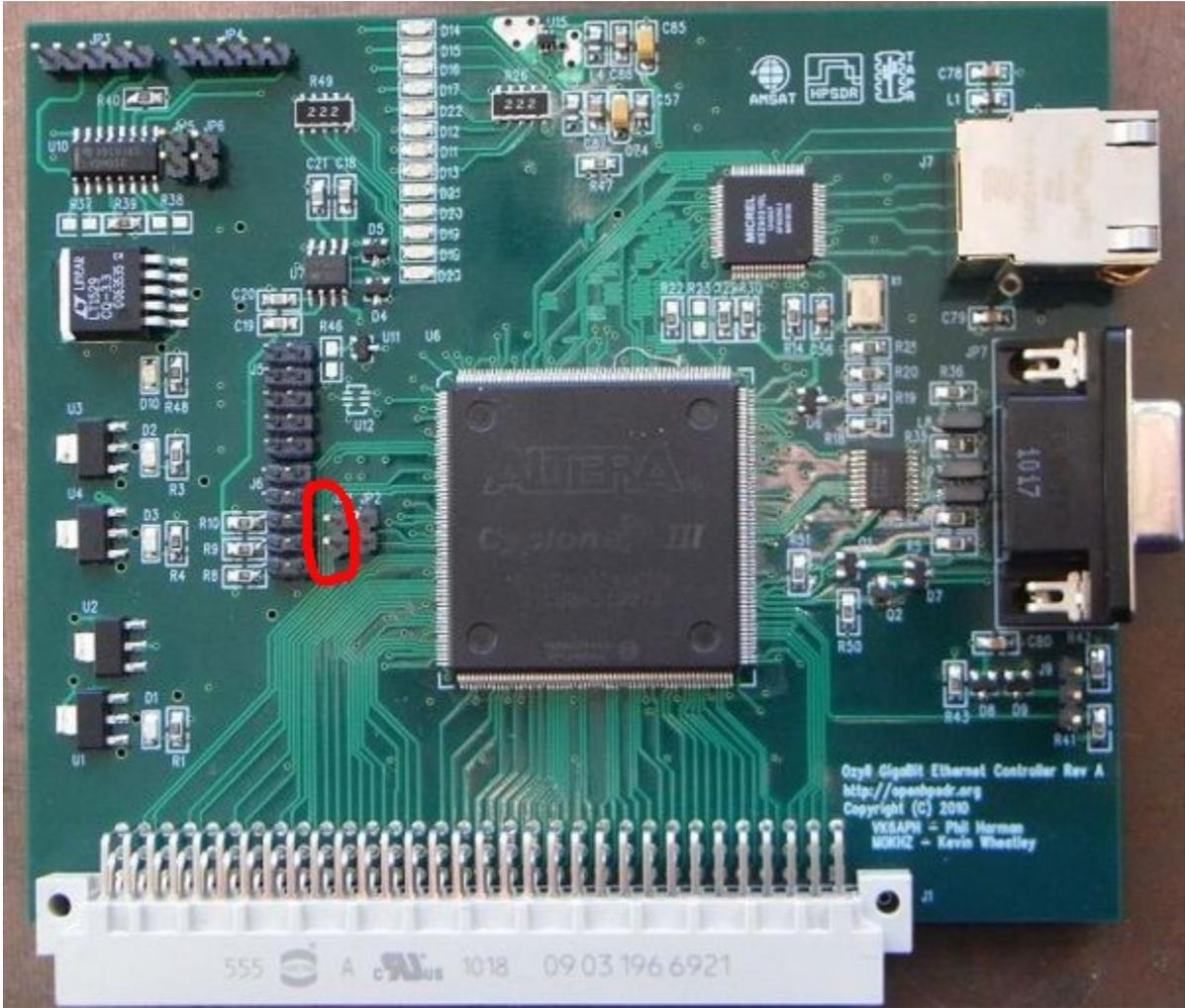
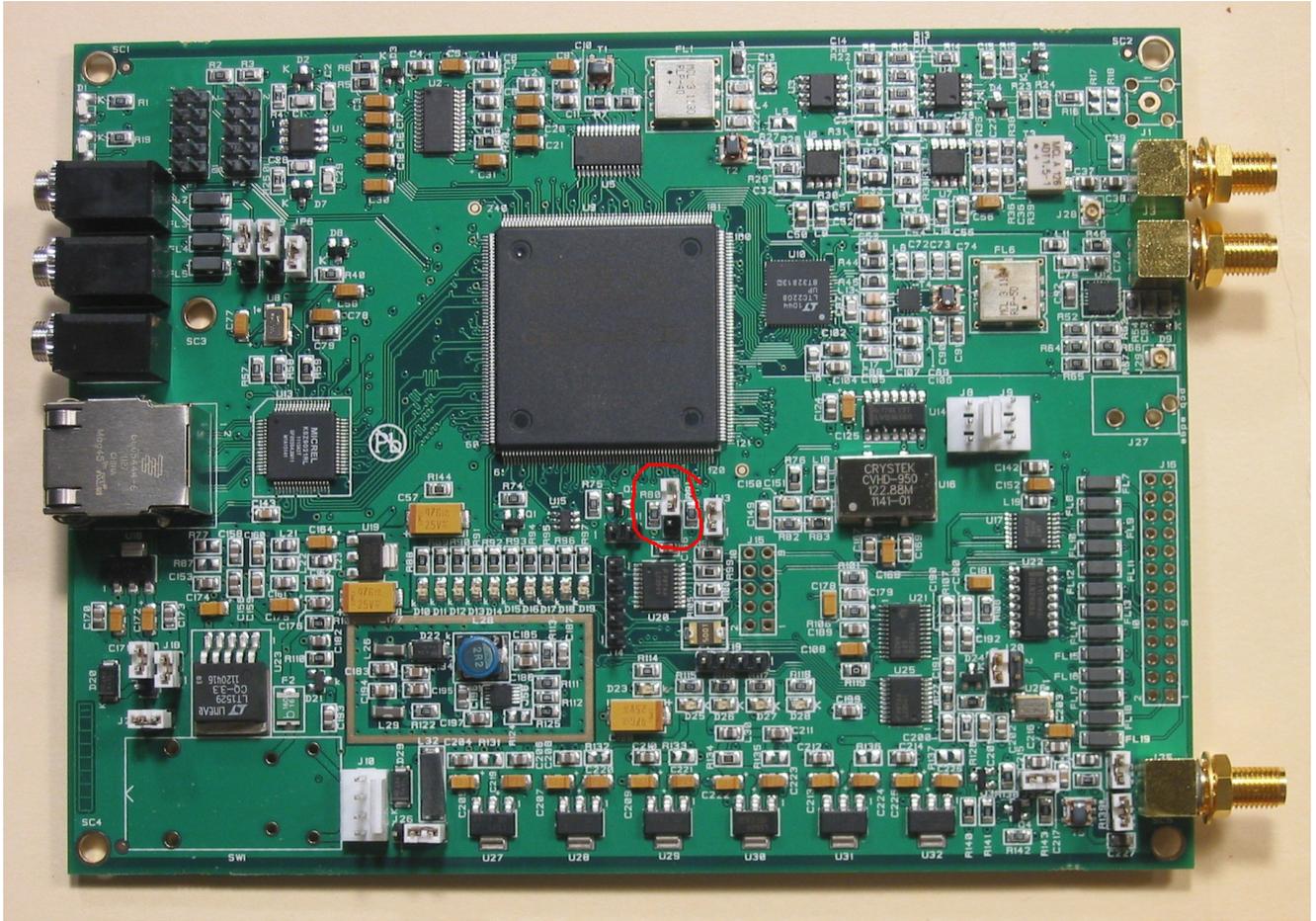


Figure 2. Location of Metis J1 in red circle



Figures 3. Location of Hermes J12 in red circle.



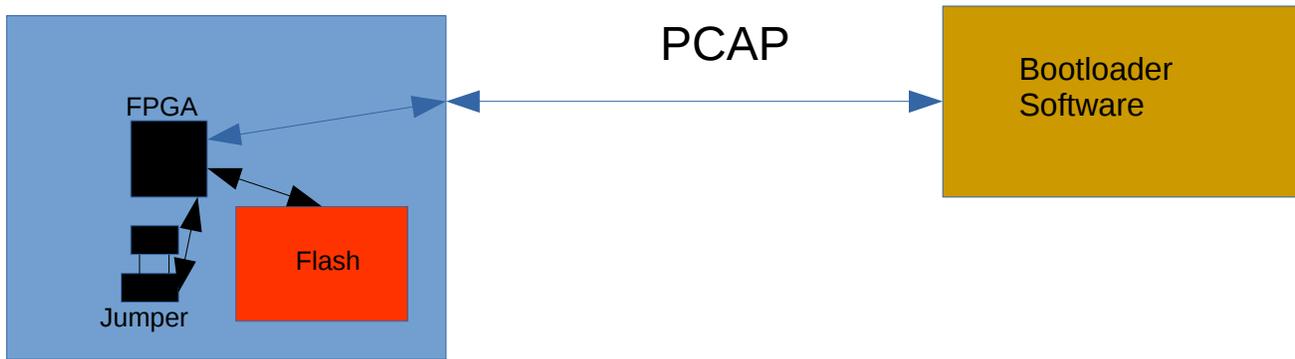
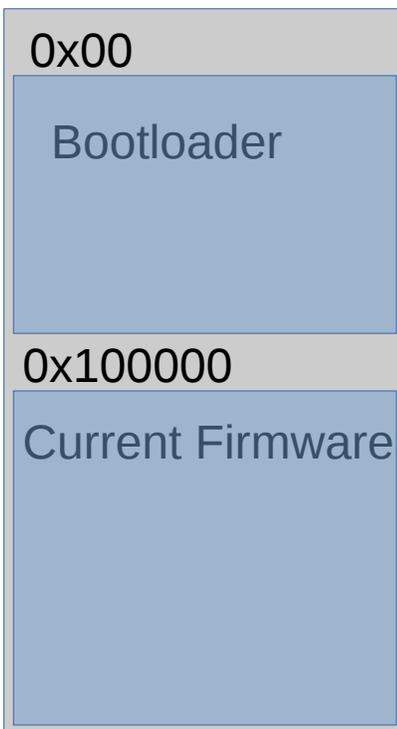


Figure 4. HPSDRBootloader configuration diagram.

Both the HPSDRBootloader and HPSDRProgrammer place the current firmware in the flash memory starting at the 0x100000 address. Neither the HPSDRBootloader or the HPSDRProgrammer will write firmware at 0x00. The following diagram of the flash memory on the Radio board describes the location of the two types of firmware. No change have been made to the bootloader firmware in the New Protocol.



Understand this arrangement if you resort to the last and hardest way to program the Radio Board.

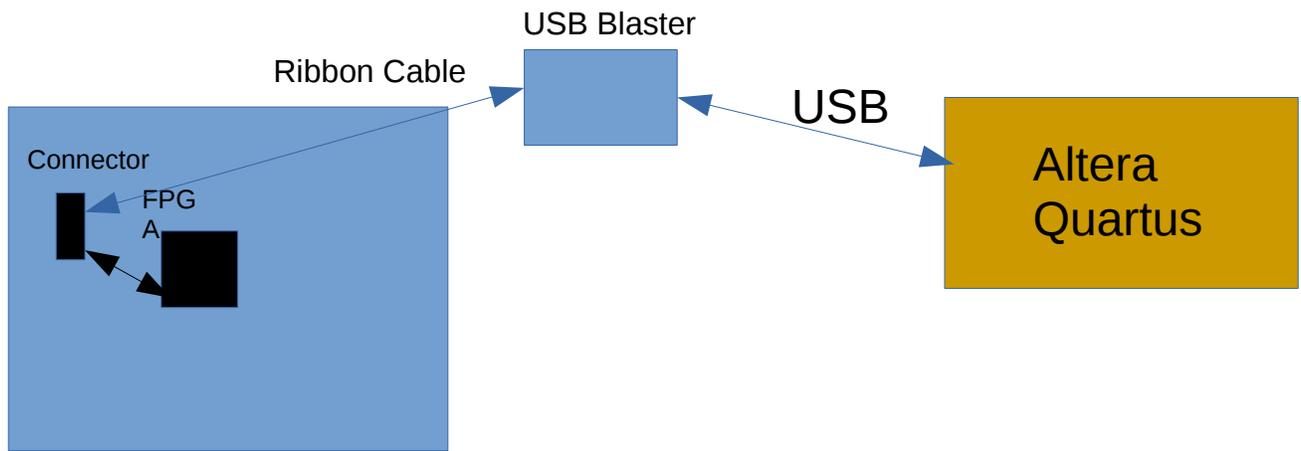
### Using Quartus and a USB Blaster

The last method of Programming the FPGA is to use a USB Blaster with the Altera Quartus Programmer software. In this process you will have to purchase or borrow a USB Blaster. You will also have to download a version compatible copy of the Altera Quartus software the free web version is OK.

The easiest way to do this is to use the USB blaster to load a new bootloader firmware into the the flash memory at 0x00 and then use the HPSDRBootloader to finish the programming process. I have had to do this a few time to replace bootloader that owners had over written the bootloader by using Altera Quartus and the USB Blaster to load the current Firmware.

A diagram of the configuration of the software and cable can be seen in Figure 4.

Figure 5 Memory Map.



*Figure 6. The Altera Quartus, USB Blaster cable configuration.*