SimpleBGC Software User Manual

Board v. 3.x
Firmware v. 2.43
GUI v. 2.43

CONTENTS

1. Overview

With the help of this manual you will learn how to connect, adjust and calibrate in a special environment the SimpleBGC 32bit controller by BaseCamElectronics.

The main principle of SimpleBGC operation is to compensate undesirable impact on the stabilizable device with brushless motors in holding mode. Motors are managed with a controller board that receives the repositioning data from the gyroscopic sensor. Gyroscopic sensor (IMU) shall be mounted together with the camera to register any undesirable repositioning. Frame sensor can also be used (Frame IMU). Thus data from two sensors will be used by stabilization board simultaneously for more precise system stabilization.

Introduction

Stabilization system controller board and its software were designed and licenced by BaseCam Electronics. SimpleBGC stands for Simple Brushless Gimbal Controller – gimbal controller for brushless motors. It is a high quality stabilization system designed for non-professional as well as professional use.
You can purchase our controller version (http://www.basecamelectronics.ru/store/) or a licensed controller version from our partners (the list of our official partners is available on our web site: http://www.basecamelectronics.ru/wheretobuy/). Different manufacturers alter the basic version of the controller (for example, by adding an integrated Bluetooth component or reducing its size etc.). Please pay attention to the relevant data sheets published on the corresponding manufacturer's web site.
Individual stabilization controllers are available as well as devices with preinstalled controllers (http://www.basecamelectronics.com/readytouse/). There are also ready gimbals without electronic stabilization system. In this case you will need to purchase a controller and install it yourself.
If you decide to assemble a stabilization system yourself, please visit our forum where you can find the necessary information (http://forum.basecamelectronics.com).
In this manual the original 32bit version of SimpleBGC controller is described as well as the multiplatform application for its adjustment.

Basic connections

Basic controller connection scheme within the camera gimbal stabilization system is shown in figure 1:

1. Overview

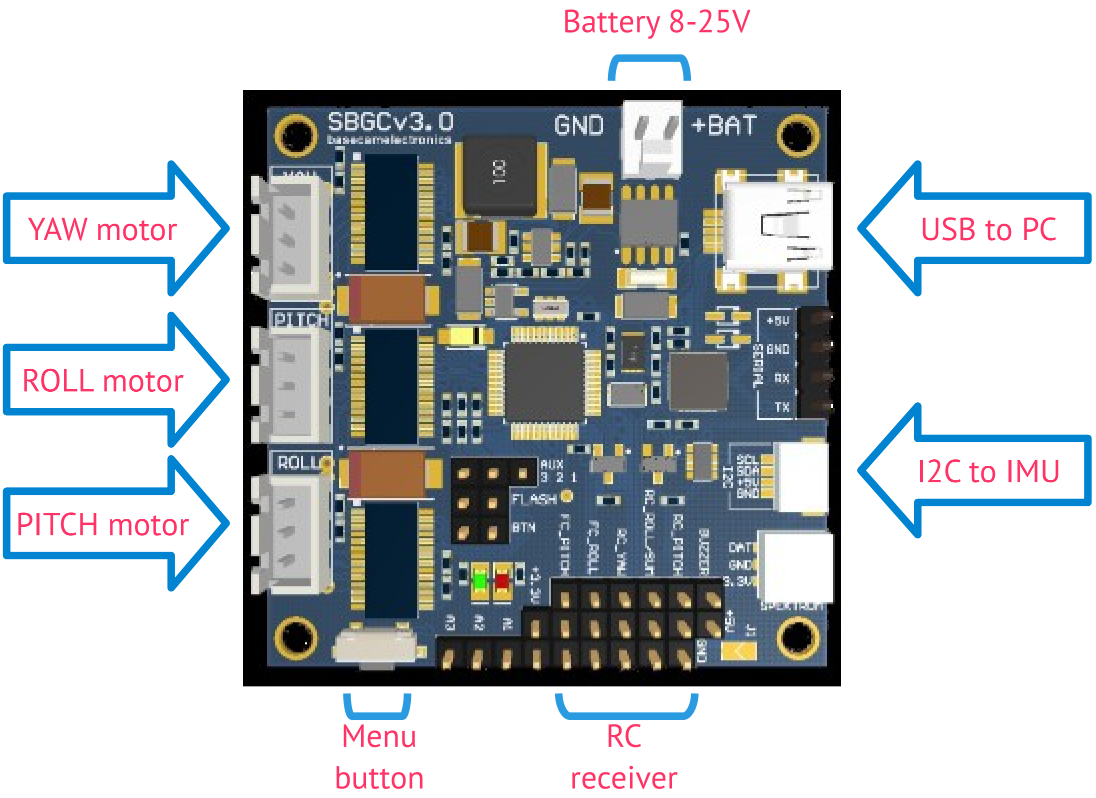


Fig.1 Basic connections

1. **USB port** is used only to connect SimpleBGC 32bit stabilization board to PC.
2. Gyroscopic **sensor** (IMU) is connected to I2C slot.
3. Each axis **motor** is connected to the corresponding motor connection.

   NOTE: It is advisable to pull each motor cable through a ferrite ring and make at least one loop around the ring to avoid high frequency interference induction interrupting the sensor and other electronic devices.

4. The stabilization controller SimpleBGC 32bit is equipped with a power cable to enable a **battery connection**. Its bared wires should be soldered to the corresponding connector on your battery. Polarity should be observed; otherwise both the controller and the battery can be damaged. The other end of the cable should be inserted into **+BAT slot.**

Never lock battery terminals; it may lead to an explosion!

NOTE: Batteries of 8 to 25V are acceptable. If you use a lithium-polymer battery (LiPo), 3S to 5S inclusive are acceptable. S stands for the quantity of cells in a battery. Each cell can be charged up to 4.2V. Consequently, a fully charged 3S LiPo is equal to 12.6V and 5S LiPo is equal to 21V.

A detailed description of a controller connection within a complete stabilization system can be found in the detailed connection scheme.

GUI installation

First you need to download the latest version of the GUI application from our web site (http://www.basecamelectronics.com/downloads/32bit/) and unpack it in any folder. To start the application you need to have Java Runtime Environment by Oracle in your system: http://www.java.com

GUI for **Windows**:

- Simply run SimpleBGC_GUI.exe

GUI for **MAC OS**:

- Run SimpleBGC_GUI.jar

> **ATTENTION:** GUI uses a virtual COM port. Create a lock file to access it.:
>
> 1. Start terminal (navigate to /Applications/Utilities and double click on Terminal)
> 2. Make folder "/var/lock" by command: sudo mkdir /var/lock
> 3. Change permissions by command: sudo chmod 777 /var/lock
> 4. Allow to run non-signed applications in System Preferences > Security & Privacy > General > Allow Applications downloaded from: Anywhere

GUI for **LINUX**:

- Run run.sh.

Connection to computer

A connection between a controller and a computer can be established via a Mini- or Micro-USB (depending on the version of the controller).

You will need to install a driver the first time you establish a connection. If the driver is not installed automatically, you can download it – follow the link
http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx

For "Tiny" version the driver for Windows can be downloaded here
http://www.st.com/web/en/catalog/tools/PF257938

After you have installed the driver and connected the controller, you will see a new virtual COM port in the system. Its name should be entered in the graphic interface window (hereinafter – GUI) upon connection.

You can connect the controller to your computer and supply power from a battery simultaneously. But be careful and **observe polarity of battery terminals** because when a USB connection is established, the in-built reverse polarity protection is off (some versions are not equipped with such protection).

Wireless connection

To connect you can also use a wireless connection through Bluetooth-to-Serial converter and USB-Bluetooth adapter from PC side. For example: HC-05, HC-06, Sparkfun BlueSMiRF converters and other.

The converter should have at least 4 outputs: Gnd, +5V, Rx, Tx. The controller is equipped with a corresponding slot marked with UART.

Bluetooth module connection is described in Appendix B.

> **NOTE:** Bluetooth module should be set for baud=115200 and parity=None or Even. Under None the board can be connected to through GUI. Under Even not only the controller can be adjusted, but also its firmware can be updated

through Bluetooth. To change Bluetooth module settings, see its manual.

Running the application

1. Attach USB cable.
2. Run GUI, select COM port from the list in the left corner of the main window and press **Connect**.
3. When the connection to the board is established all profiles will be read and downloaded, GUI will display the current profile settings. You can read the board settings again any time by pressing on the **READ** button.
4. **Make sure to have installed the latest version of firmware**. To check: open "Upgrade" tab and press "Check update". Update if a new version is available. See section "Firmware Update" for more detailed information.
5. After you have finished editing the parameters, press **WRITE** to save them to the permanent memory of the controller (EEPROM). Only the current selected profile will be saved. To restore the factory settings go to **"Board" – "Reset to defaults"**. All the parameters of the current profile will be set to defaults except for general settings and calibration data. In order to erase the settings of ALL profiles, general settings and calibration data go to menu **"Board" – "Erase EEPROM"**.
6. To switch over to the settings of another profile, choose the desired profile from the list in the upper right corner. It is not required to read the parameters by pressing **READ**. You can save different settings in 5 different profiles. Profiles can be switched over through GUI or by menu button. Please note that some settings are shared by all profiles, the settings concerning hardware component configuration in particular as well as sensor orientation, RC inputs, outputs to motors and some other. You can assign random names to profiles. They will be saved on the board and will remain unchanged when you connect to GUI from a different computer.

2. Step-by-step setup sequence

1. Adjusting the mechanics

Mount the camera on the tray and balance the gimbal in all three axes. Stabilization quality strongly depends on balance quality. To check your balance, pick your turned off gimbal in hands. Make fast motions along all axes, try to catch resonance point and swing the gimbal. If it is hard to do - gimbal is balanced correctly.

> **NOTE:** Good balance and low friction can scale down power consumption and keep good quality of stabilization.

If you rewound motors by yourself, it's recommended to check winding. Remove motors from gimbal, connect them to controller and set parameters P=0, I=0.1, D=0 for each axis and set enough POWER. Connect main power supply. Motors should spin smoothly, while rolling the sensor. Little jitter is normal due to magnetic force between rotor and stator ("cogging" effect).

Pay great attention to sensor installation. Its axes must be parallel with motor axes. Pay attention to mechanical links. They must be a VERY RIGID and backlash-free. Sensor provides feedback data for stabilization, and even any little freedom or flexibility will cause delays and low-frequency resonances. This can complicate setting of PID, and cause unstable work in real conditions (frame vibrations, wind, etc)

2. Calibrating the sensor

Gyro is calibrated every time you turn the controller on, and it takes about 4 seconds. Try to immobilize sensor (camera) as hard as you can in first seconds after powering on, while signal LED is blinking. After powering on you have 3 seconds to freeze gimbal before calibration starts.

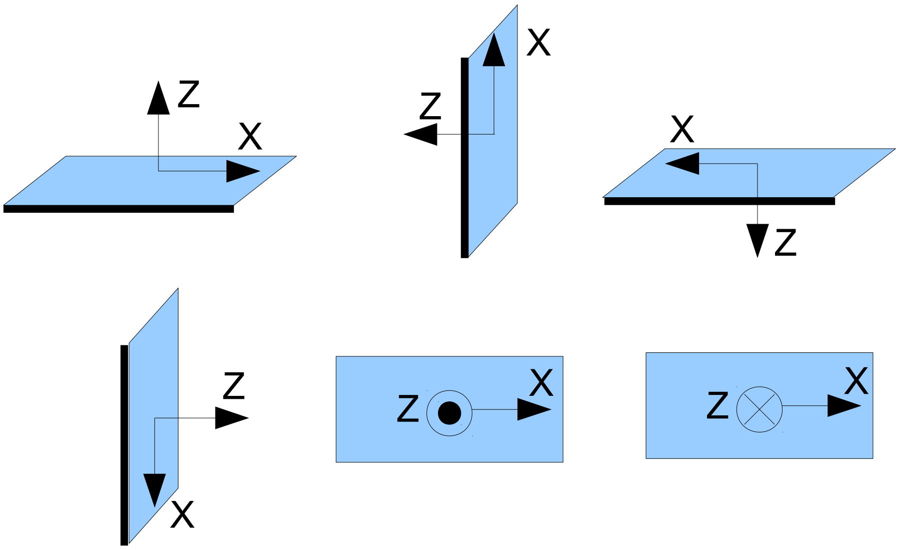
If you activated option "Skip gyro calibration at startup", gyro is not calibrated every time and controller start working immediately after powering up. Be careful and recalibrate gyro manually, if you will notice something wrong with IMU angles.

Calibrating Accelerometer

You must perform ACC calibration only once, but it's recommended to recalibrate it from time to time or when the temperature significantly changes.

- **Simple calibration mode:** set the sensor horizontally, and press CALIB.ACC in the GUI (or menu button, if it's assigned). LED will blink for 3 seconds. Try not to move sensor during calibration. At this step no matter how camera is leveled. You are calibrating the sensor, not the camera!
- **Advanced mode (recommended):** perform calibration in simple mode as above. Then turn sensor in order that each side of sensor looks up (6 positions at all, including base one). Fix the sensor in each position, press **CALIB.ACC** button in the GUI, and wait about 3-4 seconds, while LED if flashing. The order does not matter, but the base position always goes first (because the simple calibration cancels a result of advanced calibration). *You have not to press WRITE button, calibration data is written automatically after each step.*

> **NOTE:** Precise accelerometer calibration is a very important for horizon holding during dynamic flying or YAW rotation.



4. Tuning basic settings

- Connect the main power supply.
- Set **POWER** according to the motor configuration (see recommendations above)
- Auto-detect number of poles and motors direction. Do not proceed to next step until proper direction will be detected!
- Run auto-tuning for PID-controller, using default settings first time.
- Adjust PID controller settings, if required. To check stabilization quality, use peak indicator in the control panel (shown by the blue traces and blue numbers). Incline the frame by small angles and try to minimize peak values by increasing P, I and D to its maximum. You may use gyro data from "Monitoring" tab to estimate stabilization quality, too.
  Better to tune PID with the "Follow Mode" turned OFF for all axes.

  Suggested algorythm for manual PID tuning:

  1. Set I=0.01, P=10, D=10 for all axes. Gimbal should be stable at this moment. If not, decrease P and D a bit. Than start to tune each axis sequentially:
  2. Gradually increase P until motor starts oscillate (you may knock camera and see on the gyro graph, how fast oscillation decays). Increase D a little – it should damp oscillations, and decay time decreases. The lower is decay time, the better.
  3. Repeat step 2 until D reaches its maximum, when high-frequency vibration appears (you may feel it by hands and see noisy line on the gyro graph). Current P and D values are maximum for your setup, decrease them a little and go to step 4
  4. Increase I until low-frequency oscillation starts. Decrease I a little to keep gimbal stable. Now you found a maximum for all PID values for selected axis. Repeat from step 1 for other axes.
  5. When all axes are tuned in static, try to move gimbal's frame, emulating a real work. You may notice that cross-influence of axes may make gimbal not stable. In this case, decrease a little

PID values from its maximum for axes that looses

The result of good tuning – stabilization error is less than 1 degree when you slightly rock a gimbal's frame.

Further steps to improve the precision of stabilization:

- Connect and calibrate external flight controller (see Advanced Settings, External FC Gain)
- Connect, setup and calibrate second (frame) IMU (see Second IMU sensor)

5. Connecting and configuring RC

- Connect one of the free receiver's channels to RC_PITCH input, preserving right polarity

In the RC Settings tab:

- Set **SORCE**=PWM
- Assign RC_PITCH input to PITCH axis
- Leave all other axes and CMD as "no input"
- For PITCH axis, set **MIN.ANGLE**=-90, **MAX.ANGLE**=90, **ANGLE MODE**=checked, **LPF**=5, **SPEED**=10 (not used in angle mode)
- Connect the battery to the main controller and receiver, and check that RC_PITCH input receives data in the "Monitoring" tab (slider should be blue filled and reflects to stick movement)

Now you can control the camera from your RC transmitter, from -90 to 90 degrees. If you are not satisfied with the speed of movement, adjust the **I-term** setting for PITCH in the "Basic" tab.

Try the SPEED mode and feel difference with the ANGLE mode.

Connect and tune remaining axes the same way, as required.

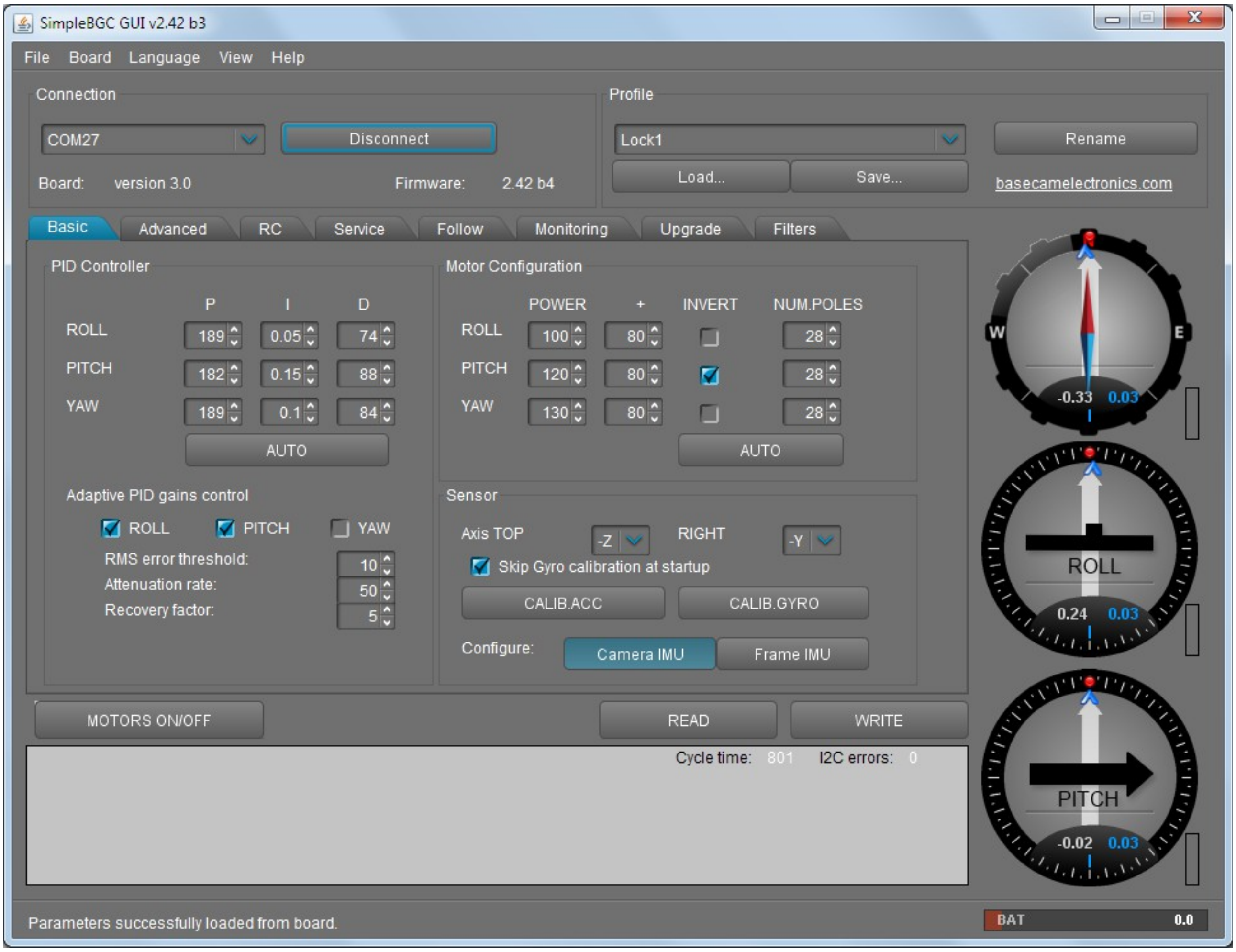
6. Testing gimbal in real conditions

Connect controller to the GUI and turn ON multirotor motors, holding it above your head. Check the vibrations on the camera by using "Monitoring" tab / ACC raw data. Try to decrease the level of vibrations using soft dampers.

> **NOTE:** Brushless motors versus traditional servos provide faster reaction, but less torque. That's why it's hard for them to fight against wind and air flows from props. If you are developing multirotor frame by yourself, try to avoid this influences (for example, lengthen arms a bit, or tilt motors away from center or place camera above props in case of H-frame). Also bear in mind, when copter moves with high speed, an air flow is deflected and can affect the gimbal.

3. GUI overview

GUI Structure



Connection: COM27

Board: version 3.0 Firmware: 2.42 b4

Profile: Lock1

PID Controller

	P	I	D
ROLL	189	0.05	74
PITCH	182	0.15	88
YAW	189	0.1	84

Adaptive PID gains control

☑ ROLL ☑ PITCH □ YAW

RMS error threshold: 10

Attenuation rate: 50

Recovery factor: 5

Motor Configuration

	POWER	+	INVERT	NUM.POLES
ROLL	100	80	□	28
PITCH	120	80	☑	28
YAW	130	80	□	28

Sensor

Axis TOP -Z RIGHT -Y

☑ Skip Gyro calibration at startup

Configure: Camera IMU Frame IMU

Cycle time: 801 I2C errors: 0

Parameters successfully loaded from board.

BAT 0.0

The GUI contains different functional blocks:

1. Configuration block in the central part of the window, organized by 'tab':
   - Basic – Basic gimbal stabilization settings. Adjusting these settings is usually adequate to achieve good camera stabilization.
   - Advanced — More precise tuning options.
   - RC – settings to control the gimbal roll/pitch/yaw orientation with RC inputs.
   - Service – Specify the behavior of the MENU button (located on the controller board or mounted externally) and tune the battery monitoring service.
   - Follow – settings related to special mode of the camera control when it follows frame.
   - Monitoring — real-time sensor data monitoring. This screen is extremely helpful in tuning your gimbal performance. Firmware Update — Firmware and GUI software versions and update options.
   - Upgrade – lets you to check the fresh version of firmware and upgrade if necessary.

- Filters – settings to setup digital filters for PID controller

2. Connection — COM-port selection and connection status.
3. Profile — Profile selection, loading, re-naming, and saving.
4. Control Panel — graphic visualization of gimbal orientation angles in three axes.
   - *Black arrows are displaying the angles, blue arrows are a 10x time magnification to provide higher precision. Red marks show target angles that gimbal should keep.*
   - *Thin blue lines shows the maximum (peak) deflection from the central, neutral point.*
   - *Blue digits show* peak *deflection amplitude. Using these numbers,* stabilization *quality* can *be estimated.*
   - *Vertical red bars to the right of the scales show actual power level, from 0 to 100%.*
   - *Gray arrows shows the angle of a stator of each motor, if known.*
5. READ, WRITE buttons are used to transfer setting from/to board.
6. MOTORS ON/OFF button is used to toggle motors state.
7. At the bottom of the screen, tips, status or error messages (in red color) are displayed . Overall cycle time and I2C error count is also displayed.
8. Battery voltage indicator with warning sector.

Board menu

This menu encapsulates options to Read/Write settings (duplicating READ, WRITE buttons), to calibrate sensors, to reset parameters to their default values, or completely reset board by erasing EEPROM.

Language menu

The GUI starts in the English version of the user interface. To change the interface language, choose one in the 'language' menu and restart the program.

View menu

You can change a visual theme from the “View” menu. For example, when using GUI outdoor, better to switch to one of the high-contrast themes.

Further in this manual each tab is described in details. At the end of this manual, you can find step-by-step tuning recommendations.

4. Basic Settings

PID and Motor settings

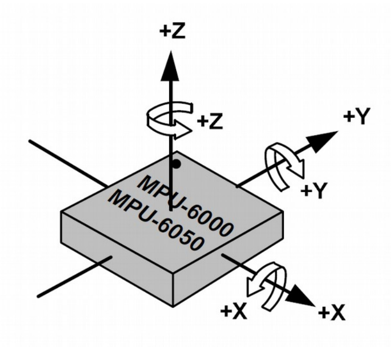
- **P,I,D – PID regulation parameters for all axes. .**
    - P – describes the power of disturbance response. Higher values means a stronger response reaction to external disturbance. Raise this value until the stabilization quality of fast disturbances will be adequate. If the "P" value is too high, oscillations of the axis will start to be present. These oscillations will get worse if there are vibrations that reach the IMU sensor board. *If oscillations occur, raise the "D" parameter by 1 or 2 units, and then try to raise the "P" value again.*
    - D – The "D" value reduces the reaction speed. This value helps to remove low-frequency oscillations. A "D" value that is too high can cause high-frequency oscillations, particularly, when the IMU sensor is exposed to vibrations. In special cases, it may be filtered out by digital filters (see below).
    - I – The "I" value changes the speed at which the gimbal moves to incoming RC commands and to move the gimbal back to neutral. *Low values result in a slow and smooth reaction to RC commands and to getting back to neutral. Increase this value to speed up the movement*
- **POWER** – maximum voltage supplied to the motors (0 - 255, where 255 means full battery voltage). Choose this parameter according to your motor characteristics. *Basic tuning:*
    - **Motors should not get too hot!** Motor temperatures of over 80C will cause permanent damage to motor magnets.
    - A Power value that is too low will not provide enough force for the motor to move the gimbal and stabilize the camera adequately. A low power value will be most noticeable in windy conditions, when the gimbal is not well balanced, or if the gimbal suffers from mechanical friction. Slowly lower the Power parameter to find its optimal value. Find the lowest value that still provides good stabilization and adequate holding torque.
    - Raising the power equals raising the "P" value of PID settings. If you raise the POWER value, you should re-tune your PID values as well.
- **"+"** - **Additional power** that will be add to the main power in case of big error (caused by missed steps). It helps to return camera to the normal position. If main power + additional power is greater than 255, the result is limited to 255.
- **INVERT** – reverse motor rotation direction. It's extremely important to choose the correct motor rotation direction to not damage your gimbal. To determine the correct direction, set the P, I, and D values to 0 and the POWER values to 80 (or higher if your motors don't produce enough force to hold/move the camera). Level the camera tray horizontally and click the AUTO button in the "Motor configuration" settings. The gimbal will make small movement to determine correct motor rotation direction. Wait for the calibration procedure to complete. Then, re-set your PID values and tune your Power values.
- **NUM.POLES** – Number of motor poles. This value needs to be equal to the number of magnets in your motor's bell. During the "auto" calibration process described above, this value is automatically detected. However, this value is sometimes not correctly determined during the "auto" calibration process and will need to be verified and possibly corrected manually. Most brushless gimbal motors are built with 14 poles (or magnets) and utilize a DLRK winding scheme. Count your motor magnets and enter this value if the value is not correct in the GUI.

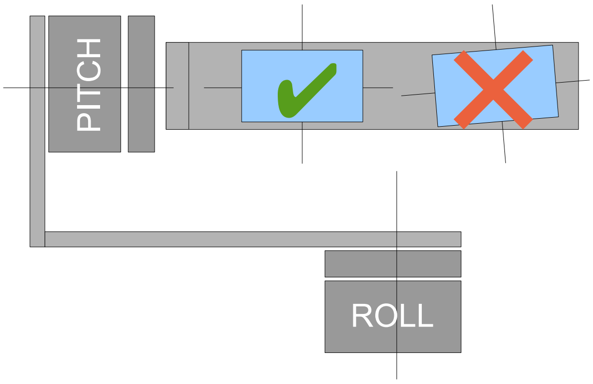
Main IMU sensor

Note: Before tuning your controller, install the camera into the gimbal firmly and ensure your gimbal's center of gravity is leveled as much as possible.

Specify your IMU sensor board's orientation and position on the gimbal . For a standard IMU sensor installation, look at the gimbal from behind just like the camera will view out from the gimbal. Viewing the gimbal in this way, the UP and Right direction will match the Z and X axis. You can place the IMU sensor in any direction, keeping its sides always parallel to the motor axis (be very accurate here, it is a very important to precisely align the sensor and mount it firmly). Configure your IMU orientation in the GUI. The correct configuration should result in the following:

- Camera pitches forward – the PITCH arrow spins clockwise in the GUI.
- Camera rolls right - ROLL arrow spins clockwise in the GUI.
- Camera yaws clockwise - YAW arrow spins clockwise.

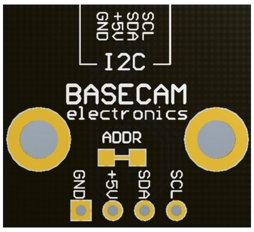




- **Skip Gyro calibration at startup** - With this option, the board starts working immediately after powering it on, using the saved calibration data from last gyroscope calibration call. However, stored calibration data may become inaccurate over time or during temperature changes. We recommend you to re-calibrate your gyro from time to time to ensure the best performance.

Second IMU sensor

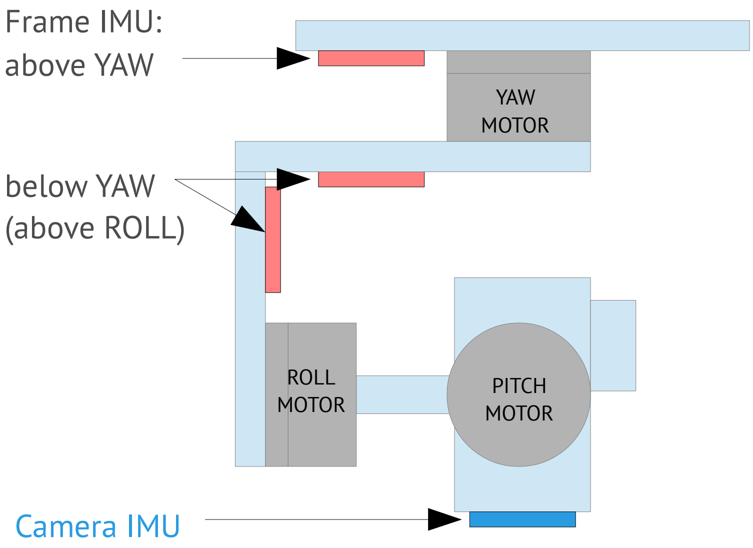
There is an option to install the second IMU sensor on the gimbal's frame. The advantage is more precise stabilization (you may use lower PID's to get the same quality) and knowing of frame tilting, that greatly helps for 3-axis system to extend the range of working angles.



Second IMU should be connected to the same I2C bus as main (in parallel). Sensors should have different I2C-address (Main IMU – 0x68, Frame IMU – 0x69). On the Basecam IMU, address 0x69 may be set by cutting the ADDR bridge, located on the back side of the sensor.

Mounting the Frame IMU

There are two options where to place the second IMU: below YAW motor and above it. In case of 2-axis stabilization, there is only one option – above ROLL motor.



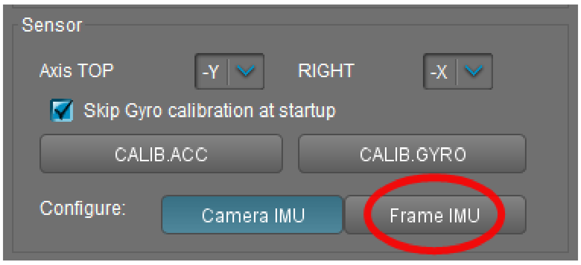
If the sensor is placed **above YAW** motor, it helps to stabilize ROLL, PITCH and YAW motors. But the system becomes less stable during long work (because the frame heading, estimated from the second IMU, may drift with time and auto-correction may not work in all cases).

If the sensor is placed **below YAW** motor, it does not help YAW axis stabilization, but works more reliable. In this position, there is additional option you can choose from: "**Below YAW + PID source**". It means, that if Frame IMU is mounted below YAW motor, it can be used as data source for PID controller. In some cases this can give better result, than the main IMU, because mechanical system "IMU-Motor" becomes more stiff when its length is shorter, and its closed-loop operation becomes more stable.

Like the main (camera) IMU, the frame IMU may be mounted in any orientation, keeping its axis parallel with motor's axis.

Configuring the frame IMU

To configure the frame IMU, first of all set its location in the "**Advanced**" tab, "Sensor" area. Write settings to the board and go to the "Basic" tab. Press the button "**Frame IMU**":



Axis TOP -Y RIGHT -X

☑ Skip Gyro calibration at startup

Configure:

If the second IMU is connected properly, this button becomes active. It means that all IMU settings now affect on the frame IMU. Change sensor orientation (axis TOP, RIGHT) and write setting to the board, if necessary (board will be restarted). After restart, calibrate the accelerometer and gyroscope like you did it for the main IMU. For the accelerometer, you can do simple calibration or extended 6-point calibration.

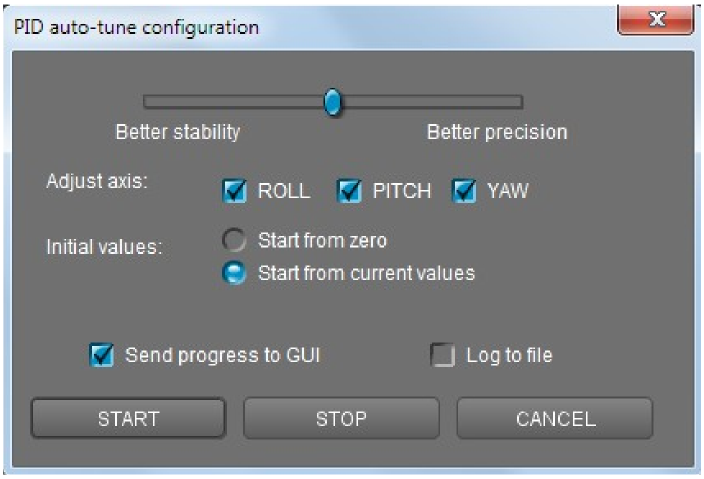
You may notice that right panels with arrows are displaying now angles not for the main, but for the frame IMU. Also, in the “Monitoring” tab, accelerometer's and gyroscope's data go for the frame IMU. It helps to properly configure an orientation of the sensor and check its calibration.

5. PID auto-tuning

This feature will be helpful for the beginners, who experienced difficulties with PID tuning.

Before you start automatic tuning, its very important to properly configure hardware of your system: motor outputs, “Power”, “Inverse” and “Number of poles” (latest 2 setting may be detected automatically, as described in the user manual). Also, main IMU position should be configured and accelerometer and gyroscope should be calibrated.

Plug-in a battery, connect board to the GUI and press “Auto” button in the PID parameters section. You will see dialog window, where you can setup auto-tuning process:


PID auto-tune configuration

Adjust axis: ☑ ROLL ☑ PITCH ☑ YAW

Initial values: ○ Start from zero ● Start from current values

☑ Send progress to GUI ☐ Log to file

Slider at the top defines the target of tuning. If its close to “Better precision”, it will try to achieve maximum gain and keep it. If close to “better stability”, it will find maximum gain and than decrease it by 30-50% to make system more stable.

You may chose which axis to tune. Best result may be reached only if to tune each axis separately. But for the first run, you can tune all axis at the same time.

If you want to use your current settings as start point, select “Start from current values” Otherwise, values will be set to zero in the beginning.

Select “Send progress to GUI” checkbox to see how PID values change in real-time during tuning process.

Select “Log to file” to write PID values together with some debug variables to the file “auto_pid_log.csv”. It may be analyzed later to better understand system behavior. There are number of tools to plot data from log files, for example http://kst-plot.kde.org

How does it work?

Tuning process does simple job: it gradually increases P,I,D values until system enters in self-excitation state. It means maximum possible gains are reached. Than it rolls back values a bit and repeats the same iteration 2 times. Averaged “good” values are stored as PID settings.

During process, you should firmly hold your gimbal in hands. You can place it on the support but check that it provides strong hold, not less than your hands.

After about minute of work, you can see that PID values have grown big enough and camera is stabilized. Now you can slightly tilt handles in all directions to emulate real-usage conditions. Find point where self-excitation is maximum, and continue tuning system in this point (a “worse case” position).

It is normal that gimbal can start to vibrate when PID values come close to theirs maximum. If any motor looses sync due to strong oscillations, you can help to restore it by hand without interrupting the process.

In some cases, you can get better result (i.e. higher PID gains) if you remove high-frequency resonances before starting automatic tuning. See section “Digital filters” for more details.

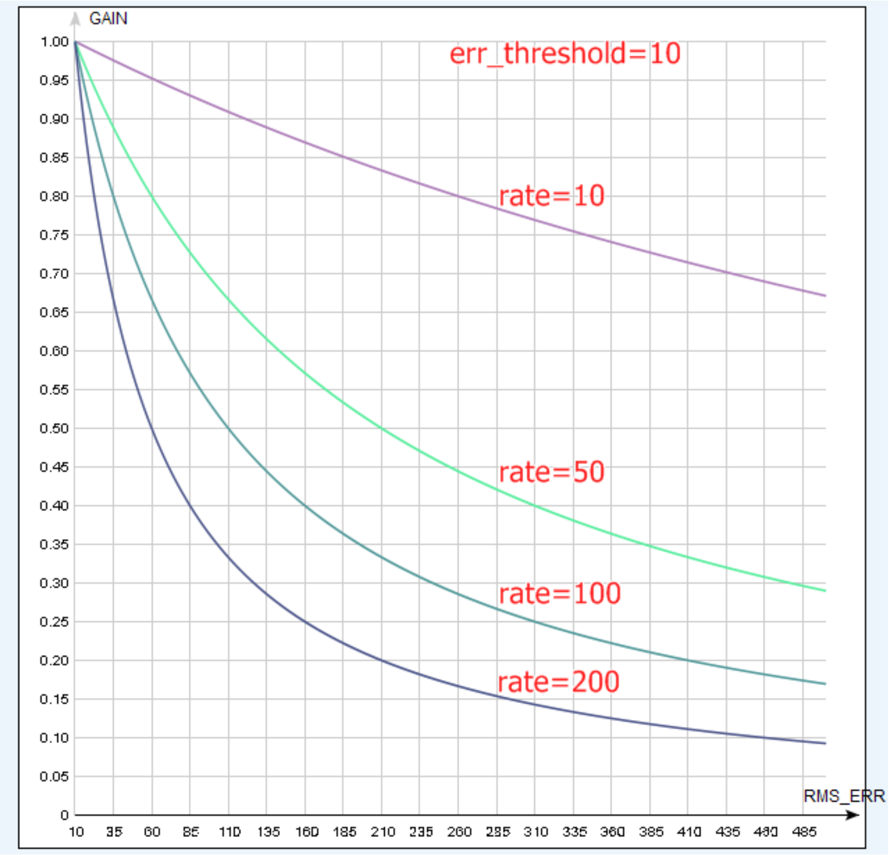
LED is flashing during tuning process. When process will finish its job, LED will light ON and new PID settings will be transferred to the GUI.

There is also a corresponding menu command, that can start PID auto-tuning without connection to the PC.

Adaptive control of PID gains

This settings group lets to adaptively decrease PID gains, when the system becomes unstable due to high PID gains. For example, system may be tuned very well in certain position, but it may become completely unstable in different position. Self-excitation may cause strong vibration, that may negatively affect gimbal construction and may be even dangerous for the camera. Also, when vibration accidentally comes, it may brake whole footage. Possible workaround is to use adaptive PID control.

- **RMS error threshold**, 0..255 - RMS (root mean square) error state variable effectively shows the level of vibrations. When it exceeds this threshold, adaptive PID algorithm comes into action. Recommended value is 10..15.
- **Attenuation rate**, 0..255 - the more this value, the more PID gains are decreased. Choose this value big enough to quiet system quickly. Effect of different rates is shown on the picture:



- **Recovery factor**, 0..10 - defines, how fast PID gains are recovered back when the system becomes stable. Too low value may increase a chance that vibration comes back in short time. Too high value may cause worse operation (because lowered PID values are kept longer). Recommended value is 5..6

6. RC Settings

SimpleBGC board provides very flexible configuration of remote controller. It supports up to 5 digital inputs, including one that supports most popular serial protocols, and 3 analog inputs. It can also output RC signal in pass-through mode or by Serial API commands. Full RC routing diagram can be found in the Appendix C of this manual.

- **RC Input Mapping** – here you can assign hardware RC inputs to target control channels. There are 5 hardware digital inputs provided on the board for RC Radio control connections, and 3 analog inputs for connecting joystick. Each input you can assign to control any of three channels, one for each axes, and one command channel. If control for an axis is not needed, leave the option at "no input".
- **RC_ROLL pin mode** – allows to configure several formats of incoming signal for RC_ROLL pin:
  - **Normal** – incoming signal is in the PWM format, that most RC-receivers generally outputs
  - **Sum-PPM** - some receivers may have this signal output. It is a PWM format modification, in which every channel transmits sequentially through one cable. In this case you do not need to connect other channels (read your receiver's user manual to check if it has SumPPM out).
  - **Futaba s-bus** – receivers made by Futaba may transmit data in special digital format, up to 16 channels by one wire. Connect it to RC_ROLL pin.
  - **Spektrum** – another digital multi-channel protocol, that is used to communicate Spektrum's satellite modules with the main module, and in its clones. There is a dedicated socket on the board (marked Spektrum) that matches standard connector.
    Starting from firmware ver. 2.43b7, **you can bind** a satellite (remote) receiver, connected to the "spektrum" port, directly from the SimpleBGC board. It will be bound as stand-alone (master) unit. To start binding, assign action "Bind RC receiver" to the hardware menu button and execute this action, or execute the same action from the "Board – Execute command" menu in the GUI. You can select any of 4 different modes prior to start binding, in the "RC" – "Other settings" tab:
    - DSM2/11ms
    - DSM2/22ms
    - DSMX/11ms
    - DSMX/22ms

    Choose a mode that a combination of your transmitter and receiver supports (10- or 11-bit modification does not matter at this moment) . Switch to Auto-detection mode after binding is done. If channels are read incorrectly, select 10bit or 11bit modification manually.
  - **SBGC Serial API 2nd UART** – in this mode, RC_ROLL input can handle Serial API commands. It lets to expand the board functionality by connecting external devices, implementing SBGC Serial API protocol. If *RC_YAW* pin is not occupied, it acts as **TX** pin of this UART, allowing to use bi-directional communication. If *RC_YAW* pin is occupied, only **RX** functionality is possible (in other words, external device can send commands to the board, but can't read answers).
    Port settings: 115200 baud, 8N1 or 8E1 - 1 stop bit, 8 data bits, parity 'none' or 'even' (auto-detected after several incoming commands).
- For each control targets you can choose appropriate hardware input from the drop-down list.
  - **RC_ROLL, RC_PITCH, RC_YAW, FC_ROLL, FC_PITCH** – hardware inputs on board that accept signal in the PWM (Pulse Width Modulation) format (excepting RC_ROLL, see above). Most RC receivers output this signal type.

- ○ **ADC1, ADC2, ADC3** — dedicated analog inputs, marked on the board as A1, A2, A3 and accepts analog signal in range from 0 to +3.3 volts. For example, joystick variable resistor provides such signal. Connect A1..A3 to the center contact of variable resistor, +3.3V and GND to side contacts. See Connection Diagram for more info.
- ○ **VIRT_CH_XX** – In case of RC_ROLL pin mode is set to multi-channel signal format, you can chose one of the virtual channels.
- ○ **API_VIRT_CH_XX** – Channels that may be set by Serial API command.

- **Control targets:**
  - ○ **ROLL, PITCH, YAW** - controls the position of the camera
  - ○ **CMD** allows to execute some actions. You can configure 2- or 3-position switch on your RC for specified channel, and assign it to CMD channel. Its range is splitted into 3 sections : LOW ,MID ,HIGH. When changing the position of your RC-switch, signal jumps from one section to another, and assigned command is executed. The full list of available commands is described in the section "**MENU BUTTON"** of this manual.
  - ○ **FC_ROLL, FC_PITCH** – is used to mark any of PWM inputs to be a signal from the external flight controller. See "External FC gain" section for details.
- **Mix channels** - you can mix 2 inputs together before applying to any of ROLL, PITCH or YAW axis. It lets to control the camera from the 2 sources (joystick and RC for example). You can adjust the proportion of the mix from 0 to 100%.
- **ANGLE MODE** — RC stick will control the camera angle directly. The full RC range will cause a camera to go from min to max angles, as specified above. If RC stick doesn't move, camera stands still. The speed of rotation depends on the "SPEED" setting and the acceleration limiter setting.
- **SPEED MODE** — RC stick will control the rotation speed. If stick is centered - camera stands still, if stick is deflected, camera starts to rotate, but does not exceed min-max range. Speed is slightly decreased near min-max borders. Speed of rotation is proportional to stick angle and the **SPEED** setting. RC control inversion is allowed in both of control modes.
- **INVERSE –** Set this checkbox to reverse direction of rotation relative to stick movement.
- **MIN.ANGLE, MAX.ANGLE** – range of the angles controlled from RC or in Follow mode. To inverse the control, set higher value first, and lower value second. For example, if you want to configure a camera to go from leveled position to down position, set 0-90 (or 90-0 to inverse).
  To disable constraints in SPEED mode, set MIN.ANGLE=MAX.ANGLE=0.
- **LPF** – Signal low-pass filtering. The higher is value, the smoother is reaction to the stick commands. This filter cuts fast stick movements, but adds some delay as a drawback.
- **INIT.ANGLE** – if RC control is not configured for any axis, (or there is no signal on the source), system will keep initial angle, specified in this field.
- **RC Sub-Trim** – allows to correct transmitter inaccuracy.
  - ○ **ROLL, PITCH, YAW trim** – central point trimming. Central point here is PWM 1500. It's better to trim it in transmitter. But in case of it is not possible (when using joystick, for example), you can use AUTO function in the GUI. Just place stick in center, and press AUTO button. Actual data becomes new center point. Press WRITE button to apply settings.
  - ○ **Dead band** — adjusts a dead band around neutral point. There's no control while RC signal is inside this range. This feature works only in SPEED mode, and helps to achieve better control by eliminating jitters of stick around neutral point.

  - **Expo curve** – adjusts the curvature of an exponential function, that allows to get precise control from RC in the range of the small values, but rough and strong control near endpoints. Works only in SPEED mode.

- **Limit Accelerations -** this option lets to limit angular accelerations in case of hard RC or Serial control (useful to prevents jerks or skipped steps, smoother camera control, less impact on the multirotor's frame). The less is value, the smoother is camera rotation under control.
- **PWM Output** – allows to pass any of virtual channel, decoded from serial input signal, to special pins that can output PWM signal. This signal can be used to drive hobby servo or IR camera trigger, for example. On the SimpleBGC 3.0 boards, these pins share PWM output function with other functions:

  Servo1 – FC_ROLL
  Servo2 – FC_PITCH
  Servo3 – RC_PITCH
  Servo4 – AUX1

  To enable servo output on any of these pins, make sure that its not specified as RC input in the GUI.

  This feature may be useful if you connect RC receiver by single wire, and want to decode signal to the separate PWM channels.

  When connecting regular hobby servo to these ports, there are two options where to get +5V to supply them:

  - Connect external power (for example from +5V BEC) to the central pin of any of RC inputs. and **cut (de-solder)** jumper J1 that passes 5V from internal voltage regulator to them.
    WARNING: two power sources joined together, may burn each other, because *switching* DC converter is used to provide 5V supply for the board, and it may conflict with the external power source.
  - **Close (solder)** jumper J1 and get +5V from internal voltage regulator.
    WARNING: before connecting servos, check their total maximum current rating, and compare it with the current rating that board can provide on the 5V line (you can find it in the hardware specifications of the board, for regular "Basecam SimpleBGC 32bit" version it is 1A).

7. Follow Mode Settings

There is a special control mode, when the camera “follows” for a tilting of the outer frame, but eliminates small frame jerking. Several modes of operation are possible:

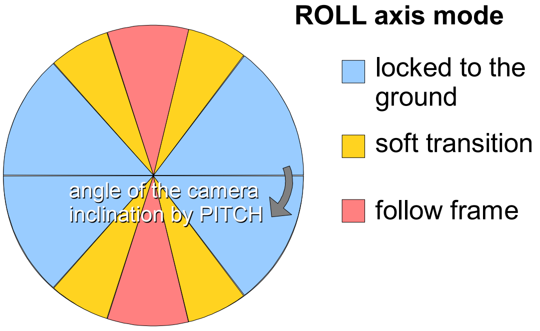
- **Disabled** – camera is locked to ground and may be rotated only from RC.
  - **Estimate frame angles from motors** - it use magnetic field for rough estimation of frame tilting. Helps to increase the range of the frame angles where the gimbal's operation is stable. To proper operation in this mode, it is strictly required to calibrate **Offset** setting (see below). Like with the Follow mode, its not recommended to use this option in flight, its dedicated for hand-held systems only.

    NOTE: This option is ignored if you connect second IMU mounted on the frame, because the data from the second IMU is more precise than from motors.)

- **Follow Flight Controller** – camera is controlled from RC together with the mixed signal from an external flight controller (FC). Almost every FC has servo outputs to drive a gimbal. It feeds the information about the frame angles to this outputs, in the PWM format that all servos understand. SimpleBGC can get this information and use it to control a camera. It is necessary to connect and calibrate external flight controller (see **EXT.FC GAIN** settings). After calibration you can setup the percentage values for ROLL and PITCH axis, so the camera will follow frame inclinations.
- **Follow PITCH, ROLL** – this mode is dedicated to hand-held systems. FC connection is not required. In this mode, the position of the outer frame by PITCH and ROLL is estimated from the motor's magnetic field. This means that if motor skips steps, position will be estimated incorrectly and operator should correct camera by hands, returning it to proper position.

  WARNING: you should use this mode carefully for FPV flying, because if the camera misses its initial direction, there is no chance to return it back automatically.

  - **Follow ROLL start, deg.** - Set the angle (in degrees) of the camera PITCH-ing up or down, where the ROLL axis enters follow mode. Below this angle, ROLL is in lock mode.
  - **Follow ROLL mix, deg.** - Set the range (in degrees) of the camera PITCH-ing, where the ROLL axis is gradually switched from the 'lock' mode to 'follow' mode (see picture)



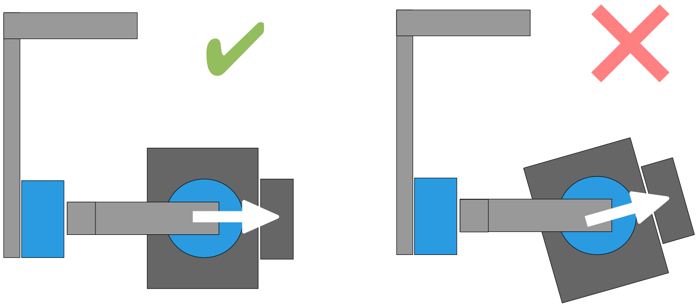
    HINT: To completely disable follow for ROLL, set these values to (90, 0). To permanently enable follow for ROLL (regardless of the camera PITCH-ing), set values to (0, 0).

- **Follow YAW** – the same as above, except it can be enabled only for YAW axis. For example, you can lock camera by ROLL and PITCH axis by selecting “Disabled” option, but still control camera by YAW by enabling “Follow YAW” option.

There are additional settings to tune follow mode:

- **Dead band, degrees**: you can set the range where the rotation of an outer frame does not affect the camera. It helps to skip small jerks when you operate gimbal by hands.

- **Expo curve:** you can specify the strength of the control when outer frame declines from neutral position. For example: when the expo curve is enabled (i.e. is not flat), small or medium declination of an outer frame will cause very fine control. But the strength of control exponentially grows when angles of declination becomes close to 60 degrees. It gives a big freedom in camera operation: from fine and smooth control to very fast movements.
- **OFFSET**: it is a very important to properly configure the initial position of the motor's magnetic poles, because all further calculations use this information. For YAW axis it allows to fine adjust a camera heading relative to a frame heading. For PITCH and ROLL axis there is an option to calibrate offset automatically. To do this, power on system, hold frame leveled, and press **AUTO** button. Don't forget to write setting when finished.
  If the camera after power on is not leveled, you need to adjust the offset setting.



- **SPEED -** adjust the speed of the camera rotation in the follow mode. Don't set big values that motors can not handle (if motor does not produce enough torque, it will skip steps and synchronization will be broken). In this case, acceleration limiter may help to have big speed but do not miss steps.
  IMPORTANT NOTE: For high SPEED values (above 50-100) its strongly recommended to set "LPF" parameter greater than zero (set it to 2-3 for example) and "Expo curve" parameter greater than 50. Otherwise, wrong system operation is possible, like vibrations and jerks under follow control, and overshoot of target.
- **LPF** – adjust low-pass filter applied to control signal. If this value set high, fast movements of the handle will be smoothed. But it requires careful operation near stop point, otherwise camera will overshoot after you stop rotation of the handle. Its recommended to not set it below 2.

Operation in the Follow Mode

At system startup in the follow mode, keep the frame horizontally and manually adjust the camera to the horizontal position, and adjust it's heading. Camera easily "jumps" between the magnetic poles. Rotate the camera by hands to desired horizontal position, it will stick to the nearest magnetic pole.

Gently rotate and tilt the frame. Turns within ± 45° will control the speed of the camera from 0 to 100%. Camera rotates in accordance with the **SPEED** settings until it's angles are not equal the frame's angles, or until given restrictions will be achieved.

If the camera moves unpredictably, perhaps its the wrong direction of rotation of the motors and you need to change the **Reverse** flag in the 'Basic' tab .

To achieve the smooth motion, increase the **LPF** parameter, increase **Expo curve**, and decrease the **SPEED** and the **Acceleration limits**. For more dynamic control, change these settings in the opposite direction.

In case of failure of stabilization due to external disturbances, the camera can completely lose synchronization with the frame . In this case, it is necessary to return it to the proper position by hands. IT IS VERY IMPORTANT to keep the frame horizontally, because at this point the frame's zero angles are calibrated.

You can switch between modes on-the-fly by activating different profiles. Camera will keep their position between modes.

8. Advanced Settings

- **AHRS** - options influencing on camera angle determination accuracy.
  - **Gyro trust** – The higher is value, the more trust to the gyro data compared with the accelerometer data when estimating angles. It can reduce errors caused by accelerations during moving, but also decreases gyro drift compensation, resulting in horizon drift over time. For smooth flying, it is recommended to set low values (40-80), which will give more stable horizon for longer time. For aggressive flying, it's better to set higher values (100-150).
  - **Accelerations compensation** – enable it to use a physical model of multirotor to compensate accelerations during flight. This option works only when external FC is connected and calibrated.
- **Serial port speed** — changes baud rate used for serial communication. Decrease it when using over-the-air serial adapters that can't work on maximum speed. The GUI can auto-detect the baud rate configured in the board.
- **PWM Frequency** — sets the PWM frequency used to drive motors by power stage. Two basic modes are available : **Low** Frequency (in audible range) and **High** Frequency (~22kHz outside audible range). Recommended mode is **High**. There is also third option present: Ultra-high (~30kHz).
- **Motor outputs** — you can assign hardware motor outs for any of stabilization axes. For example, you can use second controller for YAW stabilization and set it up this way: ROLL=disabled, PITCH=disabled, YAW=ROLL_OUT, and connect a YAW motor to hardware ROLL_OUT.
- **Sensor**
  - **Gyro LPF** – adjusts filtering gyro data. It's not recommended to set values different than 0, because it will make adjusting PID controller harder. You can experiment with this. *Starting from 2.42 this setting is replaced by Low-pass filter in the "Filters" tab.*
  - **Gyro high sensitivity** - Increases gyro sensitivity twice. Use this option for big-sized DSLR cameras, in case if your PID settings are close to upper limits, but stabilization still not good. Increasing gyro sensitivity equals to multiplying P and D values by 2.
  - **I2C Pullups Enable** - turns ON built in I2C pull-up resistors for SDA and SCL lines. Use function on only if sensor doesn't work properly. *This option has no noticeable effect for 32bit boards.*
  - **Frame IMU** – set the location of the frame IMU. See Second IMU sensor section of this manual.
- **External FC Gain** – Gain value for matching the gimbal data from your flight controller (optional). For better stabilization and utilization of some additional features, the knowledge about the frame inclination angles is required. SimpleBGC IMU doesn't provide such information. Most of FC have servo outs for connecting gimbals. This outs should be connected to SimpleBGC controller through EXT_ROLL and EXT_PITCH inputs.
  - Activate gimbal outs in FC and set range limits for angles you generally fly (for example ,+-30 degrees of frame inclination should equals full servo range about 1000-2000).
  - Deactivate all filters and smoothing of FC gimbal settings (if present).
  - In the **RC** tab, make sure that inputs EXT_ROLL, EXT_PITCH doesn't used to control gimbal. (i.e. are not chosen as source for any other RC control task).
  - In **Monitoring** tab, check availability of EXT_FC_ROLL, EXT_FC_PITCH signals, and make sure they are split to axes correctly. (Frame roll angle tilting should cause EXT_FC_ROLL change in

approximately 900..2100 range. The same is for pitch.)

- Connect power supply, and setup stabilization as described above (tune POWER, INVERT, PID)
- Push **AUTO** button in **External FC Gain** group, and smoothly incline aircraft's frame to different directions by all axes for 10-30 seconds.
- Push **AUTO** button again to complete calibration. (Calibration will stop automatically after some time too). New gains will be written into EEPROM and shown in the GUI.

NOTE: You may skip this step and leave zero values at initial setup.

9. Service Settings

Menu Button

If you've connected menu button to BTN connector on the controller, you can assign different actions to it. Action is activated by pressing button several times sequentially (1 to 5 clicks), by pressing and holding (long press).

Available actions:

- **Use profile 1..5** — loads selected profile
- **Calibrate ACC** – the accelerometer calibration, works the same way as button in the GUI.
- **Calibrate Gyro** – gyroscope calibration.
- **Swap RC PITCH – ROLL** — temporary swap RC inputs from PITCH to ROLL. In the most cases only one PITCH channel is enough to control a camera in 2-axis systems. Before a flight you can assign control from pitch channel to roll, and make a camera precisely leveled. Activating this function again swaps channels back, and saves roll position in the static memory.
- **Swap RC YAW – ROLL** — like the previous point.
- **Set tilt angles by hand** – motors will be turned off, after that you can take the camera in hands and fix it in the new position for a few seconds. Controller will save and hold the new position. This function may be useful to correct camera position before flight if there is no RC control connected.
- **Motors toggle, Motors ON, Motors OFF** - commands to change the state of the motors.
- **Reset controller**
- **Frame upside-down** – configures system to work in upside-down position. New configuration is stored to EEPROM and applied even after restart. To switch back to the normal position, execute this command again.
- **Look down** - points camera 90 degree down (or maximum allowed limit under 90, configured by the *MAX.ANGLE* parameter in the RC tab)
- **Home position** – returns camera to the initial position that is configured by the *INIT.ANGLE* parameter in the RC tab.

WARNING: There is a special action if you press menu button 10 times in series: full erase of all settings. Use this option for recovery only, if board is not accessible from the GUI.

Battery Monitoring

On the all 32-bit boards (and some 8-bit boards) there is a voltage sensor installed to monitor the main battery voltage. It is used to apply voltage drop compensation (to make PID stay stable during whole battery life-cycle), and to make low-voltage alarms and do the motor cut-off when the battery becomes discharged.

- **Calibrate -** adjusts the rate of internal multiplier to make measured voltage more precise. You need a multimeter to measure the real voltage, than enter this value in the calibration dialog.
- **Low voltage - alarm** - set the threshold to make alarm when the voltage drops below it.
- **Low voltage - stop motors** - set the threshold to stop motors when the voltage drops below it.
- **Compensate voltage drop** - set this option to automatically increase the POWER parameter (which controls the output power goes to the motors), when the battery loose voltage due to discharge

process.

- **Set defaults for** - select the battery type to fill the fields above with the default settings for selected type.

  NOTE: you can add the voltage sensor to old boards in DIY way, by soldering a voltage divider 33k/10k: 33k goes to the battery "+", 10k goes to the GND, and common point goes to the pin 19 of the 328p MCU (if this pin is grounded, de-solder it first).

Buzzer

On some boards there is an output to the buzzer (or buzzer is installed on-board), that is triggered on some events, like notification on errors or confirmation for user actions. Events are configured (turned ON or OFF) in the GUI.

You can connect an active buzzer only (which has an internal sound generator), working from 5V, currents below 40mA (check this Digikey product search for example)

If you have no buzzer connected, there is an option to beep by motors. Motors can emit sound only if they are powered and turned on.

Status LED

There are 2 LEDs on board. **Red** led lights when the power for MCU is present. **Green** or **blue** LED signals show actual state of the system:

- **LED is off** – pause before calibration, to take hands off or to level gimbal.
- **LED blinks slowly –** Calibration is in action. Freeze gimbal during this process.
- **LED blinks fast** – system error, stabilization cannot be performed. To check error description, connect to GUI.
- **LED blinks fast for short time** – confirmation for user action.
- **LED is on** – normal operation mode.
- **LED is on, but blinks irregularly –** I2C errors appears. Check in the GUI I2C errors counter.

Also, additional LEDs may present to signal serial communication on RX and TX line.

10. System Monitoring

In this tab you can see raw sensor data stream, logical RC input levels and some debug information.

- **ACC_X,Y,Z** – accelerometer data.
- **GYRO_X,Y,Z** – gyroscope data. Helps to determine quality of P and D settings. Disturb gimbal by hand and see trace. If it looks like sine wave, D setting is too low and gimbal tends to low-frequency oscillations. If some noise is always present even without any disturbance, D setting is too high and gimbal tends to high-frequency self-excitation.
- **ERR_ROLL,ERR_PITCH,ERR_YAW** – stabilization error graph. Same as peak indicators on the control panel and shows maximum deflection angle.

  HINT: Each graph can be turned on or off, scale can be adjusted for Y axis. You can pause the data transmission at any time.

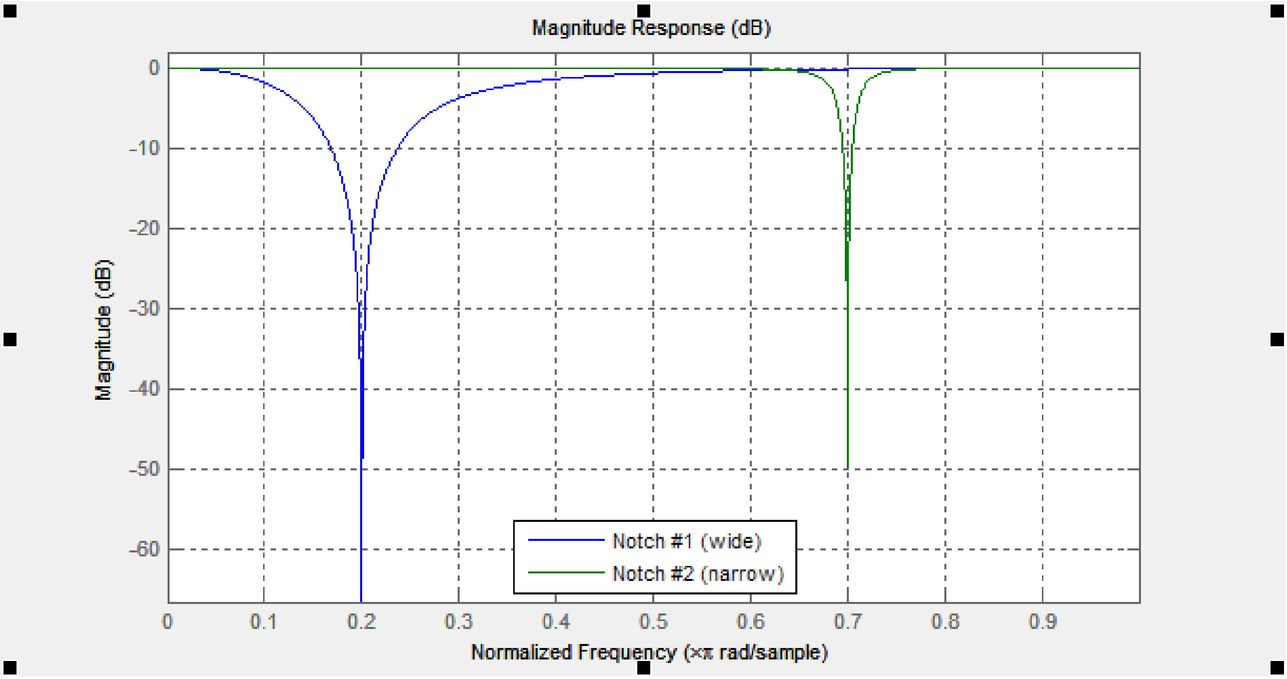
You can receive extended debug information from the board by selecting the checkbox "Receive extended debug info". Useful information you can get from the board:

- **RMS_ERR_R,RMS_ERR_P,RMS_ERR_Y** – RMS amplitude of gyro sensor data. In case of oscillations, it helps to define which axis is unstable. Its may be not so clearly from raw gyro data, because oscillations may have high frequency, far above a frame rate that GUI can receive and display.
- **FREQ_R, FREQ_P, FREQ_Y** – the main frequency of oscillation. If RMS_ERR is too small, this parameter is useless.

11. Digital Filters

This tab contains settings to configure digital filters that can help to improve quality of PID controller operation.

Notch filters



These filters can reject narrow bandwidth. They can help in case when the system has a pronounced mechanical resonance. Raising the extent feedback gain, oscillations will appear first on the mechanical resonance frequencies and does not depend on variations of P,I,D settings. In this case using one or several notch filters can help to increase feedback gain and get more accurate and stable work of PID regulator. But this filter will be useless if oscillations appear in the broad frequency range. In this case it is better to use low-pass filter.

Example: gimbal works stable, but, when camera tilts downward 60 degrees, strong vibration occurs and does not allow to increase gain of PID.

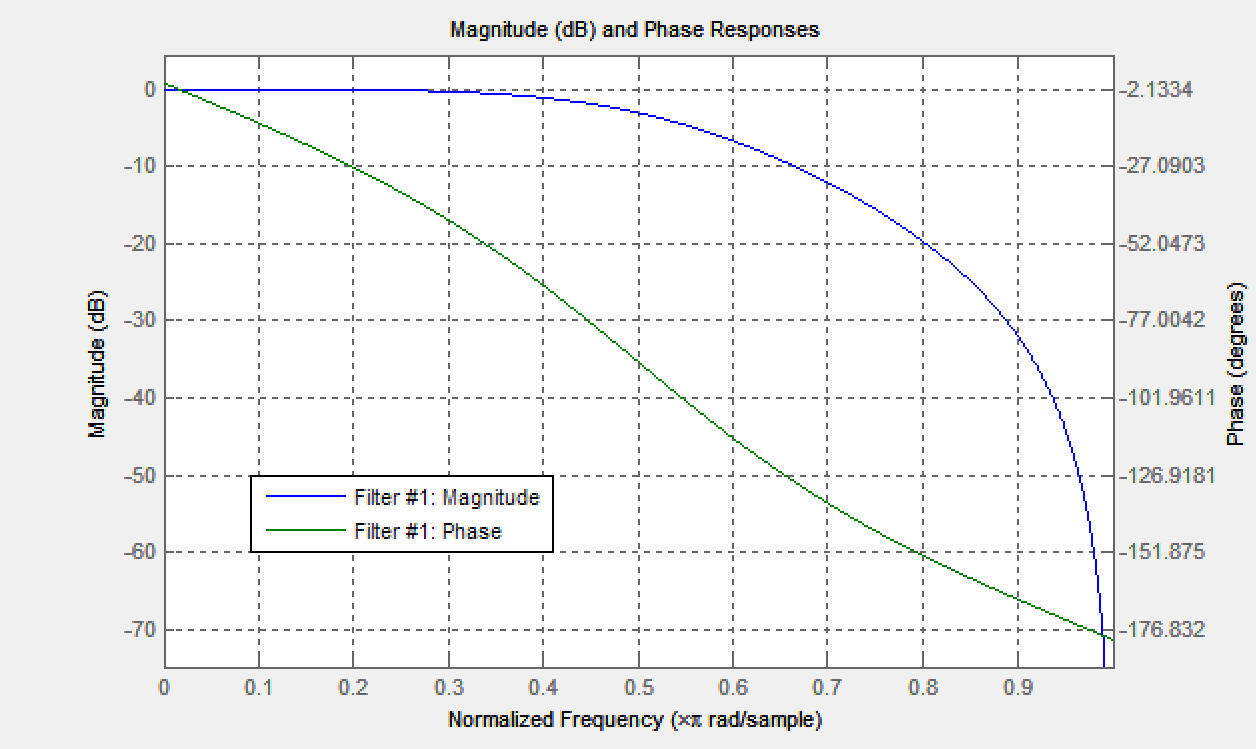
1. First, detect which axis causes vibration. To do this, in the GUI go to the tab “Monitoring” and switch on the following graphics: RMS_ERR_R, RMS_ERR_P, RMS_ERR_Y. Slow tilt the camera downward until vibrations occur. Most growing graphic will show the necessary axis. In the example it is RMS_ERR_P, Pitch axis.
2. When in steady state vibration mode, look at frequency indication: check another variable in the same tab, FREQ_P. It shows the main frequency of vibration (in our case it has value 100). Another way is to use spectroscope (for example, as application for smartphone that takes audio signal from mic), but it helps only if vibration is well-audible.
3. On the tab “Filters” fill out the parameters of the first notch filter for Pitch axis: Frequency: 100, Width: 10, checkbox “Enabled” is switched on.
4. Write the parameters to board. The vibration has been significantly reduced and its frequency shifted to 105Hz. Change the frequency of the filter to 105 Hz. Now the frequency shifted to 95 Hz. Set back value of the frequency to 100 and increase the bandwidth to 20. Now vibration on this resonance frequency is completely gone. Note, you

need to set the bandwidth as narrow as possible. Too broad bandwidth can result in decreased PID efficiency.

5. Having closed one resonance, continue increase gain of PID (Responsible for gain are the parameters P, D). Second resonance occurs on frequency 140 Hz, when we tilt the camera upward. Fill in values for second notch filter for PITCH axis to cancel this band, too, the same way as above.

In this example we have not needed to set filters for the other axes. But it can happen that resonance occurs on more than one axes. Then you will need to set filters on both axes (possible with the same frequency)

Low-pass filter



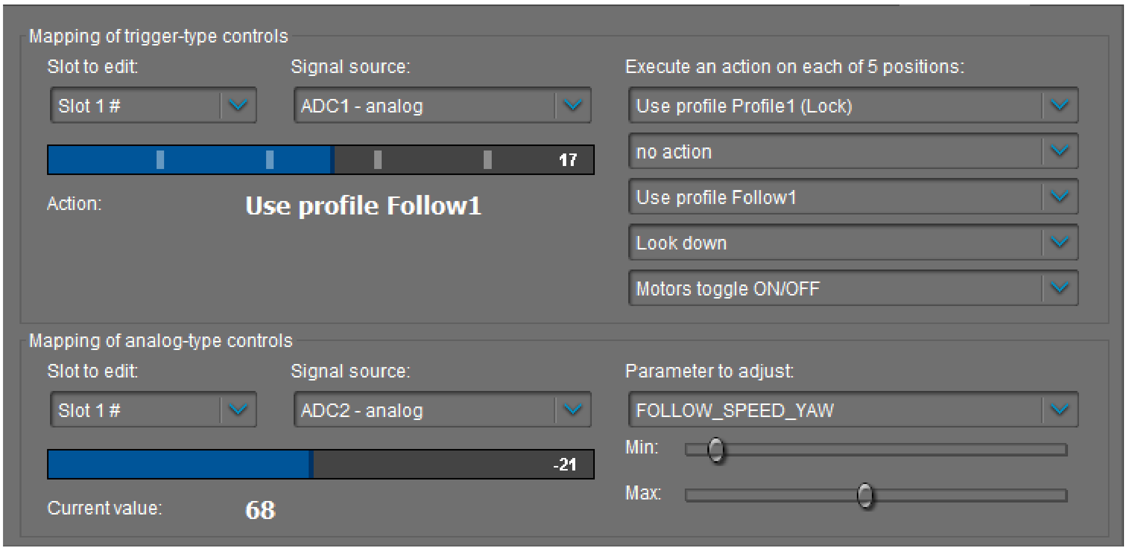
Applying this filter can be needed for large gimbals (heavy cameras with high moment of inertia) or for gimbals with reduction gear. The working frequency range for them are lower than of the lightweight gimbals. But factor D of PID-regulator increases feedback the more, the higher frequency. At a high frequencies, the response of the mechanical system can be not sufficiently precise and fast because of many reasons: high-frequency resonances, propagation delay of mechanical impact, nonlinearity due to the backlash and friction, etc. Due to this the system tends to self-excitation when gain increases. Low-pass filter reduces the gain at a high frequency and increases stability of the system. But as drawback, low-pass filter results in phase delay, which grows negative near the crossover frequency and can negatively affect the PID stability. This is a reason of the complexity of configuring this filter, and its usage is not always justified.

NOTE: Up to 2.42 version parameter Gyro LPF was responsible for LPF and provided a first order filter. Now it is not used and changed to a second order filter with more precise tuning of frequency and independent configuration for each axis.

12. Adjustable Variables

Beginning with firmware version 2.43, the controller supports not only remote control of camera angles, but also that of a large number of system parameters, allowing their change in real time. Also, it has expanded functions of various commands executed remotely - similar to channel CMD, but with a much more flexible configuration.

The tab with these settings is displayed after connecting to a 32-bit board with a firmware that supports this feature.



Mapping of trigger-type controls

Slot to edit: Slot 1 #

Signal source: ADC1 - analog

Execute an action on each of 5 positions:

- Use profile Profile1 (Lock)
- no action
- Use profile Follow1
- Look down
- Motors toggle ON/OFF

17

Action: Use profile Follow1

Mapping of analog-type controls

Slot to edit: Slot 1 #

Signal source: ADC2 - analog

Parameter to adjust: FOLLOW_SPEED_YAW

-21

Min:

Max:

Current value: 68

There are two types of control: Trigger and Analog.

- Trigger control is designed for connecting the buttons and switches in such a way that each state of the button triggers a certain command pre-assigned to this particular state. The entire range of the RC signal is divided into 5 sectors, whereby the transition from one sector to another triggers the action assigned. Up to 10 slots are available for matching the control channel set designed for 5 different functions.
- Analog Control is designed for fine adjustment of selected parameters by rotating the potentiometer on the remote control panel. It is also possible to switch between fixed values using a multi-position toggle switch, that almost all of RC transmitters have. Up to 15 slots are available for assigning the control channel to one parameter.

The source of a signal

For both types of Control, the signal source can be:

- **PWM inputs** on the board designated as RC_ROLL, RC_PITCH, RC_YAW, FC_PITCH, FC_ROLL. They use standard RC-receivers.
- **Analog inputs ADC1 - ADC3**. They can be connected to analog potentiometers with resistance value of 1-10 kOm (the end terminals are connected to GND and 3.3V, whereas the central terminal is connected to the ADC input).
- **Virtual channels** from multi-channel RC. In the event of connection of RC-receivers with a large number of channels over a single wire virtual channels of RC_VIRT_CH1 - RC_VIRT_CH32 receiver can also be used. You can read more on this in the section "RC Inputs".

- **Virtual channels** operated through the Serial API from another device. API_VIRT_CH1 - API_VIRT_CH32.

  TIP: This type of input allows independent developers to create an external control panel with any set of buttons, switches and potentiometers, serviced by a simple microprocessor (for example based on the Arduino software), which reads and transmits the state of control devices data over the wired or wireless serial-interface. Since the tuning of control functions is performed through SimpleBGC_GUI, software for such control panel can be extremely simple. Documentation of protocol «SimpleBGC Serial API specification» is available for download on our website – http://www.basecamelectronics.com

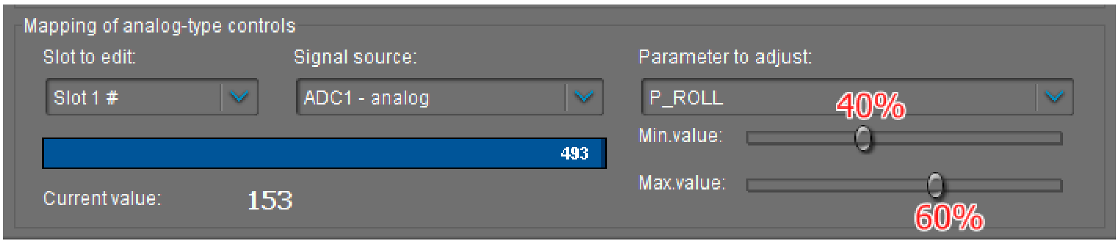
Setting control of the Trigger type

- Select a slot for tuning. Slots, where the signal source is already defined, are marked with '#' symbol.
- Select the signal source. One and the same source can be used for several slots simultaneously (but please make sure that the commands executed for individual slots do not interfere with each other).
- Assign actions to each sector. Possible actions are described in the section "Menu Button". You can leave any sector unused by specifying "no action".

After activating parameters by pressing button "Write", you will see the current RC signal level on the selected slot (for convenience, the whole range is divided into sectors), as well as the last activated action. You can check in real time whether actions are performed correctly in the case when the level of the signal has changed.

Setting control of the Analog type

- Select a slot for tuning. Slots, where the signal source is already defined, are marked with '#' symbol.
- Select the signal source. One source can be selected to control the number of variables at the same time, which can be convenient to change the value of a group of parameters by single control function.
- Select the variable that must be changed. Decoding of names of variables is presented in Table 1.
- Specify the range of variation by means of the sliders Min. and Max. For example, if the full variation range is 0-255, and you need to change it to the range 100-150, you will need to set the slider «Min.» at the mark close to 40%, and the slider «Max.» - at 60%, as shown in the picture:



Mapping of analog-type controls

Slot to edit: Slot 1 #

Signal source: ADC1 - analog

Parameter to adjust: P_ROLL

493

Current value: 153

Min.value:

Max.value:

In this case, the maximum control deviation corresponds to the parameter limit value of 153, that meets our requirements. Observing the parameter current value in real time, it is easy to estimate the required range by moving sliders.

There is a possibility to invert a control: when signal goes up, variable goes down. Just set Min. slider greater than Max slider.

You may notice that Min. and Max. sliders extend a range of a variable to ±10%. Its done for cases when RC signal is limited in range and does not cover full RC range ±500 (blue bar does not reach its limits).

After activating parameters by pressing button "Write", you will see the current RC signal level on the selected slot, as well as the current value of controlled variable.

Table 1. Decoding of names of controlled variables

Parameter's name	Description
P_ROLL, P_PITCH, P_YAW	Parameter of 'P' PID-controller
I_ROLL, I_PITCH, I_YAW	Parameter of 'I' PID-controller multiplied by 100
D_ROLL, D_PITCH, D_YAW	Parameter of 'D' PID-controller
POWER_ROLL, POWER_PITCH, POWER_YAW	Parameter 'POWER'
ACC_LIMITER	Acceleration limiter (unit of measurement: 4°/s2)
FOLLOW_SPEED_ROLL, FOLLOW_SPEED_PITCH, FOLLOW_SPEED_YAW	The speed of movement in the mode "Follow"
FOLLOW_LPF_ROLL, FOLLOW_LPF_PITCH, FOLLOW_LPF_YAW	Smoothing of operation in the mode "Follow"
RC_SPEED_ROLL, RC_SPEED_PITCH, RC_SPEED_YAW	Speed of movement when operating from the RC transmitter
RC_LPF_ROLL, RC_LPF_PITCH, RC_LPF_YAW	Smoothing of operation from the RC transmitter
RC_TRIM_ROLL, RC_TRIM_PITCH, RC_TRIM_YAW	Neutral point trimming for channels controlling the camera by ROLL, PITCH, YAW in the speed mode
RC_DEADBAND	The deadband of the RC signal for the camera control channels in the speed mode
RC_EXPO_RATE	Degree of the exponential curve depth for the RC signal
FOLLOW_MODE	Follow mode by the PITCH, ROLL angles: 0 - off, 1 - follow the flight controller, 2 - follow the gimbal's frame

RC_FOLLOW_YAW	Follow mode by the YAW angles: 0 - off, 1, 2 - follow the gimbal's frame
FOLLOW_DEADBAND	The deadband for the deflection angle of the frame in the Follow mode (unit of measurement: 0.1 degree)
FOLLOW_EXPO_RATE	Degree of the exponential curve depth for the Follow mode
FOLLOW_ROLL_MIX_START	The starting point of the zone transition to the Follow mode, degrees
FOLLOW_ROLL_MIX_RANGE	The length of the zone transition to the Follow mode, degrees
GYRO_TRUST	Trust to gyroscope compared to accelerometer

13. Firmware update

To check if the firmware upgrade is available, connect the board and press “CHECK” button. You will receive information about all available versions of firmware and can choose version to upgrade. When selecting a version in drop-down list, its full description is displayed in the text area below. To upload selected version to the board, press the **“UPGRADE”** button. Uploading process will be started. Generally, it takes about 10..30 seconds to finish. **WARNING! Do not disconnect USB cable (or break wireless connection) while firmware is uploading!**

PLEASE NOTE:

- For non-windows operating system, additional steps may be required. See notes at the end of this section.
- For “Tiny” version of the board, you need to install DFU device driver. Detailed instructions provided at the end of this section.

There is an option to configure system to check updates automatically. When new version will be issued, you will be prompted to upgrade to it.

If automatic upgrade fails just after downloading firmware from server (for example, there are may be problems to upgrade using Bluetooth connection under Mac OS), you can try to upload firmware in the manual mode. You can find downloaded firmware in the '*SimpleBGC_GUI/firmware*' folder and upload this file in the manual mode.

Uploading firmware in the manual mode.

This option intended to special cases when the board becomes bricked (GUI cannot connect to it) and you need to upload special “recovery” version of firmware, or when you experienced problems with automatic upgrade. *Use this mode carefully and only if you understand what you are doing!*

1. Disconnect any power source, USB cable.
2. Close (set) FLASH jumper on board (attach jumper to 2 pins marked as 'FLASH', shorting them)
3. Connect board to PC by USB cable
4. Run GUI, select COM port (but don't connect!) and go to "Upgrade firmware" - "Manual" tab
DO NOT PRESS "CONNECT" IN THE GUI, IF JUMPER IS CLOSED! If pressed, you need to repeat all steps from the beginning.
5. Choose firmware file (*.*hex* or *.*bin* format).
6. Select board version:
   - **v.3.x (32bit) through Virtual COM Port** – for a regular 32bit board
   - **v.3.x (32bit) through USB in DFU mode** – for a “Tiny” 32bit version
7. Press "FLASH" button and wait for process to be finished.
8. Open (remove) FLASH jumper.

If board is alive (you can connect to the GUI), you can upload firmware in manual mode without setting FLASH jumper:

1. Connect to the board normal way
2. Choose firmware file.

3. Press "FLASH" button and wait for process to be finished.

Upgrading under Mac OS and Linux

Starting from 2.42b7, its possible to upgrade firmware from GUI under Mac OS and Linux (and virtually, any other OS). Open-source tool **stm32ld** (https://github.com/jsnyder/stm32ld) is used to upload firmware to the board.

> **NOTE:** If its failed to run tool under your OS, you can compile it from sources (located in the 'SimpleBGC_GUI/bin/stm32ld-src' folder). Place a result to the 'SimpleBGC_GUI/bin' folder, renaming it to 'stm32ld_mac' for Mac OS, 'stm32ld_linux' for Linux family, and 'stm32ld' for any other OS.

Installing DFU device driver

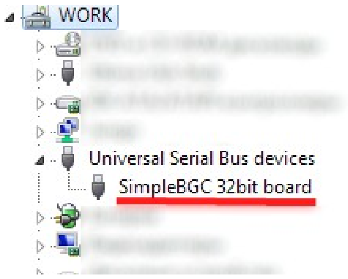
This driver is required only for “Tiny” version of the board, connected by USB. Open-source utility **dfu-util** (http://dfu-util.gnumonks.org/) is used to write firmware to this board.

Windows:

1. Download **Zadig** from the page http://zadig.akeo.ie/ (example)
2. Run Zadig. In the “Device" menu select "Load preset device.." (example)
3. Browse file "SimpleBGC_GUI/conf/SimpleBGC 32bit board.cfg"
4. Install driver **WinUSB** (example)

To check that driver is installed properly:

1. Close (set) “FLASH” jumper on the board and connect it by USB to PC (preserving this order exactly!)
2. Windows will find new device "SimpleBGC 32bit board"
3. Open (remove) jumper, re-connect USB and run GUI to upgrade firmware



Linux:

Most Linux distributions ship dfu-util in binary packages for those who do not want to compile dfu-util from source. On Debian, Ubuntu, Fedora and Gentoo you can install it through the normal software package tools. For other distributions (namely OpenSuSe, Mandriva, and CentOS) Holger Freyther was kind enough to provide binary packages through the Open Build Service.

- Copy dfu-util to “SimpleBGC_GUI/bin/dfu-util-linux” to let GUI to find and execute it

MAC OS:

Mac OS X users can also get dfu-util from Homebrew with "brew install dfu-util" or from MacPorts.

- Install macports from http://www.macports.org/install.php
- Find and install **dfu-util** from there
- Copy dfu-util to “SimpleBGC_GUI/bin/dfu-util-mac” to let GUI to find and execute it

FAQ and Troubleshooting

Q: Firmware uploading process was interrupted and board is not working now, not responding to GUI. Is it fatal?

A: No, its not fatal for your board (moreover, its impossible to damage board such way). You just need to upload special "recovery" firmware. You can find it in the "firmware" folder, named '*simplebgc_recovery_32bit*', or download from our site. Refer to instructions how to upload firmware in the manual mode. Then, you can connect to the board and upgrade to any version, as regular.

Q: I know from somebody, that there is fresh version, but I don't see it when checking for updates. Why?

A: Its normal. There are may be beta versions, available for beta-testers only, or may be different versions for different boards. You will receive only stable versions, issued for your board.

Q: Can I upgrade firmware from Mac or Linux?

A: Yes, starting from GUI 2.42b7. But check the note above.

Q: My board has no USB connector, but has bluetooth. Can I upgrade firmware?

A: Yes, you can upgrade via Bluetooth the same way as USB. Integrated module already configured properly to work for upgrade. External modules need to be configured to 115200 baud, even parity. If you have problems with re-connection to bluetooth under Mac OS, you can try to upgrade in the manual mode using “FLASH” jumper, as described above.

Q: I am using external bluetooth module and it works fine with the GUI. Can I upgrade firmware through it?

A: Yes, if you configure module to "Even" parity. To work with GUI, it may be both "Even" and "No" parity setting, but to upgrade firmware, it should be configured in the "Even" parity only. Look for instruction for your module how to configure it.

Q: Is it required to disconnect battery when upgrading firmware?

A: No, there is no matter if board powered from battery, or from USB only.

Q: After upgrade, my GUI can't connect to the board. What to do?

A: Its important that firmware and GUI both have matched versions. Changes in the firmware usually require changes in the GUI, so old GUI will not work with the fresh firmware. You can download matched GUI from our website. Link to matched GUI generally provided in the description of the firmware.

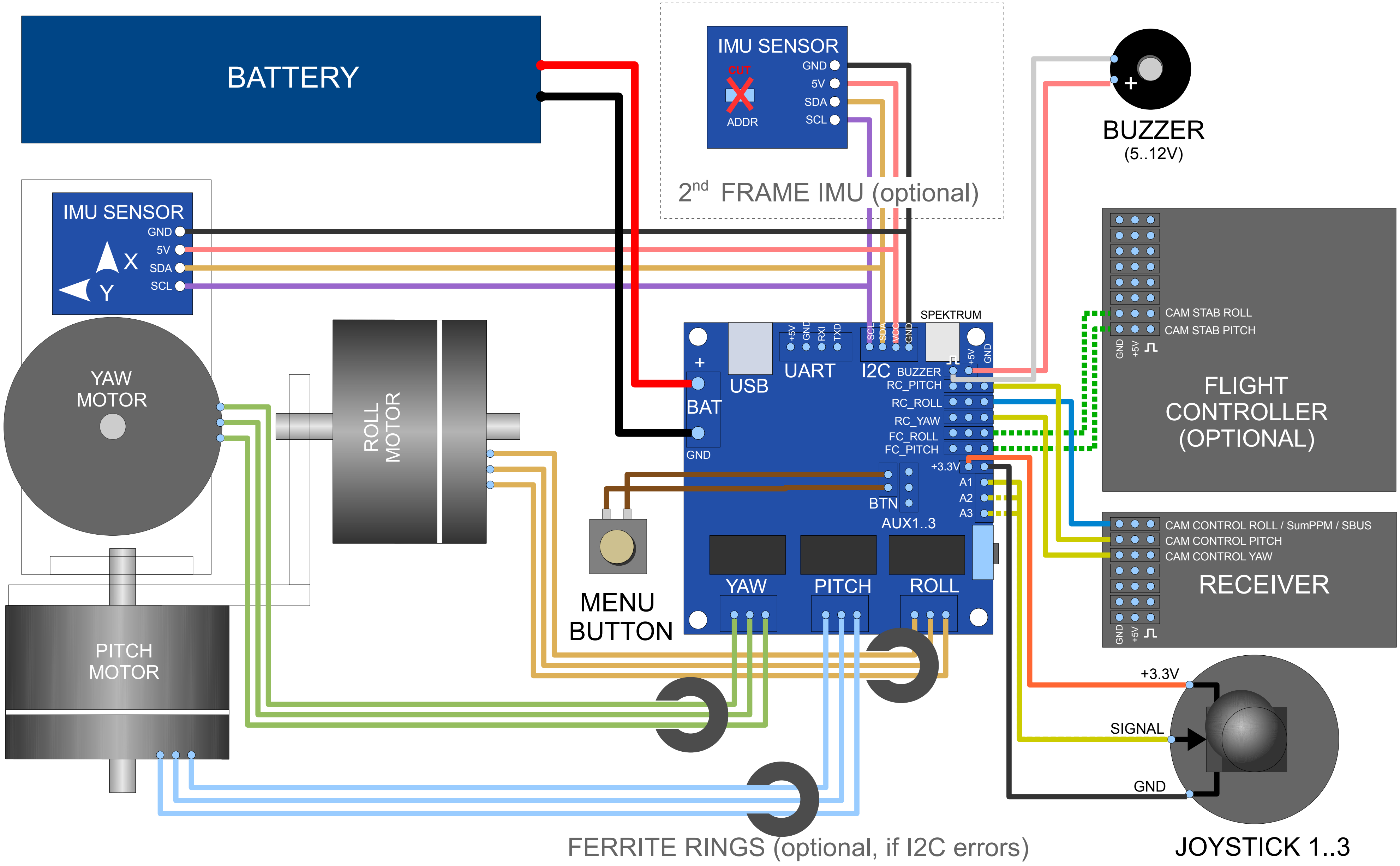
Q: I got error during uploading: "CreateProcess error=14001"

A: Some required libraries are missed in your system. You need to install Microsoft Visual C++ 2008 x86 redistributable: http://www.microsoft.com/en-us/download/details.aspx?id=5582

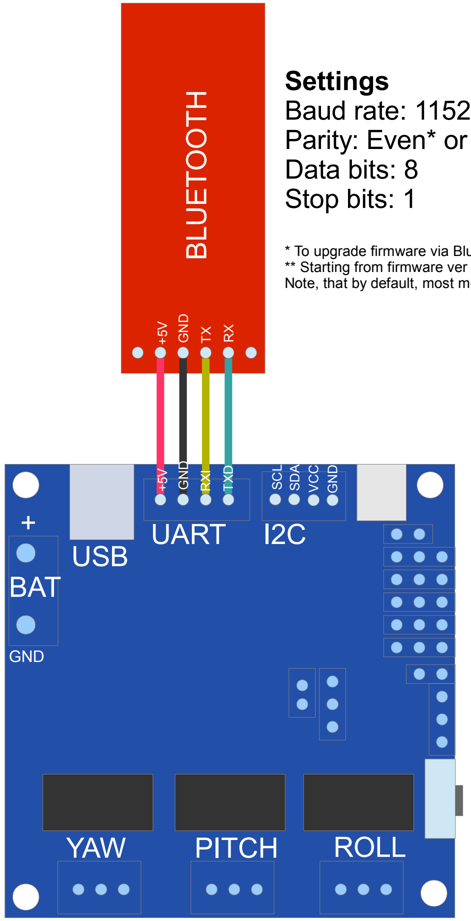
14. Possible problems and solutions

Problem	Possible causes	Solutions
Motors don't spin	-Power supply is not connected -Supply polarity inverted -POWER set to 0	-Check all connections -Set POWER between 50..200
Camera is trying to align, but falls back	-Camera is not balanced -It's an error in motor windings, or one phase is broken - POWER is not high enough	-Balance camera -Check motor winding - Increase POWER parameter
During fast YAW rotating, camera deflects by ROLL, and then slowly gets to horizon.	-Bad accelerometer calibration -Sensor is not in parallel with motor axes	-Make advanced ACC calibration by 6 positions -Align sensor with motor axes
During fast motion with acceleration, camera deflects, and then slowly gets to horizon	-This is normal effect of accelerations	-Try to increase Gyro Trust in Advanced tab
YAW arrow slowly spins in the GUI	-Slow drift is normal (less than 1 degree/minute). It's because of gyro drifts over time .	-Note to sensor Immobility during gyro calibration -Re-calibrate gyro
Camera slowly drifts by any or all axes just after power on	- Bad gyro calibration	-Re-calibrate gyro
Clicks and crunch are heard during work. LED is synchronously blinking.	-I2C errors present. Errors are possible if sensor wires are too long, or motors outs affect sensor by capacitive linkage.	-Shorter sensor wires; -Lower pullup resistors value on the sensor board; -Install spike LC-filter on motor outs (make 2-3 turns of motor cable through ferrite coil); - Install spike LC-filter on sensor wires (the same as motor filter);
High-frequency oscillations.	-Feedback self-excitation as a result of high D parameter	-Check the graphs to understand on what axis is the problem, and lower D value.
Low-frequency oscillations.	-Feedback self-excitation as a result of high D parameter or high P	Lower P, increase D
GUI cannot connect to the board.	-Wrong COM-port selected -GUI and firmware versions doesn't match.	-Try different COM-ports -Upload the latest firmware, and download matching GUI version.

SimpleBGC 3.0 (32bit) connection diagram



SimpleBGC 3.0 (32bit) bluetooth connection

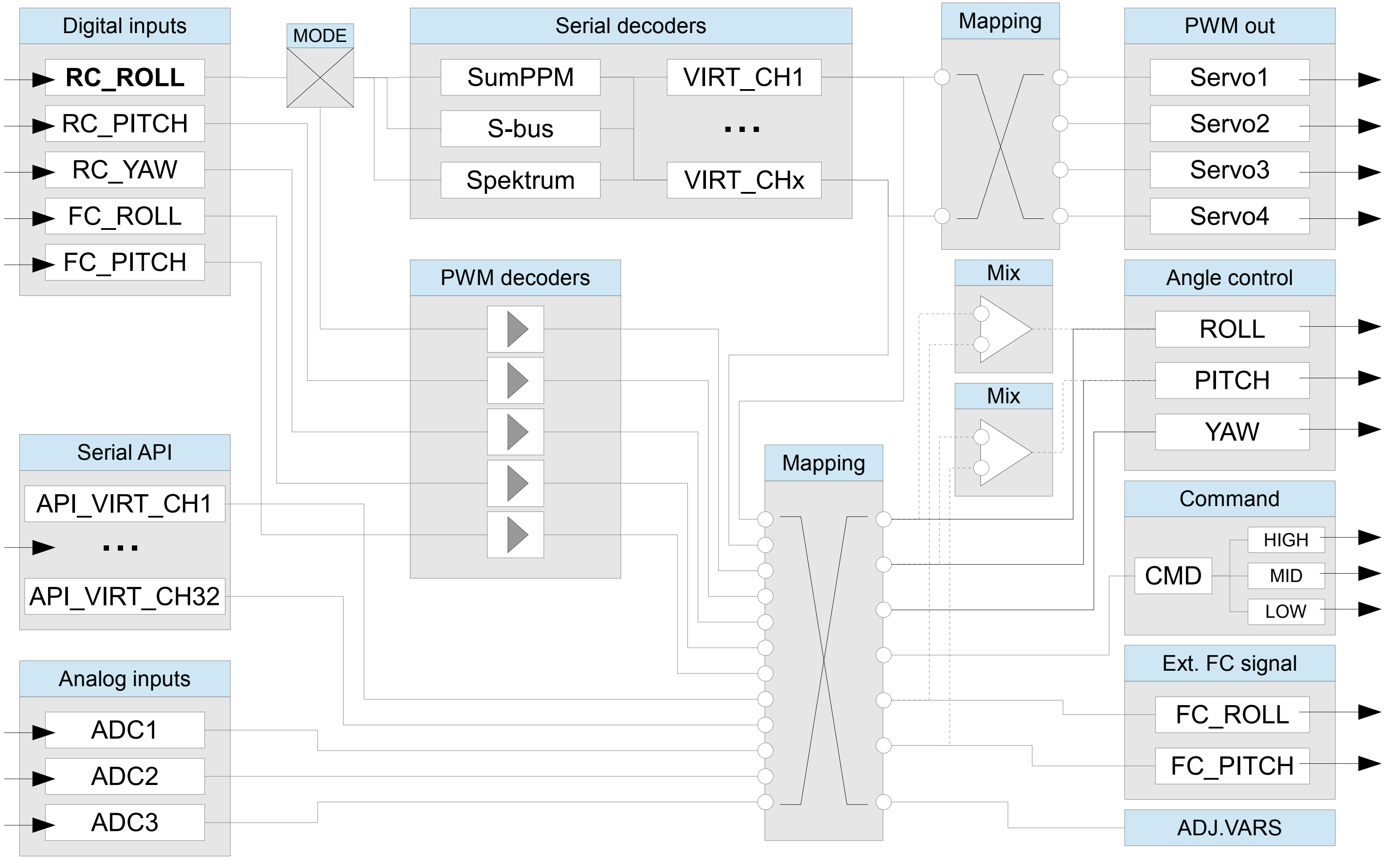


Settings
Baud rate: 115200
Parity: Even* or None**
Data bits: 8
Stop bits: 1

* To upgrade firmware via Bluetooth, only 'Even' parity will work.
** Starting from firmware ver 2.41, 'None' parity is supported, too. Note, that by default, most modules configured with 'None' parity.

SimpleBGC 32bit RC signal routing diagram

firmware ver. 2.43


Digital inputs
RC_ROLL
RC_PITCH
RC_YAW
FC_ROLL
FC_PITCH
MODE
Serial decoders
SumPPM
S-bus
Spektrum
VIRT_CH1
...
VIRT_CHx
Mapping
PWM out
Servo1
Servo2
Servo3
Servo4
PWM decoders
Mix
Mix
Angle control
ROLL
PITCH
YAW
Serial API
API_VIRT_CH1
...
API_VIRT_CH32
Mapping
Command
CMD
HIGH
MID
LOW
Ext. FC signal
FC_ROLL
FC_PITCH
Analog inputs
ADC1
ADC2
ADC3
ADJ.VARS