

# ROV PID

#master/university/roben

## List of all parameters in detail

<https://ardupilot.org/sub/docs/parameters-Sub-stable-V4.1.2.html>

### Attitude Control Parameters

#### Roll Axis:

- \* **ATC\_ANG\_RLL\_P**: Proportional gain for roll angle control.
- \* **ATC\_RAT\_RLL\_P**: Proportional gain for roll rate control.
- \* **ATC\_RAT\_RLL\_I**: Integral gain for roll rate control.
- \* **ATC\_RAT\_RLL\_D**: Derivative gain for roll rate control.

#### Pitch Axis:

- \* **ATC\_ANG\_PIT\_P**: Proportional gain for pitch angle control.
- \* **ATC\_RAT\_PIT\_P**: Proportional gain for pitch rate control.
- \* **ATC\_RAT\_PIT\_I**: Integral gain for pitch rate control.
- \* **ATC\_RAT\_PIT\_D**: Derivative gain for pitch rate control.

#### Yaw Axis:

- \* **ATC\_ANG\_YAW\_P**: Proportional gain for yaw angle control.
- \* **ATC\_RAT\_YAW\_P**: Proportional gain for yaw rate control.
- \* **ATC\_RAT\_YAW\_I**: Integral gain for yaw rate control.
- \* **ATC\_RAT\_YAW\_D**: Derivative gain for yaw rate control.

### Position Control Parameters

#### Vertical (Z-axis) Control:

- \* **PSC\_POSZ\_P**: Proportional gain for vertical position control.
- \* **PSC\_VELZ\_P**: Proportional gain for vertical velocity control.
- \* **PSC\_VELZ\_I**: Integral gain for vertical velocity control.
- \* **PSC\_VELZ\_D**: Derivative gain for vertical velocity control.

#### Horizontal (XY-axis) Control:

- \* **PSC\_POSXY\_P**: Proportional gain for horizontal position control.
- \* **PSC\_VELXY\_P**: Proportional gain for horizontal velocity control.
- \* **PSC\_VELXY\_I**: Integral gain for horizontal velocity control.

\* **PSC\_VELXY\_D**: Derivative gain for horizontal velocity control.

Sample Code to READ AND SET a parameter

```
from pymavlink import mavutil
import time

# Establish connection (adjust connection string as needed)
master = mavutil.mavlink_connection('udpin:0.0.0.0:14550')
master.wait_heartbeat()

# Parameter to read and set
param_id = b'ATC_ANG_RLL_P'

# Request current value
master.mav.param_request_read_send(
    master.target_system, master.target_component,
    param_id,
    -1
)
time.sleep(0.1)
message = master.recv_match(type='PARAM_VALUE', blocking=True, timeout=1)
if message:
    print(f"Current value of {message.param_id.decode()}: {message.param_value}")

# Set new value
new_value = 2.5
master.mav.param_set_send(
    master.target_system, master.target_component,
    param_id,
    new_value,
    mavutil.mavlink.MAV_PARAM_TYPE_REAL32
)
time.sleep(0.1)
print(f"Setting {param_id.decode()} to: {new_value}")

# Request and confirm new value
master.mav.param_request_read_send(
    master.target_system, master.target_component,
    param_id,
    -1
)
```

```
time.sleep(0.1)
message = master.recv_match(type='PARAM_VALUE', blocking=True, timeout=1)
if message:
    print(f"New value of {message.param_id.decode()}:
{message.param_value}")
else:
    print(f"Could not retrieve parameter: {param_id.decode()}")
```