



Скриптовый язык для управления подвесом на контроллере SimpleBGC 32bit

Версия прошивки: 2.5х, 2.6х
Последнее изменение: 29 мар, 2018

Назначение

Скриптовый язык предназначен для управления подвесом по заранее написанной программе. Программа загружается в контроллер и выполняется по команде с пульта управления.

В SimpleBGC GUI сделан простой редактор с проверкой синтаксиса и отображением процесса выполнения команды. Но вы можете использовать любой текстовый редактор для написания скриптов.

Общие правила

- На одной строке возможна только одна команда
- У команды можно указать один или несколько именованных параметров, разделенных пробелами. Все параметры команд необязательные, порядок следования не важен.
- Все значения задаются в десятичной записи с дробной частью, отделенной точкой. Дробную часть можно опускать.
- Можно добавлять комментарии, начиная строку со знака #

Команды записываются в виде

ANGLE **RA**(10) **PA**(10) ...
команда имя параметра значение другие параметры

Для некоторых команд можно использовать однобуквенные сокращения.

Динамически управляемые параметры*

Значения параметров в любой команде могут быть динамически заданы во время выполнения программы путем связывания их с любым источником сигнала RC. Для этого задайте имя источника сигнала и желаемый диапазон изменения параметра. Учтите, что значение параметра определяется в момент старта команды и не меняется в процессе ее выполнения. Возможные значения каналов RC указаны в Приложении 3.

Формат задания параметра:

(<имя_источника>[<значение_мин>, <значение_макс>])

Пример:

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

* Начиная с прошивки ver.2.60b3

Команды

A, ANGLE - поворот до указанного абсолютного угла со скоростью, заданной в настройках.

Команда останавливает программу и ждет, пока угол не достигнет заданного значения. Поворот идет по кратчайшему пути. Если нужно повернуть камеру на относительноный угол больше ± 180 градусов, используйте команду INC. Можно задать скорость для каждой оси. Если скорость не задана, будет использовано текущее значение скорости (заданное в настройках или команде CONFIG).

Параметры:

RA(10.0) - новый угол по оси ROLL, градусы

PA(10.0) - новый угол по оси PITCH

YA(10.0) - новый угол по оси YAW

RS(10.0) - скорость по оси ROLL, град/сек. Диапазон 0..2000

PS(10.0) - скорость по оси PITCH

YS(10.0) - скорость по оси YAW

TIMEOUT(60.0) - время ожидания, сек. Если не указан, по умолчанию 60 секунд.
(поддерживается в прошивке 2.59+)

Пример:

A RA(0) PA(0) YA(0) # вернуться в нулевые углы по всем осям

A YA(70) YS(1) # поворот YAW на 70 град. со скоростью 1 градус/сек.

I, INC - поворот на указанный относительный угол со скоростью, заданной в настройках.

Команда останавливает программу и ждет, пока текущий угол не изменится на заданное значение. Угол не должен превышать два полных оборота.

Параметры:

RA(10.0) - угол поворота по оси ROLL, градусы. Диапазон -720..720

PA(10.0) - угол поворота по оси PITCH

YA(10.0) - угол поворота по оси YAW

RS(10.0) - скорость по оси ROLL, град/сек. Диапазон 0..2000

PS(10.0) - скорость по оси PITCH

YS(10.0) - скорость по оси YAW

TIMEOUT(60.0) - время ожидания, сек. Если не указан, по умолчанию 60 секунд.
(поддерживается в прошивке 2.59+)

Пример:

I YA(270) # поворот YAW на 270 град. от текущего значения

S, SPEED - поворот с указанной скоростью, пока не встретится другая команда SPEED, ANGLE, INC или конец программы.

Камера начинает движение с указанной скоростью и выполнение программы продолжается без задержек.

Параметры:

RS(10.0) - скорость по оси ROLL, град/сек. Диапазон -2000..2000.

PS(10.0) - скорость по оси PITCH

YS(10.0) - скорость по оси YAW

Пример:

```
S YS(5.5)           # панорамирование по YAW со скоростью 5.5 градуса/сек.  
S RS(0) PS(0) YS(0) # полная остановка
```

R, RESET - устанавливаем начало отсчета для текущего угла YAW

Текущий угол становится началом отсчета (значение в нем будет равно 0).

Параметры:

нет

T, TRIGGER - переключение состояния указанных выходов

Параметры:

1(0) - ID выхода* и значение 0 - LOW, 1 - HIGH.

TIMEOUT(0.01) - выполнение программы задерживается на указанный время, сек.** По окончании ожидания состояние выхода автоматически вернется в предыдущее значение.

* Соответствие ID физическому выходу см. в справочнике в приложении 1. Выход должен быть свободен от других функций (например, не использоваться как вход в настройках RC).

** Точность отработки интервала ± 1 миллисекунда. Указание тайм-аута может быть полезно для ручной выдержки, например при съемки HDR.

Пример:

```
T 1(1) 2(1)           # переключить выходы ID=1,2 в состояние HIGH  
T 1(1) TIMEOUT(0.020) # переключить выход ID=1 его в состояние HIGH на 20мс
```

SERVO — изменение длины импульса PWM-сигнала на указанном выходе

Команда предназначена для управления сервоприводами, подключенными к выходам, поддерживающим вывод PWM-сигнала. Частота импульсов задается в GUI (RC - PWM Output – PWM Rate, Hz).

Параметры:

1(1500) – ID серво-выхода* и длительность импульса в мкс. Для стандартного сервопривода значения обычно лежат в диапазоне 900-2100. Специальное значение «-1» освобождает выход (переводит в режим входа с высоким импедансом).

TIME(3)** – время, в течение которого совершится перемещение, сек. Если не указан – выход меняется мгновенно.

TIMEOUT(1) – после достижения целевого состояния выхода, выполнение программы останавливается на указанное время, сек.

* Соответствие ID физическому выходу см. в справочнике в приложении 1. Выход должен быть свободен от других функций (например, не использоваться как вход в настройках RC).

** Для прошивки версии 2.50 и выше

Пример:

```
SERVO 1(1500) 2(1500) # установить два сервопривода в среднее положение  
SERVO 1(1000) TIME(3) # плавно изменить положение с 1500 на 1000 за 3 сек.  
SERVO 1(2000) TIMEOUT(1) # установить новое значение и подождать 1 сек.  
SERVO 1(-1) 2(-1) # освободить выходы двух сервоприводов
```

D, DELAY - задержка выполнения программы

Параметры:

TIMEOUT(10) - задержка в секундах. Точность ± 10 миллисекунд.

Пример:

D TIMEOUT(0.300) # задержка на 300 миллисекунд

W, WAIT - ожидание нужного угла или нужной скорости

Выполнение программы останавливается, пока не будет достигнут нужный угол или угловая скорость, или истечет время ожидания. При указании нескольких условий, ждем выполнения всех.

Параметры:

RA(10.0) - угол по оси ROLL, градусы

PA(10.0) - угол по оси PITCH

YA(10.0) - угол по оси YAW

RS(0) - скорость по оси ROLL, град./сек.

PS(0) - скорость по оси PITCH

YS(0) - скорость по оси YAW

TIMEOUT(10) - максимальное время ожидания, сек. Если не указан, ждем 60 сек.

Пример:

W YA(30) # ожидание достижения угла в 30 градусов по YAW
W RA(0) PA(0) YA(0) # ожидание достижения углов в 0 градусов по всем осям
W YS(0) TIMEOUT(1) # ожидание остановки по оси YAW, но не более 1 сек.

CONFIG - изменение некоторых параметров конфигурации

Параметры:

ACC_LIMIT_R(100), **ACC_LIMIT_P**(100), **ACC_LIMIT_Y**(100) - ограничение ускорений для оси ROLL, град./сек²

SPEED_R(10), **SPEED_P**(10), **SPEED_Y**(10) - скорость поворота по ROLL, град/сек. Диапазон 0..2000. По умолчанию берется из настроек.

INIT_SYSTEM_ON_FINISH(1) (ver. 2.60b3+) - если установлено значение 1 (по умолчанию), после окончания работы скрипта все установленные в нем параметры и режимы работы сбрасываются и система переходит в обычный режим. Значение 0 оставляет все параметры и режимы без изменений по выходе из скрипта.

JERK_SLOPE_R(50), **JERK_SLOPE_P**(50), **JERK_SLOPE_Y**(50) (ver. 2.66+) - ограничитель джерков для осей ROLL, PITCH, YAW - время нарастания в мс.

Параметры не сохраняются в постоянную память. При старте программы, все параметры сбрасываются в значения по умолчанию.

Пример:

S ACC_LIMIT_R(100) SPEED_Y(5) # ограничитель ускорений для оси ROLL = 100 град/сек², скорость по YAW = 5 град/сек.

MENU_CMD — выполнить команду меню (ver. 2.60b3+)

Параметры:

CMD_ID(10) - ID команды. Список команд приведен в приложении 2 (в случае отсутствия какой-то команды в списке, ее можно найти в документации Serial API). Парсер

скрипта не ждет завершения выполнения указанной команды!

TIMEOUT(1) – выполнение скрипта задерживается на указанное число секунд.

SET_ADJ_VAR — устанавливает новое значение управляемой переменной (ver. 2.61+)

Параметры:

NAME(GYRO_TRUST) – имя переменной. Все возможные имена перечислены в руководстве пользователя, в разделе "Управляемые переменные".

VALUE(100) – новое значение

Можно указать до пяти пар **NAME** – **VALUE** в одной команде.

Приложение 1. Функции физических выходов

Обозначение на плате	ID выхода 0/1	ID серво-выхода
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	-

Приложение 2: Команды меню

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
MENU_ROTATE_YAW_180 = 40
MENU_ROTATE_YAW_180_FRAME_REL = 41
MENU_SWITCH_YAW_180_FRAME_REL = 42
MENU_SWITCH_POS_ROLL_90 = 43
MENU_START_TIMELAPSE = 44
MENU_CALIB_MOMENTUM = 45
MENU_LEVEL_ROLL = 46
MENU_LOAD_PROFILE_SET1 = 48
MENU_LOAD_PROFILE_SET2 = 49
MENU_LOAD_PROFILE_SET3 = 50
MENU_LOAD_PROFILE_SET4 = 51
MENU_LOAD_PROFILE_SET5 = 52
MENU_LOAD_PROFILE_SET_BACKUP = 53
MENU_INVERT_RC_ROLL = 54
MENU_INVERT_RC_PITCH = 55
MENU_INVERT_RC_YAW = 56
MENU_SNAP_TO_FIXED_POSITION = 57
MENU_CAMERA_REC_PHOTO_EVENT = 58
MENU_CAMERA_PHOTO_EVENT = 59
MENU_MOTORS_SAFE_STOP = 60
MENU_CALIB_ACC_AUTO = 61

Приложение 3: Имена источников сигнала RC

Аппаратные входы PWM формата:

RC_ROLL_PWM
RC_PITCH_PWM
RC_YAW_PWM
FC_ROLL_PWM
FC_PITCH_PWM

RC-вход должен быть активен (т. е. назначен на управление любой осью или каналу CMD в настройках RC).

Аналоговые входы:

ADC1
ADC2
ADC3

Sum-PPM, Spektrum, s-bus:

RC_VIRT_CH_1
RC_VIRT_CH_2
..
RC_VIRT_CH_31

Соответствующий протокол должен быть выбран в настройках RC;
Каналы 20..31 имеют специальное значение: они связаны с функциями \sin , \cos от углов моторов.

Serial API virtual channels:

API_VIRT_CH_1
API_VIRT_CH_2
..
API_VIRT_CH_31

ПРИМЕР1: Снятие трехрядной сферической панорамы с шагом 60 градусов, выход AUX1 управляет затвором

```
# Используем высокую скорость для быстрого завершения панорамы
CONFIG SPEED_R(200) SPEED_P(200) SPEED_Y(200) ACC_LIMIT_R(500) ACC_LIMIT_P(1000) ACC_LIMIT_Y(500)
# Сбрасываем начало отсчета по YAW: начнем съемку с текущего азимута
R
# Выключаем спуск
T 16(0)

# Снимаем зенит: поднимаем камеру вертикально
A RA(0) PA(-90) YA(0)
# делаем кадр
T 16(1) TIMEOUT(0.100)

# Поднимаем камеру на 45 градусов для съема 1-го ряда
A RA(0) PA(-45) YA(0)
T 16(1) TIMEOUT(0.100)
# панорамирование с шагом 60 градусов по часовой стрелке
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)

# 2-й ряд камера смотрит горизонтально
A RA(0) PA(0) YA(0)
T 16(1) TIMEOUT(0.100)
# панорамирование с шагом 60 градусов против часовой стрелки (избегаем закручивания проводов)
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)

# Опускаем камеру на 45 градусов вниз для съема 3-го ряда
A RA(0) PA(30) YA(0)
# панорамирование с шагом 60 градусов по часовой стрелке
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)

# Снимаем надир
A YA(0) PA(90)
T 16(1) TIMEOUT(0.100)

# Возвращаем камеру в исходное положение и раскручиваем провода
A RA(0) PA(0)
I YA(-360)

### КОНЕЦ ПРОГРАММЫ ###
```

ПРИМЕР2: Панорамирование со скоростью 5 град/сек на 90 градусов. AUX1 управляет кнопкой записи

```
# Сбрасываем начало отсчета по YAW: начнем съемку с текущего азимута
R
# наклоняем камеру на 30 градусов вниз
A RA(0) PA(30) YA(0)
# Включаем запись
T 16(1)
# Пишем стоп-кадр 3 сек.
D TIMEOUT(3)
# Выставляем низкое значение ускорения для плавного начала движения и остановки
CONFIG ACC_LIMIT_Y(5)
# Панорамируем со скоростью 5 град./сек по часовой стрелке
S YS(5)
# Ждем поворота на 90 градусов
W YA(90)
# Останавливаем движение.
S YS(0)
# Ждем полной остановки (работает ограничитель ускорений)
W YS(0)
# Пишем стоп-кадр 3 сек.
D TIMEOUT(3)
# останавливаем запись
T 16(0)
### КОНЕЦ ПРОГРАММЫ ###
```

Example 3: Time-lapse shooting with minimized gyroscope drift

```
# 1. Refresh gyroscope calibration
# Turn motors OFF and wait 3 seconds to stop shaking
MENU_CMD CMD_ID(12) TIMEOUT(3)
# Calibrate gyroscope
MENU_CMD CMD_ID(9) TIMEOUT(4)
# Turn motors ON
MENU_CMD CMD_ID(11)

# 2. Let system to know that the frame is still, to compensate a drift of gyroscope;
# It's required to have frame IMU in the 'Above YAW' position or encoders
SET_ADJ_VAR NAME(FRAME_HEADING_ANGLE) VALUE(0)
# Set the 'gyro trust' parameter low enough to better compensate drift
SET_ADJ_VAR NAME(GYRO_TRUST) VALUE(60)

# 3. Move camera to the desired initial position
ANGLE PA(0) RA(0)
# Take a pause
DELAY TIMEOUT(5)

# 4. Start time-lapse shooting
# In this example, we pan left with the speed 0.1 degrees/sec. for 30 degrees and tilt up with the speed 0.05
degrees/sec for 15 degrees.
# We set timeout big enough to prevent the interruption of the command before target is reached.
INC YS(0.1) YA(30) PS(0.05) PA(-15) TIMEOUT(3600)
```