

GRAPH MODELS AND SIMULATION RESEARCH

# Mavlink and MSP Protocols Overview

*Hesham Hosney*

Supervisor:

Prof. Dr. Umut Durak

October 24, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	System Overview . . . . .	1
1.2	Mavlink Protocol . . . . .	2
1.3	Multiwii Serial Protocol MSP . . . . .	4
1.3.1	Header . . . . .	4
1.3.2	Size . . . . .	4
1.3.3	Type . . . . .	4
1.3.4	Data . . . . .	4
1.3.5	Checksum . . . . .	4
<b>2</b>	<b>Messages Reference</b>	<b>6</b>
2.1	Introduction . . . . .	6
2.1.1	Compatibility Matrix . . . . .	6
2.2	Messages Reference . . . . .	7
2.2.1	HeartBeat 0 . . . . .	7
2.2.2	SYS_STATUS 1 . . . . .	8
2.2.3	GPS_RAW_INT 24 . . . . .	9
2.2.4	RAW_IMU 27 . . . . .	10
2.2.5	ATTITUDE 30 . . . . .	11
2.2.6	RC_CHANNELS_RAW 35 . . . . .	12
2.2.7	SERVO_OUTPUT_RAW 36 . . . . .	13
2.2.8	MISSION_ITEM 39 . . . . .	14
2.2.9	MISSION_SET_CURRENT 41 . . . . .	15
2.2.10	MISSION_CURRENT 42 . . . . .	16
2.2.11	SET_GPS_GLOBAL_ORIGIN 48 . . . . .	17
2.2.12	MISSION_REQUEST_INT 51 . . . . .	18
2.2.13	NAV_CONTROLLER_OUTPUT 62 . . . . .	19
2.2.14	RC_CHANNELS_OVERRIDE 70 . . . . .	20
2.2.15	MISSION_ITEM_INT 73 <b>Preciser than MISSION_ITEM 39</b> . . . . .	21
2.2.16	HIL_STATE 90 . . . . .	22
2.2.17	OPTICAL_FLOW 100 . . . . .	23
2.2.18	RADIO_STATUS 109 . . . . .	24
2.2.19	SERIAL_CONTROL 126 . . . . .	25
2.2.20	HOME_POSITION 242 . . . . .	26
2.2.21	SET_HOME_POSITION 243 . . . . .	27
2.2.22	DEBUG 254 . . . . .	28
2.2.23	MAV_CMD_NAV_WAYPOINT 16 . . . . .	29

<b>3 Conclusion</b>	<b>30</b>
3.1 Summary . . . . .	30
<b>Conclusion</b>	<b>30</b>

# List of Figures

1.1	Flight Stack Overview . . . . .	1
1.2	Mavlink Message Structure . . . . .	3
1.3	MSP Message Structure . . . . .	5

## **Abstract**

In this report, we briefly describe Mavlink and MSP protocols our main aim is to provide structured documentation, which maps the relation between Mavlink and MSP common messages. we will start with an introductory chapter discussing the main flight stack and commonly used technology, an overview of Mavlink Protocol, and Multiwii Serial protocol. chapter 2 we will focus on our main goal which is mapping the messages between the two protocols. We conclude our findings the conclusion chapter.

# Chapter 1

## Introduction

Drone Software developments are gaining business and commercial interests. The industry is set to become the next high growth market with very high potential and exponential growth.

### 1.1 System Overview

In high-level abstraction, the components of the flight stack consist of three main components ground control station, drone, and Communication Layer. Figure 1.1 Describes the system overview.

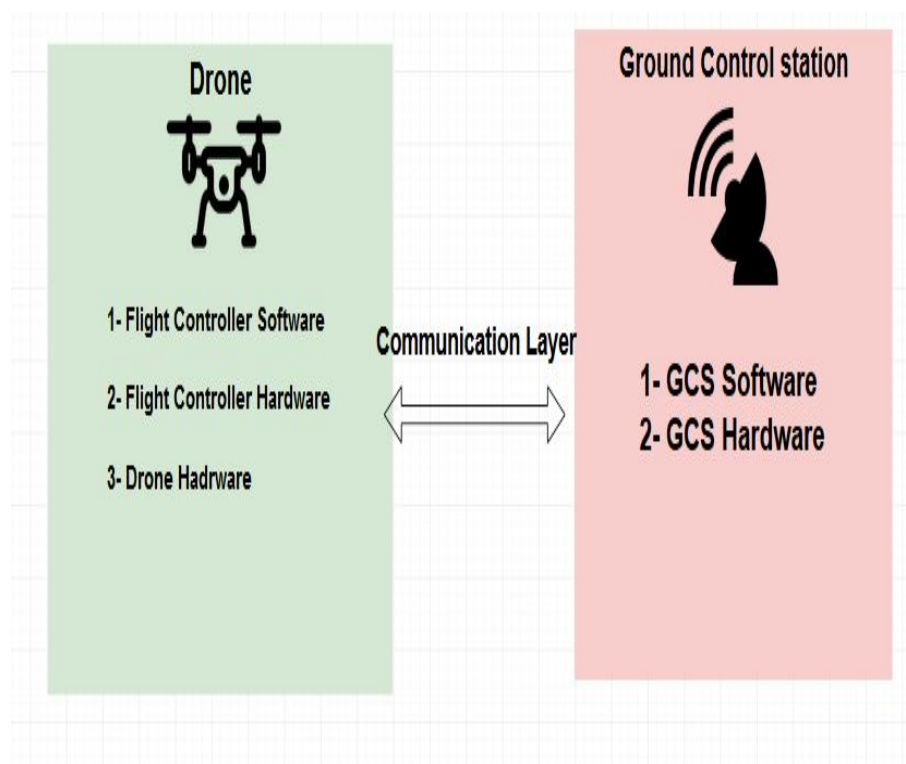


Figure 1.1: Flight Stack Overview

**The Ground Control Station** is mainly an offboard computer that can communicate with the drone to gather information, for instance, location speed battery info. in addition, it's also can send a command to do certain behaviors like landing the drone or directing the drone to a specific destination. It could also be perceived as a computer with USB in a telemetry module there is also software or a user interface layer for doing specific functionality and sending commands using Mavlink communication protocol. for example Dronekit, which is a software for controlling the drone using Python.

**Drone Hardware** consists of motors GPs probes that made up the drone. The Flight Control Hardware is connected to Drone Hardware like Navio2 based on RasperryPi The flight controller software is a compiled code that uploaded to the autopilot hardware that controls the drone's basic components For example, the open source px4 and ardupilot. Additionally, the Drone hardware and the flight controller hardware could be replaced by Software in the Loop SITL it's mainly a simulation running on the computer that facilitates the whole process as a virtual instance this will practice and facilitate testing.

**Communication Layer** is a standard protocol that used for the communication between the ground control station and the drone in a bidirectional way, in another word it is like a standard collection of messages. Mavlink[1] has a standard message structure each Mavlink message has the same message structure allowing the sending node to package the information in a consistent manner and the receiving node to interpret the incoming data consistently. Every message has a message ID which is a number that has an objective meaning. for example, a heartbeat is a message with id 0 when which means the drone is active.

## 1.2 Mavlink Protocol

Mavlink is a communication protocol which allows drones to communicate with ground stations. In other word it facilitates exchanging messages between vehicles such as drones and ground station For instance monitoring the position or performing more actions. Mavlink was released early 2009 by Lorenz Meier under LGPL.

Each Mavlink message consists of 6 bytes for the header and 9 bytes for the payload and 2 bytes for the checksum for verifying the message integrity and assuring that the message wasn't altered during the transmission. The header contains a packet start sign encoded into one byte which indicates the starting of the packet.

- Each message starts with 0xFE indicates the starting of a new message.
- Payload length indicates the length of the following payload.
- Packet sequence for sequencing the packets thus it's a method to detect packet loss.
- System Id to identify the system ground always 1-255 similar to IP address 1 for the drone and 255 for the ground control station.
- Component id to identify the component sending the message inside the system usually zero it's similar to the port number but not widely used.
- Message-id: identify the type of message in the payload for instance 0 is the heartbeat 33 it means the message is carrying out the GPs coordinates.
- Data: payload and it depends on the message-id.
- Last two bytes are for identifying the checksum.

Figure 1.2 Describes Mavlink message structure.

Byte Index	Content	Value	Explanation
0	P a c k e t start sign	v1.0: 0xFE ( v 0 . 9 :	Indicates the start of a new packet.
1	P a y l o a d length	0 - 255	Indicates length of the following payload.
2	P a c k e t sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload "means" and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, <b>excluding packet start sign, so bytes 1..(n+6)</b> Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

Figure 1.2: Mavlink Message Structure



## 1.3 Multiwii Serial Protocol MSP

MSP MultiWii Serial Protocol[2] is the de-facto standard to interact with a MultiWii flight controller (FC). Its implementation contains a list of the most common operations one would expect from a remote control/telemetry point of view. Developers can add custom functionality if required, They are three type of messages in MSP protocol.

- command – is an incoming (into FC) message without implicit outgoing response from the controller
- request – is an incoming message with implicit outgoing response (e.g. a telemetry request sent in by a remote station)
- response – is the outgoing message resulting from an incoming request.

### 1.3.1 Header

The header is three bytes and contains the message start characters \$M and a character showing which direction the message is going. < denotes going to the flight controller (command and request), > denotes coming from the flight controller (response).

### 1.3.2 Size

The fourth byte is the length (in bytes) of the data section. For example, if the data section had three INT 16 variables then the size byte would be 6.

### 1.3.3 Type

The 5th byte is the type of MSP message similar to Mavlink message ID. Value 1xx identify requests while 2xx identify commands. A full list of MSP[2]

### 1.3.4 Data

The data is where all the information is sent. Request messages have no data in them. Commands and responses do, because they contain information.

### 1.3.5 Checksum

The final byte of an MSP message is the checksum. "The checksum is the XOR of size, type and payload bytes". For a request message the checksum is equal to the type.

Figure 1.3 Depicts MSP message structure.

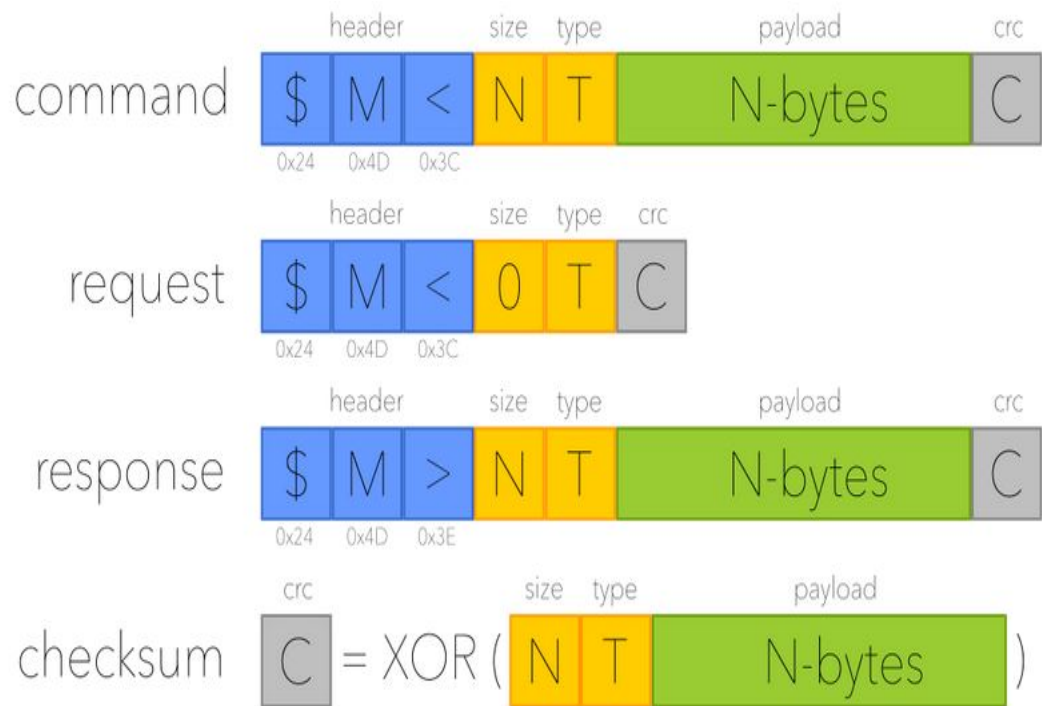


Figure 1.3: MSP Message Structure

## Chapter 2

# Messages Reference

### 2.1 Introduction

In this chapter, we document all Mavlink messages, which have a corresponding MSP message. As a noticed there are way more MAVlink messages that have no corresponding MSP message. We mapped messages on the table as Following.

- For every Mavlink message, we started with the message name and number and description.
- We wrote Mavlink message field name and type in a comma-separated for instance `system_status,u8` this means the field part of the message is `system_status` and the data type representing it is unsigned int 8 bits.
- We mapped the corresponding MSP message which does the same functionality with the message name, Message-ID, Field name(data), and Field type.

#### 2.1.1 Compatibility Matrix

Messages were highlighted with the following colors as an indication of the compatibility between Mavlink and MSP

- Green: this means the messages on both protocol are fully compatible.
- Light Grey: This means they are partially compatible however there are differences in both protocols, for instance, the data type in one protocol is int with 8 bits and unsigned int on the other.
- Yellow: This means there is no compatibility between the messages for example we didn't find an MSP message that corresponds to this particular message on Mavlink.

## 2.2 Messages Reference

### 2.2.1 HeartBeat 0

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).

Mavlink Field(Name,type)	Message	Corresponding MSP Message(Name,id,data,type)	Compatibility	Notes
type,u8		MSP_IDENT,100,MULTIPLY,u8	Yes	-
autopilot		Not Found	No	-
base_mode,u8		No Found	No	-
custom_mode,u32		Not Found	No	-
system_status,u8		Not Found	No	-
mavlink_version,u8		MSP_IDENT,100,MSP_VERSION,u8	Yes	-

### 2.2.2 SYS\_STATUS 1

The general system state. If the system is following the MAVLink standard, the system state is mainly defined by three orthogonal states/modes: The system mode, which is either LOCKED (motors shut down and locked), MANUAL (system under RC control), GUIDED (system with autonomous position control, position setpoint controlled manually) or AUTO (system guided by path/waypoint planner). The NAV\_MODE defined the current flight state: LIFTOFF (often an open-loop maneuver), LANDING, WAYPOINTS or VECTOR. This represents the internal navigation state machine. The system status shows whether the system is currently active or not and if an emergency occurred. During the CRITICAL and EMERGENCY states the MAV is still considered to be active, but should start emergency procedures autonomously. After a failure occurred it should first move from active to critical to allow manual intervention and then move to emergency after a certain timeout.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
onboard_control_sensors_present		Not Found		No	-
onboard_control_sensors_enabled		Not Found		No	-
onboard_control_sensors_health		Not Found		No	-
load,u16		Not Found		No	-
voltage_battery,u16		MSP_ANALOG,110,vbat,u8		Partially	Mavlink u16 ,unit is mv MSP u8, unit 0.1 volt
current_battery,i16		MSP_ANALOG,110,amperage,u16		Partially	Mavlink i16 MSP u16
battery_remaining,i8		MSP_MISC,114,conf.vbatscale,u8		Partially	Mavlink i8, MSP u8
drop_rate_comm,u16		Not Found		No	-
errors_comm,u16		MSP_STATUS,101,i2c_errors_count,u16		Yes	-
errors_count1,u16		Not Found		No	-
errors_count2,u16		Not Found		No	-
errors_count3,u16		Not Found		No	-
errors_count4,u16		Not Found		No	-

### 2.2.3 GPS\_RAW\_INT 24

The global position, as returned by the Global Positioning System (GPS). This is NOT the global position estimate of the system, but rather a RAW sensor value. See message GLOBAL\_POSITION for the global position estimate.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
time_usec,u32		Not Found		No	-
port,u8		Not Found		No	-
fix_type,u8		GPS_FIX,106,GPS_FIX,u8		Yes	-
lat,i32		GPS_coord[LAT],106,GPS_FIX,u32		Partially	Mavlink i32, MSP u32
long,i32		GPS_coord[LON],106,GPS_FIX,u32		Partially	Mavlink i32, MSP u32
alt,i32		GPS_altitude,106,GPS_FIX,u16		Partially	Mavlink i32, MSP u16
eph,u16		MSP_GPSSTATISTICS,-,eph,u16		Yes	-
epv,u16		MSP_GPSSTATISTICS,-,eph,u16		Yes	-
vel,u16		GPS_altitude,106,GPS_speed,u16		Yes	-
cog,u16		Not Found		No	-
satellites_visible,u8		MSP_RAW_GPS,106,GPS_numSat,u8		Yes	-
cog,u16		Not Found		No	-

### 2.2.4 RAW\_IMU 27

The RAW IMU readings for a 9DOF sensor, which is identified by the id (default IMU1). This message should always contain the true raw values without any scaling to allow data capture and system debugging.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
time_usec,u64		Not Found		No	-
xacc,i16		MSP_RAW_IMU,102,accx,i16		Yes	-
yacc,i16		MSP_RAW_IMU,102,accy,i16		Yes	-
zacc,i16		MSP_RAW_IMU,102,accz,i16		Yes	-
xgyro,i16		MSP_RAW_IMU,102,gyrx,i16		Yes	-
ygyro,i16		MSP_RAW_IMU,102,gyry,i16		Yes	-
zgyro,i16		MSP_RAW_IMU,102,gyrz,i16		Yes	-
xmag,i16		MSP_RAW_IMU,102,magx,i16		Yes	-
ymag,i16		MSP_RAW_IMU,102,magy,i16		Yes	-
zmag,i16		MSP_RAW_IMU,102,magz,i16		Yes	-
id,u8		Not Found		No	-
temperature,i16		Not Found		No	-

### 2.2.5 ATTITUDE 30

The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right).

Mavlink Field(Name,type)	Message	Corresponding sage(Name,id,data,type)	MSP	Mes-	Compatibility	Notes
time_boot_ms,u32		Not Found			No	-
roll,f		MSP_ATTITUDE,108,angx,i16			Partially	Mavlink uses f and unit is rad MSP uses i16, unit is 0.1 rad
pitch,f		MSP_ATTITUDE,108,angy,i16			Partially	Mavlink uses f and unit is rad MSP uses i16, unit is 0.1 rad
yaw,f		MSP_ATTITUDE,108,heading,i16			Partially	Mavlink uses f and unit is rad MSP uses i16, unit is 0.1 rad



## 2.2.6 RC\_CHANNELS\_RAW 35

The RAW values of the RC channels received. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. A value of UINT16\_MAX implies the channel is unused. Individual receivers/transmitters might violate this specification.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
time_boot_ms,u32		Not Found		No	-
port,u8		Not Found		No	-
chan1_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan2_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan3_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan4_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan5_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan6_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan7_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
chan8_raw,u16		MSP_RC,105,rcData[RC_CHANS],[u16;16]		Yes	-
rsi,u8		MSP_ANALOG,110,rsi,u16		Partially	Mavlink u8 MSP u16

### 2.2.7 SERVO\_OUTPUT\_RAW 36

The RAW values of the servo outputs (for RC input from the remote, use the RC\_CHANNELS messages). The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100

Mavlink Field(Name,type)	Message	Corresponding sage(Name,id,data,type)	MSP Mes-	Compatibility	Notes
time_usec,u32		Not Found		No	-
port,u8		Not Found		No	-
servo1_raw,u16		MSP_SERVO,103,Servo*8,[u16;16]		Yes	-

### 2.2.8 MISSION\_ITEM 39

Message encoding a mission item. This message is emitted to announce the presence of a mission item and to set a mission item on the system. The mission item can be either in x, y, z meters (type: LOCAL) or x:lat, y:lon, z:altitude. Local frame is Z-down, right handed (NED), global frame is Z-up, right handed (ENU).

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
target_system,u8		Not Found		No	-
target_component,u8		Not Found		No	-
seq,u16		Not Found		No	-
frame,u8		Not Found		No	-
command,u16		Not Found		No	-
autocontinue,u8		Not Found		No	-
param1,f		MSP_SET_WP,209,p1,u16		Partially	Mavlink f MSP u16
param2,f		MSP_SET_WP,209,p2,u16		Partially	Mavlink f MSP u16
param3,f		MSP_SET_WP,209,p3,u16		Partially	Mavlink f MSP i32
param4,f		Not Found		No	-
x,f		MSP_SET_WP,209,lat,i32		Partially	Mavlink f MSP u16
y,f		MSP_SET_WP,209,long,i32		Partially	Mavlink f MSP u16
z,f		MSP_SET_WP,209,altitude,i32		Partially	Mavlink f MSP u16
mission_type		Not Found		No	-

### 2.2.9 MISSION\_SET\_CURRENT 41

Set the mission item with sequence number seq as current item. This means that the MAV will continue to this mission item on the shortest path (not following the mission items in-between).

Mavlink Message Field(Name,type)	Corresponding MSP Message(Name,id,data,type)	Compatibility	Notes
seq,u16	MSP_SET_WP,209,wp_no,u8	Partially	Mavlink u16, MSP u8

### 2.2.10 MISSION\_CURRENT 42

Message that announces the sequence number of the current active mission item. The MAV will fly towards this mission item.

Mavlink Message Field(Name,type)	Corresponding MSP Message(Name,id,data,type)	Compatibility	Notes
seq,u16	MSP_WP,118,wp_no,u8	Partially	Mavlink u16, MSP u8

### 2.2.11 SET\_GPS\_GLOBAL\_ORIGIN 48

Sets the GPS co-ordinates of the vehicle local origin (0,0,0) position. Vehicle should emit GPS\_GLOBAL\_ORIGIN irrespective of whether the origin is changed. This enables transform between the local coordinate frame and the global (GPS) coordinate frame, which may be necessary when (for example) indoor and outdoor settings are connected and the MAV should move from in- to outdoor.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
target_system,u8		Not Found		No	-
latitude,i32		MSP_SET_RAW_GPS,201,lat,u32		Partially	Mavlink i32 MSP u32
longitude,i32		MSP_SET_RAW_GPS,201,lon,u32		Partially	Mavlink i32 MSP u32
altitude,i32		MSP_SET_RAW_GPS,201,alt,u32		Partially	Mavlink i32 MSP u32
time_usec,u64		Not Found		No	-

### 2.2.12 MISSION\_REQUEST\_INT 51

Request the information of the mission item with the sequence number seq. The response of the system to this message should be a MISSION\_ITEM\_INT message.

Mavlink Message Field(Name,type)	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
target_system,u8	Not Found		No	-
target_component,u8	Not Found		No	-
seq,u16	Not Found		No	-
mission_type,u8	Not Found		No	-

### 2.2.13 NAV\_CONTROLLER\_OUTPUT 62

The state of the fixed wing navigation and position controller.

Mavlink Message Field(Name,type)	Corresponding MSP Message(Name,id,data,type)	Mes-	Compatibility	Notes
nav_roll,f	Not Found		No	-
nav_pitch,f	Not Found		No	-
nav_bearing,i16	Not Found		No	-
target_bearing,i16	MSP_NAV_STATUS,-,target_bearing,i16		Yes	-
wp_dist,u16	Not Found		No	-
alt_error,f	Not Found		No	-
aspd_error,f	Not Found		No	-
xtrack_error,f	Not Found		No	-



### 2.2.14 RC\_CHANNELS\_OVERRIDE 70

The RAW values of the RC channels sent to the MAV to override info received from the RC radio. A value of UINT16\_MAX means no change to that channel. A value of 0 means control of that channel should be released back to the RC radio. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. Individual receivers/transmitters might violate this specification.

Mavlink Field(Name,type)	Message	Corresponding sage(Name,id,data,type)	MSP Mes-	Compatibility	Notes
target_system,u8		Not Found		No	-
chan1_raw,u16		MSP_RC,105,rcData[RC_CHANS], [u16;16]		Yes	-

### 2.2.15 MISSION\_ITEM\_INT 73 **Preciser than MISSION\_ITEM 39**

Message encoding a mission item. This message is emitted to announce the presence of a mission item and to set a mission item on the system. The mission item can be either in x, y, z meters (type: LOCAL) or x:lat, y:lon, z:altitude. Local frame is Z-down, right handed (NED), global frame is Z-up, right handed (ENU).

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
target_system,u8		Not Found		No	-
target_component,u8		Not Found		No	-
seq,u16		Not Found		No	-
frame,u8		Not Found		No	-
command,u16		Not Found		No	-
current,u8		Not Found		No	-
autocontinue,u8		Not Found		No	-
param1,f		MSP_SET_WP,209,p1,u16		Partially	Mavlink f MSP u16
param2,f		MSP_SET_WP,209,p2,u16		Partially	Mavlink f MSP u16
param3,f		MSP_SET_WP,209,p3,u16		Partially	Mavlink f MSP i32
param4,f		Not Found		No	-
x,i32		MSP_SET_WP,209,lat,i32		Partially	Mavlink f MSP u16
y,i32		MSP_SET_WP,209,long,i32		Partially	Mavlink f MSP u16
z,f		MSP_SET_WP,209,altitude,i32		Partially	Mavlink f MSP u16
mission_type		Not Found		No	-

### 2.2.16 HIL\_STATE 90

Sent from simulation to autopilot. This packet is useful for high throughput applications such as hardware in the loop simulations.

Mavlink Message Field(Name,type)	Corresponding MSP Message(Name,id,data,type)	Compatibility	Notes
time_usec,u64	Not Found	No	-
roll,f	MSP_HIL_STATE,-,ROLL,u16	Partially	Mavlink f MSP u16
pitch,f	MSP_HIL_STATE,-,PITCH,u16	Partially	Mavlink f MSP u16
yaw,f	MSP_HIL_STATE,-,YAW,u16	Partially	Mavlink f MSP u16
rollspeed,f	Not Found	No	-
pitchspeed,f	Not Found	No	-
yawspeed,f	Not Found	No	-
lat,i32	Not Found	No	-
lon,i32	Not Found	No	-
alt,i32	Not Found	No	-
vx,i16	Not Found	No	-
vy,i16	Not Found	No	-
vz,i16	Not Found	No	-
xacc,i16	Not Found	No	-
yacc,i16	Not Found	No	-
zacc,i16	Not Found	No	-

## 2.2.17 OPTICAL\_FLOW 100

Optical flow from a flow sensor (e.g. optical mouse sensor)

Mavlink Message Field(Name,type)	Corresponding MSP Message(Name,id,data,type)	Mes-	Compatibility	Notes
time_usec,u64	Not Found		No	-
sensor_id,u8	Not Found		No	-
flow_x,i16	-, -,flowRateX,u16		Partially	Mavlink i16 and MSP u16
flow_y,i16	-, -,flowRateY,u16		Partially	Mavlink i16 and MSP u16
flow_comp_m_x,f	Not Found		No	-
flow_comp_m_y,f	Not Found		No	-
quality,u8	Not Found		No	-
ground_distance,f	Not Found		No	-
flow_rate_x,f	Not Found		No	-
flow_rate_y,f	Not Found		No	-

## 2.2.18 RADIO\_STATUS 109

Status generated by radio and injected into MAVLink stream.

Mavlink Field (Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
rss, u8		-, -, localrss, uchar		Partially	Mavlink u8 and uchar MSP
remrss, u8		-, -, remrss, uchar		Partially	Mavlink u8 and uchar MSP
txbuf, u8		-, -, txbuf, uchar		Partially	Mavlink u8 and uchar MSP
noise, u8		-, -, noise, uchar		Partially	Mavlink u8 and uchar MSP
remnoise, u8		-, -, remnoise, uchar		Partially	Mavlink u8 and uchar MSP
rxerrors, u16		-, -, rxerrors, u16		Yes	-
fixed, u16		-, -, fixed_errors, u16		Yes	-

### 2.2.19 SERIAL\_CONTROL 126

Control a serial port. This can be used for raw access to an onboard serial peripheral such as a GPS or telemetry radio. It is designed to make it possible to update the devices firmware via MAVLink messages or change the devices settings. A message with zero bytes can be used to change just the baudrate.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
device,u8		Not Found		No	-
flags,u8		Not Found		No	-
timeout,i16		Not Found		No	-
baudrate,u32		MSP_CF_SERIAL_CONFIG,- ,msp_baudrateIndex,u8		Partially	Mavlink u32 MSP u8
count,u8		Not Found		No	-
data,u[8;70]		Not Found		No	-

## 2.2.20 HOME\_POSITION 242

This message can be requested by sending the MAV\_CMD\_GET\_HOME\_POSITION command. The position the system will return to and land on. The position is set automatically by the system during the takeoff in case it was not explicitly set by the operator before or after. The position the system will return to and land on. The global and local positions encode the position in the respective coordinate frames, while the q parameter encodes the orientation of the surface. Under normal conditions it describes the heading and terrain slope, which can be used by the aircraft to adjust the approach. The approach 3D vector describes the point to which the system should fly in normal flight mode and then perform a landing sequence along the vector.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
latitude,i32		MSP_WP,118,lat,i32		Yes	-
longitude,i32		MSP_WP,118,lon,i32		Yes	-
altitude,i32		MSP_WP,118,altitude,i32		Yes	-
x,f		Not Found		No	-
y,f		Not Found		No	-
z,f		Not Found		No	-
q,f[4]		Not Found		No	-
approach_x,f		Not Found		No	-
approach_y,f		Not Found		No	-
approach_z,f		Not Found		No	-
time_usec,u64		Not Found		No	-

### 2.2.21 SET\_HOME\_POSITION 243

The position the system will return to and land on. The position is set automatically by the system during the takeoff in case it was not explicitly set by the operator before or after. The global and local positions encode the position in the respective coordinate frames, while the q parameter encodes the orientation of the surface. Under normal conditions it describes the heading and terrain slope, which can be used by the aircraft to adjust the approach. The approach 3D vector describes the point to which the system should fly in normal flight mode and then perform a landing sequence along the vector.

Mavlink Field(Name,type)	Message	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
latitude,i32		MSP_SET_WP,209,lat,i32		Yes	-
longitude,i32		MSP_SET_WP,209,lon,i32		Yes	-
altitude,i32		MSP_SET_WP,209,altitude,i32		Yes	-
x,f		Not Found		No	-
y,f		Not Found		No	-
z,f		Not Found		No	-
q,f[4]		Not Found		No	-
approach_x,f		Not Found		No	-
approach_y,f		Not Found		No	-
approach_z,f		Not Found		No	-
time_usec,u64		Not Found		No	-



### 2.2.22 DEBUG 254

Send a debug value. The index is used to discriminate between values. These values show up in the plot of QGroundControl as DEBUG N.

Mavlink Message Field(Name,type)	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
time_boot_ms,u32	Not Found		No	-
ind,u8	Not Found		No	-
value,f	MSP_DEBUG,-,debug,u32		Partially	Mavlink f MSP u32

### 2.2.23 MAV\_CMD\_NAV\_WAYPOINT 16

Navigate to waypoint.

Mavlink Message Field(Name,type)	Corresponding MSP sage(Name,id,data,type)	Mes-	Compatibility	Notes
Hold,undefined	Not found		No	-
Pass Radius,undefined	Not found		No	-
Accept Radius,undefined	Not found		No	-
Yaw,undefined	Not found		No	-
Latitude,undefined	MSP_SET_WP,209,lat,u32		Partially	-
Longitude,undefined	MSP_SET_WP,209,long,u32		Partially	-
altitude,undefined	MSP_SET_WP,209,Althold,u32		Partially	-

## Chapter 3

# Conclusion

### 3.1 Summary

In this report, we discussed the relation between Mavlink and Multiwii Serial Protocol MSP. We aimed to map the message relation between the two protocols. We started by a brief discussion and the prerequisite background about the two protocols and finally, we listed all the messages that did a similar functionality highlighting the compatibility matrix between the two protocols.

# Bibliography

- [1] Mavlink Protocol Messages <https://mavlink.io/en/messages/common.html>
- [2] Mutliwii Serial Protocol [http://www.mutliwii.com/wiki/index.php?title=Multiwii\\_Serial\\_Protocol](http://www.mutliwii.com/wiki/index.php?title=Multiwii_Serial_Protocol)
- [3] Mutliwii Serial Protocol <https://github.com/iNavFlight/inav/wiki/MSP-V2#msp-v2-message-catalogue>
- [4] Mutliwii Serial Protocol 2 <https://github.com/iNavFlight/inav/wiki/MSP-V2#msp-v2-message-catalogue>
- [5] Mutliwii Serial Protocol <http://www.stefanocottafavi.com/msp-the-multiwii-serial-protocol/>