Parameters

The ROSflight firmware has several dozen parameters which it uses to customize performance. Parameters are considered semi-static variables. That is, parameters do not change during flight, but they may change between vehicles. Examples of parameters you may wish to change are:

- Fixed-wing vehicle flag
- PID gains
- Mixer choice
- IMU low-pass filter constant
- RC receiver type

and so on. All parameter access is enabled via ROS services advertised by `rosflight_io` while the flight controller is connected.

Parameter Interface

Getting Parameter Values

Sometimes it is handy to ask the flight controller what the current value of a parameter is. This is accomplished using the `param_get` service. As an example, let's retrieve the roll angle controller P gain.

```
rosservice call /param_get PID_ROLL_ANG_P
```

You should get a response similar to (this happens to be the default value with floating-point error)

```
exists: True
value: 0.15000000596
```

Changing Parameters

Parameters are changed via the `param_set` service. As an example, let's change the roll angle controller P gain. (I will assume that the flight controller is connected and `rosflight_io` is running in the root namespace).

```
rosservice call /param_set PID_ROLL_ANG_P 0.08
```

You should get a prompt from `rosflight_io` saying

```
[ INFO] [1491672408.585339558]: Parameter PID_ROLL_ANG_P has new value 0.08
[ WARN] [1491672408.585508849]: There are unsaved changes to onboard parameters
```

Notice that the parameters have been set, but not saved. Parameter changes take effect immediately, however they will not persist over a reboot unless you *write* them to the non-volatile memory. This brings us to the next task.

Writing Parameters

To ensure that parameter values persist between reboots, you must write the parameters to the non-volatile memory. This is done by calling `param_write`

```
rosservice call /param_write
```

Error

Parameter writing can only happen if the flight controller is disarmed. If the param write failed for some reason, you may want to make sure you are disarmed and try again.

Backing Up and Loading Parameters from File

It is good practice to backup your parameter configuration in case you have to re-flash your firmware or you want to share configurations between vehicles. We can do this via the `param_save_to_file` and `param_load_from_file` services.

First, let's back up our current parameter configuration:

```
rosservice call /param_save_to_file ~/parameters.yml
```

Parameters are saved in YAML format. You must also specify the absolute file name of where you would like your parameters to be saved. The current active set of parameters will be saved, regardless of what is saved in non-volatile memory on the flight controller.

Now, let's say we want to re-load this parameter file

```
rosservice call /param_load_from_file ~/parameters.yml
```

Again, you must specify the absolute file name of the file to be loaded

Fixed-Wing Parameter Configuration

Because ROSflight ships with default parameters for multirotors, you will probably want to change the following parameters if you want to fly a fixed wing aircraft.

Parameter	Description	Type	Fixed Wing Value

Parameter	Description	Type	Fixed Wing Value
MOTOR_PWM_UPDATE	Refresh rate of motor commands to motors and servos (Hz) - See motor documentation	int	50
ARM_SPIN_MOTORS	Enforce MOTOR_IDLE_PWM	int	false
MOTOR_IDLE_THR	min throttle command sent to motors when armed (Set above 0.1 to spin when armed)	float	0.1
ARM_CHANNEL	RC switch channel mapped to arming [0 indexed, -1 to disable]	int	4
FIXED_WING	switches on passthrough commands for fixedwing operation	int	true
MIXER	Which mixer to choose - See Mixer documentation	int	10
ELEVATOR_REV	reverses elevator servo output	int	0/1
AIL_REV	reverses aileron servo output	int	0/1
RUDDER_REV	reverses rudder servo output	int	0/1
CAL_GYRO_ARM	Calibrate gyros when arming - generally only for multirotors	int	false

Description of all Parameters

This is a list of all parameters on ROSflight, their types, default values, and minimum and maximum recommended setting:

Parameter descriptions

Parameter	Description	Type	Default Value	Min	M
BAUD_RATE	Baud rate of MAVlink communication with onboard computer	int	921600	9600	9
SYS_ID	Mavlink System ID	int	1	1	2
STRM_HRTBT	Rate of heartbeat streaming (Hz)	int	1	0	1
STRM_STATUS	Rate of status streaming (Hz)	int	10	0	1
STRM_ATTITUDE	Rate of attitude stream (Hz)	int	200	0	1
STRM_IMU	Rate of IMU stream (Hz)	int	250	0	

Parameter	Description	Type	Default Value	Min	M
STRM_MAG	Rate of magnetometer stream (Hz)	int	50	0	7
STRM_BARO	Rate of barometer stream (Hz)	int	50	0	1
STRM_AIRSPEED	Rate of airspeed stream (Hz)	int	20	0	5
STRM_SONAR	Rate of sonar stream (Hz)	int	40	0	4
STRM_SERVO	Rate of raw output stream	int	50	0	4
STRM_RC	Rate of raw RC input stream	int	50	0	5
PARAM_MAX_CMD	saturation point for PID controller output	float	1.0	0	1
PID_ROLL_RATE_P	Roll Rate Proportional Gain	float	0.070f	0.0	1
PID_ROLL_RATE_I	Roll Rate Integral Gain	float	0.000f	0.0	1
PID_ROLL_RATE_D	Rall Rate Derivative Gain	float	0.000f	0.0	1
PID_PITCH_RATE_P	Pitch Rate Proportional Gain	float	0.070f	0.0	1
PID_PITCH_RATE_I	Pitch Rate Integral Gain	float	0.0000f	0.0	1
PID_PITCH_RATE_D	Pitch Rate Derivative Gain	float	0.0000f	0.0	1
PID_YAW_RATE_P	Yaw Rate Proportional Gain	float	0.25f	0.0	1
PID_YAW_RATE_I	Yaw Rate Integral Gain	float	0.0f	0.0	1
PID_YAW_RATE_D	Yaw Rate Derivative Gain	float	0.0f	0.0	1
PID_ROLL_ANG_P	Roll Angle Proportional Gain	float	0.15f	0.0	1
PID_ROLL_ANG_I	Roll Angle Integral Gain	float	0.0f	0.0	1
PID_ROLL_ANG_D	Roll Angle Derivative Gain	float	0.05f	0.0	1
PID_PITCH_ANG_P	Pitch Angle Proportional Gain	float	0.15f	0.0	1

Parameter	Description	Type	Default Value	Min	M
PID_PITCH_ANG_I	Pitch Angle Integral Gain	float	0.0f	0.0	1
PID_PITCH_ANG_D	Pitch Angle Derivative Gain	float	0.05f	0.0	1
X_EQ_TORQUE	Equilibrium torque added to output of controller on x axis	float	0.0f	-1.0	1
Y_EQ_TORQUE	Equilibrium torque added to output of controller on y axis	float	0.0f	-1.0	1
Z_EQ_TORQUE	Equilibrium torque added to output of controller on z axis	float	0.0f	-1.0	1
PID_TAU	Dirty Derivative time constant - See controller documentation	float	0.05f	0.0	1
MOTOR_PWM_UPDATE	Refresh rate of motor commands to motors - See motor documentation	int	490	0	1
MOTOR_IDLE_THR	min throttle command sent to motors when armed (Set above 0.1 to spin when armed)	float	0.1	0.0	1
FAILSAFE_THR	Throttle sent to motors in failsafe condition (set just below hover throttle)	float	0.3	0.0	1
ARM_SPIN_MOTORS	Enforce MOTOR_IDLE_THR	int	true	0	1
FILTER_INIT_T	Time in ms to initialize estimator	int	3000	0	1
FILTER_KP	estimator proportional gain - See estimator documentation	float	0.5f	0	1
FILTER_KI	estimator integral gain - See estimator documentation	float	0.05f	0	1
FILTER_QUAD_INT	Perform a quadratic averaging of LPF gyro data prior to integration (adds ~20 us to estimation loop on F1 processors)	int	1	0	

Parameter	Description	Type	Default Value	Min	M
FILTER_MAT_EXP	1 - Use matrix exponential to improve gyro integration (adds ~90 us to estimation loop in F1 processors) 0 - use euler integration	int	1	0	1
FILTER_USE_ACC	Use accelerometer to correct gyro integration drift (adds ~70 us to estimation loop)	int	1	0	1
CAL_GYRO_ARM	True if desired to calibrate gyros on arm	int	false	0	1
GYRO_LPF_ALPHA	Low-pass filter constant - See estimator documentation	float	0.3f	0	1
ACC_LPF_ALPHA	Low-pass filter constant - See estimator documentation	float	0.5f	0	1
GYRO_X_BIAS	Constant x-bias of gyroscope readings	float	0.0f	-1.0	1
GYRO_Y_BIAS	Constant y-bias of gyroscope readings	float	0.0f	-1.0	1
GYRO_Z_BIAS	Constant z-bias of gyroscope readings	float	0.0f	-1.0	1
ACC_X_BIAS	Constant x-bias of accelerometer readings	float	0.0f	-2.0	2
ACC_Y_BIAS	Constant y-bias of accelerometer readings	float	0.0f	-2.0	2
ACC_Z_BIAS	Constant z-bias of accelerometer readings	float	0.0f	-2.0	2
ACC_X_TEMP_COMP	Linear x-axis temperature compensation constant	float	0.0f	-2.0	2
ACC_Y_TEMP_COMP	Linear y-axis temperature compensation constant	float	0.0f	-2.0	2

Parameter	Description	Type	Default Value	Min	M
ACC_Z_TEMP_COMP	Linear z-axis temperature compensation constant	float	0.0f	-2.0	2
MAG_A11_COMP	Soft iron compensation constant	float	1.0f	-999.0	9
MAG_A12_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A13_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A21_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A22_COMP	Soft iron compensation constant	float	1.0f	-999.0	9
MAG_A23_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A31_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A32_COMP	Soft iron compensation constant	float	0.0f	-999.0	9
MAG_A33_COMP	Soft iron compensation constant	float	1.0f	-999.0	9
MAG_X_BIAS	Hard iron compensation constant	float	0.0f	-999.0	9
MAG_Y_BIAS	Hard iron compensation constant	float	0.0f	-999.0	9
MAG_Z_BIAS	Hard iron compensation constant	float	0.0f	-999.0	9
BARO_BIAS	Barometer measurement bias (Pa)	float	0.0f	0	i
GROUND_LEVEL	Altitude of ground level (m)	float	1387.0f	-1000	1

Parameter	Description	Type	Default Value	Min	M
DIFF_PRESS_BIAS	Differential Pressure Bias (Pa)	float	0.0f	-10	1
RC_TYPE	Type of RC input 0 - PPM, 1 - SBUS	int	1	0	1
RC_X_CHN	RC input channel mapped to x-axis commands [0 - indexed]	int	0	0	3
RC_Y_CHN	RC input channel mapped to y-axis commands [0 - indexed]	int	1	0	3
RC_Z_CHN	RC input channel mapped to z-axis commands [0 - indexed]	int	3	0	3
RC_F_CHN	RC input channel mapped to F-axis commands [0 - indexed]	int	2	0	3
RC_ATT_OVRD_CHN	RC switch mapped to attitude override [0 indexed, -1 to disable]	int	4	4	7
RC_THR_OVRD_CHN	RC switch channel mapped to throttle override [0 indexed, -1 to disable]	int	4	4	7
RC_ATT_CTRL_CHN	RC switch channel mapped to attitude control type [0 indexed, -1 to disable]	int	-1	4	7
ARM_CHANNEL	RC switch channel mapped to arming (only if PARAM_ARM_STICKS is false) [0 indexed, -1 to disable]	int	-1	4	7
RC_NUM_CHN	number of RC input channels	int	6	1	8
SWITCH_5_DIR	RC switch 5 toggle direction	int	1	-1	1
SWITCH_6_DIR	RC switch 6 toggle direction	int	1	-1	1
SWITCH_7_DIR	RC switch 7 toggle direction	int	1	-1	1

Parameter	Description	Type	Default Value	Min	M
SWITCH_8_DIR	RC switch 8 toggle direction	int	1	-1	1
RC_OVRD_DEV	RC stick deviation from center for overrride	float	0.1	0.0	1
OVRD_LAG_TIME	RC stick deviation lag time before returning control (ms)	int	1000	0	1
MIN_THROTTLE	Take minimum throttle between RC and computer at all times	int	true	0	1
RC_ATT_MODE	Attitude mode for RC sticks (0: rate, 1: angle). Overridden if RC_ATT_CTRL_CHN is set.	int	1	0	1
RC_MAX_ROLL	Maximum roll angle command sent by full deflection of RC sticks	float	0.786f	0.0	3
RC_MAX_PITCH	Maximum pitch angle command sent by full stick deflection of RC sticks	float	0.786f	0.0	3
RC_MAX_ROLLRATE	Maximum roll rate command sent by full stick deflection of RC sticks	float	3.14159f	0.0	9
RC_MAX_PITCHRATE	Maximum pitch command sent by full stick deflection of RC sticks	float	3.14159f	0.0	3
RC_MAX_YAWRATE	Maximum pitch command sent by full stick deflection of RC sticks	float	1.507f	0.0	3
MIXER	Which mixer to choose - See Mixer documentation	int	Mixer::INVALID_MIXER	0	1
FIXED_WING	switches on passthrough commands for fixedwing operation	int	false	0	1
ELEVATOR_REV	reverses elevator servo output	int	0	0	

Parameter	Description	Type	Default Value	Min	M
AIL_REV	reverses aileron servo output	int	0	0	1
RUDDER_REV	reverses rudder servo output	int	0	0	1
FC_ROLL	roll angle (deg) of flight controller wrt to aircraft body	float	0.0f	-180	1
FC_PITCH	pitch angle (deg) of flight controller wrt to aircraft body	float	0.0f	-180	1
FC_YAW	yaw angle (deg) of flight controller wrt to aircraft body	float	0.0f	-180	1
ARM_THRESHOLD	RC deviation from max/min in yaw and throttle for arming and disarming check (us)	float	0.15	0	5