# Variables and Conditions for the PMDG

## Introduction

To find out what is happening in the PMDG 777X, 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_777X_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 777X 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 737 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_777X_Data**" structure of the SDK. All names start with "**AS_PMDG_T7_**", so in this table that prefix is 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_T7_**". 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 field, "**ICE_WindowHeatBackUp_Sw_OFF**", refers to a set of switches with guards, and they refer to a left and right one, which is also specified in the event name. 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 – Backup Window Heat** | | | | |
| ICE_WindowHeatBackUp_Sw_OFF | bool | 0-1 | | EVT_OH_ICE_BU_WINDOW_HEAT_*L* (also R)<br>EVT_OH_ICE_BU_WINDOW_HEAT_*L*_GUARD |
| **Overhead Maintenance Panel – Standby Power** | | | | |
| ELEC_StandbyPowerSw | number | | 0: OFF, 1: AUTO, 2: BAT | EVT_OH_ELEC_STBY_PWR_SWITCH<br>EVT_OH_ELEC_STBY_PWR_GUARD |
| **Overhead Maintenance Panel – Flight Controls Hydraulic Valve Power** | | | | |
| FCTL_WingHydValve_Sw_SHUT_OFF | bool | 0-2 | NOTE: 0: L, 1: R, 2: C | EVT_OH_HYD_VLV_PWR_WING_*L* (also R and C)<br>EVT_OH_HYD_VLV_PWR_WING_*L*_GUARD |
| FCTL_TailHydValve_Sw_SHUT_OFF | bool | 0-2 | | EVT_OH_HYD_VLV_PWR_TAIL_*L* (also R and C)<br>EVT_OH_HYD_VLV_PWR_TAIL_*L*_GUARD |
| FCTL_annunTailHydVALVE_CLOSED | bool | 0-2 | | |
| FCTL_annunWingHydVALVE_CLOSED | bool | 0-2 | | |
| **Overhead Maintenance Panel – APU Maintenance** | | | | |
| APU_Power_Sw_TEST | bool | | | EVT_OH_APU_TEST_SWITCH<br>EVT_OH_APU_TEST_SWITCH_GUARD |
| **Overhead Maintenance Panel – EEC Maintenance** | | | | |
| ENG_EECPower_Sw_TEST | bool | 0-1 | | EVT_OH_EEC_TEST_*L*_SWITCH (also R)<br>EVT_OH_EEC_TEST_*L*_SWITCH_GUARD |

| | | | | |
|---|---|---|---|---|
| Overhead Maintenance Panel – Electrical | | | | |
| ELEC_TowingPower_Sw_BATT | bool | | | EVT_OH_ELEC_TOWING_PWR_SWITCH<br>EVT_OH_ELEC_TOWING_PWR_GUARD |
| ELEC_annunTowingPowerON_BATT | bool | | | |
| **Overhead Maintenance Panel – Cargo Temperature Selectors** | | | | |
| AIR_CargoTemp_Selector | number | 0-1 | 0: OFF, 1: LOW, 2: HIGH | EVT_OH_AIRCOND_TEMP_SELECTOR_CARGO_AFT<br>EVT_OH_AIRCOND_TEMP_SELECTOR_CARGO_BULK |
| **Overhead Panel – ADIRU** | | | | |
| ADIRU_Sw_On | bool | | | EVT_OH_ADIRU_SWITCH |
| ADIRU_annunOFF | bool | | | |
| ADIRU_annunON_BAT | bool | | | |
| **Overhead Panel – Flight Controls** | | | | |
| FCTL_ThrustAsymComp_Sw_AUTO | bool | | | |
| FCTL_annunThrustAsymCompOFF | bool | | | |
| **Overhead Panel – Electrical** | | | | |
| ELEC_CabUtilSw | bool | | | EVT_OH_ELEC_CAB_UTIL |
| ELEC_annunCabUtilOFF | bool | | | |
| ELEC_IFEPassSeatsSw | bool | | | EVT_OH_ELEC_IFE |
| ELEC_annunIFEPassSeatsOFF | bool | | | |
| ELEC_Battery_Sw_ON | bool | | | EVT_OH_ELEC_BATTERY_SWITCH |
| ELEC_APUGen_Sw_ON | bool | | | EVT_OH_ELEC_APU_GEN_SWITCH |
| ELEC_APU_Selector | number | | 0: OFF, 1: ON, 2: START | EVT_OH_ELEC_APU_SEL_SWITCH |
| ELEC_annunAPU_FAULT | bool | | | |
| ELEC_BusTie_Sw_AUTO | bool | 0-1 | | EVT_OH_ELEC_BUS_*TIE1*_SWITCH (also TIE2) |
| ELEC_annunBusTieISLN | bool | 0-1 | | |
| ELEC_ExtPwrSw | bool | 0-1 | | EVT_OH_ELEC_GRD_PWR_*PRIM*_SWITCH (also SEC) |
| ELEC_annunExtPowr_ON | bool | 0-1 | | |
| ELEC_annunExtPowr_AVAIL | bool | 0-1 | | |
| ELEC_Gen_Sw_ON | bool | 0-1 | | EVT_OH_ELEC_*GEN1*_SWITCH (also GEN2) |
| ELEC_annunGenOFF | bool | 0-1 | | |
| ELEC_BackupGen_Sw_ON | bool | 0-1 | | EVT_OH_ELEC_BACKUP_*GEN1*_SWITCH (also GEN2) |
| ELEC_annunBackupGenOFF | bool | 0-1 | | |
| ELEC_IDGDiscSw | bool | 0-1 | | EVT_OH_ELEC_DISCONNECT*1*_SWITCH (also 2) |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_OH_ELEC_DISCONNECT*1*_GUARD |
| ELEC_annunIDGDiscDRIVE | bool | 0-1 | | |
| | | | | EVT_OH_ELEC_GND_TEST_SWITCH<br>EVT_OH_ELEC_GND_TEST_GUARD |
| **Overhead Panel – Wipers** | | | | |
| WIPERS_Selector | number | 0-1 | 0: OFF, 1: INT, 2: LOW, 3: HIGH | EVT_OH_WIPER_LEFT_*SWITCH* (also RIGHT) |
| Overhead Panel – Emergency Lights | | | | |
| LTS_EmerLightsSelector | number | | 0: OFF, 1: ARMED, 2: ON | EVT_OH_EMER_EXIT_LIGHT_SWITCH<br>EVT_OH_EMER_EXIT_LIGHT_GUARD |
| **Overhead Panel – Service Interphone** | | | | |
| COMM_ServiceInterphoneSw | bool | | | EVT_OH_SERVICE_INTERPHONE_SWITCH |
| **Overhead Panel – Passenger Oxygen** | | | | |
| OXY_PassOxygen_Sw_On | bool | | | EVT_OH_OXY_PASS_SWITCH<br>EVT_OH_OXY_PASS_GUARD |
| OXY_annunPassOxygenON | bool | | | |
| | | | | EVT_OH_OXY_SUPRNMRY_SWITCH<br>EVT_OH_OXY_SUPRNMRY_GUARD |
| | | | | EVT_OXY_TEST_RESET_SWITCH_*L* (also R)<br>EVT_OXY_TEST_RESET_SWITCH_OBS_*L*<br>EVT_OXY_NORM_EMER_*L*<br>EVT_OXY_NORM_EMER_OBS_*L* |
| **Overhead Panel – Window Heat** | | | | |
| ICE_WindowHeat_Sw_ON | bool | 0-3 | (left, left-fwd, right-fwd, right) | EVT_OH_ICE_WINDOW_HEAT_*1* (also 2, 3, and 4) |
| ICE_annunWindowHeatINOP | bool | 0-3 | (left, left-fwd, right-fwd, right) | |
| **Overhead Panel – Hydraulics** | | | | |
| HYD_RamAirTurbineSw | bool | | | EVT_OH_HYD_RAM_AIR<br>EVT_OH_HYD_RAM_AIR_COVER |
| HYD_annunRamAirTurbinePRESS | bool | | | |
| HYD_annunRamAirTurbineUNLKD | bool | | | |
| HYD_PrimaryEngPump_Sw_ON | bool | 0-1 | | EVT_OH_HYD_*ENG1* (also ENG2) |
| HYD_PrimaryElecPump_Sw_ON | bool | 0-1 | | EVT_OH_HYD_*ELEC1* (also ELEC2) |
| HYD_DemandElecPump_Selector | number | 0-1 | 0: OFF, 1: AUTO, 2: ON | EVT_OH_HYD_DEMAND_*ELEC1* (also ELEC2) |
| HYD_DemandAirPump_Selector | numbe | 0-1 | 0: OFF, 1: AUTO, 2: ON | EVT_OH_HYD_*AIR1* (also AIR2) |

| | | | | |
|---|---|---|---|---|
| | r | | | |
| HYD_annunPrimaryEngPumpFAULT | bool | 0-1 | | |
| HYD_annunPrimaryElecPumpFAULT | bool | 0-1 | | |
| HYD_annunDemandElecPumpFAULT | bool | 0-1 | | |
| HYD_annunDemandAirPumpFAULT | bool | 0-1 | | |
| **Overhead Panel – Passenger Signs** | | | | |
| SIGNS_NoSmokingSelector | numbe r | | 0: OFF, 1: AUTO, 2: ON | EVT_OH_NO_SMOKING_LIGHT_SWITCH |
| SIGNS_SeatBeltsSelector | numbe r | | 0: OFF, 1: AUTO, 2: ON | EVT_OH_FASTEN_BELTS_LIGHT_SWITCH |
| **Overhead Panel – Flight Deck Lights** | | | | |
| LTS_DomeLightKnob | numbe r | | 0-150 | EVT_OH_DOME_SWITCH |
| LTS_CircuitBreakerKnob | numbe r | | 0-150 | EVT_OH_CB_LIGHT_CONTROL |
| LTS_OvereadPanelKnob | numbe r | | 0-150 | EVT_OH_PANEL_LIGHT_CONTROL |
| LTS_GlareshieldPNLlKnob | numbe r | | 0-150 | EVT_OH_GS_PANEL_LIGHT_CONTROL |
| LTS_GlareshieldFLOODKnob | numbe r | | 0-150 | EVT_OH_GS_FLOOD_LIGHT_CONTROL |
| LTS_Storm_Sw_ON | bool | | | EVT_OH_LIGHTS_STORM |
| LTS_MasterBright_Sw_ON | bool | | | EVT_OH_MASTER_BRIGHT_PUSH |
| LTS_MasterBrigntKnob | numbe r | | 0-150 | EVT_OH_MASTER_BRIGHT_ROTATE |
| LTS_IndLightsTestSw | numbe r | | 0: TEST, 1: BRT, 2: DIM | EVT_OH_LIGHTS_IND_LTS_SWITCH |
| | | | | EVT_COCKPIT_LIGHTS_TOGGLE<br>EVT_PANEL_LIGHTS_TOGGLE<br>EVT_FLOOD_LIGHTS_TOGGLE |
| **Overhead Panel – External Lights** | | | | |
| LTS_LandingLights_Sw_ON | bool | 0-2 | NOTE: order left, right, nose | EVT_OH_LIGHTS_LANDING_L<br>EVT_OH_LIGHTS_LANDING_R<br>EVT_OH_LIGHTS_LANDING_NOSE<br>EVT_LDG_LIGHTS_TOGGLE (all together) |
| LTS_Beacon_Sw_ON | bool | | | 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<br>EVT_OH_LIGHTS_R_TURNOFF<br>EVT_TURNOFF_LIGHTS_TOGGLE (both together) |
| LTS_Taxi_Sw_ON | bool | | | EVT_OH_LIGHTS_TAXI |
| LTS_Strobe_Sw_ON | bool | | | EVT_OH_LIGHTS_STROBE |
| **Overhead Panel – APU and Cargo Fire** | | | | |
| FIRE_CargoFire_Sw_Arm | bool | 0-1 | | EVT_OH_FIRE_CARGO_ARM_*FWD* (also AFT) |
| FIRE_annunCargoFire | bool | 0-1 | | |
| FIRE_CargoFireDisch_Sw | bool | | | EVT_OH_FIRE_CARGO_DISCH<br>EVT_OH_FIRE_CARGO_DISCH_GUARD |
| FIRE_annunCargoDISCH | bool | | | |
| FIRE_FireOvhtTest_Sw | bool | | | EVT_OH_FIRE_OVHT_TEST |
| FIRE_APUHandle | number | | 0: In/normal, 1: Pulled out<br>2: turned left, 2: turned right | EVT_OH_FIRE_HANDLE_APU_TOP<br>EVT_OH_FIRE_HANDLE_APU_BOTTOM |
| FIRE_APUHandleUnlock_Sw | bool | | | EVT_OH_FIRE_UNLOCK_SWITCH_APU |
| FIRE_annunAPU_BTL_DISCH | bool | | | |
| | | | | EVT_OH_FIRE_CARGO_ARM_MAIN_DECK<br>EVT_OH_FIRE_CARGO_DISCH_DEPR |
| **Overhead Panel – Engines** | | | | |
| ENG_EECMode_Sw_NORM | bool | 0-1 | | EVT_OH_EEC_*L*_SWITCH (also R)<br>EVT_OH_EEC_*L*_GUARD |
| ENG_Start_Selector | number | 0-1 | 0: START, 1: NORM | EVT_OH_ENGINE_*L*_START (also R) |
| ENG_Autostart_Sw_ON | bool | | | EVT_OH_ENGINE_AUTOSTART |
| ENG_annunALTN | bool | 0-1 | | |
| ENG_annunAutostartOFF | bool | | | |
| | | | | EVT_OH_THRUST_ASYM_COMP |
| **Overhead Panel – Fuel** | | | | |
| FUEL_CrossFeedFwd_Sw | bool | | | EVT_OH_FUEL_CROSSFEED_FORWARD |
| FUEL_CrossFeedAft_Sw | bool | | | EVT_OH_FUEL_CROSSFEED_AFT |
| FUEL_PumpFwd_Sw | bool | 0-1 | | EVT_OH_FUEL_PUMP_*1*_FORWARD (also 2) |
| FUEL_PumpAft_Sw | bool | 0-1 | | EVT_OH_FUEL_PUMP_*1*_AFT (also 2) |

| | | | | |
|---|---|---|---|---|
| FUEL_PumpCtr_Sw | bool | 0-1 | | EVT_OH_FUEL_PUMP_*L*_CENTER (also R) |
| FUEL_JettisonNozle_Sw | bool | 0-1 | | EVT_OH_FUEL_JETTISON_NOZZLE_*L* (also R)<br>EVT_OH_FUEL_JETTISON_NOZZLE_*L*_GUARD |
| FUEL_JettisonArm_Sw | bool | | | EVT_OH_FUEL_JETTISON_ARM |
| FUEL_FuelToRemain_Sw_Pulled | bool | | | EVT_OH_FUEL_TO_REMAIN_PULL |
| FUEL_FuelToRemain_Selector | number | | | EVT_OH_FUEL_TO_REMAIN_ROTATE |
| FUEL_annunFwdXFEED_VALVE | bool | | | |
| FUEL_annunAftXFEED_VALVE | bool | | | |
| FUEL_annunLOWPRESS_Fwd | bool | 0-1 | | |
| FUEL_annunLOWPRESS_Aft | bool | 0-1 | | |
| FUEL_annunLOWPRESS_Ctr | bool | 0-1 | | |
| FUEL_annunJettisonNozleVALVE | bool | 0-1 | | |
| FUEL_annunArmFAULT | bool | | | |
| | | | | EVT_OH_FUEL_PUMP_AUX |
| **Overhead Panel – Anti-Ice** | | | | |
| ICE_WingAntiIceSw | number | | 0: OFF, 1: AUTO, 2: ON | EVT_OH_ICE_WING_ANTIICE |
| ICE_EngAntiIceSw | number | 0-1 | 0: OFF, 1: AUTO, 2: ON | EVT_OH_ICE_ENGINE_ANTIICE_*1* (also 2) |
| **Overhead Panel – Air Conditioning** | | | | |
| AIR_Pack_Sw_AUTO | bool | 0-1 | | EVT_OH_AIRCOND_PACK_SWITCH_*L* (also R) |
| AIR_TrimAir_Sw_On | bool | 0-1 | | EVT_OH_AIRCOND_TRIM_AIR_SWITCH_*L* (also R) |
| AIR_RecircFan_Sw_On | bool | 0-1 | | EVT_OH_AIRCOND_RECIRC_FAN_*UPP*_SWITCH (also LWR) |
| AIR_TempSelector | number | 0-1 | 0: C .. 60: W<br>Flight deck only: …70 MAN | EVT_OH_AIRCOND_TEMP_SELECTOR_FLT_DECK<br>EVT_OH_AIRCOND_TEMP_SELECTOR_CABIN |
| AIR_AirCondReset_Sw_Pushed | bool | | | EVT_OH_AIRCOND_RESET_SWITCH |
| AIR_EquipCooling_Sw_AUTO | bool | | | EVT_OH_AIRCOND_EQUIP_COOLING_SWITCH |
| AIR_Gasper_Sw_On | bool | | | EVT_OH_AIRCOND_GASPER_SWITCH |
| AIR_annunPackOFF | bool | 0-1 | | |
| AIR_annunTrimAirFAULT | bool | 0-1 | | |
| AIR_annunEquipCoolingOVRD | bool | | | |
| | | | | EVT_OH_AIRCOND_RECIRC_FANS_SWITCH<br>EVT_OH_AIRCOND_TEMP_SELECTOR_LWR_CARGO_FWD<br>EVT_OH_AIRCOND_TEMP_SELECTOR_LWR_CARGO_AFT |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_OH_AIRCOND_TEMP_SELECTOR_MAIN_CARGO_FWD<br>EVT_OH_AIRCOND_TEMP_SELECTOR_MAIN_CARGO_AFT<br>EVT_OH_AIRCOND_MAIN_DECK_FLOW_SWITCH<br>EVT_OH_AIRCOND_ALT_VENT_SWITCH<br>EVT_OH_AIRCOND_ALT_VENT_GUARD |
| **Overhead Panel – Bleed Air** | | | | |
| AIR_EngBleedAir_Sw_AUTO | bool | 0-1 | | EVT_OH_BLEED_ENG_*1*_SWITCH (also 2) |
| AIR_APUBleedAir_Sw_AUTO | bool | | | EVT_OH_BLEED_APU_SWITCH |
| AIR_IsolationValve_Sw | bool | 0-1 | | EVT_OH_BLEED_ISOLATION_VALVE_SWITCH_*L* (also R) |
| AIR_CtrIsolationValve_Sw | bool | | | EVT_OH_BLEED_ISOLATION_VALVE_SWITCH_C |
| AIR_annunEngBleedAirOFF | bool | 0-1 | | |
| AIR_annunAPUBleedAirOFF | bool | | | |
| AIR_annunIsolationValveCLOSED | bool | 0-1 | | |
| AIR_annunCtrIsolationValveCLOSED | bool | | | |
| **Overhead Panel – Pressurization** | | | | |
| AIR_OutflowValve_Sw_AUTO | bool | 0-1 | | EVT_OH_PRESS_VALVE_SWITCH_*1* (also 2) |
| AIR_OutflowValveManual_Selector | number | 0-1 | 0: OPEN, 1: neutral, 2: CLOSE | EVT_OH_PRESS_VALVE_SWITCH_MANUAL_*1* (also 2) |
| AIR_LdgAlt_Sw_Pulled | bool | | | EVT_OH_PRESS_LAND_ALT_KNOB_PULL |
| AIR_LdgAlt_Selector | number | | 0: DECR, 1: neutral, 2: INCR | EVT_OH_PRESS_LAND_ALT_KNOB_ROTATE |
| AIR_annunOutflowValve_MAN | bool | 0-1 | | |
| **Overhead Panel – Miscellaneous** | | | | |
| | | | | EVT_OH_CAMERA_LTS_SWITCH<br>EVT_OH_PRIM_FLT_COMPUTERS<br>EVT_OH_PRIM_FLT_COMPUTERS_GUARD<br>EVT_OH_CVR_TEST<br>EVT_OH_CVR_ERASE |
| **Forward Panel – Center** | | | | |
| GEAR_Lever | number | | 0: UP, 1: DOWN | EVT_GEAR_LEVER |
| GEAR_LockOvrd_Sw | bool | | | EVT_GEAR_LEVER_UNLOCK |
| GEAR_AltnGear_Sw_DOWN | bool | | | EVT_GEAR_ALTN_GEAR_DOWN<br>EVT_GEAR_ALTN_GEAR_DOWN_GUARD |
| GPWS_FlapInhibitSw_OVRD | bool | | | EVT_GPWS_FLAP_OVRD_SWITCH |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_GPWS_FLAP_OVRD_GUARD |
| GPWS_GearInhibitSw_OVRD | bool | | | EVT_GPWS_GEAR_OVRD_SWITCH<br>EVT_GPWS_GEAR_OVRD_GUARD |
| GPWS_TerrInhibitSw_OVRD | bool | | | EVT_GPWS_TERR_OVRD_SWITCH<br>EVT_GPWS_TERR_OVRD_GUARD |
| GPWS_GSInhibit_Sw | bool | | | EVT_GPWS_GS_INHIBIT_SWITCH |
| GPWS_annunGND_PROX_top | bool | | | |
| GPWS_annunGND_PROX_bottom | bool | | | |
| BRAKES_AutobrakeSelector | number | | 0: RTO, 1: OFF, 2: DISARM<br>3: “1” … 5: MAX AUTO | EVT_ABS_AUTOBRAKE_SELECTOR |
| | | | | EVT_GPWS_RWY_OVRD_SWITCH<br>EVT_GPWS_RWY_OVRD_GUARD |
| **Forward Panel – ISFD** | | | | |
| ISFD_Baro_Sw_Pushed | bool | | | 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_ISFD_BARO |
| | | | | EVT_STANDBY_ASI_KNOB<br>EVT_STANDBY_ASI_KNOB_PUSH<br>EVT_STANDBY_ALTIMETER_KNOB<br>EVT_STANDBY_ALTIMETER_KNOB_PUSH |
| **Forward Panel – Left** | | | | |
| ISP_Nav_L_Sw_CDU | bool | | | EVT_FWD_NAV_SOURCE_L |
| ISP_DsplCtrl_L_Sw_Altn | bool | | | EVT_FWD_DSPL_CTRL_SOURCE_L |
| ISP_AirDataAtt_L_Sw_Altn | bool | | | EVT_FWD_AIR_DATA_ATT_SOURCE_L |
| DSP_InbdDspl_L_Selector | number | | 0: ND, 1: NAV, 2: MFD, 3:EICAS | EVT_DSP_INDB_DSPL_L |
| EFIS_HdgRef_Sw_Norm | bool | | | EVT_EFIS_HDG_REF_SWITCH<br>EVT_EFIS_HDG_REF_GUARD |
| EFIS_annunHdgRefTRUE | bool | | | |
| BRAKES_BrakePressNeedle | number | | 0-100 (0-4000 PSI) | |

| | | | | |
|---|---|---|---|---|
| BRAKES_annunBRAKE_SOURCE | bool | | | |
| **Forward Panel – Right** | | | | |
| ISP_Nav_R_Sw_CDU | bool | | | EVT_FWD_NAV_SOURCE_R |
| ISP_DsplCtrl_R_Sw_Altn | bool | | | EVT_FWD_DSPL_CTRL_SOURCE_R |
| ISP_AirDataAtt_R_Sw_Altn | bool | | | EVT_FWD_AIR_DATA_ATT_SOURCE_R |
| ISP_FMC_Selector | number | | 0: LEFT, 1: AUTO, 2: RIGHT | EVT_FWD_FMC_SELECTOR |
| DSP_InbdDspl_R_Selector | number | | 0: EICAS, 1: MFD, 2: ND, 3: PFD | EVT_DSP_INDB_DSPL_R |
| **Left- & Right-Sidewalls** | | | | |
| AIR_ShoulderHeaterKnob | number | 0-1 | 0-150 | EVT_FWD_*LEFT*_SHOULDER_HEATER (also RIGHT) |
| AIR_FootHeaterSelector | number | 0-1 | 0: OFF, 1: LOW, 2: HIGH | EVT_FWD_*LEFT*_FOOT_HEATER |
| LTS_LeftFwdPanelPNLKnob | number | | 0-150 | EVT_FWD_LEFT_PANEL_LIGHT_CONTROL |
| LTS_LeftFwdPanelFLOODKnob | number | | 0-150 | EVT_FWD_LEFT_FLOOD_LIGHT_CONTROL |
| LTS_LeftOutbdDsplBRIGHTNESSKnob | number | | 0-150 | EVT_FWD_LEFT_OUTBD_BRIGHT_CONTROL |
| LTS_LeftInbdDsplBRIGHTNESSKnob | number | | 0-150 | EVT_FWD_LEFT_INBD_BRIGHT_CONTROL |
| | | | | EVT_FWD_LEFT_INBD_TERR_BRIGHT_CONTROL |
| LTS_RightFwdPanelPNLKnob | number | | 0-150 | EVT_FWD_RIGHT_PANEL_LIGHT_CONTROL |
| LTS_RightFwdPanelFLOODKnob | number | | 0-150 | EVT_FWD_RIGHT_FLOOD_LIGHT_CONTROL |
| LTS_RightInbdDsplBRIGHTNESSKnob | number | | 0-150 | EVT_FWD_RIGHT_INBD_BRIGHT_CONTROL |
| LTS_RightOutbdDsplBRIGHTNESSKnob | number | | 0-150 | EVT_FWD_RIGHT_OUTBD_BRIGHT_CONTROL |
| | | | | EVT_FWD_RIGHT_INBD_TERR_BRIGHT_CONTROL |
| **Chronometers** | | | | |
| CHR_Chr_Sw_Pushed | bool | 0-1 | | EVT_CHRONO_*L*_CHR (also R) |
| CHR_TimeDate_Sw_Pushed | bool | 0-1 | | EVT_CHRONO_*L*_TIME_DATE_PUSH |
| CHR_TimeDate_Selector | numbe | 0-1 | 0: UTC, 1: MAN | EVT_CHRONO_*L*_TIME_DATE_SELECT |

| | | | | |
|---|---|---|---|---|
| | r | | | |
| CHR_Set_Selector | numbe r | 0-1 | 0: RUN, 1: HLDY, 2: MM, 3: HD | EVT_CHRONO_*L*_SET |
| CHR_ET_Selector | numbe r | 0-1 | 0: RESET, 1: HLD, 2: RUN | EVT_CHRONO_*L*_ET |
| **Glareshield – EFIS** | | | | |
| EFIS_MinsSelBARO | bool | 0-1 | | EVT_EFIS_*CPT*_MINIMUMS_RADIO_BARO (also FO) |
| EFIS_BaroSelHPA | bool | 0-1 | | EVT_EFIS_*CPT*_BARO_IN_HPA |
| EFIS_VORADFSel1 | numbe r | 0-1 | 0: VOR, 1: OFF, 2: ADF | EVT_EFIS_*CPT*_VOR_ADF_SELECTOR_L |
| EFIS_VORADFSel2 | numbe r | 0-1 | 0: VOR, 1: OFF, 2: ADF | EVT_EFIS_*CPT*_VOR_ADF_SELECTOR_R |
| EFIS_ModeSel | numbe r | 0-1 | 0: APP, 1: VOR, 2: MAP, 3:PLAN | EVT_EFIS_*CPT*_MODE |
| EFIS_RangeSel | numbe r | 0-1 | 0: 10 … 6: 640 | EVT_EFIS_*CPT*_RANGE |
| EFIS_MinsKnob | numbe r | 0-1 | 0-99 | EVT_EFIS_*CPT*_MINIMUMS |
| EFIS_BaroKnob | numbe r | 0-1 | 0-99 | EVT_EFIS_*CPT*_BARO |
| EFIS_MinsRST_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_MINIMUMS_RST |
| EFIS_BaroSTD_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_BARO_STD |
| EFIS_ModeCTR_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_MODE_CTR |
| EFIS_RangeTFC_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_RANGE_TFC |
| EFIS_WXR_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_WXR |
| EFIS_STA_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_STA |
| EFIS_WPT_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_WPT |
| EFIS_ARPT_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_ARPT |
| EFIS_DATA_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_DATA |
| EFIS_POS_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_POS |
| EFIS_TERR_Sw_Pushed | bool | 0-1 | | EVT_EFIS_*CPT*_TERR |
| | | | | EVT_EFIS_*CPT*_FPV<br>EVT_EFIS_*CPT*_MTRS |
| **Glareshield – MCP** | | | | |
| MCP_IASMach | numbe | | Mach if < 10.0 | |

| | | | | |
|---|---|---|---|---|
| | r | | | |
| MCP_IASBlank | bool | | | |
| MCP_Heading | numbe<br>r | | | |
| MCP_Altitude | numbe<br>r | | | |
| MCP_VertSpeed | numbe<br>r | | | |
| MCP_FPA | numbe<br>r | | | |
| MCP_VertSpeedBlank | bool | | | |
| MCP_FD_Sw_On | bool | 0-1 | | EVT_MCP_FD_SWITCH_*L* (also R) |
| MCP_ATArm_Sw_On | bool | 0-1 | | EVT_MCP_AT_ARM_SWITCH_*L* (also R) |
| MCP_BankLimitSel | numbe<br>r | | 0: AUTO, 1: 5 … 5: 25 | EVT_MCP_BANK_ANGLE_SELECTOR |
| MCP_AltIncrSel | bool | | False: AUTO, True: 1000 | EVT_MCP_ALT_INCR_SELECTOR |
| MCP_DisengageBar | bool | | | EVT_MCP_DISENGAGE_BAR |
| MCP_Speed_Dial | numbe<br>r | | 0-99 | EVT_MCP_SPEED_SELECTOR<br>EVT_MCP_IAS_SET<br>EVT_MCP_MACH_SET |
| MCP_Heading_Dial | numbe<br>r | | 0-99 | EVT_MCP_HEADING_SELECTOR<br>EVT_MCP_HDGTRK_SET |
| MCP_Altitude_Dial | numbe<br>r | | 0-99 | EVT_MCP_ALTITUDE_SELECTOR<br>EVT_MCP_ALT_SET |
| MCP_VS_Wheel | numbe<br>r | | 0-99 | EVT_MCP_VS_SELECTOR<br>EVT_MCP_VS_SET<br>EVT_MCP_FPA_SET |
| MCP_HDGDial_Mode | numbe<br>r | | 0: HDG, 1: TRK | |
| MCP_VSDial_Mode | numbe<br>r | | 0: VS, 1: FPA | |
| MCP_AP_Sw_Pushed | bool | 0-1 | | EVT_MCP_AP_*L*_SWITCH (also R) |
| MCP_CLB_CON_Sw_Pushed | bool | | | EVT_MCP_CLB_CON_SWITCH |
| MCP_AT_Sw_Pushed | bool | | | EVT_MCP_AT_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_FPA_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_Speeed_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_HDG_TRK_Toggle_Sw_Pushed | bool | | | EVT_MCP_HDG_TRK_SWITCH |
| MCP_VS_FPA_Toggle_Sw_Pushed | bool | | | EVT_MCP_VS_FPA_SWITCH |
| MCP_annunAP | bool | 0-1 | | |
| MCP_annunAT | bool | | | |
| MCP_annunLNAV | bool | | | |
| MCP_annunVNAV | bool | | | |
| MCP_annunFLCH | bool | | | |
| MCP_annunHDG_HOLD | bool | | | |
| MCP_annunVS_FPA | bool | | | |
| MCP_annunALT_HOLD | bool | | | |
| MCP_annunLOC | bool | | | |
| MCP_annunAPP | bool | | | |
| | | | | EVT_MCP_TOGA_SCREW_L<br>EVT_MCP_TOGA_SCREW_R<br>EVT_AT_ARM_SWITCHES |
| **Glareshield – Display Select Panel** | | | | |
| DSP_L_INBD_Sw | bool | | | EVT_DSP_L_INBD_SWITCH |
| DSP_R_INBD_Sw | bool | | | EVT_DSP_R_INBD_SWITCH |
| DSP_LWR_CTR_Sw | bool | | | 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_AIR_Sw | bool | | | EVT_DSP_AIR_SWITCH |
| DSP_DOOR_Sw | bool | | | EVT_DSP_DOOR_SWITCH |
| DSP_GEAR_Sw | bool | | | EVT_DSP_GEAR_SWITCH |
| DSP_FCTL_Sw | bool | | | EVT_DSP_FCTL_SWITCH |
| DSP_CAM_Sw | bool | | | EVT_DSP_CAM_SWITCH |
| DSP_CHKL_Sw | bool | | | EVT_DSP_CHKL_SWITCH |
| DSP_COMM_Sw | bool | | | EVT_DSP_COMM_SWITCH |
| DSP_NAV_Sw | bool | | | EVT_DSP_NAV_SWITCH |
| DSP_CANC_RCL_Sw | bool | | | EVT_DSP_CANC_RCL_SWITCH |
| DSP_annunL_INBD | bool | | | |
| DSP_annunR_INBD | bool | | | |
| DSP_annunLWR_CTR | bool | | | |
| **Glareshield – Master Warning/Caution** | | | | |
| WARN_Reset_Sw_Pushed | bool | 0-1 | | EVT_MASTER_WARNING_RESET_*LEFT* (also RIGHT) |
| WARN_annunMASTER_WARNING | bool | 0-1 | | |
| WARN_annunMASTER_CAUTION | bool | 0-1 | | |
| **Glareshield – Click-spots** | | | | |
| | | | | EVT_CLICKSPOT_ISFD<br>EVT_CLICKSPOT_STBY_ADI<br>EVT_CLICKSPOT_STBY_ASI<br>EVT_CLICKSPOT_STBY_ALTIMETER<br>EVT_CLICKSPOT_GMCS_ZOOM |
| **Glareshield – Miscellaneous** | | | | |
| | | | | EVT_CLOCK_*L* (also R)<br>EVT_MAP_LIGHT_*L*<br>EVT_GLARESHIELD_MIC_*L*<br>EVT_CHART_LIGHT_*L*<br>EVT_WORKTABLE_LIGHT_*L* |
| **Forward Aisle Stand Panel** | | | | |
| ISP_DsplCtrl_C_Sw_Altn | bool | | | EVT_PED_DSPL_CTRL_SOURCE_C |
| LTS_UpperDsplBRIGHTNESSKnob | number | | 0-150 | EVT_PED_UPPER_BRIGHT_CONTROL |
| LTS_LowerDsplBRIGHTNESSKnob | number | | 0-150 | EVT_PED_LOWER_BRIGHT_CONTROL |
| EICAS_EventRcd_Sw_Pushed | bool | | | EVT_PED_EICAS_EVENT_RCD |

| | | | | |
|---|---|---|---|---|
| CDUs | | | | |
| CDU_annunEXEC | bool | 0-2 | | EVT_CDU_*L*_EXEC (also C and R) |
| 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, R) |
| **Control Stand – Flight Controls** | | | | |
| FCTL_AltnFlaps_Sw_ARM | bool | | | EVT_ALTN_FLAPS_ARM<br>EVT_ALTN_FLAPS_ARM_GUARD |
| FCTL_AltnFlaps_Control_Sw | number | | 0: RET, 1: OFF, 2: EXT | EVT_ALTN_FLAPS_POS |
| FCTL_StabCutOutSw_C_NORMAL | bool | | | EVT_CONTROL_STAND_STABCUTOUT_SWITCH_C_GUARD<br>EVT_CONTROL_STAND_STABCUTOUT_SWITCH_C |
| FCTL_StabCutOutSw_R_NORMAL | bool | | | EVT_CONTROL_STAND_STABCUTOUT_SWITCH_R_GUARD<br>EVT_CONTROL_STAND_STABCUTOUT_SWITCH_R |
| FCTL_AltnPitch_Lever | number | | 0: nose down, 1: neutral,<br>2: nose up | EVT_CONTROL_STAND_ALT_PITCH_TRIM_LEVER |
| FCTL_Speedbrake_Lever | number | | 0..100<br>0: Down<br>25: Armed<br>26-100 Deployed | EVT_CONTROL_STAND_SPEED_BRAKE_LEVER<br>EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_DOWN<br>EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_ARM<br>EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_UP<br>EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_50 |
| FCTL_Flaps_Lever | number | | 0: Up<br>1: 1<br>2: 5<br>3: 15<br>4: 20<br>5: 25<br>6: 30 | EVT_CONTROL_STAND_FLAPS_LEVER<br>EVT_CONTROL_STAND_FLAPS_LEVER_0<br>EVT_CONTROL_STAND_FLAPS_LEVER_1<br>EVT_CONTROL_STAND_FLAPS_LEVER_5<br>EVT_CONTROL_STAND_FLAPS_LEVER_15<br>EVT_CONTROL_STAND_FLAPS_LEVER_20<br>EVT_CONTROL_STAND_FLAPS_LEVER_25<br>EVT_CONTROL_STAND_FLAPS_LEVER_30 |
| ENG_FuelControl_Sw_RUN | bool | 0-1 | | EVT_CONTROL_STAND_*ENG1*_START_LEVER (also ENG2) |
| BRAKES_ParkingBrakeLeverOn | bool | | | EVT_CONTROL_STAND_PARK_BRAKE_LEVER |
| | | | | EVT_YOKE_AP_DISC_SWITCH |
| | | | | EVT_CONTROL_STAND_REV_THRUST*1*_LEVER (also 2) |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_CONTROL_STAND_FWD_THRUST*1*_LEVER<br>EVT_CONTROL_STAND_AT*1*_DISENGAGE_SWITCH<br>EVT_CONTROL_STAND_TOGA*1*_SWITCH |
| **Aft Aisle Stand Panel – Audio Control** | | | | |
| COMM_SelectedMic | number | 0-2 | 0: VHFL<br>1: VHFC<br>2: VHFR<br>3: FLT<br>4: CAB<br>5: PA<br>6: HFL<br>7: HFR<br>8: SAT1<br>9: SAT2<br>-1 if none selected | EVT_ACP_*CAPT*_MIC_VHFL (also FO and OBS)<br>EVT_ACP_*CAPT*_MIC_VHFC<br>EVT_ACP_*CAPT*_MIC_VHFR<br>EVT_ACP_*CAPT*_MIC_FLT<br>EVT_ACP_*CAPT*_MIC_CAB<br>EVT_ACP_*CAPT*_MIC_PA<br>EVT_ACP_*CAPT*_MIC_HFL<br>EVT_ACP_*CAPT*_MIC_HFR<br>EVT_ACP_*CAPT*_MIC_SAT1<br>EVT_ACP_*CAPT*_MIC_SAT2 |
| COMM_ReceiverSwitches | number | 0-2 | Bit 0: VHFL<br>Bit 1: VHFC<br>Bit 2: VHFR<br>Bit 3: FLT<br>Bit 4: CAB<br>Bit 5: PA<br>Bit 6: HFL<br>Bit 7: HFR<br>Bit 8: SAT1<br>Bit 9: SAT2<br>Bit 10: SPKR<br>Bit 11: VORADF<br>Bit 12: APP | EVT_ACP_*CAPT*_REC_VHFL (also FO and OBS)<br>EVT_ACP_*CAPT*_REC_VHFC<br>EVT_ACP_*CAPT*_REC_VHFR<br>EVT_ACP_*CAPT*_REC_FLT<br>EVT_ACP_*CAPT*_REC_CAB<br>EVT_ACP_*CAPT*_REC_PA<br>EVT_ACP_*CAPT*_REC_HFL<br>EVT_ACP_*CAPT*_REC_HFR<br>EVT_ACP_*CAPT*_REC_SAT1<br>EVT_ACP_*CAPT*_REC_SAT2<br>EVT_ACP_*CAPT*_REC_SPKR<br>EVT_ACP_*CAPT*_REC_VORADF<br>EVT_ACP_*CAPT*_REC_APP |
| | | | | EVT_ACP_*CAPT*_MIC_INT_SWITCH<br>EVT_ACP_*CAPT*_FILTER_SELECTOR<br>EVT_ACP_*CAPT*_VOR_ADF_SELECTOR<br>EVT_ACP_*CAPT*_APP_SELECTOR |
| COMM_OBSAudio_Selector | number | | 0: CAPT, 1: normal, 2: FO | EVT_PED_OBS_AUDIO_SELECTOR |
| | | | | EVT_PED_CALL_GND<br>EVT_PED_CALL_PANEL_FIRST_SWITCH<br>EVT_PED_CALL_CREW_REST |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_PED_CALL_SUPRNMRY<br>EVT_PED_CALL_CARGO<br>EVT_PED_CALL_CARGO_AUDIO<br>EVT_PED_CALL_MAIN_DK_ALERT<br>EVT_PED_CALL_PANEL_LAST_SWITCH |
| **Aft Aisle Stand Panel – Radio Control** | | | | |
| COMM_SelectedRadio | numbe<br>r | 0-2 | 0: VHFL<br>1: VHFC<br>2: VHFR<br>3: HFL<br>5: HFR (4 not used) | EVT_*COM1*_VHFL_SWITCH (also COM2 and COM3)<br>EVT_*COM1*_VHFC_SWITCH<br>EVT_*COM1*_VHFR_SWITCH<br>EVT_*COM1*_HFL_SWITCH<br>EVT_*COM1*_HFR_SWITCH |
| 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_*COM1*_AM_SWITCH (also COM2 and COM3)<br>EVT_*COM1*_HF_SENSOR_KNOB<br>EVT_*COM1*_OUTER_SELECTOR<br>EVT_*COM1*_INNER_SELECTOR |
| **Aft Aisle Stand Panel – TCAS** | | | | |
| XPDR_XpndrSelector_R | bool | | True: right, False: left | EVT_TCAS_XPNDR |
| XPDR_AltSourceSel_ALTN | bool | | True: ALIN, False: NORM | EVT_TCAS_ALTSOURCE |
| XPDR_ModeSel | numbe<br>r | | 0: STBY, 1: ALT RPTG OFF<br>2: XPNDR, 3: TA ONLY<br>4: TA/RA | EVT_TCAS_MODE |
| XPDR_Ident_Sw_Pushed | bool | | | EVT_TCAS_IDENT |
| | | | | EVT_TCAS_KNOB_L_OUTER<br>EVT_TCAS_KNOB_L_INNER<br>EVT_TCAS_KNOB_R_OUTER<br>EVT_TCAS_KNOB_R_INNER<br>EVT_TCAS_TEST |
| **Aft Aisle Stand Panel – Weather Radar** | | | | |
| | | | | EVT_WXR_L_TFR<br>EVT_WXR_L_WX<br>EVT_WXR_L_WX_T<br>EVT_WXR_L_MAP<br>EVT_WXR_L_GC |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_WXR_AUTO<br>EVT_WXR_L_R<br>EVT_WXR_TEST<br>EVT_WXR_R_TFR<br>EVT_WXR_R_WX<br>EVT_WXR_R_WX_T<br>EVT_WXR_R_MAP<br>EVT_WXR_R_GC<br>EVT_WXR_L_TILT_CONTROL<br>EVT_WXR_L_GAIN_CONTROL<br>EVT_WXR_R_TILT_CONTROL<br>EVT_WXR_R_GAIN_CONTROL |
| **Aft Aisle Stand Panel – Engine Fire** | | | | |
| FIRE_EngineHandle | number | 0-1 | 0: IN, 1: Pulled out<br>2: turned left, 3: turned right | EVT_FIRE_HANDLE_ENGINE_*1*_TOP (also 2)<br>EVT_FIRE_HANDLE_ENGINE_*1*_BOTTOM |
| FIRE_EngineHandleUnlock_Sw | bool | 0-1 | | EVT_FIRE_UNLOCK_SWITCH_ENGINE_*1* (also 2) |
| FIRE_annunENG_BTL_DISCH | bool | 0-1 | | |
| **Aft Aisle Stand Panel – Aileron & Rudder Trim** | | | | |
| FCTL_AileronTrim_Switches | number | | 0: Left wing down, 1: Neutral<br>2: Right wing down | EVT_FCTL_AILERON_TRIM |
| FCTL_RudderTrim_Knob | number | | 0: Nose left, 1: Neutral,<br>2: Nose right | EVT_FCTL_RUDDER_TRIM |
| FCTL_RudderTrimCancel_Sw_Pushed | bool | | | EVT_FCTL_RUDDER_TRIM_CANCEL |
| **Aft Aisle Stand Panel – Evacuation Panel** | | | | |
| EVAC_Command_Sw_ON | bool | | | EVT_PED_EVAC_SWITCH<br>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 | | | |
| **Aft Aisle Stand Panel – Flood & Floor Lights** | | | | |
| LTS_AisleStandPNLKnob | number | | 0-150 | EVT_PED_PANEL_LIGHT_CONTROL |
| LTS_AisleStandFLOODKnob | number | | 0-150 | EVT_PED_FLOOD_LIGHT_CONTROL |
| LTS_FloorLightsSw | number | | 0: BRT, 1: OFF, 2: DIM | EVT_PED_FLOOR_LIGHTS |

| | | | | |
|---|---|---|---|---|
| | | | | EVT_PED_LOWER_TERR_BRIGHT_CONTROL |
| **Pedestal – Miscellaneous** | | | | |
| | | | | EVT_DATA_LINK_ACPT_*LEFT* (also RIGHT)<br>EVT_DATA_LINK_CANC_*LEFT*<br>EVT_DATA_LINK_RJCT_*LEFT* |
| | | | | EVT_PED_*L*_CCD_SIDE (also R)<br>EVT_PED_*L*_CCD_INBD<br>EVT_PED_*L*_CCD_LWR |
| **Additional Data – Door State** | | | | |
| DOOR_state | number | 0-15 | 0: open<br>1: closed<br>2: closed and armed<br>3: closing<br>4: opening | EVT_DOOR_1L<br>EVT_DOOR_1R<br>EVT_DOOR_2L<br>EVT_DOOR_2R<br>EVT_DOOR_3L (just aft of wing, marked 4L for -300)<br>EVT_DOOR_3R (just aft of wing, marked 4R for -300)<br>EVT_DOOR_4L (marked 5L for -300)<br>EVT_DOOR_4R (marked 5L for -300)<br>EVT_DOOR_OVERWING_EXIT_L<br>EVT_DOOR_OVERWING_EXIT_R<br>EVT_DOOR_CARGO_FWD<br>EVT_DOOR_CARGO_AFT<br>EVT_DOOR_CARGO_MAIN (Freighter)<br>EVT_DOOR_CARGO_BULK<br>EVT_DOOR_FWD_ACCESS<br>EVT_DOOR_EE_ACCESS |
| **Addition Data – Miscellaneous** | | | | |
| ENG_StartValve | bool | 0-1 | True: open, False: closed | |
| AIR_DuctPress | number | 0-1 | PSI | |
| FUEL_QtyCenter | number | | LBS | |
| FUEL_QtyLeft | number | | LBS | |
| FUEL_QtyRight | number | | LBS | |
| FUEL_QtyAux | number | | LBS | |

| | | | | |
|---|---|---|---|---|
| IRS_aligned | bool | | True: at least one is aligned | |
| AircraftModel | number | | 1: -200, 2: -200ER, 3: -300<br>4: -200LR, 5: -200F, 6: -300ER | |
| WeightInKg | bool | | True: KG, False: LBS | |
| GPWS_V1CallEnabled | bool | | | |
| GroundConnAvailable | bool | | | |
| FMC_TakeoffFlaps | number | | Degrees, 0 if not set | |
| FMC_V1 | number | | Knots, 0 if not set | |
| FMC_VR | number | | Knots, 0 if not set | |
| FMC_V2 | number | | Knots, 0 if not set | |
| FMC_LandingFlaps | number | | Degrees, 0 if not set | |
| FMC_LandingVREF | number | | Knots, 0 if not set | |
| FMC_CruiseAlt | number | | Feet, 0 if not set | |
| FMC_LandingAltitude | number | | Feet, -32767 if not available | |
| FMC_TransitionAlt | number | | Feet | |
| FMC_TransitionLevel | number | | Feet | |
| FMC_PerfInputComplete | bool | | | |
| FMC_DistanceToTOD | number | | Nm, 0.0 if passed, <0 if n/a | |
| FMC_DistanceToDest | number | | Nm, <0 if n/a | |
| FMC_flightNumber | char | 0-8 | | |

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

## Controlling the 777X

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 777X 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_NOSE”, and value 536870912, will do the same as clicking with the left mouse button on the nose landing light switch.

### Using Data Events

The Honeycomb Alpha Yoke has a rotary knob for the magnetos on GA aircraft. On the 777 we do not have those, but the 777X 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_T7_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 mapps 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 777X 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_T7_LTS_LandingLights_Sw_ON”, index 0 (the left switch), unit “bool”, and value “0”. This will make the condition trigger on the left fixed 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_T7_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 777X 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_T7” 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 not 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_T7_MCP_annunHDG_HOLD, bool =1 | |
| FCU-NAV | NAV | 1, 1 | “Nav hold” mode is on, in the 777 profile “VOR LOC”. | L:AS_PMDG_T7_MCP_annunLOC, bool =1 | |
| FCU-APR | APR | 1, 2 | “Approach” mode is on. | L:AS_PMDG_T7_MCP_annunAPP, bool =1 | |
| FCU-REV | REV | 1, 3 | “Back course hold” is on, in the 777 profile “LNAV”. | L:AS_PMDG_T7_MCP_annunLNAV, bool =1 | |
| FCU-ALT | ALT | 1, 4 | “Altitude hold” mode is on, in the 777 profile “VNAV”. | L:AS_PMDG_T7_MCP_annunVNAV, bool =1 | |
| FCU-VS | VS | 1, 5 | “Vertical speed” mode is on. | L:AS_PMDG_T7_MCP_annunFLCH, bool =1 | |
| FCU-IAS | IAS | 1, 6 | “IAS hold” or “Speed hold” mode is on, in the 777 profile “AT” on. | L:AS_PMDG_T7_MCP_annunAT, bool =1 | |
| GCU-AP | AUTO PILOT | 1, 7 | Autopilot is switched on. For the 777 profile this means any of the two switches is highlighted. | L:AS_PMDG_T7_MCP_annunAP, 0, bool =1<br>L:AS_PMDG_T7_MCP_annunAP, 1, 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<br>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<br>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<br>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_T7_WARN_annunMASTER_WARNING, 0, bool =1<br>L:AS_PMDG_T7_WARN_annunMASTER_WARNING, 1, bool =1 | OR |
| ANC-ENG_FIRE | ENGINE FIRE | 2, 7 | Lights when the “Fire Overheat Test” switch is pushed. | L: AS_PMDG_T7_FIRE_FireIvhtTest_Sw, bool =1 | |
| ANC-OIL | LOW OIL PRESSURE | 3, 0 | Unused | | |
| ANC-FUEL | LOW FUEL PRESSURE | 3, 1 | The overhead fuel annunciators. | L:AS_PMDG_T7_FUEL_annunLOWPRESS_Fwd, 0, bool =1<br>L:AS_PMDG_T7_FUEL_annunLOWPRESS_Fwd, 1, bool =1<br>L:AS_PMDG_T7_FUEL_annunLOWPRESS_Aft, 0, bool =1<br>L:AS_PMDG_T7_FUEL_annunLOWPRESS_Aft, 1, bool =1<br>L:AS_PMDG_T7_FUEL_annunLOWPRESS_Ctr, 0, bool =1<br>L:AS_PMDG_T7_FUEL_annunLOWPRESS_Ctr, 1, bool =1 | OR |
| ANC-ANTI_ICE | ANTI ICE | 3, 2 | The anti-ice annunciators. | L:AS_PMDG_T7_ICE_annunWindowHeatINOP, 0, bool =1<br>L:AS_PMDG_T7_ICE_annunWindowHeatINOP, 1, bool =1<br>L:AS_PMDG_T7_ICE_annunWindowHeatINOP, 2, bool =1<br>L:AS_PMDG_T7_ICE_annunWindowHeatINOP, 3, bool =1 | OR |
| ANC-STARTER | STARTER ENGAGED | 3, 3 | | | |
| ANC-APU | APU | 3, 4 | The APU fault annunciator. | L:AS_PMDG_T7_ELEC_annunAPU_FAULT, bool =1 | |
| ANC-MSTR_CTN | MASTER CAUTION | 3, 5 | Master caution annunciator. | L:AS_PMDG_T7_WARN_annunMASTER_CAUTION, 0, bool =1<br>L:AS_PMDG_T7_WARN_annunMASTER_CAUTION, 1, bool =1 | OR |

| | | | | | |
|---|---|---|---|---|---|
| ANC-VACUUM | VACUUM | 3, 6 | The air conditioning annunciators. | L:AS_PMDG_T7_AIR_annunPackOFF, 0, bool =1<br>L:AS_PMDG_T7_AIR_annunPackOFF, 1, bool =1<br>L:AS_PMDG_T7_AIR_annunTrimAirFAULT, 0, bool =1<br>L:AS_PMDG_T7_AIR_annunTrimAirFAULT, 1, bool =1<br>L:AS_PMDG_T7_AIR_annunEngBleedAirOFF, 0, bool =1<br>L:AS_PMDG_T7_AIR_annunEngBleedAirOFF, 1, bool =1<br>L:AS_PMDG_T7_AIR_annunAPUBleedAirOFF, bool =1<br>L:AS_PMDG_T7_AIR_annunIsolationValveCLOSED, 0, bool =1<br>L:AS_PMDG_T7_AIR_annunIsolationValveCLOSED, 1, bool =1 | OR |
| ANC-HYD | LOW HYD PRESSURE | 3, 7 | The hydraulics annunciators. | L:AS_PMDG_T7_HYD_annunPrimaryEngPumpFAULT, 0, bool<br>L:AS_PMDG_T7_HYD_annunPrimaryEngPumpFAULT, 1, bool<br>L:AS_PMDG_T7_HYD_annunPrimaryElecPumpFAULT, 0, bool<br>L:AS_PMDG_T7_HYD_annunPrimaryElecPumpFAULT, 1, bool<br>L:AS_PMDG_T7_HYD_annunDemandElecPumpFAULT, 0, bool<br>L:AS_PMDG_T7_HYD_annunDemandElecPumpFAULT, 1, bool<br>L:AS_PMDG_T7_HYD_annunDemandAirPumpFAULT, 0, bool<br>L:AS_PMDG_T7_HYD_annunDemandAirPumpFAULT, 1, 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_T7_BRAKES_ParkingBrakeLeverOn, bool =1 | |
| ANC-VOLTS | LOW VOLTS | 4, 2 | The electrical systems annunciator. | L:AS_PMDG_T7_ELEC_annunGenOFF, 0, bool =1<br>L:AS_PMDG_T7_ELEC_annunGenOFF, 1, bool =1<br>L:AS_PMDG_T7_ELEC_annunCabUtilOFF, bool =1<br>L:AS_PMDG_T7_ELEC_annunBusTieISLN, 0, bool =1<br>L:AS_PMDG_T7_ELEC_annunBusTieISLN, 1, bool =1<br>L:AS_PMDG_T7_ELEC_annunIDGDiscDRIVE, 0, bool =1<br>L:AS_PMDG_T7_ELEC_annunIDGDiscDRIVE, 1, bool =1 | OR |
| ANC-DOOR | DOOR | 4, 3 | Any of the doors is not closed and armed. | L:AS_PMDG_T7_DOOR_state, 0, number !=2<br>L:AS_PMDG_T7_DOOR_state, 1, number !=2<br>L:AS_PMDG_T7_DOOR_state, 2, number !=2<br>L:AS_PMDG_T7_DOOR_state, 3, number !=2<br>L:AS_PMDG_T7_DOOR_state, 4, number !=2<br>L:AS_PMDG_T7_DOOR_state, 5, number !=2<br>L:AS_PMDG_T7_DOOR_state, 6, number !=2<br>L:AS_PMDG_T7_DOOR_state, 7, number !=2 | OR |

In addition, all LEDs are also OR- linked to L:AS_PMDG_T7_LTS_IndLightsTestSw,number = 0, so that you can switch on all mapped LEDs with the IND LTS switch in the TEST position.