Phantom 3 Drone Repair / Diagnosis

Contents

1 Overview

User reports drone is unable to sync to mobile phone. Upon testing drone, I find that the gimbal is not able to get a proper level base. It continously moves around.

2 Let's try a Logic Analyzer

Unfortunately a lot of my work in not in notes, although I thought I had written it down, but in any case... I've looked at this drone a bit already. The issues are likely in the camera / gimbal assembly. The price of a new assembly is $200 on ebay, and extremely expensive. The drone used is about $240 or $300 on ebay now (Jan - June 2019).

The pin connector that goes to the gimbal board from the drone is about 8 pins. Not bad. The ribbon connector on the gimbal, that goes to the camera is some obscene 50+ pin, double stacked monster. Not easy to decipher without schematics. There are a few motors on the camera board,

which all seem to work. The issue is with the self test of the gimbal, at which point the drone never stabilizes. The user reported that he had a crash, but he also reported that he replaced the gimbal board.

2.1 Drone Mobile App

Looking at the drone in detail, it relies on a mobile phone app to work with the drone. This app doesn't work on my old phone. I tried an Ipad model one, and that didn't work either. The only phone I was able to get the app to work on was a more recent apple phone (that I do not own). So, right away we are having trouble interfacing to this device. The future doesn't bode well for this drone. What will 10 years in the future be like?

I've somewhat given up on any interest in repairing this. It's a black box, and cheaply made. Instead, I'm going to tap into the 8 wires going hetween the drone and the gimbal and just take a look. I've needed an excuse to use my Logic Analyzer for a while, and here's a good one. Let's see what / if we can learn, if anything.

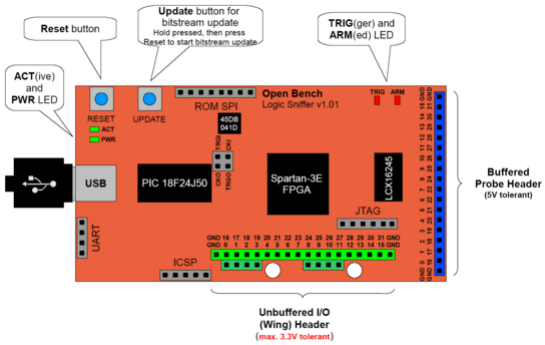


Figure 1: Open Bench Logic Sniffer

2.1.1 Open Bench Logic Sniffer

The Open Bench Logic Sniffer is the first result that comes up when you search for Open Source Logic Analyzer. It's from 2010 by Dangerous Prototypes whom also made the Bus Pirate.

First off, I need a case for this board. It's exposed. I found two on thingiverse. Let's fire up the 3d printer and get one of those made. Looking at the two cases, they are nothing more than bottoms for the boards. Helpful, but I would've liked a top on these. A bit too rushed. (see cad folder). I'll just throw some electrical tape on the bottom, about as good as the cases.

2.1.2 Abandoned Project?

The github[1] repo https://github.com/GadgetFactory/OpenBench-Logic-Sniffer/issues seems to be quiet. Is the project abandoned?

2.2 Setup Cont...

Let's get this thing. Running. On devuan ascii (d9), follow the quick start guide here:

http://dangerousprototypes.com/docs/Logic_Sniffer_quick_start_guide Linked from the main logic sniffer page.

It looks like the Sump program runs as a shell script (without any installation), so you can follow that page where they link to here:

http://www.lxtreme.nl/ols/

and download the latest client. Then extract, untar and run the program. Unfortunately there is no verifcation of the tar files that I see. A bit shady. A package for a distribution is warranted here... Also looks like the Sump client is customized for Open Bench Logic Sniffer, and as a result, hasn't been updated as the project has been somewhat quiet...

2.3 Basic Test

The demonstrations section shows some basic tests. They use a bus pirate, but instead, you can just plug in an Arduino Uno (much easier). Here's what you do.

- Plug in Arduino, set it to Serial Read example, confirm that some data is being spit out.
- Plug in TX of UART on Uno to pin 0-7 of the OLS (open logic sniffer). Also, obviously you'll need to tie the grounds together.
- Set the sampling frequency to be lower than the 200MHz default. Lower sampling frequency means longer sampling times, but less detail.
- Do a sample, then go into UART decode mode in the menus, choose auto detect speed, and run a sample, making sure to assign the TX pin of the UART decode to whatever your TX pin is connected to in OLS (this is somewhat obvious, so just fool around with the program until you get it).

[1] Not a fan

- Note that what speed works for your given application may vary. With a 9600 baud UART, I can do 50KHz sampling and get everything, but 5KHz returns garbage. Again, the lower the sampling rate, the longer the sample.

Everything should work. The decoding on the main time line view is poor, but the UART printout works reasonably well in its own menu.[2]

Some things to do: enable side measurement window. And go to preferences - theme - Logic Sniffer. Looks better.

I think I've seen enough, let's try to connect it to the Drone.

3 Drone Logic Sniffing

I was looking at the feet of the drone, and noticed something I had not seen before. There are two wires (antennas or thermocouples) on two legs, and then on a third, there was a board, with four wires going to it. The wires appear to be I2C, as there is a GND, and an SDA, SCL wire. I plugged the OLS into this, and started recording, but then the battery died. I did realize that in order to get the sampling rate right, you'll have to first probe with an Oscilloscope, and at least look at the signal first. Then after you have an idea of what you are dealing with, you program OLS and go after decoding.

3.1 I2C Decode

The pins on the board indicate SDA and SCL, which is I2C. Using the Open Logic Sniffer, at 2MHz speed, I am able to get 12ms of sampling. Unfortunately, this is not enough to read more than one packet. The packets are sent from the Phantom at a rate of one per every 20ms. Thus, the limitations of this speed show up.

Another drawback, is that it's not seamless to re-sample, when using OLS. In order to get I2C decoding, I have to begin capture, then go into the measurement menus, and click analyze. In order to get a 2nd sample, I have to close that analyation window, and click begin capture again, and then open a new I2C decode window.[3] The workflow, is poor for this use case. Ideally, I could capture, and re-analyze in the I2C window (perhaps one button), without needing to click through menus.

[2] Although export is limited, which I will get to.

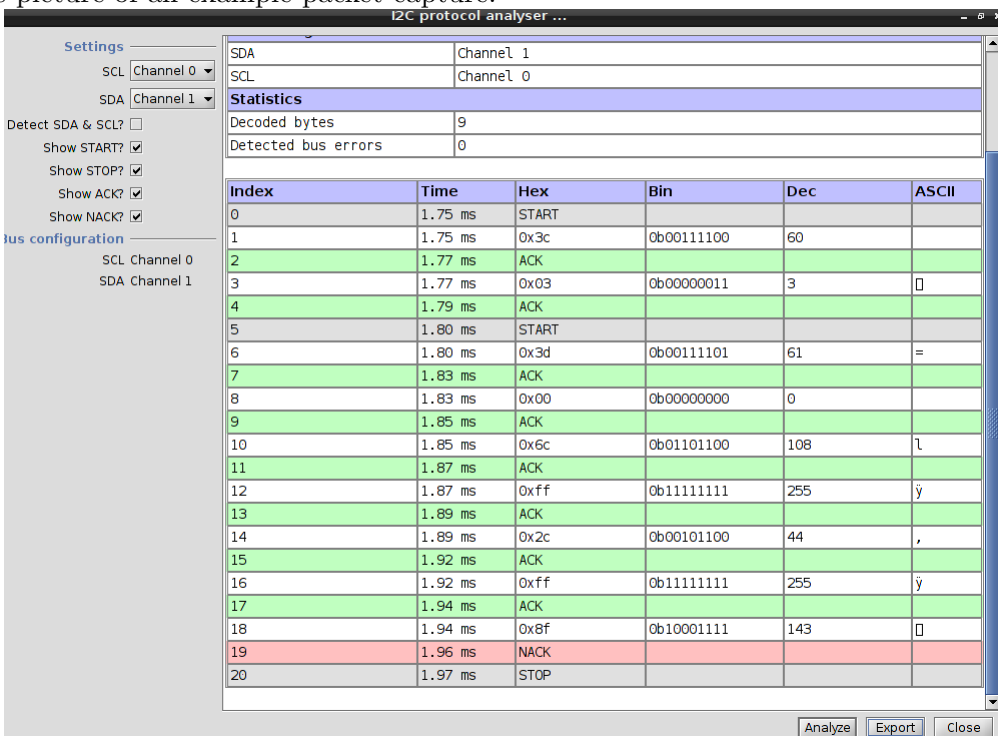
[3] And if you aren't using the default pins, e.g. in an SPI protocol decode, you have to change all the pin values again...

The limited sampling issue is addressed on the Dangerous Prototypes website: *"The major difference is that the OLS has a limited number of samples, while the hobby USB analyzers can theoretically take infinite samples. Most of our debugging is done with the first few hundred samples, so this feature isnt usually important to our work, your situation may be different. Future versions of the OLS will definitely have more sample storage, and we can work an infinite sampling mode into a future firmware update too."* Seems that they acknowledge this as a flaw.

In any case, I'll have to get a Saleae, I think... In order to do further testing. I'll buy one used. I don't wish to use a clone, and I can't afford to pay $150 for one.

Seems like the OLS is more a demonstration of making an Open Source Logic Sniffer, rather than an actual developed open source logic sniffer. At least in this scenario of 2MHz speeds with packets being burst. Perhaps it works better for slow sampling rates.

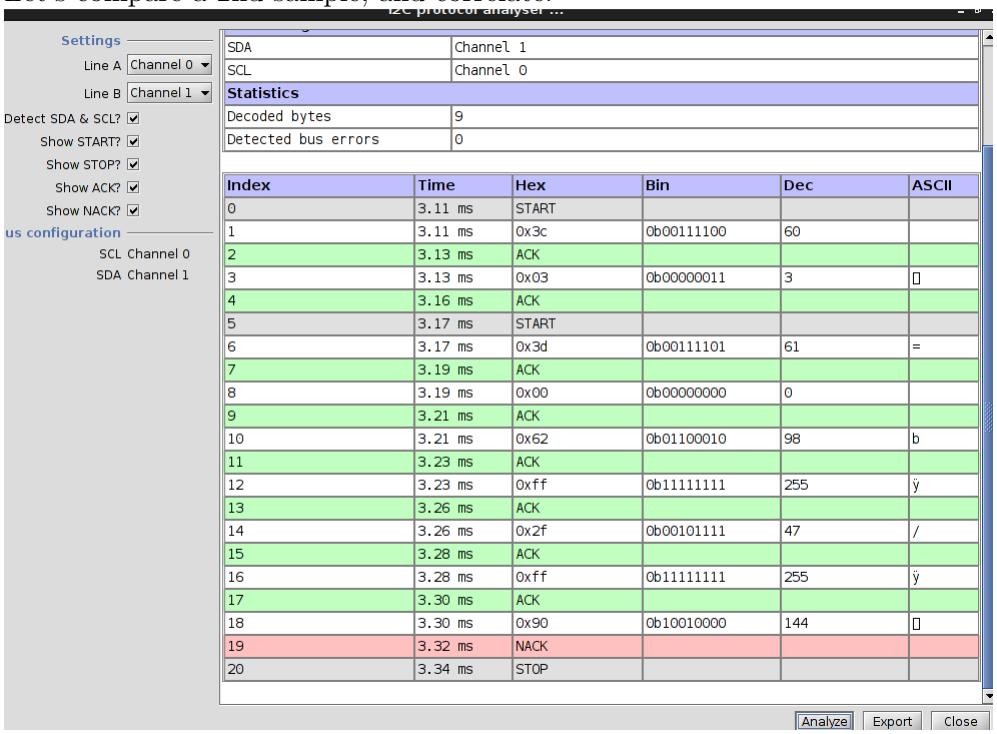
Anyways, what did we find? First off, you must get the sampling rate right. At 1MHz there were decoding errors, but at 2MHz, no errors. See this picture of an example packet capture:



Exporting it, results in a csv file that lacks the binary output... Not sure

why. It does have ascii. Another limited option. [4] From the output, we can see what is likely a bit being written from the Phantom main board, before the sensor on the leg sends out the data.

Let's compare a 2nd sample, and correlate.



Here we can see that the write code is ascii 60, address is 3, then the return data is a number, followed by a 0, a number followed by 255 (0xFF), another number, another 0xFF, and then a final number before closing the connection. Clearly the foot part is some kind of sensor, maybe vibration to detect a landing.

[4] I don't have a problem with this product being undeveloped, but I do have a problem with the way its touted as a working product. Call a spade a spade. This is an open source project that is in development. By no way is it mature enough to be sold to the average hobbyist. They are too eager to tout the product as an open source logic analyzer. Well, it is, but it's not finished... It's an in-development open source logic analyzer. Basic functionalities in the software and hardware are lacking. Don't sugar coat it. Call a spade a spade. Maybe it's my fault for not realizing the company is called Dangerous Prototypes. I mean, it's in the name... Prototypes. Though the bus pirate is generally considered mature enough for the hobbyist. The problem is that if you search Open Source Logic Analyzer (as I did) this shows up as the main result. In defenes of the Logic Analyzer, it is still good for what it is. Perhaps if development hadn't stopped, they would've been able to fully develop the missing features.

3.2 Tapping into Main Wires

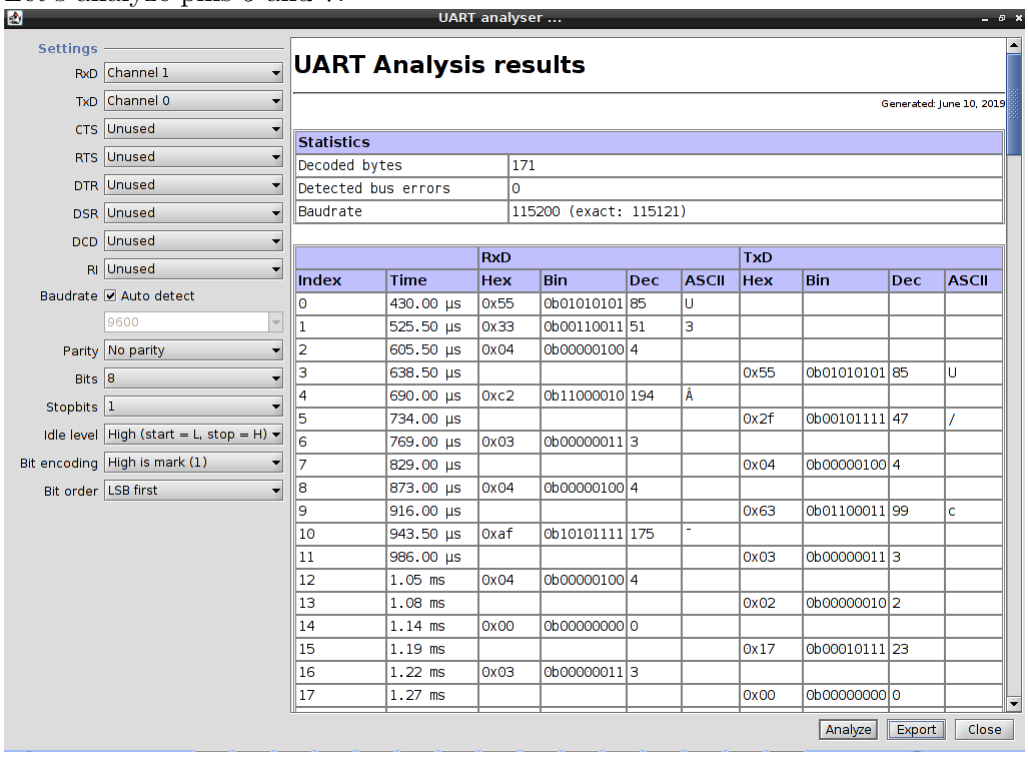
To tap into the wires, I will use a sharp exacto blade[5], to slice the insulation off, then connect the logic probe to the exposed wiring. Being careful not to cut the wiring, I peeled off the insulation. One mistake I made, was making a straight line of cuts. It would've been better to stagger, or offset the cuts, to avoid shorts. Actually, I can still do that. The Ground pin, is pin 1.

Let's check the signals with a scope to make sure they aren't over 7V, otherwise, I will break my OLS.

Pins 1 being ground, I'm not sure what 2 is. Pins 3, and 4 are 12V. Yikes!!! Can't connect there. Pin 5 appears to be data. Pin 6 is about 3.3V. Pin 7 is some kind of data. Pin 8 is about 3.3v. Looks like we only need pin 5 and 7.

The motors are getting stinky and hot, being unconnected to the drone, and trying endlessly to run. There doesn't appear to be a failsafe in the firmware to stop the motors from burning themselves out. I have to power the drone off after a couple of minutes.

Let's analyze pins 5 and 7.



[5]Exacto 11SS blades

It appears to be a UART at 115200. RX and TX between the drone and the gimbal board.

What is it sending? It's too hard to get this via sampling with the OLS. I'd be better off connecting the drone to my laptop and using a USB-to-Serial adapter. Let's do that, now that I know the baudrate.

3.2.1 Confirming Baudrate from an oscilloscope

How to confirm the baudrate? Get a sample of a bit on the UART on the scope, find its timing, and contrast with the speed. I have a High signal that is 10us roughly. 10us is 0.000010 seconds. 1 / 115200 is 0.000008 seconds. The numbers line up. 1 second == 1,000 milli == 1,000,000 micro == 1,000,000,000 nano == 1,000,000,000,000 pico seconds. [6]

Unfortunately, the UART is all garbage, or at least not decipherable. Samples in resources. It's possible there is too much capacitance on the line, or something is hurting the signal, as per:

http://web.archive.org/web/https:// reverseengineering.stackexchange.com/questions/12523 /routers-serial-port-only-outputs-garbage

3.3 Open Bench Logic Sniffer Review

Overall, it's a good project, but I would like to see in a project like this:

- A) Not abandon it, or if it's abandoned, indicate that development has slowed down in an obvious place.
- B) Tout it as what it is, an in-development open source logic sniffer. They should emphasize that basic functions are lacking. Tell the whole story.
- C) Would've like to see them personally sell it. I feel that other companies have benefitted more from their products than they have.

3.4 Other Approaches to troubleshooting the Drone

There is a USB port. I bet I could get debugging info out of that if I had the right tool. There is also a USB port on the camera gimbal. Clearly, I am missing a tool that I need to do the job. There must be an easier way.

As a concluding note, I gave it a try, and I made some progress, but no cigar here. I can always try again later.

[6] Almost seems like these should be named neg-thousand, or neg-million, neg-billion, and neg-trillion... Or similar. Would that not be simpler?

4 References

http://dangerousprototypes.com/docs/Open_Bench_Logic_Sniffer

https://github.com/o-gs/dji-firmware-tools/issues/109

http://web.archive.org/web/https:/ /reverseengineering.stackexchange.com/questions/12523 /routers-serial-port-only-outputs-garbage