Scripting Language Reference

SimpleBGC 32bit

Firmware ver.: 2.5x, 2.6x
Updated: 30. Oct. 2017

Overview

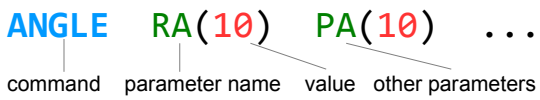
Scripting language is intended to control a gimbal by user-written program. This program is uploaded to controller and executed there without a connection to PC. Program can be started by a command from remote controller (RC), menu button or GUI button.

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:



Some commands may have one-letter abbreviation.

Parameters controlled in real-time*

Values of parameters in any command can be dynamically updated during the execution of a program by linking them to any of RC signal sources. Just specify the signal source name and the desired range of the parameter's value, and RC signal will be captured and mapped to this range at the start of the command. Note that the value is not updated during the command execution. Available RC signal source names are listed in the Appendix 3.

Notation:

```
(<source_name>[<min_value>,<max_value>])
```

Example:

```
ANGLE RA(ADC1[-50,50]) PA(RC_VIRT_CH_1[0,90])
```

* Starting from firmware ver.2.60b3

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

TIMEOUT(60.0) – wait timeout. If not specified, it is 60 seconds by default.
(supported in firmware ver.2.59+)
```

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

TIMEOUT(60.0) – wait timeout. If not specified, it is 60 seconds by default.
(supported in firmware ver.2.59+)
```

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.
TIME(3)** – travel time to target position, in seconds
TIMEOUT(1) – when target is reached, 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).
**Firmware ver. 2.5x and above

Example:

```
SERVO 1(1500) 2(1500)     # move two servos to a middle position
SERVO 1(1000) TIME(3)     # slowly move from 1500 to 1000 in 3 seconds
SERVO 1(2000) TIMEOUT(1)  # set new value and wait 1 second
SERVO 1(-1) 2(-1)         # free two outputs
```

D, DELAY - delay of program execution

Parameters:

```
TIMEOUT(10) – delay, in seconds. Accuracy ±1 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), ACC_LIMIT_P(100), ACC_LIMIT_Y(100) – acceleration limit for the
ROLL, PITCH, YAW axes, degrees/sec²
SPEED_R(10), SPEED_P(10), SPEED_Y(10) – speed for the ROLL, PITCH, YAW axes,
degree/sec. Range is 0..2000. Initial value is taken from the “RC SPEED” parameters
INIT_SYSTEM_ON_FINISH(1) (ver. 2.60b3+) - if set to 1 (default), system will be
re-initialized when script finishes (all parameters and modes of operation will be
returned to default). If set to 0, system will be left in it's current state.
JERK_SLOPE_R(50), JERK_SLOPE_P(50), JERK_SLOPE_Y(50) (ver. 2.66+) – jerk limiter
for the ROLL, PITCH, YAW axis, the rise/fall time in ms
```

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.
```

MENU_CMD — execute a menu command (ver. 2.60b3+)

Parameters:

```
CMD_ID(10) - command ID to run. Full list of commands is listed in the Appendix
2. In case of any command is missed there, it can be found in the Serial API
documentation.  Script parser does not wait for command to be finished!
TIMEOUT(1) - program execution is delayed by a given time, in seconds.
```

SET_ADJ_VAR — set the value of adjustable variable (ver. 2.61+)

Parameters:

NAME(GYRO_TRUST) – name of variable. All names are provided in the SBGC32 User Manual, "Adjustable variables" section.

VALUE(100) – new value to set.

You can specify up to 5 pairs NAME - VALUE in a single command.

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	-

Appendix 2: Menu commands

```
MENU_CMD_NO = 0
MENU_CMD_PROFILE1 = 1
MENU_CMD_PROFILE2 = 2
MENU_CMD_PROFILE3 = 3
MENU_CMD_SWAP_PITCH_ROLL = 4
MENU_CMD_SWAP_YAW_ROLL = 5
MENU_CMD_CALIB_ACC = 6
MENU_CMD_RESET = 7
MENU_CMD_SET_ANGLE = 8
MENU_CMD_CALIB_GYRO = 9
MENU_CMD_MOTOR_TOGGLE = 10
MENU_CMD_MOTOR_ON = 11
MENU_CMD_MOTOR_OFF = 12
MENU_CMD_FRAME_UPSIDE_DOWN = 13
MENU_CMD_PROFILE4 = 14
MENU_CMD_PROFILE5 = 15
MENU_CMD_AUTO_PID = 16
MENU_CMD_LOOK_DOWN = 17
MENU_CMD_HOME_POSITION = 18
MENU_CMD_RC_BIND = 19
MENU_CMD_CALIB_GYRO_TEMP = 20
MENU_CMD_CALIB_ACC_TEMP = 21
MENU_CMD_BUTTON_PRESS = 22
MENU_CMD_RUN_SCRIPT1 = 23
MENU_CMD_RUN_SCRIPT2 = 24
MENU_CMD_RUN_SCRIPT3 = 25
MENU_CMD_RUN_SCRIPT4 = 26
MENU_CMD_RUN_SCRIPT5 = 27
MENU_CMD_CALIB_MAG = 33
MENU_CMD_LEVEL_ROLL_PITCH = 34
MENU_CMD_CENTER_YAW = 35
MENU_CMD_UNTWIST_CABLES = 36
MENU_CMD_SET_ANGLE_NO_SAVE = 37
MENU_HOME_POSITION_SHORTEST = 38
MENU_CENTER_YAW_SHORTEST = 39
```

Appendix 3: RC signal source names

Hardware inputs for PWM format:

```
RC_ROLL_PWM
RC_PITCH_PWM
RC_YAW_PWM
FC_ROLL_PWM
FC_PITCH_PWM
```

RC input should be enabled (i.e. assigned to control of any axis or CMD channel in the RC settings).

Analog inputs (joystick connection):

```
ADC1
ADC2
ADC3
```

Sum-PPM or serial protocols (Spektrum, s-bus):

```
RC_VIRT_CH_1
RC_VIRT_CH_2
..
RC_VIRT_CH_31
```

Desired serial protocol should be selected in the RC settings;
Channels 20..31 have special meaning: they are linked to the sin, cos of the angles of motors.

Serial API virtual channels (can be set by the external devices or mobile applications connected by SBGC Serial API protocol):

```
API_VIRT_CH_1
API_VIRT_CH_2
..
API_VIRT_CH_31
```

Example 1: 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 ###
```

Example 2: Pan at 5 deg/sec by 90 degrees. AUX1 operates 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 ###
```

Example 3: Time-lapse shooting with minimized gyroscope drift

```
# Let system to know that the frame is still, to compensate a drift of gyroscope;
SET_ADJ_VAR NAME(FRAME_HEADING_ANGLE) VALUE(0)
# Set the 'gyro trust' parameter low enough to better compensate drift of gyroscope
SET_ADJ_VAR NAME(GYRO_TRUST) VALUE(60)
# (Optional) move camera to the desired initial position. Skip this command to start from the current
position
#ANGLE PA(0) RA(0)
# Pan left with the speed 0.1 degrees/sec and tilt up with the speed half slower.
SPEED YS(0.1) PS(-0.05)
# Wait 10 minutes
DELAY TIMEOUT(600)
```