

\$3 Bluetooth HID Module (HC05 with RN42 firmware)

Download

18 Steps

Share Favorite I Made it! Collection



Bluetooth HID (human interface device) modules will let you create bluetooth peripherals that can pair with mobile devices out-of-the-box (like bluetooth keyboards, gamepads, mice, etc.).

If we're looking to build our own bluetooth HID device, the hefty price of a single HID module can spike the budget of a project (like the BlueSMiRF HID for \$40 USD, or the BlueFruit EZ-KEY HID for \$20 USD).

An inexpensive bluetooth serial module commonly used by hobbyists, the HC-05, sells for \$3 USD, but is not HID capable by default. With a bit of a firmware hack though, we can turn this cheaper module into a bluetooth HID module.

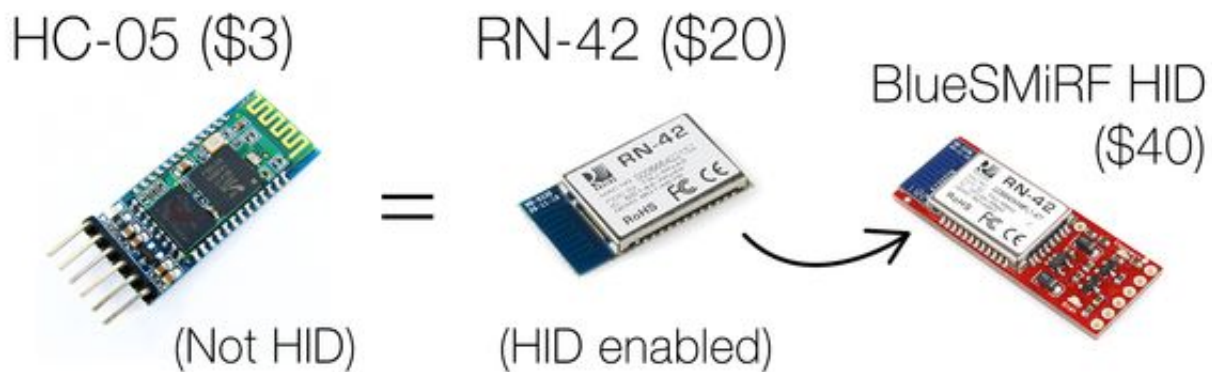
It may look like there's a lot of steps below, but it's actually just a very thorough guide. The flashing process of the HC-05 itself takes no longer than a minute after we have extracted the firmware from the RN-42 once.

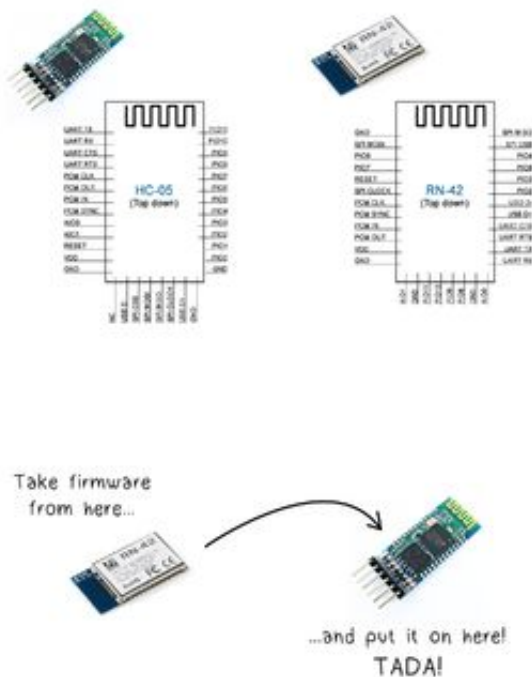
Step 1: Watch the video!



Check out this quick video, for a run-through all the steps below from start to finish.

Step 2: A comparison of the two modules





This cheaper module is the HC-05 and sells for around \$3 on ebay.

By itself, this module isn't HID capable. But luckily, this module actually uses the same chip (the BC417) as the RN-42 bluetooth module, which also powers the \$40 BlueSMiRF HID modules.

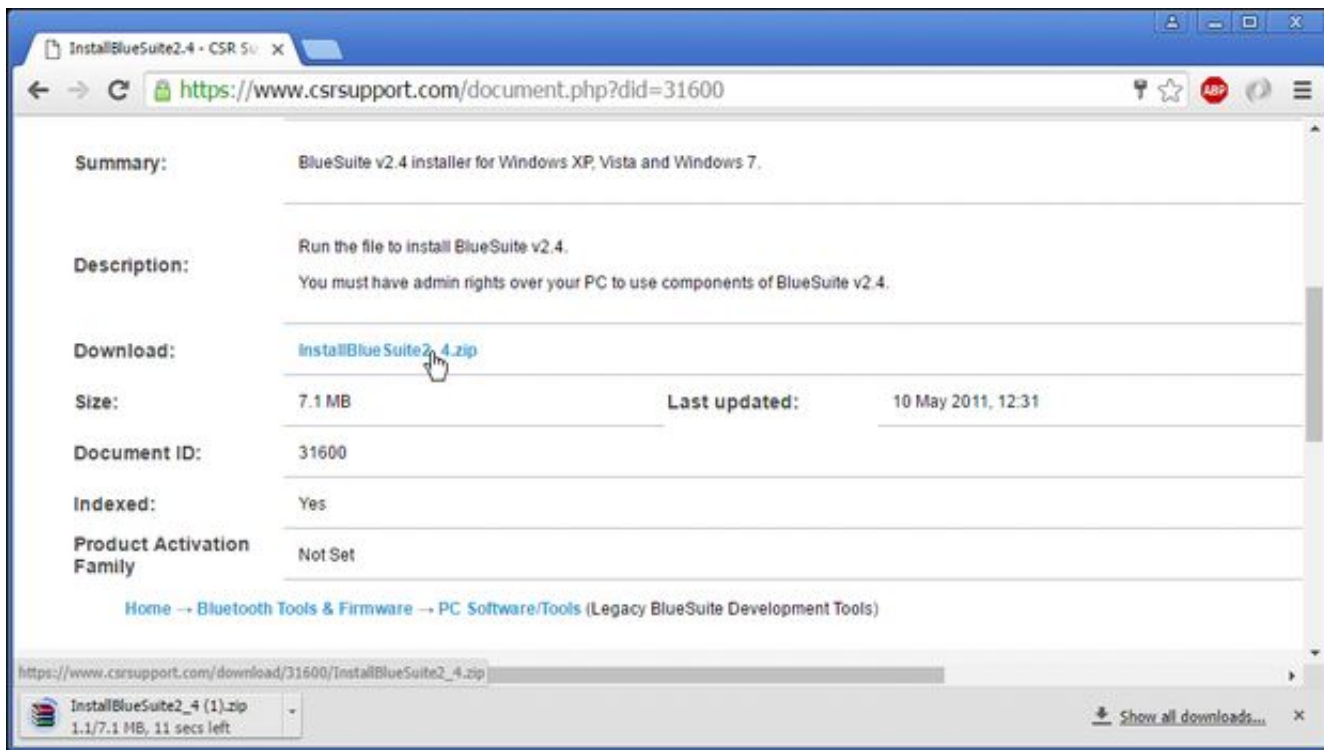
The only differences between these modules are the firmware and the location of the pin outs.

So here's the plan - to make our \$3 module HID enabled, we'll just have to copy over the firmware from the RN-42 chip to the HC-05.

It's as simple as that!

Where to buy

Step 3: Get the BlueSuite tools



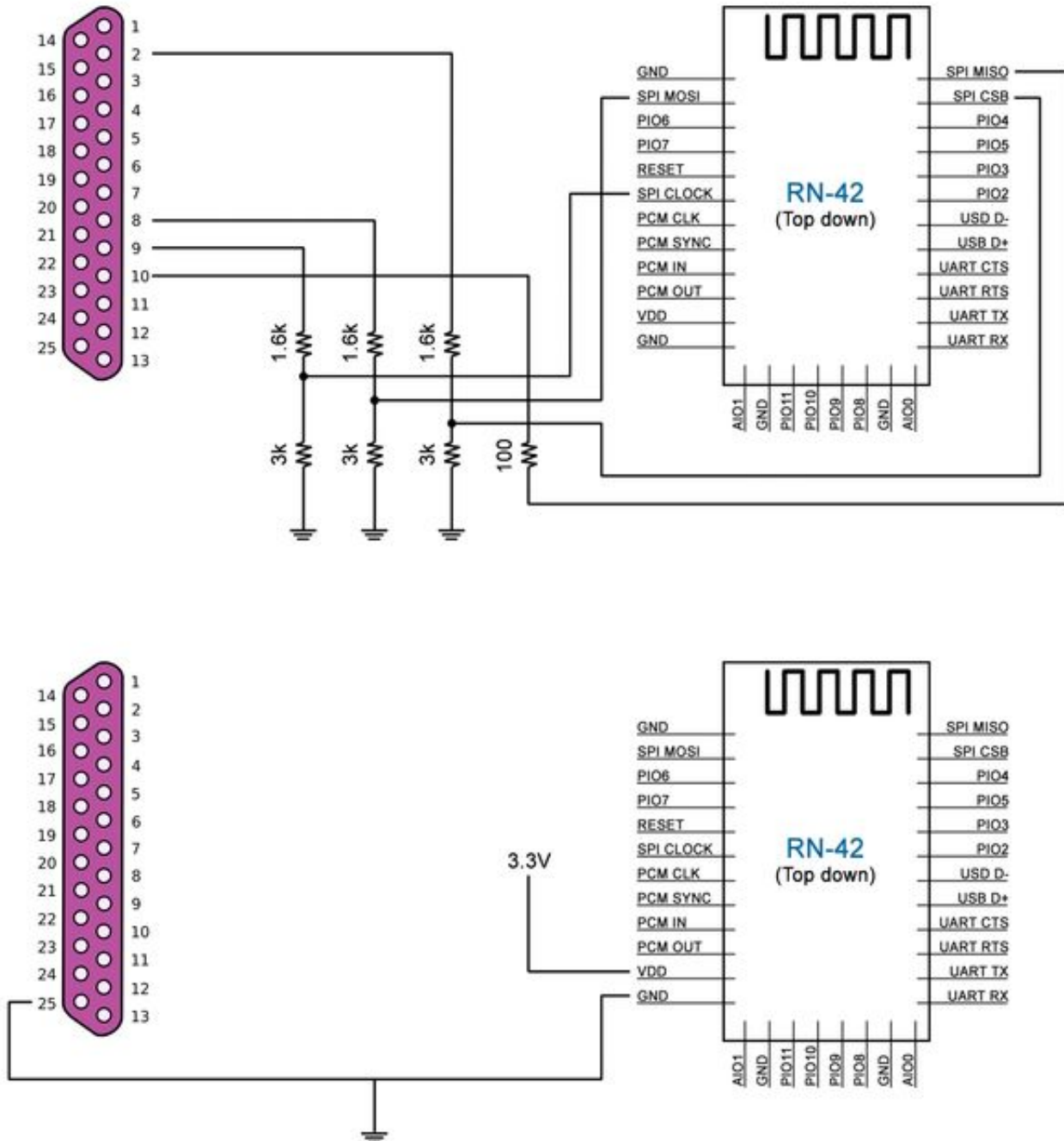
The chip manufacturer, CSR, has made their tools publicly available.

Head over to this link:

to download the manufacturer's BlueSuite toolset, which is what we'll be using to dump the module's firmware using the parallel port of a PC.

The download will require a quick, free of charge registration.

Step 4: Building the programmer



For the chip to communicate with our PC, we'll need to connect the SPI pins into these pins of the parallel port.

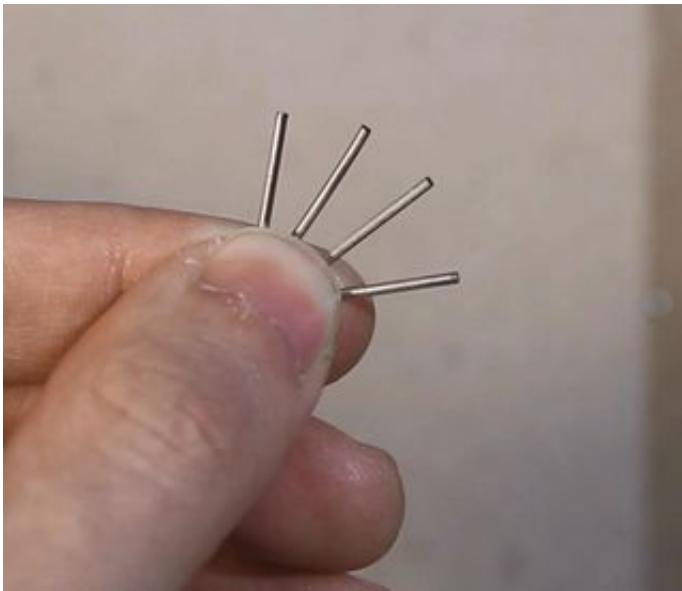
We'll need use a few resistors to create voltage dividers that will step down the parallel port's 5v transmission levels to the chip's 3.3v levels.

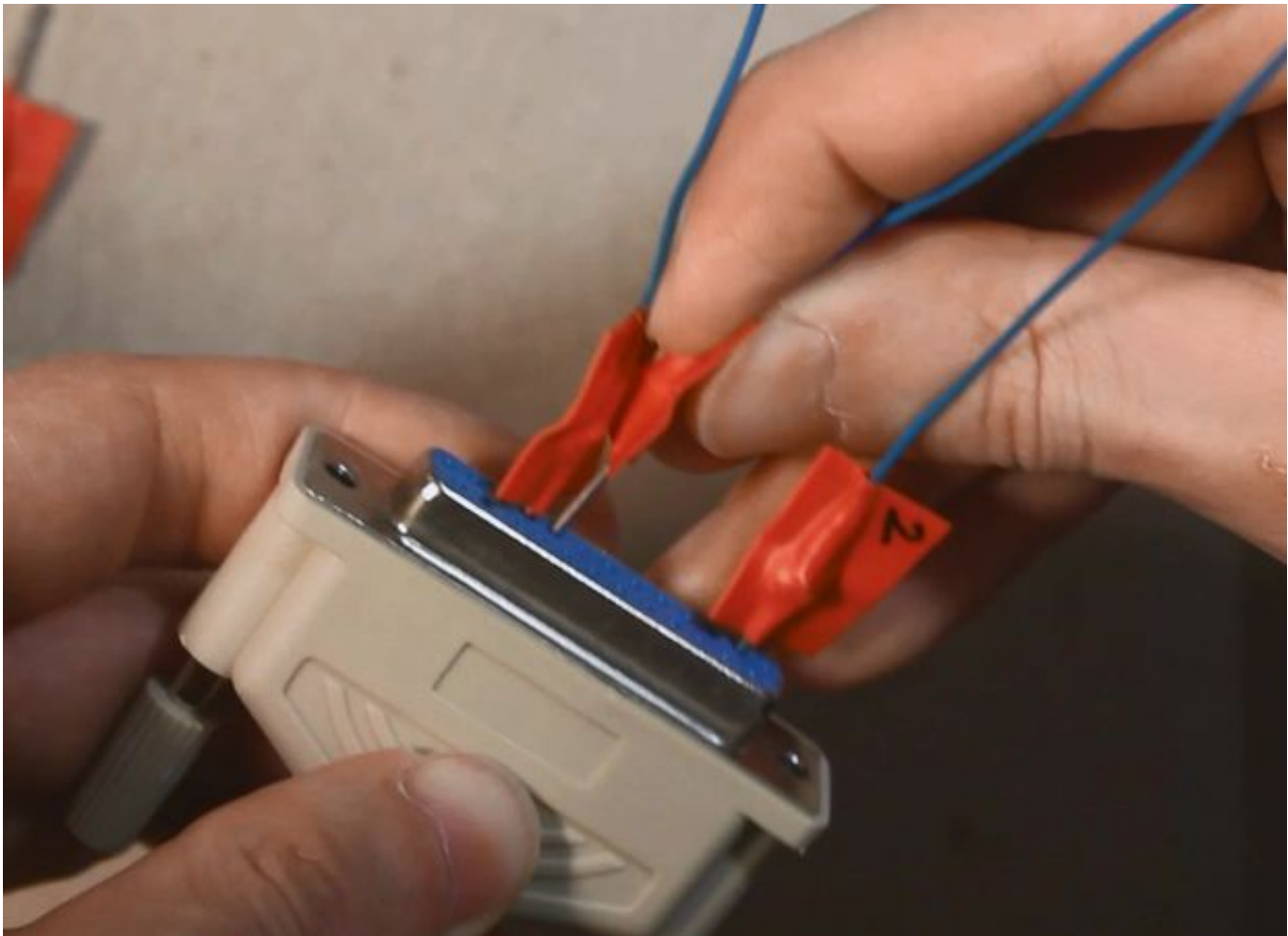
The MISO pin will transmit to the parallel port at 3.3v, which the PC will still recognize.

We'll simply add an 100ohm resistor here to prevent the the module from being overloaded.

We'll also need an external 3.3v power source to power the chip.

Step 5: Get a parallel cable

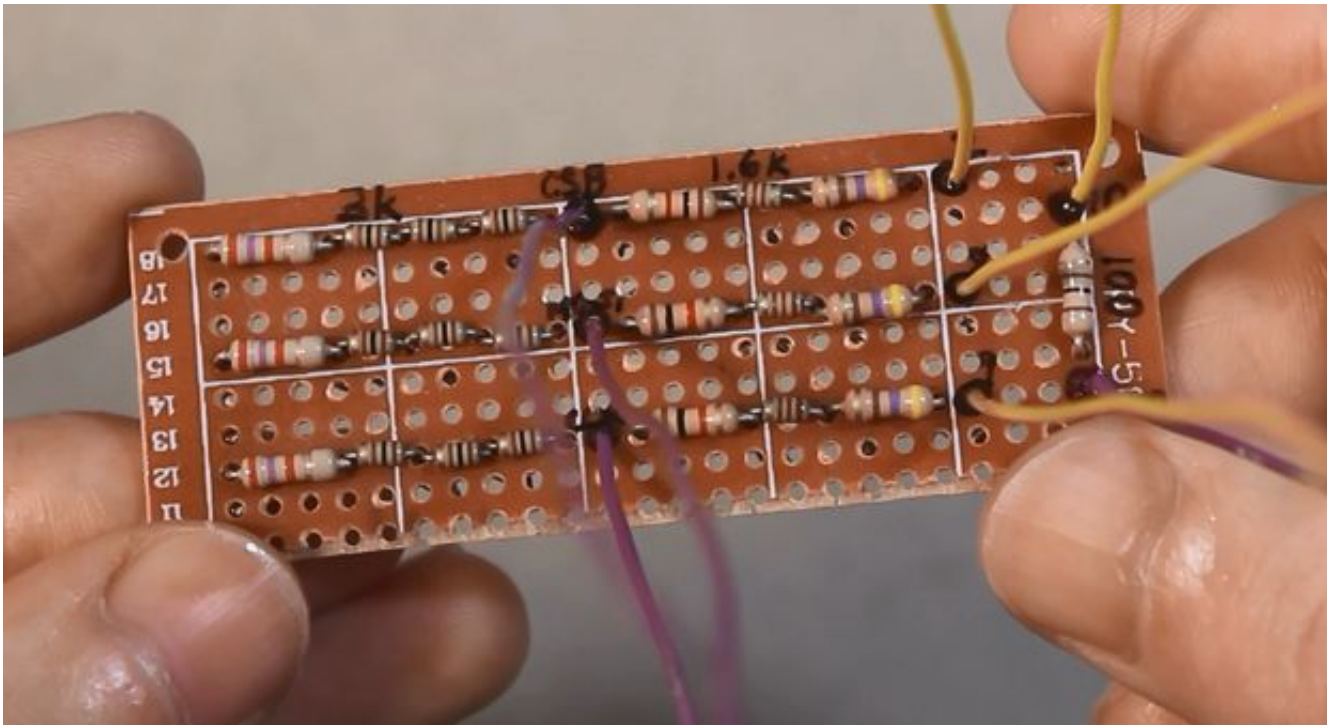




Here's the parallel cable I'll be using, the male end will connect to the PC and the female end will connect to the chip.

Paper clips will fit snugly inside the slots of the female end, so we'll cut up a few pieces and solder them to some loose wires so that we can connect it to our breadboard.

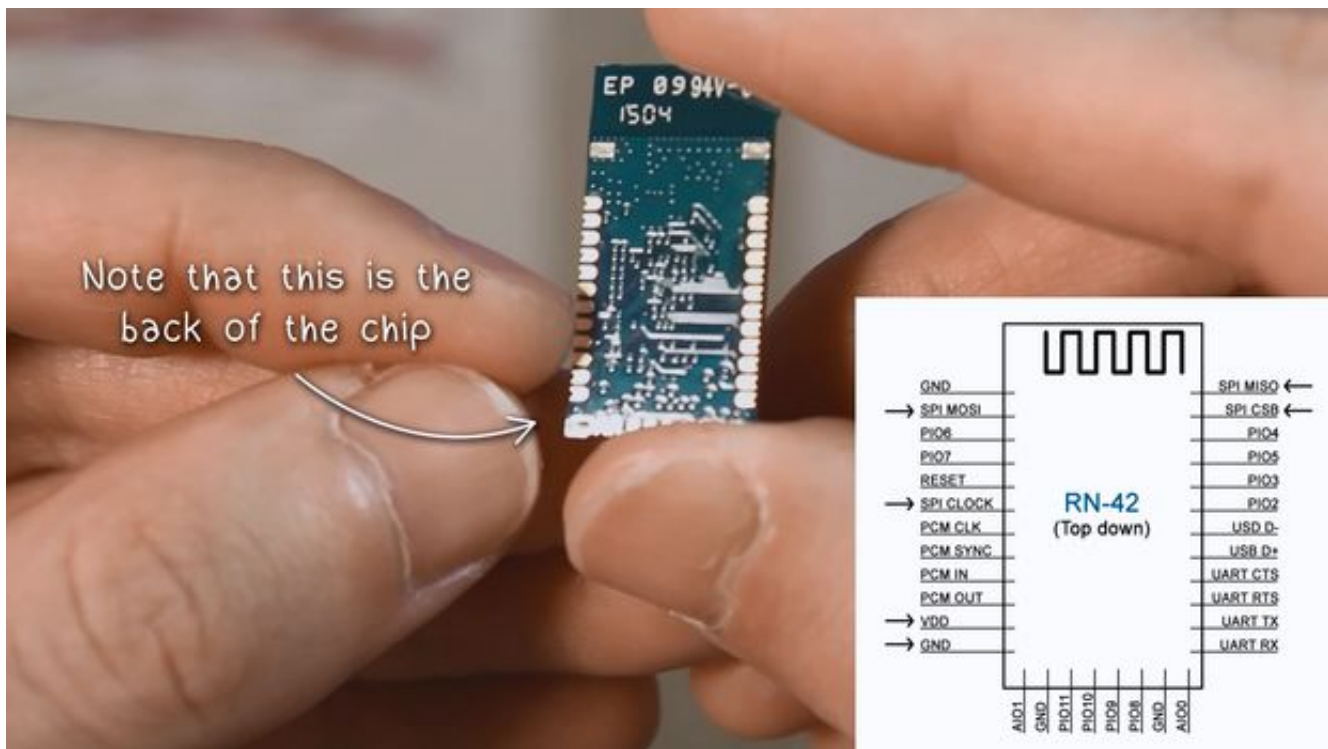
Step 6: Put together the voltage dividers

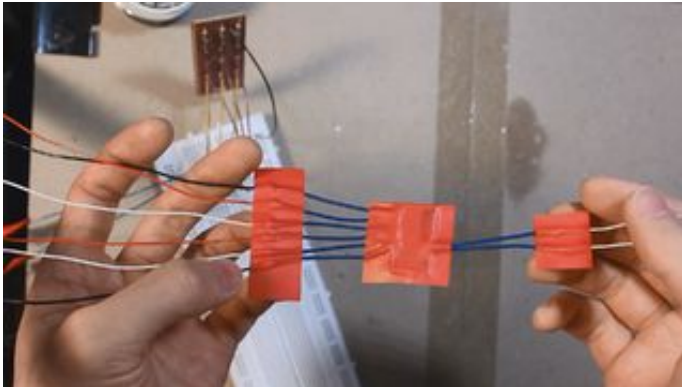
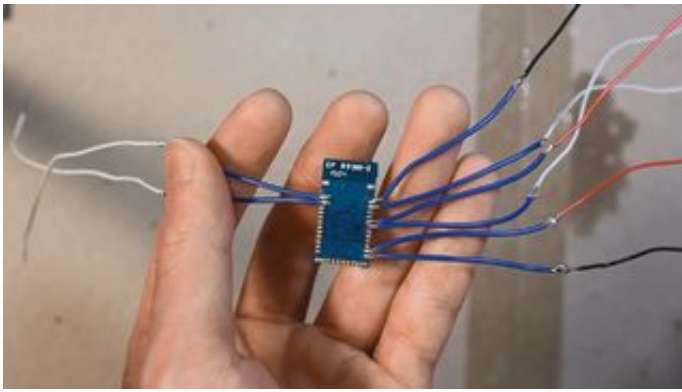


Here are the voltage dividers I made from salvaged resistors, as per the schematic in step 4.

The values of the resistors used in the voltage dividers don't have to be the same, as long as the two resistors in each voltage divider has a 2:1 ratio.

Step 7: Connect the RN-42 pins





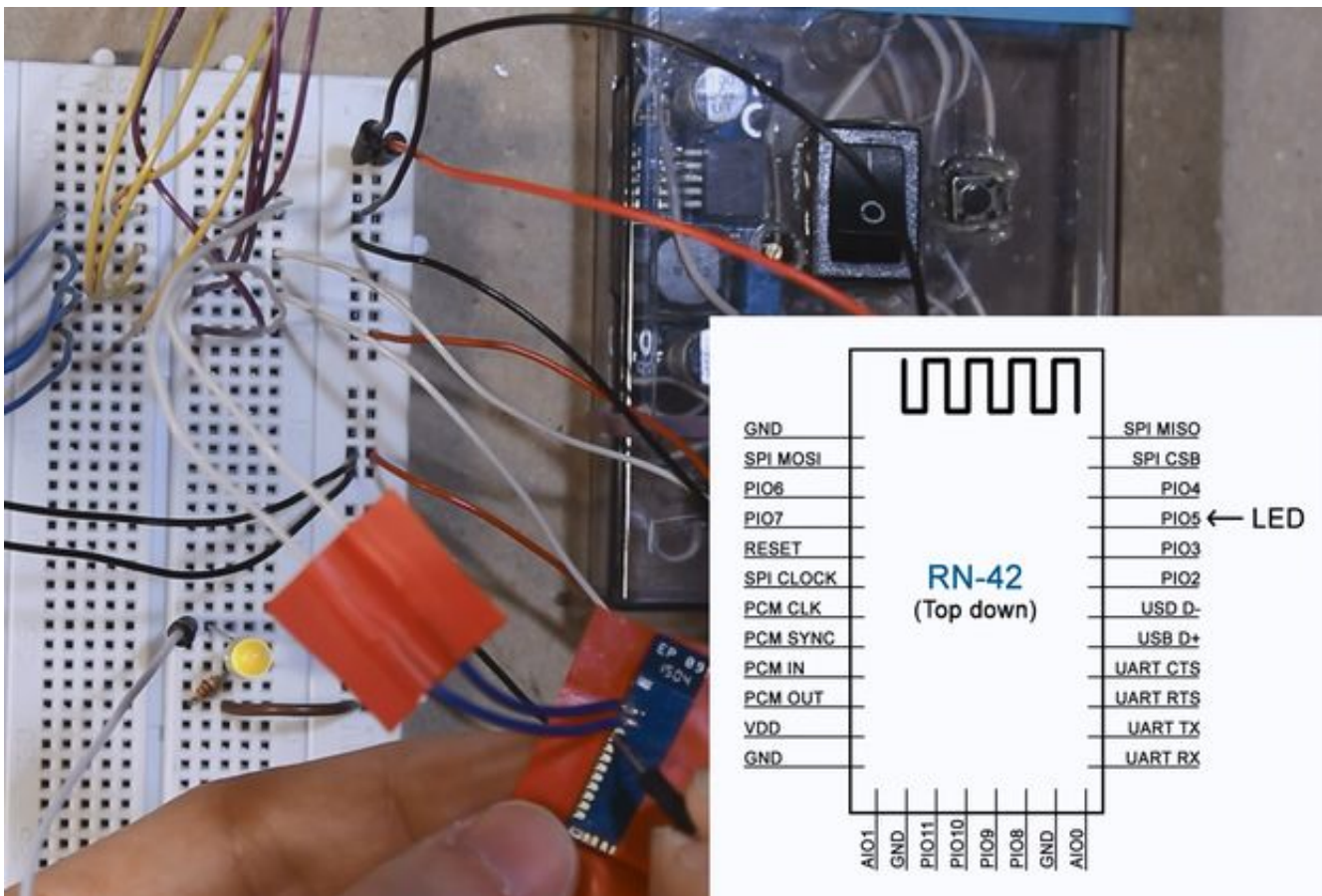
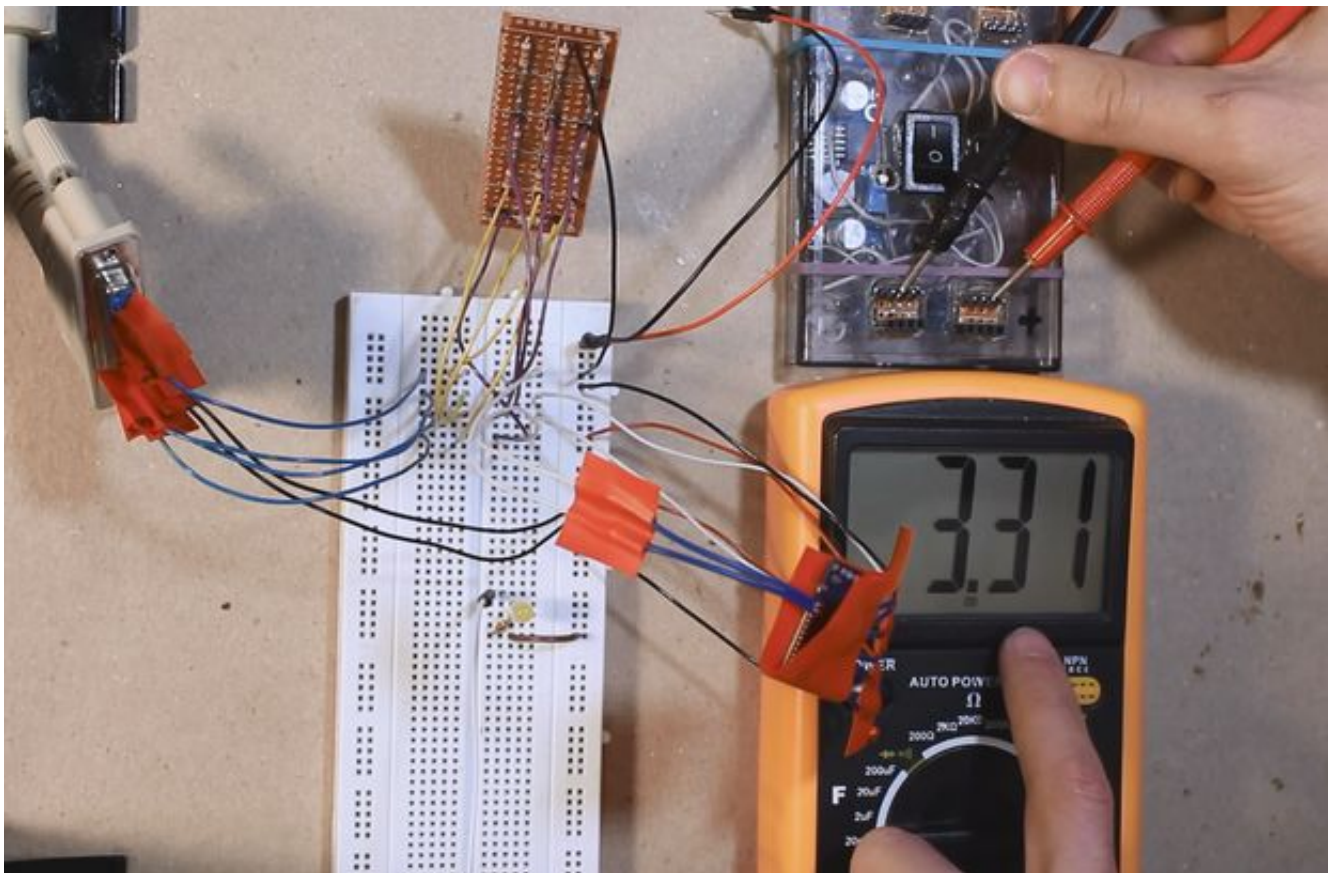
Here is the RN-42 module that we'll need to extract the firmware from.

We'll need to access the SPI (MOSI,CLOCK,MISO,CSB) pins, as well as the VDD and GND pins.

If you want to make your life easier, you could get a breakout board instead of actually soldering wires to the pins.

If you do solder the pins directly, make sure you keep the wires separated with a bit of tape so nothing shorts unintentionally.

Step 8: Power up the chip

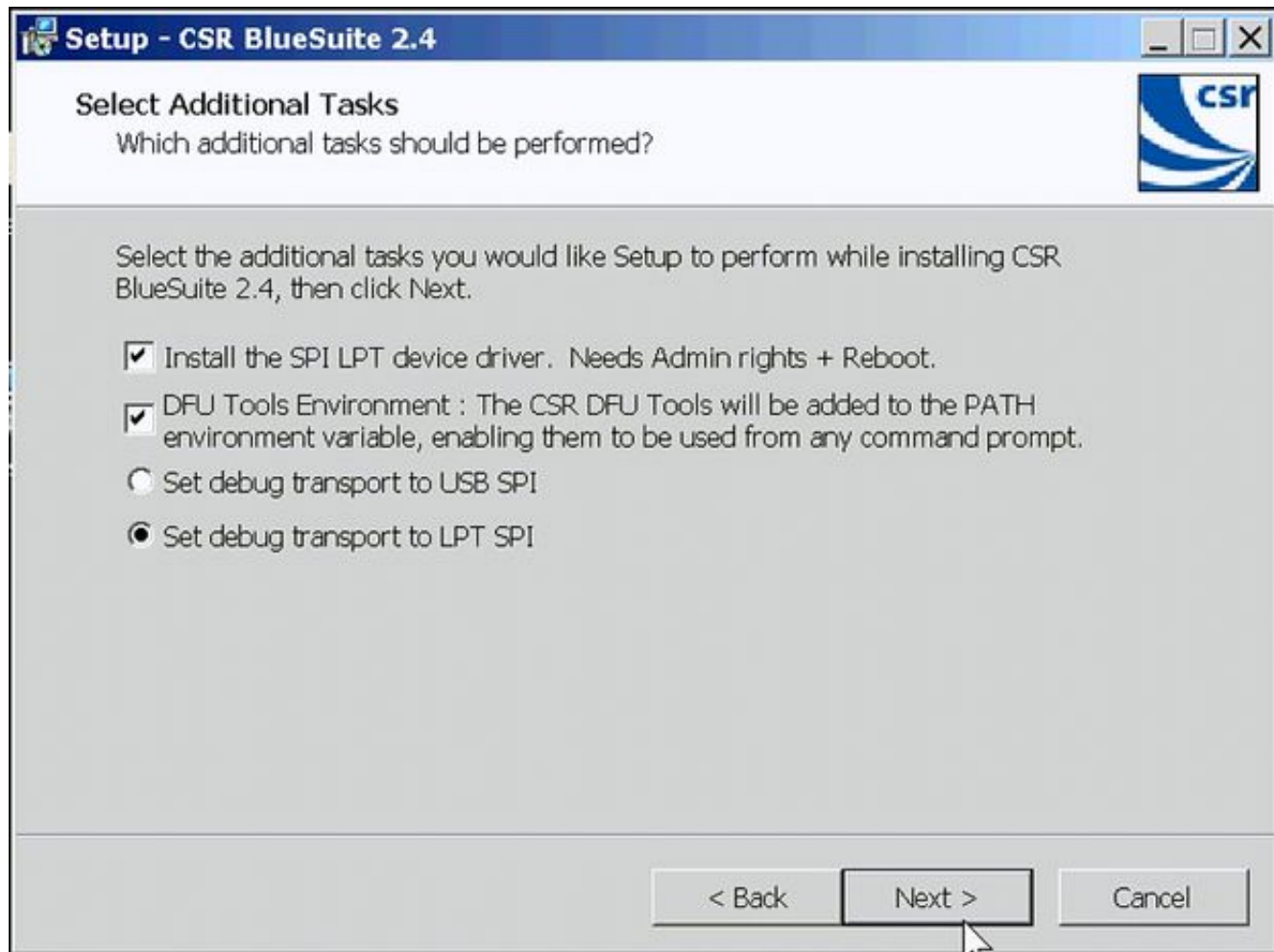


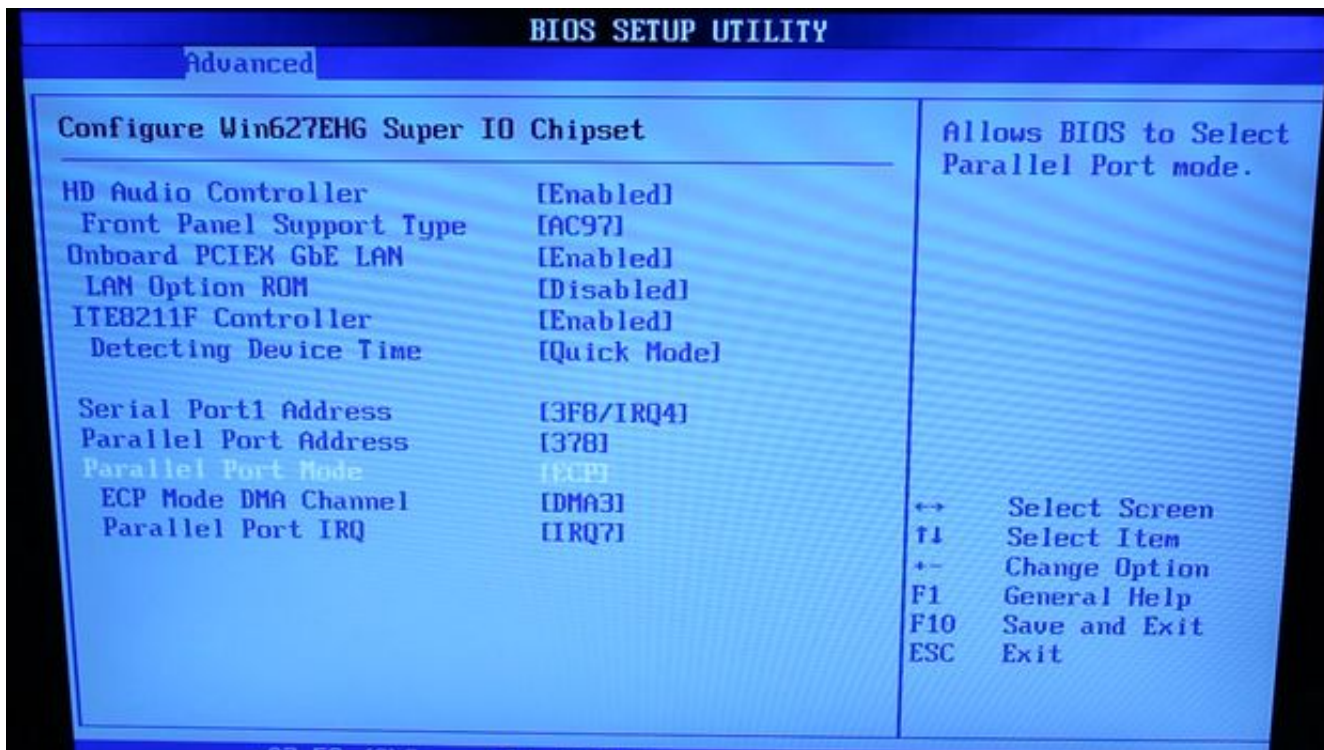
Now we're going to plug everything into our breadboard, and power up the chip.

I'll be using my regulator to output a 3.3V supply to the chip.

If you hook up an LED on PIO5, you can check the status of the module. When the chip is just turned on, you should see the LED blink.

Step 9: Install BlueSuite



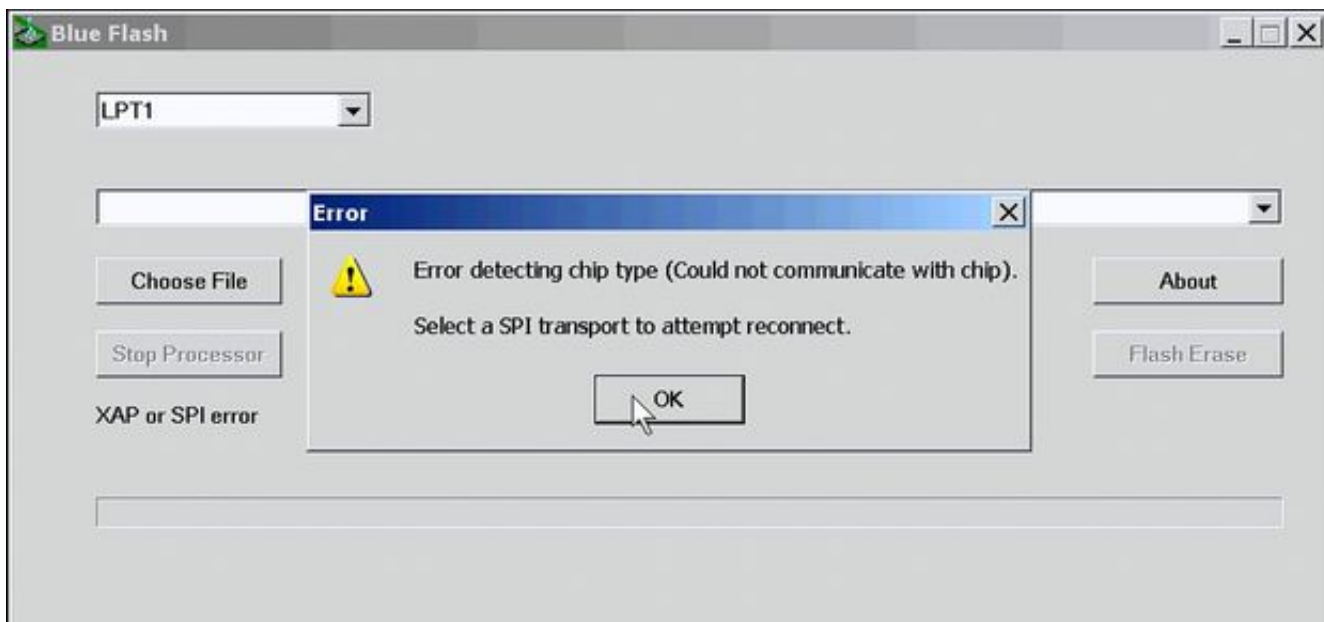


On the PC side, the BlueSuite tools have been reported to work best on a Windows XP machine, so that's what I'm using here.

Install the package we downloaded earlier - and check the box in the installation dialog to install the LPT device driver and also set the debug transport to LPT.

Reboot when prompted, then head into the BIOS to make sure that the Parallel port mode is set to ECP.

Step 10: Some possible BlueSuite errors



I've seen two errors from BlueSuite before, and they're not very descriptive as to what's actually wrong.

Here's how I solved both these errors:

1) Legacy Plug and Play detection not enabled (first picture)

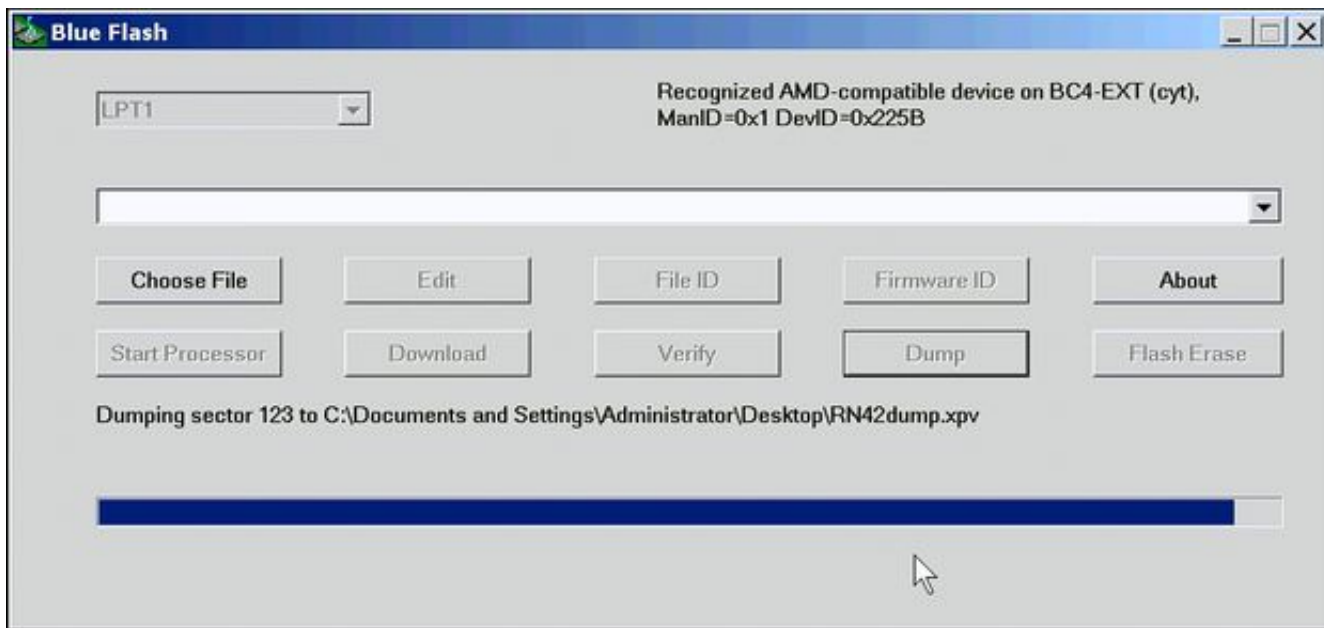
- Go to System Properties (right-click My Computer)

- Under the Hardware tab, go to the Device Manager
- ECP Printer Port, go to its properties
- Under port settings, check Enable legacy Plug and Play detection
- Hit OK and reboot

2) Chip not detected (second picture)

- Either the chip is not powered, under powered, or there's a faulty connection
- Check the wiring and try again

Step 11: Dump the RN-42 firmware



If everything went well, then you should be able to access this dialog.

If you hit Stop Processor, you'll get an acknowledgement that the chip is recognized.

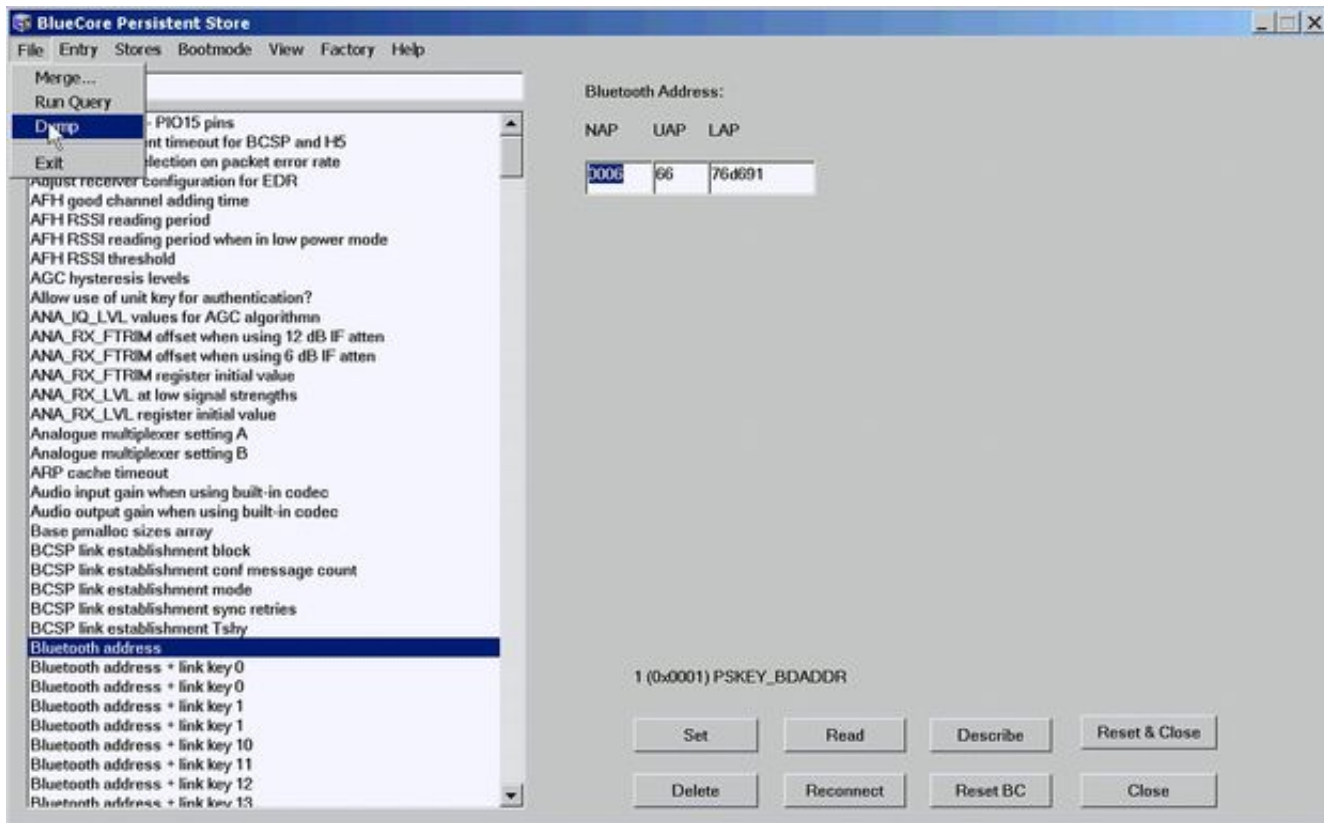
The Firmware ID button will let you see what firmware version your chip currently has.

But for now, we'll hit Dump and choose a location to save our firmware.

Fortunately, we only ever have to do this once. As long as you keep this dump, you'll be able to flash any HC-05 chip with this HID firmware.

When that's done, you'll have two files. Tuck them away, and fire up PSTool.

Step 12: Dump the RN-42 config



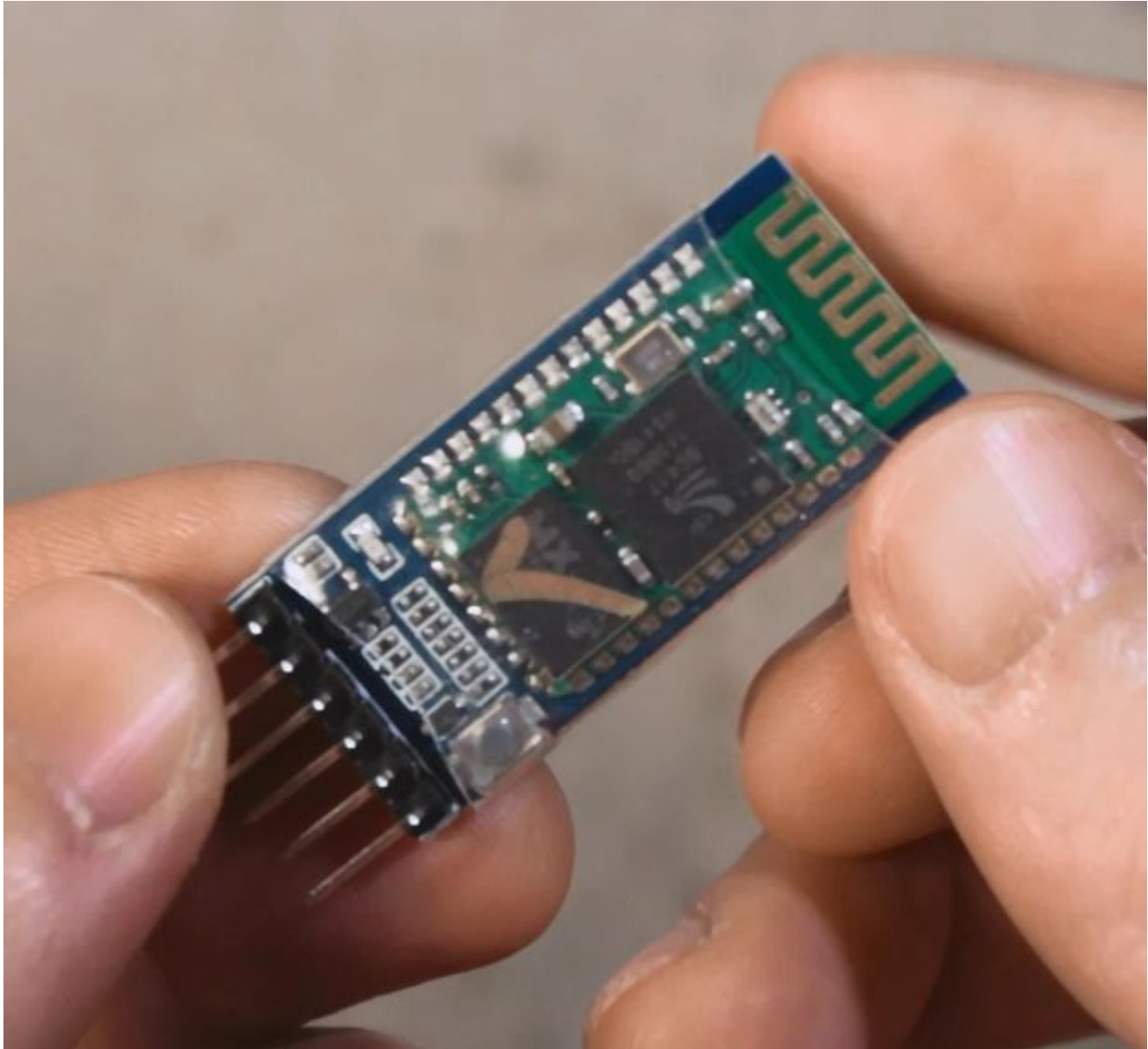
PSTool will let us manage the settings of the chip, like bluetooth address, baud rate, etc.

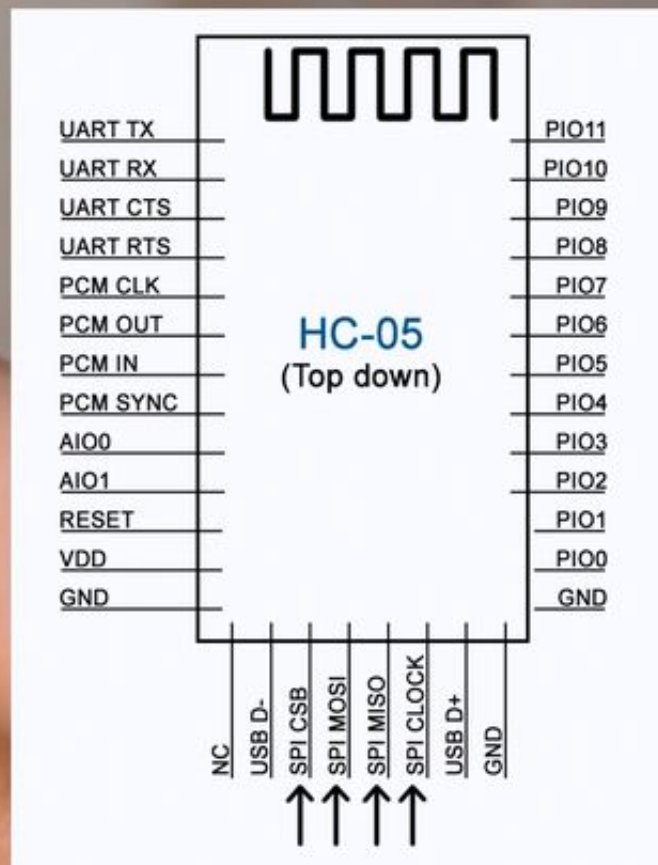
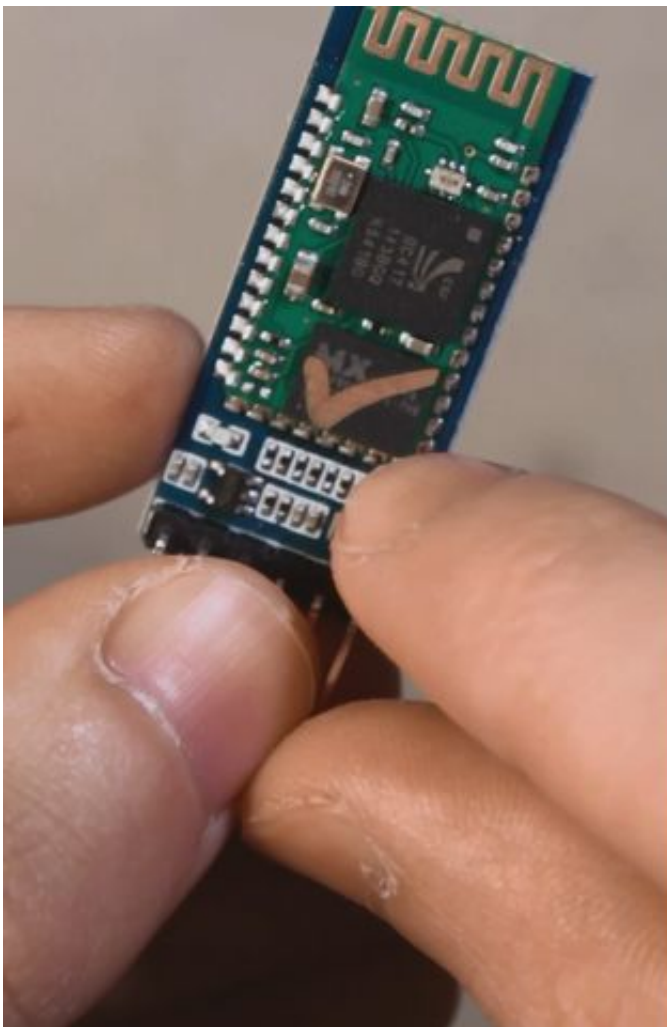
We'll need to make a dump of all these settings too. Goto File > Dump, then choose a location to save

the config file.

Once we have the three files (two from the firmware dump, one config file), we're done with the RN-42 chip.

Step 13: Prepare the HC-05



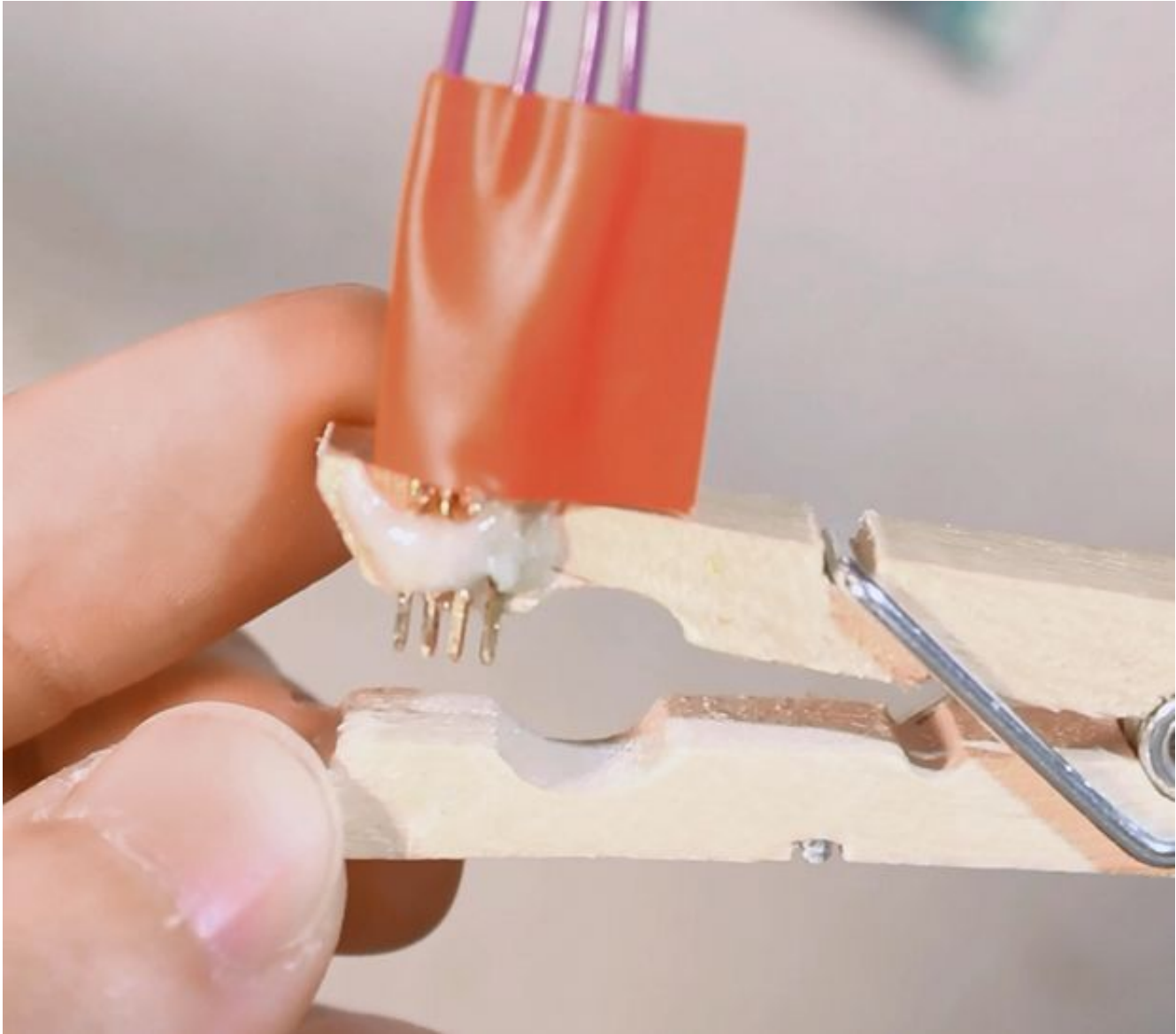


Here's our \$3 module, the HC-05.

This one comes with a breakout board for some of the pins. We'll need to cut away the shrink wrapped housing to access the SPI pins.

If we look at the pin out of the HC-05, we'll notice that the pin placement is different from the RN-42. For the HC-05, the SPI pins are conveniently grouped together on the bottom.

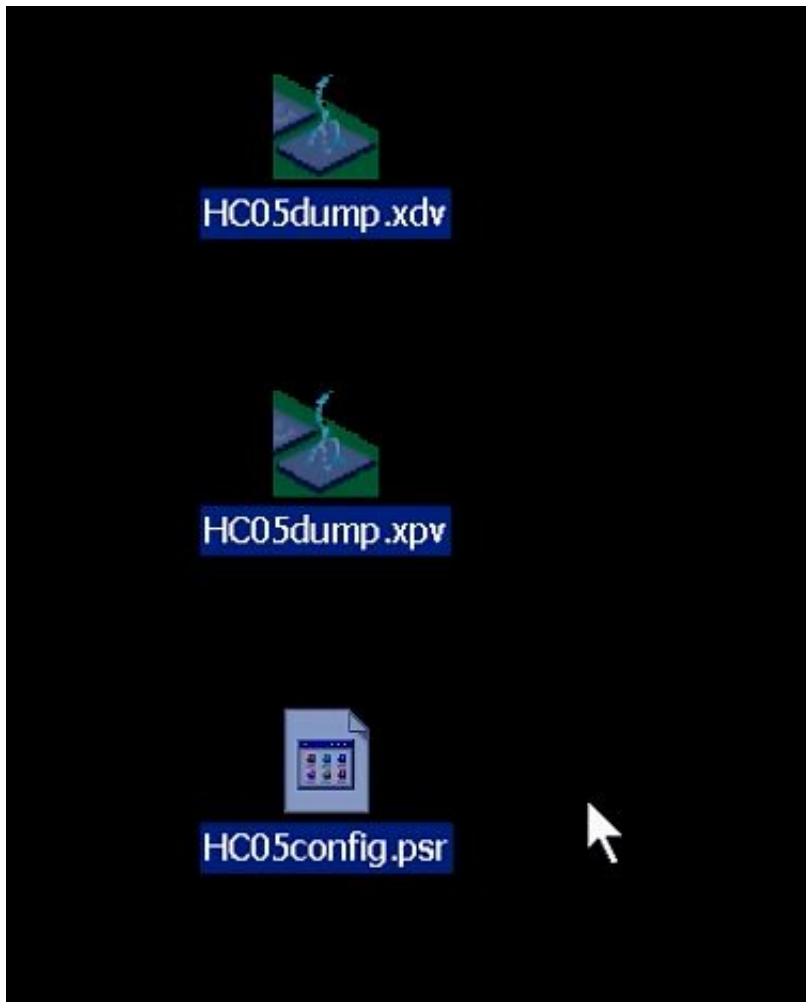
Step 14: Make a clothespin programmer



We'll want an easier way to flash these chips in the future so we don't want to be soldering wires to the pins directly.

And here's what I came up with - a clothespin with pins and wires attached. All we need to do is clamp it onto the chip and plug it into our breadboard.

Step 15: Prepare to flash the HC-05

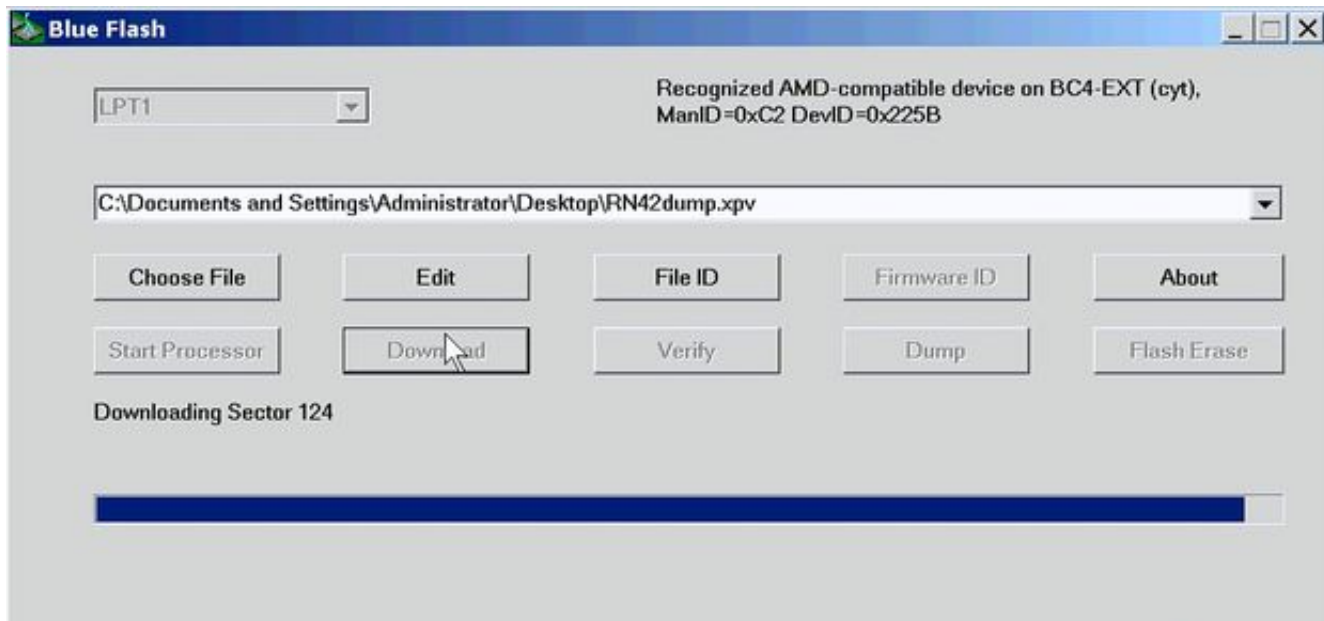


First, make a dump of the HC-05 following the exact same steps as we did with the RN-42 module, for safe-keeping.

When we flash the HC-05, all the settings of the chip will be overwritten by the RN-42 settings.

We need to preserve the bluetooth address and antenna settings of the HC-05, so open up the HC-05 config file in a text editor, and preserve the three entries: PSKEY_BDADDR, PSKEY_ANA_FTRIM, PSKEY_ANA_FREQ. Save this config file as another file.

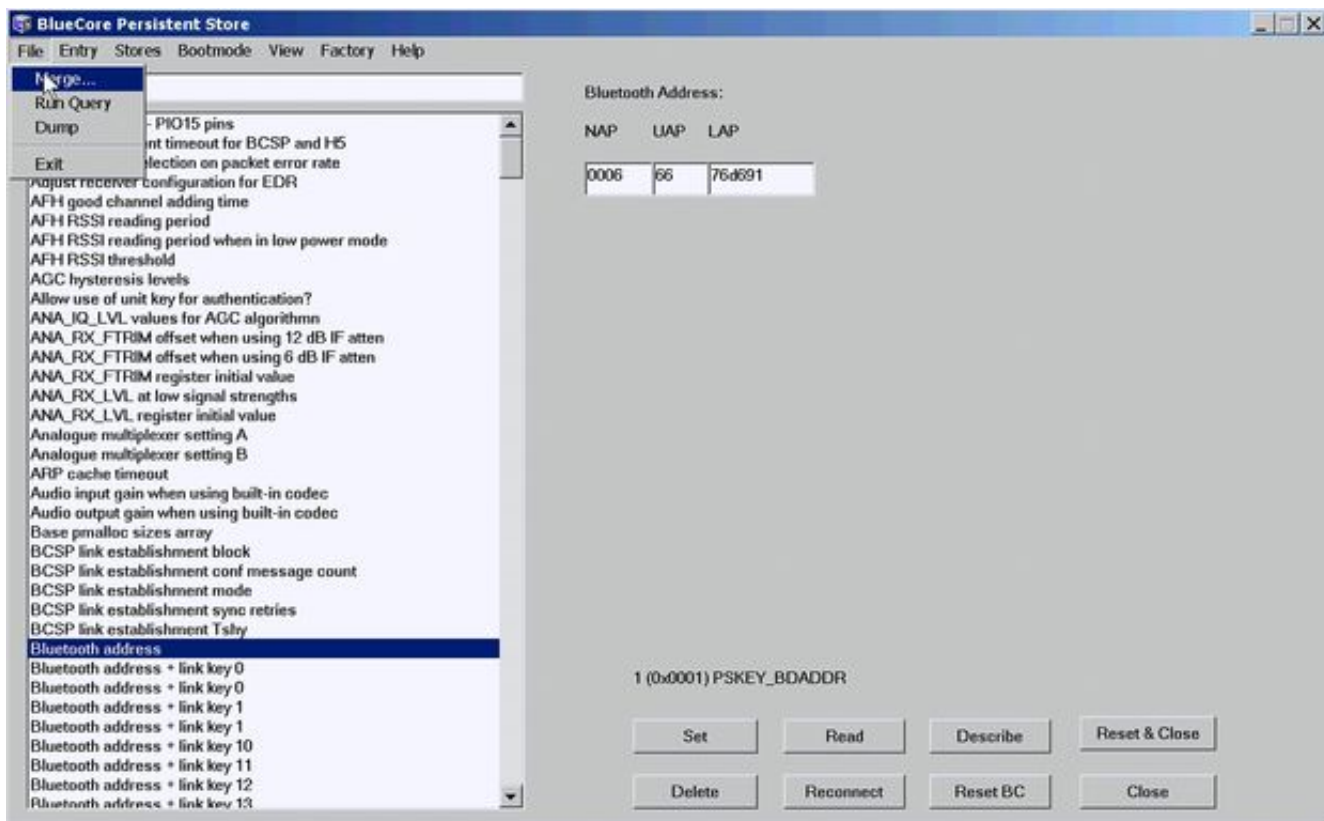
Step 16: Flash the HC-05



Now we're going to flash the chip.

Run BlueFlash, stop the processor, hit Choose File, select the RN-42 dump, and hit Download.

Step 17: Revert a few HC-05 configs



We'll need to revert the bluetooth address and antenna settings of the HC-05.

So open up PStool, go to File > Merge, and select the config file containing the three entries, PSKEY_BDADDR, PSKEY_ANA_FTRIM, PSKEY_ANA_FREQ, that we saved from step 15.

That's it! We're done!

Step 18: Conclusion

This might seem like a long process, but after doing it once, it'll take you no longer than a minute to flash a chip.

I'd like to thank Robin Gross (and his blog, Byron's blog) for being one of the first to document how to build a programmer for the BC417 chip.

You can check out his blog here: <http://byron76.blogspot.ca/>

That's all I have for you today!

In an upcoming project, I'll be showing you how to build an actual device with this chip, so come back and don't miss it!

If you liked this Instructable, then perhaps you'll like some of my other projects!

You can check them out over at my [YouTube channel](#).

Thanks guys! I'll see you next week!

