Storm32 brushless gimbal manual

External link: https://goo.gl/qejTYK

Description

Lifetime warranty

Electrical wiring diagram with camera

Part 1 - Installing your gimbal

Part 2 - Configuring your gimbal

- Calibrating the sensor
- PID Tuning
- Tuning Procedure
  - The Starting Point
  - The P Parameter
  - The I Parameter
- Configure the RC Inputs
- GUI Settings
- Wiring for PWM Signals
- Wiring for SUM-PPM Signals
- Wiring for Digital Signals
    - Spektrum Satellite
    - S-Bus
    - HoTT SUMD
    - Multiplex SRXL
- Wiring for Joysticks
  - DJI Naza/A2 Pitch Control

Part 3 - Powering your gimbal
- GUI installation
  - How does Read, Write, and Store work?
  - How to enter precise parameter values?
  - Which drivers are needed for the USB?
  - Storing the parameter values to the EEPROM doesn't seem to work
  - Flashing with the USB-TTL adapter produces Windows error messages

Part 4: Hold versus Pan Mode
- What does stabilization mean?
- Hold versus Pan
- Fine Tuning the Pan Mode
- Deadbands
  - Pitch and Roll Deadbands
  - Yaw Deadband
- Switching Between Pan/Hold
- Further Tuning the Pan Mode

Part 5: Advanced Functions
- Second IMU Support
- Lipo Saver
- Voltage Correction
- Beeps
- Standby
- Recenter Camera
- IR Camera Remote Control
- Pwm Out
- Scripts

- Motion Control
- MAVLink
- APM / Pixhawk 2 gimbal control
- NT Logger
- Third IMU

Profile of your gimbal

Storm32 original user manual
- Optional video transmitter manual

Thank you for your purchase in our shop. We hope you will find full satisfaction in our product.

Description

The gimbal secures small and middle cameras

It is made of ABS and carbon material for the strongest possible build.

The gimbal has been specially designed and has been thoroughly tested.

This gimbal can be attached to different frames. Accessories are available to attach to specific frames.
Ask us if you want a particular accessory.

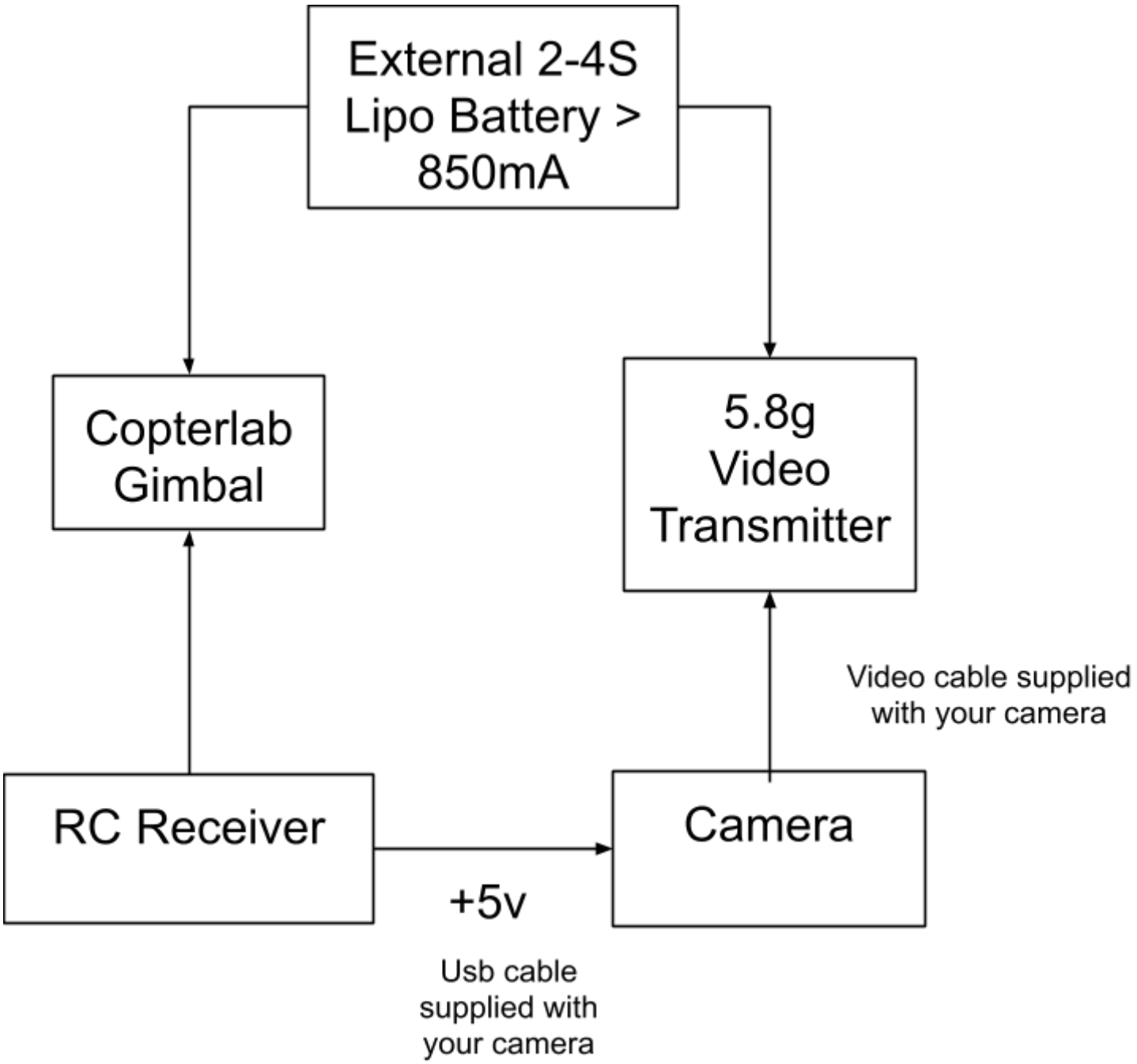
If you ordered a controller card, it will be configured automatically.
Settings are suitable for your camera and voltage lipo battery chosen with your order. It may be necessary to change to other models of camera.

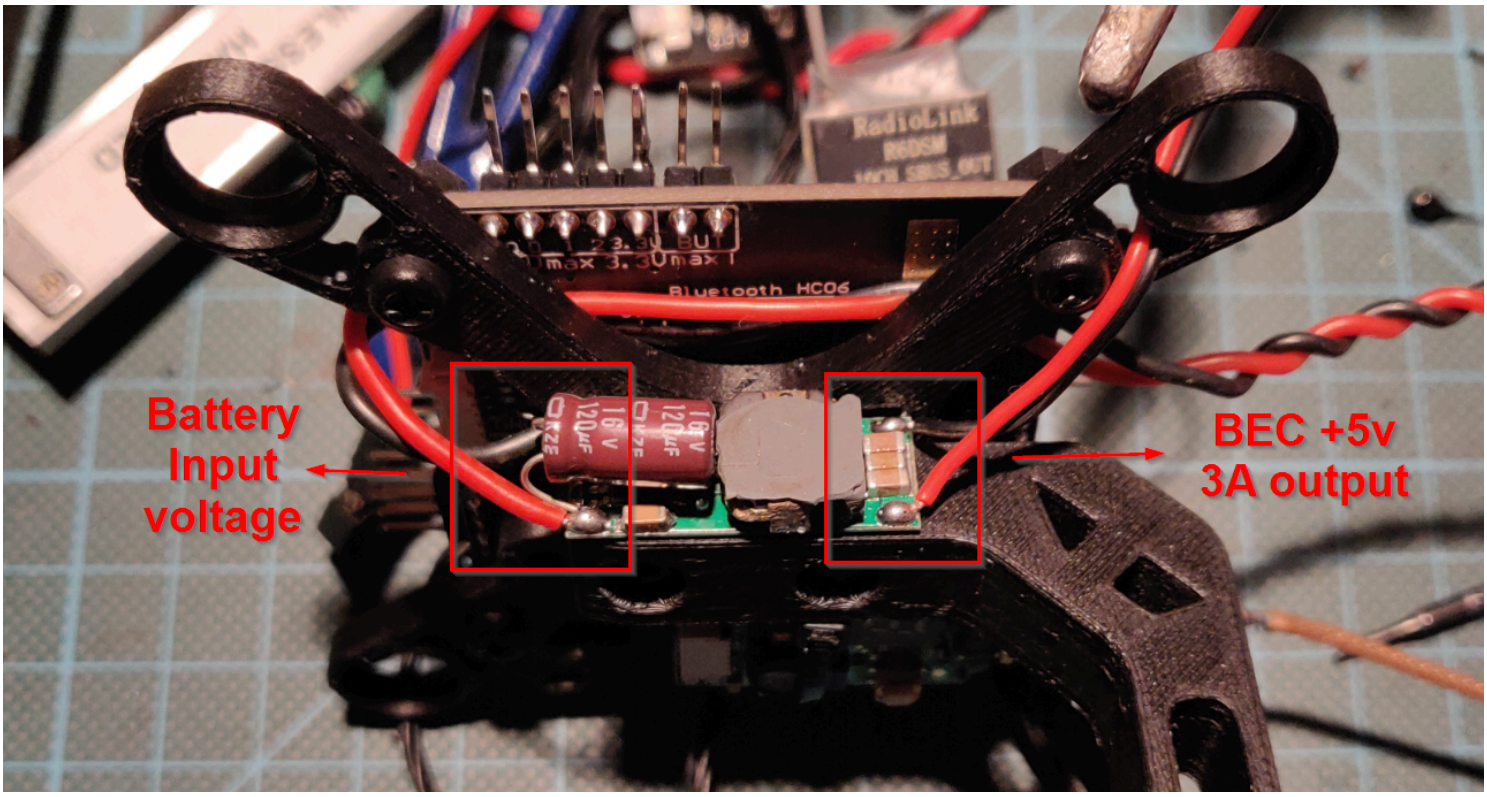
Lifetime warranty

If you break your gimbal, we can send you gimbal parts for free. Just send $15 for worldwide shipping costs.
We would like you to send us photos of any broken parts.
Electronic / motor parts are not under warranty but can be replaced.

For nano gimbals and some 2 axis models:

Electrical wiring diagram with camera


External 2-4S Lipo Battery > 850mA
Copterlab Gimbal
5.8g Video Transmitter
Video cable supplied with your camera
RC Receiver
Camera
+5v
Usb cable supplied with your camera



Part 1 - Installing your gimbal

Installing your camera into the housing.
- Attach your gimbal over your frame, taking care to isolate the connectors from the rest of the frame.
- Check connections of all modules of the platform
- Connect the control board on a lipo battery type (2S to 6S)
- Keep the gimbal to initialize itself on a stable surface, please don't climb/move it while this operation.

Initialization sequence is a bit long and very sensitive.

Initialization Steps: strtMOTOR - SETTLE - CALIBRATE - LEVEL - AUTODIR - RELEVEL - NORMAL

- strtMOTOR: The motors are turned on, and moved into the position specified by the startup motor position parameters.
- SETTLE: In order to calibrate, the gimbal has to be stable, it should not be moved. For a copter that is quite simple as the copter is on the ground. A handheld device needs to be put at a table to remain in position for a few seconds.
- CALIBRATE: Calibrates the sensors. Takes a second or so.
- LEVEL: This is the point where the motors will moved such to level the camera in pitch and roll axis. The yaw motor is under power, so you can feel some resistance, but it is not moved.
- AUTODIR: Here the motors are moved slightly around and from the sensor changes the controller determines the motor directions.
- RELEVEL: Since autodir moved the camera, it is quickly brought back in a horizontal position.
- NORMAL: This is the final stage; the PID controller is activated and the gimbal is operational. Whatever you do, the camera should remain now stable.

If your gimbal doesn't have enough strength to level, you can help it to level correctly. Motors will beep when gimbal is ready.

Led Signals: The progress of the initialization is also indicated by the green LED on the board: During initialization it flashes with varying frequency. When initialization has finished and NORMAL state has been reached, it goes solid. If

beeps are activated, then the motors will emit a sound at the end of initialization, which can be very convenient. The current state of the controller can also be seen in the status line, the GUI, or the Data Display. The red LED blinks with a frequency of 1 Hz, except in certain fault conditions or special states, in which case it blinks very fast.

Operational Range: With a 2nd IMU enabled you can pitch and yaw the camera indefinitely. Without 2nd IMU, the pitch range is limited to +-45°. In both cases the roll angle is maximal +-80° or so. Larger roll angles are not possible since then the pitch and yaw axes become aligned, leading to gimbal lock.

Quick Troubleshooting:

- The gimbal controller levels the camera, but only very, very slowly: The gimbal has been moved before the initialization has finished. Wait until the NORMAL state has been reached (green LED = solid) before moving the gimbal.
- The gimbal moves constantly around, never finding the level position: The pitch and roll motors are not connected to Mot0 (Pitch) and Mot1 (Roll) but reverse. Hence the control logic does measure that pitch has to be changed by -5°, applies the proper movement sequence to the motor but all that happens is that suddenly the roll value is off by -5°. Make sure the motors are connected correctly.
- Camera turns upside down or things like that: The IMU orientation is wrong.
- Gimbal starts shaking, makes high frequency noises and things like that: This would be normal as we have not yet tuned the PID values for the motor control loop. In case that happens, set all motors the P, I and D values to very low numbers but not zero (P=0.10; I=5.0; D=0.0050) and write them to the board. This will cause the gimbal to be slow when correcting movements but at least you can prove all is functional.
- The motors receive power only briefly after startup but are when shut off and the red and green led start blinking fast, and the controller remains in LEVEL state: The controller could not level the camera within a certain time and hence shut off the motors for safety. This can happen for various reasons, such as that there is a mechanical constraint to the camera, one or more motors do not operate properly because of e.g. a broken motor wire or bad connections, or that the gimbal has not been assembled fully.

Part 2 - Configuring your gimbal

Calibration is already made by Copterlab team. Your gimbal can be used out of box.
We don't advise you to setup it by yourself as this is a complex task.

Dependings manufacturing your storm32 gimbal version could be v0.90 or v0.96 (depending stock)

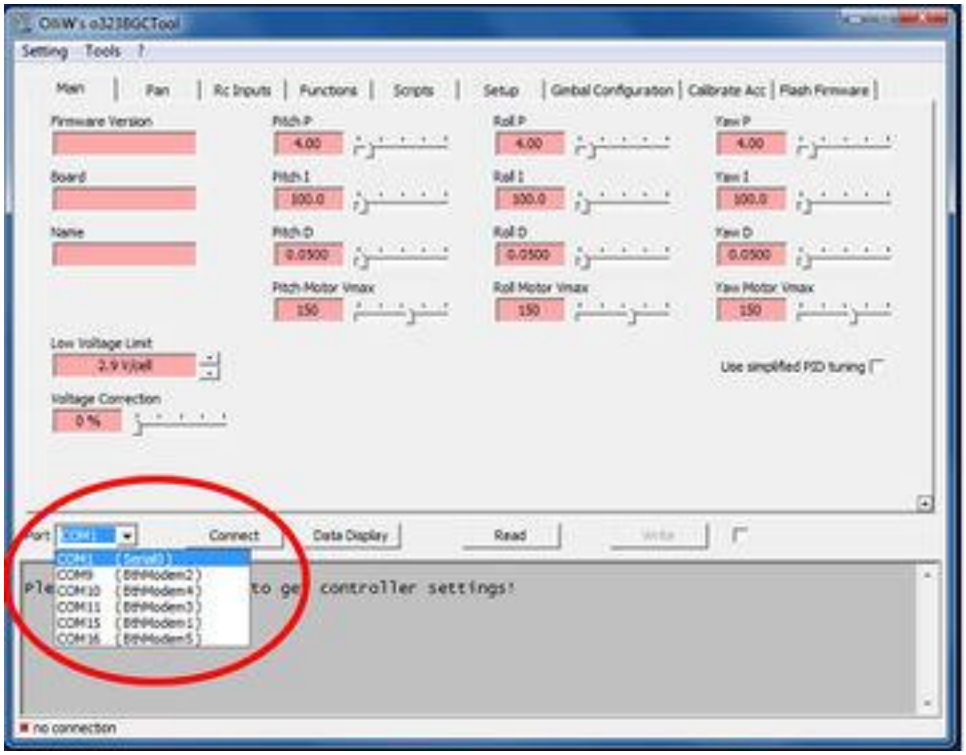
Download and install OlliW application on
V0.90 :
http://www.olliw.eu/downloads/storm32/o323bgc/o323bgc-release-v090-v20160110.zip
V0.96 :
http://www.olliw.eu/downloads/storm32/o323bgc/o323bgc-release-v096-v20160319.zip

USB drivers if needed : -http://www.ftdichip.com/Drivers/CDM/CDM%20v2.12.00%20WHQL%20Certified.exe

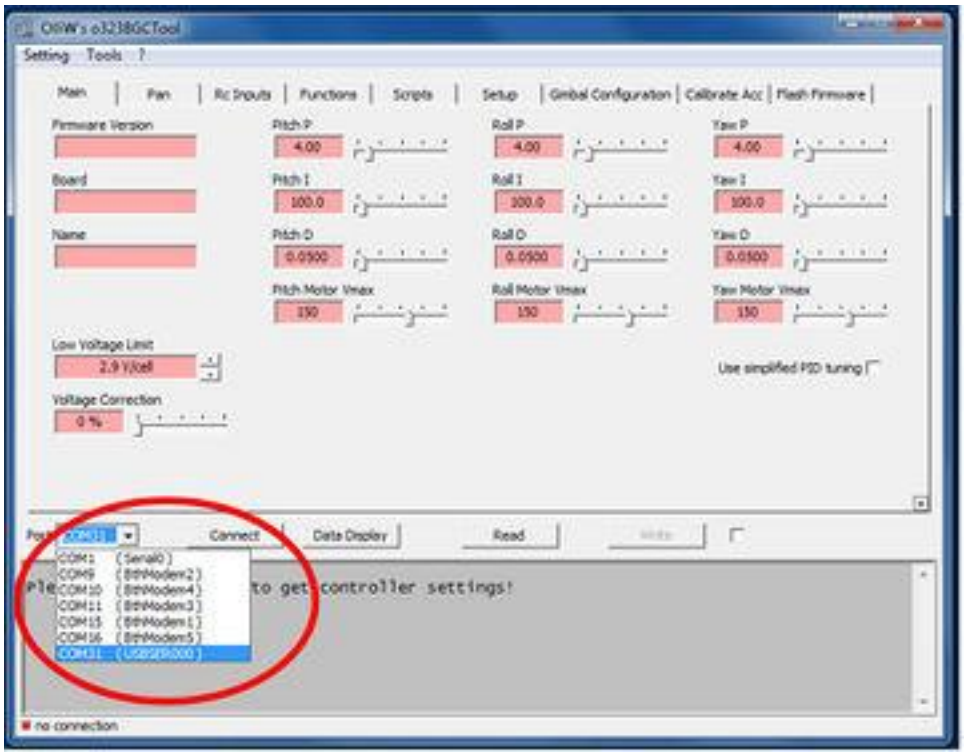
Generally Windows will install the driver for your card if Arduino units have been already installed, you can download drivers here: https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

Launch OlliW application

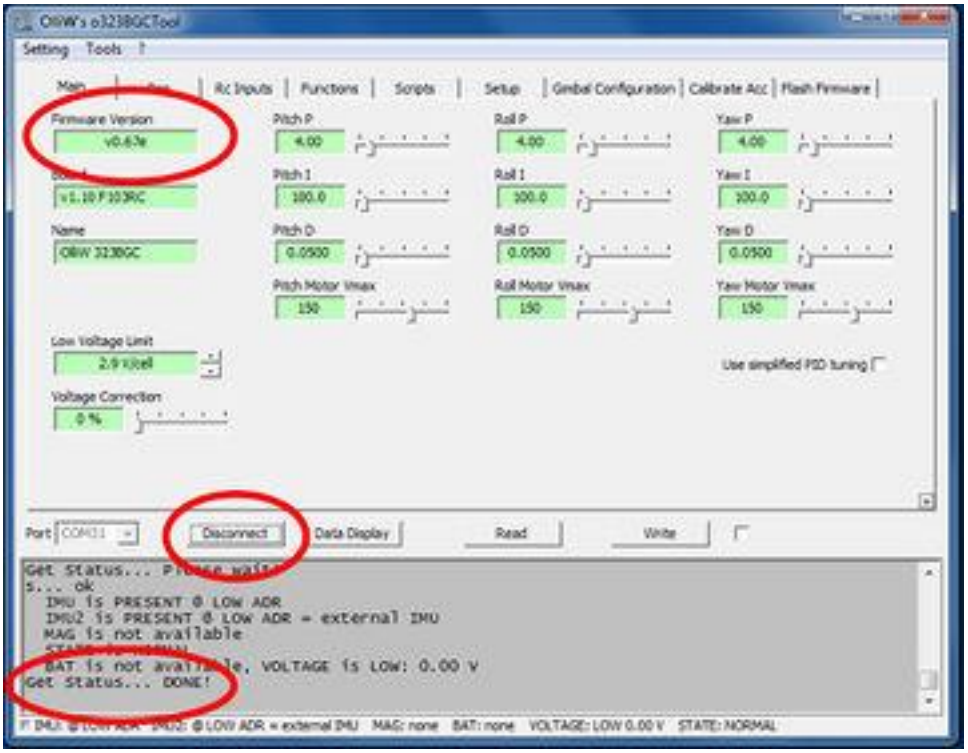
1. If connected, disconnect a USB cable and/or USB-TTL adapter. Click on the COM port selector in the left bottom and memorize the list.



2. Then plug in the STorM32 board via the USB cable to power it. When clicking on the drop down for the COM port selector, it should now show an additional one.
3. If the board has a firmware installed and this is the first-time connection via USB to your PC, then Windows will install the USB driver for the STorM32 board. This can take quite some time. Do ***NOT*** interrupt the install process.



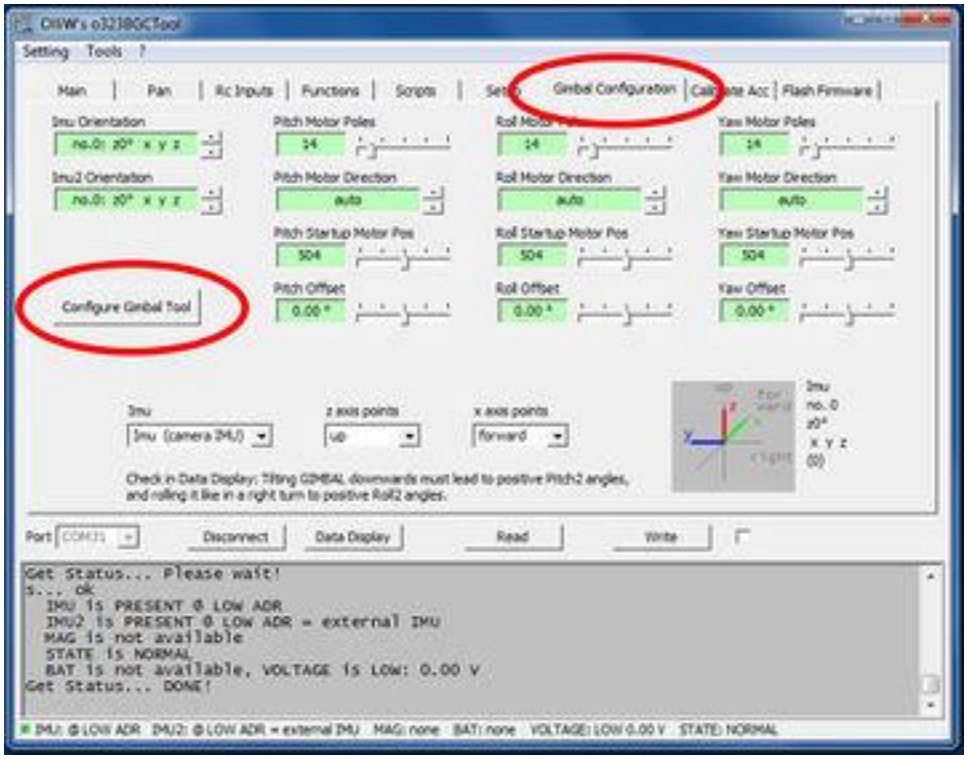
4. Go to e.g. the [GUI:Main] tab and click on [Connect] or [Read] to validate the connection with the board.

5. 

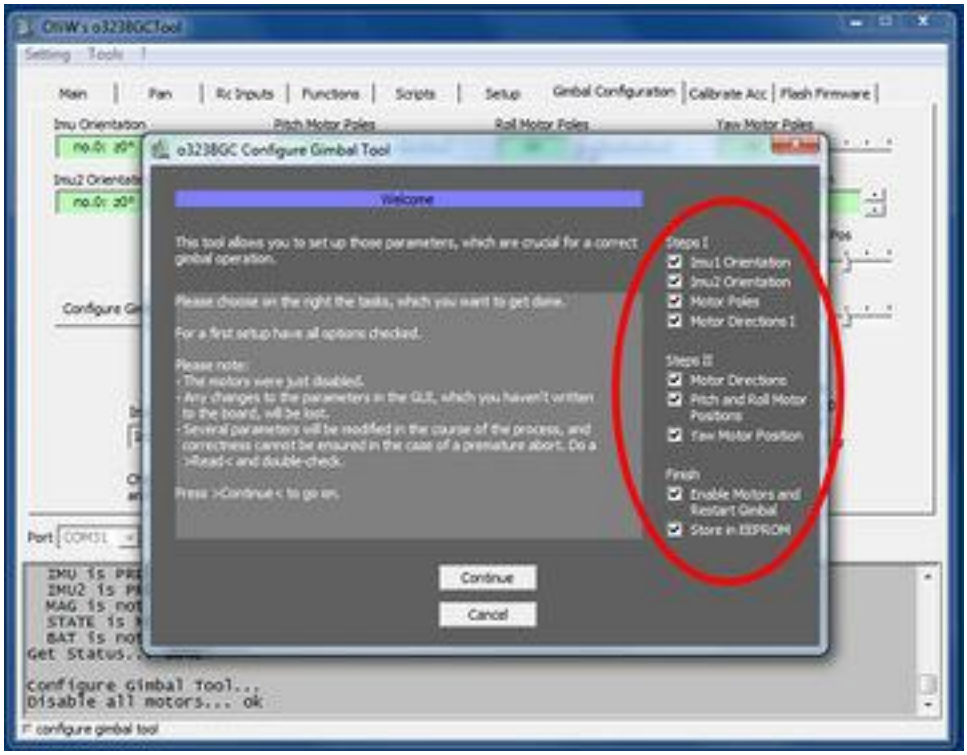
Calibrating the sensor

As already displayed, you don't need to do these tasks. Ask to Copterlab team, your gimbal profile if you need it.

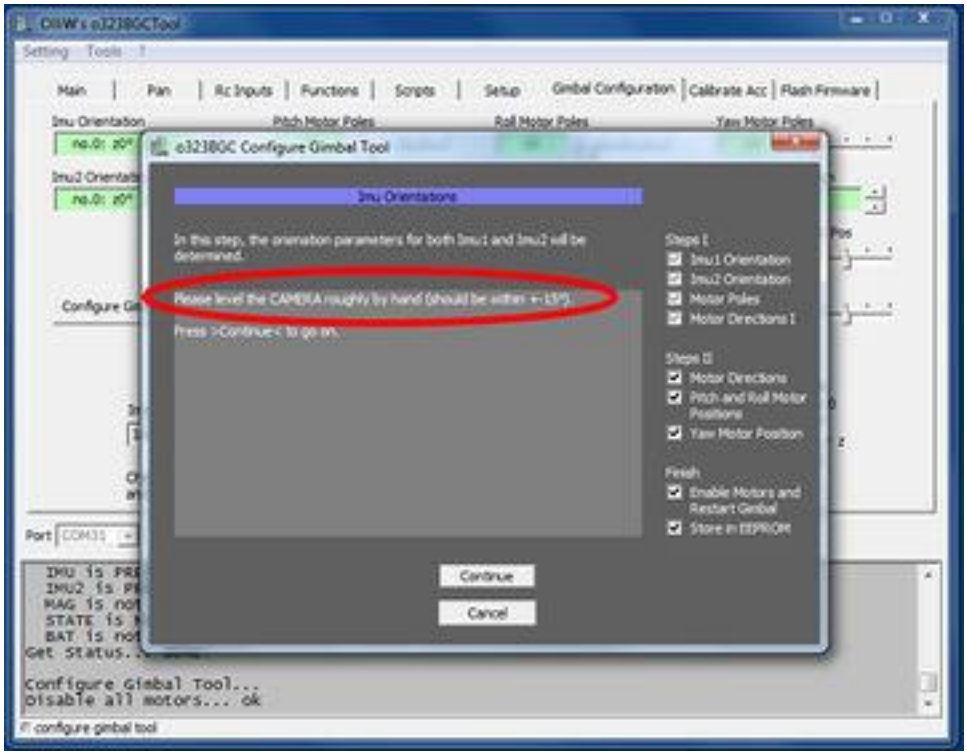
Change to the **Gimbal configuration** tab. There you find the parameter fields for the IMU orientations, motor poles, motor directions and motor startup positions. To set them, run the **Configure Gimbal Tool** by hitting the so named button.

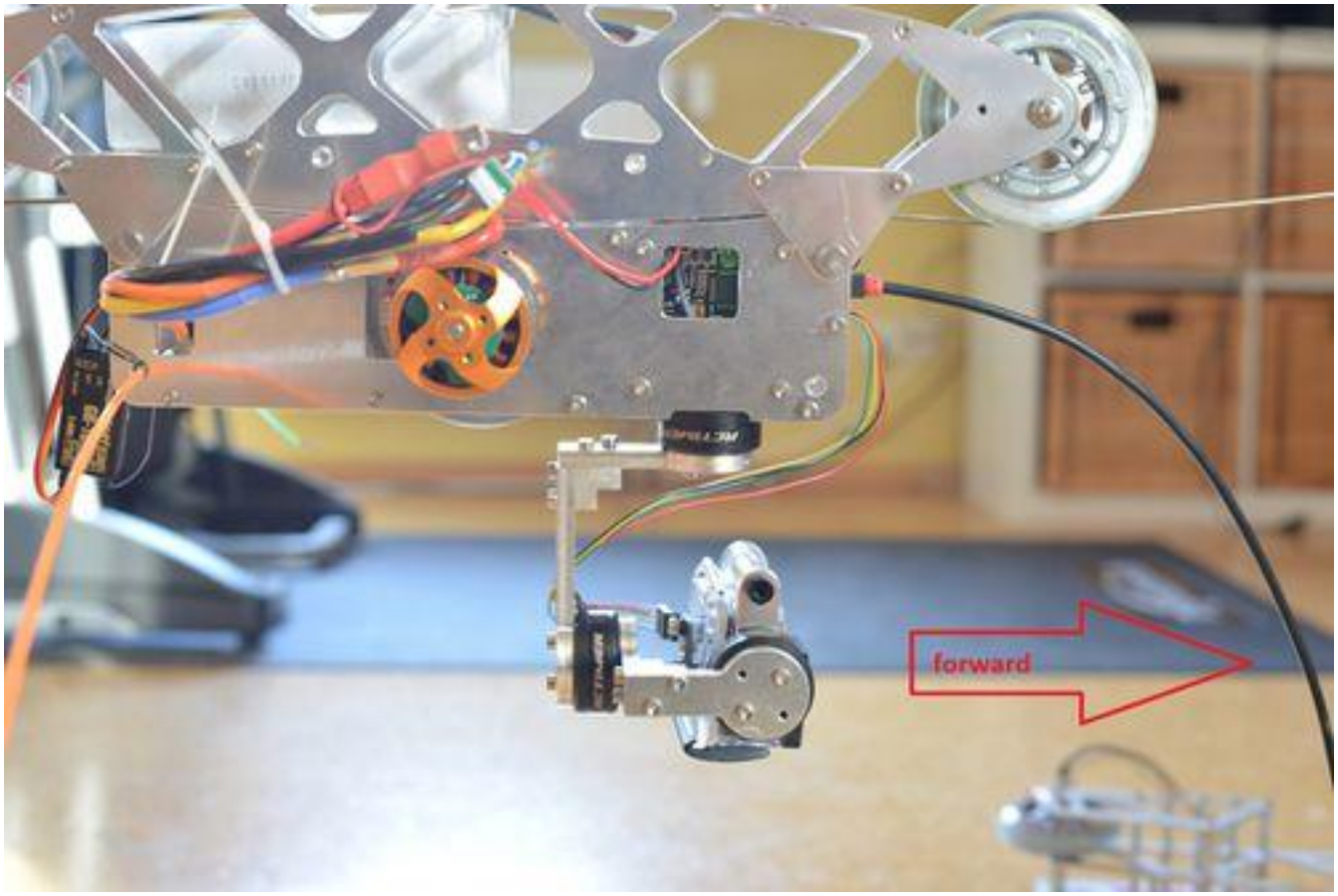


In the welcome screen you can select what to configure. The individual steps are grouped into **Steps I** and **Steps II**, and some finishing steps. You want to do all, and hence just click on **Continue** .

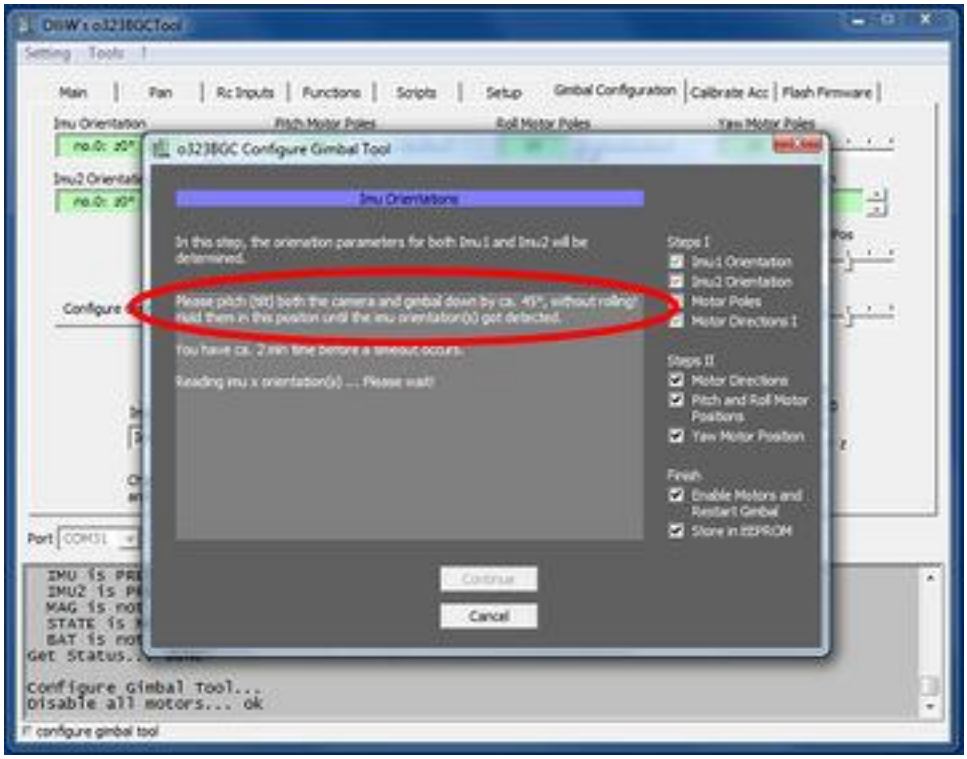


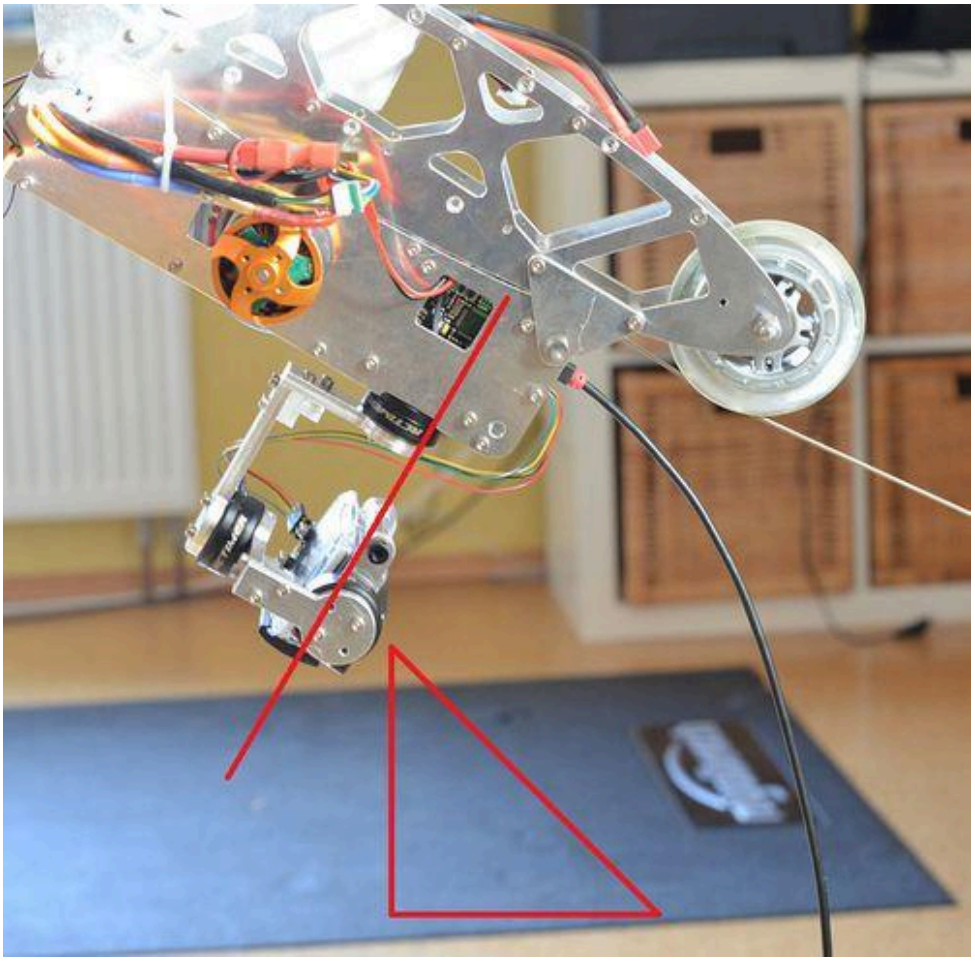
You are now asked to position the gimbal and camera. The gimbal should be in its standard default position. Adjust the camera manually such that it is level and points to the forward direction of your copter. You might not be able to do that perfectly because of the motor magnets, but you should get that better than to within 15°. Once the camera is in forward position click on **Continue**.



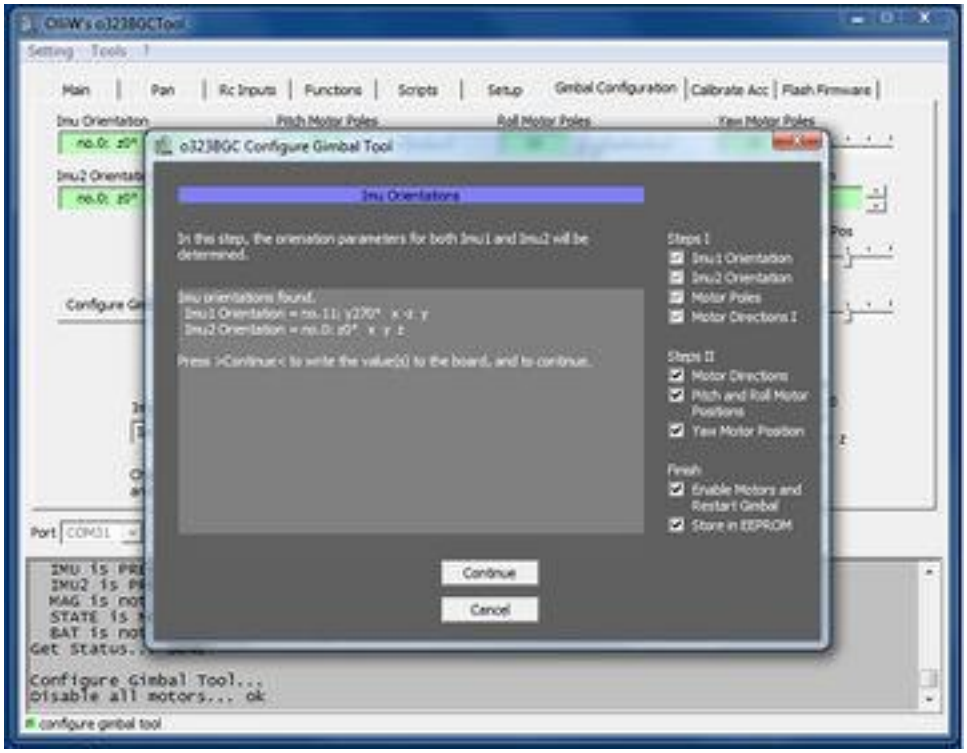


Now you are asked to pitch the entire frame inclusive camera downwards by 45°, as if you would want to film the ground in front of the copter. So you lift the frame such that both, frame and camera, point downwards. Importantly, the camera should not move and keep its position relative to the frame. If it does move then you should consider balancing the camera better. Also, avoid roll movements. Anyway, the goal in this step is to measure the effect of a downturn by 45° degree on the IMU's signals.

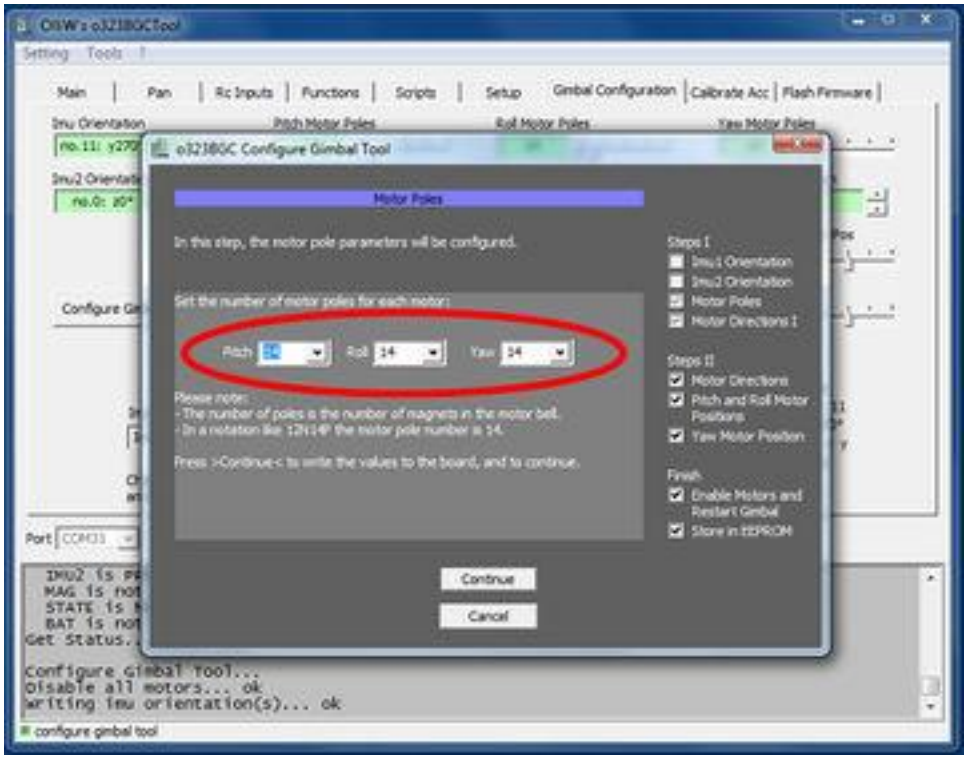




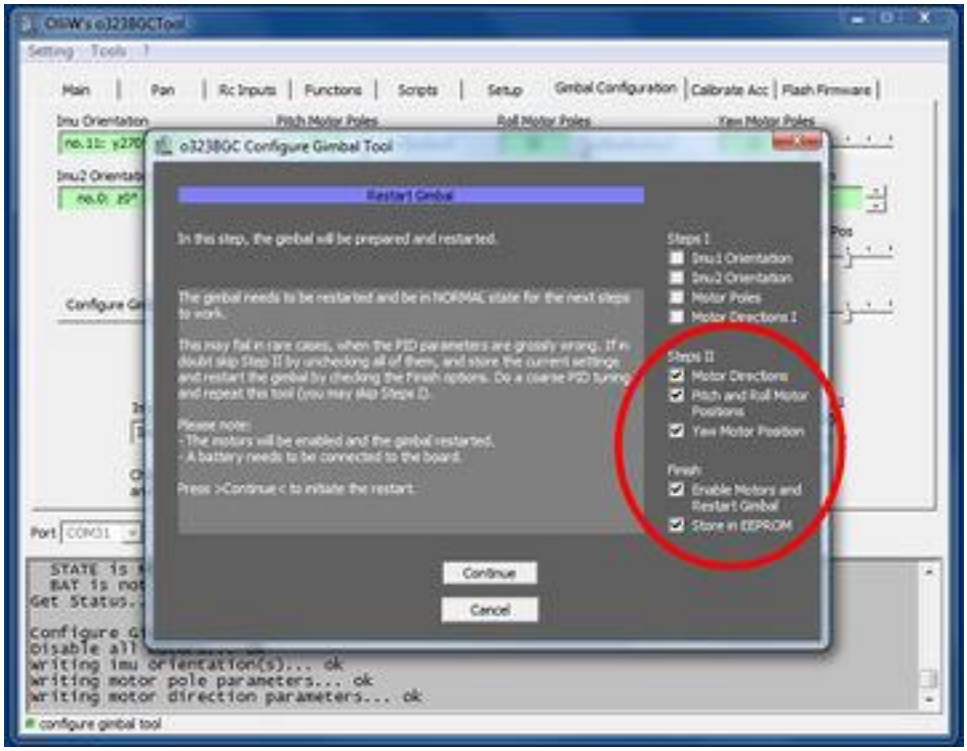
Once the IMU orientations were determined, their values are shown and you can (should) put the gimbal back to normal position. When you can continue with the next step.



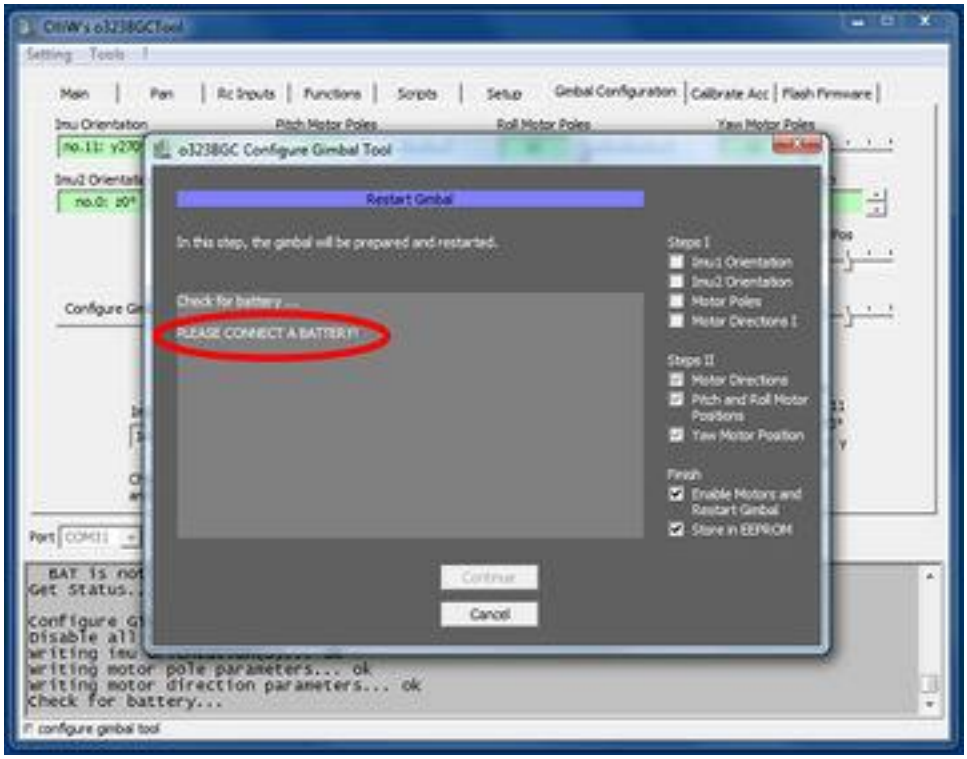
The next important step is to set the motor pole counts for each motor. This is information you should get from the motor vendor. Data like N12P14 means 14 poles.



On the next screen you are informed that all motor direction values will be set to “auto”. Click **Continue**. This completes the settings of **Steps I**, and you can proceed with adjusting the parameters of **Steps II**.

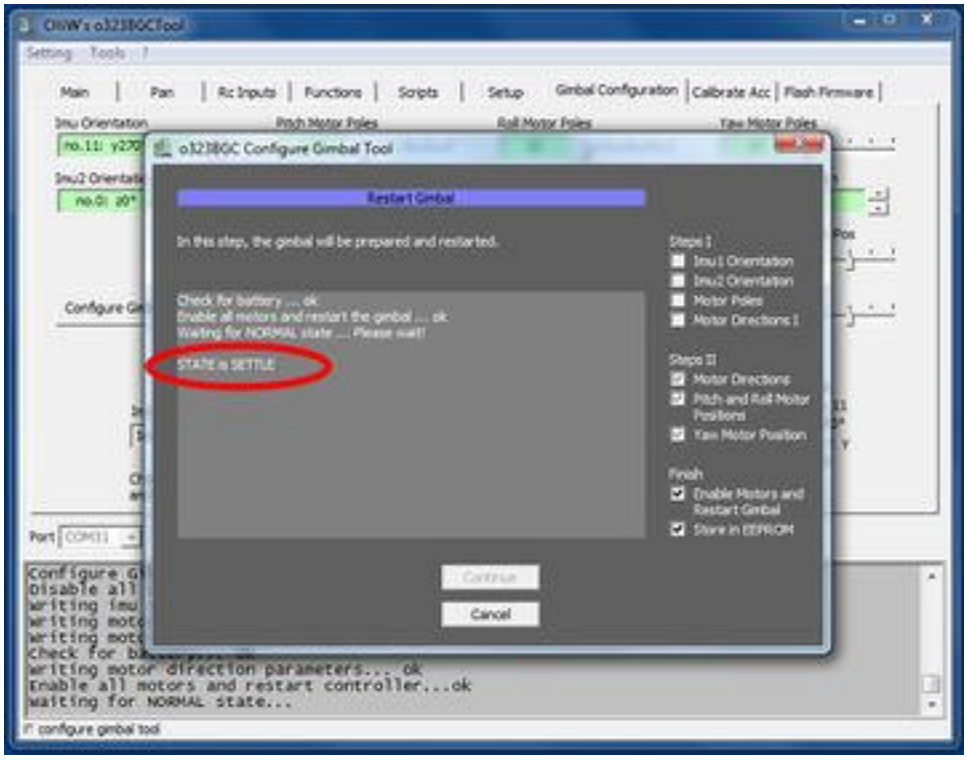


For the next steps to work, the gimbal has to be started up with enabled motors, and a battery must be connected. The GUI will check for that, and will ask you to connect a battery if required. Do as advised.



The gimbal will now go through it's initialization steps, which you can follow in the screen. Please wait until it reaches the NORMAL state. You should see the green LED go solid, as well as hear a beep. You are reminded to keep the gimbal in normal position and at rest during all this.

If this step doesn't complete or an error occurs see the Quick Trouble Shooting section below, and please take any recommendations from there seriously.

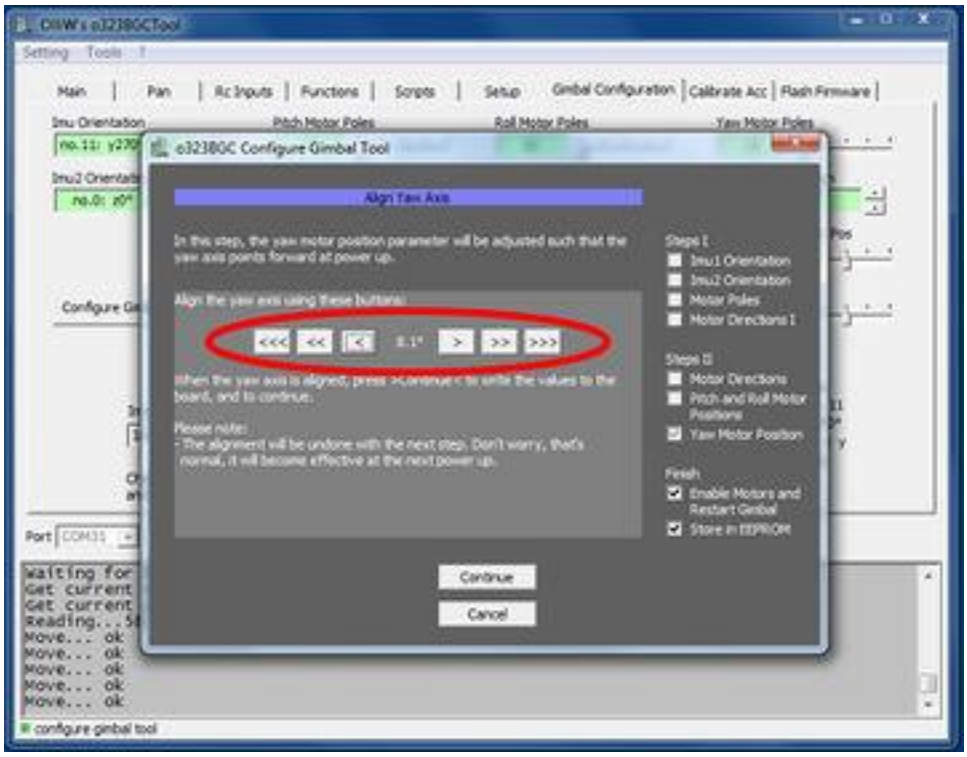


On the next screen you are informed that the motor direction values will be determined for all motors. Click **Continue**.

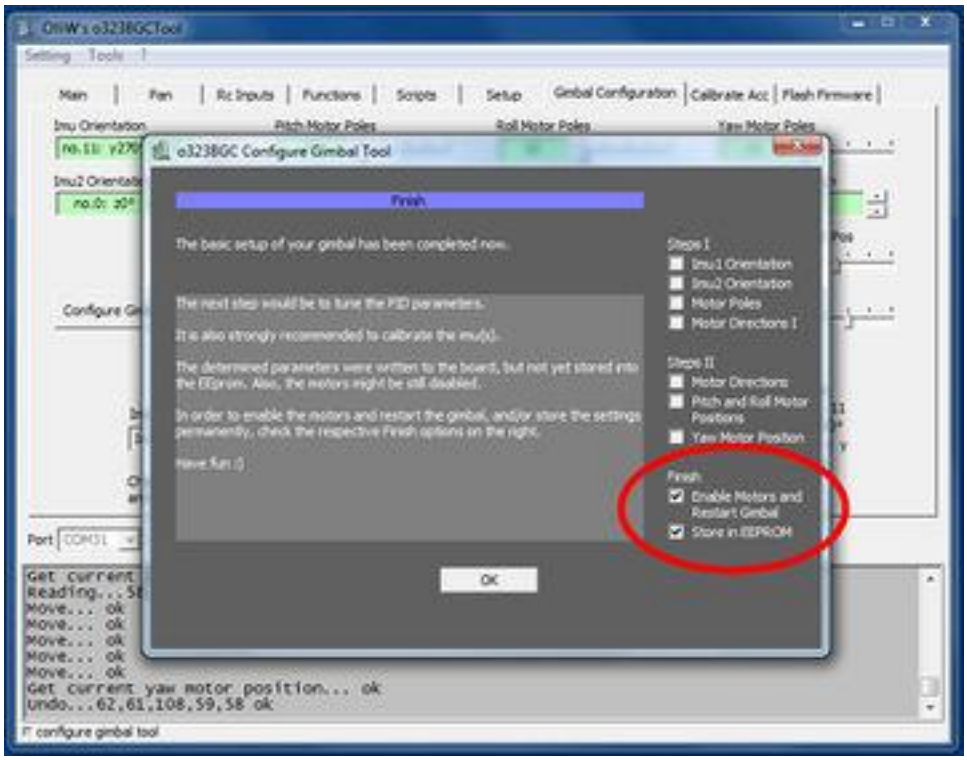
In a further screen you are informed that the motor startup positions for the pitch and roll motors are determined. Click **Continue**.

Now you are asked for another important step, namely to align the camera such as to point forward. Use the buttons to turn the camera until you're satisfied. The goal of this step is to align the camera with the 2nd IMU. A precise alignment

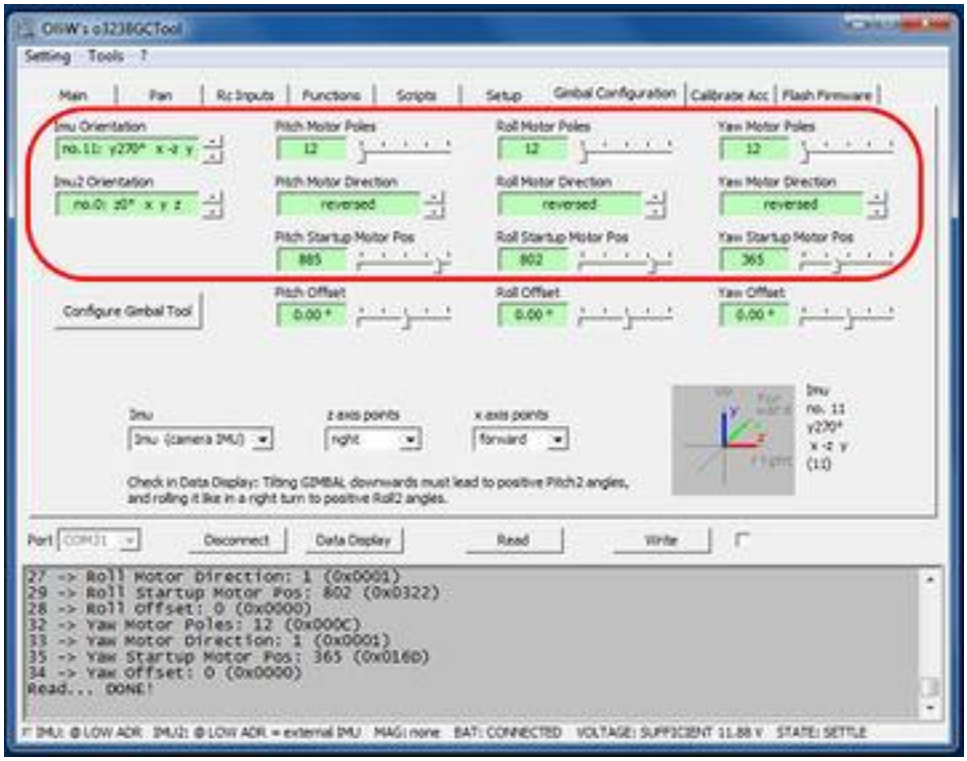
is required for the 2nd IMU function to work correctly. You don't have to overdo however, a visual accuracy of the alignment is sufficient.



This completes the settings of **Steps II**. What is left is to store all values in non-volatile memory, and to restart the gimbal.



The results of the above steps can be seen in the **Gimbal Configuration** tab, which got updated with the new IMU orientations and motor parameters. Also the motors are enabled, and the system is functional now.



PID Tuning

Calibration is already made by Copterlab team. Your gimbal can be used out of box.
We don't advise you to setup it by yourself as this is a complex task.

Before any tuning, the gimbal needs to be balanced. Balancing means to bring the center of gravity (COG) of the mounted camera into the center of all three gimbal motor axes (depending on the gimbal construction it is also part of the balancing to ensure that all three gimbal motor axes cross in someone points). A quick test, which may be sufficient for larger cameras, is to move the camera by hand while all gimbal motors are turned off: The camera should remain in its position whenever you let it go, as demonstrated in the video below.

A perfectly balanced gimbal has many advantages. Otherwise you will get artefacts, such as:

- When the gimbal as a whole is accelerated, the camera would rotate out of position due to residual forces - see the image below.
- The motors need to apply more power to keep the camera in position.
- When turning the camera, e.g. with the RC stick, the center of gravity would change its positions with respect to the center of the gimbal motor axes.
- The PID control actions on different axes get correlated.
- Vibrations in the gimbal are more easily induced.

Balancing not only brings advantages, a minimum degree of balancing is in fact mandatory for the gimbal and controller to function at all. It's as simple as that: The better the balancing the better the gimbal performance. This holds true especially for a 3-axis gimbal.

You will always have to compromise, because e.g. the gimbal does not allow for infinitely precise adjustments, or because of friction in the motors, and so on. Hence take considerable effort in balancing the gimbal as good as possible, but at some point one of course has to let go.

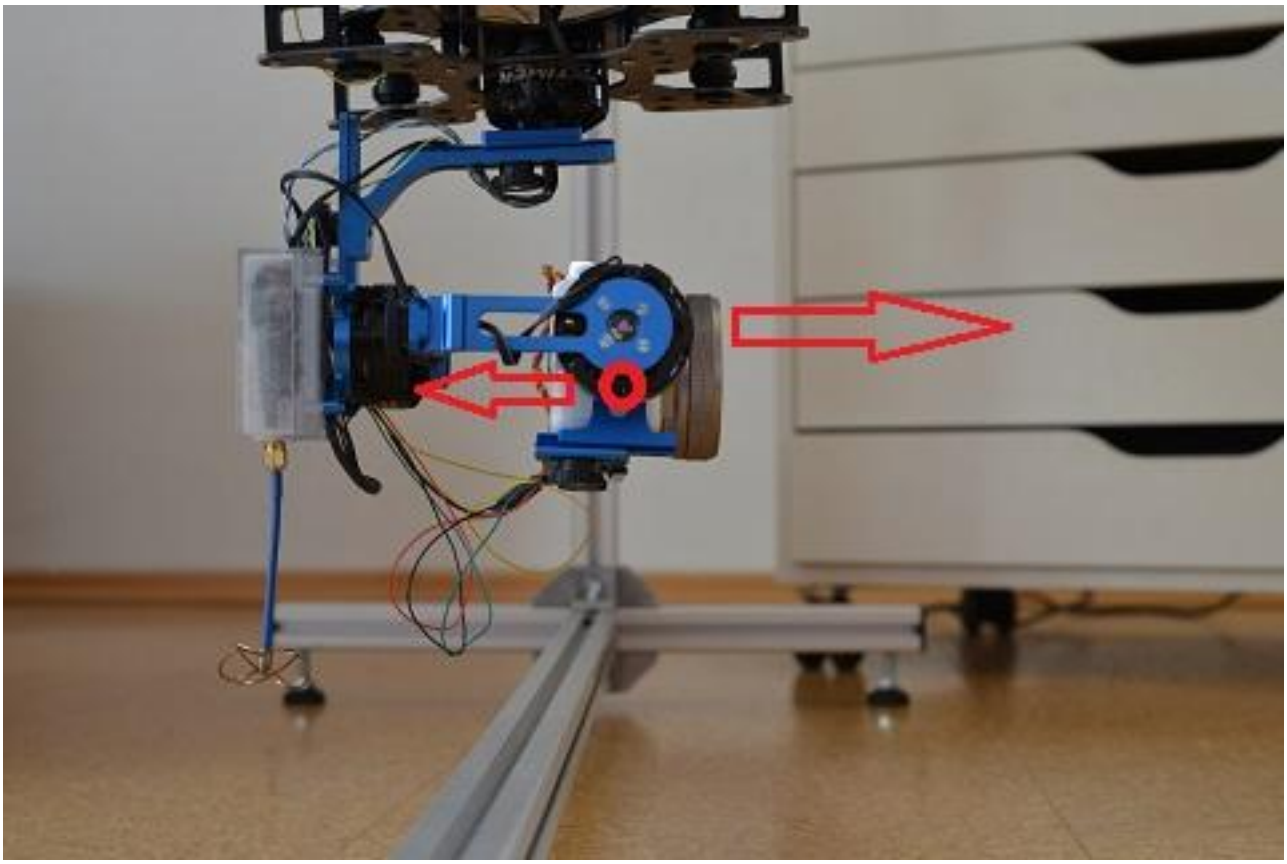
The picture below shows a very badly balanced situation. The center of gravity is quite below the pitch motor axis. At first this might actually look like a good idea, since the camera gets "self-stabilized" by gravity. However, consider now the gimbal being accelerated to the right. Because the center of gravity is below the pitch axis, where would be a force

on the camera and it would tilt downwards. The PID controller would have to fight against that, which makes it much harder or even impossible for it to stabilize the camera.

The video shows a well balanced gimbal. Here it is possible to move the camera to any orientation and let it free, without it turning back to a preferred orientation.

Please also consider these tips:

- It is **important** to electrically disconnect the motors from the board, since then they can rotate more freely than just being disabled in the GUI.
- A much more severe test to check balance is to heavily shake the gimbal, up and down, left and right, forward and backward. When balanced, the camera will stay in the position it was before the shaking started. Do this for different camera positions.
- A 3-axis gimbal requires more precise balancing than a 2-axis gimbal. Especially the yaw axis needs attention.
- In general, the smaller the camera the more important is the balance.



Tuning Procedure

Calibration is already made by Copterlab team. Your gimbal can be used out of box.
We don't advise you to setup it by yourself as this is a complex task.

The tuning of one axis might impact the other axis, nevertheless it is best to tune one axis after the other. Accordingly, the first step in tuning a brand new gimbal is to disable all motors except of the pitch motor, and find the proper tuning parameters for it. Then the roll motor is enabled and the corresponding PID parameters are tuned. Finally, the yaw axis is done.

Our goal in the PID tuning is to find the highest value for each of the P, I and D parameters which still work stably. The higher the values the better distortions can be compensated, and the more accurate the camera will be hold

in position. If a value however is too large, then the PID controller will overcompensate, and we will notice oscillations, noise, or even erratic behavior in the worst case.

To find the sweet point we will start at a low value, increase it until the gimbal starts to misbehave, and lower it until it works again. This will be done first for the D parameter, when the P value, and last but not least the I term.

Motor Vmax must not to raise more than 1v

The Starting Point

Set the P, I and D values to zero.

Also, ensure that:

All three axes should be in hold mode, i.e., in the [GUI:Pan] tab the parameters Pitch Pan, Roll Pan and Yaw Pan should be set to zero or alternatively the Pan Mode Default Setting set to “hold hold hold”.

Voltage Correction should be set to “0%”.

A 2nd IMU should be deactivated, i.e., in the [GUI:Setup] tab the parameter Imu2 Configuration should be set to “off”.

The battery should neither be fully charged nor nearly discharged, i.e., have a voltage of about 3.7 V per cell.

The D Parameter

With P and I set to zero, there is no position correction at all, that is the motor will simply stay in its current position.

The Derivative term of the PID control loop is not directly related to the movement of the camera. It is a high frequency correction looking ahead, in order to dampen what in future could become oscillations. If the D value is too high, then the frequency of the correction gets visible and/or audible. Therefore, by keeping P and I zero and just increasing D, one can find a point at which an audible high-frequency noise, produced by the gimbal motor, appears. The D value should be as large as possible without any noise occurring. Once a good value is found, one should move the motor by hand to double check that no noise appears for any gimbal position. If a D value is working great in almost all positions except one, then that would be indication of an imperfectly balanced gimbal. Go then back to start.

An example tuning is illustrated in the video. Listen to it (switch on audio). Initially D = 0.17, and the gimbal is shaking like mad, with low frequency and high amplitudes. Then the value is reduced to D = 0.10, and the camera position becomes stable, but a high-frequency noise can clearly be heard. With D = 0.05 the frequency gets even higher. We want the frequency to be that high that no noise is audible. At D = 0.04 the noise is suddenly gone.

Before the video was taken, the following actually happened. First D = 0.04 was picked, and everything was fine. With increasing it to 0.05, everything was still fine. At D = 0.06 heavy vibrations were visible, hence the previous D = 0.05 value was set, but now vibrations were still present! In this situation one could manually move the motor into a different position and the vibrations would go away. What happened? At D = 0.06 the resonance frequency of the control loop was hit, but with a slight decrease to 0.05 the frequency remained very similar and the control loop was not able to get out of it by itself. So, to be on the safe side D = 0.04 was chosen, as for this value the shaking stopped by itself. The message is that it's quite natural to observe a hysteresis, that is that when coming from large D values one needs to go to smaller D values to stop a vibration than the other way around. Choose the smaller D value, at which the vibrations stop.

The P Parameter

Before starting the tuning of the Proportional term of the PID controller, set the I parameter to a low value, for instance 5, which is the lowest non-zero value.

The P constant is multiplied with the positional error to get the correction amount, exactly as in the example discussed in the introduction. The higher the P factor is the larger the calculated correction will be. If it is too high, then the PID controller will have the tendency to overshoot the target position, and when it is much too high it even produces totally erratic movements. Hence, the goal is to have both a high P value and a smooth movement in all motor positions.

The video shows the procedure. After five seconds, P and I are set to non-zero values and the gimbal moves into a level position. Then the P value is increased. Once P = 15 is reached, some weird shakes start to appear when the motor is released from a tilted position, which get worse and worse, and with P = 23 the camera shakes for a while even. Only once the P value is set back to 13 a smooth movement from all positions is achieved again.

The I Parameter

Now the Integral term can be adjusted. The goal is, similar to the other parameters, to find the largest possible value without introducing negative side effects.

In fact, the I term plays a much more important role for the stabilization accuracy than the D and P parameters. In practical terms this means that it is more important to set a large I value, or vice versa, that one doesn't lose too much if D and P are not at their highest possible values. So, it is actually not a bad idea to lower D and P, as this increases e.g. the reliability of the PID controller behavior.

When you look carefully again at the last few seconds of the previous video, where I was 5, you'll see that the camera could not be kept level when the gimbal was moved. The I term is responsible for this part of the movement.

In the video, the I value is initially set to 480. I in comparison to the behavior for I = 5, a perfect movement is now obtained: fast speed, the camera remains horizontal, it truly is horizontal, everything is fine. Maybe one can increase even further. At I = 700 the same situation, everything looks nice. At I = 1200, however, the camera isn't horizontally aligned, and - if you look carefully at the yellow cable of the IMU connector - you see that it is vibrating (sorry, not very visible in the video). So, I = 1200 is too high for sure. There were slight vibrations in some positions of the camera even at I = 700, so I was set to 620 at the end.

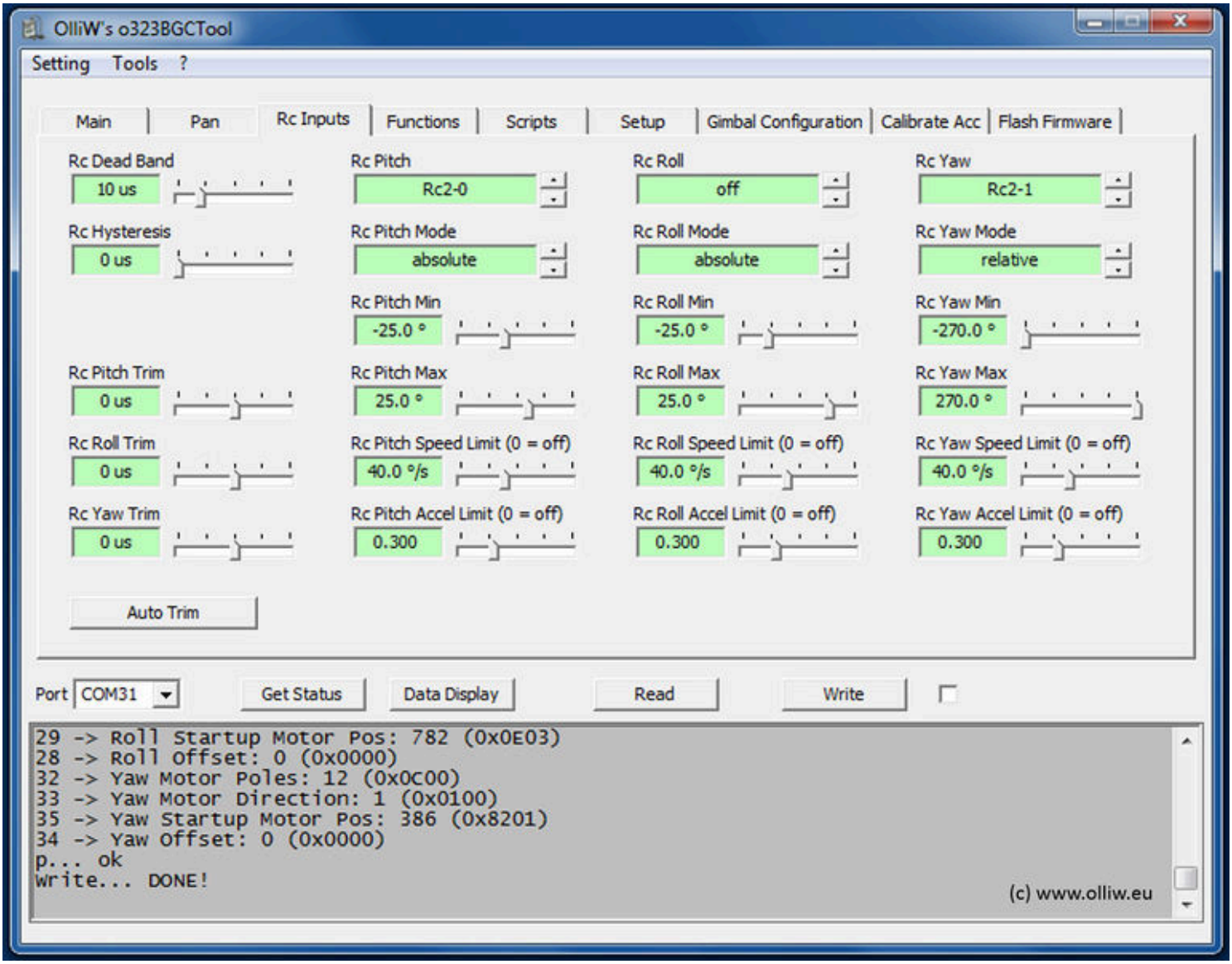
Configure the RC Inputs

You should now have a self-stabilizing gimbal. No matter what movements the gimbal support is undergoing, the camera remains stable in position.

If so, you may want to set the yaw axis back to pan mode. In the Pan tab set the Yaw Pan value to a non-zero value (default is 2.0) and the Pan Mode Default Setting to "hold hold pan".

While the current setting might be perfectly sufficient for basic operation, normally the gimbal shall also be moved via a RC stick or a joystick, or functions such as switching between hold and pan modes or a pan deadband are desired.

As an example the pitch and yaw axis shall be controlled by a transmitter. The receiver outputs shall be connected to the controller board at the RC2-0 and RC2-1 pins for pitch and yaw control, respectively.



The orientation of the camera can be controlled by a variety of external input signals such as from a receiver, a joystick, or buttons. The source of the input signal and the exact behavior of the camera in relation to the input is configured in the **Inputs** tab, which is described in the first chapter below. Physically the signal sources need to be connected to certain pins on the controller.

GUI Settings

The response of the camera to an input signal, e.g. a movement of a transmitter stick, is determined by these parameters:

- Rc Dead Band: Defines how wide the neutral range around the input signal's center is. It becomes relevant e.g. then transmitter/receivers are used to control the relative movements, the Yaw axis usually. If you find a slow but constant movement on the Yaw axis because the transmitter stick does not re-position itself perfectly to center, then increase the deadband range a bit. 10 us requires a precise setting, double it or more. A typical value would be 15 us. Applies to all input channels.
- Rc Hysteresis: Defines a neutral range around the current value of the input signal. It becomes relevant e.g. then the input signal fluctuates with time even though the transmitter stick is not moved. A typical value could be 5 to 10 us. Applies to all input channels.

- Rc Pitch Trim, Rc Roll Trim, Rc Yaw Trim: These values allow for a fine adjustment of the neutral camera orientation for each axis. Applies to all input channels, also a joystick if selected as input. The [Auto Trim] button reads the current input values and adjusts the trim values such as to make them zero.

There are further fields for precisely configuring the camera movements. They can be adjusted for Pitch, Roll and Yaw individually, but function identically for each axis (with one exception). Let's use Yaw as example:

- Rc Yaw: This defines what pin this signal is taken from, and wherewith implicitly of what type it is (PWM, Sum-PPM, analog, digital).
- Rc Yaw Mode: This defines how the stick movements affect the camera orientation. Four modes are possible for yaw:

"absolute centered": The stick controls the orientation of the camera, but such that when the stick is in neutral position the camera points exactly forward. When the stick is held completely to one side the camera orientation is given by the min (or max) degrees set by Rc Yaw Min (or Rc Yaw Max). In this mode the deadband set by Rc Dead Band is applied. This mode is the default mode (as it allows to bring back the camera to the startup orientation).

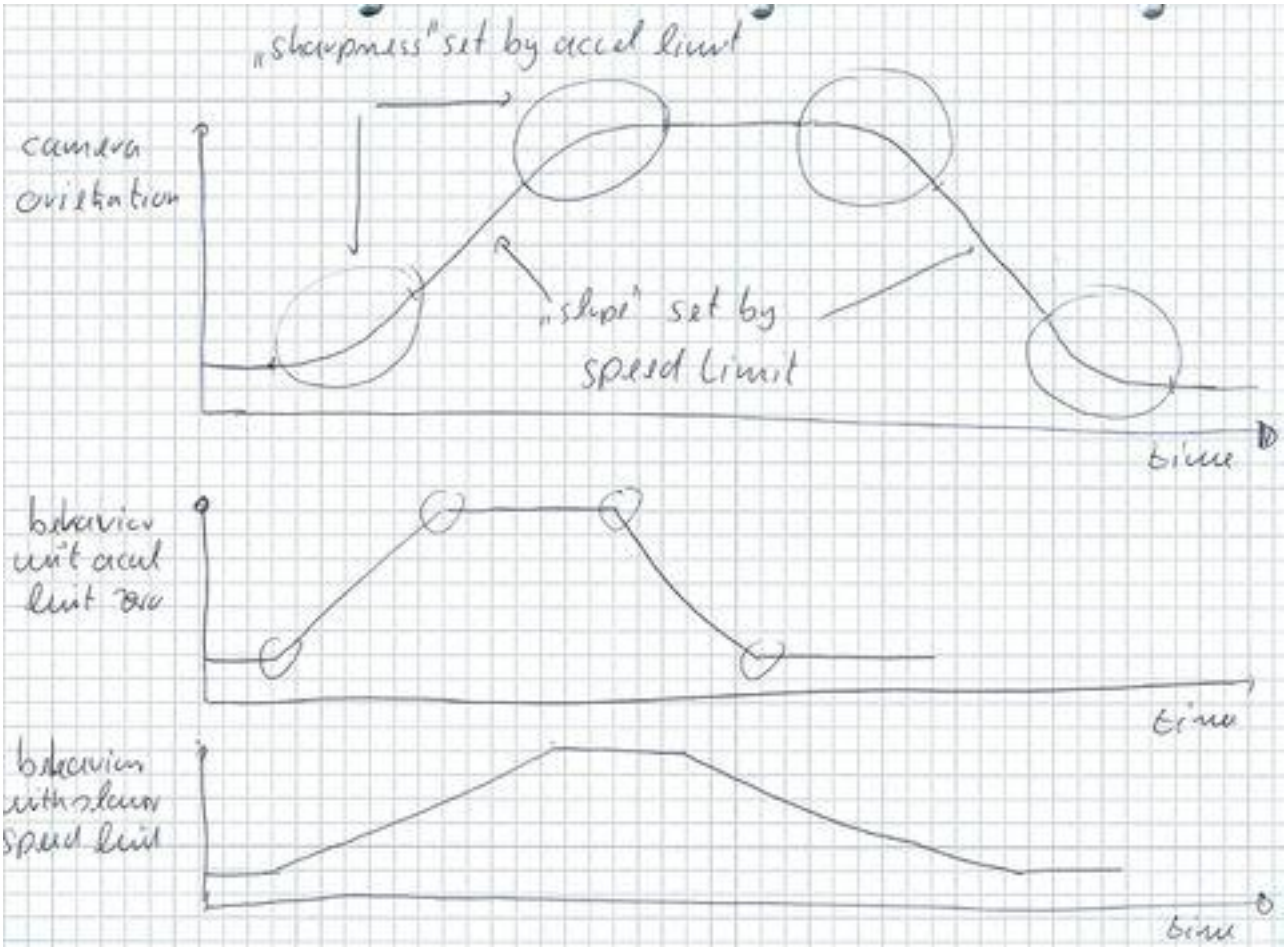
"relative": The stick controls the movement of the camera. That is, stick in neutral position means no movement, and stick fully to one side means moving at full speed. This mode can e.g. be useful for a stick that positions itself back to center automatically.

"absolute": In this mode the stick controls the orientation of the camera. The camera's orientation is obtained by linearly interpolating between the min and max degrees set by Rc Yaw Min and Rc Yaw Max. A deadband is not applied. This mode can e.g. be useful for the pitch axis. Note that in this mode the camera moves to the center position at startup even when Rc Yaw is set to "off" (which might be confusing but is the correct behavior).

"relative turn around": For the yaw axis also this setting is available. It operates like the "relative" mode, but switches off the Min/Max limits. The camera can then be turned around indefinitely.

- Rc Yaw Min, Rc Yaw Max: The minimum and maximum value in degrees allowed for yaw movements. You can move the camera between these two extremes only. If the minimum value is greater than the maximum value, then the direction is reversed.
- Rc Yaw Speed Limit: The maximal speed with which the camera is turned in degrees per second. When panning the camera you do not want an extremely fast turn rate. Neither would that be controllable via a stick nor would the gimbal be able to cope with the forces. Hence you have the option to limit the turn rate to an upper speed limit.
- Rc Yaw Accel Limit: The maximal acceleration with which the camera can be moved. The larger the value the more the acceleration is reduced. When the stick is e.g. moved from full left to full right, the camera should not make the same abrupt movements. It should slowly turn faster and faster according to the given acceleration limit, as well as should slowly get slower and slower.

The effect of the Rc Yaw Speed Limit and Rc Yaw Accel Limit parameter values are also explained in this graphic:

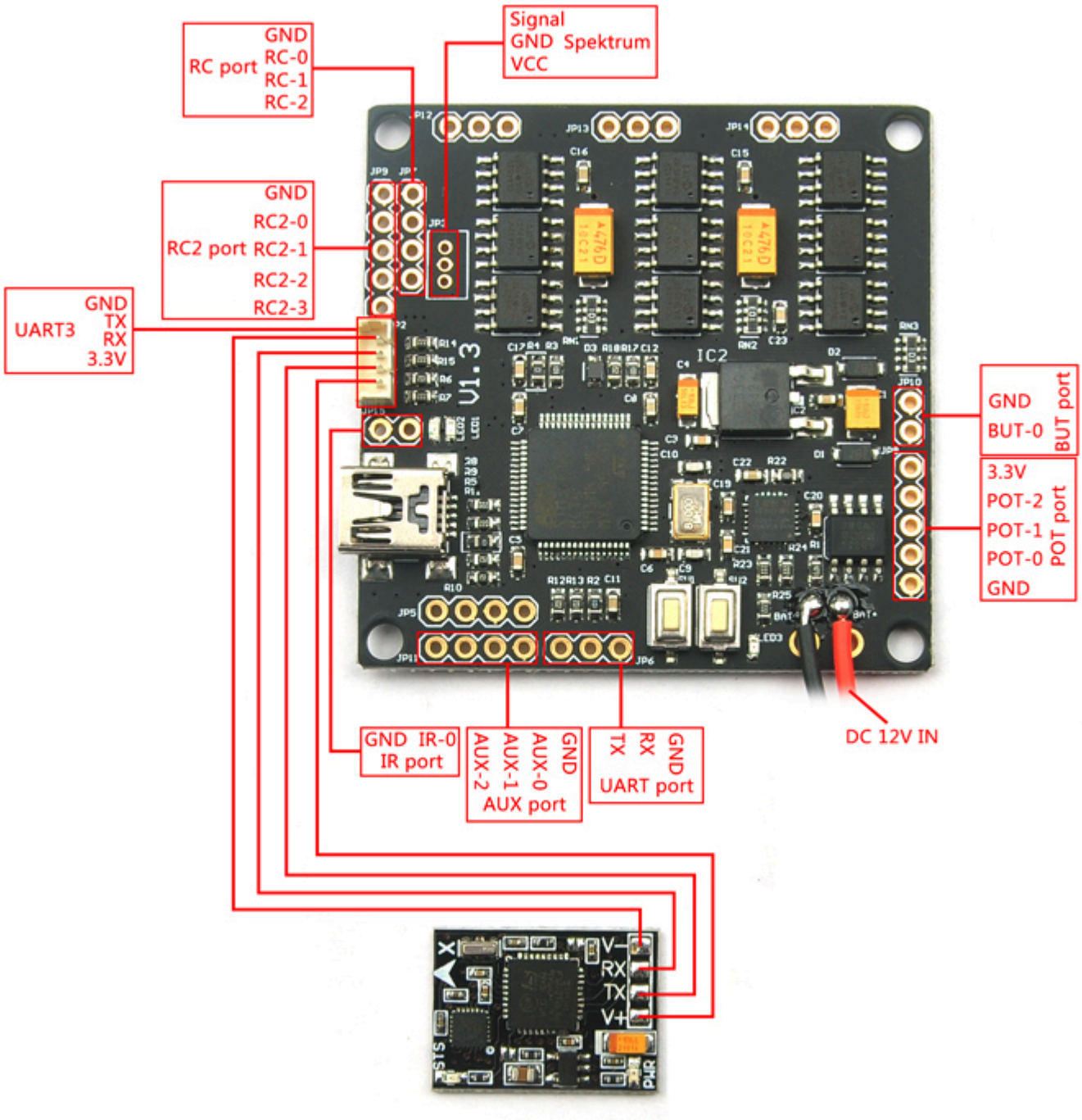


Wiring for PWM Signals

For each channel the RC receiver exposes a three pin header, which provides GND (black/brown wire), +5 V (red wire) and the RC signal (white/orange wire). Internally the GND pins of all channels are connected to each other, and likewise for the +5 V pins. The STorM32-BGC board however has only two GND pins plus the various signal pins, and no +5 V, at the RC and RC2 ports (see Pins and Connectors). This is to save space and reduce the number of cables, but also for safety (the STorM32-BGC board works internally with +3.3 V only).

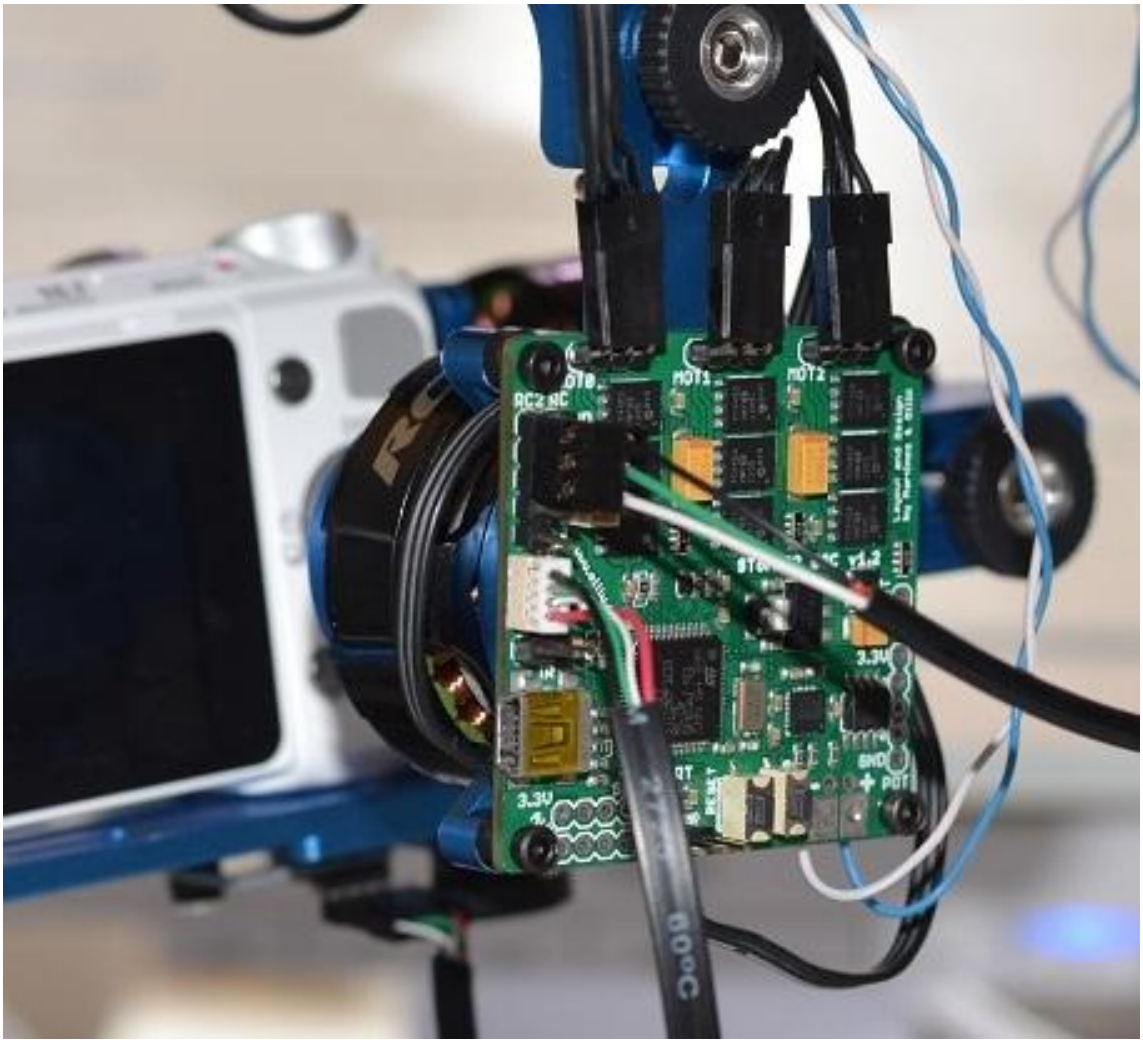
Therefore a typical connection (using the RC2 port as example) will look like this:

- One of the receiver's GND pin is connected to RC2-GND, which is the pin next to the label (see image below, the black GND cable is the top most)
- The receiver's channel 3 signal pin is connected to RC2-0 (green cable below) used for pitch
- The receiver's channel 4 signal pin is connected to RC2-1 (white cable below) used for yaw
- Roll is usually not controlled by the RC and hence not connected. There isn't any reason why the camera horizon shouldn't be level, isn't it?

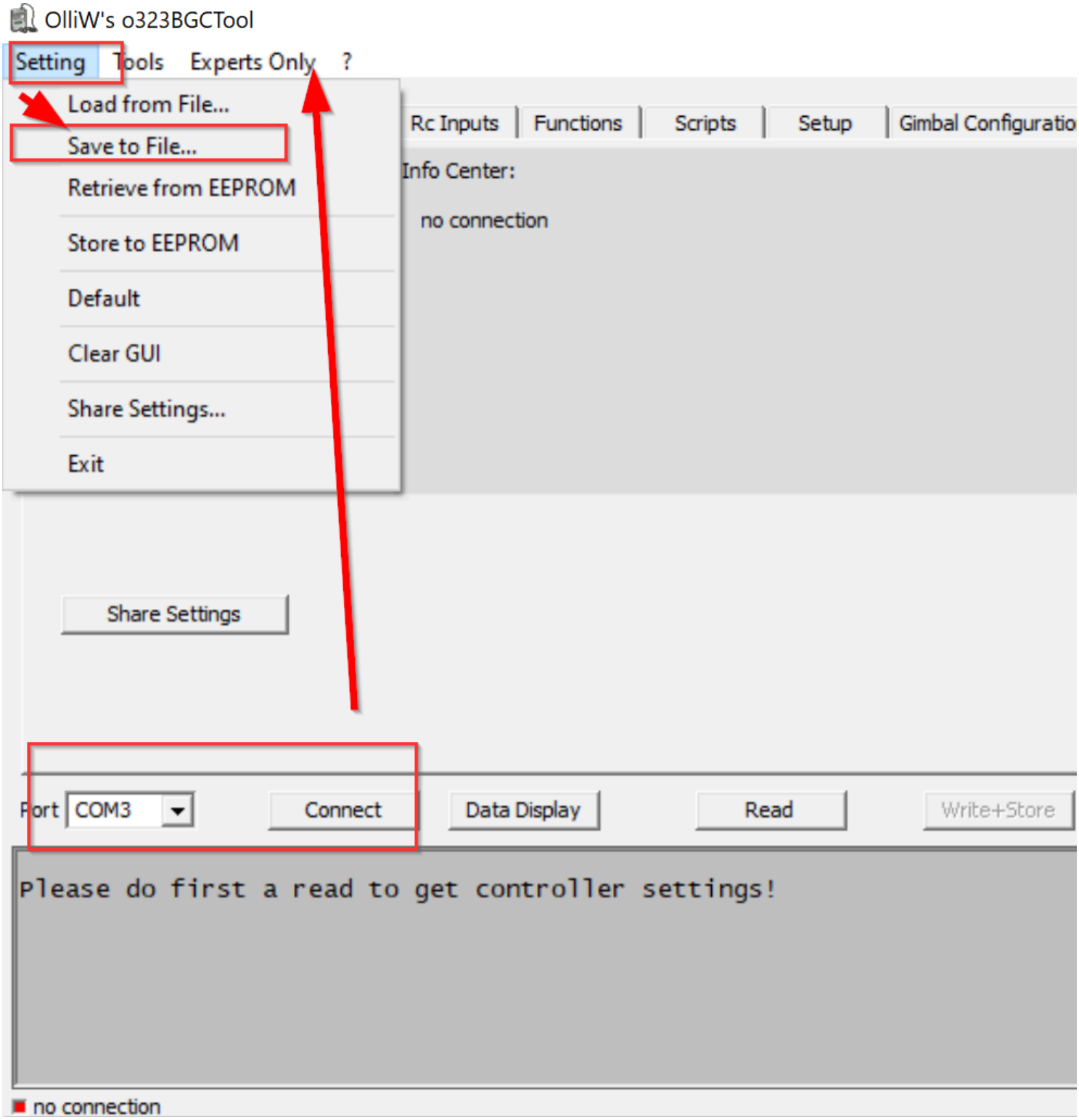


Storm32 NT diagram

Comment: *The +5 V line of the receiver **must not** be connected to the STorM32-BGC board. The board is powered by the battery directly; the receiver by e.g. a separate BEC module, which is usually part of the ESC.*

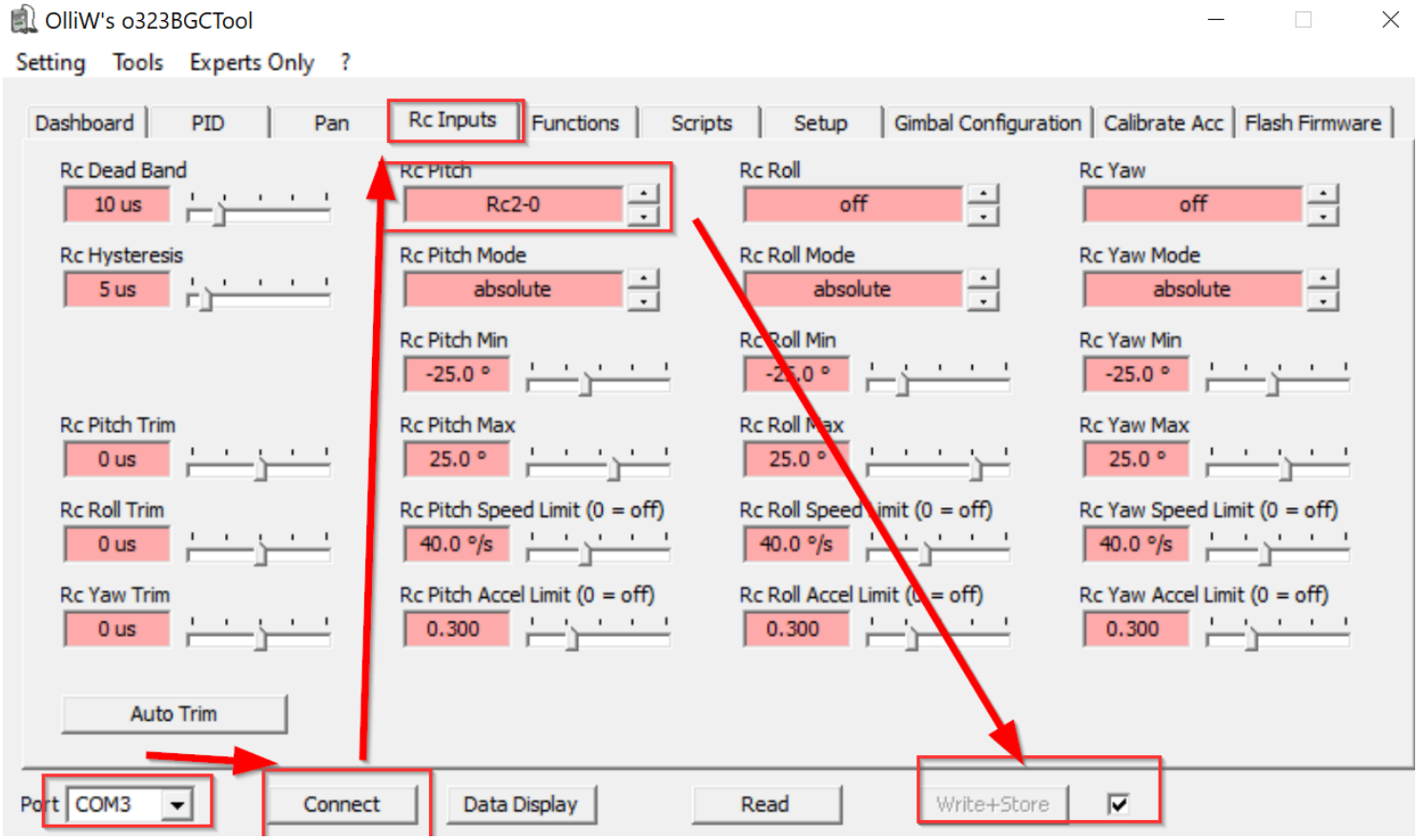


Before doing any change, we *highly* recommend you to backup the controller settings

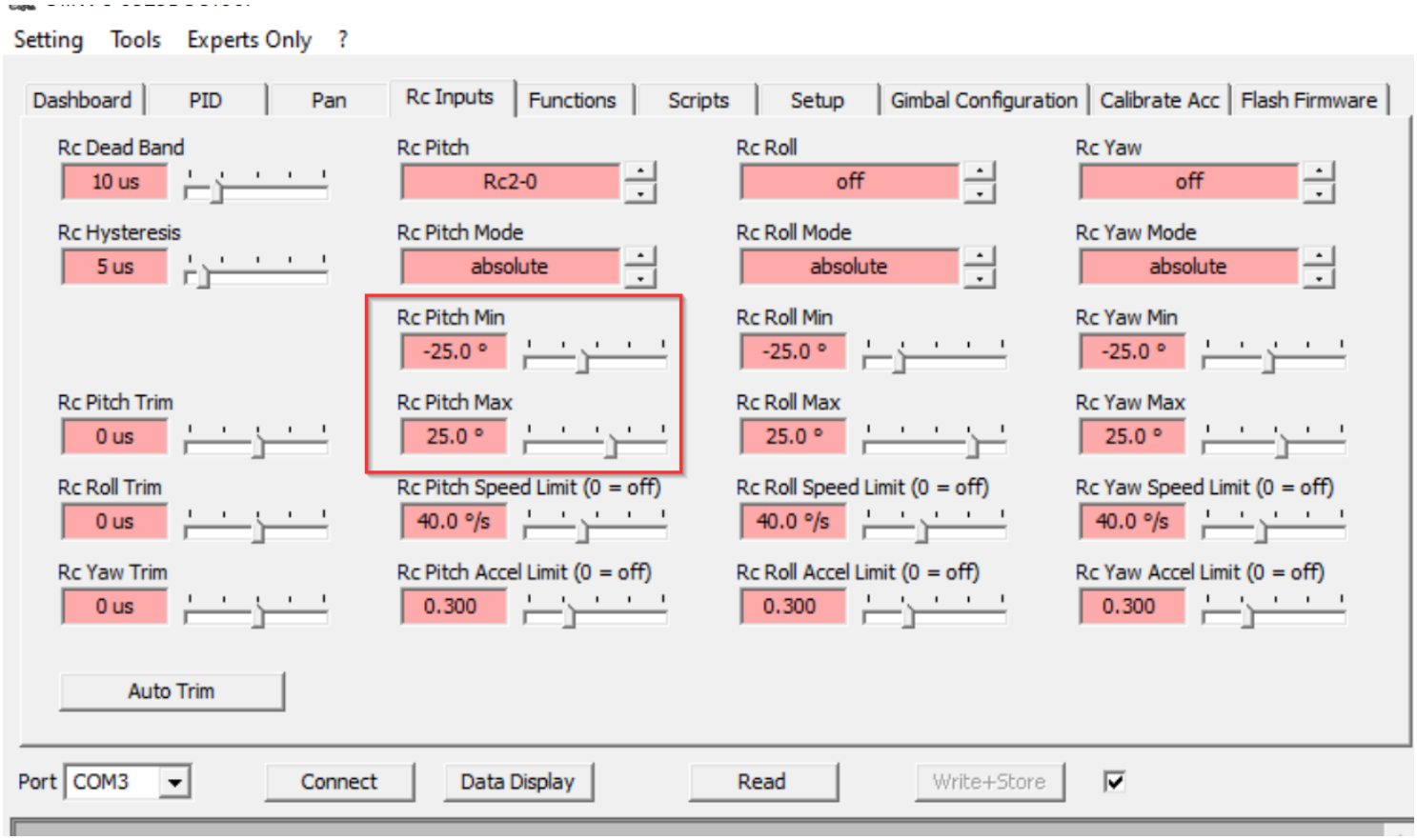


Locate the RC2-0 input pin (it can be another pin entry if you want) and connect it with a cable to your PWM output receiver

Now check if the Rc pitch entry is mapped to RC2-0 on your board



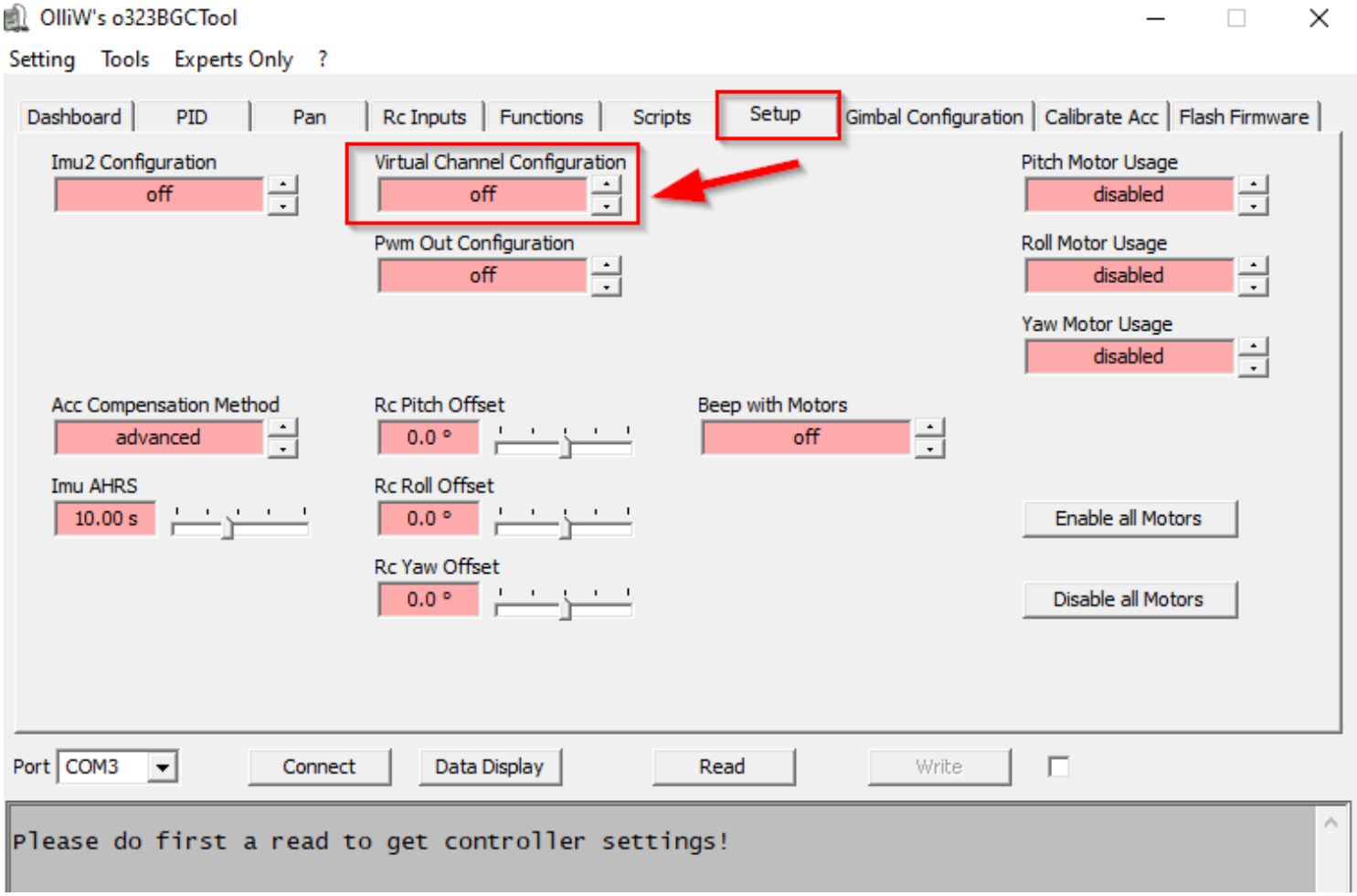
The angle range can be changed here



Last thing to check is the gimbal controller set for PWM signals

"off" value means PWM activated

You can change it to another protocol if you want



Wiring for SUM-PPM Signals

For using a Sum-PPM signal, also known as CPPM, two things need to be done:

- Connect the Sum-PPM signal to the RC-2 pin on the RC port (see Pins and Connectors). Also a GND connection is of course required.
- Configure the parameter Virtual Channel Configuration, which is found in the **Setup** tab, as a Sum-PPM. Select e.g. “sum ppm 8” if your Sum-PPM signal transmits eight channels.

The data transmitted by the Sum-PPM signal can then be used in any available function by selecting “Virtual-1” to “Virtual-12” as input.

Comment: *With activated Sum-PPM, the “Rc-2” input channel returns zero value.*

Wiring for Digital Signals

All STorM32 boards support Spektrum satellite, HoTT SUMD and Multiplex SRXL receivers. The v1.3 board has also support for the S-Bus (Futaba, FrySky, OrangeRX).

For using any of these signals, the procedure is as follows:

- Connect the digital data line to the RC-0 pin on the RC port (see Pins and Connectors). Also a GND connection is of course required.
- Configure the parameter Virtual Channel Configuration, which is found in the [GUI:Setup] tab, to the respective digital signal.

The data transmitted by the data line can then be used in any available function by selecting “Virtual-1” to “Virtual-16” as input. The “Rc-0” input channel will however return zero value.

Spektrum Satellite

Configure the parameter Virtual Channel Configuration to “spektrum 10 bit” or “spektrum 11 bit”, depending on the type of the protocol in use.

The v1.2/v1.3 boards provide a dedicated Spektrum port. On the v1.1 boards the satellite's black wire needs to be connected to GND, the orange/red wire to one of the 3.3V pins, and the grey/white signal wire to the RC-0 pin (see Pins and Connectors).

Comment: *The STorM32 board can power the satellite, if no other sinks are connected to the board. On v1.1/v1.2 boards it may happen though, that the current draw is to high.*

S-Bus

Configure the parameter Virtual Channel Configuration to “sbus”.

Comment: *v1.1 boards do not support the S-Bus.*

HoTT SUMD

Configure the parameter Virtual Channel Configuration to “hott sumd”.

Rcgroups user fpvberlin has produced a nice tutorial for setting up HoTT SUMD, see here: How to use Graupner HoTT with the Storm32bgc.

Comment: *The HoTT receiver needs a 5 V power supply, which cannot be obtained from the STorM32 board. Use a BEC or any other 5 V source.*

Multiplex SRXL

Configure the parameter Virtual Channel Configuration to “srxl”.

Comment: *The Multiplex receiver needs a 5 V power supply, which cannot be obtained from the STorM32 board. Use a BEC or any other 5 V source.*

Wiring for Joysticks

There are many different types of joysticks, and not all cases can hence be considered here. The most common situation is that of a resistive joystick, there the joystick positions are detected by potentiometers, one potentiometer for each axis.

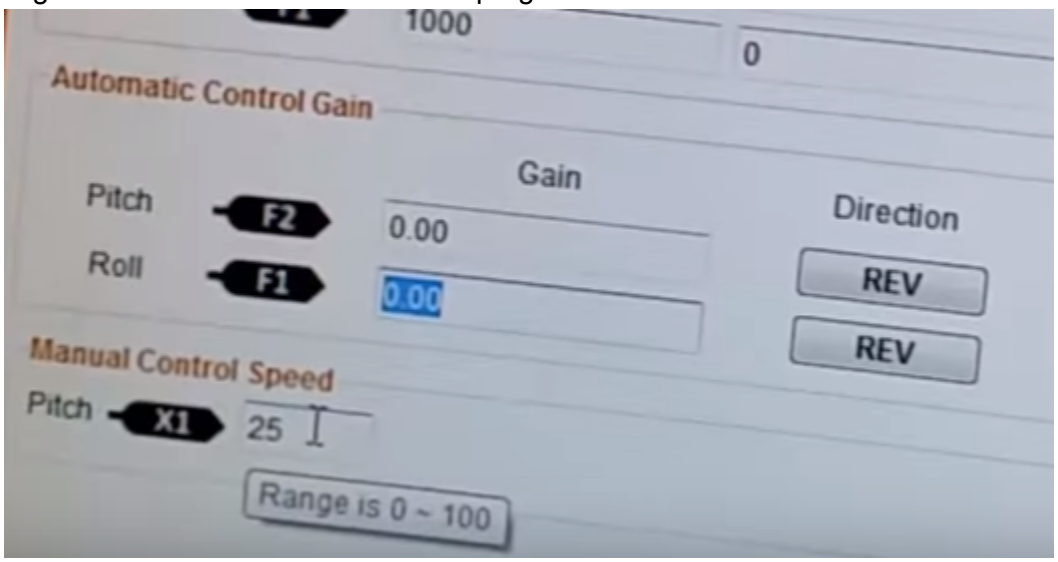
Each potentiometer is connected to the POT port (see Pins and Connectors) as follows:

- The center tap (slider, wiper) goes to one of the pins POT-0, POT-1, or POT-2.
- One of the ends goes to the GND pin, and the other end to the 3.3 V pin.
- Be careful to not produce short cuts.

The potentiometer position or related joystick can then be used for control in any available function by selecting the “Pot-0”, “Pot-1”, or “Pot-2” option, depending on whether the potentiometer is connected to the POT-0, POT-1, or POT-2 pin.

DJI Naza/A2 Pitch Control

- Go to RC settings TAB and change Pitch control minimum and maximum angle to -90 and + 90 degrees.
- Enable the gimbal control in the assistant PC program of the Naza on the Gimbal tab.



You can adjust pitch control between -1000 and 1000. It is recommended to set automatic gain control to 0 or 1.0 for pitch.

Connect only the GND and Signal pins of

- your free receiver channel
- or in F2 port in DJI Naza or in DJI Phantom FC
- or S1 on Quanum Nova
- or D2 connected to RC-Pitch in A2

to the pitch control port of the gimbal controller. The GND is the bottom and the Signal is the top pin on the board.

For pitch control you only need a simple male to male servo wire or 2 DuPont cables. If you use a servo wire please disconnect the middle power red wire and use only white or yellow as signal and black or brown as ground. Note: Never use middle pin under the signal (this is usually red servo wire) as it is the power supply and is not needed for pitch or tilt control and can damage the board and will void warranty.

Part 3 - Powering your gimbal

NOTE: Please make sure that the camera is mounted on the gimbal before it's powered on, as without the counterweight the gimbal cannot work properly!

When connecting the power, the gimbal should be left in a stable position until it completes initialization (5-10 seconds). If the gimbal is moved or shaken during the calibration period it can cause unbalanced operation and malfunction.
If the gimbal does not level correctly, disconnect and reconnect it with the battery, hold your camera in leveled position and wait until the gimbal starts up.

GUI installation

(Only for PC computer)

How does Read, Write, and Store work?

[Read] reads the currently active settings from the board to the GUI.

[Write] writes the values of the GUI to the board and makes them effective immediately. The changes last until the board is reset or powered down.

[Store] makes the board stores its current values into the EEPROM so that they become permanent. The settings stored in the EEPROM will be used at power up.

[Write+Store] does first a Write and then a Store.

Thus: With the [Write] button, all parameter values are copied to the board, and become effective immediately. They are however not stored permanently in the EEPROM, i.e., changes will be lost after a power down. To store changes permanently do a [Write+Store], e.g., by clicking the check box next to the [Write] button which turns it into a [Write+Store] button.

How to enter precise parameter values?

The o323BGCTool GUI doesn't allow to enter values in the text fields directly. The sliders can be moved using the mouse, or via the up/down keys when selected. The later allows precise adjustment of the values.

Which drivers are needed for the USB?

On Win7 the driver is usually installed automatically. This may take some minutes, be patient! It is important to wait until Win7 has finished the install before doing anything else. The correct Virtual Com Port (VCP) driver, usable also for WinXP and Win8/8.1, can be obtained from the STM download page. ***Comment:*** *It is crucial to follow the instructions in the readme.txt file, otherwise it won't work!*.

Storing the parameter values to the EEPROM doesn't seem to work

Please double-check that the correct firmware for your board has been flashed. The problem typically happens then e.g. the firmware for a F103RB is flashed onto a board with a F103RC.

Flashing with the USB-TTL adapter produces Windows error messages

If upon flashing you get Windows errors such as mfc110.dll is missing, msvcr110.dll is missing, or application was unable to start correctly (0xc000007b), then first check that you're not using an outdated GUI version. Otherwise, you need to install the VC++ runtime libraries. Download vcredist_x64.exe or vcredist_x86.exe depending on your Win system from here, and run the exe file.

Part 4: Hold versus Pan Mode

Generally speaking, two situations can be distinguished as regards the gimbal behavior: The camera should remain stable in relationship to the ground (hold mode) or in relationship to the gimbal frame (pan mode). The STorM32 controller allows you to choose the mode for each of the three axes independently.

What does stabilization mean?

The term "stabilization" can mean different things depending on the application, and on the axis we talk about.

Imagine you have put your camera on a tripod and are now trying to rotate the camera by 360° for a panorama shot. In pan mode the goal is a stabilization in the sense of removing camera shakes but in overall to follow the turn of the tripod. In other words, the yaw axis stabilization acts like a virtual tripod. The other type is called hold mode, meaning

no matter what you do, the camera will hold its current position. Here, no matter what direction you point at with the tripod, the camera will compensate for this movement and will remain static, i.e. hold its initial position relative to the ground.

The parameters to adjust all this are found in the **Pan** tab in the GUI.

Hold versus Pan

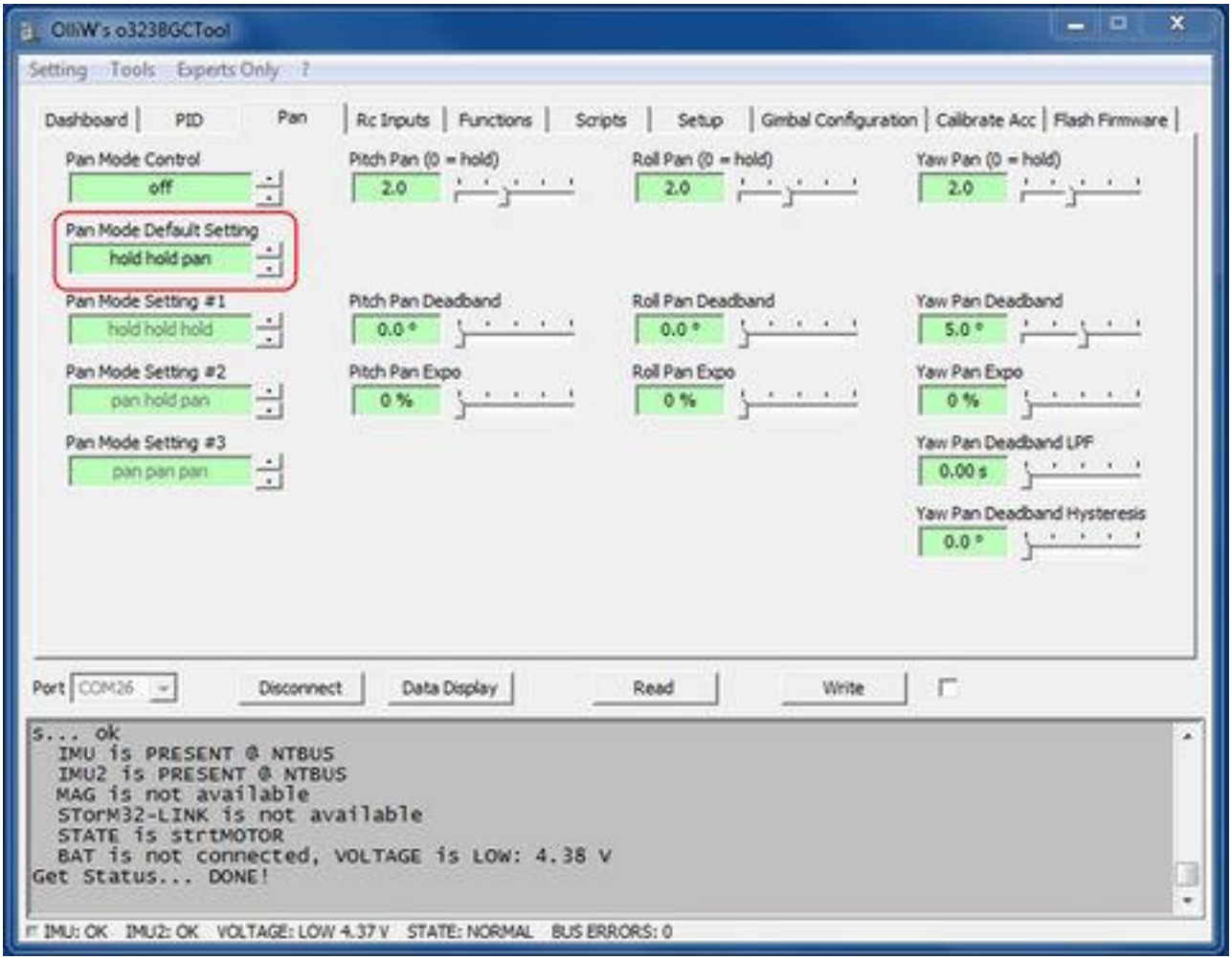
Which mode you want for which axis is defined in the Pan Mode Default Setting parameter.

The standard setting is “hold hold pan” in order to lock the camera in pitch and roll but allow yaw movements. This is, for most situations, the appropriate setting: What should happen if you accidentally tilt the camera forward to the ground? Nothing, i.e., the gimbal should compensate this unwanted movement and keep the horizon in the video at the same level. Same thing with the roll axis. Just because you do not hold the gimbal perfectly level, the horizon should tilt to one side? No, the horizon should be kept level. For yaw movements, however, a different behavior is normally desired. If you would have enabled the hold mode for yaw and would turn the gimbal by 180°, you would film the gimbal frame since the camera would remain stable relative to ground. Doesn't make much sense. Instead the camera should have followed the 180° turn, which is achieved by “pan”. So, the “hold hold pan” setting does make sense for most cases.

Now imagine a hand-held gimbal used in the mountains. Since the pitch is set to “hold” as default, you would either film the ground or the mountains on the other side of the valley, but not your target, maybe 15° higher. One way to adjust the pitch would be via the joystick, which however would be inconvenient. An alternative is to enable pan on the pitch axis, and use “pan hold pan” for the Pan Mode Default Setting parameter. Then the gimbal would dampen any shakes in the pitch axis but in general you can point the camera up and down by moving the gimbal frame.

A completely different example would be a gimbal mounted in an airplane and to record movies from the pilots point of view. When the plane is pointing downwards the camera should look downwards, when the plane is flying a right turn the horizon should tilt. The only task of the gimbal is to remove shakes and allow the "pilot" to look left/right/up/down. That would ask for a Pan Mode Default Setting of “pan pan pan”.

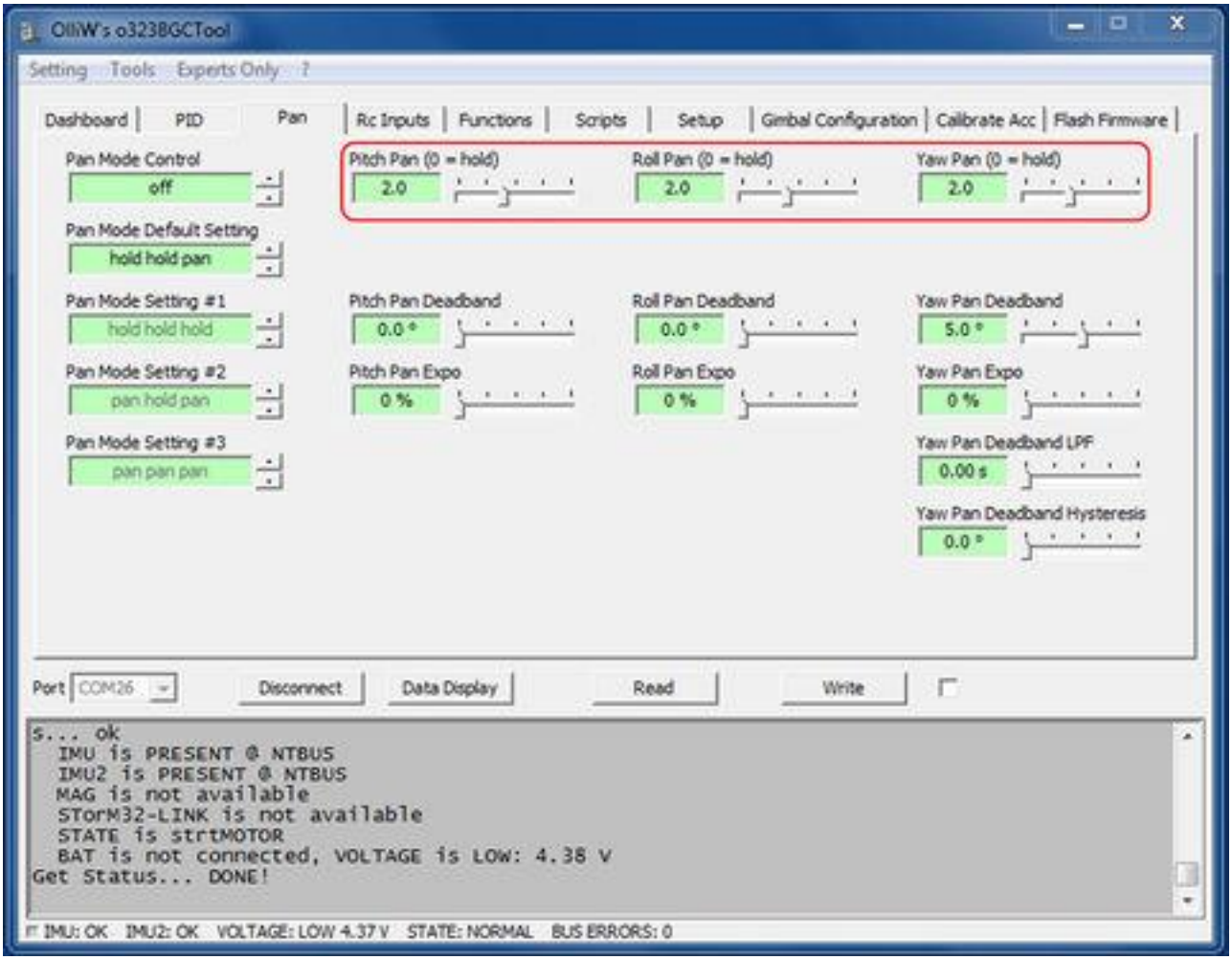
Comment: *It is strongly advised against using a brushless gimbal as an FPV camera! The purpose of the example was just to explain things.*



Fine Tuning the Pan Mode

How does the controller differentiate between unwanted shakes versus true movements? In the **Pan** tab the parameters Pitch Pan, Roll Pan and Yaw Pan are found, with which you determine the follow speeds in pan mode for each axis. A zero means that the follow speed is zero, which effectively corresponds to hold mode. A non-zero value enables the pan mode, whereby a larger pan value means a faster following.

Importantly, whether these settings become active or not depends on the Pan Mode Default Setting parameter: If “hold” is specified for an axis there, then this axis will be in hold mode irrespective of the pan speed value. The Pan Mode Default Setting overrule the pan speed setting.



Deadbands

The STorM32 controller provides sophisticated deadband mechanisms, which are controlled through various settings. Generally, these settings are effective only when the respective axis is in pan mode; they do not affect the behavior in hold mode.

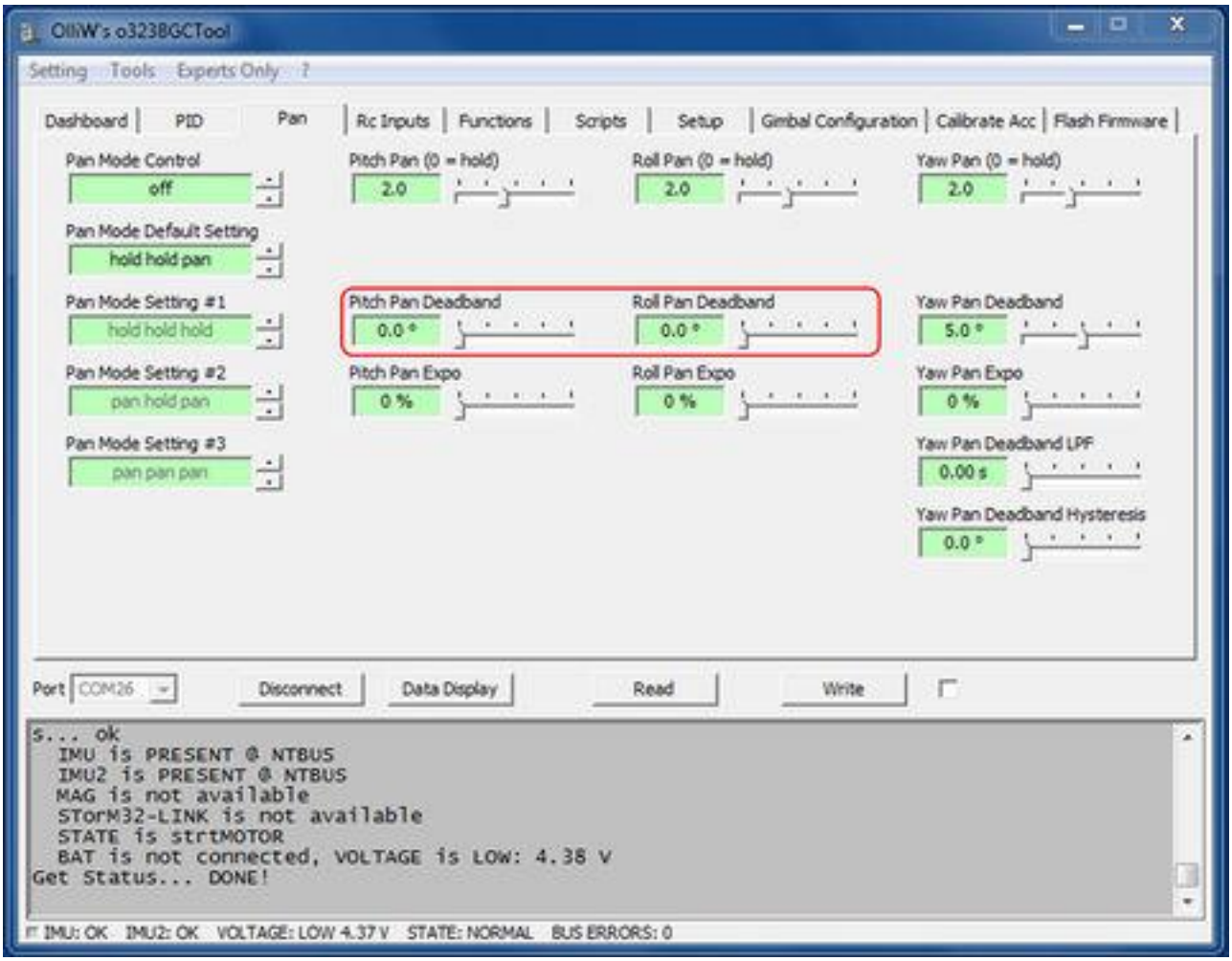
The deadband feature allows you to eliminate any camera movement along the selected axis, when the deviation of the camera from the gimbal center falls within a certain angle range, the deadband. That means that the camera will follow the gimbal frame only when the relative angle between camera and gimbal frame is larger than the given deadband.

The details of the mechanism differ for the pitch&roll axes and the yaw axis, and are thus described separately.

Pitch and Roll Deadbands

For the pitch or roll axis the deadband behavior is determined by the Pan Deadband parameter for the respective axis. Depending on whether the deadband is smaller or larger than 15° the "default" or the "plane" pan mode becomes effective:

- ***default pan mode*** (deadband < 15°): The deadband center follows the gimbal frame. This behavior is e.g. appropriate for hand-held gimbals, to stabilize the video at the current camera position.
- ***plane pan mode*** (deadband > 15°): The deadband center is held fixed at the normal forward orientation. This behavior allows one e.g. to prevent that the camera hits some mechanical limits upon too large pitch or roll movements.



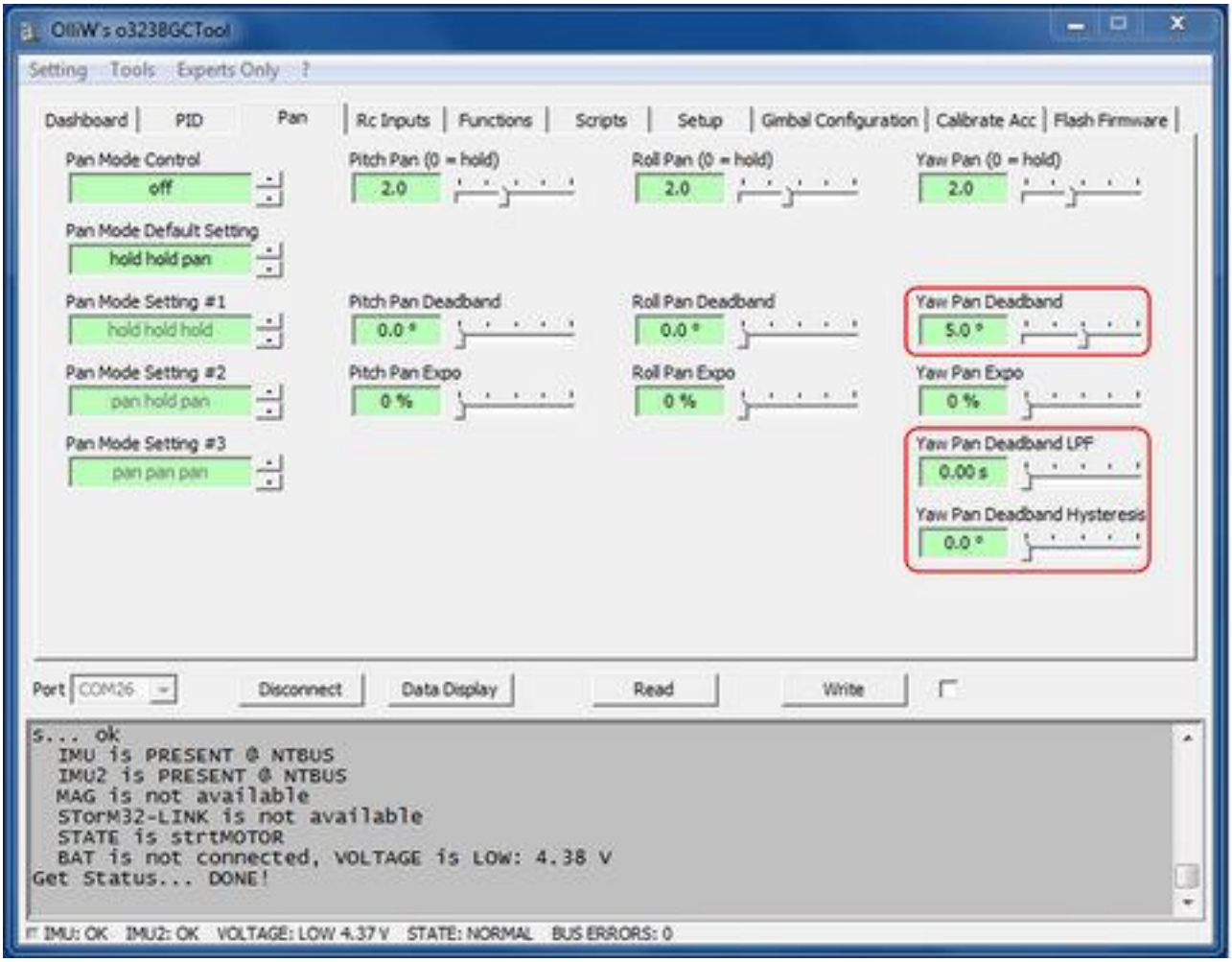
Yaw Deadband

The deadband behavior for the yaw axis is determined by the three parameters Yaw Pan Deadband, Yaw Pan Deadband LPF, and Yaw Pan Deadband Hysteresis. In general, the deadband for yaw works like the ***default pan mode*** for the pitch/roll axis, but provides for more adjustments in order to offer greater flexibility, but also to "fight" yaw drift.

In contrast to pitch and roll, the gyro drift for yaw cannot be compensated for by the accelerometers. The STorM32 controller is capable of compensating the gyro yaw drift when in pan mode, but it's impossible when in hold mode. Since the deadband works essentially like an angle-controlled transition from hold to pan mode, and vice versa, the situation arises that when the camera is within the deadband (= hold) it drifts slowly towards one edge of the deadband, where the controller switches to pan, which pulls back the camera to within the deadband (=hold), where it drifts to the deadband edge, is pulled back, and so on. This can in unfortunate situations lead to a wiggle in the yaw axis. The Yaw Pan Deadband LPF and Yaw Pan Deadband Hysteresis parameters provide two means to alleviate the situation.

The Yaw Pan Deadband LPF parameter allows to slow down the transition from pan to hold and vice versa. It leads to smooth behavior, and tends to pull the camera into the deadband region, which both helps with the gyro yaw drift. A typical value could be 1.5 secs.

The Pan Deadband Hysteresis parameter helps to avoid the unwanted effects due to the gyro drift by dynamically increasing the effective deadband whenever an axis is within the deadband. A typical value could be 2°.



Switching Between Pan/Hold

Sometimes it might be desirable to switch during a shot from pan to hold or vice versa. The switch can be made via any of the available inputs, selectable in the Pan Mode Control field. The following discusses the use of buttons, as it might be appropriate for handheld gimbals.

Buttons can be connected to the AUX-0, AUX-1, and AUX-2 pins (see Pins and Connectors), which allows you to switch through up to four preset pan modes specified in the parameters Pan Mode Default Setting to Pan Mode Setting #3. Let's consider as example two buttons connected to AUX-0 and AUX-1, respectively, and these settings:

- Pan Mode Control = “Aux-01 switch” (meaning that buttons are connected to Aux-0 and Aux-1)
- Pan Mode Default Setting = “hold hold pan”
- Pan Mode Setting #1 = “hold hold hold”
- Pan Mode Setting #2 = “pan pan pan”
- Pan Mode Setting #3 = “pan hold pan”

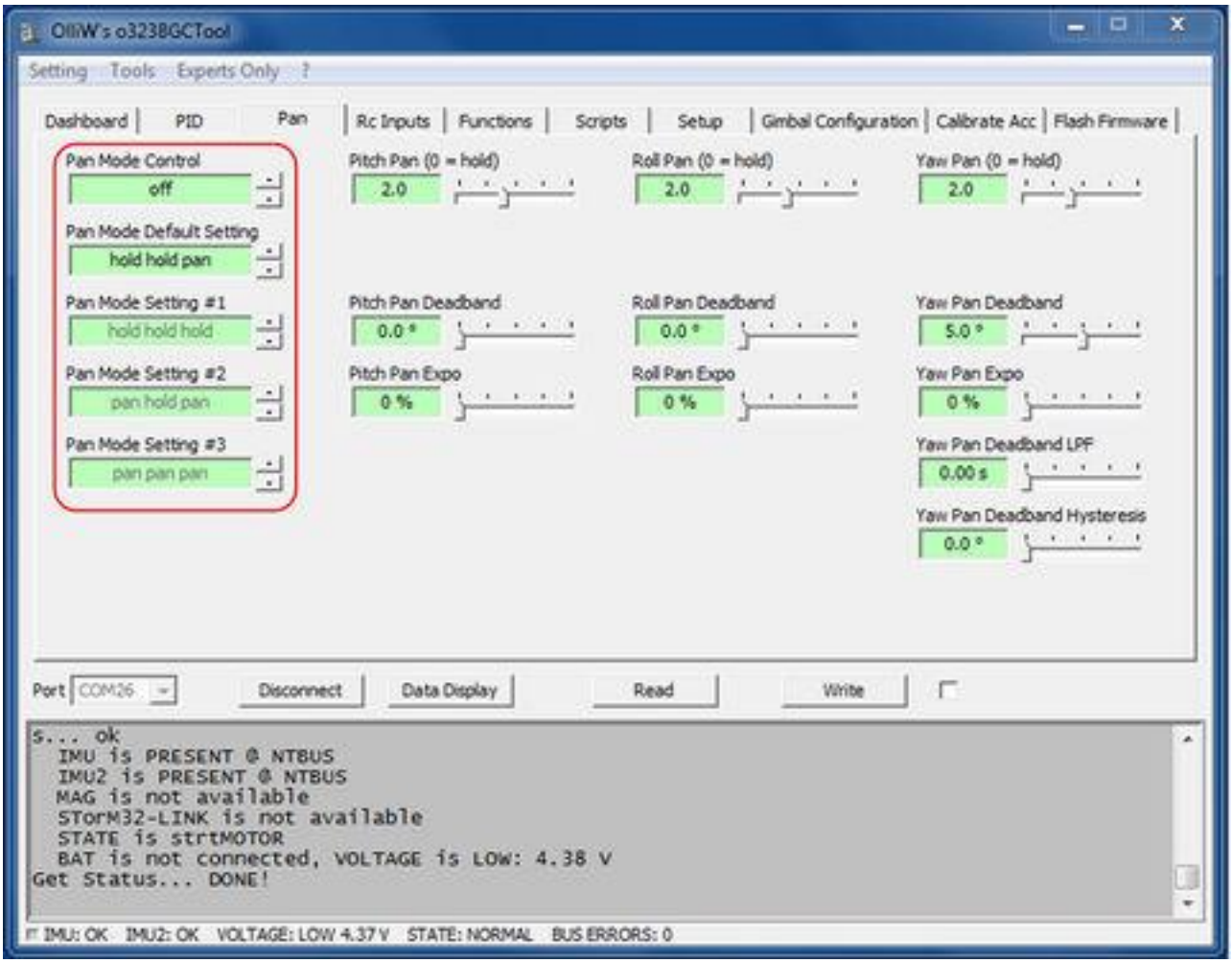
If none of the two buttons are pressed or switched on, then Pan Mode Default Setting is active, which in our example is “hold hold pan”. This means that the pitch and roll axes will be in hold mode, and yaw in pan mode.

If only the button connected to AUX-0 is pressed/switched on, then the Pan Mode Setting #1 becomes active. In our example all three axes are then in hold mode.

If only the button connected to AUX-1 is pressed/switched on, then the Pan Mode Setting #2 is selected and “pan pan pan” would be activated.

Finally, if both buttons are pressed/switched on, then the Pan Mode Setting #3 is selected or “pan hold pan” in our example.

Instead of buttons you of course can choose also any other available input, such as e.g. an PWM input or an PPM input, to switch between the four pan mode settings.

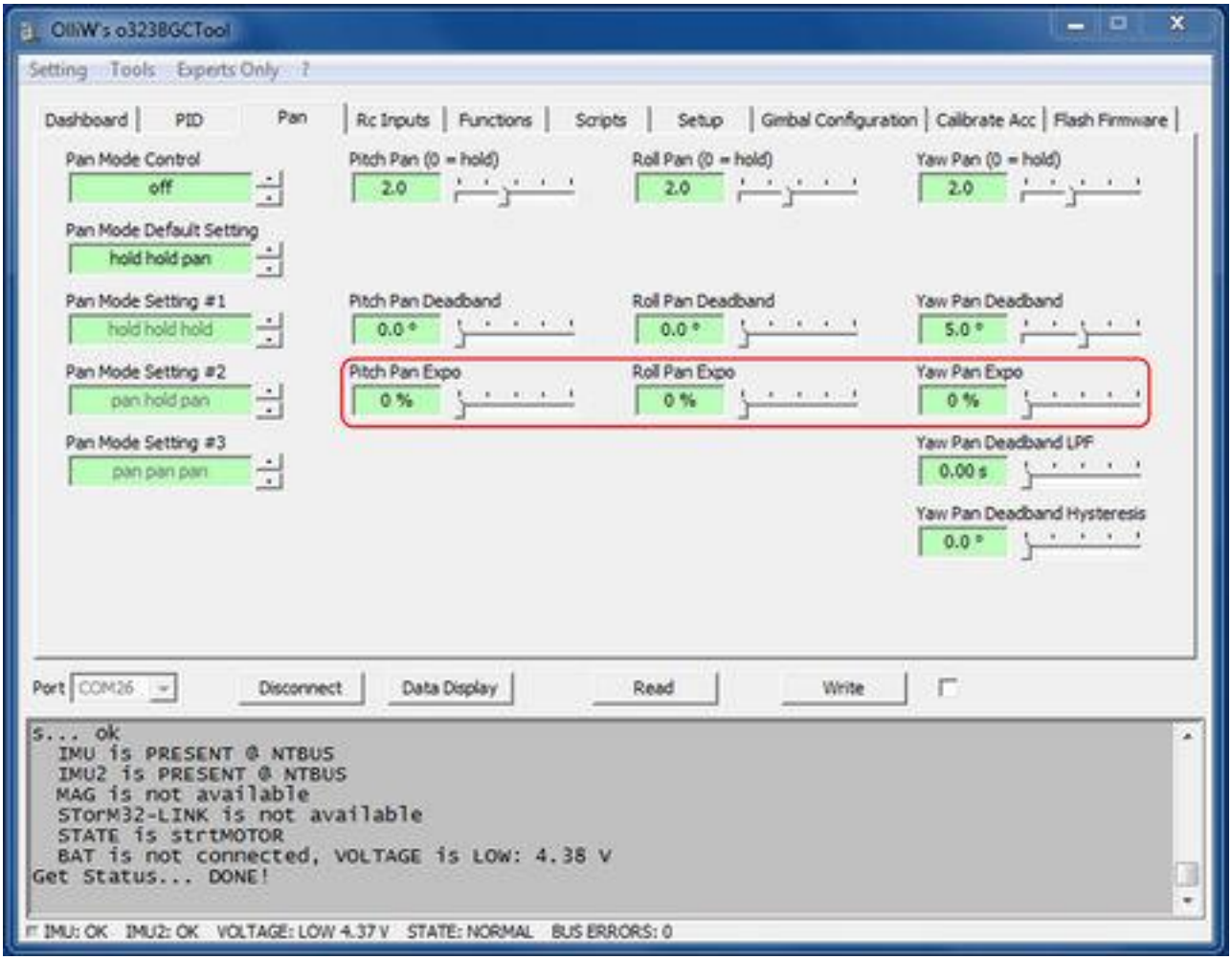


Further Tuning the Pan Mode

The STorM32 controller provides further useful options to adjust the pan behavior, namely the Pan Expo parameters. Again, these settings are effective only when the respective axis is in pan mode, and do not affect the hold mode.

Similar to Pan, the Pan Expo parameter also determines the follow speed, but in contrast to Pan increases the follow speed roughly exponentially with the deviation from the center position. The values can be set from 0% to 100%, where larger values correspond to stronger exponential increase, and 0% disables them. In combination with Pan they provide a large flexibility in adjusting the camera follow behavior upon turns.

The Pan Expo can also be used as a sort of "soft-limiter" to prevent camera lags by too large an angle, e.g., to keep landing skids out of the video or to prevent mechanical damage.



Part 5: Advanced Functions

The STorM32 board provides many user features. Many of them can be activated or controlled by an input signal, and are accessible as so-called functions (see Inputs and Functions). Other features are accessible via specific parameter fields, or the **Tools** menu.

Second IMU Support

One of the major features of the STorM32 controller is its support of a second imu (activated on Copterlab gimbals), providing unmatched performance figures. For a detailed account see the article on Using a 2nd IMU.

Lipo Saver

The parameter Low Voltage Limit in the **Main** tab allows activating a lipo saver. When the battery voltage falls below the specified voltage per cell, the motors are disabled to minimize current draw. This condition is signaled also by the fast blinking green and red led. The board needs to be re-powered or reset to resume normal operation.

Comment: *Lipo batteries may be safely discharged down to 3.0 V/cell, under load.*

Comment: *The lipo cell number is determined automatically by the firmware when a battery is connected to the board. In that calculation the maximum cell voltage is assumed to be 4.2 V. Connecting discharged batteries may lead to*

incorrect cell number determination. Also, using cells with maximal cell voltage very different from 4.2 V may confuse the mechanism.

Comment: *The voltage can be calibrated by adjusting the ADC Calibration parameter in the* ***Tool*** *window.*

Voltage Correction

The parameter Voltage Correction in the **Main** tab allows activation of the voltage correction mechanism.

This mechanism attempts to compensate for the effects of a (slowly) varying battery voltage on the PID controller tuning. It does it by modifying the actual motor power. i.e. Vmax values proportionally to the voltage change. The proportionality factor is given by the parameter Voltage Correction. For instance, with a Voltage Correction of 100%, a drop in the battery voltage by 25% leads to an increase of the actually used Vmax value by 25%. With a Voltage Correction of 50%, a voltage drop by 25% leads to a 12.5% increase in Vmax, and so on.

The mechanism can significantly reduce the dependence of the gimbal performance on the battery voltage, compensating for the slow voltage drop with time when running from a battery. It is not designed to handle short-time power bursts due to e.g. sudden aggressive flight maneuvers. It also cannot accomplish what could be achieved with a truly constant (regulated) voltage.

Comment: *The voltage can be calibrated by adjusting the ADC Calibration parameter in the* ***Expert Tool*** *window.*

Beeps

The parameter Beep with Motors in the **Setup** tab allows to activate the emission of beeps by the motors upon certain events. Two settings are available:

- “Basic” (activated by default)
- “full”

Comment: *The beeps are audible only when the motors are enabled. Hence, beeps are e.g. not heard when the board is powered only by the USB.*

Standby

This function can be activated and its trigger source be specified by the parameter Standby in the [GUI:Functions] tab.

When in normal operation, it will disable the motors. When in standby mode, it will enable the motors and do a quick relevel. It will not go through the full initialization process and e.g. will not do a gyro calibration. Hence the standby function allows it to quickly resume operation. It also means however that if the camera position has been heavily changed or the system has been put into other unfavorable conditions, that the relevel may fail.

Recenter Camera

This function can be activated and its trigger source be specified by the parameter Recenter Camera in the Functions tab.

It will recenter the camera, i.e. will undo any change in camera position due to a manual control signal. For the yaw axis, it does not undo the cumulative center position changes when switching between hold and pan mode.

IR Camera Remote Control

This function can be activated and its trigger source be specified by the parameter IR Camera Control in the Functions tab. The camera type is selected by the parameter Camera Model.

It provides a remote control of the shutter and video functions of the camera (if available). For remote controlling Sony Nex and Canon cameras one can connect an IR led directly to the IR port (see Pins and Connectors). Panasonic cameras can be remote controlled by adding a small piece of hardware, as described here.

In addition, or alternatively, also a NT camera module can be used, see NT Camera.

The camera shutter and video on/off can be adjusted via these fields in the Functions tab:

- IR Camera Setting #1
- IR Camera Setting #2
- Time Interval

Pwm Out

This function needs to be enabled via the parameter Pwm Out Configuration in the Setup tab, and the PWM output format and frequency be specified. Then it can be activated and its trigger source be specified by the parameter Pwm Out Control in the Functions tab.

When activated the PWM signal is output on the pin RC-1. The input channel “Rc-1” yields zero value.

The details of the signal and its behavior can be fine-tuned via these fields the Functions tab:

- Pwm Out Mid
- Pwm Out Min
- Pwm Out Max
- Pwm Out Speed Limit

Scripts

The innovative concept of on-board scripts, invented by the STorM32 project, allows users to modify the controller's behavior and accomplish even complicated tasks. For the details see the article on the STorM32 Scripts.

Some example script files (extension .scr) are available in the firmware folder, which should be sufficiently self-explanatory.

Motion Control

The STorM32 controller provides a rich set of commands for remote controlling the camera. These commands can be sent to the board via any of the serial ports (USB, UART, Bluetooth). A very convenient method to make use of that is motion control scripts. A typical example application would be a pano, i.e. shooting a sequence of photos which are later stitched together to a large panorama view.

Some example motion control script files (extension .mcs) are available in the firmware folder, which should be self explanatory enough to use them.

The following commands are currently (firmware v0.68) possible:

- SetParameter()
- SetAngle()
- DoCamera()
- RecenterCamera()
- SetPwmOut()
- Wait()
- TextOut()

The motion control programming language is in fact just native Perl, so you can do anything you want and it is possible with a full-fledged scripting language. You may use all control structures, define variables and even your own functions, as needed.

UART documentation: http://www.olliw.eu/storm32bgc-wiki/Serial_Communication

Python sample code : https://github.com/olliw42/storm32bgc/tree/master/py-library

C++ sample code : https://github.com/haiha95/storm32bgc_ros_data_sample/blob/main/src/main.cpp

MAVLink

The STorM32 controller supports MAVLink.

For enabling it, open the Tool window, which is accessible via the Tools menu:

- Mavlink Configuration: Set to “emit heartbeat” to activate the emission of a heartbeat message.
- Mavlink System ID: Sets the system ID of the STorM32 controller (default = 71).
- Mavlink Component ID: Sets the component ID of the STorM32 controller (default = 67).

For the supported MAVLink messages, see Serial Communication: Mavlink Communication and Serial Communication: Comments to STorM32 specific Mavlink commands.

Currently, the STorM32's MAVLink capabilities are exploited (to some extend) only by the 32-bit ArduPilot flight controllers. See also

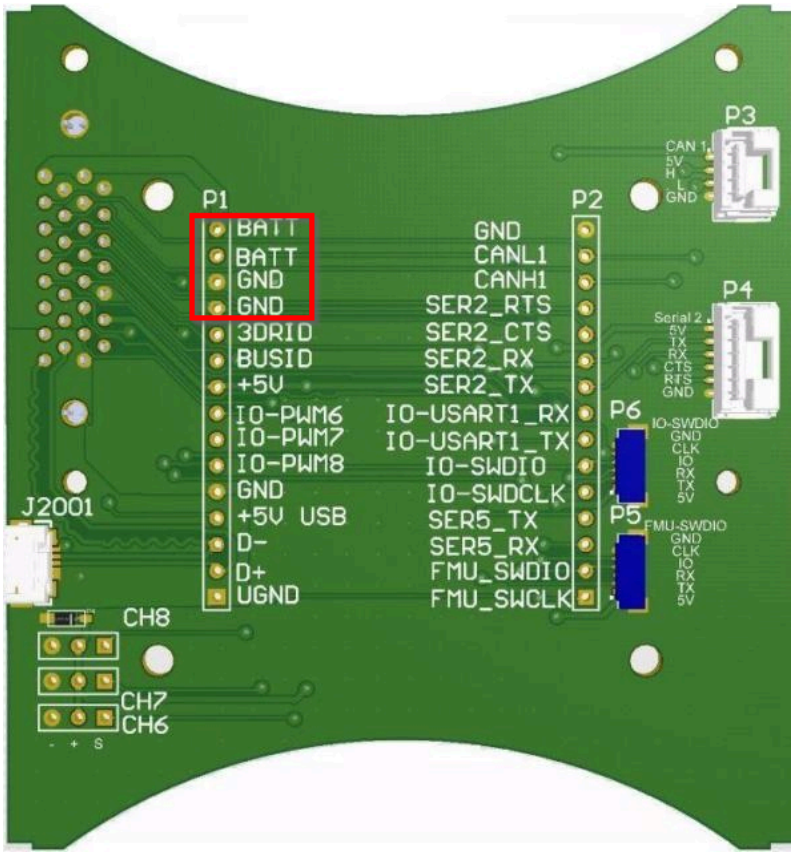
- Storm32 with Pixhawk over serial connection @rcgroups
- ArduPilot > Docs > Optional Hardware > Cameras and Gimbals > SToRM32 Gimbal Controller
- STorM32 BGC - Pixhawk/BetaCopter

Comment: One cannot simultaneously use a BT module and a serial MAVLink connection between the STorM32 and ArduPilot.

APM / Pixhawk 2 gimbal control

You need to buy Accessory Breakout Board

You can power your gimbal by using BATT and GND pins from Breakout PCB Board



Each axis can be controlled by pins PWM6, PWM7, PWM8 and GND

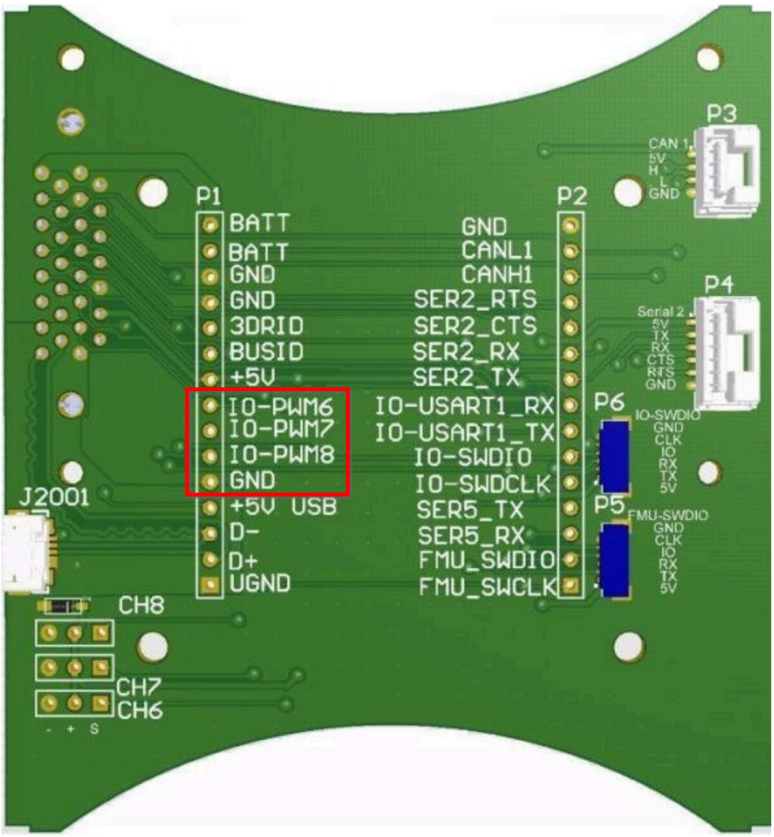


Figure: Accessory Breakout Board PCB (from below)

NT Logger

The STorM32 NT controller supports real-time logging of quite a lot of data onto micro SD cards, by attaching a NT logger module to the NT bus.

For enabling and fine-adjusting the data logging, open the **Expert Tool** window, which is accessible via the **Tools** menu, and set the NT Logging parameter.

For the details see the article NT Data Logger.

Third IMU

The STorM32 NT controller allows attaching a 3rd IMU for a comprehensive vibrations analysis, mainly for evaluating, tuning and optimizing the gimbal's damping system.

The 3rd IMU support is determined by the parameters Imu3 Configuration and Imu3 Orientation, which are found in the **Expert Tool** window, accessible via the **Tools** menu.

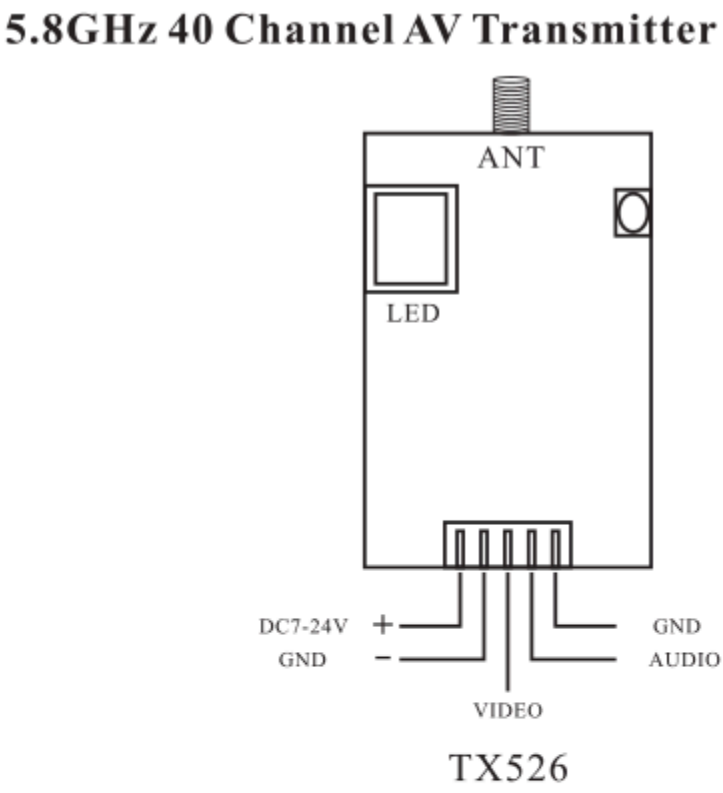
Profile of your gimbal

If you forgot your gimbal profil, ask Copterlab to send you a new one.

Storm32 original user manual

http://www.olliw.eu/storm32bgc-v1-wiki/Manuals_and_Tutorials

Optional video transmitter manual



www.eachie.com

5.8GHz 40 Channel AV Transmitter

Operation Instruction:

1. Powering on the vtx, blue LED display 0, indicating the vtx is powered on, but it is in off status without any transmission.
2. Short-press the button to change channel (CH), digital display will change synchronously. Digital display changes cyclically from 1 to 8.
3. Long-press the button for 2 second until digital display flicker. Then short-press the button to change the frequency band (FR), digital display will change synchronously. Digital display changes cyclically from A、b、E、F、r.
4. Long-press the button for 5 second until digital display flicker. Then short-press the button to change the power, 25mW (one hyphen stands for 25mW), 200mW (two hyphens stand for 200mW), and 600mW (three hyphens stand for 600mW).

Frequency and channel frequency table:

	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR1/Band A	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
FR2/Band b	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
FR3/Band E	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
FR4/Band F	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M
FR5/Band r	5658M	5695M	5732M	5769M	5806M	5843M	5880M	5917M

Thanks for reading