

# OlliW OpenTX MavSDK LUA reference

## (rev. 1.1 based on v26 firmware)

March 21<sup>st</sup>, 2021

General OpenTX LUA additions are to 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
Generic	<i>getEvent</i>	value[integer]{event}	enum, see keys.h
	<i>lockKeys</i>	value[unsigned]{mask}	-
	<i>unlockKeys</i>	-	-
	<i>isInMenu</i>	value[bool]	-

	MavSDK function	return value / parameter	Unit
Generic 1	<i>mavtelemIsEnabled</i>	value[bool]	-
	<i>isReceiving</i>	value[bool]	-
	<i>isInitialized</i>	value[bool]	-
	<i>getVersion</i>	value[string]	-
Generic 2	<i>getAutopilotType</i>	value[number]	enum MAV_AUTOPILOT
	<i>getVehicleType</i>	value[number]	enum MAV_TYPE
	<i>getFlightMode</i>	value[number]	enum PLANE_MODE or COPTER_MODE or SUB_MODE or ROVER_MODE or TRACKER_MODE
	<i>getVehicleClass</i>	value[number]	enum MAV_TYPE
	<i>getSystemStatus</i>	value[number]	enum MAV_STATE
	<i>isArmed</i>	value[bool]	-
	<i>getSystemStatusSensors</i>	table (present[number], enabled[number], health[number]) or nil	bitmap MAV_SYS_STATUS_SENSOR bitmap MAV_SYS_STATUS_SENSOR bitmap MAV_SYS_STATUS_SENSOR
IMU	<i>getAttRollDeg</i>	value[number]	°
	<i>getAttPitchDeg</i>	value[number]	°
	<i>getAttYawDeg</i>	value[number]	°
Vfr	<i>getVfrAirSpeed</i>	value[number]	m/s
	<i>getVfrGroundSpeed</i>	value[number]	m/s
	<i>getVfrAltitudeMsl</i>	value[number]	m
	<i>getVfrClimbRate</i>	value[number]	m/s
	<i>getVfrHeadingDeg</i>	value[number]	°
	<i>getVfrThrottle</i>	value[integer]	%
GPS generic	<i>getGpsCount</i>	value[integer]	bitmap
	<i>getPositionLatLonInt</i>	table (lat[integer], lon[integer])	°E7
	<i>getPositionAltitudeMsl</i>	value[number]	m
	<i>getPositionAltitudeRelative</i>	value[number]	m
	<i>getPositionHeadingDeg</i>	value[number]	°
	<i>getPositionSpeedNed</i>	table (vx[number], vy[number], vz[number])	m/s m/s m/s
GPS, 1st or only	<i>isGpsAvailable</i>	value[bool]	-
		table (fix[number], hdop[number], vdop[number], sat[number])	enum GPS_FIX_TYPE - - -
	<i>getGpsStatus</i>	value[number]	enum GPS_FIX_TYPE
	<i>getGpsFix</i>	value[number]	enum GPS_FIX_TYPE
	<i>getGpsHDop</i>	value[number]	-
	<i>getGpsVDop</i>	value[number]	-
	<i>getGpsSat</i>	value[number]	-
	<i>getGpsLatLonInt</i>	table (lat[integer], lon[integer])	°E7 °E7
	<i>getGpsAltitudeMsl</i>	value[number]	m
	<i>getGpsSpeed</i>	value[number]	m/s
	<i>getGpsCourseOverGroundDeg</i>	value[number]	°

	MavSDK function	return value / parameter	Unit
GPS, 2nd	isGps2Available	value[bool]	-
		table (fix[number], hdop[number], vdop[number], sat[number])	enum GPS_FIX_TYPE
	getGps2Status		-
	getGps2Fix	value[number]	enum GPS_FIX_TYPE
	getGps2HDop	value[number]	-
	getGps2VDop	value[number]	-
	getGps2Sat	value[number]	-
	getGps2LatLonInt	table (lat[integer], lon[integer])	°E7 °E7
	getGps2AltitudeMsl	value[number]	m
	getGps2Speed	value[number]	m/s
getGps2CourseOverGroundDeg	value[number]	°	
Battery	isBatAvailable	value[bool]	-
	isBat2Available	value[bool]	-
	getBatCount	value[integer]	-
Battery, 1st or only	getBatChargeConsumed	value[number]	mAh
	getBatEnergyConsumed	value[number]	J
	getBatTemperature	value[number]	°C
	getBatVoltage	value[number]	V
	getBatCurrent	value[number   nil]	A
	getBatRemaining	value[integer]	%
	getBatCellCount	value[integer]	-
	getBatTimeRemaining	value[integer   nil]	s
	getBatChargeState	value[integer   nil]	enum MAV_BATTERY_CHARGE_STATE
	getBatFaultBitMask	value[integer   nil]	enum MAV_BATTERY_FAULT
getBatCapacity	value[number]		
Battery, 2nd	getBat2ChargeConsumed	value[number]	mAh
	getBat2EnergyConsumed	value[number]	J
	getBat2Temperature	value[number]	°C
	getBat2Voltage	value[number]	V
	getBat2Current	value[number   nil]	A
	getBat2Remaining	value[integer]	%
	getBat2CellCount	value[integer]	-
	getBat2TimeRemaining	value[integer   nil]	s
	getBat2ChargeState	value[integer   nil]	enum MAV_BATTERY_CHARGE_STATE
	getBat2FaultBitMask	value[integer   nil]	enum MAV_BATTERY_FAULT
getBat2Capacity	value[number]		
Mission	getMission	table (count[integer], current_seq[integer])	- -
		table (seq[integer], command[integer], frame[integer], is_global[boolean], lat[integer] or x[number], lon[integer] or y[number], alt[number] or z[number])	- enum MAV_CMD_*(value) enum MAV_FRAME . °e7 or m °e7 or m °e7 or m
	getMissionItem	table (nav_bearing[number], target_bearing[number], wp_dist[number])	° ° m
	getNavController		
Messages	isStatusTextAvailable	value[bool]	-
	getStatusText	value[integer   nil] value[string   nil]	enum MAV_SEVERITY -
RF Link	getRadioRssi	value[integer]	-
	getRadioRemoteRssi	value[integer]	-
	getRadioNoise	value[integer]	2dB on SiK
	getRadioRemoteNoise	value[integer]	2dB on SiK
	getRadioRssiScaled	value[integer   nil]	
	optionGetRssiScale	value[integer]	
	optionSetRssiScale	value[integer]	
	optionIsRssiEnabled	value[bool]	-
	optionEnableRssi	value[integer]{bool}	-
	radioDisableRssiVoice	value[integer]{bool}	-

	MavSDK function	return value / parameter	Unit
AP	apIsFlying	value[bool]	-
	apIsFailsafe	value[bool]	-
	apPositionOk	value[bool]	-
	apGetArmingCheck	value[number   nil]	bitmap
	apSetFlightMode	value[integer]	enum PLANE_MODE or COPTER_MODE or SUB_MODE or ROVER_MODE or TRACKER_MODE
	apRequestBanner	none	-
	apArm	value[integer]{bool}	-
	apCopterTakeOff	value[number]{alt}	m
	apLand	none	-
apGetRangefinder	value[number]	m	
Camera	cameralsReceiving	value[bool]	-
	cameralsInitialized	value[bool]	-
	cameraGetInfo	table (compid[integer], flags[integer], has_video[bool], has_photo[bool], has_modes[bool], total_capacity[number   nil], vendor_name[string], model_name[string], firmware_version[string])	enum MAV_COMPONENT enum CAMERA_CAP_FLAGS - - - MiB - -
	cameraGetStatus	table (system_status[integer], mode[integer], video_on[boolean], photo_on[boolean], available_capacity[number   nil], battery_voltage[number   nil], battery_remaininpct[integer   nil])	enum MAV_STATE enum CAMERA_MODE - - MiB V %
	cameraSendVideoMode	none	-
	cameraSendPhotoMode	none	-
	cameraStartVideo	none	-
	cameraStopVideo	none	-
	cameraTakePhoto	none	-
Gimbal generic	gimballsReceiving	value[bool]	-
	gimballsInitialized	value[bool]	-
	gimbalGetInfo	table (compid[integer], vendor_name[string], model_name[string], custom_name[string], firmware_version[string], hardware_version[string], capability_flags[integer])	. . . . . . . . . . . . . . .
	gimbalGetStatus	table (system_status[number], custom_mode[number], is_armed[bool], prearm_ok[bool])	- - - -
	gimbalGetAttRollDeg	value[number]	°
	gimbalGetAttPitchDeg	value[number]	°
	gimbalGetAttYawDeg	value[number]	°
Gimbal protocol v1	gimbalSendNeutralMode	none	-
	gimbalSendMavlinkTargetingMode	none	-
	gimbalSendRcTargetingMode	none	-
	gimbalSendGpsPointMode	none	-
	gimbalSendSysidTargetingMode	none	-
	gimbalSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °

	MavSDK function	return value / parameter	Unit
STorm32 gimbal protocol v2	gimbalsProtocolV2	value[bool]	-
	gimbalSetProtocolV2	value[number]	-
	gimbalClientIsReceiving	value[bool]	-
	gimbalClientIsInitialized	value[bool]	-
	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
	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
	gimbalClientSetRetract	value[integer]{flags}	-
	gimbalClientSetNeutral	value[integer]{flags}	-
	gimbalClientSetLock	value1[integer]{roll_lock}, value2[integer]{pitch_lock}, value3[integer]{yaw_lock}	- - -
	gimbalClientSetFlags	value[integer]{flags}	-
	gimbalClientSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °
	gimbalClientSendControlPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °
	gimbalClientSendCmdPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °
	gimbalDeviceSendPitchYawDeg	value1[number]{pitch}, value2[number]{yaw}	° °