# Scripting Language Reference

# SimpleBGC 32bit

Firmware ver.: 2.44
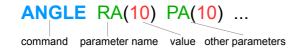Updated: 25.01.2015

# Overview

Scripting language is intended to control a gimbal by user-written program. This program is uploaded to controller and executed there without a connection to PC. Program can be started by a command from remote controller (RC), menu button or GUI button.

There is a simple editor in the SimpleBGC GUI with the syntax checking and tracing of the execution process, but you can use any text editors you like to edit scripts.

# General rules

- Each line can contain only one command
- Command may have any number of named parameters separated by space. The order of parameters is not important. All parameters are optional.
- All values must be in decimal notation with a fractional part separated by a dot. Fractional part can be omitted.
- You can add comments starting line with the # character

Command notation:

**ANGLE** RA(10) PA(10) …

command   parameter name   value  other parameters

Some commands may have one-letter abbreviation.

# Command reference

## A, ANGLE — rotation to the absolute angle with a given speed

Command stops the program and waits until the angle reaches a given value. Rotation is done by the shortest path. If you want to turn the camera on the relative angle greater than ± 180 degrees, use the INC command instead. You can set the speed for each axis. If speed is not set, will be used the current value defined in the settings or command "CONFIG"

Parameters:
```
    RA(10.0) — target angle by the ROLL axis, in degrees
    PA(10.0) - target angle by the PITCH axis
    YA(10.0) - target angle by the YAW axis

    RS(10.0) — speed by the ROLL axis, degree/sec. in range 0..2000
    PS(10.0) - speed by the PITCH axis
    YS(10.0) - speed by the YAW axis
```

Example:
```
    A RA(0) PA(0) YA(0)# go to home position by all axes
    A YA(70) YS(1)     # rotate YAW to 70 degree with the speed 1 degree/sec.
```

## I, INC - rotation by the relative angle with a given speed

Command stops the program and waits until the angle rotates by a given value.  Angle should not exceed two full turns.

Parameters:
```
    RA(10.0) — target angle by the ROLL axis, in degrees. Range -720..720
    PA(10.0) - target angle by the PITCH axis
    YA(10.0) - target angle by the YAW axis

    RS(10.0) — speed by the ROLL axis, degree/sec. in range 0..2000
    PS(10.0) - speed by the PITCH axis
    YS(10.0) - speed by the YAW axis
```

Example:
```
    I YA(270)          # rotate by YAW by 270 degree from current position
```

## S, SPEED - turn with a given speed until it encounters another command SPEED, ANGLE, INC or the end of the program.

The camera starts moving and program execution continues without delay.

Parameters:
```
    RS(10.0) — speed by ROLL axis, degree/sec. Range -2000..2000.
    PS(10.0) - speed by PITCH axis
    YS(10.0) - speed by YAW axis
```

Example:
```
    S YS(5.5)                 # panning by YAW with the speed 5.5 degree/sec
    S RS(0) PS(0) YS(0)       # full stop
```

## R, RESET — set the YAW angle = 0 for the current position

Parameters:
    no

## T, TRIGGER — trigger the state of the output pins

Parameters:
    1(0) — pin ID * and its state: 0 - LOW, 1 — HIGH.
    TIMEOUT(0.01) – program execution is delayed by a given time, in seconds.** At
    the end of the timeout specified pins will automatically return to the previous
    state.

    * Mapping between Pin ID and physical port can be found in the Appendix 1. The pin should be free of other
    functions (for example, not be used as an input in the RC settings tab).
    ** Accuracy is ±1 millisecond. Specifying a timeout can be useful for example, to manually set the shutter
    speed when shooting HDR images.

Example:
```
T 1(1) 2(1)              # switch outputs ID=1,2 to a HIGH state
T 1(1) TIMEOUT(0.020)    # switch output ID=1 to a HIGH state and return back to
LOW after 20ms
```

## SERVO — set PWM signal pulse width on the specified pin

This command is used to control the servos connected to the special pins that supports output of PWM signal. PWM rate can be changed in the GUI (RC - PWM Output - PWM Rate, Hz).

Parameters:
    1(1500) — Servo ID* and pulse duration, microseconds. For regular servo values
are in the range 900-2100. Special value -1 frees pin and makes it high-impedance
input.
    TIMEOUT(1) — program execution is delayed by a given time, in seconds.

    * Mapping between Servo ID and physical port can be found in the Appendix 1. The pin should be free of other
    functions (for example, not be used as an input in the RC settings tab).

Example:
```
SERVO 1(1500) 2(1500)    # move two servos to the middle position
SERVO 1(-1) 2(-1)        # free two outputs
SERVO 1(2000) TIMEOUT(1) # set new value to servo1 and wait 1 second
```

## D, DELAY - delay of program execution

Parameters:
    TIMEOUT(10) — delay, in seconds. Accuracy ±10 milliseconds.

Example:
```
D TIMEOUT(0.300)         # delay for 300 milliseconds
```

## W, WAIT – waiting for the specified angle or speed

Program execution is delayed until all specified conditions will be satisfied, or a given timeout will be expired.

Parameters:
```
      RA(10.0) — angle by the ROLL axis, in degrees
      PA(10.0) — angle by the PITCH axis
      YA(10.0) — angle by the YAW axis
      RS(0)    - speed by the ROLL axis, in degrees/sec.
      PS(0)    - speed by the PITCH axis
      YS(0)    - speed by the YAW axis
      TIMEOUT(10) – max. time of waiting, in seconds. Default is 60 sec.
```

Example:
```
      W YA(30)                # wait for YAW angle=30 degree
      W RA(0) PA(0) YA(0)     # wait for angle=0 for all axes
      W YS(0) TIMEOUT(1)      # wait to stop motion by YAW axis, but not more than 1
      second.
```

## CONFIG — configure some parameters used in program execution

Parameters:
```
      ACC_LIMIT_R(100) — acceleration limit for ROLL axis, degrees/sec²
      ACC_LIMIT_P(100) - acceleration limit for PITCH axis
      ACC_LIMIT_Y(100) - acceleration limit for YAW axis
      SPEED_R(10) — speed for ROLL axis, degree/sec. Range is 0..2000. Initial value is
taken from the "RC SPEED" parameters
      SPEED_P(10) — speed for PITCH axis
      SPEED_Y(10) — speed for YAW axis
```

The changed parameters are not saved to the persistent memory and will be reset to their original values when program is finished.

Example:
```
      C ACC_LIMIT_R(100) SPEED_Y(5) # set an acceleration limit for the ROLL axis to
      100 degree/sec² and a speed for YAW axis to 5 degree/sec.
```

## *Appendix 1. Mapping between pin IDs and physical ports*

| Labeling on the board | Trigger pin ID | Servo ID |
|---|---|---|
| RC_ROLL | 1 | - |
| RC_PITCH | 2 | 3 |
| RC_YAW | 5 | - |
| EXT_FC_ROLL | 3 | 1 |
| EXT_FC_PITCH | 4 | 2 |
| AUX1 | 16 | 4 |
| AUX2 | 17 | - |
| AXU3 | 18 | - |
| BUZZER | 32 | - |

### Example1: Shooting 3-row spherical panorama with increment of 60 degrees in 20 seconds, AUX1 controls the shutter.

```
# Setup a high speed for a quick completion of the panorama
CONFIG SPEED_R(200) SPEED_P(200) SPEED_Y(200) ACC_LIMIT_R(500) ACC_LIMIT_P(1000) ACC_LIMIT_Y(500)
# Reset the origin of the YAW: Start recording with the current azimuth
R
# Turn off the shutter pin
T 16(0)

# Make 1st shot in the zenith position: raise the camera vertically
A RA(0) PA(-90) YA(0)
# Make a shot
T 16(1) TIMEOUT(0.100)

# Tilt 45 degrees up to make the 1st row
A RA(0) PA(-45) YA(0)
T 16(1) TIMEOUT(0.100)
# Panning with increment of 60 degrees clockwise
A YA(60)
T 16(1) TIMEOUT(0.100)
A YA(120)
T 16(1) TIMEOUT(0.100)
A YA(180)
T 16(1) TIMEOUT(0.100)
A YA(240)
T 16(1) TIMEOUT(0.100)
A YA(300)
T 16(1) TIMEOUT(0.100)

# Camera is tilted 0 degrees to make 2nd row
A RA(0) PA(0) YA(0)
T 16(1) TIMEOUT(0.100)
# Panning with increment of 60 degrees counterclockwise (avoid double-twisting of the wires)
A YA(-60)
T 16(1) TIMEOUT(0.100)
A YA(-120)
T 16(1) TIMEOUT(0.100)
A YA(-180)
T 16(1) TIMEOUT(0.100)
A YA(-240)
T 16(1) TIMEOUT(0.100)
A YA(-300)
T 16(1) TIMEOUT(0.100)

# Tilt 45 degrees down to make the 3rd row
A RA(0) PA(45) YA(0)
T 16(1) TIMEOUT(0.100)
# Panning with increment of 60 degrees clockwise
A YA(60)
T 16(1) TIMEOUT(0.100)
A YA(120)
T 16(1) TIMEOUT(0.100)
A YA(180)
T 16(1) TIMEOUT(0.100)
A YA(240)
T 16(1) TIMEOUT(0.100)
A YA(300)
T 16(1) TIMEOUT(0.100)


# Shoot in the nadir position
A YA(0) PA(90)
T 16(1) TIMEOUT(0.100)

# Returns the camera to its original position (untwist wires)
A RA(0) PA(0)
I YA(-360)

### End of program ###
```

## *Example2: Pan at 5 deg/sec by 90 degrees. AUX1 operate the Recording*

```
# Reset the origin of the YAW: Start recording with the current azimuth
R
# Tilt the camera 30 degrees down and level it
A RA(0) PA(30) YA(0)
# Start recording
T 16(1)
# Writing freeze for 3 seconds.
D TIMEOUT(3)
# Setup low acceleration for smooth start and stop of the motion
CONFIG ACC_LIMIT_Y(5)
# Panning with a speed of 5 degree/sec. clockwise
S YS(5)
# Wait until turned by 90 degrees.
W YA(90)
# Stop panning (de-acceleration starts here)
S YS(0)
# Wait until de-acceleration is finished
W YS(0)
# Writing freeze for 3 seconds.
D TIMEOUT(3)
# Stop recording
T 16(0)
### End of program ###
```