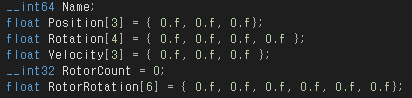
1. 동기화를 위해 필요한 정보



1. Position : 현재 드론의 위치 x, y, z값
   1. PX4: latitude, longitude, alt 추출 가능 (Global 좌표)
   2. OCTOPUS: 3D MAP 내에서의 좌표이며 Global 좌표계가 아님 (즉 PX4 의 좌표계와 불일치)
      1. Settings.json 의 Setoriginpoint 를 활용하여 3D MAP 의 중심에 대한 global 좌표계를 설정한 후 3D MAP 의 orientation, scale 을 실제 공간과 일치하게 사전 튜닝해두어야 함
      2. PX4 로부터 latitude, longitude, alt 정보를 수신한 후, 이를 3D MAP 에 투영할 수 있어야 함

### GLOBAL\_POSITION\_INT ([#33](https://mavlink.io/en/messages/common.html#GLOBAL_POSITION_INT))

[[Message]](https://mavlink.io/en/messages/common.html#messages)The filtered global position (e.g. fused GPS and accelerometers). The position is in GPS-frame (right-handed, Z-up). It is designed as scaled integer message since the resolution of float is not sufficient.

| **Field Name** | **Type** | **Units** | **Description** |
| --- | --- | --- | --- |
| time\_boot\_ms | uint32\_t | ms | Timestamp (time since system boot). |
| lat | int32\_t | degE7 | Latitude, expressed |
| lon | int32\_t | degE7 | Longitude, expressed |
| alt | int32\_t | mm | Altitude (MSL). Note that virtually all GPS modules provide both WGS84 and MSL. |
| relative\_alt | int32\_t | mm | Altitude above ground |
| vx | int16\_t | cm/s | Ground X Speed (Latitude, positive north) |
| vy | int16\_t | cm/s | Ground Y Speed (Longitude, positive east) |
| vz | int16\_t | cm/s | Ground Z Speed (Altitude, positive down) |
| hdg | uint16\_t | cdeg | Vehicle heading (yaw angle), 0.0..359.99 degrees. If unknown, set to: UINT16\_MAX |

1. Rotation : 현재 드론의 각도 쿼터니언 값
   1. OCTOPUS: x, y, z, w
   2. PX4: w, x, y, z

### ATTITUDE\_QUATERNION ([#31](https://mavlink.io/en/messages/common.html#ATTITUDE_QUATERNION))

[[Message]](https://mavlink.io/en/messages/common.html#messages)The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right), expressed as quaternion. Quaternion order is w, x, y, z and a zero rotation would be expressed as (1 0 0 0).

| **Field Name** | **Type** | **Units** | **Description** |
| --- | --- | --- | --- |
| time\_boot\_ms | uint32\_t | ms | Timestamp (time since system boot). |
| q1 | float |  | Quaternion component 1, w (1 in null-rotation) |
| q2 | float |  | Quaternion component 2, x (0 in null-rotation) |
| q3 | float |  | Quaternion component 3, y (0 in null-rotation) |
| q4 | float |  | Quaternion component 4, z (0 in null-rotation) |
| rollspeed | float | rad/s | Roll angular speed |
| pitchspeed | float | rad/s | Pitch angular speed |
| yawspeed | float | rad/s | Yaw angular speed |
| repr\_offset\_q[\*\*](https://mavlink.io/en/messages/common.html" \l "mav2_extension_field" \o "MAVLink2 extension field) | float[4] |  | Rotation offset by which the attitude quaternion and angular speed vector should be rotated for user display (quaternion with [w, x, y, z] order, zero-rotation is [1, 0, 0, 0], send [0, 0, 0, 0] if field not supported). This field is intended for systems in which the reference attitude may change during flight. For example, tailsitters VTOLs rotate their reference attitude by 90 degrees between hover mode and fixed wing mode, thus repr\_offset\_q is equal to [1, 0, 0, 0] in hover mode and equal to [0.7071, 0, 0.7071, 0] in fixed wing mode. |

### ATTITUDE\_QUATERNION\_COV ([#61](https://mavlink.io/en/messages/common.html#ATTITUDE_QUATERNION_COV))

[[Message]](https://mavlink.io/en/messages/common.html#messages)The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right), expressed as quaternion. Quaternion order is w, x, y, z and a zero rotation would be expressed as (1 0 0 0).

| **Field Name** | **Type** | **Units** | **Description** |
| --- | --- | --- | --- |
| time\_usec | uint64\_t | us | Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number. |
| q | float[4] |  | Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation) |
| rollspeed | float | rad/s | Roll angular speed |
| pitchspeed | float | rad/s | Pitch angular speed |
| yawspeed | float | rad/s | Yaw angular speed |
| covariance | float[9] |  | Row-major representation of a 3x3 attitude covariance matrix (states: roll, pitch, yaw; first three entries are the first ROW, next three entries are the second row, etc.). If unknown, assign NaN value to first element in the array. |

1. Velocity : 현재 드론의 3방향 속도 (velocity 를 사용하지 못한다면, 다른 interpolation 방법은?)
   1. OCTOPUS: unit=cm/s (not NED. 대신 unreal 의 x, y, z 사용)
   2. PX4: unit=m/s (NED)

### POSITION\_TARGET\_LOCAL\_NED ([#85](https://mavlink.io/en/messages/common.html#POSITION_TARGET_LOCAL_NED))

[[Message]](https://mavlink.io/en/messages/common.html#messages)Reports the current commanded vehicle position, velocity, and acceleration as specified by the autopilot. This should match the commands sent in SET\_POSITION\_TARGET\_LOCAL\_NED if the vehicle is being controlled this way.

| **Field Name** | **Type** | **Units** | **Values** | **Description** |
| --- | --- | --- | --- | --- |
| time\_boot\_ms | uint32\_t | ms |  | Timestamp (time since system boot). |
| coordinate\_frame | uint8\_t |  | [MAV\_FRAME](https://mavlink.io/en/messages/common.html#MAV_FRAME) | Valid options are: MAV\_FRAME\_LOCAL\_NED = 1, MAV\_FRAME\_LOCAL\_OFFSET\_NED = 7, MAV\_FRAME\_BODY\_NED = 8, MAV\_FRAME\_BODY\_OFFSET\_NED = 9 |
| type\_mask | uint16\_t |  | [POSITION\_TARGET\_TYPEMASK](https://mavlink.io/en/messages/common.html#POSITION_TARGET_TYPEMASK) | Bitmap to indicate which dimensions should be ignored by the vehicle. |
| x | float | m |  | X Position in NED frame |
| y | float | m |  | Y Position in NED frame |
| z | float | m |  | Z Position in NED frame (note, altitude is negative in NED) |
| vx | float | m/s |  | X velocity in NED frame |
| vy | float | m/s |  | Y velocity in NED frame |
| vz | float | m/s |  | Z velocity in NED frame |
| afx | float | m/s/s |  | X acceleration or force (if bit 10 of type\_mask is set) in NED frame in meter / s^2 or N |
| afy | float | m/s/s |  | Y acceleration or force (if bit 10 of type\_mask is set) in NED frame in meter / s^2 or N |
| afz | float | m/s/s |  | Z acceleration or force (if bit 10 of type\_mask is set) in NED frame in meter / s^2 or N |
| yaw | float | rad |  | yaw setpoint |
| yaw\_rate | float | rad/s |  | yaw rate setpoint |

Note) #85 대신 #87로 해결 가능하다면, #87로 1) position 과 3) velocity 가 동시에 충족되지 않을지?

### POSITION\_TARGET\_GLOBAL\_INT ([#87](https://mavlink.io/en/messages/common.html#POSITION_TARGET_GLOBAL_INT))

1. RotorCount : 사용하고 있는 프로펠러 갯수
   1. Heartbeat 을 통해 MAV\_TYPE 을 확인하면 됨

### HEARTBEAT ([#0](https://mavlink.io/en/messages/common.html#HEARTBEAT))

[[Message]](https://mavlink.io/en/messages/common.html#messages)The heartbeat message shows that a system or component is present and responding. The type and autopilot fields (along with the message component id), allow the receiving system to treat further messages from this system appropriately (e.g. by laying out the user interface based on the autopilot). This microservice is documented at https://mavlink.io/en/services/heartbeat.html

| **Field Name** | **Type** | **Values** | **Description** |
| --- | --- | --- | --- |
| type | uint8\_t | [MAV\_TYPE](https://mavlink.io/en/messages/common.html#MAV_TYPE) | Vehicle or component type. For a flight controller component the vehicle type (quadrotor, helicopter, etc.). For other components the component type (e.g. camera, gimbal, etc.). This should be used in preference to component id for identifying the component type. |
| autopilot | uint8\_t | [MAV\_AUTOPILOT](https://mavlink.io/en/messages/common.html#MAV_AUTOPILOT) | Autopilot type / class. Use MAV\_AUTOPILOT\_INVALID for components that are not flight controllers. |
| base\_mode | uint8\_t | [MAV\_MODE\_FLAG](https://mavlink.io/en/messages/common.html#MAV_MODE_FLAG) | System mode bitmap. |
| custom\_mode | uint32\_t |  | A bitfield for use for autopilot-specific flags |
| system\_status | uint8\_t | [MAV\_STATE](https://mavlink.io/en/messages/common.html#MAV_STATE) | System status flag. |
| mavlink\_version | uint8\_t\_mavlink\_version |  | MAVLink version, not writable by user, gets added by protocol because of magic data type: uint8\_t\_mavlink\_version |

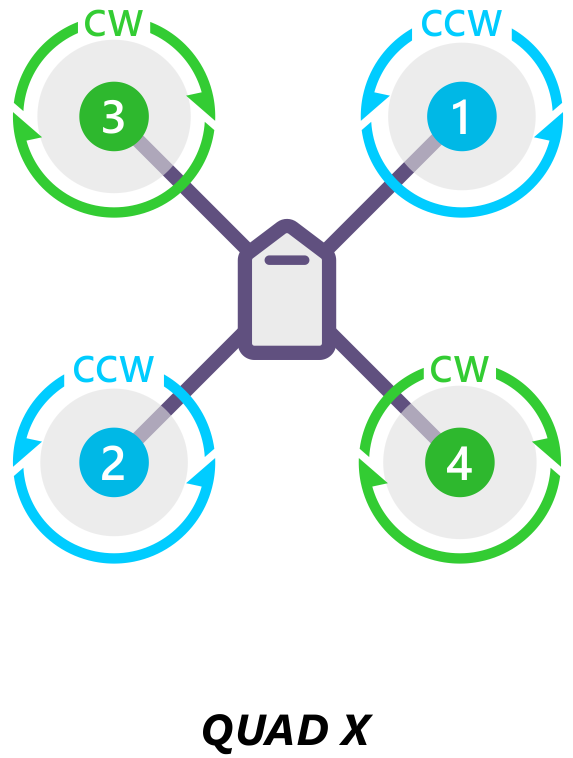
* 1. MAV\_TYPE 은 아래와 같음 (QUAR/HEXA/OCTO-ROTOR 중요)

### [MAV\_TYPE](https://mavlink.io/en/messages/common.html#MAV_TYPE)

[[Enum]](https://mavlink.io/en/messages/common.html#enums)MAVLINK component type reported in HEARTBEAT message. Flight controllers must report the type of the vehicle on which they are mounted (e.g. MAV\_TYPE\_OCTOROTOR). All other components must report a value appropriate for their type (e.g. a camera must use MAV\_TYPE\_CAMERA).

| **Value** | **Field Name** | **Description** |
| --- | --- | --- |
| 0 | [MAV\_TYPE\_GENERIC](https://mavlink.io/en/messages/common.html#MAV_TYPE_GENERIC) | Generic micro air vehicle |
| 1 | [MAV\_TYPE\_FIXED\_WING](https://mavlink.io/en/messages/common.html#MAV_TYPE_FIXED_WING) | Fixed wing aircraft. |
| 2 | [MAV\_TYPE\_QUADROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_QUADROTOR) | Quadrotor |
| 3 | [MAV\_TYPE\_COAXIAL](https://mavlink.io/en/messages/common.html#MAV_TYPE_COAXIAL) | Coaxial helicopter |
| 4 | [MAV\_TYPE\_HELICOPTER](https://mavlink.io/en/messages/common.html#MAV_TYPE_HELICOPTER) | Normal helicopter with tail rotor. |
| 5 | [MAV\_TYPE\_ANTENNA\_TRACKER](https://mavlink.io/en/messages/common.html#MAV_TYPE_ANTENNA_TRACKER) | Ground installation |
| 6 | [MAV\_TYPE\_GCS](https://mavlink.io/en/messages/common.html#MAV_TYPE_GCS) | Operator control unit / ground control station |
| 7 | [MAV\_TYPE\_AIRSHIP](https://mavlink.io/en/messages/common.html#MAV_TYPE_AIRSHIP) | Airship, controlled |
| 8 | [MAV\_TYPE\_FREE\_BALLOON](https://mavlink.io/en/messages/common.html#MAV_TYPE_FREE_BALLOON) | Free balloon, uncontrolled |
| 9 | [MAV\_TYPE\_ROCKET](https://mavlink.io/en/messages/common.html#MAV_TYPE_ROCKET) | Rocket |
| 10 | [MAV\_TYPE\_GROUND\_ROVER](https://mavlink.io/en/messages/common.html#MAV_TYPE_GROUND_ROVER) | Ground rover |
| 11 | [MAV\_TYPE\_SURFACE\_BOAT](https://mavlink.io/en/messages/common.html#MAV_TYPE_SURFACE_BOAT) | Surface vessel, boat, ship |
| 12 | [MAV\_TYPE\_SUBMARINE](https://mavlink.io/en/messages/common.html#MAV_TYPE_SUBMARINE) | Submarine |
| 13 | [MAV\_TYPE\_HEXAROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_HEXAROTOR) | Hexarotor |
| 14 | [MAV\_TYPE\_OCTOROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_OCTOROTOR) | Octorotor |
| 15 | [MAV\_TYPE\_TRICOPTER](https://mavlink.io/en/messages/common.html#MAV_TYPE_TRICOPTER) | Tricopter |
| 16 | [MAV\_TYPE\_FLAPPING\_WING](https://mavlink.io/en/messages/common.html#MAV_TYPE_FLAPPING_WING) | Flapping wing |
| 17 | [MAV\_TYPE\_KITE](https://mavlink.io/en/messages/common.html#MAV_TYPE_KITE) | Kite |
| 18 | [MAV\_TYPE\_ONBOARD\_CONTROLLER](https://mavlink.io/en/messages/common.html#MAV_TYPE_ONBOARD_CONTROLLER) | Onboard companion controller |
| 19 | [MAV\_TYPE\_VTOL\_DUOROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_DUOROTOR) | Two-rotor VTOL using control surfaces in vertical operation in addition. Tailsitter. |
| 20 | [MAV\_TYPE\_VTOL\_QUADROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_QUADROTOR) | Quad-rotor VTOL using a V-shaped quad config in vertical operation. Tailsitter. |
| 21 | [MAV\_TYPE\_VTOL\_TILTROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_TILTROTOR) | Tiltrotor VTOL |
| 22 | [MAV\_TYPE\_VTOL\_RESERVED2](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_RESERVED2) | VTOL reserved 2 |
| 23 | [MAV\_TYPE\_VTOL\_RESERVED3](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_RESERVED3) | VTOL reserved 3 |
| 24 | [MAV\_TYPE\_VTOL\_RESERVED4](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_RESERVED4) | VTOL reserved 4 |
| 25 | [MAV\_TYPE\_VTOL\_RESERVED5](https://mavlink.io/en/messages/common.html#MAV_TYPE_VTOL_RESERVED5) | VTOL reserved 5 |
| 26 | [MAV\_TYPE\_GIMBAL](https://mavlink.io/en/messages/common.html#MAV_TYPE_GIMBAL) | Gimbal |
| 27 | [MAV\_TYPE\_ADSB](https://mavlink.io/en/messages/common.html#MAV_TYPE_ADSB) | ADSB system |
| 28 | [MAV\_TYPE\_PARAFOIL](https://mavlink.io/en/messages/common.html#MAV_TYPE_PARAFOIL) | Steerable, nonrigid airfoil |
| 29 | [MAV\_TYPE\_DODECAROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_DODECAROTOR) | Dodecarotor |
| 30 | [MAV\_TYPE\_CAMERA](https://mavlink.io/en/messages/common.html#MAV_TYPE_CAMERA) | Camera |
| 31 | [MAV\_TYPE\_CHARGING\_STATION](https://mavlink.io/en/messages/common.html#MAV_TYPE_CHARGING_STATION) | Charging station |
| 32 | [MAV\_TYPE\_FLARM](https://mavlink.io/en/messages/common.html#MAV_TYPE_FLARM) | FLARM collision avoidance system |
| 33 | [MAV\_TYPE\_SERVO](https://mavlink.io/en/messages/common.html#MAV_TYPE_SERVO) | Servo |
| 34 | [MAV\_TYPE\_ODID](https://mavlink.io/en/messages/common.html#MAV_TYPE_ODID) | Open Drone ID. See https://mavlink.io/en/services/opendroneid.html. |
| 35 | [MAV\_TYPE\_DECAROTOR](https://mavlink.io/en/messages/common.html#MAV_TYPE_DECAROTOR) | Decarotor |
| 36 | [MAV\_TYPE\_BATTERY](https://mavlink.io/en/messages/common.html#MAV_TYPE_BATTERY) | Battery |

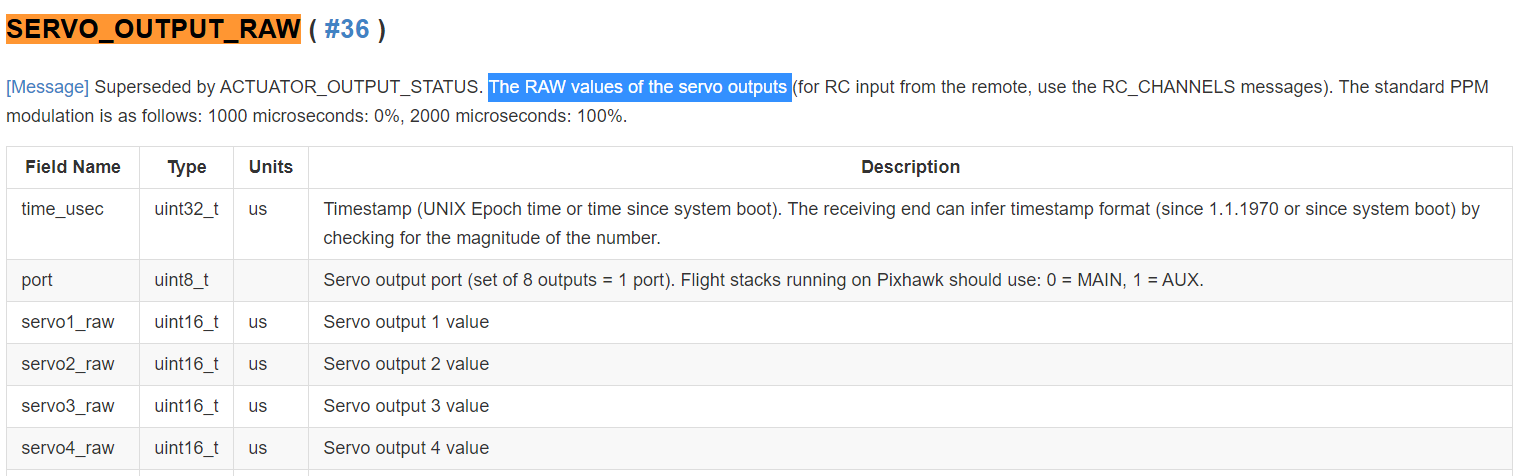
1. RotorRotation : 각 프로펠러의 회전 속도
   1. 아래 표의 Servo -> rpm 변환 식 참조
      1. PX4: us
      2. OCTOPUS: degree/s
   2. 아래와 같이 Octopus 와 PX4 간 index sequence 에 차이가 있으므로 주의 요망
      1. PX4: servo1~4 //실제 로터 1, 2, 3, 4 순서는 아래와 같음



* + 1. OCTOPUS: rot\_components[rotor\_index] //상기 순서와 일치

### SERVO\_OUTPUT\_RAW (#36)

* + <https://mavlink.io/en/messages/common.html>
  + PWM (pulse-width modulation) signal to servo



* Conversion from SERVO\_OUTPUT\_RAW 🡪 motor rotation speed (or, RPM )
  + ServoValue = SERVO\_OUTPUT\_RAW.servo{i}\_raw (range: 900, 1000~2000)

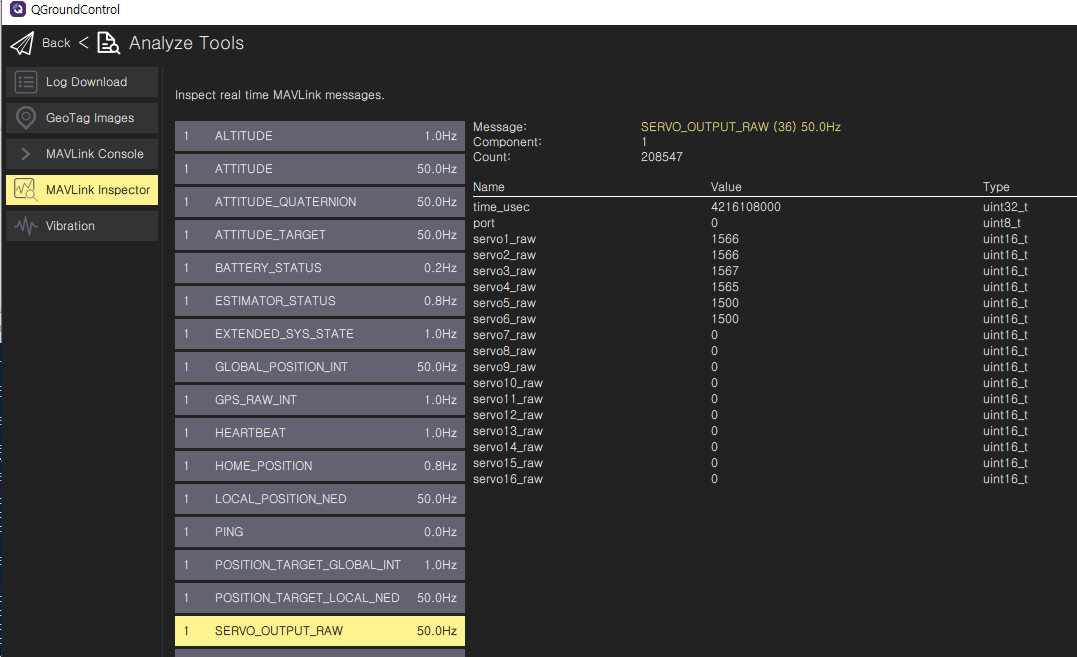
**RotorRotation =**

**if ServoValue = 900**

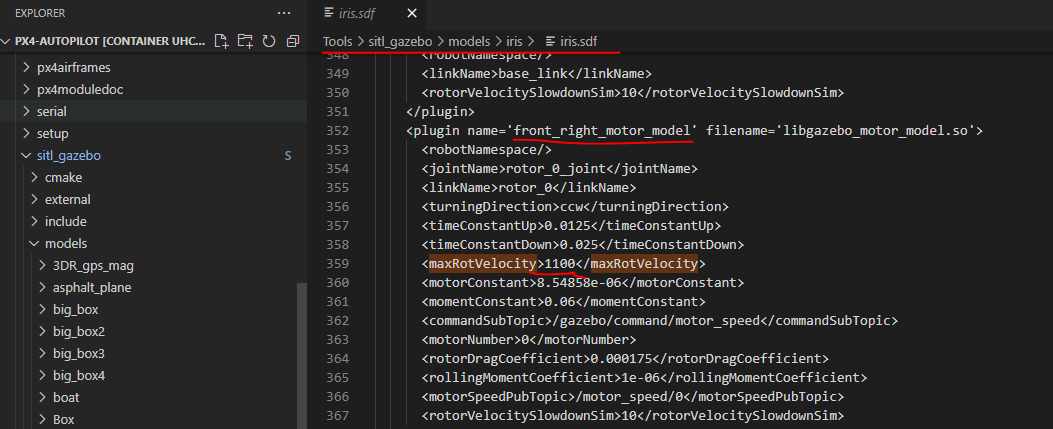
* **MotorRPM = 0 (disarm)**

**elseif (1000<= ServoValue <=2000)**

* **[rad/s] : (ServoValue – 1000 ) \* maxRotVelocity / (1000)**
* **[rpm] : (ServoValue – 1000 ) \* maxRotVelocity \* (60/(2\*pi)) / (1000)**

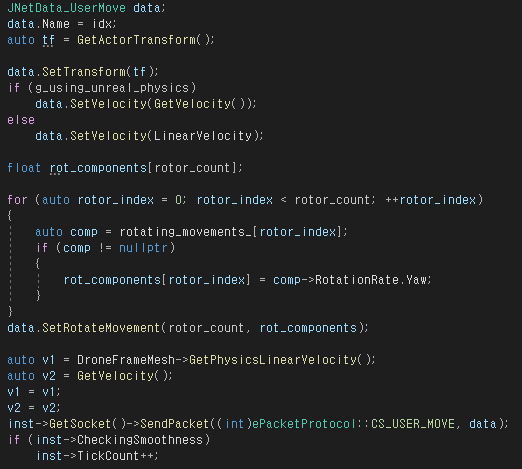


* E.g) Px4\_sitl gazebo의 default model인 iris drone의 specification
  + **maxRotVeolocity = 1100 [ rad/s ]**
  + hovering시 ServoValue =1566 🡪 **MotorRPM = 5954 rpm**

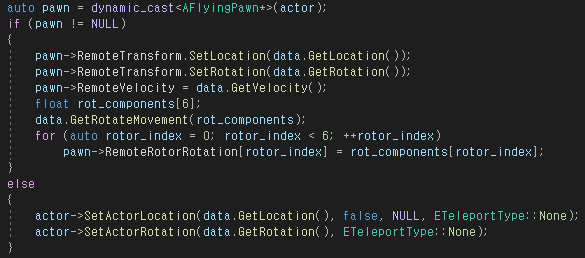


2. 데이터 처리 소스코드

송신파트



수신파트



3. 실제 정보 값

