

Variables and Conditions for the PMDG 747 QOTSII

Introduction

To find out what is happening in the PMDG 747 Queen Of The Skies II, or cause things to happen to it, you will need to interface with Prepar3D using Variables and Events. In the Prepar3D SDK, there are long lists of these that you can use, but they are somewhat limited in the sense that they do not cover everything you will encounter in more complex aircraft, such as those built by PMDG. Way back, when it was more important to even get an aircraft looking like its real-world counterpart than ensuring its behavior also matched, most of the work went into building realistic visuals, such as gauges in the cockpit. Consequently, most of the variables that are used nowadays are still following what was defined for that part, and this is most visible in the concept of the “L-var”. To be able to build complex gauges, it was necessary to allow them to store data that could be shared across several gauges. Variables are grouped and identified by a single letter that signifies what it is for, such as “A” (the first group) for the general Simulator variables, “E” for the environment (for example, time), “K” for the Keyboard and “M” for the Mouse, and “L” for “Local”. Local variables are entirely up to the builder of the gauge, and because any addon can request variables from Prepar3D by type, they also became the most popular way to communicate with addon aircraft. However, addon builders can also simply claim a block of memory and give that a name, which can be more efficient if you have a lot of (often small) variables. PMDG has chosen this second approach for their jets.

In the PMDG SDK, which is essentially a C (or C++) header file, there is a structure definition for this block of data named “`PMDG_747QOTSII_Data`”, which in Prepar3D terms is a ClientData object. This object is read-only, in the sense that you can only use it to find out what is happening. The SDK also defines a long list of “Events”, which you can send to make changes, and they mostly correspond to all the buttons and knobs you see in the (virtual) cockpit. Events only indicate which button something is being done to, so you will also need to add a second value to tell what is happening, typically a mouse click of some sort. Some buttons and knobs have more than one setting or are rotary knobs that you can turn. Setting such a button or knob to the wanted position is a bit of a problem, so the PMDG SDK also allows you to send a value. This means we can send a value of 5 for the event concerned with the auto-break, meaning we want it set to “MAX”.

The Honeycomb bridge, which sits between the Alpha Yoke and Bravo Throttle on the one side, and Prepar3D on the other, has an optional component called the PMDG Translator, which defines L-vars for the elements of the 747 QOTSII ClientData object. It also adds a second list of L-vars, corresponding to the events from the SDK. Together these allow you to send commands to the 747 when you operate the Yoke or Throttle and optionally make that dependent on the current state of the aircraft. You can also use it to program the LEDs on the annunciator panel.

Condition variables

The condition variables in the SDK are floating-point numbers, integers, characters, or booleans, using various C types fitting their size and kind. For L-vars this has been reduced to “number”, “char”, and “bool”. Some of the values occur multiple times, for example once per pump or engine, and are indexed. This is also used for strings, which are an indexed group of characters, and the last character in the array has a value of zero. Note that “bool” values, so true and false, must be checked as numerical values 1 and 0 respectively.

List of Condition Variables and Events

The table below follows the order in which the controls and indicators are defined in the “PMDG_747QOTSII_Data” structure of the SDK. All names start with “AS_PMDG_QOTSII_”, so in this table that prefix has been dropped. Where a corresponding event is defined that allows you to change the value, it is shown in the “Event” column, also without the initial “AS_PMDG_747QOTSII”. The order of the variables follows the SDK’s control structure. The last column contains the event that can be used to change the value or influence the buttons/knobs.

Where possible I tried to put events next to the corresponding fields, but some events I could not find a match. In those cases, I put them at the end of a block of most likely relevant fields. Because the events are never provided as an array, the events will have names that reflect the meaning of the index. For example, the first field, “ELEC_GenFieldReset”, refers to a set of reset switches with guards, and the event names have a number ranging from 1 to 4. Other cases will have “L” and “R” for left and right, “L”, “C”, and “R” for left, center, and right, or “CAPT” (or “CPT”), “FO”, and “OBS” for Captain, First Officer, and Observer. Other forms are also used. In those cases, I will list the events only once, and mark in bold italics which part of the name should be changed and add a small list of the other values. If the field has a name that, after the initial group prefix, begins with “annun”, then it is an annunciator light and most likely has no function as a button or switch.

A small note on the “Unit” or type of values: if the unit is “bool” for “boolean”, you should use values 0 for “false” and 1 for “true”.

Condition variable (L-var) name	Unit	Index	Values	Event
Overhead Maintenance Panel				
Electrical				
ELEC_GenFieldReset	bool	0-3		EVT_OH_ELEC_GEN_FIELD1_SWITCH (also 2, 3, and 4) EVT_OH_ELEC_GEN_FIELD1_GUARD
ELEC_APUFieldReset	bool	0-1		EVT_OH_ELEC_APU_FIELD1_SWITCH (also 2) EVT_OH_ELEC_APU_FIELD1_GUARD
ELEC_SplitSystemBreaker	bool			EVT_OH_ELEC_SPLIT_BREAKER_SWITCH EVT_OH_ELEC_SPLIT_BREAKER_GUARD
ELEC_annunGen_FIELD_OFF	bool	0-3		
ELEC_annunAPU_FIELD_OFF	bool	0-1		
ELEC_annunSplitSystemBreaker_OPEN	bool			
Fuel				
FUEL_CWTScavengePump_Sw_ON	bool			EVT_OH_FUEL_CWT_SCAVENGE_PUMP EVT_OH_FUEL_CWT_SCAVENGE_PUMP_COVER
FUEL_Reserve23Xfer_Sw_OPEN	bool			EVT_OH_FUEL_RSV_2_3_XFR EVT_OH_FUEL_RSV_2_3_XFR_COVER
EEC Maintenance				
ENG_EECPower_Sw_TEST	bool	0-3		EVT_OH_EEC_TEST_1_SWITCH (also 2, 3, and 4) EVT_OH_EEC_TEST_1_GUARD
Flight Controls Hydraulic Valve Power				

FCTL_WingHydValve_Sw_SHUT_OFF	bool	0-3		EVT_OH_HYD_VLV_PWR_WING_1 (also 2, 3, and 4) EVT_OH_HYD_VLV_PWR_WING_1_GUARD
FCTL_TailHydValve_Sw_SHUT_OFF	bool	0-3		EVT_OH_HYD_VLV_PWR_TAIL_1 (also 2, 3, and 4) EVT_OH_HYD_VLV_PWR_TAIL_1_GUARD
FCTL_annunTailHydVALVE_CLOSED	bool	0-3		
FCTL_annunWingHydVALVE_CLOSED	bool	0-3		
Air Conditioning				
AIR_LowerLobeFlowRate_Selector	number		(744 Freighter only) 0: BOTH 1: AFT LOW ... 6: AFT HIGH	EVT_OH_LOWER_LOBE_FLOW_RATE
AIR_LowerLobeAftCargoHt_Selector	number		0: OFF, 1: LOW, 2: HIGH (748 only)	EVT_OH_AIRCOND_AFT_CARGO_HT_SELECTOR
Overhead Panel				
IRS				
IRS_Selector	number	0-2	0: OFF, 1: ALIGN, 2: NAV, 3: ATT	EVT_OH_IRU_SELECTOR_L (0: L, 1: C, 2: R)
IRS_annunON_BAT	bool			
Electrical				
ELEC_annunUtilOFF	bool	0-1		
ELEC_Battery_Sw_ON	bool			EVT_OH_ELEC_BATTERY_SWITCH EVT_OH_ELEC_BATTERY_GUARD
ELEC_APU_Selector	number		0: OFF, 1: ON, 2: START	EVT_OH_ELEC_APU_SEL_SWITCH
ELEC_StandbyPowerSw	number		0: OFF, 1: AUTO, 2: BAT	EVT_OH_ELEC_STBY_PWR_SWITCH
ELEC_APUGen_Sw_ON	bool	0-1		EVT_OH_ELEC_APU_GEN1_SWITCH (also 2)
ELEC_UtilSw	bool	0-1		EVT_OH_ELEC_L_UTIL (also R)
ELEC_BusTie_Sw_AUTO	bool	0-3		EVT_OH_ELEC_BUS_TIE1_SWITCH (also 2, 3, and 4)
ELEC_annunBusTieISLN	bool	0-3		
ELEC_Gen_Sw_ON	bool	0-3		EVT_OH_ELEC_GEN1_SWITCH (also 2, 3, and 4)
ELEC_IDGDiscSw	bool	0-3		EVT_OH_ELEC_DISCONNECT1_SWITCH (also 2, 3, and 4) EVT_OH_ELEC_DISCONNECT1_GUARD
ELEC_ExtPwrSw	bool	0-1		EVT_OH_ELEC_EXT_PWR1_SWITCH (also 2)
ELEC_annunExtPowr_ON	bool	0-1		
ELEC_annunExtPowr_AVAIL	bool	0-1		
ELEC_annunAPUGen_ON	bool	0-1		
ELEC_annunAPUGen_AVAIL	bool	0-1		
ELEC_annunGenOFF	bool	0-3		

ELEC_annunIDGDiscDRIVE	bool	0-3		EVT_OH_ELEC_GND_TESTS_SWITCH EVT_OH_ELEC_GND_TESTS_GUARD EVT_OH_ELEC_TOWING_PWR
			(744F & 747-8 only)	EVT_OH_APU_START_SOURCE
Hydraulics				
HYD_EnginePump_Sw_ON	bool	0-3		EVT_OH_HYD_ENG1 (also 2, 3, and 4)
HYD_DemandPump_Selector	number	0-3	0: OFF, 1: AUTO, 2: ON 3: AUX (for selector 4 only)	EVT_OH_HYD_DEMAND1 (also 2, 3, and 4)
HYD_annunSYS_FAULT	bool	0-3		
HYD_annunSYS_FAULT	bool	1		
HYD_annunSYS_FAULT	bool	2		
HYD_annunSYS_FAULT	bool	3		
HYD_annunEnginePumpPRESS	bool	0-3		
HYD_annunDemandPumpPRESS	bool	0-3		
HYD_RamAirTurbineSw	bool		(747-8 only)	EVT_OH_HYD_RAM_AIR EVT_OH_HYD_RAM_AIR_COVER
HYD_annunRamAirTurbinePRESS	bool		(747-8 only)	
HYD_annunRamAirTurbineUNLKD	bool		(747-8 only)	
Fire Protection				
FIRE_EngineHandle	number	0-3	0: In (normal), 1: pulled out, 2: turned left, 3: turned right	EVT_OH_FIRE_HANDLE_ENGINE_1_BOTTOM (also 2, 3, 4) EVT_OH_FIRE_HANDLE_ENGINE_1_TOP
FIRE_EngineHandleUnlock_Sw	bool	0-3		EVT_OH_FIRE_UNLOCK_SWITCH_ENGINE_1 (also 2, 3, 4)
FIRE_annunENG_BTL_DISCH	bool	0-3		
FIRE_CargoFire_Sw_Arm	bool	0-1		EVT_OH_FIRE_CARGO_ARM_FWD (also AFT)
FIRE_annunCargoFire	bool	0-1		
FIRE_MainDeckFire_Sw_Arm	bool		(Freighter only)	EVT_OH_FIRE_CARGO_ARM_MAIN_DECK
FIRE_annunMainDeckFire	bool		(Freighter only)	
FIRE_CargoFireDisch_Sw	bool			EVT_OH_FIRE_CARGO_DISCH EVT_OH_FIRE_CARGO_DISCH_GUARD
FIRE_annunCargoDISCH	bool			
FIRE_FireOvhtTest_Sw	bool			EVT_OH_FIRE_OVHT_TEST EVT_OH_FIRE_OVHT_TEST_ENG
FIRE_APUHandle	number		0: In (normal), 1: pulled out, 2: turned left, 3: turned right	EVT_OH_FIRE_HANDLE_APU_BOTTOM EVT_OH_FIRE_HANDLE_APU_TOP

FIRE_APUHandleUnlock_Sw	bool			EVT_OH_FIRE_UNLOCK_SWITCH_APU
FIRE_annunAPU_BTL_DISCH	bool			
Engine Control				
ENG_EECMode_Sw_NORM	bool	0-3		EVT_OH_EEC_1_SWITCH (also 2, 3, and 4) EVT_OH_EEC_1_GUARD
ENG_Start_Sw_Pulled	bool	0-3		EVT_OH_ENGINE_1_START (also 2, 3, and 4)
ENG_ConIgnition_Sw_ON	bool			EVT_OH_IGNITION_CON
ENG_StbyIgnition_Selector	number		0: 1, 1: NORM, 2: 2	EVT_OH_IGNITION_STBY
ENG_AutoIgnition_Selector	number		GE: SINGLE/BOTH, PW: 1/BOTH/2, RR: 1/NORM/2/BOTH	EVT_OH_IGNITION_AUTO
ENG_Autostart_Sw_ON	bool		(Freighter only)	
			(GE/RR only)	EVT_OH_ENGINE_AUTOSTART
			(PW only)	EVT_OH_ENGINE_AUTOSTART_1 (also 2, 3, and 4) EVT_OH_ENGINE_AUTOSTART_1_GUARD
ENG_Start_Light	bool	0-3		
ENG_annunALTN	bool	0-3		
				EVT_OH_EEC_CH_SEL_1_SWITCH (also 2, 3, and 4)
Fuel Panel				
FUEL_CrossFeed_Sw	bool	0-3		EVT_OH_FUEL_XFEED_1 (also 2, 3, and 4) EVT_OH_FUEL_XFEED_2_GUARD (also 3, NOT 1 and 4)
FUEL_MainPumpFwd_Sw	bool	0-3		EVT_OH_FUEL_PUMP_MAIN_FWD_1 (also 2, 3, and 4)
FUEL_MainPumpAft_Sw	bool	0-3		EVT_OH_FUEL_PUMP_MAIN_AFT_1 (also 2, 3, and 4)
FUEL_OvrPumpFwd_Sw	bool	0-1		EVT_OH_FUEL_PUMP_OVRD_FWD_2 (also 3, NOT 1, 4)
FUEL_OvrPumpAft_Sw	bool	0-1		EVT_OH_FUEL_PUMP_OVRD_AFT_2 (also 3, NOT 1, and 4)
FUEL_PumpStab_Sw	bool	0-1	(Passenger only)	EVT_OH_FUEL_PUMP_L_STAB (also R)
FUEL_PumpCtr_Sw	bool	0-1		EVT_OH_FUEL_PUMP_L_CENTER (also R)
FUEL_XferMain14_Sw	bool			EVT_OH_FUEL_XFER_MAIN_1_4 EVT_OH_FUEL_XFER_MAIN_1_4_COVER
			(747-8 only)	EVT_OH_FUEL_RSV_1_4_XFR EVT_OH_FUEL_RSV_1_4_XFR_COVER
FUEL_JettisonNozle_Sw	bool	0-1		EVT_OH_FUEL_JETTISON_NOZZLE_L (also R) EVT_OH_FUEL_JETTISON_NOZZLE_L_GUARD
FUEL_JettisonArm_Selector	number		No MLW option: 0/1: not used, 2: OFF, 3: A, 4: B MLW option: 0: SEL_A, 1: MLW_A,	EVT_OH_FUEL_JETTISON_SELECTOR

			2: OFF, 3: MLW_B, 4: SEL_B	
FUEL_FuelToRemain_Selector	number		0: DECR, 1: Neutral, 1: INCR	EVT_OH_FUEL_TO_REMAIN_KNOB
FUEL_annunXFEED_VALVE	bool	0-3		
FUEL_annunPRESS_MainFwd	bool	0-3		
FUEL_annunPRESS_MainAft	bool	0-3		
FUEL_annunPRESS_Ovrdfwd	bool	0-1		
FUEL_annunPRESS_OvrdfwdAft	bool	0-1		
FUEL_annunPRESS_Stab	bool	0-1		
FUEL_annunPRESS_Ctr	bool	0-1		
FUEL_annunJettisonNozleVALVE	bool	0-1		
				EVT_OH_FUEL_PUMP_AUX_L EVT_OH_FUEL_PUMP_AUX_R
Anti-Ice				
ICE_WingAntilceSw	number		0: OFF, 1: AUTO (748 only), 2: ON	EVT_OH_ICE_WING_ANTIICE
ICE_EngAntilceSw	number	0-3	0: OFF, 1: AUTO (748 only), 2: ON	EVT_OH_ICE_ENGINE_ANTIICE_1 EVT_OH_ICE_ENGINE_ANTIICE_2 EVT_OH_ICE_ENGINE_ANTIICE_3 EVT_OH_ICE_ENGINE_ANTIICE_4
ICE_annunEngAntilceVALVE	bool	0-3		
ICE_annunWingAntilceVALVE	bool			
ICE_WindowHeat_Sw_ON	bool	0-1		EVT_OH_ICE_WINDOW_HEAT_L EVT_OH_ICE_WINDOW_HEAT_R
			(747-8 only)	EVT_OH_WINDHT_ANTIFOG_PWR_SIDE_L (also R) EVT_OH_WINDHT_ANTIFOG_PWR_SIDE_L_GUARD EVT_OH_WINDHT_ANTIFOG_PWR_FWD_L EVT_OH_WINDHT_ANTIFOG_PWR_FWD_L_GUARD
ICE_annunWindowHeatINOP	bool	0-1		
Rain Protection				
WIPERS_Selector	number	0-1	0: OFF, 1: INT (748), 2: LOW, 3: HIGH	EVT_OH_WIPER_SWITCH_L (also R)
WASHER_Sw	bool	0-1		EVT_OH_WASHER_SWITCH_L (also R)
RAIN_REP_Sw	bool	0-1		EVT_OH_RAIN_REP_SWITCH_L (also R)
Lights Panel				
LTS_DomeLightKnob	number		0-150	EVT_OH_DOME_SWITCH
LTS_CktBkrOverheadKnob	number		0-150	

LTS_GlareshieldPNLIKnob	number		0-150	EVT_OH_GS_PANEL_LIGHT_CONTROL
LTS_GlareshieldFLOODKnob	number		0-150	EVT_OH_GS_FLOOD_LIGHT_CONTROL
LTS_AisleStandPNLKnob	number		0-150	EVT_OH_AISLE_STAND_PANEL_LIGHT_CONTROL
LTS_AisleStandFLOODKnob	number		0-150	EVT_OH_AISLE_STAND_FLOOD_LIGHT_CONTROL
LTS_Storm_Sw_ON	bool			EVT_OH_LIGHTS_STORM
LTS_IndLightsTestSw	number		0: TEST, 1: BRT, 2: DIM	EVT_OH_LIGHTS_IND_LTS_SWITCH
LTS_LandingLights_Sw_ON	bool	0-3	Order: Outbrd L then R, Inbrd L then R	EVT_OH_LIGHTS_LANDING_OUTBD_L (also R) EVT_OH_LIGHTS_LANDING_INBD_L EVT_LDG_LIGHTS_TOGGLE (all together)
LTS_Beacon_Sw	number		0: lower, 1: off, 2: both	EVT_OH_LIGHTS_BEACON
LTS_NAV_Sw_ON	bool			EVT_OH_LIGHTS_NAV
LTS_Logo_Sw_ON	bool			EVT_OH_LIGHTS_LOGO
LTS_Wing_Sw_ON	bool			EVT_OH_LIGHTS_WING
LTS_RunwayTurnoff_Sw_ON	bool	0-1		EVT_OH_LIGHTS_L_TURNOFF (also R) EVT_TURNOFF_LIGHTS_TOGGLE
LTS_Taxi_Sw_ON	bool			EVT_OH_LIGHTS_TAXI
LTS_Strobe_Sw_ON	bool			EVT_OH_LIGHTS_STROBE
				EVT_OH_PANEL_LIGHT_CONTROL EVT_OH_FLIGHT_DECK_ACCES_LIGHTS
Air Pressurization				
AIR_LdgAlt_PushOn_Sw	bool			EVT_OH_PRESS_LAND_ALT_PUSH_ON
AIR_LdgAlt_Selector	number		0: DECR, 1: neutral, 2: INCR	EVT_OH_PRESS_LAND_ALT_KNOB_ROTATE
AIR_OutflowValveMan_Sw	bool	0-1		EVT_OH_PRESS_VALVE_MAN_L (also R)
AIR_OutflowValveNeedle	number	0-1		
AIR_OutflowValves_Selector	number			EVT_OH_PRESS_VALVE_CTRL
AIR_CabinAltAuto_Selector	number			EVT_OH_PRESS_AUTO_SELECT
AIR_SmokeEvacHandle_Pulled	bool			EVT_OH_SMOKE_EVAC_HANDLE
Air Conditioning				
AIR_Pack_Selector	number	0-2	0: OFF (748), 1: NORM (748), 2: A, 3: B	EVT_OH_AIRCOND_PACK_SWITCH_L (also C and R)
AIR_TrimAir_Sw_On	bool	0-1	("_R" on 747-8 only)	EVT_OH_AIRCOND_TRIM_AIR_SWITCH EVT_OH_AIRCOND_TRIM_AIR_SWITCH_R
AIR_RecircFan_Sw_On	bool	0-1	(Passenger only)	EVT_OH_AIRCOND_RECIRC_FAN_UPP_SWITCH (also LWR)
AIR_TempSelector	number	0-5	0: Flt deck, 1: main deck fwd, 2: main deck aft, 3: lower lobe fwd,	EVT_OH_AIRCOND_TEMP_SELECTOR_FLT_DECK (Passenger only)

			4: lower lobe aft, 5: pass	EVT_OH_AIRCOND_TEMP_SELECTOR_PASS (Freighter only)
			Values: 0: C ... 60: W ... 70: MAN or ALTN	EVT_OH_AIRCOND_TEMP_SELECTOR_MAIN_DECK_FWD EVT_OH_AIRCOND_TEMP_SELECTOR_MAIN_DECK_AFT EVT_OH_AIRCOND_TEMP_SELECTOR_LWR_LOBE_FWD EVT_OH_AIRCOND_TEMP_SELECTOR_LWR_LOBE_AFT
AIR_PackReset_Sw_Pushed	bool			EVT_OH_AIRCOND_PACK_RST_SWITCH
AIR_EquipCooling_Selector	number		0: STBY, 1: NORM, 2: OVRD	EVT_OH_AIRCOND_EQUIP_COOLING_SWITCH
AIR_HighFlow_Sw_On	bool			EVT_OH_AIRCOND_HI_FLOW_SWITCH
AIR_Gasper_Sw_On	bool		(Passenger 744 only)	EVT_OH_AIRCOND_GASPER_SWITCH
AIR_FltDeckFan_Sw_On	bool		(Freighter only)	EVT_OH_AIRCOND_FLT_DECK_FAN_SWITCH
AIR_AftCargoHeat_Sw_On	bool		(744 only)	EVT_OH_AIRCOND_AFT_CARGO_HT_SWITCH
AIR_ZoneReset_Sw_Pushed	bool		(744 only)	EVT_OH_AIRCOND_ZONE_RST_SWITCH
AIR_AltVent_Sw_On	bool		(748 only)	EVT_OH_AIRCOND_ALT_VENT_SWITCH EVT_OH_AIRCOND_ALT_VENT_GUARD
AIR_AltVent_Selector	number		(748 only) 0: CLOSE, 1: NEUTRAL, 2: OPEN	EVT_OH_AIRCOND_ALT_VENT_SELECTOR
AIR_annunPackOFF	bool	0-2	(748 only)	
AIR_annunPack_SYS_FAIL	bool			
AIR_annunZone_SYS_FAIL	bool		(744 only)	
AIR_annunAftCragoHeat_TEMP	bool		(744 only)	
			(Passenger 747-8 only)	EVT_OH_AIRCOND_AFT_CARGO_TEMP
Bleed Air				
AIR_EngBleedAir_Sw_ON	bool	0-3		EVT_OH_BLEED_ENG_1_SWITCH (also 2, 3, and 4)
AIR_APUBleedAir_Sw_ON	bool			EVT_OH_BLEED_APU_SWITCH
AIR_IsolationValve_Sw	bool	0-1		EVT_OH_BLEED_ISOLATION_VALVE_SWITCH_L (also R)
AIR_annunEngBleedAirOFF	bool	0-3		
AIR_annunAPUBleedAirVALVE	bool			
AIR_annunIsolationVALVE	bool	0-1		
AIR_annun_SYS_FAULT	bool	0-3	(744 only)	
Overhead – Miscellaneous				
LTS_EmerLightsSelector	number		0: OFF, 1: ARMED, 2: ON)	EVT_OH_EMER_EXIT_LIGHT_SWITCH EVT_OH_EMER_EXIT_LIGHT_GUARD
COMM_CAPTAudio_Selector	number		0: NORM, 1: VHF-L DIRECT (Freighter)	EVT_CAPT_AUDIO_SELECTOR

COMM_OBSAudio_Selector	number		0: CAPT, 1: NORM, 2: FO	EVT_OBS_AUDIO_SELECTOR
COMM_ServiceInterphoneSw	bool			EVT_OH_SERVICE_INTERPHONE_SWITCH
COMM_CargoCabinInterphoneSw	bool			EVT_OH_CABIN_CARGO_INTERPHONE_SWITCH
OXY_Oxygen_Sw_On	bool		PASS OXY or SUPRNMRY OXY (freighter)	EVT_OH_OXYGEN_SWITCH EVT_OH_OXYGEN_GUARD
FCTL_YawDamper_Sw	bool	0-1		EVT_OH_YAW_DAMPER_UPR (also LWR)
FCTL_annunYawDamperINOP	bool	0-1		
				EVT_OH_CENTER_AIR_DATA_SELECTOR EVT_OH_CVR_TEST EVT_OH_CVR_ERASE EVT_OH_EMU_MAINT_PWR_SWITCH EVT_OH_EMU_MAINT_PWR_GUARD EVT_OH_FIRE_SQUIB_TEST1 EVT_OH_FIRE_SQUIB_TEST2 EVT_OH_FIRE_SQUIB_TEST_FWD EVT_OH_FIRE_SQUIB_TEST_AFT
Glareshield – Warnings				
WARN_Reset_Sw_Pushed	bool	0-1		EVT_MASTER_WARNING_RESET_L (also R)
WARN_annunMASTER_WARNING	bool	0-1		
WARN_annunMASTER_CAUTION	bool	0-1		
Glareshield – ISFD				
EFIS_MinsSelBARO	bool	0-1		EVT_EFIS_CPT_MINIMUMS_RADIO_BARO (0: CPT, 1: FO)
EFIS_BaroSelHPA	bool	0-1		EVT_EFIS_CPT_BARO_IN_HPA (0: CPT, 1: FO)
EFIS_VORADFSel1	number	0-1	0: VOR, 1: OFF, 2: ADF	EVT_EFIS_CPT_VOR_ADF_SELECTOR_L (0: CPT, 1: FO)
EFIS_VORADFSel2	number	0-1	0: VOR, 1: OFF, 2: ADF	EVT_EFIS_CPT_VOR_ADF_SELECTOR_R (0: CPT, 1: FO)
EFIS_ModeSel	number	0-1	0: APP, 1: VOR, 2: MAP, 3: PLAN	EVT_EFIS_CPT_MODE (0: CPT, 1: FO)
EFIS_RangeSel	number	0-1	0: 10, ... 6: 640	EVT_EFIS_CPT_RANGE (0: CPT, 1: FO)
EFIS_MinsKnob	number	0-1	0-99	EVT_EFIS_CPT_MINIMUMS (0: CPT, 1: FO)
EFIS_BaroKnob	number	0-1	0-99	EVT_EFIS_CPT_BARO (0: CPT, 1: FO)
EFIS_MinsRST_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_MINIMUMS_RST (0: CPT, 1: FO)
EFIS_BaroSTD_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_BARO_STD (0: CPT, 1: FO)
EFIS_ModeCTR_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_MODE_CTR (0: CPT, 1: FO)
EFIS_RangeTFC_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_RANGE_TFC (0: CPT, 1: FO)
EFIS_WXR_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_WXR (0: CPT, 1: FO)
EFIS_STA_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_STA (0: CPT, 1: FO)

EFIS_WPT_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_WPT (0: CPT, 1: FO)
EFIS_ARPT_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_ARPT (0: CPT, 1: FO)
EFIS_DATA_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_DATA (0: CPT, 1: FO)
EFIS_POS_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_POS (0: CPT, 1: FO)
EFIS_TERR_Sw_Pushed	bool	0-1		EVT_EFIS_CPT_TERR (0: CPT, 1: FO)
				EVT_ALT_NEFIS_SWITCH EVT_EFIS_CPT_MTRS (also FO) EVT_EFIS_CPT_FPV
Glareshield – Display Select Panel				
DSP_L_INBD_Sw	bool		(748 only)	EVT_DSP_L_INBD_SWITCH
DSP_R_INBD_Sw	bool		(748 only)	EVT_DSP_R_INBD_SWITCH
DSP_LWR_CTR_Sw	bool		(748 only)	EVT_DSP_LWR_CTR_SWITCH
DSP_ENG_Sw	bool			EVT_DSP_ENG_SWITCH
DSP_STAT_Sw	bool			EVT_DSP_STAT_SWITCH
DSP_ELEC_Sw	bool			EVT_DSP_ELEC_SWITCH
DSP_HYD_Sw	bool			EVT_DSP_HYD_SWITCH
DSP_FUEL_Sw	bool			EVT_DSP_FUEL_SWITCH
DSP_ECS_Sw	bool			
DSP_DRS_Sw	bool			EVT_DSP_DOOR_SWITCH
DSP_GEAR_Sw	bool			EVT_DSP_GEAR_SWITCH
DSP_FCTL_Sw	bool		(748 only)	EVT_DSP_FCTL_SWITCH
DSP_INFO_Sw	bool		(748 only)	EVT_DSP_INFO_SWITCH
DSP_CHKL_Sw	bool		(748 only)	EVT_DSP_CHKL_SWITCH
DSP_NAV_Sw	bool		(748 only)	EVT_DSP_NAV_SWITCH
DSP_CANC_Sw	bool		(CANC for 744, CANC/RCL for 748)	EVT_DSP_CANC_SWITCH
DSP_RCL_Sw	bool		(744 only)	EVT_DSP_RCL_SWITCH
DSP_annunL_INBD	bool			
DSP_annunR_INBD	bool			
DSP_annunLWR_CTR	bool			
Glareshield – Mode Control Panel				
MCP_IASMach	number		Mach if < 10.0	EVT_MCP_IAS_SET (sets value if IAS active) EVT_MCP_MACH_SET (sets <i>value</i> * 0.01, if MACH active)
MCP_IASBlank	bool			
MCP_Heading	number			EVT_MCP_HDG_SET

MCP_Altitude	number			EVT_MCP_ALT_SET
MCP_VertSpeed	number			EVT_MCP_VS_SET (sets <i>value</i> – 10000, if VS window open)
MCP_VertSpeedBlank	bool			
MCP_FD_Sw_On	bool	0-1		EVT_MCP_FD_SWITCH_L (also R)
MCP_ATArm_Sw_On	bool			EVT_MCP_AT_ARM_SWITCH
MCP_BankLimitSel	number		0: AUTO, 1: 5, 2: 10, 3: 15, 4: 20, 5: 25	EVT_MCP_BANK_ANGLE_SELECTOR
MCP_DisengageBar	bool			EVT_MCP_DISENGAGE_BAR
MCP_Speed_Dial	number			EVT_MCP_SPEED_SELECTOR
MCP_Heading_Dial	number			EVT_MCP_HEADING_SELECTOR
MCP_Altitude_Dial	number			EVT_MCP_ALTITUDE_SELECTOR
MCP_VS_Wheel	number			EVT_MCP_VS_SELECTOR
MCP_AP_Sw_Pushed	bool	0-2		EVT_MCP_AP_L_SWITCH (also C and R)
MCP_THR_Sw_Pushed	bool			EVT_MCP_THR_SWITCH
MCP_SPD_Sw_Pushed	bool			EVT_MCP_SPD_SWITCH
MCP_LNAV_Sw_Pushed	bool			EVT_MCP_LNAV_SWITCH
MCP_VNAV_Sw_Pushed	bool			EVT_MCP_VNAV_SWITCH
MCP_FLCH_Sw_Pushed	bool			EVT_MCP_LVL_CHG_SWITCH
MCP_HDG_HOLD_Sw_Pushed	bool			EVT_MCP_HDG_HOLD_SWITCH
MCP_VS_Sw_Pushed	bool			EVT_MCP_VS_SWITCH
MCP_ALT_HOLD_Sw_Pushed	bool			EVT_MCP_ALT_HOLD_SWITCH
MCP_LOC_Sw_Pushed	bool			EVT_MCP_LOC_SWITCH
MCP_APP_Sw_Pushed	bool			EVT_MCP_APP_SWITCH
MCP_Speed_Sw_Pushed	bool			EVT_MCP_SPEED_PUSH_SWITCH
MCP_Heading_Sw_Pushed	bool			EVT_MCP_HEADING_PUSH_SWITCH
MCP_Altitude_Sw_Pushed	bool			EVT_MCP_ALTITUDE_PUSH_SWITCH
MCP_IAS_MACH_Toggle_Sw_Pushed	bool			EVT_MCP_IAS_MACH_SWITCH
MCP_annunAP	bool	0-2		
MCP_annunTHR	bool			
MCP_annunSPD	bool			
MCP_annunLNAV	bool			
MCP_annunVNAV	bool			
MCP_annunFLCH	bool			
MCP_annunHDG_HOLD	bool			

MCP_annunVS	bool			
MCP_annunALT_HOLD	bool			
MCP_annunLOC	bool			
MCP_annunAPP	bool			
				EVT_MCP_TOGA_SCREW_L EVT_MCP_TOGA_SCREW_R
Glareshield – Miscellaneous				
DSP_InbdDspl_L_Selector	number		0: EICAS, 1: NORM, 2: PFD (744) 0: ND, 1: NAV, 2: MFD, 3: EICAS (748)	EVT_DSP_INB_DSPL_L EVT_LEFT_INBD_TERR_BRIGHT_CONTROL
DSP_LwrDspl_L_Selector	number		0: EICAS PRI, 1: NORM, 2: ND (744)	EVT_DSP_LWR_DSPL_L
DSP_InbdDspl_R_Selector	number		0: PFD, 1: NORM, 2: EICAS (744) 0: EICAS, 1: MFD, 2: NAV, 3: PFD (748)	EVT_DSP_INB_DSPL_R EVT_RIGHT_INBD_TERR_BRIGHT_CONTROL
DSP_LwrDspl_R_Selector	number		0: ND PRI, 1: NORM, 2: EICAS (744)	EVT_DSP_LWR_DSPL_R
ISP_FMC_Selector	number		0: LEFT, 1: AUTO, 2: RIGHT	EVT_FWD_FMC_SELECTOR
ISP_EIU_C_Selector	number		0: L, 1: C, 2: R	EVT_FWD_EIU_SOURCE_C
LTS_UpperDsplBRIGHTNESSKnob	number		0-150	EVT_FWD_UPPER_BRIGHT_CONTROL
LTS_LowerDsplBRIGHTNESSKnob	number		0-150	EVT_FWD_LOWER_BRIGHT_CONTROL
EICAS_EventRcd_Sw_Pushed	bool			EVT_FWD_EICAS_EVENT_RCD
EFIS_HdgRef_Sw_Norm	bool			EVT_EFIS_HDG_REF_SWITCH
FCTL_AltFlaps_Sw_ARM	bool			EVT_ALT_FLAPS_ARM
FCTL_AltFlaps_Control_Sw	number		0: RET, 1: OFF, 2: EXT	EVT_ALT_FLAPS_POS EVT_GLARESHIELD_PTT_L (also R) EVT_CLOCK_L (also R)
Gears				
GEAR_Lever	number		0: RET, 1: OFF, 2: DOWN	EVT_GEAR_LEVER EVT_GEAR_LEVER_OFF
GEAR_LockOvrD_Sw	bool			EVT_GEAR_LEVER_UNLOCK
GEAR_AltGearNoseBody_Sw_DPushed	bool			EVT_GEAR_ALT_GEAR_NOSE_BODY EVT_GEAR_ALT_GEAR_NOSE_BODY_GUARD
GEAR_AltGearWing_Sw_DPushed	bool			EVT_GEAR_ALT_GEAR_WING EVT_GEAR_ALT_GEAR_WING_GUARD

Ground Proximity Warning System

GPWS_GSInhibit_Sw	Bool		EVT_GPWS_GS_INHIBIT_SWITCH
GPWS_annunGND_PROX_top	bool		
GPWS_annunGND_PROX_bottom	bool		
GPWS_FlapInhibitSw_OVRD	bool		EVT_GPWS_FLAP_OVRD_SWITCH EVT_GPWS_FLAP_OVRD_GUARD
GPWS_GearInhibitSw_OVRD	bool		EVT_GPWS_GEAR_OVRD_SWITCH EVT_GPWS_GEAR_OVRD_GUARD
GPWS_TerrInhibitSw_OVRD	bool		EVT_GPWS_TERR_OVRD_SWITCH EVT_GPWS_TERR_OVRD_GUARD
			EVT_GPWS_RWY_OVRD_SWITCH EVT_GPWS_RWY_OVRD_GUARD
			EVT_FWD_LOWER_TERR_BRIGHT_CONTROL

ISDF

ISFD_Baro_Sw_Pushed	bool		EVT_ISFD_BARO EVT_ISFD_BARO_PUSH
ISFD_RST_Sw_Pushed	bool		EVT_ISFD_ATT_RST
ISFD_Minus_Sw_Pushed	bool		EVT_ISFD_MINUS
ISFD_Plus_Sw_Pushed	bool		EVT_ISFD_PLUS
ISFD_APP_Sw_Pushed	bool		EVT_ISFD_APP
ISFD_HP_IN_Sw_Pushed	bool		EVT_ISFD_HP_IN
			EVT_CLICKSPOT_ISFD EVT_STANDBY_ADI_CAGE_KNOB EVT_STANDBY_ALT_BARO_KNOB EVT_STANDBY_ADI_APPR_MODE

Forward Left Panel

ISP_FltDir_L_Selector	number	0: L, 1: C, 2: R	EVT_FWD_FD_SOURCE_L
ISP_Nav_L_Selector	number	0: FMC L, 1: FMC R, 2: CDU L, 3: CDU C	EVT_FWD_NAV_SOURCE_L
ISP_EIU_L_Selector	number	0: L, 1: C, 2: R	EVT_FWD_EIU_SOURCE_L
ISP_IRS_L_Selector	number	0: L, 1: C, 2: R	EVT_FWD_IRS_SOURCE_L
ISP_AirData_L_Selector	number	0: L, 1: C, 2: R	EVT_FWD_AIR_DATA_SOURCE_L
BRAKES_BrakePressNeedle	number	0-100 (corresponds to 0-4000 PSI)	
BRAKES_annunBRAKE_SOURCE	bool		

Forward Right Panel

ISP_FltDir_R_Selector	number		0: R, 1: C, 2: L	EVT_FWD_FD_SOURCE_R
ISP_Nav_R_Selector	number		0: FMC R, 1: FMC L, 2: CDU R, 3: CDU C	EVT_FWD_NAV_SOURCE_R
ISP_EIU_R_Selector	number		0: R, 1: C, 2: L	EVT_FWD_EIU_SOURCE_R
ISP_IRS_R_Selector	number		0: R, 1: C, 2: L	EVT_FWD_IRS_SOURCE_R
ISP_AirData_R_Selector	number		0: R, 1: C, 2: L	EVT_FWD_AIR_DATA_SOURCE_R
Left Sidewall – Cockpit Lights				
LTS_LeftFwdPanelPNLKnob	number		0-150	EVT_LEFT_PANEL_LIGHT_CONTROL
LTS_LeftFwdPanelFLOODKnob	number		0-150	EVT_LEFT_FLOOD_LIGHT_CONTROL
LTS_LeftOutbdDsplBRIGHTNESSKnob	number		0-150	EVT_LEFT_OUTBD_BRIGHT_CONTROL
LTS_LeftInbdDsplBRIGHTNESSKnob	number		0-150	EVT_LEFT_INBD_BRIGHT_CONTROL
Right Sidewall – Cockpit lights				
LTS_RightFwdPanelPNLKnob	number		0-150	EVT_RIGHT_PANEL_LIGHT_CONTROL
LTS_RightFwdPanelFLOODKnob	number		0-150	EVT_RIGHT_FLOOD_LIGHT_CONTROL
LTS_RightInbdDsplBRIGHTNESSKnob	number		0-150	EVT_RIGHT_INBD_BRIGHT_CONTROL
LTS_RightOutbdDsplBRIGHTNESSKnob	number		0-150	EVT_RIGHT_OUTBD_BRIGHT_CONTROL
				EVT_COCKPIT_LIGHTS_TOGGLE EVT_PANEL_LIGHTS_TOGGLE EVT_FLOOD_LIGHTS_TOGGLE
Left and Right Sidewalls – Cockpit Airconditioning				
AIR_ShoulderHeaterKnob	number	0-1	0: HI, 1: LO, 2: OFF	EVT_FWD_LEFT_SHOULDER_HEATER (also RIGHT)
AIR_FootHeaterSelector	number	0-1	0: HI, 1: LO, 2: OFF	EVT_FWD_LEFT_FOOT_HEATER (also RIGHT)
AIR_WShldAirSelector	number	0-1	0: ON, 1: OFF	EVT_FWD_LEFT_WSHLD_AIR (also RIGHT)
Chronometers				
CHR_Chr_Sw_Pushed	bool	0-1		EVT_CHRONO_L_CHR (also R)
CHR_Date_Sw_Pushed	bool	0-1		EVT_CHRONO_L_DATE (also R)
CHR_Set_Selector	number	0-1	0: RUN, 1: HLDY, 2: MM, 3: HD	EVT_CHRONO_L_SET (also R)
CHR_ET_Selector	number	0-1	0: RESET, 1: HLD, 2: RUN	EVT_CHRONO_L_ET (also R)
Yokes				
				EVT_YOKE_L_AP_DISC_SWITCH EVT_YOKE_R_AP_DISC_SWITCH EVT_YOKE_L_DIGIT_WHEEL_1 EVT_YOKE_L_DIGIT_WHEEL_2 EVT_YOKE_L_DIGIT_WHEEL_3 EVT_YOKE_R_DIGIT_WHEEL_1

				EVT_YOKE_R_DIGIT_WHEEL_2 EVT_YOKE_R_DIGIT_WHEEL_3 EVT_YOKE_L_SLIDER EVT_YOKE_R_SLIDER
Control Stand – Flight controls				
FCTL_StabCutOutSw_2_NORMAL	bool			EVT_CONTROL_STAND_STABCUTOUT_2_SWITCH EVT_CONTROL_STAND_STABCUTOUT_2_SWITCH_GUARD
FCTL_StabCutOutSw_3_NORMAL	bool			EVT_CONTROL_STAND_STABCUTOUT_3_SWITCH EVT_CONTROL_STAND_STABCUTOUT_3_SWITCH_GUARD
FCTL_AltPitch_Switches	number		0: Node down, 1: neutral, 2: nose up	EVT_CONTROL_STAND_ALT_PITCH_TRIM_SWITCHES
FCTL_Speedbrake_Lever	number		0-100, 0: DOWN, 5: ARMED, 26-100: DEPLOYED	EVT_CONTROL_STAND_SPEED_BRAKE_LEVER EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_DOWN EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_ARM EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_FLT_DET EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_UP
FCTL_Flaps_Lever	number		0: UP 1: 1 2: 5 3: 10 4: 20 5: 25 6: 30	EVT_CONTROL_STAND_FLAPS_LEVER EVT_CONTROL_STAND_FLAPS_LEVER_0 EVT_CONTROL_STAND_FLAPS_LEVER_1 EVT_CONTROL_STAND_FLAPS_LEVER_5 EVT_CONTROL_STAND_FLAPS_LEVER_10 EVT_CONTROL_STAND_FLAPS_LEVER_20 EVT_CONTROL_STAND_FLAPS_LEVER_25 EVT_CONTROL_STAND_FLAPS_LEVER_30
ENG_FuelControl_Sw_RUN	bool	0-3		EVT_CONTROL_STAND_ENG1_FUEL_CTRL_SWITCH (2,3,4) EVT_FUEL_CTRL_SWITCHES_ALL_CUTOFF
BRAKES_ParkingBrakeLeverOn	bool			EVT_CONTROL_STAND_PARK_BRAKE_LEVER EVT_CONTROL_STAND_REV_THRUST1_LEVER (2, 3, 4) EVT_CONTROL_STAND_FWD_THRUST1_LEVER (2, 3, 4) EVT_CONTROL_STAND_TOGA1_SWITCH (also 2) EVT_CONTROL_STAND_AT1_DISENGAGE_SWITCH (also 2)
Forward Aisle Stand Panel – CDUs				
CDU_annunEXEC	bool	0-2		
CDU_annunDSPY	bool	0-2		
CDU_annunFAIL	bool	0-2		
CDU_annunMSG	bool	0-2		

CDU_annunOFST	bool	0-2		
CDU_BrtKnob	number	0-2	0: DECR, 1: neutral, 2: INCR	EVT_CDU_L_BRITENESS (also C and R)
				EVT_CDU_L_ZOOM (also C and R)
				EVT_CDU_L_SCRATCHPAD
Aft Aisle Stand Panel – Communications				
COMM_SelectedMic	number	0-2	-1: No microphone selected 0: VHFL, 1: VHFC, 2: VHFR, 3: FLT, 4: CAB, 5: PA, 6: HFL, 7: HFR, 8: SAT1, 9: SAT2, 10: SPKR, 11: VOR/ADF, 12: APP	EVT_ACP_CAPT_MIC_VHFL (also FO and OBS) EVT_ACP_CAPT_MIC_VHFC EVT_ACP_CAPT_MIC_VHFR EVT_ACP_CAPT_MIC_FLT EVT_ACP_CAPT_MIC_CAB EVT_ACP_CAPT_MIC_PA EVT_ACP_CAPT_MIC_HFL EVT_ACP_CAPT_MIC_HFR EVT_ACP_CAPT_MIC_SAT1 EVT_ACP_CAPT_MIC_SAT2
COMM_ReceiverSwitches	number	0-2	Bit 0: VHFL Bit 1: VHFC Bit 2: VHFR, Bit 3: FLT Bit 4: CAB Bit 5: PA, Bit 6: HFL Bit 7: HFR Bit 8: SAT1 Bit 9: SAT2 Bit 10: SPKR Bit 11: VOR/ADF Bit 12: APP	EVT_ACP_CAPT_REC_VHFL (also FO and OBS) EVT_ACP_CAPT_REC_VHFC EVT_ACP_CAPT_REC_VHFR EVT_ACP_CAPT_REC_FLT EVT_ACP_CAPT_REC_CAB EVT_ACP_CAPT_REC_PA EVT_ACP_CAPT_REC_HFL EVT_ACP_CAPT_REC_HFR EVT_ACP_CAPT_REC_SAT1 EVT_ACP_CAPT_REC_SAT2 EVT_ACP_CAPT_REC_SPKR EVT_ACP_CAPT_REC_VORADF EVT_ACP_CAPT_REC_APP
COMM_SelectedRadio	number	0-2	0: VHFL, 1: VHFC, 2: VHFR, 3: HFL, 5: HFR (4 not used)	EVT_ACP_CAPT_MIC_INT_SWITCH (also FO and OBS)
COMM_RadioTransfer_Sw_Pushed	bool	0-2		EVT_COM1_TRANSFER_SWITCH (also COM2 and COM3)
COMM_RadioPanelOff	bool	0-2		EVT_COM1_PNL_OFF_SWITCH (also COM2 and COM3)
COMM_annunAM	bool	0-2		
				EVT_ACP_CAPT_FILTER_SELECTOR (also FO and OBS) EVT_ACP_CAPT_VOR_ADF_SELECTOR EVT_ACP_CAPT_APP_SELECTOR

				EVT_COM1_HF_SENSOR_KNOB (also COM2 and COM3) EVT_COM1_OUTER_SELECTOR EVT_COM1_INNER_SELECTOR EVT_COM1_VHFL_SWITCH EVT_COM1_VHFC_SWITCH EVT_COM1_VHFR_SWITCH EVT_COM1_HFL_SWITCH EVT_COM1_AM_SWITCH EVT_COM1_HFR_SWITCH EVT_COM1_TEST_SWITCH
				EVT_COM1_DATA_SWITCH (also _COM2_ and _COM3_) EVT_COM1_KEY_PAD_1 EVT_COM1_KEY_PAD_2 EVT_COM1_KEY_PAD_3 EVT_COM1_KEY_PAD_4 EVT_COM1_KEY_PAD_5 EVT_COM1_KEY_PAD_6 EVT_COM1_KEY_PAD_7 EVT_COM1_KEY_PAD_8 EVT_COM1_KEY_PAD_9 EVT_COM1_KEY_PAD_RCL EVT_COM1_KEY_PAD_0 EVT_COM1_KEY_PAD_CLR
Pedestal – Transponder				
XPDR_XpndrSelector_R	bool		True: right, false: left	
XPDR_ModeSel	number		0: STBY, 1: ALT RPTG OFF, 2: XPNDR, 3: TA ONLY, 4: TA/RA	EVT_TCAS_MODE
XPDR_Ident_Sw_Pushed	bool		(747-400 & 747-8)	EVT_TCAS_IDENT
			(747-400 only)	EVT_TCAS_KNOB_L_OUTER EVT_TCAS_KNOB_L_INNER EVT_TCAS_KNOB_R_OUTER EVT_TCAS_KNOB_R_INNER EVT_TCAS_TEST
			(747-400 & 747-8)	EVT_TCAS_XPNDR
			(747-8 only)	EVT_TCAS_AIRSPACE_SELECTOR EVT_TCAS_KEYPAD_0 (also 1 through 7)

			EVT_TCAS_KEYPAD_CLR
Pedestal – Weather Radar			
			EVT_WXR_RANGE_SELECTOR EVT_WXR_SYSTEM_SWITCH EVT_WXR_GAIN_CONTROL EVT_WXR_MODE_SELECTOR EVT_WXR_TILT_CONTROL
			EVT_WXR_C2100_L_GAIN_CONTROL EVT_WXR_C2100_L_TILT_CONTROL EVT_WXR_C2100_R_GAIN_CONTROL EVT_WXR_C2100_R_TILT_CONTROL EVT_WXR_C2100_OFFSET_R EVT_WXR_C2100_AUTO EVT_WXR_C2100_L_R EVT_WXR_C2100_TEST EVT_WXR_C2100_L_TFR (also R) EVT_WXR_C2100_L_WX EVT_WXR_C2100_L_WX_T EVT_WXR_C2100_L_MAP EVT_WXR_C2100_L_GC
Pedestal – Auto-break selector			
BRAKES_AutobrakeSelector	number	0: RTO, 1: OFF, 2: DISARM, 3-5: 1-max	EVT_ABS_AUTOBRAKE_SELECTOR
Pedestal – Trimmings			
FCTL_AileronTrim_Switches	number	0: left wing down, 1: neutral, 2: right wing down	EVT_FCTL_AILERON_TRIM
FCTL_RudderTrim_Knob	number	0: nose left, 1: neutral, 2: nose right	EVT_FCTL_RUDDER_TRIM EVT_FCTL_RUDDER_CTR
Pedestal – Signs			
SIGNS_NoSmokingSelector	number	0: OFF, 1: AUTO, 1: ON	EVT_PED_NO_SMOKING_LIGHT_SWITCH
SIGNS_SeatBeltsSelector	number	0: OFF, 1: AUTO, 1: ON	EVT_PED_FASTEN_BELTS_LIGHT_SWITCH
Pedestal – Evacuation			
EVAC_Command_Sw_ON	bool		EVT_PED_EVAC_SWITCH EVT_PED_EVAC_SWITCH_GUARD
EVAC_PressToTest_Sw_Pressed	bool		EVT_PED_EVAC_TEST_SWITCH
EVAC_HornSutOff_Sw_Pulled	bool		EVT_PED_EVAC_HORN_SHUTOFF

EVAC_LightIlluminated	bool			
Pedestal – Miscellaneous				
				EVT_FLT_DK_DOOR_SELECTOR EVT_MAIN_DECK_ALERT_SWITCH EVT_MAIN_DECK_ALERT_COVER EVT_CONTROL_STAND_CCD_SEL_L EVT_CONTROL_STAND_CCD_TURN_L EVT_CONTROL_STAND_CCD_SEL_R EVT_CONTROL_STAND_CCD_TURN_R
				EVT_ATTENDANT_ADVISORY EVT_PED_CALL_UD EVT_PED_CALL_CREW_REST_LEFT EVT_PED_CALL_CARGO EVT_PED_CALL_GND EVT_PED_CALL_KEY_NXT EVT_PED_CALL_KEY_RST EVT_PED_CALL_KEY_1 EVT_PED_CALL_KEY_2 EVT_PED_CALL_KEY_3 EVT_PED_CALL_KEY_4 EVT_PED_CALL_KEY_5 EVT_PED_CALL_KEY_6 EVT_PED_CALL_KEY_P
Miscellaneous Info				
DOOR_state	number	0-19	0: open, 1: closed, 2: closed and armed, 3: closing, 4: opening	EVT_DOOR_1L EVT_DOOR_1R EVT_DOOR_2L EVT_DOOR_2R EVT_DOOR_3L EVT_DOOR_3R EVT_DOOR_4L EVT_DOOR_4R EVT_DOOR_5L EVT_DOOR_5R EVT_DOOR_UPPER_DECK_L EVT_DOOR_UPPER_DECK_R EVT_DOOR_CARGO_FWD

				EVT_DOOR_CARGO_AFT EVT_DOOR_CARGO_BULK EVT_DOOR_CARGO_SIDE EVT_DOOR_CARGO_NOSE EVT_DOOR_MAIN_ELEC EVT_DOOR_CTR_ELEC EVT_DOOR_FD_OVERHEAD
ENG_StartValve	bool	0-3		
AIR_DuctPress	number	0-2	(748 only) value in PSI	
FUEL_TankQty	number	0-8	Value in LBS. 0: main1, 1: main2, 2: main3, 3: main4, 4: L wing, 5: R wing, 6: CWT, 7: STAB, 8: CTR res	
IRS_aligned	bool			
AircraftModel	number		1: 400, 2: 400BCF, 3: 400M, 4: 400D, 5: 400ER, 6: 400F, 7: 400ERF, 8: 8j, 9: 8F	
WeightInKg	bool			
GPWS_V1CallEnabled	bool			
GroundConnAvailable	bool			
FMC_TakeoffFlaps	number		In degrees, 0 if not set	
FMC_V1	number		In knots, 0 if not set	
FMC_VR	number		In knots, 0 if not set	
FMC_V2	number		In knots, 0 if not set	
FMC_LandingFlaps	number		In degrees, 0 if not set	
FMC_LandingVREF	number		In knots, 0 if not set	
FMC_CruiseAlt	number		In feet, 0 if not set	
FMC_LandingAltitude	number		In feet, -32767 if not set	
FMC_TransitionAlt	number		In feet	
FMC_TransitionLevel	number		In feet	
FMC_PerfInputComplete	bool			
FMC_DistanceToTOD	number		In NM, 0 if passed, <0 if unknown	
FMC_DistanceToDest	number		In NM, <0 if unknown	
FMC_flightNumber	char	0-8		

ELEC_annunBatteryOFF	bool			
FIRE_annunCargoDEPRESS	bool		(freighter only)	
MCP_panelPowered	bool			
COMM_RadioPanelPowered	bool	0-2		
COMM_AudioControlPanelPowered	bool	0-2		
TCAS_ATC_panelPowered	bool			
FIRE_HandleIllumination	bool	0-4	Index 0-3: engines, 4: APU	
WheelChocksSet	bool			
Unmatched events				
				EVT_DSP_AIR_SWITCH
				EVT_DATA_LINK_ACPT_L
				EVT_DATA_LINK_CANC_L
				EVT_DATA_LINK_RJCT_L
				EVT_DATA_LINK_ACPT_R
				EVT_DATA_LINK_CANC_R
				EVT_DATA_LINK_RJCT_R
				EVT_PVD_DIMMER_L
				EVT_PVD_ON_OFF_L
				EVT_PVD_DIMMER_R
				EVT_PVD_ON_OFF_R
				EVT_CHART_LIGHT_L
				EVT_CHART_LIGHT_R
				EVT_MAP_LIGHT_L_ROTATE
				EVT_MAP_LIGHT_R_ROTATE
				EVT_MAP_LIGHT_L_PULL_PUSH
				EVT_MAP_LIGHT_R_PULL_PUSH
				EVT_MAP_LIGHT_OBS_ROTATE
				EVT_MAP_LIGHT_OBS_PULL_PUSH
				EVT_OBSERVER_PANEL_LIGHT
				EVT_SPOT_LIGHT_1
				EVT_SPOT_LIGHT_2
				EVT_SPOT_LIGHT_3
				EVT_RMI_LEFT_SELECTOR
				EVT_RMI_RIGHT_SELECTOR
				EVT_OXY_TEST_RESET_SWITCH_L
				EVT_OXY_TEST_RESET_SWITCH_R
				EVT_OXY_EMER_TEST_L

Left out are the events targeted at typing things into the CDUs and EFB, as well as any other knobs and buttons they have.

Controlling the 747 Queen Of The Skies II

With the list above you have all you need to fill in the “Variable” sections in the Honeycomb configurator, apart from *what* to send as value. Basically, you have two choices: You can send mouse events, or data events. The PMDG code determines which it is by looking at the value; if the value is 8192 or higher, you are sending mouse events, lower values are interpreted as data events.

Using Mouse Events

The PMDG 747 QOTSII SDK defines mouse events: using single bits of a 32-bit value. As said, the lowest value is 8192 decimal or 0x00002000 hexadecimal, which is the 13th bit. The Honeycomb configurator only recognizes decimal values, so you will have to use the table below to determine what to enter:

Bit	Value (dec)	Value (hex)	Meaning
13	8192	0x00002000	Scroll the mouse wheel downwards, one click.
14	16384	0x00004000	Scroll the mouse wheel upwards, one click.
15	32768	0x00008000	“Wheel skip” (I am unsure what this means.)
16	65536	0x00010000	Invert the direction of the mouse wheel.
17	131072	0x00020000	Release the left button.
18	262144	0x00040000	Release the middle button.
19	524288	0x00080000	Release the right button.
21	2097152	0x00200000	“Down repeat” (I am unsure what this means.)
22	4194304	0x00400000	“Move” (Useless without mouse location)
23	8388608	0x00800000	Drag with left button (Useless without mouse location)
24	16777216	0x01000000	Drag with middle button (Useless without mouse location)
25	33554432	0x02000000	Drag with right button (Useless without mouse location)
26	67108864	0x04000000	Double-click with left button
27	134217728	0x08000000	Double-click with middle button
28	268435456	0x10000000	Double-click with right button
29	536870912	0x20000000	Click with left button
30	1073741824	0x40000000	Click with middle button
31	2147483648	0x80000000	Click with right button

So, sending type “L”, name “EVT_OH_LIGHTS_LANDING_OUTBD_L”, and value 536870912, will do the same as clicking with the left mouse button on the switch for the left-outboard landing light.

Using Data Events

The Honeycomb Alpha Yoke has a rotary knob for the magnetos on GA aircraft. On the 747 we do not have those, but the 747 QOTSII profile provided by Aerosoft has a nice alternative use: the autobrake knob. In the L-var table you can find “BRAKES_AutobrakeSelector”, where it states that the values are 0 for “RTO” (Rejected Take-Off), and 1 to 5 for the individual settings. (“OFF”, “DISARM”, 1, 2, and “MAX AUTO”) So the profile sends type “L”, name “AS_PMDG_747QOTSII_EVT_ABS_AUTOBRAKE_SELECTOR”, value 1 for the “OFF” settings, 2 for “R”, 3 for “L”, 4 for “BOTH”, and 5 for “START”. You could let it send mouse clicks, but that would only make the knob change once. This way you have a guaranteed setting that corresponds with the knob on the Alpha.

Further examples

The Aerosoft provided profiles maps the light knobs like they work on the overhead, so instead of up-for-on, they are down-for-on. Also, as mentioned before, the landing lights are switched on or off using the guard, which is not available on all models. The 747 QOTSII SDK provides “EVT_LDG_LIGHTS_TOGGLE”, but then you do not know for sure which way they flip. You can solve this by using a “Condition”:

- In the configurator, select the landing lights switch, and click the “Land Light ON” button.
- Open the “Conditions” and “Variables” lists and delete any entries currently there.
- Set “Condition” to “CUSTOM VAR”, key “L”, name “AS_PMDG_747QOTSII_LTS_LandingLights_Sw_ON”, index 0 (the “Outboard left” switch), unit “bool”, and value “0”. This will make the condition trigger on the outboard left landing light being off.
- Set the “Show / hide variable area” slider to the right, so it becomes blue.
- Set “Variable” to “CUSTOM VAR”, key “L”, name “AS_PMDG_747QOTSII_EVT_LDG_LIGHTS_TOGGLE”, no value.

With this you should now have a single entry to toggle all landing lights on. Make a similar entry for “Land Light OFF”, but now testing on value 1, so it toggles only if the landing lights are currently on. Similar “toggle” events are available for the runway turnoff lights, the cockpit lights, and the logo lights.

Controlling the Honeycomb Bravo’s LEDs

The Bravo’s LEDs are subdivided into three groups: Autopilot switch backlighting, gear indicators, and the annunciator panel. If you want to change their programming, it can help to make a short list. In the tables below, the “name” column refers to the name in the “Select LED” drop-down list of the Honeycomb Configurator. In column two is listed the label on the button. Columns three and four show numbers that you may find when you look in the “BFC_Throttle_Config.json” file (or one of the saved profiles), as “ByteIndex” and “BitIndex”. This is done because the LEDs are controlled using 8-bit values (bytes) but should not be of concern to you. In the next column I list their meaning, as set in the default 747 QOTSII profile from the Aerosoft site, and finally what condition is used to test it.

How to Interpret the Condition

The condition is written as it follows:

<key> “:” <variable> “,” <unit> <comparison>

Or:

<key> “:” <variable> “,” <index> “,” <unit> <comparison>

In this notation, the “key” is typically the letter “L”, as we are mostly looking at the L-vars provided by the PMDG Translator component. After the colon comes the actual name of the L-var, **with the “AS_PMDG_747QOTSII” prefix included**. Next comes an optional index, such as in the tables below for the gear annunciators. The unit will normally be “number” or “bool”, and getting it right is important, because otherwise the correct value will not be found. In the JSON file you may see a space between the comma and the unit name. **This space will disappear in an upcoming update!** This means any profiles you have made need to be adjusted, because the Prepar3D plugin will no longer recognize the L-var references.

Lastly, the comparison is an operator from the list here, followed by a value. The comparison operators are:

- “=” or “==” for “equals”.
- “!=” for “not equal to”.
- “<” and “<=” for “less than” and “less or equal to”.
- “>” and “>=” for “greater than” and “greater or equal to”.

If you add more than one condition, you must select a “Condition-Link”, which chooses “AND” or “OR” to be between those conditions. Choose once per LED, it will hold for all conditions.

The Autopilot Switch Backlighting

The autopilot switches have white labels that can light up. I have read that there may be different labels on different production runs of the Bravo, but the order is strictly left-to-right.

Name	Label	Byte , Bit	Remarks	Programming	
FCU-HDG	HDG	1, 0	“Heading hold” mode is on.	L:AS_PMDG_747QOTSII_MCP_annunHDG_HOLD, bool =1	
FCU-NAV	NAV	1, 1	“Nav hold” mode is on, in the 747 profile “VOR LOC”.	L:AS_PMDG_747QOTSII_MCP_annunLOC, bool =1	
FCU-APR	APR	1, 2	“Approach” mode is on.	L:AS_PMDG_747QOTSII_MCP_annunAPP, bool =1	
FCU-REV	REV	1, 3	“Back course hold” is on, in the 747 profile “LNAV”.	L:AS_PMDG_747QOTSII_MCP_annunLNAV, bool =1	
FCU-ALT	ALT	1, 4	“Altitude hold” mode is on, in the 747 profile “VNAV”.	L:AS_PMDG_747QOTSII_MCP_annunVNAV, bool =1	
FCU-VS	VS	1, 5	“Vertical speed” mode is on.	L:AS_PMDG_747QOTSII_MCP_annunVS, bool =1	
FCU-IAS	IAS	1, 6	“IAS hold” or “Speed hold” mode is on.	L:AS_PMDG_747QOTSII_MCP_annunSPD, bool =1	
GCU-AP	AUTO PILOT	1, 7	Autopilot is switched on. For the 747 profile this means any of the three switches is highlighted.	L:AS_PMDG_747QOTSII_MCP_annunAP, 0, bool =1 L:AS_PMDG_747QOTSII_MCP_annunAP, 1, bool =1 L:AS_PMDG_747QOTSII_MCP_annunAP, 2, bool =1	OR

Landing Gear Status Lights

The landing gear lights can be red or green, or switched off. Note that the 747 profile uses the generic conditions, which, in the configurator, do not need a key or unit to be set.

Name	Byte, Bit	Remarks	Programming	
LDG-L_GREEN	2, 0	Left main gear is UP.	GEAR LEFT POSITION =1	
LDG-L_RED	2, 1	Left main gear is IN TRANSIT.	GEAR LEFT POSITION >0 GEAR LEFT POSITION <1	AND
LDG-N_GREEN	2, 2	Nose wheel is UP.	GEAR CENTER POSITION =1	
LDG-N_RED	2, 3	Nose is IN TRANSIT.	GEAR CENTER POSITION >0 GEAR CENTER POSITION <1	AND
LDG-R_GREEN	2, 4	Right main gear is UP.	GEAR RIGHT POSITION =1	
LDG-R_RED	2, 5	Right main gear is IN TRANSIT.	GEAR RIGHT POSITION >0 GEAR RIGHT POSITION <1	AND

The Annunciator Panel

The annunciator panel uses partial phrases, which sometimes makes it a bit more difficult to find a matching condition if your aircraft does not really provide easy measurements for it, but overall, it works quite well. The lights are numbered left to right, first all the top-row ones, then the bottom row. Most of these are in the profile programmed to follow the annunciators in the big button on the glareshield, which contains several lights, so they are almost all interpreted as warnings.

Name	Label	Byte, Bit	Remarks	Programming	
ANC-MSTR_WARNG	MASTER WARNING	2, 6	Either of the master warning lights is on	L:AS_PMDG_747QOTSII_WARN_annunMASTER_WARNING, 0, bool =1 L:AS_PMDG_747QOTSII_WARN_annunMASTER_WARNING, 1, bool =1	OR
ANC-ENG_FIRE	ENGINE FIRE	2, 7	Any of the fire handles are illuminated. (4 engines + APU)	L: AS_PMDG_747QOTSII_FIRE_HandleIllumination, 0, bool =1 L: AS_PMDG_747QOTSII_FIRE_HandleIllumination, 1, bool =1 L: AS_PMDG_747QOTSII_FIRE_HandleIllumination, 2, bool =1 L: AS_PMDG_747QOTSII_FIRE_HandleIllumination, 3, bool =1 L: AS_PMDG_747QOTSII_FIRE_HandleIllumination, 4, bool =1	OR
ANC-OIL	LOW OIL PRESSURE	3, 0	Unused		
ANC-FUEL	LOW FUEL PRESSURE	3, 1	The overhead fuel annunciators.	L:AS_PMDG_747QOTSII_FUEL_annunXFEED_VALVE, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunXFEED_VALVE, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunXFEED_VALVE, 2, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunXFEED_VALVE, 3, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainFwd, 0, bool =1	OR

				L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainFwd, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainFwd, 2, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainFwd, 3, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainAft, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainAft, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainAft, 2, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_MainAft, 3, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ovrdfwd, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ovrdfwd, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ovrdfwd, 2, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ovrdfwd, 3, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Stab, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Stab, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ctr, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_Ctr, 1, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_JettisonNozle, 0, bool =1 L:AS_PMDG_747QOTSII_FUEL_annunPRESS_JettisonNozle, 1, bool =1	
ANC-ANTI_ICE	ANTI ICE	3, 2	The anti-ice annunciators.	L:AS_PMDG_747QOTSII_ICE_annunWindowHeatINOP, 0, bool =1 L:AS_PMDG_747QOTSII_ICE_annunWindowHeatINOP, 1, bool =1 L:AS_PMDG_747QOTSII_ICE_annunEngAntilceVALVE, 0, bool =1 L:AS_PMDG_747QOTSII_ICE_annunEngAntilceVALVE, 1, bool =1 L:AS_PMDG_747QOTSII_ICE_annunEngAntilceVALVE, 2, bool =1 L:AS_PMDG_747QOTSII_ICE_annunEngAntilceVALVE, 3, bool =1 L:AS_PMDG_747QOTSII_ICE_annunWingAntilceVALVE, bool =1	OR
ANC-STARTER	STARTER ENGAGED	3, 3	One of the engines is starting up.	L:AS_PMDG_747QOTSII_ENG_Start_Light, 0, bool = 1 L:AS_PMDG_747QOTSII_ENG_Start_Light, 1, bool = 1 L:AS_PMDG_747QOTSII_ENG_Start_Light, 2, bool = 1 L:AS_PMDG_747QOTSII_ENG_Start_Light, 3, bool = 1	OR
ANC-APU	APU	3, 4		L:AS_PMDG_747QOTSII_FIRE_HandleIllumination, 4, bool=1	
ANC-MSTR_CTN	MASTER CAUTION	3, 5	Master caution annunciator.	L:AS_PMDG_747QOTSII_WARN_annunMASTER_CAUTION, 0, bool =1 L:AS_PMDG_747QOTSII_WARN_annunMASTER_CAUTION, 1, bool =1	OR
ANC-VACUUM	VACUUM	3, 6	The air conditioning annunciators.	L:AS_PMDG_747QOTSII_AIR_annun_SYS_FAULT, 0, bool =1 L:AS_PMDG_747QOTSII_AIR_annun_SYS_FAULT, 1, bool =1 L:AS_PMDG_747QOTSII_AIR_annun_SYS_FAULT, 2, bool =1 L:AS_PMDG_747QOTSII_AIR_annun_SYS_FAULT, 3, bool =1 L:AS_PMDG_747QOTSII_AIR_annunPack_SYS_FAIL, bool =1	OR

				L:AS_PMDG_747QOTSII_AIR_annunZone_SYS_FAIL, bool =1	
ANC-HYD	LOW HYD PRESSURE	3, 7	The hydraulics annunciators.	L:AS_PMDG_747QOTSII_HYD_annunSYS_FAULT, 0, bool L:AS_PMDG_747QOTSII_HYD_annunSYS_FAULT, 1, bool L:AS_PMDG_747QOTSII_HYD_annunSYS_FAULT, 2, bool L:AS_PMDG_747QOTSII_HYD_annunSYS_FAULT, 3, bool	OR
ANC-AUX-FUEL	AUX FUEL PUMP	4, 0	(unused)		
ANC-PRK_BRK	PARKING BRAKE	4, 1	The parking brakes are set.	L:AS_PMDG_747QOTSII_BRAKES_ParkingBrakeLeverOn, bool =1	
ANC-VOLTS	LOW VOLTS	4, 2	The electrical systems annunciator.	L:AS_PMDG_747QOTSII_ELEC_annunGenOFF, 0, bool =1 L:AS_PMDG_747QOTSII_ELEC_annunGenOFF, 1, bool =1 L:AS_PMDG_747QOTSII_ELEC_annunGenOFF, 2, bool =1 L:AS_PMDG_747QOTSII_ELEC_annunGenOFF, 3, bool =1 L:AS_PMDG_747QOTSII_ELEC_annunUtilOFF, 0, bool =1 L:AS_PMDG_747QOTSII_ELEC_annunUtilOFF, 1, bool =1	OR
ANC-DOOR	DOOR	4, 3	Any of the doors is not closed and armed.	L:AS_PMDG_747QOTSII_DOOR_state, 0, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 1, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 2, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 3, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 4, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 5, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 6, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 7, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 8, number !=2 L:AS_PMDG_747QOTSII_DOOR_state, 9, number !=2	OR

In addition, all LEDs are also OR- linked to L:AS_PMDG_747QOTSII_LTS_IndLightsTestSw,number = 0, so that you can switch on all mapped LEDs with the IND LTS switch in the TEST position.