

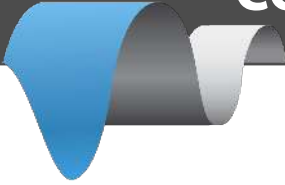


# **SimpleBGC 32bit controllers Using with encoders**

Last edit date: 23 October 2014

Version: 0.5

# CONTENTS



1. Encoders in the SimpleBGC project.....	3
2. Installing encoders.....	4
3. Testing encoders.....	7
4. Calibrating encoders.....	8
5. FAQ.....	11

# 1. Encoders in the SimpleBGC project

## 1. Encoders in the SimpleBGC project

Using encoders with SimpleBGC board gives the following advantages:

1. Prevents motors from losing synchronization and skipping steps.
2. Provides the information about frame angle and angle of the camera relative to frame, that can be important for some kind of applications.
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. Option to adjust camera position by hands.
7. Increase the precision of stabilization by applying FOC-specific compensations.

Despite the implementation of some new features is still in progress, encoders now give completely new feel of operating the system, and recommended to use in the professional-grade products.

**For using Encoders with SimpleBGC 32bit special firmware is needed. It is provided by BaseCam Electronics by request at [info@basecamelectronics.com](mailto:info@basecamelectronics.com).**

**You can get trial version of the firmware with limitation for tests.**

### 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 communication.

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 big resolution is not required. But ABSOLUTE angular error should be as low as possible, especially with big number of poles, where full electrical angle is small.

## 2. Installing encoders

### 2. Installing encoders

**AS5048A,B version** comes as a 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. On image below is shown the encoder installed on our system:



**AMT203 encoder** has pass-through 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).

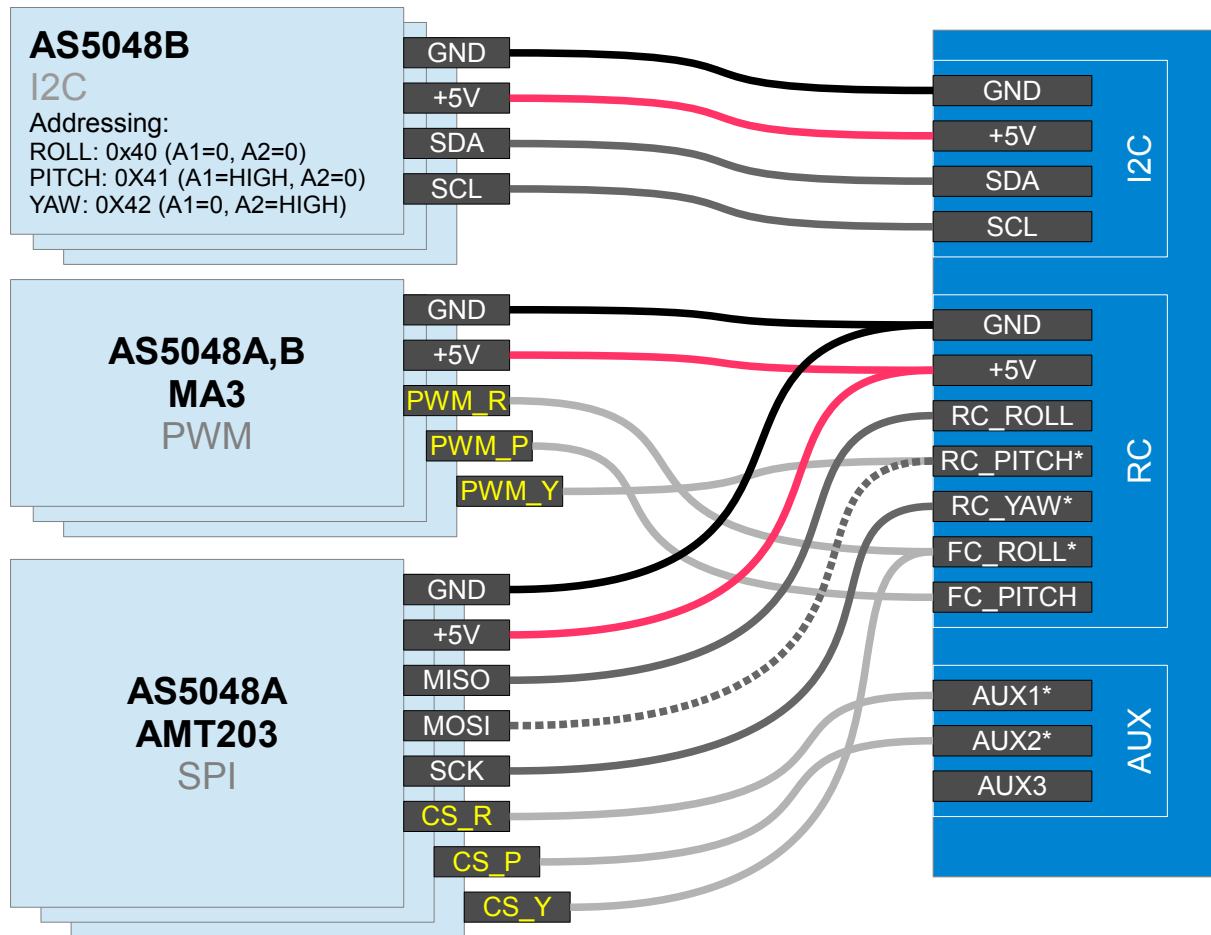


**MA3 magnetic encoder** that comes in 10 bit and 12 bit version (10bit has 1kHz refresh rate, 12bit has 244Hz refresh rate). <http://www.usdigital.com/products/encoders/absolute/rotary/shaft/MA3>



## 2. Installing encoders

### Connecting encoders to the regular SimpleBGC 32bit board

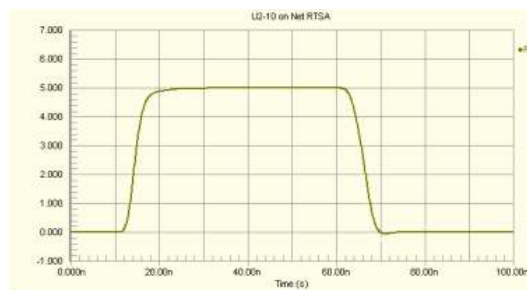
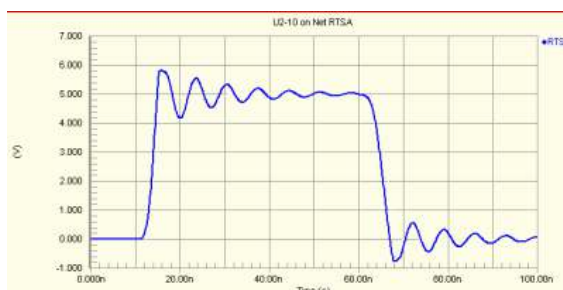


\* These pins acts as output in case of SPI connection. Be carefull and avoid short circuit or improper connection to other devices.

## 2. Installing encoders

### 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 resistors on the SCK, MOSI line are strictly required! Use 33..60 Ohm resistor in the series on SCK, MOSI line on the board side. You can adjust their values more precisely by observing signal slope on the scope on the encoder side. It should be flat and sharp enough, without spikes caused by signal reflections (see picture):
- 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.



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

### NOTE:

RC\_ROLL input in this version of firmware is occupied by SPI bus, so its special meaning (Sum-PPM, S-bus, Spektrum) is moved to AUX3.



### 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 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 on the “Encoders” tab to 0.
- Power on gimbal, tune PID. Stabilization should work without jitters at this moment.
- Press **CALIB.EL FIELD** button. You have 10 seconds to tilt frame +-20..30 degrees VERY SLOWLY for each axis, several times for all axes.
- On completion, calibration data will be transferred to the GUI and encoders will start working.

#### 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.

#### Reuse 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)

#### 2. Calibrate offset

- IMU should be calibrated properly before this step (accelerometer and gyroscope).
- You may keep gimbal working, or turn motors off. Stabilization is not required.
- Press **CALIB. OFFSET** button. You have 5 seconds to level outer frame and hold it leveled. Camera position is not critical but better to keep it leveled, too. You may fix all by hands, small jitter is not critical.
- On the 5<sup>th</sup> second, IMU's and encoder's angles will be read, offset calibrated and transferred to the GUI.

#### Checking calibration

The “Frame angle” value, displayed in the GUI, should reflect tilting of the outer frame. If outer frame is fixed, the frame angle should not change when you tilt the camera.

#### Reuse of calibration data

You can use this data for other gimbals in batch, if encoder is mounted the same way and has the same offset in the angle. This relates to optical encoder, because for magnetic encoder it's practically impossible to place magnet without calibration.

### 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 setting is 0.5 sec. To disable this feature for any axis, set it to 0. In this case, camera resists against any external disturbance infinite time.



## 4. Calibrating encoders

**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.

From **2.42b7** version of the firmware, a new motor control algorithm is used – simplified (sensorless) 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.

Compared to previous version with equal POWER settings, P and D should be increased 1.75x times. Or you can increase POWER by the same value instead.

**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.

How to estimate this value for any motor by experiment:

- System should be tuned with encoders and should be stable. Set this parameter to 0, it will not affect system operation.
- Rotate frame with moderate speed. You will see that balance marker (blue ball in the GUI) will shift from its center position depending on the speed and direction of rotation.
- Gradually increase this value to minimize balance error (it should not more depend on the frame movements)

This property is for motor only, and does not depend on the other system parameters. It means that the estimated value can be used among different setups with the same motor.

Remarks:

- For a small 14-pole motors, this value is close to zero, and has almost no positive effect.
- For a big 42-pole motors with big induction this value is about 30..50
- This value can help if POWER setting is low, and motor has low KV factor. Otherwise, maximum speed of rotation will be limited by  $KV * \text{Batter Voltage} * \text{POWER} / 255$

**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.

### Choosing motors

To get maximum from you setup, its recommended to choose motors with the lower winding resistance compared to regular brushless motors that are present on the market (most of them designed to work with constant current and have too big resistance). Because there is no DC component of the current in normal operation mode, 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 full battery voltage) and motor gives maximum torque. To limit the amount of time when motor operates under high current, use heating-

## 4. Calibrating encoders

cooling model that will chop current after seconds of work in lock mode.

When designing motor, you need to take into account that the maximum current under full rated voltage in the stall mode (i.e. under pressure) should not cause over-saturation of magnetic field in metal core (a maximum allowed value of magnetic field depends on the core material). Comparing to regular brushless motor, where stall mode is outside of normal operation conditions, in our case, stall mode should be considered as normal operation conditions during limited period of time.

## 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)?**

No, its not supported now. If you are using encoder-enabled firmware, ALL axes should have them installed and configured.

In future, I will think about adding simple I2C encoder reading for YAW only, to the main branch of firmware. It will not be used in FOC, just to estimate YAW angle and provide reliable Follow FPV mode.

**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)