OlliW MAVLink augmented OpenTX LUA function reference, detailed edition (rev. 1.1 based on v26 firmware)

March 21st, 2021

General OpenTX LUA additions are to be be called directly - example: *getEvent()* , MavSDK library function calls need to be prepended with *mavsdk* and a dot - example: *mavsdk.mavtelemIsEnabled()*
Getters are listed in blue, setters in green.

	General OpenTX LUA additions	return value / parameter	Unit	Internal C++ function/wrapper	Value stems internally from or calls function(s)	MAVLink message	MAVLink msg field(s)	Data type & unit	Comments
Generic	getEvent	value[integer](event)	enum, see keys.h	luaGetEvent	s_evt	-	-	-	returns only locked keys and rotary events
	lockKeys	value[unsigned](mask)	-	luaLockKeys	sets s_evt_lockmask, allows only ENTER, MODEL, EXIT, TELEM, RADIO to be locked	-	-	-	gets set for max 500ms, OpenTX internal setting
	unlockKeys	-	-	luaUnlockKeys	clears s_evt_lockmask	-	-	-	OpenTX internal setting
	isInMenu	value[bool]	-	luaIsInMenu	true if menuLevel > 0	-	-	-	OpenTX internal setting

	MavSDK function	return value / parameter	Unit	MavSDK internal C++ function/wrapper	Value stems internally from or calls function(s)	MAVLink message	MAVLink msg field(s)	Data type & unit	Comments
Generic 1	mavtelemIsEnabled	value[bool]	-	luaMavsdkMavTelemIsEnabled	g_eeGeneral.auxSerialMode g_eeGeneral.aux2SerialMode	-	-	-	OpenTX radio SYSTEM settings check
	isReceiving	value[bool]	-	luaMavsdkIsReceiving	mavlinkTelem.isReceiving()	all except RADIO_STATUS	-	-	
	isInitialized	value[bool]	-	luaMavsdkIsInitialized	mavlinkTelem.autopilot.is_receiving mavlinkTelem.autopilot.is_initialized	Any when compid == autopilot.compid and all requests done	-	-	
	getVersion	value[string]	-	luaMavsdkMavTelemVersion	OWVERSIONONLYSTR	-	-	-	Constant in opentx.h, e.g. "v22" or "v22rc01"
Generic 2	getAutopilotType	value[number]	enum MAV_AUTOPILOT	luaMavsdkGetAutopilotType	mavlinkTelem.autopilottype	#0 HEARTBEAT	autopilot	uint8_t [enum]	
	getVehicleType	value[number]	enum MAV_TYPE	luaMavsdkGetVehicleType	mavlinkTelem.vehicletype	#0 HEARTBEAT	type	uint8_t [enum]	
	getFlightMode	value[number]	enum PLANE_MODE or COPTER_MODE or SUB_MODE or ROVER_MODE or TRACKER_MODE	luaMavsdkGetFlightMode	mavlinkTelem.flightmode	#0 HEARTBEAT	custom_mode	uint32_t [enum]	enum type depends on vehicletype
	getVehicleClass	value[number]	enum MAV_TYPE	luaMavsdkGetVehicleClass	mavlinkTelem.vehicletype	#0 HEARTBEAT	type	uint8_t [enum]	
	getSystemStatus	value[number]	enum MAV_STATE	luaMavsdkGetSystemStatus	mavlinkTelem.autopilot.system_status	#0 HEARTBEAT	system_status	uint8_t [enum]	
	isArmed	value[bool]	-	luaMavsdkIsArmed	mavlinkTelem.autopilot.is_armed	#0 HEARTBEAT	base_mode	uint8_t [enum]	
	getSystemStatusSensors	table (present[number], enabled[number], health[number]) or nil	bitmap MAV_SYS_STATUS_SENSOR bitmap MAV_SYS_STATUS_SENSOR bitmap MAV_SYS_STATUS_SENSOR	luaMavsdkGetSystemStatusSensors	mavlinkTelem.sysstatus.sensors_present mavlinkTelem.sysstatus.sensors_enabled mavlinkTelem.sysstatus.sensors_health	#1 SYS_STATUS #1 SYS_STATUS #1 SYS_STATUS	onboard_control_sensors_present onboard_control_sensors_enabled onboard_control_sensors_health	uint32_t [bitmap] uint32_t [bitmap] uint32_t [bitmap]	returns nil if not mavlinkTelem.sysstatus.received
IMU	getAttRollDeg	value[number]	°	luaMavsdkGetAttRollDeg	mavlinkTelem.att.roll_rad * 180/PI	#30 ATTITUDE	roll	float [rad]	-PI to +PI
	getAttPitchDeg	value[number]	°	luaMavsdkGetAttPitchDeg	mavlinkTelem.att.pitch_rad * 180/PI	#30 ATTITUDE	pitch	float [rad]	-PI to +PI
	getAttYawDeg	value[number]	°	luaMavsdkGetAttYawDeg	mavlinkTelem.att.yaw_rad * 180/PI	#30 ATTITUDE	yaw	float [rad]	-PI to +PI
Vfr	getVfrAirSpeed	value[number]	m/s	luaMavsdkGetVfrAirSpeed	mavlinkTelem.vfr.airspd_mps	#74 VFR_HUD	airspeed	float [m/s]	
	getVfrGroundSpeed	value[number]	m/s	luaMavsdkGetVfrGroundSpeed	mavlinkTelem.vfr.groundspd_mps	#74 VFR_HUD	groundspeed	float [m/s]	
	getVfrAltitudeMsl	value[number]	m	luaMavsdkGetVfrAltitudeMsl	mavlinkTelem.vfr.alt_m	#74 VFR_HUD	alt	float [m]	
	getVfrClimbRate	value[number]	m/s	luaMavsdkGetVfrClimbRate	mavlinkTelem.vfr.climbrate_mps	#74 VFR_HUD	climb	float [m/s]	
	getVfrHeadingDeg	value[number]	°	luaMavsdkGetVfrHeadingDeg	mavlinkTelem.vfr.heading_deg	#74 VFR_HUD	heading	int16_t [°]	0-360, 0=north
	getVfrThrottle	value[integer]	%	luaMavsdkGetVfrThrottle	mavlinkTelem.vfr.thro_pct	#74 VFR_HUD	throttle	uint16_t [%]	0 to 100
GPS generic	getGpsCount	value[integer]	bitmap	luaMavsdkGetGpsCount	mavlinkTelem.gps_instancemask	#24 GPS_RAW_INT #124 GPS2_RAW	any		
	getPositionLatLonInt	table (lat[integer], lon[integer])	°E7 °E7	luaMavsdkGetPositionLatLonInt	mavlinkTelem.gposition.lat mavlinkTelem.gposition.lon	#33 GLOBAL_POSITION_INT #33 GLOBAL_POSITION_INT	lat lon	int32_t [°E7] int32_t [°E7]	need to divide with 10 million to get ° need to divide with 10 million to get °
	getPositionAltitudeMsl	value[number]	m	luaMavsdkGetPositionAltitudeMsl	mavlinkTelem.gposition.alt_mm/1000	#33 GLOBAL_POSITION_INT	alt	int32_t [mm]	
	getPositionAltitudeRelative	value[number]	m	luaMavsdkGetPositionAltitudeRelative	mavlinkTelem.gposition.relative_alt_mm/1000	#33 GLOBAL_POSITION_INT	relative_alt	int32_t [mm]	Altitude above ground
	getPositionHeadingDeg	value[number]	°	luaMavsdkGetPositionHeadingDeg	mavlinkTelem.gposition.hdg_cdeg/100	#33 GLOBAL_POSITION_INT	hdg	uint16_t [c°]	0 to 359.99°, UINT16_MAX = unknown
	getPositionSpeedNed	table (vx[number], vy[number], vz[number])	m/s m/s m/s	luaMavsdkGetPositionSpeedNed	mavlinkTelem.gposition.vx_cmps/100 mavlinkTelem.gposition.vy_cmps/100 mavlinkTelem.gposition.vz_cmps/100	#33 GLOBAL_POSITION_INT #33 GLOBAL_POSITION_INT #33 GLOBAL_POSITION_INT	vx vy vz	int16_t [cm/s] int16_t [cm/s] int16_t [cm/s]	
GPS, 1st or only	isGpsAvailable	value[bool]	-	luaMavsdkIsGps1Available	mavlinkTelem.gps_instancemask & 0x01	#24 GPS_RAW_INT	any		
	getGpsStatus	table (fix[number], hdop[number], vdop[number], sat[number])	enum GPS_FIX_TYPE - - -	luaMavsdkGetGps1Status	mavlinkTelem.gps1.fix mavlinkTelem.gps1.hdop/100 mavlinkTelem.gps1.vdop/100 mavlinkTelem.gps1.sat	#24 GPS_RAW_INT #24 GPS_RAW_INT #24 GPS_RAW_INT #24 GPS_RAW_INT	fix_type eph epv satellites_visible	uint8_t [enum] uint16_t uint16_t uint8_t	valid range 0 to 8 UINT16_MAX = unknown UINT16_MAX = unknown UINT8_MAX = unknown
	getGpsFix	value[number]	enum GPS_FIX_TYPE	luaMavsdkGetGps1Fix	mavlinkTelem.gps1.fix	#24 GPS_RAW_INT	fix_type	uint8_t [enum]	valid range 0 to 8
	getGpsHDop	value[number]	-	luaMavsdkGetGps1HDop	mavlinkTelem.gps1.hdop/100	#24 GPS_RAW_INT	eph	uint16_t	UINT16_MAX = unknown
	getGpsVDop	value[number]	-	luaMavsdkGetGps1VDop	mavlinkTelem.gps1.vdop/100	#24 GPS_RAW_INT	epv	uint16_t	UINT16_MAX = unknown
	getGpsSat	value[number]	-	luaMavsdkGetGps1Sat	mavlinkTelem.gps1.sat	#24 GPS_RAW_INT	satellites_visible	uint8_t	UINT8_MAX = unknown, currently no special handling
	getGpsLatLonInt	table (lat[integer], lon[integer])	°E7 °E7	luaMavsdkGetGps1LatLonInt	mavlinkTelem.gps1.lat mavlinkTelem.gps1.lon	#24 GPS_RAW_INT #24 GPS_RAW_INT	lat lon	int32_t [°E7] int32_t [°E7]	need to divide with 10 million to get ° need to divide with 10 million to get °
	getGpsAltitudeMsl	value[number]	m	luaMavsdkGetGps1AltitudeMsl	mavlinkTelem.gps1.alt_mm/1000	#24 GPS_RAW_INT	alt	int32_t [mm]	
	getGpsSpeed	value[number]	m/s	luaMavsdkGetGps1Speed	mavlinkTelem.gps1.vel_cmps/100	#24 GPS_RAW_INT	vel	uint16_t [cm/s]	>=UINT16_MAX outputs nil
	getGpsCourseOverGroundDeg	value[number]	°	luaMavsdkGetGps1CourseOverGroundDeg	mavlinkTelem.gps1.cog_cdeg/100	#24 GPS_RAW_INT	cog	uint16_t [0.01°]	0 to 359.99°, >=UINT16_MAX outputs nil
GPS, 2nd	isGps2Available	value[bool]	-	luaMavsdkIsGps2Available	mavlinkTelem.gps_instancemask & 0x02	#124 GPS2_RAW	any		
	getGps2Status	table (fix[number], hdop[number], vdop[number], sat[number])	enum GPS_FIX_TYPE - - -	luaMavsdkGetGps2Status	mavlinkTelem.gps2.fix mavlinkTelem.gps2.hdop/100 mavlinkTelem.gps2.vdop/100 mavlinkTelem.gps2.sat	#124 GPS2_RAW #124 GPS2_RAW #124 GPS2_RAW #124 GPS2_RAW	fix_type eph epv satellites_visible	uint8_t [enum] uint16_t uint16_t uint8_t	valid range 0 to 8 UINT16_MAX = unknown UINT16_MAX = unknown UINT8_MAX = unknown
	getGps2Fix	value[number]	enum GPS_FIX_TYPE	luaMavsdkGetGps2Fix	mavlinkTelem.gps2.fix	#124 GPS2_RAW	fix_type	uint8_t [enum]	valid range 0 to 8
	getGps2HDop	value[number]	-	luaMavsdkGetGps2HDop	mavlinkTelem.gps2.hdop/100	#124 GPS2_RAW	eph	uint16_t	UINT16_MAX = unknown
	getGps2VDop	value[number]	-	luaMavsdkGetGps2VDop	mavlinkTelem.gps2.vdop/100	#124 GPS2_RAW	epv	uint16_t	UINT16_MAX = unknown
	getGps2Sat	value[number]	-	luaMavsdkGetGps2Sat	mavlinkTelem.gps2.sat	#124 GPS2_RAW	satellites_visible	uint8_t	UINT8_MAX = unknown, currently no special handling
	getGps2LatLonInt	table (lat[integer], lon[integer])	°E7 °E7	luaMavsdkGetGps2LatLonInt	mavlinkTelem.gps2.lat mavlinkTelem.gps2.lon	#124 GPS2_RAW #124 GPS2_RAW	lat lon	int32_t [°E7] int32_t [°E7]	need to divide with 10 million to get ° need to divide with 10 million to get °
	getGps2AltitudeMsl	value[number]	m	luaMavsdkGetGps2AltitudeMsl	mavlinkTelem.gps2.alt_mm/1000	#124 GPS2_RAW	alt	int32_t [mm]	
	getGps2Speed	value[number]	m/s	luaMavsdkGetGps2Speed	mavlinkTelem.gps2.vel_cmps/100	#124 GPS2_RAW	vel	uint16_t [cm/s]	>=UINT16_MAX outputs nil
	getGps2CourseOverGroundDeg	value[number]	°	luaMavsdkGetGps2CourseOverGroundDeg	mavlinkTelem.gps2.cog_cdeg/100	#124 GPS2_RAW	cog	uint16_t [0.01°]	>=UINT16_MAX outputs nil
Battery	isBatAvailable	value[bool]	-	luaMavsdkIsBat1Available	mavlinkTelem.bat_instancemask & 0x01	#147 BATTERY_STATUS	id	uint8_t	id must be < 8
	isBat2Available	value[bool]	-	luaMavsdkIsBat2Available	mavlinkTelem.bat_instancemask & 0x02	#147 BATTERY_STATUS	id	uint8_t	id must be < 8
	getBatCount	value[integer]	-	luaMavsdkGetBatCount	mavlinkTelem.bat_instancemask	#147 BATTERY_STATUS	id	uint8_t	id must be < 8
Battery, 1st or only	getBatChargeConsumed	value[number]	mAh	luaMavsdkGetBat1ChargeConsumed	mavlinkTelem.bat1.charge_consumed_mAh	#147 BATTERY_STATUS	current_consumed	int32_t [mAh]	negative outputs nil
	getBatEnergyConsumed	value[number]	J	luaMavsdkGetBat1EnergyConsumed	mavlinkTelem.bat1.energy_consumed_hJ * 100	#147 BATTERY_STATUS	energy_consumed	int32_t [100J]	negative outputs nil
	getBatTemperature	value[number]	°C	luaMavsdkGetBat1Temperature	mavlinkTelem.bat1.temperature_cC/100	#147 BATTERY_STATUS	temperature	int16_t [0.01°C]	>=INT16_MAX outputs nil
	getBatVoltage	value[number]	V	luaMavsdkGetBat1Voltage	mavlinkTelem.bat1.voltage_mV/1000	#147 BATTERY_STATUS	voltage[10] voltages_ext[4]	uint16_t[10] [mV] uint16_t[4] [mV]	
	getBatCurrent	value[number|nil]	A	luaMavsdkGetBat1Current	mavlinkTelem.bat1.current_cA/100	#147 BATTERY_STATUS	current_battery	int16_t [10mA]	-1 outputs nil
	getBatRemaining	value[integer]	%	luaMavsdkGetBat1Remaining	mavlinkTelem.bat1.remaining_pct	#147 BATTERY_STATUS	battery_remaining	int8_t [%]	-1 outputs nil
	getBatCellCount	value[integer]	-	luaMavsdkGetBat1CellCount	mavlinkTelem.bat1.cellcount	#147 BATTERY_STATUS #147 BATTERY_STATUS	voltage[10] voltages_ext[4]	uint16_t[10] [mV] uint16_t[4] [mV]	negative outputs nil
	getBatTimeRemaining	value[integer|nil]	s	luaMavsdkGetBat1TimeRemaining	mavlinkTelem.bat1.time_remaining	#147 BATTERY_STATUS	time_remaining	int32_t [s]	if time_remaining == 0 outputs nil
	getBatChargeState	value[integer|nil]	enum MAV_BATTERY_CHARGE_STATE	luaMavsdkGetBat1ChargeState	mavlinkTelem.bat1.charge_state	#147 BATTERY_STATUS	charge_state	uint8_t [enum]	if undefined, outputs nil
	getBatFaultBitMask	value[integer|nil]	enum MAV_BATTERY_FAULT	luaMavsdkGetBat1FaultBitMask	mavlinkTelem.bat1.fault_bitmask	#147 BATTERY_STATUS	fault_bitmask	uint32_t [enum]	if state is !(failed or unhealty) outputs nil
	getBatCapacity	value[number]		luaMavsdkGetBat1Capacity	mavlinkTelem.param.BATT_CAPACITY	#22 PARAM_VALUE	param_value	float	negative outputs nil, unit mAh in 50 mAh steps in ArduPilot

	MavSDK function	return value / parameter	Unit	MavSDK internal C++ function/wrapper	Value stems internally from or calls function(s)	MAVLink message	MAVLink msg field(s)	Data type & unit	Comments
Battery, 2nd	getBat2ChargeConsumed	value[number]	mAh	luaMavsdkGetBat2ChargeConsumed	mavlinkTelem.bat2.charge_consumed_mAh	#147 BATTERY_STATUS	current_consumed	int32_t [mAh]	negative outputs nil
	getBat2EnergyConsumed	value[number]	J	luaMavsdkGetBat2EnergyConsumed	mavlinkTelem.bat2.energy_consumed_hJ * 100	#147 BATTERY_STATUS	energy_consumed	int32_t [100J]	negative outputs nil
	getBat2Temperature	value[number]	°C	luaMavsdkGetBat2Temperature	mavlinkTelem.bat2.temperature_cC/100	#147 BATTERY_STATUS	temperature	int16_t [0.01°C]	>=INT16_MAX outputs nil
							voltage[10]	uint16_t[10] [mV]	
	getBat2Voltage	value[number]	V	luaMavsdkGetBat2Voltage	mavlinkTelem.bat2.voltage_mV/1000	#147 BATTERY_STATUS	voltages_ext[4]	uint16_t[4] [mV]	
	getBat2Current	value[number|nil]	A	luaMavsdkGetBat2Current	mavlinkTelem.bat2.current_cA/100	#147 BATTERY_STATUS	current_battery	int16_t [10mA]	-1 outputs nil
	getBat2Remaining	value[integer]	%	luaMavsdkGetBat2Remaining	mavlinkTelem.bat2.remaining_pct	#147 BATTERY_STATUS	battery_remaining	int8_t [%]	-1 outputs nil
						#147 BATTERY_STATUS	voltage[10]	uint16_t[10] [mV]	
	getBat2CellCount	value[integer]	-	luaMavsdkGetBat2CellCount	mavlinkTelem.bat2.cellcount	#147 BATTERY_STATUS	voltages_ext[4]	uint16_t[4] [10mV]	negative outputs nil
	getBat2TimeRemaining	value[integer|nil]	s	luaMavsdkGetBat2TimeRemaining	mavlinkTelem.bat2.time_remaining	#147 BATTERY_STATUS	time_remaining	int32_t [s]	if time_remaining == 0 outputs nil
	getBat2ChargeState	value[integer|nil]	enum MAV_BATTERY_CHARGE_STATE	luaMavsdkGetBat2ChargeState	mavlinkTelem.bat2.charge_state	#147 BATTERY_STATUS	charge_state	uint8_t [enum]	if undefined, outputs nil
	getBat2FaultBitMask	value[integer|nil]	enum MAV_BATTERY_FAULT	luaMavsdkGetBat2FaultBitMask	mavlinkTelem.bat2.fault_bitmask	#147 BATTERY_STATUS	fault_bitmask	uint32_t [enum]	if state is !(failed or unhealty) outputs nil
									negative outputs nil,
	getBat2Capacity	value[number]		luaMavsdkGetBat2Capacity	mavlinkTelem.param.BATT2_CAPACITY	#22 PARAM_VALUE	param_value	float	unit mAh in 50 mAh steps in ArduPilot
Mission		table (count[integer],	-		mavlinkTelem.mission.count	#44 MISSION_COUNT	count	uint16_t	
	getMission	current_seq[integer])	-	luaMavsdkGetMission	mavlinkTelem.mission.seq_current	#42 MISSION_CURRENT	seq	uint16_t	
		table (seq[integer],	-		mavlinkTelem.missionItem.seq	#73 MISSION_ITEM_INT	seq	uint16_t	starts at 0, no gaps
		command[integer],	enum MAV_CMD_*(value)		mavlinkTelem.missionItem.command	#73 MISSION_ITEM_INT	command	uint16_t [enum]	
		frame[integer],	enum MAV_FRAME		mavlinkTelem.missionItem.frame	#73 MISSION_ITEM_INT	frame	uint8_t [enum]	coordinate system
		is_global[boolean],	.		mavlinkTelem.missionItem.frame	#73 MISSION_ITEM_INT	frame	uint8_t [enum]	coordinate system
		lat[integer] or x[number],	°e7 or m		mavlinkTelem.missionItem.x or .x/10000	#73 MISSION_ITEM_INT	x	int32_t [°e7] or [m*e4]	global °e7, local m*e4
		lon[integer] or y[number],	°e7 or m		mavlinkTelem.missionItem.y or y//10000	#73 MISSION_ITEM_INT	y	int32_t [°e7] or [m*e4]	global °e7, local m*e4
	getMissionItem	alt[number] or z[number])	°e7 or m	luaMavsdkGetMissionItem	mavlinkTelem.missionItem.z or z/10000	#73 MISSION_ITEM_INT	z	float [m]	global alt m, local z m
		table (nav_bearing[number],	°		mavlinkTelem.navControllerOutput.nav_bearing	#62 NAV_CONTROLLER_OUTPUT	nav_bearing	int16_t [°]	
		target_bearing[number],	°		mavlinkTelem.navControllerOutput.target_bearing	#62 NAV_CONTROLLER_OUTPUT	target_bearing	int16_t [°]	
	getNavController	wp_dist[number])	m	luaMavsdkGetNavControllerOutput	mavlinkTelem.navControllerOutput.wp_dist	#62 NAV_CONTROLLER_OUTPUT	wp_dist	uint16_t [m]	
Messages							severity	uint8_t [enum]	valid range 0 to 7
	isStatusTextAvailable	value[bool]	-	luaMavsdkIsStatusTextAvailable	not mavlinkTelem.statustext.fifo.isEmpty()	#253 STATUSTEXT	text	char[50]	without null termination character
		value[integer|nil]	enum MAV_SEVERITY				severity	uint8_t [enum]	
	getStatusText	value[string|nil]	-	luaMavsdkGetStatusText	mavlinkTelem.statustext.fifo	#253 STATUSTEXT	text	char[50]	if nothing in buffer, outputs nil, nil
RF Link					mavlinkTelem.radio.rssi, or	#109 RADIO_STATUS	rssi	uint8_t	valid range 0-254, 255 = invalid
					mavlinkTelem.radio.rssi65, or	#65 RC_CHANNELS	rssi	uint8_t	valid range 0-254, 255 = invalid
	getRadioRssi	value[integer]	-	luaMavsdkGetRadioRssi	mavlinkTelem.radio.rssi35	#35 RC_CHANNELS_RAW	rssi	uint8_t	valid range 0-254, 255 = invalid
	getRadioRemoteRssi	value[integer]	-	luaMavsdkGetRadioRemoteRssi	mavlinkTelem.radio.remrssi	#109 RADIO_STATUS	remrssi	uint8_t	valid range 0-254, 255 = invalid
	getRadioNoise	value[integer]	2dB on SiK	luaMavsdkGetRadioNoise	mavlinkTelem.radio.noise	#109 RADIO_STATUS	noise	uint8_t	valid range 0-254, 255 = invalid
	getRadioRemoteNoise	value[integer]	2dB on SiK	luaMavsdkGetRadioRemoteNoise	mavlinkTelem.radio.remnoise	#109 RADIO_STATUS	remnoise	uint8_t	valid range 0-254, 255 = invalid
						#109 RADIO_STATUS or	rssi	uint8_t	
					mavlinkTelem.radio.rssi_scaled,	#65 RC_CHANNELS or	rssi	uint8_t	
	getRadioRssiScaled	value[integer|nil]	-	luaMavsdkGetRadioRssiScaled	calculated from rssi and g_model.mavlinkRssiScale	#35 RC_CHANNELS_RAW	rssi	uint8_t	if #109 or #65 or #35 are not receiving, outputs nil
	optionGetRssiScale	value[integer]		luaMavsdkOptionGetRssiScale	g_model.mavlinkRssiScale	-	-	-	OpenTX internal function
	optionSetRssiScale	value[integer]	-	luaMavsdkOptionSetRssiScale	g_model.mavlinkRssiScale = value, limited from 0 to 255	-	-	-	OpenTX internal function
	optionIsRssiEnabled	value[bool]	-	luaMavsdkOptionIsRssiEnabled	g_model.mavlinkRssi	-	-	-	OpenTX internal function
	optionEnableRssi	value[integer]{bool}		luaMavsdkOptionEnableRssi	g_model.mavlinkRssi = value ? 1 : 0	-	-	-	OpenTX internal function
	radioDisableRssiVoice	value[integer]{bool}	-	luaMavsdkRadioDisableRssiVoice	if value>0 mavlinkTelem.radio.rssi_voice_disabled = true else fals	-	-	-	OpenTX internal function
AP	apIsFlying	value[bool]	-	luaMavsdkApIsFlying	not mavlinkTelem.autopilot.is_standby	#0 HEARTBEAT	system_status	uint8_t [enum]	
	apIsFailsafe	value[bool]	-	luaMavsdkApIsFailsafe	mavlinkTelem.autopilot.is_critical	#0 HEARTBEAT	system_status	uint8_t [enum]	
									true if EKF_POS_HORIZ_ABS &
	apPositionOk	value[bool]	-	luaMavsdkApPositionOk	mavlinkTelem.apPositionOk()	#193 EKF_STATUS_REPORT	flags	uint16_t [enum]	EKF_VELOCITY_HORIZ
									returns nil if
	apGetArmingCheck	value[number|nil]	bitmap	luaMavsdkApGetArmingCheck	mavlinkTelem.param.ARMING_CHECK	#22 PARAM_VALUE	param_value	float	mavlinkTelem.param.ARMING_CHECK < 0
			enum PLANE_MODE or COPTER_MODE						
			or SUB_MODE or ROVER_MODE or						
	apSetFlightMode	value[integer]	TRACKER_MODE	luaMavsdkApSetFlightMode	mavlinkTelem.apSetFlightMode(value)	176 MAV_CMD_DO_SET_MODE	2: Custom Mode	[enum]	value = ap_flight_mode, according vehicle type
	apRequestBanner	none	-	luaMavsdkApRequestBanner	mavlinkTelem.apRequestBanner()	42428 MAV_CMD_DO_SEND_BANNER	-	-	
	apArm	value[integer]{bool}	-	luaMavsdkApArm	mavlinkTelem.apArm()	400 MAV_CMD_COMPONENT_ARM_DISARM	1: Arm	-	if value > 0, arms
	apCopterTakeOff	value[number]{alt}	m	luaMavsdkApCopterTakeOff	mavlinkTelem.apCopterTakeOff(value)	22 MAV_CMD_NAV_TAKEOFF	7: Altitude	[m]	value = Altitude
	apLand	none	-	luaMavsdkApLand	mavlinkTelem.apLand()	21 MAV_CMD_NAV_LAND	-	-	
	apGetRangefinder	value[number]	m	luaMavsdkApGetRangefinder	mavlinkTelem.rangefinder.distance	#173 RANGEFINDER	distance	float [m]	
Camera					if (mavlinkTelem.camera.is_receiving > 0) true				
	cameraIsReceiving	value[bool]	-	luaMavsdkCameraIsReceiving	else false	any from camera.compid	-	-	
					if (((mavlinkTelem.camera.is_receiving > 0) and	any when _msg.compid == camera.compid			
	cameraIsInitialized	value[bool]	-	luaMavsdkCameraIsInitialized	mavlinkTelem.camera.is_initialized) true else false	and no requests waiting	-	-	
		table (compid[integer],	enum MAV_COMPONENT		mavlinkTelem.camera.compid	#0 HEARBEAT	_msg.compid (header, not payload!)	uint8_t [enum]	
		flags[integer],	enum CAMERA_CAP_FLAGS		mavlinkTelem.cameraInfo.flags	#259 CAMERA_INFORMATION	flags	uint32_t [enum]	
		has_video[bool],	-		mavlinkTelem.cameraInfo.has_video	#259 CAMERA_INFORMATION	flags & 1	uint32_t [enum]	
		has_photo[bool],	-		mavlinkTelem.cameraInfo.has_photo	#259 CAMERA_INFORMATION	flags & 2	uint32_t [enum]	
		has_modes[bool],	-		mavlinkTelem.cameraInfo.has_modes	#259 CAMERA_INFORMATION	flags & 4	uint32_t [enum]	
		total_capacity[number|nil],	MiB		mavlinkTelem.cameraInfo.total_capacity_MiB	#261 STORAGE_INFORMATION	total_capacity (only when READY, else NAN)	float [MiB]	
		vendor_name[string],	-		mavlinkTelem.cameraInfo.vendor_name	#259 CAMERA_INFORMATION	vendor_name	uint8_t[32]	
		model_name[string],	-		mavlinkTelem.cameraInfo.model_name	#259 CAMERA_INFORMATION	model_name	uint8_t[32]	
	cameraGetInfo	firmware_version[string])	-	luaMavsdkCameraGetInfo	mavlinkTelem.cameraInfo.firmware_version	#259 CAMERA_INFORMATION	firmware_version	uint32_t	Dev, Patch, Minor, Major
		table (system_status[integer],	enum MAV_STATE		mavlinkTelem.camera.system_status	#0 HEARBEAT	system_status	unit8_t [enum]	
		mode[integer],	enum CAMERA_MODE		mavlinkTelem.cameraStatus.mode	#260 CAMERA_SETTINGS	mode_id	uint8_t [enum]	converted to boolean, true if IMAGE
		video_on[boolean],	-		mavlinkTelem.cameraStatus.video_on	#262 CAMERA_CAPTURE_STATUS	video_status, if > 0 outputs true, else false	uint8_t	converted to boolean
		photo_on[boolean],	-		mavlinkTelem.cameraStatus.photo_on	#262 CAMERA_CAPTURE_STATUS	image_status, if > 0 outputs true, else false	uint8_t	converted to boolean
		available_capacity[number|nil],	MiB		mavlinkTelem.cameraStatus.available_capacity_MiB	#262 CAMERA_CAPTURE_STATUS or	available_capacity	float [MiB]	
						#261 STORAGE_INFORMATION	available_capacity (only when READY, else NAN)	float [MiB]	
		battery_voltage[number|nil],	V		mavlinkTelem.cameraStatus.battery_voltage_V	#147 BATTERY_STATUS	sum voltages/1000, if all UINT16_MAX then NAN	uint16_t[10] [mV]	
	cameraGetStatus	battery_remaininpct[integer|nil])	%	luaMavsdkCameraGetStatus	mavlinkTelem.cameraStatus.battery_remaining_pct	#147 BATTERY_STATUS	battery_remaining	int8_t [%]	range 0 to 100, -1 if unknown
							1: 0		
							2: Camera Mode = CAMERA_MODE_VIDEO = 1		
							3: 0		
							4: 0		
	cameraSendVideoMode	none	-	luaMavsdkCameraSendVideoMode	mavlinkTelem.sendCameraSetVideoMode()	530 MAV_CMD_SET_CAMERA_MODE	7: 0	-	
							1: 0		
							2: Camera Mode = CAMERA_MODE_IMAGE = 0		
							3: 0		
							4: 0		
	cameraSendPhotoMode	none	-	luaMavsdkCameraSendPhotoMode	mavlinkTelem.sendCameraSetPhotoMode()	530 MAV_CMD_SET_CAMERA_MODE	7: 0	-	
							1: Stream ID = 0		
							2: Status Frequency = 0.2 = 5 s period	[Hz]	
	cameraStartVideo	none	-	luaMavsdkCameraStartVideo	mavlinkTelem.sendCameraStartVideo()	2500 MAV_CMD_VIDEO_START_CAPTURE	3 to 7: 0		
							1: Steam ID = 0		
	cameraStopVideo	none	-	luaMavsdkCameraStopVideo	mavlinkTelem.sendCameraStopVideo()	2501 MAV_CMD_VIDEO_STOP_CAPTURE	2 to 7: 0		
							1: Reserved = 0	-	
							2: Interval = 0	[s]	
							3: Total Images = 1	-	
							4: Sequence Number = 0	-	
	cameraTakePhoto	none	-	luaMavsdkCameraTakePhoto	mavlinkTelem.sendCameraTakePhoto()	2000 MAV_CMD_IMAGE_START_CAPTURE	5 to 7: 0	-	

	MavSDK function	return value / parameter	Unit	MavSDK internal C++ function/wrapper	Value stems internally from or calls function(s)	MAVLink message	MAVLink msg field(s)	Data type & unit	Comments

Gimbal generic	gimbalIsReceiving	value[bool]	-	luaMavsdkGimbalIsReceiving	if (mavlinkTelem.gimbal.is_receiving > 0) true
					else false
						any from gimbal.compid	-	-	

	gimbalIsInitialized	value[bool]	-	luaMavsdkGimbalIsInitialized	if ((mavlinkTelem.gimbal.is_receiving > 0) and mavlinkTelem.gimbal.is_initialized) true else false	#0 HEARTBEAT	any	-	at least one HEARTBEAT from gimbal
	gimbalGetInfo	table (compid[integer], vendor_name[string], model_name[string], custom_name[string], firmware_version[string], hardware_version[string], capability_flags[integer])	enum MAV_COMPONENT	luaMavsdkGimbalGetInfo	mavlinkTelem.gimbal.compid mavlinkTelem.gimbaldeviceInfo.vendor_name mavlinkTelem.gimbaldeviceInfo.model_name mavlinkTelem.gimbaldeviceInfo.custom_name mavlinkTelem.gimbaldeviceInfo.firmware_version mavlinkTelem.gimbaldeviceInfo.hardware_version mavlinkTelem.gimbaldeviceInfo.cap_flags	#0 HEARTBEAT #283 GIMBAL_DEVICE_INFORMATION #283 GIMBAL_DEVICE_INFORMATION #283 GIMBAL_DEVICE_INFORMATION #283 GIMBAL_DEVICE_INFORMATION #283 GIMBAL_DEVICE_INFORMATION #283 GIMBAL_DEVICE_INFORMATION	msg.compid (header, not payload!) vendor_name model_name custom_name firmware_version hardware_version cap_flags + custom_capflags	uint8_t [enum] char[32] char[32] char[32] uint32_t uint32_t uint16_t + uint16_t	 Dev, Patch, Minor, Major bitmap + bitmap
	gimbalGetStatus	table (system_status[number], custom_mode[number], is_armed[bool], prearm_ok[bool])	- - - -	luaMavsdkGimbalGetStatus	mavlinkTelem.gimbal.system_status mavlinkTelem.gimbal.custom_mode mavlinkTelem.gimbal.is_armed mavlinkTelem.gimbal.prearm_ok	#0 HEARTBEAT #0 HEARTBEAT #0 HEARTBEAT #0 HEARTBEAT	system_status custom_mode base_mode custom_mode	uint8_t uint32_t uint8_t uint8_t -> bool	
	gimbalGetAttRollDeg	value[number]	°	luaMavsdkGimbalGetAttRollDeg	mavlinkTelem.gimbalAtt.roll_deg	#30 ATTITUDE	roll * 180/PI	float [rad]	
	gimbalGetAttPitchDeg	value[number]	°	luaMavsdkGimbalGetAttPitchDeg	mavlinkTelem.gimbalAtt.pitch_deg	#30 ATTITUDE	pitch * 180/PI	float [rad]	
	gimbalGetAttYawDeg	value[number]	°	luaMavsdkGimbalGetAttYawDeg	mavlinkTelem.gimbalAtt.yaw_deg_relative	#30 ATTITUDE	yaw * 180/PI	float [rad]	
Gimbal protocol v1	gimbalSendNeutralMode	none	-	luaMavsdkGimbalSendNeutralMode	mavlinkTelem.sendGimbalTargetingMode(1)	204 MAV_CMD_DO_MOUNT_CONFIGURE	1: mode = 1	-	
	gimbalSendMavlinkTargetingMode	none	-	luaMavsdkGimbalSendMavlinkTargetingMode	mavlinkTelem.sendGimbalTargetingMode(2)	204 MAV_CMD_DO_MOUNT_CONFIGURE	1: mode = 2	-	
	gimbalSendRcTargetingMode	none	-	luaMavsdkGimbalSendRcTargetingMode	mavlinkTelem.sendGimbalTargetingMode(3)	204 MAV_CMD_DO_MOUNT_CONFIGURE	1: mode = 3	-	
	gimbalSendGpsPointMode	none	-	luaMavsdkGimbalSendGpsPointMode	mavlinkTelem.sendGimbalTargetingMode(4)	204 MAV_CMD_DO_MOUNT_CONFIGURE	1: mode = 4	-	
	gimbalSendSysIdTargetingMode	none	-	luaMavsdkGimbalSendSysIdTargetingMode	mavlinkTelem.sendGimbalTargetingMode(5)	204 MAV_CMD_DO_MOUNT_CONFIGURE	1: mode = 5	-	
	gimbalSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °	luaMavsdkGimbalSendPitchYawDeg	mavlinkTelem.sendGimbalPitchYawDeg (value1, value2)	205 MAV_CMD_DO_MOUNT_CONTROL	1: Pitch = value1 2: Roll = 0 3: Yaw = value2 4: Altitude = 0 5: Latitude = 0 6: Longitude = 0 7: Mode = gimbalmanagerOut.mount_mode	[°] or [°/s] [°] or [°/s] [°] or [°/s] [m] [enum]	
STorM32 gimbal protocol v2	gimbalIsProtocolV2	value[bool]	-	luaMavsdkIsGimbalProtocolV2	mavlinkTelem.isStorm32GimbalProtocolV2()	-	-	-	returns _storm32_gimbal_protocol_v2
	gimbalSetProtocolV2	value[number]	-	luaMavsdkSetGimbalProtocolV2	mavlinkTelem.setStorm32GimbalProtocolV2(value)	-	-	-	sets _storm32_gimbal_protocol_v2=value

	gimbalClientIsReceiving	value[bool]	-	luaMavsdkGimbalClientIsReceiving	if (mavlinkTelem.gimbalmanager.is_receiving > 0) true
					else false
						#62011 STORM32_GIMBAL_MANAGER_STATUS	any	-	3.3 sec timeout

	gimbalClientIsInitialized	value[bool]	-	luaMavsdkGimbalClientIsInitialized	if ((mavlinkTelem.gimbalmanager.is_receiving > 0) and mavlinkTelem.gimbalmanager.is_initialized) true else false	#62011 STORM32_GIMBAL_MANAGER_STATUS	any and no requests waiting	-	-
	gimbalClientGetInfo	table (gimbal_manager_id[integer], gimbal_id[integer], device_capability_flags[integer], manager_capability_flags[integer])	enum MAV_COMPONENT enum MAV_COMPONENT enum MAV_STORM32_\ GIMBAL_DEVICE_CAP_FLAGS enum MAV_STORM32_\ GIMBAL_MANAGER_CAP_FLAGS	luaMavsdkGimbalClientGetInfo	mavlinkTelem.gimbalmanager.compid mavlinkTelem.gimbal.compid mavlinkTelem.gimbalmanagerInfo.\ device_cap_flags mavlinkTelem.gimbalmanagerInfo.\ manager_cap_flags	#62011 STORM32_GIMBAL_MANAGER_STATUS #0 HEARTBEAT #62010 STORM32_GIMBAL_MANAGER\ _INFORMATION #62010 STORM32_GIMBAL_MANAGER\ _INFORMATION	msg.compid (header, not payload!) msg.compid (header, not payload!) device_cap_flags manager_cap_flags	uint8_t [enum] uint8_t [enum] uint32_t [enum] uint32_t [enum]	
	gimbalClientGetStatus	table (supervisor[integer], device_flags[integer], manager_flags[integer], profile[integer])	enum MAV_STORM32_\ GIMBAL_MANAGER_CLIENT enum MAV_STORM32_\ GIMBAL_DEVICE_FLAGS enum MAV_STORM32_\ GIMBAL_MANAGER_FLAGS enum MAV_STORM32_\ GIMBAL_MANAGER_PROFILE	luaMavsdkGimbalClientGetStatus	mavlinkTelem.gimbalmanagerStatus.supervisor mavlinkTelem.gimbalmanagerStatus.device_flags mavlinkTelem.gimbalmanagerStatus.manager_flags mavlinkTelem.gimbalmanagerStatus.profile	#62011 STORM32_GIMBAL_MANAGER_STATUS (all 4)	supervisor device_flags manager_flags profile	uint8_t [enum] uint16_t [enum] uint16_t [enum] uint8_t [enum]	 0 = none 0 = default
	gimbalClientSetRetract	value[integer]{flags}	-	luaMavsdkGimbalClientSetRetract	mavlinkTelem.setStorm32GimbalClientRetract(value)	-	-	-	sets gimbalmanagerOut.device_flags
	gimbalClientSetNeutral	value[integer]{flags}	-	luaMavsdkGimbalClientSetNeutral	mavlinkTelem.setStorm32GimbalClientNeutral(value)	-	-	-	sets gimbalmanagerOut.device_flags
	gimbalClientSetLock	value1[integer]{roll_lock}, value2[integer]{pitch_lock}, value3[integer]{yaw_lock}	- - -	luaMavsdkGimbalClientSetLock	mavlinkTelem.setStorm32GimbalClientLock (value1, value2, value3)	-	-	-	gimbalmanagerOut.device_flags
	gimbalClientSetFlags	value[integer]{flags}	-	luaMavsdkGimbalClientSetFlags	mavlinkTelem.setStorm32GimbalClientFlags(value)	-	-	-	sets gimbalmanagerOut.manager_flags
	gimbalClientSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °	luaMavsdkGimbalClientSendPitchYawDeg	mavlinkTelem.sendStorm32GimbalManagerPitchYawDeg(value1, value2)	#62013 STORM32_GIMBAL_MANAGER_CONTROL_PITCHYAW	target_system = _sysid target_component = gimbalmanager.compid gimbal_id = gimbal.compid client = 3 device_flags = _t_storm32GM_gdflags manager_flags = _t_storm32GM_gmflags pitch = value1*PI/180 yaw = value2*PI/180 pitch_rate = NAN yaw_rate = NAN	uint8_t uint8_t uint8_t uint8_t [enum] uint16_t [enum] uint16_t [enum] float [rad] float [rad] float [rad/s] float [rad/s]	
	gimbalClientSendControlPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °	luaMavsdkGimbalClientSendControlPitch\ YawDeg	mavlinkTelem.sendStorm32GimbalManagerControl- PitchYawDeg(value1, value2)	#62012 STORM32_GIMBAL_MANAGER_CONTROL	target_system = _sysid target_component = gimbalmanager.compid gimbal_id = gimbal.compid client = 3 device_flags = _t_storm32GM_control_gdflags manager_flags = _t_storm32GM_control_gmflags q = calculated from value1 and value2 angular_velocity_x = NAN angular_velocity_y = NAN angular_velocity_z = NAN	uint8_t uint8_t uint8_t uint8_t uint16_t uint16_t float[4] float float float	
	gimbalClientSendCmdPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °	luaMavsdkGimbalClientSendCmdPitchYawDeg	mavlinkTelem.sendStorm32GimbalManagerCmd- PitchYawDeg(value1, value2)	#62002 MAV_CMD_STORM32_DO_GIMBAL_MANAGER_CONTROL_PITCHYAW	1: Pitch angle = value1 2: Yaw angle = value2 3: Pitch rate = NaN 4: Yaw rate = NaN 5: Gimbal device flags = _t_storm32GM_cmd_gdflags 6: Gimbal manager flags = _t_storm32GM_cmd_gmflags 7: Gimbal and cliend Ids = 3 * 256 + gimbal.compid	[°] [°] [°/s] [°/s]	
	gimbalDeviceSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °	luaMavsdkGimbalDeviceSendPitchYawDeg	mavlinkTelem.sendStorm32GimbalDevicePitch\ YawDeg (value1, value2)	#62002 STORM32_GIMBAL_DEVICE_CONTROL	target_system = _sysid target_component = gimbal.compid flags = _t_storm32GD_flags q = calculated from value1 and value2 angular_velocity_x = NAN angular_velocity_y = NAN angular_velocity_z = NAN	uint8_t uint8_t uint16_t [enum] float[4] float [rad/s] float [rad/s] float [rad/s]	

	MavSDK function	return value / parameter	Unit	MavSDK internal C++ function/wrapper	Value stems internally from or calls function(s)	MAVLink message	MAVLink msg field(s)	Data type & unit	Comments
AP EXPERIMENTAL	apSetGroundSpeed	value[number]{speed}	m/s	luaMavsdkApSetGroundSpeed	mavlinkTelem.apSetGroundSpeed(value)	178 MAV_CMD_DO_CHANGE_SPEED	1: Speed Type = 1 2: Speed = value 3: Throttle = -1 4: Relative = 1 (relative)	 [m/s} 	
	apSimpleGotoPosIntAltRel	value1[integer]{lat}, value2[integer]{lon}, value3[number]{alt}	m*e4 m*e4 m	luaMavsdkApSimpleGotoPosIntAltRel	mavlinkTelem.apSimpleGotoPosAlt (value1, value2, value3)	#73 MISSION_ITEM_INT 16 MAV_CMD_NAV_WAYPOINT	target_system = _sysid target_componetn = autopilot.compid seq = 0 frame = MAV_FRAME_GLOBAL_RELATIVE_ALT command = MAV_CMD_NAV_WAYPOINT current = 2 (=ArduPlane speciality!) autocontinue = 0 param1 = 1: Hold = 0 param2 = 2: Accept Radius = 0 param3 = 3: Pass Radius = 0 param4 = 4: Yaw = 0 x = 5: Latitude = value1 y = 6: Longitude = value2 z = 7: Altitude = value3 mission_type = MAV_MISSION_TYPE_MISSION	uint8_t uint8_t uint16_t uint8_t [enum] uint16_t [enum] uint8_t uint8_t float [s] float [m] float [m] float [°] int32_t [m*e4] int32_t [m*e4] float [m] uint8_t enum	
	apGotoPosIntAltRel	value1[integer]{lat}, value2[integer]{lon}, value3[number]{alt}	°E7 °E7 m	luaMavsdkApGotoPosIntAltRel	mavlinkTelem.apGotoPosAltYawDeg (value1, value2, value3, NAN)	#86 MAVLINK_MSG_ID_SET_POSITION_\ TARGET_GLOBAL_INT	time_boot_ms = get_tmr10ms()*10 target_system = _sysid target_component = autopilot.compid coordinate_frame = MAV_FRAME_GLOBAL_\ RELATIVE_ALT_INT type_mask = if alt != NaN then 0x0DF8 else 0x0DFC lat_int = value1 lon_int = value2 alt = if value3 != NaN then value 3, else 1 vx = 0 vy = 0 vz = 0 afx = 0 afy = 0 afz = 0 yaw = 0 yaw_rate = 0	uint32_t [ms] uint8_t uint8_t uint8_t [enum] uint16_t [bitmap] int32_t [°E7] int32_t [°E7] float [m] float [m/s] float [m/s] float [m/s] float [m/s²] float [m/s²] float [m/s²] float [rad] float [rad/s]	
	apGotoPosIntAltRelYawDeg	value1[integer]{lat}, value2[integer]{lon}, value3[number]{alt}, value4[number]{yaw}	°E7 °E7 m °	luaMavsdkApGotoPosIntAltRelYawDeg	mavlinkTelem.apGotoPosAltYawDeg (value1, value2, value3, value4)	#86 MAVLINK_MSG_ID_SET_POSITION_\ TARGET_GLOBAL_INT	time_boot_ms = get_tmr10ms()*10 target_system = _sysid target_component = autopilot.compid coordinate_frame = MAV_FRAME_GLOBAL_\ RELATIVE_ALT_INT type_mask = 0x09F8 (yaw and alt OK), 0x0DF8 (yaw=NaN, alt OK) 0x09fC (yaw OK, alt=NaN), 0x0DFC (alt and yaw=NaN) lat_int = value1 lon_int = value2 alt = if value3 != NaN then value 3, else 1 vx = 0 vy = 0 vz = 0 afx = 0 afy = 0 afz = 0 yaw = if value4 != NaN then value4*PI/180 else 0 yaw_rate = 0	uint32_t [ms] uint8_t uint8_t uint8_t enum uint16_t bitmap int32_t [°E7] int32_t [°E7] float [m] float [m/s] float [m/s] float [m/s] float [m/s²] float [m/s²] float [m/s²] float [rad] float [rad/s]	
	apGotoPosIntAltRelVel	value1[integer]{lat}, value2[integer]{lon}, value3[number]{alt}, value4[number]{vx}, value5[number]{vy}, value6[number]{vz}	°E7 °E7 m m/s m/s m/s	luaMavsdkApGotoPosIntAltRelVel	mavlinkTelem.apGotoPosAltVel (value1, value2, value3, value4, value5, value6)	#86 MAVLINK_MSG_ID_SET_POSITION_\ TARGET_GLOBAL_INT	time_boot_ms = get_tmr10ms()*10 target_system = _sysid target_component = autopilot.compid coordinate_frame = MAV_FRAME_GLOBAL_ RELATIVE_ALT_INT type_mask = 0x0DC0 lat_int = value1 lon_int = value2 alt = value3 vx = value4 vy = value5 vz = value6 afx = 0 afy = 0 afz = 0 yaw = 0 yaw_rate = 0	uint32_t [ms] uint8_t uint8_t uint8_t enum uint16_t bitmap int32_t [°E7] int32_t [°E7] float [m] float [m/s] float [m/s] float [m/s] float [m/s²] float [m/s²] float [m/s²] float [rad] float [rad/s]	

	apSetYawDeg	value1[number]{yaw},
		value2[number]{relative}
			°
			bool
				luaMavsdkApSetYawDeg	if (value2 ~= nil and value2) mavlinkTelem.apSetYawDeg(value1, true)
					else mavlinkTelem.apSetYawDeg(value1, false)
						115 MAV_CMD_CONDITION_YAW	1:Angle = if arg2 then
							fmodf(abs(value1), 360.0f) else
							fmodf(value1, 360.0f)
							2: Angular Speed = 0
							3: Direction = if arg2 then
							(if value1<0 then CCW else CW)
							else CCW
							4: Relative = if arg2 then 1 else 0
								[°]
								
								[°/s]
								-
								
								-	

	apCopterFlyClick	none	-	luaMavsdkApCopterFlyClick	mavlinkTelem.apCopterFlyClick()	42001 MAV_CMD_SOLO_BTN_FLY_CLICK	-	-	
	apCopterFlyHold	value[number]{alt}	m	luaMavsdkApCopterFlyHold	mavlinkTelem.apCopterFlyHold(value)	42002 MAV_CMD_SOLO_BTN_FLY_HOLD	1: Takeoff Altitude: value	[m]	
	apCopterFlyPause	none	-	luaMavsdkApCopterFlyPause	mavlinkTelem.apCopterFlyPause()	42003 MAV_CMD_SOLO_BTN_PAUSE_CLICK	1: Shot Mode = 0	-	
Qshot EXPERIMENTAL	qshotSendCmdConfigure	value1[integer]{mode}, value2[integer]{shot_state}	enum MAV_QSHOT_MODE -	luaMavsdkQShotSendCmdConfigure	mavlinkTelem.sendQShotCmdConfigure (value1, value2)	62020 MAV_CMD_QSHOT_DO_CONFIGURE	1: mode = value1 2: shot_state = value2	[enum]	
	qshotSendStatus	value1[integer]{mode}, value2[integer]{shot_state}	enum MAV_QSHOT_MODE -	luaMavsdkQShotSendStatus	mavlinkTelem.sendQShotStatus(value1, value2)	#62020 QSHOT_STATUS	1: mode = value1 2: shot_state = value2	uint16_t [enum] uint16_t	
	qshotGetStatus	table (mode[integer], shot_state[integer])	enum MAV_QSHOT_MODE -	luaMavsdkQShotGetStatus	mavlinkTelem.qshot.mode mavlinkTelem.qshot.shot_state	#62020 QSHOT_STATUS #62020 QSHOT_STATUS	mode shot_state	uint16_t [enum] uint16_t	
	qshotButtonState	value[integer]{state}	-	luaMavsdkQShotButtonState	mavlinkTelem.sendQShotButtonState(value)	#257 BUTTON CHANGE	time_boot_ms = get_tmr10ms()*10 last_change_ms = 0 state = value	uint32_t [ms] uint32_t [ms] uint8_t	

Debug EXPERIMENTAL	getTaskStats	table (time[integer],
		max[integer],
		load[integer])
			500ns
			500ns
			500ns
				luaMavsdkGetTaskStats	mavlinkTaskRunTime()
					mavlinkTaskRunTimeMax()
					mavlinkTaskLoad()
						-
						-
						-
							-
							-
							-
								uint16_t
								uint16_t
								uint16_t	