MAVLink Provider

Version 1.0.0

User's guide

February 14, 2022

【Remarks】

【Revision History】

Version	Date	Content
1.0.0	2022-2-14	First edition

Contents

1. Introduction

This document is a user's guide for the CAO provider "MAVLink Provider" that executes messages and commands using the MAVLink (Micro Air Vehicle Link) protocol with CAOProv.FACOM.MAVLink_CLIENT.dll for drones on the network.

This document describes the features of this MAVLink provider and the implemented methods..

For more information on MAVLink, please refer to the "MAVLink Developer Guide" from the link below.

※https://mavlink.io/en/

2. Overview of provider

2.1. Overview

The MAVLink provider connects to the drone on the network via UDP or TCP / IP communication, sends messages and commands using the MAVLink protocol, and controls the drone.

The MAVLink provider is targeted for version 2.0 of the MAVLink protocol.

To use the MAVLink provider, you need to install the ORiN SDK or manually register the registry by referring to the following.

Table 2-1 MAVLink_CLIENT provider

File name	CaoProv.FAcom.MAVLink_CLIENT.dll
ProgID	CaoProv.FAcom.MAVLink_CLIENT
Registry registration	%ORIN2%\DotNet\BAT\RegistAsm.bat CaoProv.FAcom.MAVLink_CLIENT.dll
Remove registry registration	%ORIN2%\DotNet\BAT\UnregistAsm.bat CaoProv.FAcom.MAVLink_CLIENT.dll

2.2. Method and Property

2.2.1. CaoWorkspace::AddControllermethod

The MAVLink provider performs the initial processing of socket communication when the Controller object is created. Specify the drone with an option string when connecting.

Syntax **AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR> <bstrPcName:BSTR>[,<bstrOption:BSTR>])**

bstrCtrlName : [in] Controller name
bstrProvName : [in] Provider name. Fixed value="CaoProv.FAcom.MAVLink_CLIENT"
bstrPcName : [in] Machine name where provider operates
bstrOption : [in] Option character string

Table 2-2 Option character string of CaoWorkspace::AddController

Option	Description
IP=<IP address>	Set the IP address of the drone to connect to. IP address supports only IPv4 format.
Port=<port number>	Set the port number for socket communication with the drone.
ConnMode=<communication method>	Set the protocol used for connection. 0 : UDP 1 : TCP/IP
Debug=<debug mode>	Set the availability of debug mode. 0 : don't use 1 : use (default:0)

2.2.2. CaoController::Execute method

Execute the command for the connected drone. For details on the commands, refer to "3.1 Command List".

Format Execute(<bstrCommand:BSTR>, <vntParam:ARRAY | BSTR>)

bstrCommand : [in] command name
vntParam : [in] parameter

2.2.3. CaoController::AddVariable method

Create a variable object to get system version information and manufacturer information.

Format AddVariable(<bstrName:BSTR> [,<bstrOption:BSTR>])

bstrName :[in] command name
bstrOption :[in] Option character string

Table 2-3 List of system variables

Variable Name	Data type	Description	Attribute	
			get	put
@MAKER	VT_BSTR	Returns the manufacturer string of the system.	○	-
@VERSION	VT_BSTR	Returns the system version string.	○	-
QUEUECNT	VT_I4	Returns the number of queues for received data.	○	-

2.2.4. CaoController::OnMessage event

When the MAVLink provider receives telemetry from the drone, it passes the data as an OnMessage event of the CaoController class.

For the telemetry notified by the OnMessage event, refer to "3.2 Telemetry List".

3. Command reference

3.1. Command list

Below is a list of commands that can be used with the MAVLink provider.

An error will occur if a command not listed is set.

Command names are case insensitive.

All command types are character type (VT_BSTR). All response types to commands is an array of object types.

See the following sections for details on each command.

Table 3-1 Command list

Command	Description
TAKEOFF	Send takeoff instructions.
LAND	Send landing instructions.
MISSION_COUNT	・Send notification of flight plan upload start. ・Delete the uploaded flight plan.
MISSION_ITEM_INT	Information such as waypoint details and speed specifications will be sent.
SET_MODE	Sends a mode change request.
DO_REPOSITION	Send a stop instruction to the drone in flight.
DO_CHANGE_SPEED	Send a speed change instruction.
COMPONENT_ARM_DISARM	Sends motor start / stop instructions.
DISCONNECT	Disconnect from the drone.

3.1.1. CaoController::Execute("TAKEOFF") command

Send takeoff instructions.

Format Execute (<TAKEOFF>, <pitch, yaw, latitude, longitude, altitude>)

The details of each parameter of TAKEOFF are as follows.

TABLE 3-2 TAKEOFF Parameter details

Parameter	Unit	Description
pitch	deg	Airframe pitch.
yaw	deg	Airframe yaw.
latitude	-	Latitude after takeoff. Set non-numerical value when taking off on the spot.
longitude	-	Longitude after takeoff. Set non-numerical value when taking off on the spot.
altitude	m	Altitude.

Return value COMMAND_ACK

If the TAKEOFF transmission is successful, a COMMAND_ACK message will be returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-3 Response to TAKE OFF

Parameter	Value	Description
message ID	77	ID for each message.
command ID	22	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.2. CaoController::Execute("LAND") command

Send landing instructions.

Format Execute (<LAND>, <abort alt, land mode, yaw, latitude, longitude, altitude>)

The details of each parameter of LAND are as follows.

Table 3-4 LAND Parameter details

Parameter	Unit	Description
abort alt	m	Minimum target altitude in case of landing failure. If "0", the system default value is used.
land mode	-	Landing mode.
yaw	deg	Airframe yaw.
latitude	-	Latitude of landing point. Set a non-numerical value when landing on the spot.
longitude	-	Longitude of landing point。 Set a non-numerical value when landing on the spot.
altitude	m	Landing point altitude。

Return value COMMAND_ACK

If the LAND is successfully sent, a COMMAND_ACK message is returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-5 Response of LAND

Parameter	Value	Description
message ID	77	ID for each message.
command ID	21	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.3. CaoController::Execute("MISSION_COUNT") command

This command can use the following two functions depending on the parameters.

・Send notification of flight plan upload start

・Delete the uploaded flight plan.

Format Execute (<MISSION_COUNT>, <count, mission type>)

Details of each parameter of MISSION_COUNT are as follows.

Table 3-6 MISSION_COUNT Parameter details

Parameter	Unit	Description
count	-	Number of mission items to send.
mission type	-	The type of mission.

Return Value Response to upload start notification
MISSION_REQUEST_INT

If the upload start notification is sent successfully, the MISSION_REQUEST_INT message will be returned as a response. See the list below for more information on MISSION_REQUEST_INT.

Table 3-7 Response to upload start notification

Parameter	Value	Description
message ID	51	ID for each message.
target system		Command receiver system ID.
target component		The component ID of the command receiver.
count		The number of mission items received.
mission type		The type of mission.

Response to flight plan deletion

MISSION_ACK

If the flight plan deletion is successfully sent, a MISSION_ACK message will be returned in response. See the list below for more information on MISSION_ACK.

Table 3-8 Response to flight plan deletion

Parameter	Value	Description
message ID	47	ID for each message.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.
type		Mission execution result.
mission type		The type of mission.

3.1.4. CaoController::Execute("MISSION_ITEM_INT") command

Information such as waypoint details and speed specifications will be sent.

See Microservice – Mission Protocol in the MAVLink Developer Guide for more information.

※https://mavlink.io/en/services/mission.html

Format Execute (<MISSION_ITEM_INT>,<seq, frame, command, current, autocontinue, param1, param2, param3, param4, x, y, z, mission type>)

Details of each parameter of MISSION_ITEM_INT are as follows.

Table 3-9 MISSION_ITEM_INT Parameter details

Parameter	Unit	Description
seq	-	Waypoint ID. Set from 0 to serial numbers.
frame	-	Coordinate system.
command	-	The ID of the command to execute.
current	-	Current setting. Set "1" for the first item to be executed in the flight plan. Set "0" for items other than the above.
autocontinue	-	Automatic continuation setting. Set "0" to prevent the next item from being executed automatically. Set "1" to make it automatic.
param1	-	Parameter 1 of each command.
param2	-	Parameter 2 of each command.
param3	-	Parameter 3 of each command.
param4	-	Parameter 4 of each command.
x	-	Parameter 5 of each command.
y	-	Parameter 6 of each command.
z	-	Parameter 7 of each command.
mission type	-	The type of mission.

Return Value If the information you sent is not the last item in your flight plan
MISSION_REQUEST_INT

As a response if the information you sent is not the last item in your flight plan The MISSION_REQUEST_INT message is returned. See the list below for more information on MISSION_REQUEST_INT.

Table 3-10 Response if not the last item

Parameter	Value	Description
message ID	51	ID for each message.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.
count		The number of mission items received.
mission type		The type of mission.

If the information you sent is the last item in your flight plan
MISSION_ACK

If the information you sent is the last item in the flight plan, a MISSION_ACK message will be returned in response. See the list below for more information on MISSION_ACK.

Table 3-11 Response if it is the last item

Parameter	Value	Description
message ID	47	ID for each message.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.
type		Mission execution result.
mission type		The type of mission.

3.1.5. CaoController::Execute("SET_MODE") command

Sends a mode change request.

Format Execute (<SET_MODE>, <base mode, custom mode>)

The details of each parameter of SET_MODE are as follows.

Table 3-12 SET_MODE Parameter details

Parameter	Unit	Description
base mode	-	Base mode.
custom mode	-	Autopilot-specific mode.

Return Value COMMAND_ACK

If the SET_MODE transmission is successful, a COMMAND_ACK message is returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-13 Response to SET_MODE

Parameter	Value	Description
message ID	77	ID for each message.
command ID	176	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.6. CaoController::Execute("DO_REPOSITION") command

Send a stop instruction to the drone in flight.

Format Execute (<DO_REPOSITION>, < speed, bitmask, yaw, latitude, longitude, altitude>)

The details of each parameter of DO_REPOSITION are as follows.

Table 3-14 DO_REPOSITION Parameter details

Parameter	Unit	Description
speed	m/s	Ground speed. If "-1" is set, the default value will be used.
bitmask	-	Bit mask for option flags.
yaw	deg	Airframe yaw.
latitude	-	Latitude of the stop position. If you want to stop on the spot, set a non-numeric value.
longitude	-	Longitude of stop position. If you want to stop on the spot, set a non-numeric value.
altitude	m	Altitude of the stop position. If you want to stop on the spot, set a non-numeric value.

Return Value COMMAND_ACK

If the DO_REPOSITION is successfully sent, a COMMAND_ACK message will be returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-15 Response to DO_REPOSITION

Parameter	Value	Description
message ID	77	ID for each message.
command ID	192	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.7. CaoController::Execute("DO_CHANGE_SPEED") command

Send speed change instructions.

Format Execute (<DO_CHAMGE_SPEED>, <speed type, speed, throttle, relative>)

The details of each parameter of DO_CHANGE_SPEED are as follows.

Table 3-16 DO_CHANGE_SPEED Parameter details

Parameter	Unit	Description
speed type	-	The type of speed.
speed	m/s	speed. If "-1" is set, it will not be changed.
throttle	%	The amount of throttle. If "-1" is set, it will not be changed.
relative	-	Evaluation methods. Absolute if "0" is set. Relative when "1" is set.

Return Value COMMAND_ACK

If DO_CHANGE_SPEED is sent successfully, a COMMAND_ACK message will be returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-17 Response to DO_CHANGE_SPEED

Parameter	Value	Description
message ID	77	ID for each message.
command ID	178	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.8. CaoController::Execute("COMPONENT_ARM_DISARM") command

Sends motor start or stop instructions

Format Execute (<COMPONENT_ARM_DISARM>, <arm, force>)

The details of each parameter of COMPONENT_ARM_DISARM are as follows.

Table 3-18 COMPONENT_ARM_DISARM Parameter details

Parameter	Unit	Description
arm	-	Motor start / stop setting. Stop if "0" is set. Start when "1" is specified.
force	-	Operation forced setting. If "0" is set, it will be executed only in the situation where fail-safe is not applied. If "21196" is set, it will be forcibly executed.

Return Value COMMAND_ACK

If COMPONENT_ARM_DISARM is successfully sent, a COMMAND_ACK message is returned as a response. See the list below for more information on COMMAND_ACK.

Table 3-19 Response to DO_CHANGE_SPEED

Parameter	Value	Description
message ID	77	ID for each message.
command ID	400	The ID of the command in the request for the response.
result		Execution result of the sent command.
progress		The reason why the command execution failed.
result param 2		Additional content of result.
target system		The system ID of the command receiver.
target component		The component ID of the command receiver.

3.1.9. CaoController::Execute("DISCONNECT") command

Disconnect from the drone.

Format Execute (<DISCONNECT>)

DISCONNECT requires no parameters.

Return Value None

If DISCONNECT is sent successfully, there will be no response with MAVLink specific messages. Returns the following string as a response.

"DISCONNECT"

3.2. Telemetry list

Below is a list of telemetry (messages) notified by the MAVLink provider's OnMessage event. See the following sections for details on each message.

Table 3-20 Telemetry list

Message	Description
HEARTBEAT	Confirms the existence of the communication partner and notifies that it is responding.
BATTERY_STATUS	Notify you of information about the battery.
GPS_RAW_INT	Notify GPS related information.
GLOBAL_POSITION_INT	Notify information about coordinates.
EXTENDED_SYS_STATE	Notify the takeoff and landing status of the aircraft.
MISSION_ITEM_REACHED	Notify you of information regarding the completion of flight plan execution.
MISSION_CURRENT	Notify you of information about the flight plan in progress.

3.2.1. CaoController::OnMessage("HEARTBEAT")

Confirms the existence of the communication partner and notifies that it is responding. It holds the version of the MAVLink protocol and the state of the system.

The details of the data received by HEARTBEAT are as follows.

Table 3-21 HEARTBEAT details

Parameter	Type	Description
custom mode	uint32_t	Bitfield used for autopilot-specific flags.
type	uint8_t	The type of aircraft.
autopilot	uint8_t	Types of autopilot
base mode	uint8_t	System mode bitmap.
system status	uint8_t	System status.
mavlink version	uint8_t_mavlink_version	MAVLink version.

3.2.2. CaoController::OnMessage("BATTERY_STATUS")

Notifies you of information about the battery. It holds information such as battery type and voltage.

The details of the data received by BATTERY_STATUS are as follows.

Table 3-22 BATTERY_STATUS details

Parameter	Type	Unit	Description
current consumed	int32_t	mAh	Discharge capacity. In case of -1, it is not measured.
energy consumed	int32_t	hJ	amount of work. In case of -1, it is not measured.
temperature	int16_t	cdegC	Battery temperature. If unknown, INT16_MAX is set.
voltages	uint16_t[10]	mV	Battery voltage for cells 1-10. If the individual voltage is unknown, the voltage of the entire battery is set to cell 0 and UINT16_MAX is set to all other cells.
current battery	int16_t	cA	Battery current. In case of -1, the current is not measured.
id	uint8_t		Battery ID.
battery function	uint8_t		Battery function.
type	uint8_t		Battery type.
battery remaining	int8_t	%	The remaining battery capacity. If it is -1, the remaining amount is not measured.
time remaining	int32_t	s	Battery remaining time. In case of -1, the remaining time is not estimated.
charge state	uint8_t		The state of the degree of discharge.
voltage ext	uint16_t[4]	mV	Voltage of cells 11-14. If it is 0, it means that the cell does not exist.

3.2.3. CaoController::OnMessage("GPS_RAW_INT")

Notifies you of GPS related information. It holds information such as latitude, longitude, and the number of supplementary satellites.

The details of the data received by GPS_RAW_INT are as follows.

Table 3-23 GPS_RAW_INT details

Parameter	Type	Unit	Description
time usec	uint64_t	us	Elapsed time since the system booted.
latitude	int32_t	degE7	latitude.
longitude	int32_t	degE7	longitude.
altitude	int32_t	mm	Altitude.
eph	uint16_t		GPS HDOP horizontal dilution position. If unknown, UINT16_MAX is set.
epv	uint16_t		GPS VDOP vertical dilution position. If unknown, UINT16_MAX is set.
vel	uint16_t	cm/s	GPS ground speed. If unknown, UINT16_MAX is set.
cog	uint16_t	cdeg	Direction of movement. If unknown, UINT16_MAX is set.
fix type	uint8_t		GPS correction type.
satellites visible	uint8_t		Supplementary number of satellites. If unknown, UINT8_MAX is set.
alt ellipsoid	int32_t	mm	Altitude.
h acc	uint32_t	mm	Positional uncertainty.
v acc	uint32_t	mm	High degree of uncertainty.
vel acc	uint32_t	mm	Speed uncertainty.
hdg acc	uint32_t	degE5	Uncertainty of orientation.

3.2.4. CaoController::OnMessage("GLOBAL_POSITION_INT")

Notifies you of information about coordinates. It holds information such as latitude, longitude, and altitude.

The details of the data received by GLOBAL_POSITION_INT are as follows.

Table 3-24 GLOBAL_POSITION_INT details

Parameter	Type	Unit	Description
time boot ms	uint32_t	ms	Elapsed time since the system booted.
latitude	int32_t	degE7	latitude.
longitude	int32_t	degE7	longitude.
altitude	int32_t	mm	Altitude.
relative alt	int32_t	mm	Height above ground level.
vx	int16_t	cm/s	Ground speed on the x-axis. (North is positive)
vy	int16_t	cm/s	Ground speed on the y-axis. (East is positive)
vz	int16_t	cm/s	Ground speed on the z-axis. (The bottom is positive)
hdg	uint16_t	cdeg	Aircraft orientation (Yaw). Range from 0.0 to 359.99. If unknown, UINT16_MAX is set.

3.2.5. CaoController::OnMessage("EXTENDED_SYS_STATE")

Notifies the takeoff and landing status of the aircraft. It maintains the VTOL status and takeoff and landing status.

The details of the data received by EXTENDED_SYS_STATE are as follows.

Table 3-25 EXTENDED_SYS_STATE details

Parameter	Type	Description
vtol state	uint8_t	VTOL status.
landed state	uint8_t	Takeoff and landing state.

3.2.6. CaoController::OnMessage("MISSION_ITEM_REACHED")

Notifies you of information regarding the completion of flight plan execution. Holds the sequence number of completed mission items.

The details of the data received by MISSION_ITEM_REACHED are as follows.

Table 3-26 MISSION_ITEM_REACHED details

Parameter	Type	Description
seq	uint16_t	Sequence number of completed mission items

3.2.7. CaoController::OnMessage("MISSION_CURRENT")

Get information about the flight plan in progress. Holds the sequence number of running mission items.

The details of the data received by MISSION_CURRENT are as follows.

Table 3-27 MISSION_CURRENT details

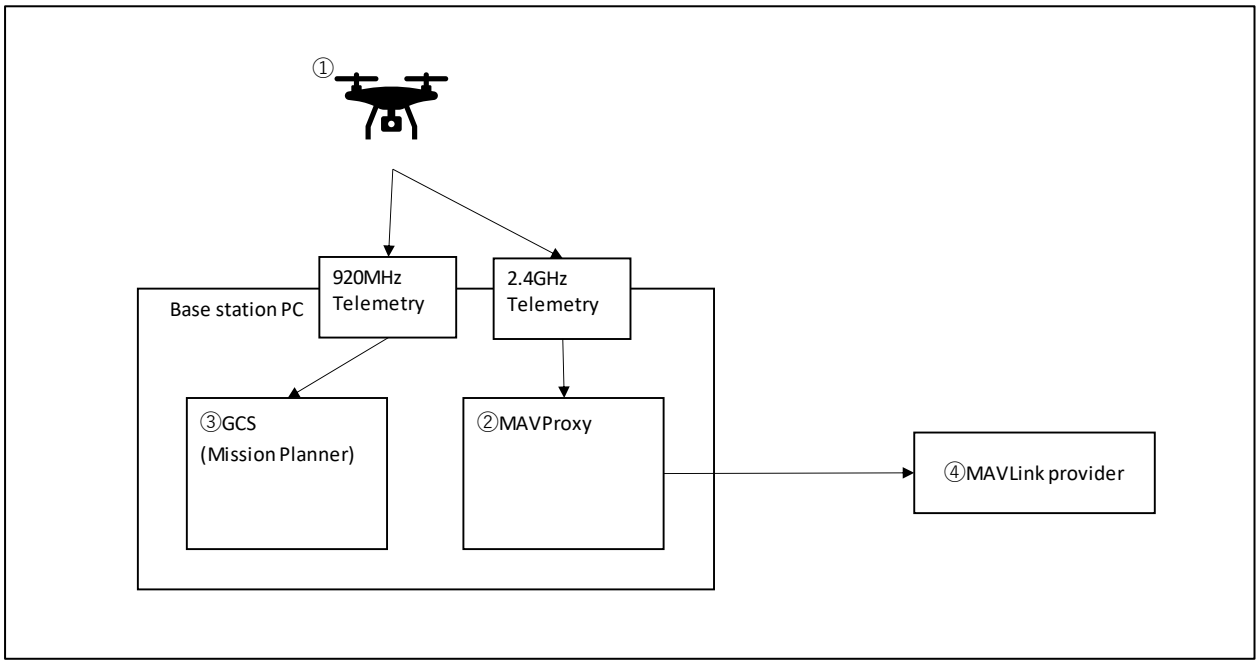
Parameter	Type	Description
seq	uint16_t	Sequence number of the currently running mission item

4. Reference

4.1. System configuration / execution procedure example

The following is an example of a system configuration that connects a drone and a MAVLink provider.

Figure 4-1 System configuration example



The procedure for connecting to the drone is as follows.

① Start the drone
② Start MAVProxy
③ Launch Mission Planner and connect to the drone
④ Execute AddControl of MAVLink provider

4.2. Mission execution sequence

A mission is a drone flight plan, which is sometimes called a flight plan.
Shows a summary of instructions such as the route the drone flies, actions such as takeoff and landing, and speed changes. In order to start a mission, you need to change the drone to mission mode and start the motor with the mission uploaded to the drone.

The sequence to start a mission using the MAVLink provider is shown in the figure below.

Figure 4-2 Mission execution sequence diagram