Idiot's Guide

to

Olliw's GA250 Firmware

Table of Contents

1. Introduction

The GA250 is with its price of $9.99 a very reasonable MEMS gyro. The hardware itself seems to be capable, but unfortunately is its firmware not utilizing it to its full potential. But not only that, the firmware has proven to be completely unusable in certain configurations. To make a long story short, what the GA250 needs, is a better firmware. Even better, when lots of parameter could be adjusted.

With OlliW's firmware, the GA250 works a lot better and there are many adjustable parameters. These parameters ***can't be adjusted by the transmitter***, but through a Serial/USB connection by ***using your computer***. For this purpose, Olli created a tool he named “AVRConfig”. From here all parameters could be changed.

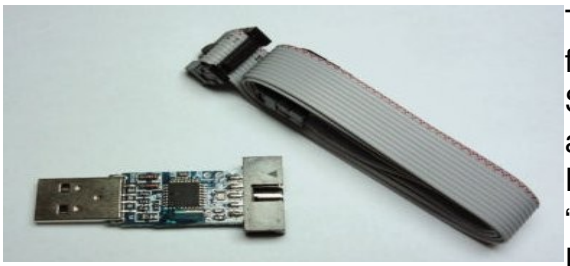
To make it more convenient at the flying field, Olli also created a separate hardware called the ProgBox. It is basically a Robbe Roxxy ESC programmer, which needs to be re-flashed with Olli's programming box firmware. So with the “ProgBox” you can also change all the gyros parameters. But its not mandatory to have it, as all settings could be changed conveniently from your computer. So there are two different ways, to adjust the settings of your gyro.

2. What do we need?

I am not an electronics expert, but did the one or the other hack I found on the web to my gadgets. Basic soldering skills are necessary, because you need to solder some cables to the GA250 board for flashing the bootloader.

To reflash the gyro and later change the settings, there are two little devices you need. Both could be found easily at eBay. The sellers from Hong Kong usually ship these devices for free to you and each of them is below $10 USD. One of them is now also available from HobbyKing.

> AVR Programmer



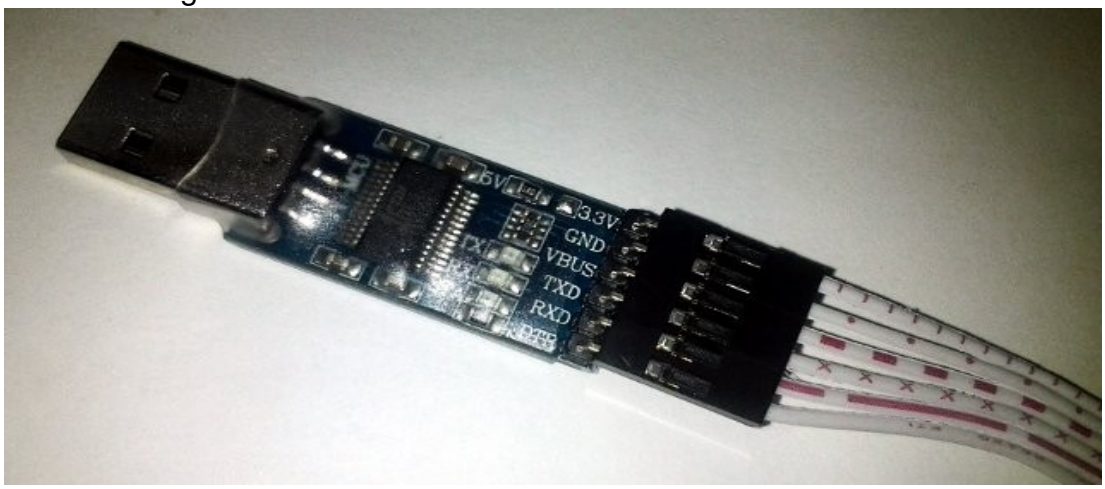
The AVR programmer will only be used once, to flash the gyro. Later a connection through the Servo plug will be used, to update the firmware and change the settings. There are plenty of AVR Programmers available. I can recommend the “USBASP”. I got mine from Hong Kong through Ebay. Or you just buy it from HobbyKing, who just started to stock it as well:

http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=21321&aff=387352

However, it is not necessary that it is this particular programmer, **any AVR programmer** will do.

> FTDL Adapter (serial Adapter)

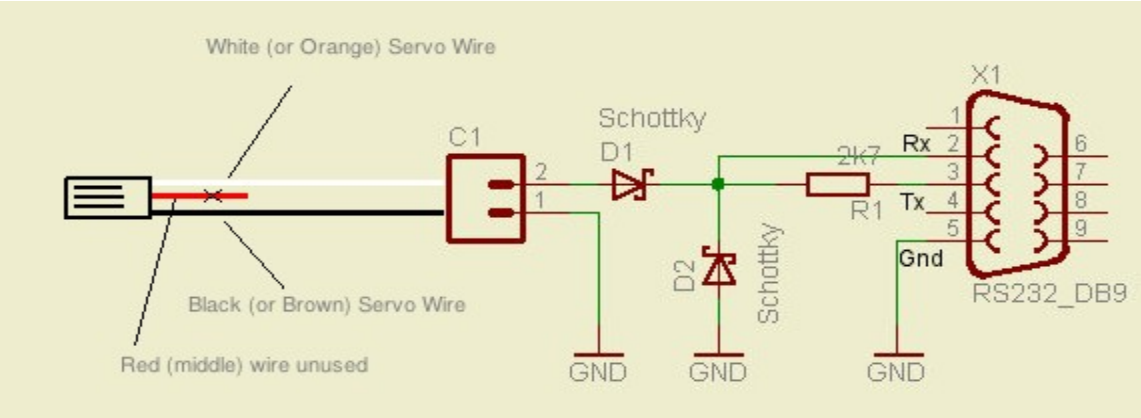
With this adapter, you will make a serial connection through the servo plug of the gyro. This is the tool you will use most of the time, when you are dealing with the gyro. There is a certain USB to serial adapter with the FT232RL chip on it. You can check eBay and search for that chip. These are also sold as “Arduino USB Programmer”. I bought mine at eBay for $9.99 with free shipment from Hong Kong. So try to search for “FT232RL” or “Arudiono USB Programmer”.



> The DIY alternative to the FTDL Adapter

You can build a special serial adapter by yourself, which could be hooked up to any USB serial port adapter or the serial port of your computer. For that a few parts from the electronic shop near to your place are necessary.

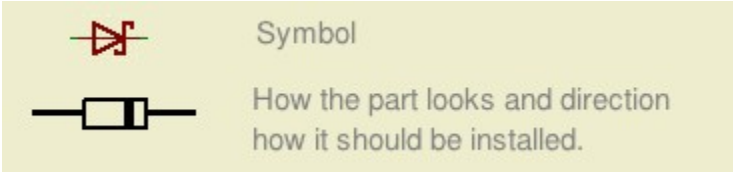
2x Shottky Diodes (BAT85, SB 120, …)
1x 2.7k Resistor (for me 2x 1K=2K worked)
1x DB9 male connector
1x Servo Cable



The adapter soldered together looks like this:

There is probably a smarter way to build it, but I am not an expert so I just followed the wiring diagram as close as possible. This adapter works for me very well. You just have to

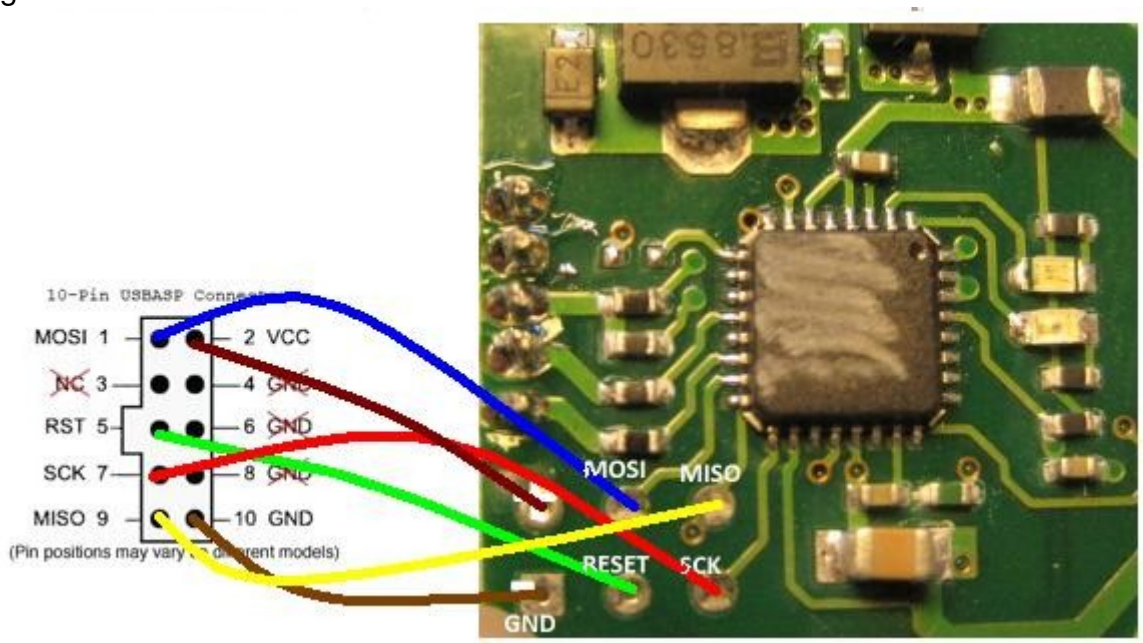
be careful with the diodes. They have a black bar as a marking in which direction the current flows. Similar to its symbol in the wiring diagram. Its very important, to install the diodes in the right direction. To simplify it, here a simple illustration:



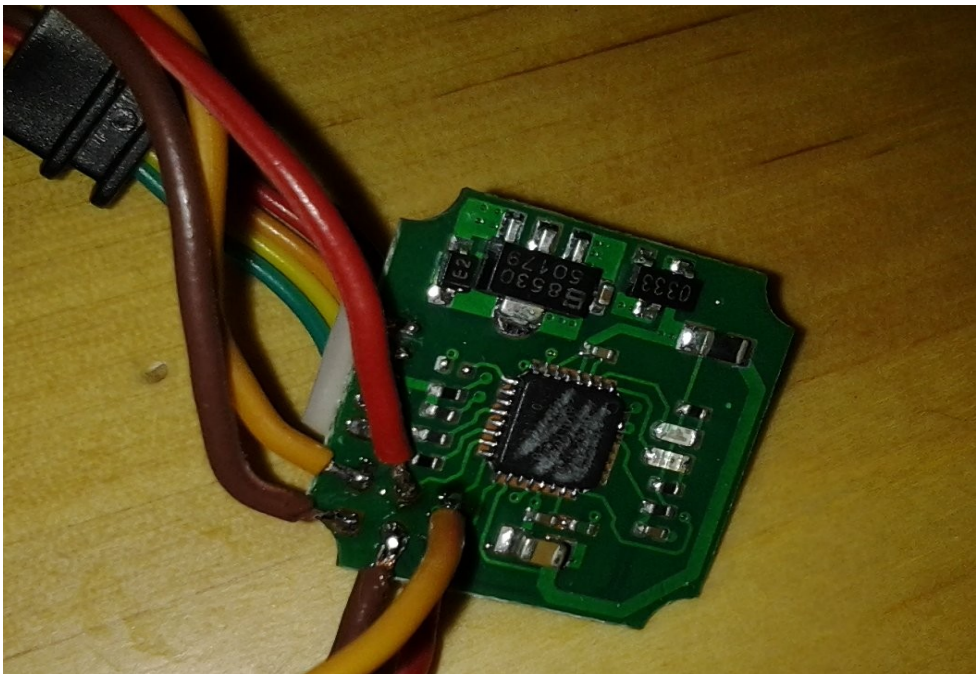
Points the arrow into the other direction, the diodes should be installed the other way around.

3. Flashing the Bootloader

So lets get our hands dirty and flash the gyro! That's the part, which requires some soldering. Even when you bought both, the AVR Programmer **and** the FT232RL serial adapter, soldering here can't be avoided. So we need to open the gyro from the back, by unscrewing the 4 tiny screws. After that, both parts of the casing can be removed. We are looking at the naked GA250 board. On the board are 6 soldering points, we need to connect our programmer to. When you are using the USBASP, you can use the diagram below. Otherwise you just have to check for the pinout of your programmers plug, to make the right connections.



The soldered and ready for flashing gyro looked like this when I did it:



Soldering the wires on the tiny pads is not an easy task. Be careful when you are doing it. Use small wires, I used servo wire. Once you are done with the soldering, the hardware part is completed.
So lets move on to the “Software Part”. We want to flash the Gyro with the Bootloader. Yes, Bootloader! The OlliW Firmware will go into the gyro with a different process. We want to get rid of the extra wires as soon as possible and we don't want to solder them back, when the firmware is updated.

The Bootloader will do the “real” flashing job for us in the future. So the gyro doesn't even has to leave the helicopter in the future, for getting a firmware update!

But back to the bootloader. To flash it I like to use “avrdude”. Avrdude is a commandline program and its available for all operating Systems (Windows, Linux & MacOS). So lets get prepared and download all the stuff we need on our computers.

The latest verion (5.11) of avrdude could be downloaded here:
http://helix.air.net.au/avrdude-5.11-Patch7610-win32.zip

And of course the other important part is the OlliW Firmware:
http://www.olliw.eu/uploads/olliw-ga250-cp-gyro-v014-v014b-v015-2011-12-30.zip

This is the version of early January 2012. You might also want to check the (german) project page as well and look for the latest version (http://www.olliw.eu/2011/ga250-gyro-firmware/).

> Whats in the firmware zip archive?

Well here is a simple list of the files in the archive from Olliws website. Basically its all in there what you will need:

GA250-Cp-Gyro-v014.dev	
AvrConfig.exe	**Tool to change gyro settings.**
AvrConfig.ini	
AvrConfig.tls	
AVRootloader.dev	
AVRootloader.exe	**The gyro FW flashing tool.**
AVRootloader.ini	
BL_4GA250CpGyro_BlackServoPlug.hex	**Bootloader using the servo plug.**
BL_4GA250CpGyro_RedGainPlug.hex	Bootloader using the gain plug.
BL_4RobbeProgger_m88.hex	BL for the (optional) ProgBox.
BL_4RobbeProgger_m88pa.hex	BL for the (optional) ProgBox.
FT_PROG.exe	Tool for the FT232RL Adapter
FTD2XX_NET.dll	Corresponding DLL
Ga250_Cp_Gyro_Firmware_v014.hex	**Firmware for the Gyro**
GA250_ProgBox_v014b_m88.hex	Firmware for the (opt.) ProgBox

> Flash the Bootloader wit Avrdude

For the next step, we need the red marked file "**BL_4GA250CpGyro_BlackServoPlug.hex**" and avrdude. This is the bootloader. There are two versions of the bootloader, one is using the servo plug in the future for configuration of the gyro and the other one the red gain plug. This is just a matter of taste. I found the servo plug a lot more convenient, especially on a small 250 sized helicopter, its quite difficult to pull out a plug from the receiver. The servo plug can be somewhere outside of the heli and that makes it easy. So in this tutorial we will use the version for the **servo plug**.

When you plug in your USBASP, the LED on the gyro will light up. **The gyro gets its power now from the AVR programmer**. Nothing should be connected to one of the three gyro plugs. You just need the gyro connected through the 6 wires from your programmer.

For the next step you need to have your avrdude ready and the firmware archive unzipped. Avrdude should be somewhere in the Windows path, or you just drop it into the firmware folder. Then open "cmd" (START->Run: cmd), CD into the firmware folder and execute the following commands.

1) connect the programmer with the attached gyro now and check the connection with avrdude by excecuting the following command

```
avrdude -c usbasp  -p m8 -U lfuse:r:-:i
```

Here is a sample output of avrdude, with a working connection:

```
Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e9307
avrdude: reading lfuse memory:
Reading | ################################################## | 100% 0.00s
avrdude: writing output file "<stdout>"
:01000000BF40
:00000001FF
avrdude: safemode: Fuses OK
avrdude done.  Thank you.
```

2) then excecute the following commands

```
avrdude -c usbasp  -p m8 -e
avrdude -c usbasp  -p m8 -U lfuse:w:0xBF:m
avrdude -c usbasp  -p m8 -U hfuse:w:0xDC:m
avrdude -c usbasp  -p m8 -U flash:w:BL_4GA250CpGyro_BlackServoPlug.hex
```

Output from avrdude of the last command, which finally flashes the bootloader:

```
Reading | ################################################## | 100% 0.01s
avrdude: Device signature = 0x1e9307
avrdude:  NOTE:  FLASH  memory  has  been  specified,  an  erase  cycle  will  be
performed
         To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude:  warning:  cannot  set  sck  period.  please  check  for  usbasp  firmware
update.
avrdude: reading input file "BL_4GA250CpGyro_BlackServoPlug.hex"
avrdude:  input  file  BL_4GA250CpGyro_BlackServoPlug.hex  auto  detected  as  Intel
Hex
avrdude: writing flash (8180 bytes):
Writing | ################################################## | 100% 4.28s
avrdude: 8180 bytes of flash written
avrdude: verifying flash memory against BL_4GA250CpGyro_BlackServoPlug.hex:
avrdude:      load      data      flash      data      from      input      file
BL_4GA250CpGyro_BlackServoPlug.hex:
avrdude:  input  file  BL_4GA250CpGyro_BlackServoPlug.hex  auto  detected  as  Intel
Hex
avrdude: input file BL_4GA250CpGyro_BlackServoPlug.hex contains 8180 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 2.21s
avrdude: verifying ...
avrdude: 8180 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done.  Thank you.
```

The first command in 2) erases the old gyro firmware. **From here, there is no way back.** We can't flash back the old firmware. Be aware of that. The 2nd and 3rd step sets the fuses in the microcontroller and the 4th command finally flashes the bootloader into the gyro.

When you are **not** using the "usbasp" as a programmer, you need to change the commandline parameter accordingly to your programmer.

After flashing, you can disconnect your programmer from the gyro. You can keep the

cables soldered for the moment, just to be safe. But nothing should be connected to them for the next step.

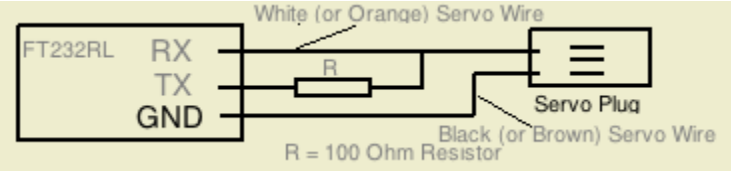
My suggestion is, to put the gyro now back into his “natural environment” and hook it to a receiver, with ESC and a Lipo battery. Just like the way you would have it later in your heli (connection of both, gain and rudder plug in the receiver is here required). Only the servo plug just should remain empty. When you power on the receiver (connect Lipo to ESC), **the gyro wont' light up**.

4. Flashing the Olliw Firmware

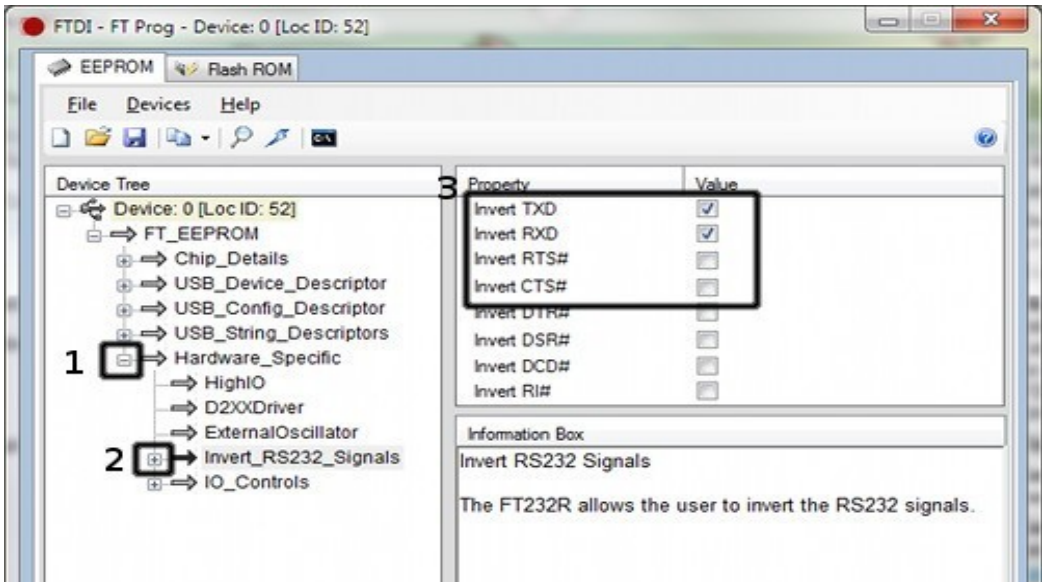
Time for the final act! Let's flash the latest OlliW firmware into the gyro. For that we need either the serial DIY adapter, or the FT232L USB to serial adapter. When you use the FT232RL adapter, one additional step is necessary before we can proceed. **When you are using the DIY adapter, you can skip to the next step.**

> **Preparing the FT232L adapter**

To connect the adapter through the servo plug of the gyro, we need to fix a servo wire on it. You will also need a 100 Ohm resistor for that. Here is a simple wiring diagram:



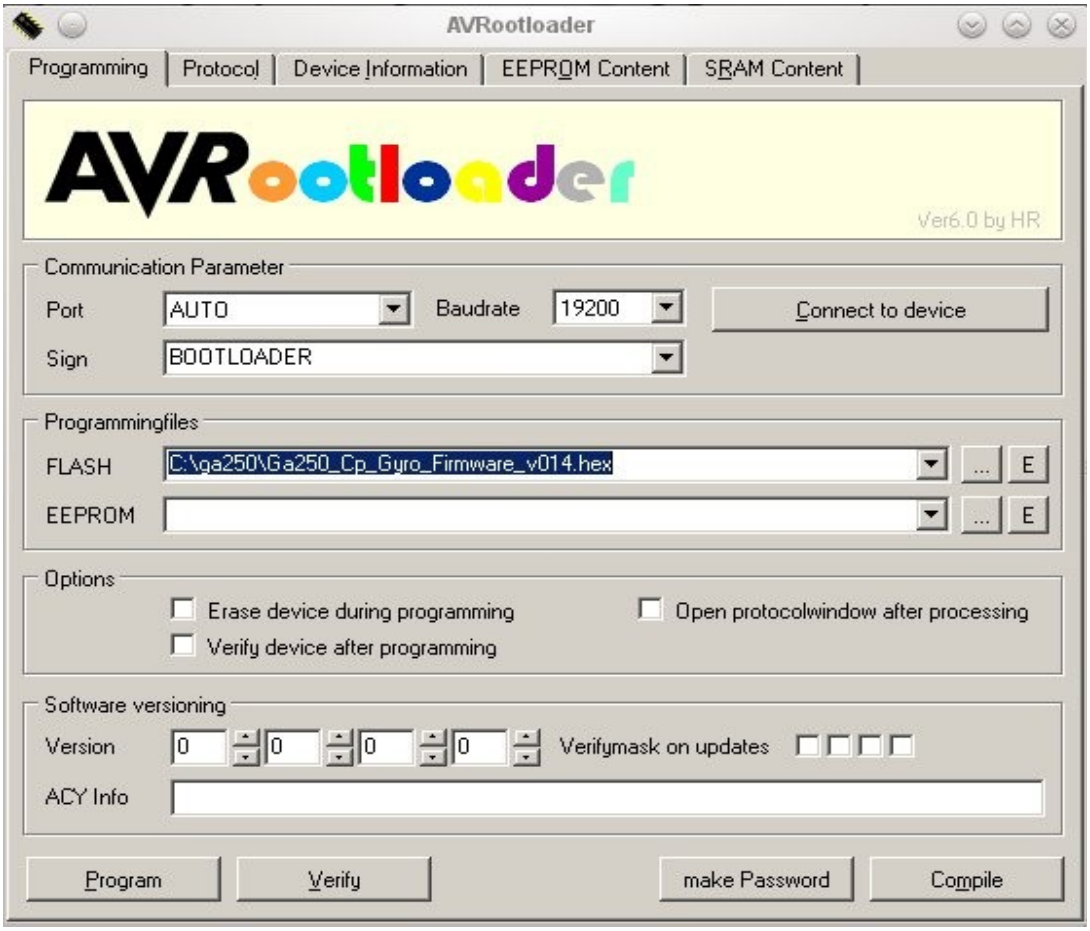
To be able to connect to the gyro, also the RX and TX signals of the FT232RL need to be inverted. For that, you just have to open FTProg.exe in the fimware folder. Then press the scan button to search for your adapter and change the settings as illustrated in the screenshot. When done, press the flash symbol in the toolbar of FTProg, to save the new settings.



Now your FT232RL adapter is ready to be used! If you are using the DIY serial adapter, the mentioned steps are unnecessary and needed to be skipped as mentioned earlier.

> Flashing the Firmware with AVRootloader

Open AVRootloader from the firmware folder, choose the correct firmware file "`Ga250_Cp_Gyro_Firmware_v014.hex`" and correct COM port. To keep it at AUTO, sometimes doesn't work. Plug the servo wire from your serial adapter (DIY one or the



FT232) into the servo plug and press "Connect to device". Then connect the Lipo to your ESC and AVRootloader should show "connected".

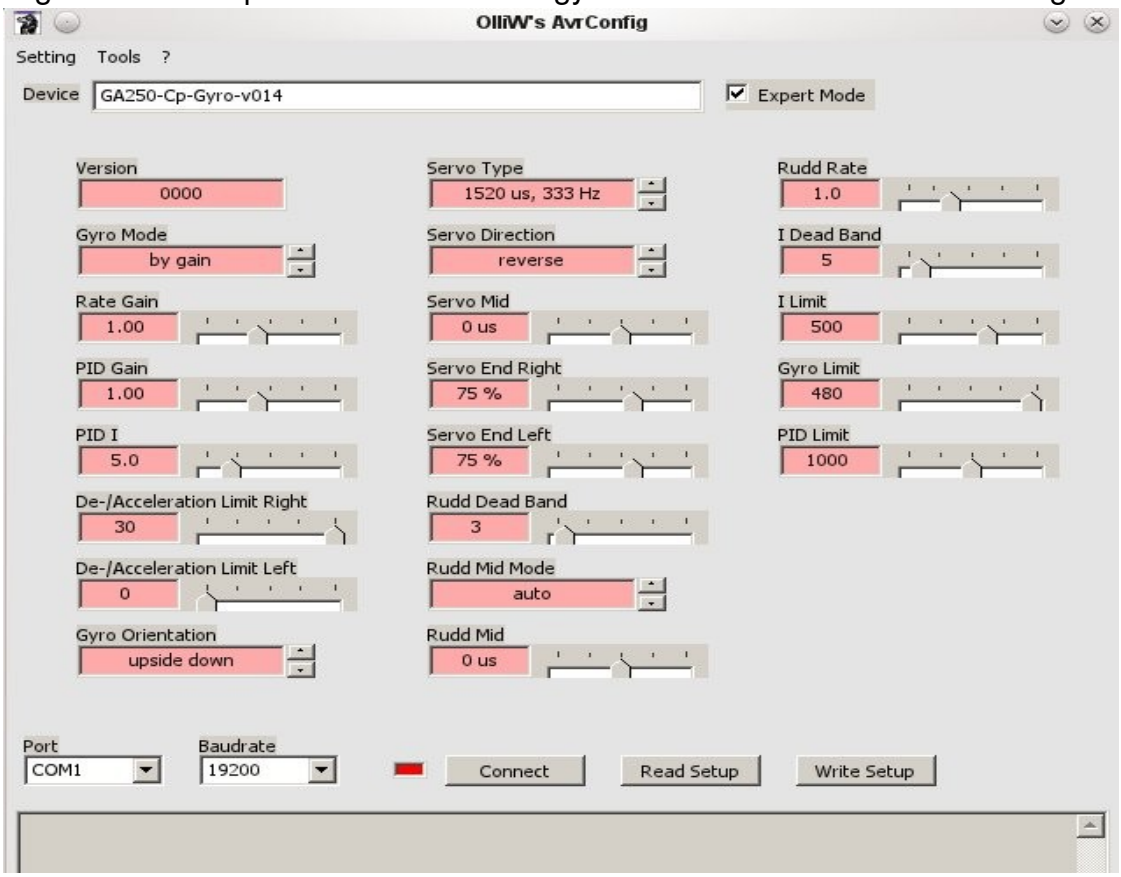
When you switch to the "Protocol" tab, you can see that AVRootloader is sending "Keepalives" to the gyro. That ensures, that the connection stays. This tab is also the only good source, to get some information about the flash process. My suggestion is, to press "Program" to flash the gyro and when its done, to quickly click "Disconnect". Then look at the log in the Protocol tab. From there you can get some confirmation about the flash. When you don't click disconnect quickly, it will continue to write plenty of "keepalive" messages into the log and you might have to scroll up a lot :-).

The flash process with AVRootloader is safe, nothing to worry about. Here you can't break anything. We are just communicating with the bootloader and the firmware code is written into a different part of the gyro flash memory. The process can be repeated as many times as you like.

Remove the serial connection cable from the servo plug and unplug/replug the lipo from the ESC. When the connection and the flash worked, and the LED goes from blue to red after power up, you can unsolder the cables from the GA250 board and put the gyro back into its casing.

5. Gyro configuration

To change some vital parameters from our gyro we need to launch AVRConfig from the



firmware folder. The gyro can be already installed in your helicopter. In the screenshot you can see AVRConfig with the default values. Plenty of parameters can be adjusted. For the beginning, you want to check:

- Servo Type
- Servo Direction
- Gyro Orientation
- Servo End Right
- Servo End Left

In this early version of this guide, I will not further elaborate too much on the other gyro settings. But I will give at least one example and also provide Olli's standard procedure, as a way to get to a working setup. In a later release, it can be interesting to share configurations which worked well in different setups.

In AVRConfig you need to manually choose the right COM Port. The procedure is the same like with the AVRootloader. Plug the serial adapter cable into your computer first. Start AVRConfig and connect the servo plug of the adapter with the gyro. Press the connect button on AVRConfig and finally connect your Lipo to the ESC. The red box next to connect will turn green. Press “Read Setup” and you see all parameters changing from Red to Green background. Once you change a parameter, the background will change to Red color again. Press Write Setup to store your settings in the gyro. To verify its always good, to check with Read again.

6. Standard setup procedure

Olli provided some instructions, which are basically a procedure to find the right values for the most important gyro parameters. Here is my German to English translation of that.

1) Gyro in Rate-Mode: Set up your gyro in rate, you start dealing with only one parameter first- the Gain. When your helicopter is not acting like it should in rate mode, chances are low that it does in HH mode. When you face some issues here to stabilize your tail, make sure that you iron out any mechanical issues before you continue.
2) Gyro in HH-Mode and De-Accelartion Limit = 30: The Rudderfilter/limiter is at 30 almost completely off and now you just have to work with two parameters: Gain and PID I. Start with a lowered PID I and fly some pirouettes. Look how the tail stops after the pirouette. Is the tail after a stop kind of pulled back again, your PID I is too low. When you get a bounce after a stop, its too high. Try to adjust your PID I and Gain, until your tail ends a piroutte stop, with one short but strong backbounce. Thats the ideal setting.
3) Gyro in HH-Mode, Gain and PID I when the right parameters are found as in (2), start to lower the De-/Acceleration limit Right step by step. The short but strong backbounce should become lesser and lesser. Find the right value for you. You can elimiate the backbounce completely, but then your tail might feel sluggish. I like to have a slight backbounce, the tail feels more locked in for me.
4) Go flying and adjust your gain until there is no wagging in any kind of flight situation.

7. Example Configuration HK-250

My HK-250 helicopter is flying like a charm with the GA250 and this firmware. Before that, the GA250 was unusable. None of the cheap gyros worked really well for me. But the GA250 now does. The tail just holds perfectly.

But before I am explaining my GA250 configuration, I want to elaborate on the setup of my Helicopter. I am using as a transmitter a Turnigy 9X from Hobby King. The Turnigy 9X seems to be not as accurate as the Spectrum DX Olli is using. We faced in the first tests massive issues due to a drifting rudder signal. The rudder signal from then transmitter turned out to be a bit fuzzy. But we found a solution, which will be of help for everybody else who is facing issues with his TX as well.

<u>My HK-250 (T-Rex 250 clone) Configuration:</u>

- 3x HXT900 on Cyclic

- DS420 as tail servo
- HK-SS 18/20 ESC
- Separate 6V HobbyKing BEC
- 3800 KV motor with 12T pinion, Headspeed ~4100 RPM
- 3S Zippy Flightmax 20C 800mAh Lipo
- Hobbyking FG Mainblades
- Gaui CF Tail Pushrod
- Align alloy tail arm and landing gear

Everything else on this heli is still HobbyKing, just two Align parts.

And here is my GA250 configuration. To help with the drift of my TX, Olli implemented “Rudd Dead Band”. So I was able to set the following parameter:

- Rudd Dead Band: 15
- Rudd Mid: 2 us (it was slightly off center, this corrects it)
- I Dead Band: 5 (probably not necessary and can remain default)

There is a second issue with my TX regarding the gain. Its for example a known problem, that when you use the HK401b Gyro together with the T9X, you have to set the gain in rate mode to +60. Otherwise this Gyro wouldn't switch into HH mode and gain for HH would be ignored. So I was afraid, that my gain displayed in HH is far away from the real figure. That's why I switched to

- Gyro Mode: heading hold (in this mode the gain of the gyro is determined by the parameter PID Gain, the parameters Rate Gain and the Gain setting at your Tx are ignored)

In the future I might revert that back to “by gain” to be able to make adjustments from my TX. But at the moment I am very satisfied how everything works, so I keep it like that.

Now we come to the more interesting part that makes the gyro behave like you want. I just played with two parameters and ended up with these settings

- PID Gain: 0.50
- PID I: 3.0

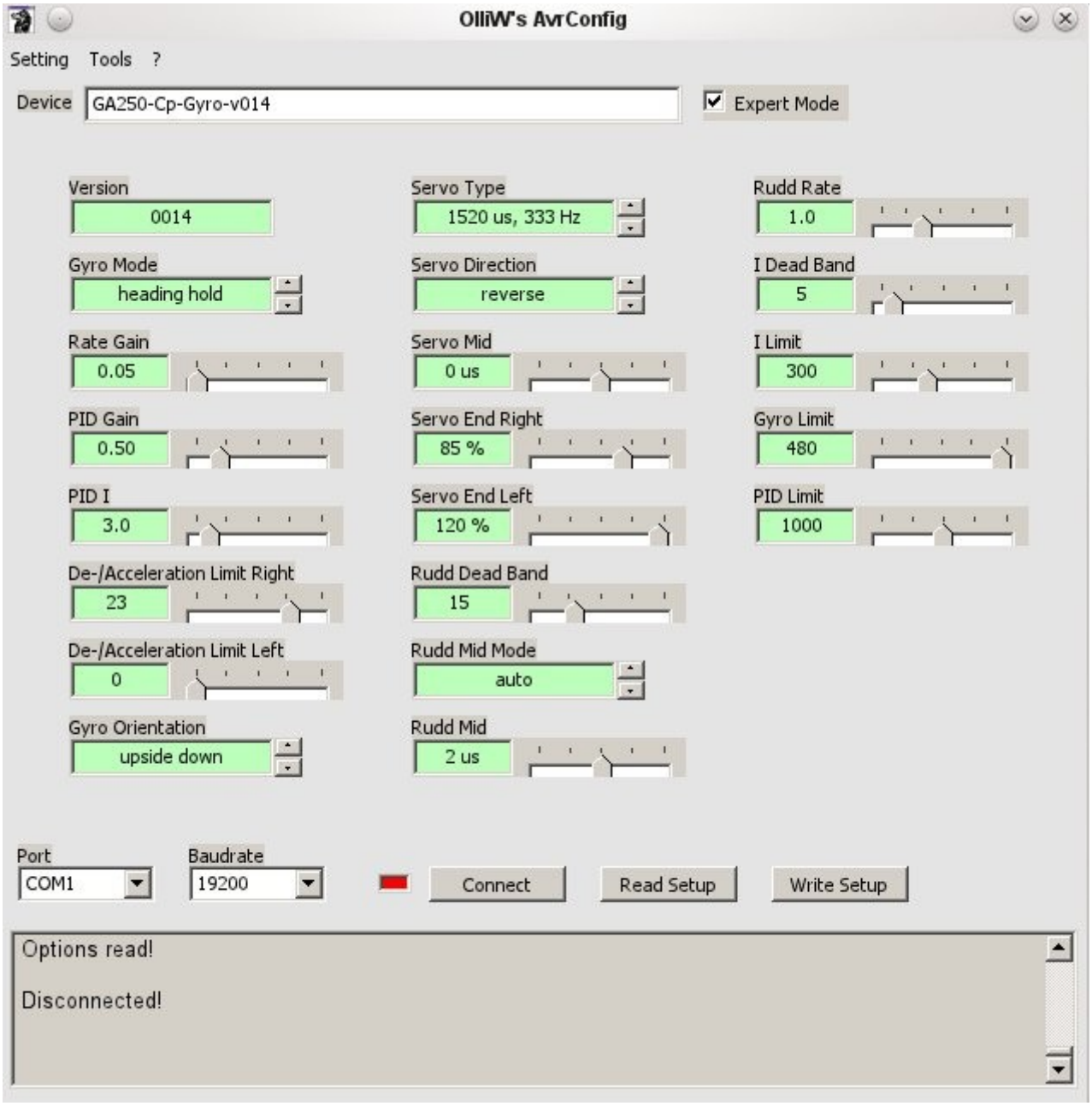
The “PID I” is the essential part which influences the behavior of the gyro a lot. Higher PID I will make the gyro more sensitive and active. The default 5.0 was way too high for a 250 helicopter. With 3.0 the tail doesn't wag. The “PID Gain” represents the Gain you would usually adjust from your TX. In this setup now its fixed to 50%. The adjustment of the PID I was the key for me, to get a working tail. Most of you wouldn't touch the “PID Gain” paramemter, you only need that, when you are NOT controlling your gain through the transmitter like I do.

To avoid bounce, I had to tweak this parameters

- De-/Acceleration Limit Right: 23
- De-/Acceleration Limit Left: 0

Having Left at 0 means, that it uses the same value as right. With something more than 25 I had a bounce. With 23 the taill locks in crisp and no bounce at all.

And that's it. For the rest, just have a look at the screenshot below. I am considering myself in the RC Heli Hobby still a beginner. So I am not an expert, but that's probably the reason, why this guide is hopefully helpful and more people will try the firmware and join the project.



Appendix A: LED Codes

To avoid confusion, here is a little LED code reference.

→ Startup

(No firmware is flashed yet → Bootloader is waiting, no LED activity)

1) Firmware is flashed → Bootloader checks whether AVRootloader, AvrConfig or Progbox is connected. While he is doing that, the blue LED flashes.
2) One of these three is connected, through the whole activity the BLUE led will remain on. After disconnection of ProgBox or PC, the gyro continues to 2)
3) Firmware is waiting for a signal from the Receiver, red LED flashes fast, blue is off
4) Firmware is waiting for a stable rudder signal, red flashes, blue is off
5) Startup is finished, red is off and blue on (or Off) depending on rate or HH mode