Using encoders with the SimpleBGC 32 controllers

Last edit date: 20.09.2016

Version: 0.11

© Basecamelectronics® 2013-2016

CONTENTS

1. General information.....3
2. Installing encoders.....4
3. Testing encoders.....9
4. Calibrating encoders.....10
5. FAQ.....13

1. General information

Using encoders with the SimpleBGC32 controller board gives the following advantages:

1. Prevents motors from loosing synchronization and skipping steps.
2. Provides full information about the position of a camera relative to a frame, that provides better model than 2nd IMU and makes use of 2nd IMU obsolete. This information extends the range of working positions and functionality of a gimbal.
3. Can significantly decrease power consumption by using field-oriented control strategy to drive motors.
4. Increases instant torque the same way.
5. Provides possibility to get information about the camera balance and makes automatic balancing (with help of extra DC servomotors and moving counterweights).
6. Allows to adjust camera position by hands.
7. Increase the precision of stabilization and the maximum speed of rotation that gimbal is able to compensate.

Encoders give completely new feel of operating the system, and recommended to use in the professional-grade products.

The stock firmware that comes with the regular SimpleBGC32 controllers does not support encoders. For using encoders, the special firmware is needed. You can get a trial or buy fully-functional encoder firmware here: http://www.basecamelectronics.com/encoders/

Gimbal manufacturers, please contact us at info@basecamelectronics.com

Note, that with this special firmware, all motors should have encoders installed! It is not possible to have one motor with encoder and other two without.

List of supported encoders

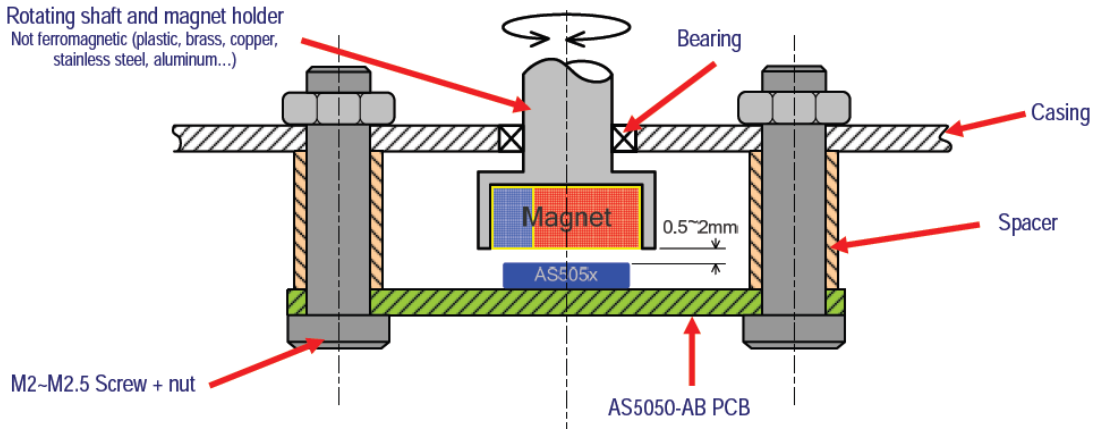
AMT203 – absolute (capacitive technology) 12bit, SPI communication
AS5048A,B – absolute (magnetic) 14bit, PWM, I2C or SPI communication, not assembled.
MA3 – absolute (magnetic) 10-12bit, assembled. PWM 1kHz and 244Hz interfaces are supported.
Analog – potentiometer or any other type where analog output is proportional to angle
AS5600 – absolute magnetic encoder with Analog, PWM 460Hz or I2C interface are supported (note that for I2C connection, address selection is not possible)
SBGC32_I2C_Drv expansion board – acts as external motor driver combined with encoder interface, controlled by I2C bus. See http://www.basecamelectronics.com/sbgc32_i2c_drv/ for details.
RLS Orbis *(ver.2.61+)* – absolute off-axis magnetic encoder. PWM 549Hz and SPI interfaces are supported.

Type of encoder can be selected during tuning process in the GUI.
There is no big difference between 12bit and 14bit resolution, because encoders are not used in the closed-loop PID controller, so high resolution is not strictly required. But ABSOLUTE angular error should be as low as possible, especially if working together with the motors having big number of pole pairs.

© Basecamelectronics® 2013-2016

2. Installing encoders

AS5048A,B version comes as an IC chip that should be soldered on the custom-made PCB with simple circuit (2 capacitors and connector), example is provided in the datasheet. This IC should be placed above specially polarized magnet, installed on the motor's shaft cup. Recommendations on choosing magnet and positions tolerances can be found in the datasheet.



For AS5048B connected to I2C bus, device address should be selected depending on motor where its installed:

ROLL: 0x40 (A1=0, A2=0)
PITCH: 0X41 (A1=HIGH, A2=0)
YAW: 0X42 (A1=0, A2=HIGH)

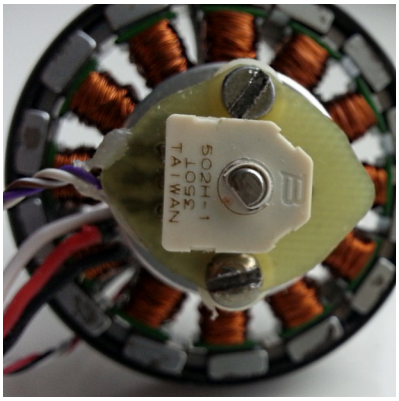
Also you can assign address to axis later in the GUI.
IMPORTANT NOTE: AS5048B is not compatible with the **"High speed I2C"** option!

AMT203 encoder has pass-throw hole and comes fully assembled with variety set of shaft adapters and centering tool. You only need to make cabling (cable and connector not included).

MA3 magnetic encoder that comes in 10 bit and 12 bit version (10bit has 1kHz refresh rate, 12bit has 244Hz refresh rate). http://www.usdigital.com/products/encoders/absolute/rotary/shaft/MA3

© Basecamelectronics® 2013-2016

Analog type may be a potentiometer, magnetic or other types with analog output, where voltage is linearly proportional to the angle of rotation of the shaft. Potentiometer-based encoder is a cheapest type. Analog encoders do not allow to have infinite 360 degree of shaft rotation, because there are forbidden areas at both ends of its range.



This type of encoder is available from many manufacturers, for example on the digikey.com in the category Sensors, Transducers > Position Sensors - Angle, Linear Position Measuring

Tips on analog type:

- Its better to choose lower resistance (about 1..3k) to minimize sensitivity to EMI noise
- If cable is long, twist it or shield, to minimize EMI noise affection
- You have to add mechanical limiters on the axis, to prevent entering to forbidden area where encoder does not provide a measurement. See datasheet to find working angles for selected model. Additionally, specify a software limiters in the GUI to softly stop near mechanical limiters, to not fight against them (see parameter *Limits (min,max)*)
- The more number of poles motor has, the more absolute precision is required from the encoder. Because precision is defined by the linearity of potentiometer, always check a specification of selected model and compare guaranteed linearity with the size of sector for single pole (360/number of poles). For normal work, maximum absolute error should be significantly less than sector size.

Sensitivity (scale) and offset for analog type is automatically set during regular procedure of calibration, like for other types of encoders.

AS5600 magnetic encoder can be connected by its OUT pin, that is by default is configured as analog output, and configured as "Analog" type in the GUI. Also, you can re-program it to the PWM output mode, and choose "AS5600 PWM" type in the GUI. But note, that In both modes, encoder does not support full 360 degree of rotation, because it has hysteresis +-40 points near zero angle (1.75 degree).

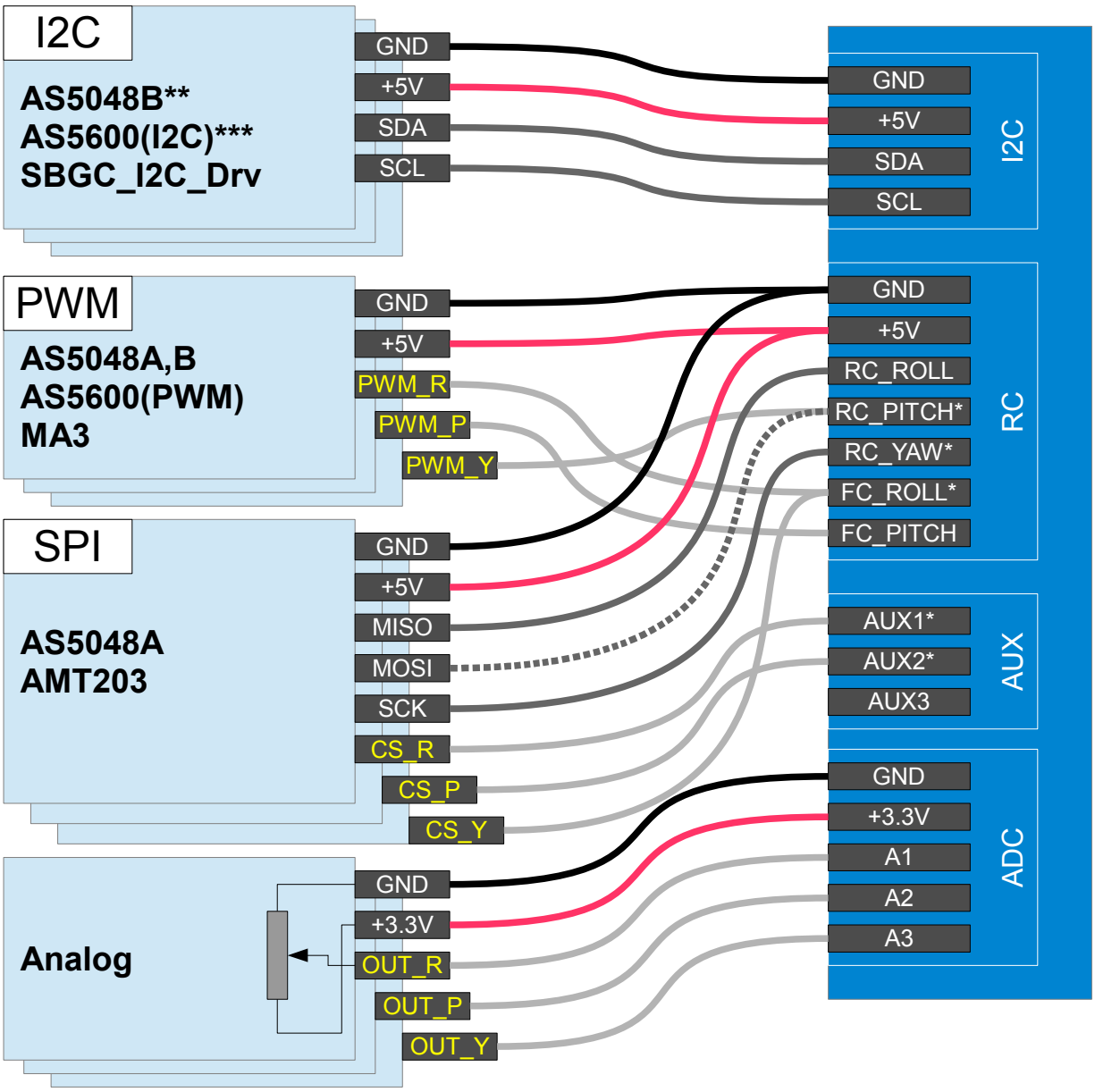
HINT: For the PWM output mode, you have to program CONF register* by I2C interface:
CONF = { WD(0) FTH(001) SF(11) }, { PWMF(10) OUTS(10) HYST(00) PM(00) } = {xx000111}, {10100000} = **0x07, 0xA0** - PWM at 460Hz, fastest response, no watchdog, no hysteresis, normal power mode.
After programming*, CONF should be written into the non-volatile memory. See AS5600 datasheet for details.
* If powered from 3.3V, 10uF buffer capacitor is required on VDD3V3 pin to ground to burn settings into OTP memory.

NOTE: starting from 2.60 firmware, AS5600 encoder will be automatically programmed into PWM mode, when connected by the I2C bus first time and selected as "AS5600 (PWM)" in the GUI. Only 1 encoder at a time may be

connected by I2C. When programming is done, I2C lines may be disconnected. **WARNING! Its impossible to use encoder in analog mode after programming is done!**

The best way is to connect it by I2C interface – in this case it will support full 360 degree rotation. But only single AS5600 encoder may be present on I2C bus. If combined with the 2x I2C_Drv expansion modules, you can use AS5600 encoders for all three motors.

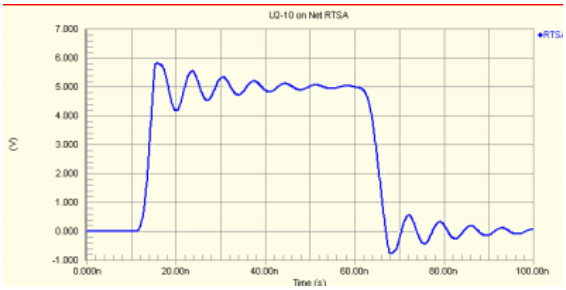
Connecting encoders to the regular SimpleBGC 32bit board

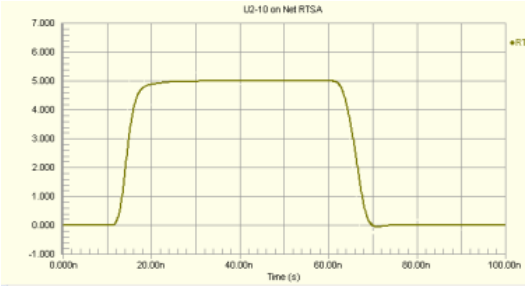


* These pins acts as output in case of SPI connection. Be carefull and avoid short circuit or improper connection to other devices.
** AS5048B is not compatible with the "I2C high-speed" option.
*** Only single AS5600 unit can be connected — it's address is fixed.

Tips on connection

- SPI or PWM is preferred than I2C. I2C can be used only for system with short cable.
- For AS5048A there is an option to use 3-wire SPI: MOSI line can be omitted and connected to VDD3 (high level) on the encoder side. But in this configuration will not be available diagnostic info from encoder.
- For long SPI cable (more than 30cm) resistors on the SCK, MOSI line are strictly required! Use 33..60 Ohm resistor in the series on any output line (SCK, MOSI from the board side, MISO from the encoder side). You can adjust their values more precisely by observing signal slope on the scope on the receiver side. It should be flat and sharp enough, without spikes caused by signal reflections (see picture below).
- AS5048 PWM connection gives less resolution (12bit compared to 14bit over I2C or SPI) and can give less precision, but PWM is an easiest way of connection.
- Analog signal is very sensitive to EMI noise, its recommended to use shielded cable in case of long distance (for big gimbals, its better to choose different type of encoder).
- For all PWM encoder types, its better to setup them such way, that zero-crossing point (where raw data crosses zero) is located far away from normal working angles, because near this point PWM signal can rapidly jitter between 0% and 100% duty cycle that may cause problems in PWM capturing.





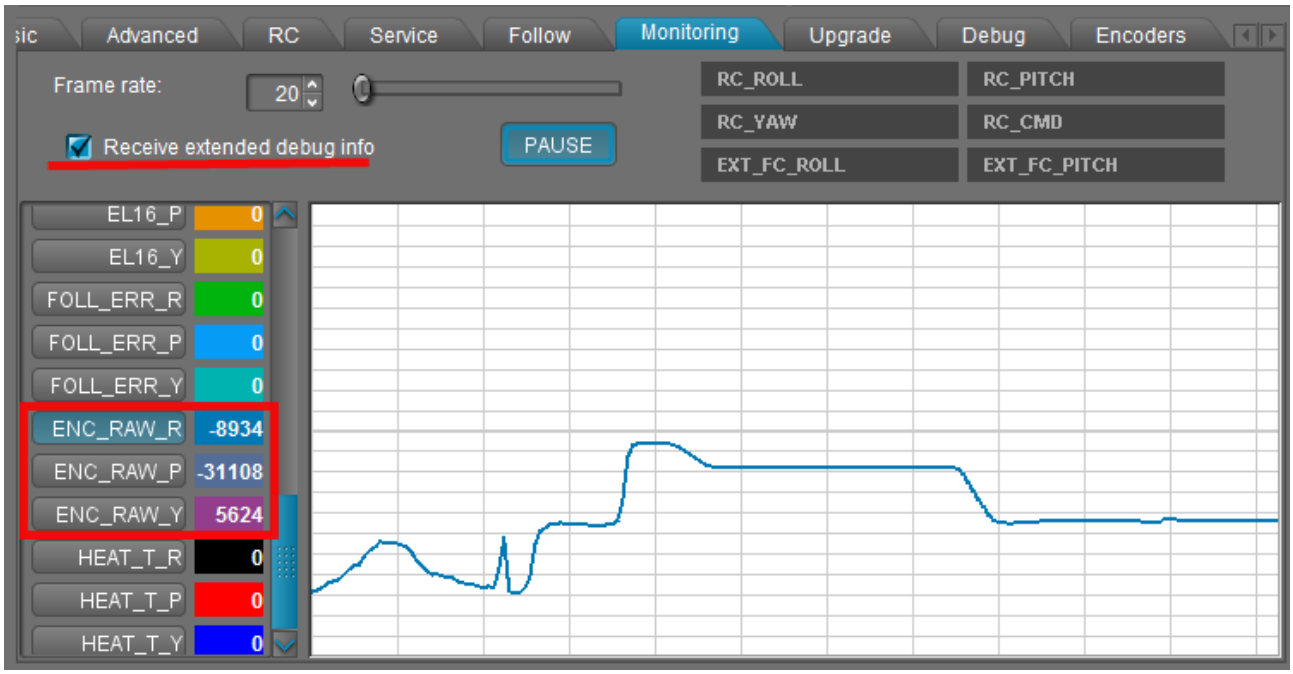
Bad signal (direct connection) and good signal (with resistor in series)

IMPORTANT NOTES:

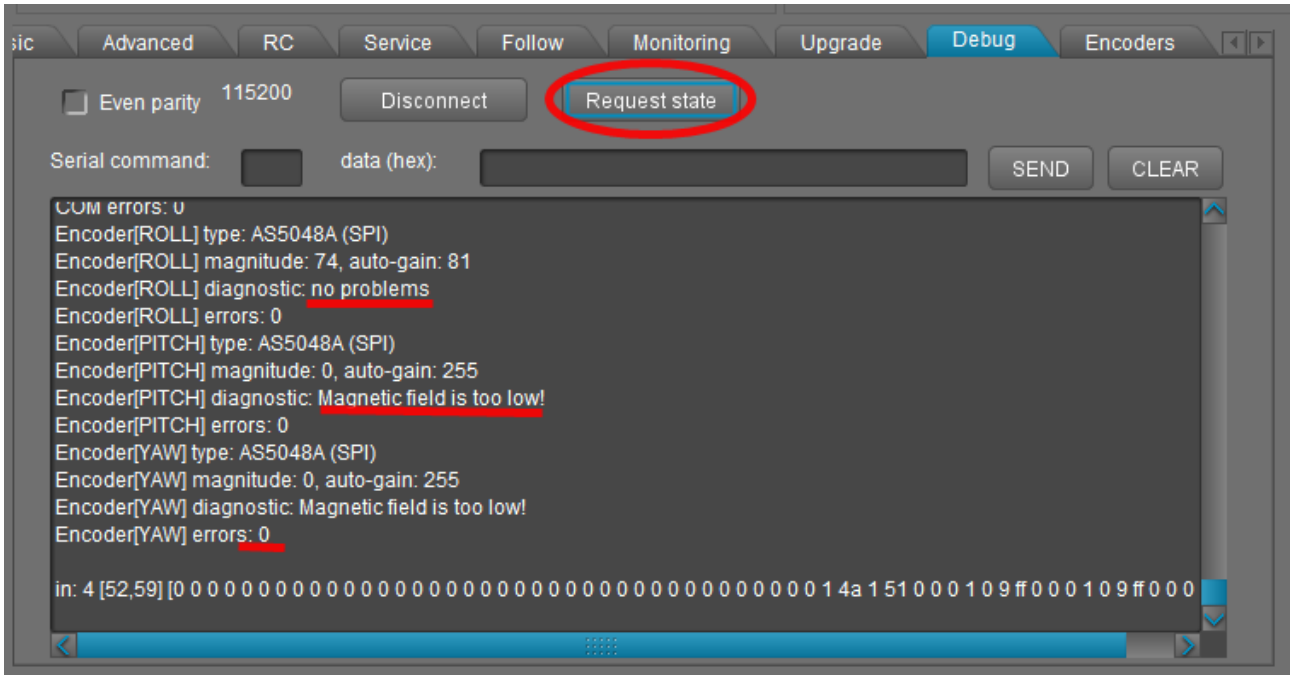
- In the "Regular" and "Tiny" controllers and their modifications, RC_ROLL input in this version of firmware is occupied by the SPI bus, so its alternative functions (Sum-PPM and serial) are moved to the AUX3 input. "Serial functions" include Spektrum, S-bus and SBGC Serial API interfaces.
- If RC_ROLL input is not occupied by the SPI bus, starting from the firmware 2.60, you can return all serial functions to it's origin. To do this, disable the check-box "Swap RC_SERIAL ⟷ UART2 ports" in the "Advanced" tab
- For "Extended" and "Pro" versions of controller, there are dedicated ports for SPI and PWM interfaces, so RC_ROLL pin functionality is unaffected.

3. Testing encoders

Connect from GUI to the board with encoder-enabled firmware. Select encoder type in the “Encoders” tab, write settings and restart the controller. If encoders are connected and configured properly, you will see a raw data from the encoders in the “Monitoring” tab:



For some types of encoders you can request additional diagnostic information. Go to the "Debug" tab in the GUI and request system state:



Error count available for all types of encoders, and shows how many reads were failed. If you see “DISABLED” for encoder type, it means that auto-checking at startup was failed. Generally it means that something wrong with connection or encoder is not working at all.

4. Calibrating encoders

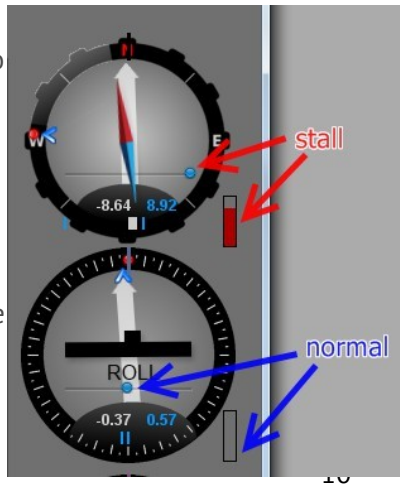
1. Calibrate electrical angle offset and direction

- Tune mechanics. Perfect balance is very important here (at least in the leveled camera position)!!!
- Calibrate poles and direction (find out proper setting for "Invert" checkbox) in the "Basic" tab, as usual, or enter known values. Do not change direction after this step!
- Reset all calibration values in the "Encoders" tab to zero.
- Power on gimbal, tune PID. Stabilization should work without jitters at this moment, and big precision is not required.
- Disable Follow mode or any kind of RC control.
- Press **CALIB.EL FIELD** button.
  - *Firmware ver prior to 2.60:* you have about 20 seconds to tilt frame +-10..30 degrees VERY SLOWLY for each axis, several times for all axes. If any axis will loose sync, reset controller and restart calibration. Also you can tilt a camera instead of a frame by mean of RC controller, but very slowly, too.
  - *Firmware ver. after 2.60:* you do not need to move frame. Just fix it or hold in hands firmly – calibration will be done automatically, each motor in series.
- On completion, calibration data will be transferred to the GUI and encoders will start working.
- Check that parameter "**Encoder/motor gearing ratio**" is equals to 1.0 or very close to this value. Otherwise, scale factors of encoder and motor do not match. It means that motor is not properly configured (check number of poles), or calibration is done improperly.
- There is a simple error checking is present: if estimated scale factor differs a lot from the specified value (or 1.0 by default), "Emergency fault" will be generated and motors will be turned OFF. You need to check resulting values and accept them, or clear them and repeat calibration after restarting of a system.
- Set **Heating factor, Cooling factor** to 0 to disable current limitation at this moment. Increase **POWER** to that level, where motor effectively resists external disturbance, but the current is safe for motor and motor driver (test it at a stall condition, where power indicator goes to maximum). The long operation in stall condition may overheat motor or controller. You can limit POWER to safe level, or setup heating-cooling model to have big current in short time, but limit it at a stall condition after a while, or cut off power completely.

This is a short video that shows procedure of calibration of encoders (for old version of firmware): http://www.youtube.com/watch?v=sAnWG2ipavE

Checking calibration

A "balance error" value (displayed in the GUI as blue ball on the angle panels) should be about neutral position for each axis during work, and go to its limits under pressure. This indicator shows shifts in the balance and may be used as reference to help balancing camera. Power indicator is zero in normal condition and goes to maximum at a stall condition, when motor resists to push.



Reusing of calibration data

You can use this data for other gimbals in batch, if encoder is mounted the same way, has the same offset in data, and the motor EXACTLY matches the previous one (including mounting of rotor, stator, winding, connection)

© Basecamelectronics® 2013-2016

Backing-up a calibration data
Always make a EEPROM backup for each item you are sending to a customer. If system will accidentally loose all setting, or customer will brake something, it will be useful to have an option to restore factory default settings and calibration, especially in case of complicated calibration procedure that is hard to repeat.

2. Calibrate offset

- You may keep gimbal working, or turn motors off. Stabilization is not required.
- Put the camera to "normal" position relative to the frame, where all motors forms zero angle to a frame.
- Press **CALIB. OFFSET** button. Current angles read by encoders, will be set as zero offset.
- New settings will be written to EEPROM and transferred to the GUI.

Checking calibration
In the GUI, white arrows show the angle of each motor relative to a frame. They should point up when the camera is in "home" position relative to a frame, that mean zero angle. When motors rotate, white arrows move.

Notes on geared setup
Controller supports 2 types of gearing: encoder/motor and motor/frame. If gear ratio is not equal to 1.0 in any of this cases, there is a potential problem with the detection of initial angles of motors at system startup. In case of encoder/motor gearing is not equal to 1.0, it may be also a problem with the proper electrical field offset detection. Use the following rules to let system to start properly:

- Hold the frame in normal position (where encoders was calibrated) at startup.
- Connect and configure 2nd IMU. If present, its data will be combined with the main IMU and used to detect relative angle of each motor. But its impossible to detect YAW motor angle by this method, because IMU's are not referenced globally by YAW axis.
- Chose the lowest gearing ratios, if possible. Remember that direct drive is the most optimal choice for this application.
- Try to minimize back-lashes. Otherwise, it will be very hard to get good PID values and good precision.

Settings, related to the operation of the system with encoders

Manual set time – if camera is rotated by hands and fixed in new position, firmware waits some time. If camera is held during this time, new position is fixed as target position. Default value is 0.5 sec. To disable this feature for any axis, set it to 0. In this case, camera resists against any external disturbance for infinite time.

Heating factor, Cooling factor – these are settings for simple "Heat accumulation/decay model". Boost power, when applied, increases virtual temperature of the motor in the model, according to heating factor. When temperature rises, it chops the output power. Cooling factor specify the rate of heat emitted to the outer space. Tune these values to make model close to real life. It lets to feed big power to motors for a short time without the risk of overheating them.

To control actual state of the model, you can check **HEAT_T_X** debug variables in the "Monitoring" tab. It shows virtual "temperature", where 0 means power is not limited at all, and 100 means power is completely off. To test how model work, apply force to the motor to put it in the "locked" state and see how **POWER_X** and **HEAT_T_X** graphs are evolved. Adjust model settings to provide safe motor temperature under "lock" conditions.

© Basecamelectronics® 2013-2016

Starting from **2.42b7** version of the firmware, a new motor control algorithm is used – simplified version of Field Oriented Control (FOC), that is optimized for power consumption and linearity. In steady state (without motion and with perfect balance) motor takes almost zero current. POWER setting in the GUI defines maximum power that motor takes under lock condition, and the rate of PID gain (as before, if you increase POWER, overall PID gains are proportionally increased). BOOST POWER is not more used and should be set to 0.

Motor's magnetic linkage – this parameter is used to make more precise model of motor in the FOC algorithm. Its depends on motor size, number of poles, number of turns in winding.
Starting from 2.60 version of firmware, this parameter was moved to the "Advanced" tab and can be calibrated automatically. Refer to SBGC32 User Manual for more information.

Motor/frame gearing ratio – if a geared drive is used to drive any axis, encoder should be mounted directly on the motor's shaft. Set the "number of poles" parameter to actual number of poles that motor has, and set proper gearing ratio value to this field.

Encoder/motor gearing ratio - this parameter is used for analog type of encoder, to calibrate its sensitivity (scale factor). For all other types it should be disabled, excepting the case when the mechanical gearing ratio is not equal to 1.0. If disabled, it is updated during the calibration process and allows to detect possible problems, if it differs a lot from 1.0.

Limits (min, max) – software limits applied to relative angle measured by encoder. if specified, when motor reaches this limits, it will be locked to maintain relative angle. If motor had a big speed before it reach limit, it can overshoot limit and than return back. Its useful to have a software limits set slightly before a hardware (mechanical) limits, otherwise motor will fight against hard limit, applying full power to overcome it.

Choosing motors

To get maximum from you setup, its recommended to choose motors with the lower winding resistance compared to regular gimbal brushless motors that are present on the market (most of them designed to work with constant current and have too big resistance). Because normally there is no DC component of the current, motor can have less resistance, less size and work without overheat. Under external disturbance, such motor may take bigger current and give bigger torque to fight against it. In the locked state, maximum voltage is applied (set by POWER setting where 255 means ~70% of full battery voltage) and motor provides maximum torque. To limit the amount of time when motor operates under high current, use heating-cooling model that will chop current after seconds of work in lock mode.

But resistance should not be as low, as for regular brushless motors, where stall mode is outside of normal operation conditions. Instead, in our case, stall mode should be considered as normal operation conditions during limited period of time. By this parameter, gimbal motors may be compared with the DC servo motors, used in robotics and industry.

5. FAQ

Could encoder be used on yaw only? (On aerial gimbal I am now using potentiometer on yaw, but would like to replace it with encoder, but not add too much complexity, i.e. not use encoder for pitch and roll) Would current boards support this kind of "simple" use in future (also other than test use)?

Yes, starting from 2.56 ver. of regular firmware, it is possible to use single encoder on YAW motor only (you do not need to update to "encoder" license for this). Encoder is used to read motor angle only, with no changes to the motor control algorithm.

If you are using encoder-enabled firmware, ALL axes should have encoders installed and configured.

I have GB90 and GB85 motors. after reading the documentation I understand they might not be ideal to be used with the encoders. Will they work, what negative effect high resistance/inductance has?

Encoders will work with them, but precision of angle reading is more crucial. For example, with magnetic encoders, demand to precise installation of magnet over IC is bigger.

If inductance is high, KV factor is low. That mean that such motor will not compensate fast speed rotations (speed is limited by Battery Voltage * KV)

What would be ideal resistance for GB90 motor?

All depends on maximum current that controller may provide. For regular SimpleBGC 32 bit board, where current is limited by several Amps, their actual resistance 10-20 Ohm is good. For more powerful board, resistance 3-10 Ohm is more preferred to get maximum torque and high KV factor. The same is truth for any other motor.

What kind of results are you getting with encoders? The biggest problem I have with my current handheld is jitters when gimbal is moved, especially on follow mode or with joystick, at certain (medium) speeds.

All benefits of using encoders are listed in this document. But it will not help to overcome jitter, if its caused by cogging effect that is a property of a motor, and will remain with encoders, too. Reducing pole number can help in this case (frequency of jitter goes down and becomes in the area of reach of PID loop), as a choosing different motor with uniform O-ring magnet instead of separated magnets.

I calibrated encoders properly, but torque is much less than in the setup without encoders, even with POWER set to maximum.

Case1: push to motor by hand. If the power indicator in the GUI does not go to maximum, it seems that heating-cooling model limits it. Set Cooling factor, Heating factor = 0 for all axes.

Case2: under push, power indicator goes to maximum, but torque is still low. Repeat calibration. Check that "Encoder/Motor gearing ratio" is close to 1.0 after calibration. Check that encoder angle reading is correct in the "Monitoring" tab (ENC_RAW_x value reflects shaft rotation and is smooth, has no steps, spikes, etc.)

After all calibrations motor works well only in small range of angles and start to spin erratically outside it.

Check that the "NUMBER OF POLES" parameter is set correctly. Even error in 1 point will cause wrong operation of FOC algorithm. Another possible problem may be non-linearity of encoder reading, caused by bad magnet or wrong placement of a magnet over chip for a magnetic type of encoder.

I can not get reliable results with encoders and calibrations do not help.

First of all, check that encoders provide consistent data and they are assigned to proper axes. In the "Monitoring" tab, look at the raw values from encoders, and check that they reflect the actual angles precisely: rotate motor by hands to a fixed angle (for example 90 degrees), and observe that the increment in the encoder's raw value is 4096 points.

© Basecamelectronics® 2013-2016