Variables and Conditions for the PMDG 737 NGXu

Introduction

To find out what is happening in the PMDG 737 NGXu, 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_NG3_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 737 NGXu 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_NG3_Data**" structure of the SDK. 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_737**". All names start with "**AS_PMDG_737NGX_**" or "**AS_PMDG_737NGXu_**", so in this table the initial "**AS_PMDG_737**" is dropped. 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, "**NGX_ENG_EECSwitch**", 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
Aft Overhead – ADIRU				
NGX_IRS_DisplaySelector	number		Positions: 0-4	NGX_EVT_ISDU_DSPL_SEL
NGX_IRS_SysDisplay_R	bool		True: R False: L	
NGX_IRS_annunGPS	bool			
NGX_IRS_annunALIGN	bool	0-1		
NGX_IRS_annunON_DC	bool	0-1		
NGX_IRS_annunFAULT	bool	0-1		
NGX_IRS_annunDC_FAIL	bool	0-1		
NGX_IRS_ModeSelector	number	0-1	0: Off, 1: Align 2: Nav, 3: Att	NGX_EVT_IRU_MSU_***LEFT*** (also RIGHT)
				NGXu_EVT_IRU_MSU_RIGHT_INOUT (also RIGHT)
Aft Overhead – ISDU				
				NGX_EVT_ISDU_DSPL_SEL_BRT NGX_EVT_ISDU_SYS_DSPL NGX_EVT_ISDU_KBD_***0*** (also 1 through 9) NGX_EVT_ISDU_KBD_ENT NGX_EVT_ISDU_KBD_CLR
Aft Overhead –PSEU				
NGX_WARN_annunPSEU	bool			
Aft Overhead – Service Interphone				

NGX_COMM_ServiceInterphoneSw	bool			NGX_EVT_OH_SERVICE_INTERPHONE_SWITCH
Aft Overhead – Lights				
NGX_LTS_DomeWhiteSw	number		0: Dim, 1: Off 2: Bright	NGX_EVT_OH_DOME_SWITCH
Aft Overhead – Engines				
NGX_ENG_EECSwitch	bool	0-1		NGX_EVT_OH_EEC_***L***_GUARD (also R) NGX_EVT_OH_EEC_***L***_SWITCH
NGX_ENG_annunREVERSER	bool	0-1		
NGX_ENG_annunENGINE_CONTROL	bool	0-1		
NGX_ENG_annunALTN	bool	0-1		
Aft Overhead – Oxygen				
NGX_OXY_Needle	number		0-240	
NGX_OXY_SwNormal	bool		True: Normal False: On	NGX_EVT_OH_OXY_PASS_GUARD NGX_EVT_OH_OXY_PASS_SWITCH
NGX_OXY_annunPASS_OXY_ON	bool			
				NGX_EVT_OH_OXY_RED_BUTTON_***L*** (also R) NGX_EVT_OH_OXY_TEST_RESET_SWITCH_***L***
Aft Overhead – Gear				
NGX_GEAR_annunOvhdLEFT	bool			
NGX_GEAR_annunOvhdNOSE	bool			
NGX_GEAR_annunOvhdRIGHT	bool			
Aft Overhead – Flight recorder				
NGX_FLTREC_SwNormal	bool		True: Normal False: Test	NGX_EVT_OH_FLTREC_GUARD NGX_EVT_OH_FLTREC_SWITCH
NGX_FLTREC_annunOFF	bool			
Forward Overhead – Flight controls				
NGX_FCTL_FltControl_Sw	number	0-1	0: STBY/RUD 1: Off, 2: On	NGX_EVT_OH_FCTL_***A***_GUARD (also B) NGX_EVT_OH_FCTL_***A***_SWITCH
NGX_FCTL_Spoiler_Sw	bool	0-1	True: On False: Off	NGX_EVT_OH_SPOILER_***A***_GUARD (also B) NGX_EVT_OH_SPOILER_***A***_SWITCH
NGX_FCTL_YawDamper_Sw	bool			NGX_EVT_OH_YAW_DAMPER
NGX_FCTL_AltnFlaps_Sw_ARM	bool		True: Arm False: Off	NGX_EVT_OH_ALT_FLAPS_MASTER_GUARD NGX_EVT_OH_ALT_FLAPS_MASTER_SWITCH
NGX_FCTL_AltnFlaps_Control_Sw	number		0: Up, 1: Off, 2: Down	NGX_EVT_OH_ALT_FLAPS_POS_SWITCH
NGX_FCTL_annunFC_LOW_PRESSURE	bool	0-1		

NGX_FCTL_annunYAW_DAMPER	bool			
NGX_FCTL_annunLOW_QUANTITY	bool			
NGX_FCTL_annunLOW_PRESSURE	bool			
NGX_FCTL_annunLOW_STBY_RUD_ON	bool			
NGX_FCTL_annunFEEL_DIFF_PRESS	bool			
NGX_FCTL_annunSPEED_TRIM_FAIL	bool			
NGX_FCTL_annunMACH_TRIM_FAIL	bool			
NGX_FCTL_annunAUTO_SLAT_FAIL	bool			
Forward Overhead – Navigation / Displays				
NGX_NAVDIS_VHFNavSelector	number		0: Both on 1 1: Normal 2: Both on 2	NGX_EVT_OH_NAVDSP_VHF_NAV_SEL
NGX_NAVDIS_IRSSelector	number		0: Both on L 1: Normal 2: Both on R	NGX_EVT_OH_NAVDSP_IRS_SEL
NGX_NAVDIS_FMCSelector	number		0: Both on L 1: Normal 2: Both on R	NGX_EVT_OH_NAVDSP_FMC_SEL
NGX_NAVDIS_SourceSelector	number		0: All on 1 1: Auto 2: All on 2	NGX_EVT_OH_NAVDSP_DISPLAYS_SOURCE_SEL
NGX_NAVDIS_ControlPaneSelector	number		0: Both on 1 1: Normal 2: Both on 2	NGX_EVT_OH_NAVDSP_CONTROL_PANEL_SEL
Forward Overhead – Fuel				
NGX_FUEL_FuelTempNeedle	number			
NGX_FUEL_CrossFeedSw	bool			NGX_EVT_OH_FUEL_CROSSFEED
NGX_FUEL_PumpFwdSw	bool	0-1	Left/right forward	NGX_EVT_OH_FUEL_PUMP_***1***_FORWARD (also 2)
NGX_FUEL_PumpAftSw	bool	0-1	Left / right aft	NGX_EVT_OH_FUEL_PUMP_***1***_AFT (also 2)
NGX_FUEL_PumpCtrSw	bool	0-1	Left / right center	NGX_EVT_OH_FUEL_PUMP_***L***_CENTER (also R)
NGX_FUEL_annunENG_VALVE_CLOSED	bool	0-1	0: Closed 1: Open 2: In transit	
NGX_FUEL_annunSPAR_VALVE_CLOSED	bool	0-1	0: Closed 1: Open 2: In transit	
NGX_FUEL_annunFILTER_BYPASS	bool	0-1		

NGX_FUEL_annunXFEED_VALVE_OPEN	number		0: Closed 1: Open 2: In transit	
NGX_FUEL_annunLOWPRESS_Fwd	bool	0-1		
NGX_FUEL_annunLOWPRESS_Aft	bool	0-1		
NGX_FUEL_annunLOWPRESS_Ctr	bool	0-1		
				NGXu_EVT_OH_FUEL_AFT_BLD NGXu_EVT_OH_FUEL_AUX_AFT_A NGXu_EVT_OH_FUEL_AUX_AFT_B NGXu_EVT_OH_FUEL_AUX_FWD_A NGXu_EVT_OH_FUEL_AUX_FWD_B NGXu_EVT_OH_FUEL_FWD_BLD NGXu_EVT_OH_FUEL_GND_XFR_GUARD NGXu_EVT_OH_FUEL_GND_XFR_SW
Forward Overhead – Electrical				
NGX_ELEC_annunBAT_DISCHARGE	bool			
NGX_ELEC_annunTR_UNIT	bool			
NGX_ELEC_annunELEC	bool			
NGX_ELEC_DCMeterSelector	number		0-7	NGX_EVT_OH_ELEC_DC_METER
NGX_ELEC_ACMeterSelector	number		0-6	NGX_EVT_OH_ELEC_AC_METER
NGX_ELEC_BatSelector	number		0: Off, 1: Bat 2: On	NGX_EVT_OH_ELEC_BATTERY_GUARD NGX_EVT_OH_ELEC_BATTERY_SWITCH
NGX_ELEC_CabUtilSw	bool			NGX_EVT_OH_ELEC_CAB_UTIL
NGX_ELEC_IFEPassSeatSw	bool			NGX_EVT_OH_ELEC_IFE
NGX_ELEC_annunDRIVE	bool	0-1		
NGX_ELEC_annunSTANDBY_POWER_OFF	bool			
NGX_ELEC_IDGDisconnectSw	bool	0-1		NGX_EVT_OH_ELEC_DISCONNECT_***1***_GUARD (also 2) NGX_EVT_OH_ELEC_DISCONNECT_1_SWITCH
NGX_ELEC_StandbyPowerSelector	number		0: Bat, 1: off 2: auto	NGX_EVT_OH_ELEC_STBY_PWR_GUARD NGX_EVT_OH_ELEC_STBY_PWR_SWITCH
NGX_ELEC_annunGRD_POWER_AVAILABLE	bool			
NGX_ELEC_GrdPwrSw	bool			NGX_EVT_OH_ELEC_GRD_PWR_SWITCH
NGX_ELEC_BusTransSw_AUTO	bool			NGX_EVT_OH_BUS_TRANSFER_GUARD NGX_EVT_OH_BUS_TRANSFER_SWITCH
NGX_ELEC_GenSw	bool	0-1		NGX_EVT_OH_ELEC_GEN***1***_SWITCH (also 2)
NGX_ELEC_APUGenSw	bool	0-1		NGX_EVT_OH_ELEC_APU_GEN***1***_SWITCH (also 2)
NGX_ELEC_annunTRANSFER_BUS_OFF	bool	0-1		

NGX_ELEC_annunSOURCE_OFF	bool	0-1		
NGX_ELEC_annunGEN_BUS_OFF	bool	0-1		
NGX_ELEC_annunAPU_GEN_OFF_BUS	bool			
				NGX_EVT_OH_ELEC_GALLEY NGX_EVT_OH_ELEC_MAINT_SWITCH
Forward Overhead – APU				
NGX_APU_EGTNeedle	number			
NGX_APU_annunMAINT	bool			
NGX_APU_annunLOW_OIL_PRESSURE	bool			
NGX_APU_annunFAULT	bool			
NGX_APU_annunOVERSPEED	bool			
Forward Overhead – Wipers				
NGX_OH_WiperLSelector	number		0: park 1: interval 2: low 3: high	NGX_EVT_OH_WIPER_LEFT_CONTROL
NGX_OH_WiperRSelector	number			NGX_EVT_OH_WIPER_RIGHT_CONTROL
Forward Overhead – Center overhead controls & indicators				
NGX_LTS_CircuitBreakerKnob	number		0-150	NGX_EVT_OH_CB_LIGHT_CONTROL
NGX_LTS_OvereadPanelKnob	number		0-150	NGX_EVT_OH_PANEL_LIGHT_CONTROL
NGX_AIR_EquipCoolingSupplyNORM	bool			NGX_EVT_OH_EC_SUPPLY_SWITCH
NGX_AIR_EquipCoolingExhaustNORM	bool			NGX_EVT_OH_EC_EXHAUST_SWITCH
NGX_AIR_annunEquipCoolingSupplyOFF	bool			
NGX_AIR_annunEquipCoolingExhaustOFF	bool			
NGX_LTS_annunEmerNOT_ARMED	bool			
NGX_LTS_EmerExitSelector	number		0: off, 1: armed, 2: on	NGX_EVT_OH_EMER_EXIT_LIGHT_GUARD NGX_EVT_OH_EMER_EXIT_LIGHT_SWITCH
NGX_COMM_NoSmokingSelector	number		0: off, 1: auto, 2: on	NGX_EVT_OH_NO_SMOKING_LIGHT_SWITCH
NGX_COMM_FastenBeltsSelector	number		0: off, 1: auto, 2: on	NGX_EVT_OH_FASTEN_BELTS_LIGHT_SWITCH
				NGX_EVT_OH_ATTND_CALL_SWITCH
NGX_COMM_annunCALL	bool			
NGX_COMM_annunPA_IN_USE	bool			
Forward Overhead – Anti-ice				
NGX_ICE_annunOVERHEAT	bool	0-3		
NGX_ICE_annunON	bool	0-3		
NGX_ICE_WindowHeatSw	bool	0-3		NGX_EVT_OH_ICE_WINDOW_HEAT_1 (also 2, 3, and 4)

NGX_ICE_annunCAPT_PITOT	bool			
NGX_ICE_annunL_ELEV_PITOT	bool			
NGX_ICE_annunL_ALPHA_VANE	bool			
NGX_ICE_annunL_TEMP_PROBE	bool			
NGX_ICE_annunFO_PITOT	bool			
NGX_ICE_annunR_ELEV_PITOT	bool			
NGX_ICE_annunR_ALPHA_VANE	bool			
NGX_ICE_annunAUX_PITOT	bool			
NGX_ICE_ProbeHeatSw	bool	0-1		NGX_EVT_OH_ICE_PROBE_HEAT_***1*** (also 2)
NGX_ICE_annunVALVE_OPEN	bool	0-1		
NGX_ICE_annunCOWL_ANTI_ICE	bool	0-1		
NGX_ICE_annunCOWL_VALVE_OPEN	bool	0-1		
NGX_ICE_WingAntiIceSw	bool			NGX_EVT_OH_ICE_WING_ANTIICE
NGX_ICE_EngAntiIceSw	bool	0-1		NGX_EVT_OH_ICE_ENGINE_ANTIICE_***1*** (also 2)
Forward Overhead – Hydraulics				
NGX_HYD_annunLOW_PRESS_eng	bool	0-1		
NGX_HYD_annunLOW_PRESS_elec	bool	0-1		
NGX_HYD_annunOVERHEAT_elec	bool	0-1		
NGX_HYD_PumpSw_eng	bool	0-1		NGX_EVT_OH_HYD_ENG***1*** (also 2)
NGX_HYD_PumpSw_elec	bool	0-1		NGX_EVT_OH_HYD_ELEC***1*** (also 2)
Forward Overhead – Air systems				
NGX_AIR_TempSourceSelector	number		0-6	
NGX_AIR_TrimAirSwitch	bool			NGX_EVT_OH_TRIM_AIR_SWITCH_TOGGLE
				(737-600/700 only) NGX_EVT_OH_AIRCOND_TEMP_SELECTOR_CABIN NGX_EVT_OH_AIRCOND_TEMP_SELECTOR_CONT NGX_EVT_OH_AIRCOND_TEMP_SOURCE_SELECTOR (737-800/900 only) NGX_EVT_OH_AIRCOND_TEMP_SELECTOR_AFT_800 NGX_EVT_OH_AIRCOND_TEMP_SELECTOR_CONT_800 NGX_EVT_OH_AIRCOND_TEMP_SELECTOR_FWD_800 NGX_EVT_OH_AIRCOND_TEMP_SOURCE_SELECTOR_800 NGX_EVT_OH_AIRCOND_TRIM_AIR_SWITCH_800
NGX_AIR_annunZoneTemp	bool	0-2		
NGX_AIR_annunDualBleed	bool			
NGX_AIR_annunRamDoorL	bool			

NGX_AIR_annunRamDoorR	bool			
NGX_AIR_RecircFanSwitch	bool	0-1		NGX_EVT_OH_BLEED_RECIRC_FAN_***L***_SWITCH (also R)
NGX_AIR_PackSwitch	number	0-1	0: off, 1: auto, 2: high	NGX_EVT_OH_BLEED_PACK_***L***_SWITCH (also R)
NGX_AIR_BleedAirSwitch	bool	0-1		NGX_EVT_OH_BLEED_ENG_***1***_SWITCH (also 2)
NGX_AIR_APUBleedAirSwitch	bool			NGX_EVT_OH_BLEED_APU_SWITCH
NGX_AIR_IsolationValveSwitch	number		0: close, 1: auto, 2: open	NGX_EVT_OH_BLEED_ISOLATION_VALVE_SWITCH
				NGX_EVT_OH_BLEED_TRIP_RESET_BUTTON
NGX_AIR_annunPackTripOff	bool	0-1		
NGX_AIR_annunWingBodyOverheat	bool	0-1		
NGX_AIR_annunBleedTripOff	bool	0-1		
NGXu_AIR_DisplayFltAlt	char	0-5	(Use mouse events for knob)	NGX_EVT_OH_PRESS_FLT_ALT_SET NGX_EVT_OH_PRESS_FLT_ALT_KNOB
NGXu_AIR_DisplayLandAlt	char	0-5	(Use mouse events for knob)	NGX_EVT_OH_PRESS_LAND_ALT_SET NGX_EVT_OH_PRESS_LAND_ALT_KNOB
NGX_AIR_OutflowValveSwitch	number		0: close 1: neutral 2: manual	NGX_EVT_OH_PRESS_VALVE_SWITCH
NGX_AIR_PressurizationModeSelector	number		0: auto, 1: altn 2: man	NGX_EVT_OH_PRESS_SELECTOR
Forward Overhead – Bottom overhead				
NGX_LTS_LandingLtRetractableSw (on some models)	number	0-1	0: retract 1: extend, 2: on	NGX_EVT_OH_LIGHTS_***L***_RETRACT (also R) NGX_EVT_OH_LAND_LIGHTS_GUARD
NGX_LTS_LandingLtFixedSw	bool	0-1		NGX_EVT_OH_LIGHTS_***L***_FIXED (also R)
			(all landing lights)	NGX_EVT_LDG_LIGHTS_TOGGLE
NGX_LTS_RunwayTurnoffSw	bool	0-1		NGX_EVT_OH_LIGHTS_***L***_TURNOFF (also R) NGX_EVT_TURNOFF_LIGHTS_TOGGLE
NGX_LTS_TaxiSw	bool			NGX_EVT_OH_LIGHTS_TAXI
NGX_APU_Selector	number		0: off, 1: on,2: start	NGX_EVT_OH_LIGHTS_APU_START
NGX_ENG_StartSelector	number	0-1	0: grd, 1: off 2: cont, 3: flt	NGX_EVT_OH_LIGHTS_***L***_ENGINE_START (also R)
NGX_ENG_IgnitionSelector	number		0: ign L, 1: both 2: ign R	NGX_EVT_OH_LIGHTS_IGN_SEL
NGX_LTS_LogoSw	bool			NGX_EVT_OH_LIGHTS_LOGO NGXu_EVT_LOGO_LIGHTS_TOGGLE

NGX_LTS_PositionSw	number		0: steady, 1: off 2: strobe & steady	NGX_EVT_OH_LIGHTS_POS_STROBE
NGX_LTS_AntiCollisionSw	bool			NGX_EVT_OH_LIGHTS_ANT_COL
NGX_LTS_WingSw	bool			NGX_EVT_OH_LIGHTS_WING
NGX_LTS_WheelWellSw	bool			NGX_EVT_OH_LIGHTS_WHEEL_WELL
				NGX_EVT_OH_LIGHTS_COMPASS
Overhead – Miscellaneous				
				NGX_EVT_OH_BLEED_OVHT_TEST_BUTTON NGX_EVT_OH_CAB_ALT_HORN_CUTOUT_BUTTON NGX_EVT_OH_GRND_CALL_SWITCH NGX_EVT_OH_ICE_TAT_TEST NGX_EVT_OH_ICE_WINDOW_HEAT_TEST NGX_EVT_OH_LE_DEVICES_TEST_SWITCH NGX_EVT_OH_WARNING_TEST_MACH_IAS_1_PUSH NGX_EVT_OH_WARNING_TEST_MACH_IAS_2_PUSH NGX_EVT_OH_WARNING_TEST_STALL_1_PUSH NGX_EVT_OH_WARNING_TEST_STALL_2_PUSH NGX_EVT_OH_WING_BODY_OVERHEAT_TEST_PUSH NGXu_EVT_OH_CARGO_SMOKE NGXu_EVT_OH_CARGO_SMOKE_BELL_CUTOUT NGXu_EVT_OH_CARGO_SMOKE_GUARD NGXu_EVT_OH_CARGO_SMOKE_TEST
Glareshield – Warnings				
NGX_WARN_annunFIRE_WARN	bool	0-1		NGX_EVT_FIRE_WARN_LIGHT_***LEFT*** (also RIGHT)
NGX_WARN_annunMASTER_CAUTION	bool	0-1		NGX_EVT_MASTER_CAUTION_LIGHT_***LEFT*** (also RIGHT)
NGX_WARN_annunFLT_CONT	bool			NGX_EVT_SYSTEM_ANNUNCIATOR_PANEL_***LEFT*** (also RIGHT)
NGX_WARN_annunIRS	bool			
NGX_WARN_annunFUEL	bool			
NGX_WARN_annunELEC	bool			
NGX_WARN_annunAPU	bool			
NGX_WARN_annunOVHT_DET	bool			
NGX_WARN_annunANTI_ICE	bool			
NGX_WARN_annunHYD	bool			
NGX_WARN_annunDOORS	bool			
NGX_WARN_annunENG	bool			
NGX_WARN_annunOVERHEAD	bool			
NGX_WARN_annunAIR_COND	bool			

Glareshield – EFIS Control Panels				
NGX_EFIS_MinsSelBARO	bool	0-1		NGX_EVT_EFIS_***CPT***_MINIMUMS_RADIO_BARO (also FO)
NGX_EFIS_BaroSelHPA	bool	0-1		NGX_EVT_EFIS_***CPT***_BARO_IN_HPA (also FO)
NGX_EFIS_VORADFSel1	number	0-1	0: VOR, 1: off, 2: ADF	NGX_EVT_EFIS_***CPT***_VOR_ADF_SELECTOR_L (also FO)
NGX_EFIS_VORADFSel2	number	0-1		NGX_EVT_EFIS_***CPT***_VOR_ADF_SELECTOR_R (also FO)
NGX_EFIS_ModeSel	number	0-1	0: APP, 1: VOR, 2: MAP, 3: PLAN	NGX_EVT_EFIS_***CPT***_MODE (also FO)
NGX_EFIS_RangeSel	number	0-1	0-5	NGX_EVT_EFIS_***CPT***_RANGE (also FO)
				NGX_EVT_EFIS_***CPT***_ARPT (also FO) NGX_EVT_EFIS_***CPT***_BARO NGX_EVT_EFIS_***CPT***_BARO_STD NGX_EVT_EFIS_***CPT***_DATA NGX_EVT_EFIS_***CPT***_FPV NGX_EVT_EFIS_***CPT***_MINIMUMS NGX_EVT_EFIS_***CPT***_MINIMUMS_RST NGX_EVT_EFIS_***CPT***_MODE_CTR NGX_EVT_EFIS_***CPT***_MTRS NGX_EVT_EFIS_***CPT***_POS NGX_EVT_EFIS_***CPT***_RANGE_TFC NGX_EVT_EFIS_***CPT***_STA NGX_EVT_EFIS_***CPT***_TERR NGX_EVT_EFIS_***CPT***_WPT NGX_EVT_EFIS_***CPT***_WXR NGX_EVT_OH_EFIS_HDG_REF_TOGGLE
Glareshield – Mode Control Panel				
NGX_MCP_Course	number	0-1	(Use mouse events for selectors)	NGX_EVT_MCP_COURSE_SELECTOR_***L*** (also R) NGX_EVT_MCP_CRS_***L***_SET
NGX_MCP_IASMach	number		Mach if < 10.0 (Use mouse events for selectors)	NGX_EVT_MCP_SPEED_SELECTOR NGX_EVT_MCP_IAS_SET NGX_EVT_MCP_MACH_SET NGX_EVT_MCP_CO_SWITCH NGX_EVT_MCP_SPD_INTV_SWITCH
NGX_MCP_IASBlank	bool			
NGX_MCP_IASOverspeedFlash	bool			
NGX_MCP_IASUnderspeedFlash	bool			
NGX_MCP_Heading	number		(Use mouse events	NGX_EVT_MCP_HEADING_SELECTOR

			for selectors)	NGX_EVT_MCP_HDG_SET
NGX_MCP_Altitude	number		(Use mouse events for selectors)	NGX_EVT_MCP_ALTITUDE_SELECTOR NGX_EVT_MCP_ALT_SET NGX_EVT_MCP_ALT_INTV_SWITCH
NGX_MCP_VertSpeed	number		(Use mouse events for selectors)	NGX_EVT_MCP_VS_SELECTOR NGX_EVT_MCP_VS_SET
NGX_MCP_VertSpeedBlank	bool			
NGX_MCP_FDSw	bool	0-1		NGX_EVT_MCP_FD_SWITCH_***L*** (also R)
NGX_MCP_ATArmSw	bool			NGX_EVT_MCP_AT_ARM_SWITCH NGX_EVT_CONTROL_STAND_AT***1***_DISENGAGE_SWITCH (also 2)
NGX_MCP_BankLimitSel	number		0-4	NGX_EVT_MCP_BANK_ANGLE_SELECTOR
NGX_MCP_DisengageBar	bool			NGX_EVT_MCP_DISENGAGE_BAR
NGX_MCP_annunFD	bool	0-1		
NGX_MCP_annunATArm	bool			
NGX_MCP_annunN1	bool			
NGX_MCP_annunSPEED	bool			NGX_EVT_MCP_SPEED_SWITCH
NGX_MCP_annunVNAV	bool			NGX_EVT_MCP_VNAV_SWITCH
NGX_MCP_annunLVL_CHG	bool			NGX_EVT_MCP_LVL_CHG_SWITCH
NGX_MCP_annunHDG_SEL	bool			NGX_EVT_MCP_HDG_SEL_SWITCH
NGX_MCP_annunLNAV	bool			NGX_EVT_MCP_LNAV_SWITCH
NGX_MCP_annunVOR_LOC	bool			NGX_EVT_MCP_VOR_LOC_SWITCH
NGX_MCP_annunAPP	bool			NGX_EVT_MCP_APP_SWITCH
NGX_MCP_annunALT_HOLD	bool			NGX_EVT_MCP_ALT_HOLD_SWITCH
NGX_MCP_annunVS	bool			NGX_EVT_MCP_VS_SWITCH
NGX_MCP_annunCMD_A	bool			NGX_EVT_MCP_CMD_A_SWITCH
NGX_MCP_annunCWS_A	bool			NGX_EVT_MCP_CWS_A_SWITCH
NGX_MCP_annunCMD_B	bool			NGX_EVT_MCP_CMD_B_SWITCH
NGX_MCP_annunCWS_B	bool			NGX_EVT_MCP_CWS_B_SWITCH
Forward Panel – Clock				
				NGX_EVT_CHRONO_***L***_CHR (also R) NGX_EVT_CHRONO_***L***_ET NGX_EVT_CHRONO_***L***_MINUS NGX_EVT_CHRONO_***L***_PLUS NGX_EVT_CHRONO_***L***_RESET NGX_EVT_CHRONO_***L***_SET NGX_EVT_CHRONO_***L***_TIME_DATE

				NGX_EVT_CLOCK_***L***
Forward Panel – Miscellaneous				
NGX_MAIN_NoseWheelSteeringSwNORM	bool		False: ALT	NGX_EVT_NOSE_WHEEL_STEERING_SWITCH_GUARD NGX_EVT_NOSE_WHEEL_STEERING_SWITCH
NGX_MAIN_annunBELOW_GS	bool	0-1		
NGX_MAIN_MainPanelDUSel	number	0-1	0-4, FO seq reversed	NGX_EVT_DSP_***CPT***_MAIN_DU_SELECTOR (also FO)
NGX_MAIN_LowerDUSel	number	0-1	0-2, FO seq reversed	NGX_EVT_DSP_***CPT***_LOWER_DU_SELECTOR (also FO)
NGX_MAIN_annunAP	bool	0-1		NGX_EVT_DSP_***CPT***_AP_RESET_SWITCH (also FO)
NGX_MAIN_annunAT	bool	0-1		NGX_EVT_DSP_***CPT***_AT_RESET_SWITCH (also FO)
NGX_MAIN_annunFMC	bool	0-1		NGX_EVT_DSP_***CPT***_FMC_RESET_SWITCH (also FO)
NGX_MAIN_DisengageTestSelector	number	0-1	0: 1, 1: off, 2: 2	NGX_EVT_DSP_***CPT***_DISENGAGE_TEST_SWITCH (also FO)
NGX_MAIN_annunSPEEDBRAKE_ARMED	bool			
NGX_MAIN_annunSPEEDBRAKE_DO_NOT_ARM	bool			
NGX_MAIN_annunSPEEDBRAKE_EXTENDED	bool			
NGX_MAIN_annunSTAB_OUT_OF_TRIM	bool			
NGX_MAIN_LightsSelector	number		0: TEST, 1: BRT 2: DIM	NGX_EVT_DSP_CPT_MASTER_LIGHTS_SWITCH
NGX_MAIN_RMISelector1_VOR	bool			NGX_EVT_RMI_LEFT_SELECTOR
NGX_MAIN_RMISelector2_VOR	bool			NGX_EVT_RMI_RIGHT_SELECTOR
NGX_MAIN_N1SetSelector	number		0: 2, 1: 1, 2: AUTO 3: BOTH	NGX_EVT_MPM_N1SET_SELECTOR NGX_EVT_MPM_N1SET_CONTROL
NGX_MAIN_SpdRefSelector	number		0: SET, 1: AUT 2: V1, 3: VR 4: WT, 5: VREF 6: Bug	NGX_EVT_MPM_SPEED_REFERENCE_SELECTOR NGX_EVT_MPM_SPEED_REFERENCE_CONTROL
NGX_MAIN_FuelFlowSelector	number		0: RESET, 1: RATE 2: USED	NGX_EVT_MPM_FUEL_FLOW_SWITCH
NGX_MAIN_AutobrakeSelector	number		0: RTO, 1-5	NGX_EVT_MPM_AUTOBRAKE_SELECTOR NGXu_EVT_MPM_AUTOBRAKE_SELECTOR_INOUT
NGX_MAIN_annunANTI_SKID_INOP	bool			
NGX_MAIN_annunAUTO_BRAKE_DISARM	bool			
NGX_MAIN_annunLE_FLAPS_TRANSIT	bool			
NGX_MAIN_annunLE_FLAPS_EXT	bool			

NGX_MAIN_TEFlapsNeedle	number	0-1	 0: 0 1: 1 2: 2 3: 5 4: 10 5: 15 6: 25 7: 30 8: 40	NGX_EVT_CONTROL_STAND_FLAPS_LEVER NGX_EVT_CONTROL_STAND_FLAPS_LEVER_0 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_1 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_2 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_5 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_10 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_15 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_25 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_30 NGX_EVT_CONTROL_STAND_FLAPS_LEVER_40
NGX_MAIN_annunGEAR_transit	bool	0-2		
NGX_MAIN_annunGEAR_locked	bool	0-2		
NGX_MAIN_GearLever	number		0: UP, 1: OFF 2: DOWN	NGX_EVT_GEAR_LEVER NGX_EVT_GEAR_LEVER_OFF NGX_EVT_GEAR_LEVER_UNLOCK
NGX_MAIN_BrakePressNeedle	number			
				NGX_EVT_MPM_MFD_C_R_BUTTON NGX_EVT_MPM_MFD_ENG_BUTTON NGX_EVT_MPM_MFD_SYS_BUTTON
System Annunciator				
NGX_HGS_annun_AIII	bool			NGX_EVT_SYSTEM_ANNUNCIATOR_PANEL_LEFT NGX_EVT_SYSTEM_ANNUNCIATOR_PANEL_RIGHT
NGX_HGS_annun_NO_AIII	bool			
NGX_HGS_annun_FLARE	bool			
NGX_HGS_annun_RO	bool			
NGX_HGS_annun_RO_CTN	bool			
NGX_HGS_annun_RO_ARM	bool			
NGX_HGS_annun_TO	bool			
NGX_HGS_annun_TO_CTN	bool			
NGX_HGS_annun_APCH	bool			
NGX_HGS_annun_TO_WARN	bool			
NGX_HGS_annun_Bar	bool			
NGX_HGS_annun_FAIL	bool			
Lower Forward Panel				
NGX_LTS_MainPanelKnob	number	0-1	0-150	NGX_EVT_LWRMAIN_***CAPT***_MAIN_PANEL_BRT (also FO)
NGX_LTS_BackgroundKnob	number		0-150	NGX_EVT_LWRMAIN_CAPT_BACKGROUND_BRT
NGX_LTS_AFDSFloodKnob	number		0-150	NGX_EVT_LWRMAIN_CAPT_AFDS_BRT
NGX_LTS_OutbdDUBrtKnob	number	0-1	0-127	NGX_EVT_LWRMAIN_***CAPT***_OUTBD_DU_BRT (also FO)

NGX_LTS_InbdDUBrtKnob	number	0-1	0-127	NGX_EVT_LWRMAIN_***CAPT***_INBD_DU_BRT (also FO)
NGX_LTS_InbdDUMapBrtKnob	number	0-1	0-127	NGX_EVT_LWRMAIN_***CAPT***_INBD_DU_INNER_BRT (also FO)
NGX_LTS_UpperDUBrtKnob	number		0-127	NGX_EVT_LWRMAIN_CAPT_UPPER_DU_BRT
NGX_LTS_LowerDUBrtKnob	number		0-127	NGX_EVT_LWRMAIN_CAPT_LOWER_DU_BRT
NGX_LTS_LowerDUMapBrtKnob	number		0-127	NGX_EVT_LWRMAIN_CAPT_LOWER_DU_INNER_BRT
NGX_GPWS_annunINOP	bool			
NGX_GPWS_FlapInhibitSw_NORM	bool			NGX_EVT_GPWS_FLAP_INHIBIT_GUARD NGX_EVT_GPWS_FLAP_INHIBIT_SWITCH
NGX_GPWS_GearInhibitSw_NORM	bool			NGX_EVT_GPWS_GEAR_INHIBIT_GUARD NGX_EVT_GPWS_GEAR_INHIBIT_SWITCH
NGX_GPWS_TerrInhibitSw_NORM	bool			NGX_EVT_GPWS_TERR_INHIBIT_GUARD NGX_EVT_GPWS_TERR_INHIBIT_SWITCH
Integrated Standby Flight Display and Analog Standby Instruments				
				NGX_EVT_ISFD_APP NGX_EVT_ISFD_ATT_RST NGX_EVT_ISFD_BARO NGX_EVT_ISFD_BARO_PUSH NGX_EVT_ISFD_HP_IN NGX_EVT_ISFD_MINUS NGX_EVT_ISFD_PLUS NGX_EVT_STANDBY_ADI_APPR_MODE NGX_EVT_STANDBY_ADI_CAGE_KNOB NGX_EVT_STANDBY_ALT_BARO_KNOB
Control Stand				
NGX_CDU_annunEXEC	bool	0-1		
NGX_CDU_annunCALL	bool	0-1		
NGX_CDU_annunFAIL	bool	0-1		
NGX_CDU_annunMSG	bool	0-1		
NGX_CDU_annunOFST	bool	0-1		
NGX_CDU_BrtKnob	number	0-1	0-127	NGX_EVT_CDU_L_BRITENESS NGX_EVT_CDU_R_BRITENESS
Control Stand – Flight Controls				
				NGX_EVT_CONTROL_STAND_FWD_THRUST1_LEVER NGX_EVT_CONTROL_STAND_FWD_THRUST2_LEVER
				NGX_EVT_CONTROL_STAND_REV_THRUST1_LEVER NGX_EVT_CONTROL_STAND_REV_THRUST2_LEVER
				NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER

				NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_50PCT NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_ARM NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_DOWN NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_FLT_DET NGX_EVT_CONTROL_STAND_SPEED_BRAKE_LEVER_UP
				NGX_EVT_CONTROL_STAND_TOGA1_SWITCH NGX_EVT_CONTROL_STAND_TOGA2_SWITCH
				NGX_EVT_CONTROL_STAND_TRIM_WHEEL
				NGX_EVT_TILLER NGX_EVT_FCTL_AILERON_TRIM NGX_EVT_FCTL_RUDDER_TRIM NGX_EVT_STAB_TRIM_OVRD_SWITCH NGX_EVT_STAB_TRIM_OVRD_SWITCH_GUARD
				NGX_EVT_YOKE_***L***_AP_DISC_SWITCH (also R) NGX_EVT_YOKE_***L***_COUNTER_1 NGX_EVT_YOKE_***L***_COUNTER_2 NGX_EVT_YOKE_***L***_COUNTER_3
				NGXu_EVT_HIDE_YOKE_***CAPT*** (also FO) NGXu_EVT_***CA***_ARMREST_LEFT_SWITCH (also FO) NGXu_EVT_***CA***_ARMREST_RIGHT_SWITCH (also FO)
NGX_TRIM_StabTrimMainElecSw_NORMAL	bool			NGX_EVT_CONTROL_STAND_STABTRIM_ELEC_SWITCH_GUARD NGX_EVT_CONTROL_STAND_STABTRIM_ELEC_SWITCH
NGX_TRIM_StabTrimAutoPilotSw_NORMAL	bool			NGX_EVT_CONTROL_STAND_STABTRIM_AP_SWITCH_GUARD NGX_EVT_CONTROL_STAND_STABTRIM_AP_SWITCH
NGX_PED_annunParkingBrake	bool			NGX_EVT_CONTROL_STAND_PARK_BRAKE_LEVER
NGX_FIRE_OvhtDetSw	number	0-1	0: A, 1: NORMAL, 2: B	NGX_EVT_FIRE_OVHT_DET_SWITCH_***1*** (also 2)
NGX_FIRE_annunENG_OVERHEAT	bool	0-1		
NGX_FIRE_DetTestSw	number		0: FAULT/INOP 1: neutral 2: OVHT/FIRE	NGX_EVT_FIRE_DETECTION_TEST_SWITCH
NGX_FIRE_HandlePos	number	0-2	0: In 1: Blocked 2: Out 3: Turned left 4: Turned right	NGX_EVT_FIRE_HANDLE_ENGINE_1_TOP NGX_EVT_FIRE_HANDLE_ENGINE_1_BOTTOM NGX_EVT_FIRE_HANDLE_APU_TOP NGX_EVT_FIRE_HANDLE_APU_BOTTOM NGX_EVT_FIRE_HANDLE_ENGINE_2_TOP NGX_EVT_FIRE_HANDLE_ENGINE_2_BOTTOM

NGX_FIRE_HandleIlluminated	bool	0-2		
NGX_FIRE_annunWHEEL_WELL	bool			
NGX_FIRE_annunFAULT	bool			
NGX_FIRE_annunAPU_DET_INOP	bool			
NGX_FIRE_annunAPU_BOTTLE_DISCHARGE	bool			
NGX_FIRE_annunBOTTLE_DISCHARGE	bool	0-1		
NGX_FIRE_ExtinguisherTestSw	number		0: 1, 1: neutral 2: 2	NGX_EVT_FIRE_EXTINGUISHER_TEST_SWITCH
NGX_FIRE_annunExtinguisherTest	bool	0-2	Left, Right, APU	
NGX_CARGO_annunExtTest	bool	0-1	Fwd, Aft	
NGX_CARGO_DetSelect	number	0-1	0: A, 1: ORM, 2: B	NGX_EVT_CARGO_FIRE_DET_SEL_SWITCH_***FWD*** (also AFT)
NGX_CARGO_ArmedSw	bool	0-1		NGX_EVT_CARGO_FIRE_ARM_SWITCH_***FWD*** (also AFT)
NGX_CARGO_annunFWD	bool			
NGX_CARGO_annunAFT	bool			
NGX_CARGO_annunDETECTOR_FAULT	bool			
NGX_CARGO_annunDISCH	bool			
				NGX_EVT_CARGO_FIRE_DISC_SWITCH NGX_EVT_CARGO_FIRE_DISC_SWITCH_GUARD NGX_EVT_CARGO_FIRE_TEST_SWITCH
Control Stand – HGS				
NGX_HGS_annunRWY	bool			
NGX_HGS_annunGS	bool			
NGX_HGS_annunFAULT	bool			
NGX_HGS_annunCLR	bool			
Control Stand – Weather Radar Panel				
				NGX_EVT_WXR_AUTO NGX_EVT_WXR_TEST NGX_EVT_WXR_***L***_GAIN_CONTROL (also R) NGX_EVT_WXR_***L***_GC NGX_EVT_WXR_***L***_MAP NGX_EVT_WXR_***L***_TFR NGX_EVT_WXR_***L***_TILT_CONTROL NGX_EVT_WXR_***L***_WX NGX_EVT_WXR_***L***_WX_T
Control Stand – COM radio panel				
				NGX_EVT_***COM1***_AM_SWITCH (also COM2 and COM3)

				NGX_EVT_***COM1***_HF_SENSOR_KNOB NGX_EVT_***COM1***_HF1_SWITCH NGX_EVT_***COM1***_HF2_SWITCH NGX_EVT_***COM1***_INNER_SELECTOR NGX_EVT_***COM1***_OUTER_SELECTOR NGX_EVT_***COM1***_PNL_OFF_SWITCH NGX_EVT_***COM1***_TEST_SWICTH NGX_EVT_***COM1***_TRANSFER_SWITCH NGX_EVT_***COM1***_VHF1_SWITCH NGX_EVT_***COM1***_VHF2_SWITCH NGX_EVT_***COM1***_VHF3_SWITCH
Control Stand – NAV panel				
				NGX_EVT_***NAV1***_INNER_SELECTOR (also NAV2) NGX_EVT_***NAV1***_OUTER_SELECTOR NGX_EVT_***NAV1***_TEST_SWICTH NGX_EVT_***NAV1***_TRANSFER_SWITCH
Control Stand – ADF panel				
				NGX_EVT_ADF_INNER_SELECTOR NGX_EVT_ADF_MIDDLE_SELECTOR NGX_EVT_ADF_MODE_SELECTOR NGX_EVT_ADF_OUTER_SELECTOR NGX_EVT_ADF_TONE_SWITCH NGX_EVT_ADF_TRANSFER_SWITCH
Control Stand – TCAS Panel				
NGX_XPDR_XpndrSelector_2	bool		False: 1, True: 2	NGX_EVT_TCAS_XPNDR
NGX_XPDR_AltSourceSel_2	bool		False: 1, True: 2	NGX_EVT_TCAS_ALTSOURCE
NGX_XPDR_ModeSel	number		0-4	NGX_EVT_TCAS_MODE
NGX_XPDR_annunFAIL	bool			
				NGX_EVT_TCAS_IDENT NGX_EVT_TCAS_KNOB1 NGX_EVT_TCAS_KNOB2 NGX_EVT_TCAS_KNOB3 NGX_EVT_TCAS_KNOB4 NGX_EVT_TCAS_TEST
Control Stand – Miscellaneous				
NGX_LTS_PedFloodKnob	number		0-150	
NGX_LTS_PedPanelKnob	number		0-150	

NGX_TRIM_StabTrimSw_NORMAL	bool			
NGX_PED_annunLOCK_FAIL	bool			
NGX_PED_annunAUTO_UNLK	bool			
NGX_PED_FltDkDoorSel	number			NGX_EVT_FLT_DK_DOOR_KNOB
Extra variables				
NGX_ENG_StartValve	bool	0-1	True: open	NGX_EVT_CONTROL_STAND_ENG1_START_LEVER NGX_EVT_CONTROL_STAND_ENG2_START_LEVER
NGX_AIR_DuctPress	number	0-1	PSI	
NGX_COMM_Attend_PressCount	number		Counter	
NGX_COMM_GrdCall_PressCount	number		Counter	
NGX_COMM_SelectedMic	number	0: CAPT 1: FO 2: OBS (Events for each)	0: VHF1 1: VFH2 2: VHF3 3: HF1 4: HF2 5: FLT 6: SVC 7: PA	NGX_EVT_ACP_***CAPT***_MIC_VHF1 (also FO and OBS) NGX_EVT_ACP_***CAPT***_MIC_VHF2 NGX_EVT_ACP_***CAPT***_MIC_VHF3 NGX_EVT_ACP_***CAPT***_MIC_HF1 NGX_EVT_ACP_***CAPT***_MIC_HF2 NGX_EVT_ACP_***CAPT***_MIC_FLT NGX_EVT_ACP_***CAPT***_MIC_SVC NGX_EVT_ACP_***CAPT***_MIC_PA
NGX_FUEL_QtyCenter	number		Lbs	
NGX_FUEL_QtyLeft	number		Lbs	
NGX_FUEL_QtyRight	number		Lbs	
NGX_IRS_aligned	bool		True: at least 1 is aligned	
Information from settings and FMS				
NGX_AircraftModel	number		1: -600, 2: -700, 3: -700 BW, 4: -700 SSW, 5: -800, 6: -800 BW, 7: -800 SSW 8: -900, 9: -900 BW, 10: -900 SSW, 11: -900ER BW, 12: -900ER SSW	
NGX_WeightInKg	bool		True: KG, False: LBS	
NGX_GPWS_V1CallEnabled	bool			
NGX_GroundConnAvailable	bool			
NGX_FMC_TakeoffFlaps	number		0 if not set	
NGX_FMC_V1	number		0 if not set	
NGX_FMC_VR	number		0 if not set	
NGX_FMC_V2	number		0 if not set	
NGX_FMC_LandingFlaps	number		0 if not set	
NGX_FMC_LandingVREF	number		0 if not set	
NGX_FMC_CruiseAlt	number		0 if not set	

NGX_FMC_LandingAltitude	number		-32767 if not available	
NGX_FMC_TransitionAlt	number			
NGX_FMC_TransitionLevel	number			
NGX_FMC_PerfInputComplete	bool			
NGX_FMC_DistanceToTOD	number		0.0 if passed, <0 if not available	
NGX_FMC_DistanceToDest	number		<0 if not available	
NGX_FMC_flightNumber	char	0-8		
			New since SP2	
NGX_COMM_ReceiverSwitches	number	0-2 0: CAPT 1: FO 2: OVH	Bit 0: VHF1 Bit 1: VHF1 Bit 2: VHF3 Bit 3: HF1 Bit 4: HF2 Bit 5: FLT Bit 6: SVC Bit 7: PA Bit 8: NAV1 Bit 9: NAV2 Bit 10: ADF1 Bit 11: ADF2 Bit 12: MKR Bit 13: SPKR	NGX_EVT_ACP_***CAPT***_REC_VHF1 (also FO and OBS) NGX_EVT_ACP_***CAPT***_REC_VHF2 NGX_EVT_ACP_***CAPT***_REC_VHF3 NGX_EVT_ACP_***CAPT***_REC_HF1 NGX_EVT_ACP_***CAPT***_REC_HF2 NGX_EVT_ACP_***CAPT***_REC_FLT NGX_EVT_ACP_***CAPT***_REC_SVC NGX_EVT_ACP_***CAPT***_REC_PA NGX_EVT_ACP_***CAPT***_REC_NAV1 NGX_EVT_ACP_***CAPT***_REC_NAV2 NGX_EVT_ACP_***CAPT***_REC_ADF1 NGX_EVT_ACP_***CAPT***_REC_ADF2 NGX_EVT_ACP_***CAPT***_REC_MKR NGX_EVT_ACP_***CAPT***_REC_SPKR
				NGX_EVT_ACP_***CAPT***_ALT_NORM_SWITCH (also FO and OBS) NGX_EVT_ACP_***CAPT***_FILTER_SWITCH NGX_EVT_ACP_***CAPT***_MASK_BOOM_SWITCH NGX_EVT_ACP_***CAPT***_RT_IC_SWITCH
				NGX_EVT_ACP_***FO***_VOL_ADF1 (also OBS, no CAPT!) NGX_EVT_ACP_***FO***_VOL_ADF2 NGX_EVT_ACP_***FO***_VOL_MKR NGX_EVT_ACP_***FO***_VOL_NAV1 NGX_EVT_ACP_***FO***_VOL_NAV2
NGX_MAIN_annunAP_Amber	bool	0-1		
NGX_MAIN_annunAT_Amber	bool	0-1		
NGX_ICE_WindowHeatTestSw	number		0: OVHT 1: Neutral 2: PWR TEST	
NGX_DOOR_annunFWD_ENTRY	bool			NGX_EVT_DOOR_FWD_L

NGX_DOOR_annunFWD_SERVICE	bool			NGX_EVT_DOOR_FWD_R
NGX_DOOR_annunAIRSTAIR	bool			NGX_EVT_DOOR_AIRSTAIR
NGX_DOOR_annunLEFT_FWD_OVERWING	bool			NGX_EVT_DOOR_OVERWING_EXIT_L
NGX_DOOR_annunRIGHT_FWD_OVERWING	bool			NGX_EVT_DOOR_OVERWING_EXIT_R
NGX_DOOR_annunFWD_CARGO	bool			NGX_EVT_DOOR_CARGO_FWD
NGX_DOOR_annunEQUIP	bool			NGX_EVT_DOOR_EQUIPMENT_HATCH
NGX_DOOR_annunLEFT_AFT_OVERWING	bool			
NGX_DOOR_annunRIGHT_AFT_OVERWING	bool			
NGX_DOOR_annunAFT_CARGO	bool			NGX_EVT_DOOR_CARGO_AFT
NGX_DOOR_annunAFT_ENTRY	bool			NGX_EVT_DOOR_AFT_L
NGX_DOOR_annunAFT_SERVICE	bool			NGX_EVT_DOOR_AFT_R
NGX_AIR_annunAUTO_FAIL	bool			
NGX_AIR_annunOFFSCHED_DESCENT	bool			
NGX_AIR_annunALTN	bool			
NGX_AIR_annunMANUAL	bool			
NGX_AIR_CabinAltNeedle	number			
NGX_AIR_CabinDPNeedle	number			
NGX_AIR_CabinVSNeedle	number			
NGX_AIR_CabinValveNeedle	number		0.0-1.0	
NGX_AIR_TemperatureNeedle	number		Degrees C	
NGX_AIR_DuctPressNeedle	number	0-1	Degrees C	
NGX_ELEC_MeterDisplayTop	char	0-12	Top line, 3 x 4 digits/symbols, zero terminated	
NGX_ELEC_MeterDisplayBottom	char	0-12	Bottom line, zero terminated	
NGX_IRS_DisplayLeft	char	0-6	Left display, zero terminated	
NGX_IRS_DisplayRight	char	0-7	Right display, zero terminated	
NGX_IRS_DisplayShowsDots	bool			
NGXu_FUEL_AuxFwd	bool	0-1		
NGXu_FUEL_AuxAft	bool	0-1		
NGXu_FUEL_FWDBleed	bool			
NGXu_FUEL_AFTBleed	bool			
NGXu_FUEL_GNDXfr	bool			
NGX_FUEL_annunENG_VALVE_CLOSED	number	0-1		
NGX_FUEL_annunSPAR_VALVE_CLOSED	number	0-1		
NGXu_ADF_StandbyFrequency	number		Freq times 10	
Auto-Flight System indicators				
NGXu_AFS_AutothrottleServosConnected	bool			

NGXu_AFS_ControlsPitch	bool			
NGXu_AFS_ControlsRoll	bool			
Electronics				
NGXu_ELEC_BusPowered	bool	0-15		
NGXu_MCP_indication_powered	bool			
New for NGXu				
NGXu_MAIN_annunCABIN_ALTITUDE	bool			
NGXu_MAIN_annunTAKEOFF_CONFIG	bool			
NGXu_CVR_annunTEST	bool			

Some There are several events that couldn't easily fit somewhere in the table above:

Condition variable	Unit	Index	Values	Event
Overhead – Cockpit Voice Recorder				
				NGX_EVT_OH_CVR_ERASE NGX_EVT_OH_CVR_TEST
Overhead – Cockpit Lights				
				NGX_EVT_COCKPIT_LIGHTS_OFF NGX_EVT_COCKPIT_LIGHTS_ON NGX_EVT_COCKPIT_LIGHTS_TOGGLE
Main Panel				
Tests, Alarms, Alarm cutouts, and Fires				
				NGX_EVT_CONTROL_STAND_HORN_CUTOUT_SWITCH NGX_EVT_FIRE_BELL_CUTOUT_SWITCH NGX_EVT_FIRE_UNLOCK_SWITCH_APU NGX_EVT_FIRE_UNLOCK_SWITCH_ENGINE_1 NGX_EVT_FIRE_UNLOCK_SWITCH_ENGINE_2 NGX_EVT_HGS_FAIL_SWITCH NGX_EVT_GPWS_SYS_TEST_BTN
Miscellaneous				
				NGX_EVT_CTRL_ACCELERATION_DISABLE NGX_EVT_CTRL_ACCELERATION_ENABLE NGX_EVT_DSP_CPT_BELOW_GS_INHIBIT_SWITCH NGX_EVT_DSP_FO_BELOW_GS_INHIBIT_SWITCH NGX_EVT_PED_FLOOD_CONTROL NGX_EVT_PED_PANEL_CONTROL NGX_EVT_SELCAL_HF1_SWITCH NGX_EVT_SELCAL_HF2_SWITCH

				NGX_EVT_SELCAL_VHF1_SWITCH NGX_EVT_SELCAL_VHF2_SWITCH NGX_EVT_SELCAL_VHF3_SWITCH NGX_EVT_WLAN_GUARD NGX_EVT_WLAN_SWITCH NGXu_EVT_ALT_GEAR_EXT_DOOR NGXu_EVT_ALT_GEAR_EXT_HANDLE_LEFT NGXu_EVT_ALT_GEAR_EXT_HANDLE_NOSE NGXu_EVT_ALT_GEAR_EXT_HANDLE_RIGHT NGXu_EVT_CHART_BRT_L NGXu_EVT_CHART_BRT_R NGXu_EVT_COMBINER_COVER NGXu_EVT_GRIMES_LIGHT_CA NGXu_EVT_HGS_EYEPOINT NGXu_EVT_JUMPSEAT_STOW_EXTEND NGXu_EVT_MAP_BRT_L NGXu_EVT_MAP_BRT_L_PUSHPULL NGXu_EVT_MAP_BRT_R NGXu_EVT_MAP_BRT_R_PUSHPULL NGXu_EVT_MIC_L NGXu_EVT_MIC_R NGXu_EVT_NON_HGS_EYEPOINT NGXu_EVT_OH_LAV_SMOKE_BELL_CUTOUT NGXu_EVT_OH_LAV_SMOKE_TEST NGXu_EVT_OH_LIGHTS_L_ENGINE_START_INOUT NGXu_EVT_OH_LIGHTS_R_ENGINE_START_INOUT NGXu_EVT_RELEASE_PUSHBACK_TUG NGXu_EVT_SPOTLIGHT_L NGXu_EVT_SPOTLIGHT_OBS NGXu_EVT_SPOTLIGHT_R

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 737-NGXu

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 737 NGXu 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 “AS_PMDG_737NGX_EVT_OH_LAND_LIGHTS_GUARD”, and value 2147483648, will do the same as clicking with the right mouse button on the landing light guard, which is the bar some 737 NG models have next to the landing lights. This guard has no state (there is no L-var for it in the left-most column), so using mouse events are the only option.

Using Data Events

The Honeycomb Alpha Yoke has a rotary knob for the magnetos on GA aircraft. On the 737 NG we do not have those, but the 737 NGXu profile provided by Aerosoft has a nice alternative use: the autobrake knob. In the L-var table you can find “NGX_MAIN_AutobrakeSelector”, where it states that the values are 0 for “RTO” (Rejected Take-Off), and 1 to 5 for the individual settings. (“OFF”, 1, 2, 3, and “MAX”) So the profile sends type “L”, name “AS_PMDG_737NGX_EVT_MPM_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 737 NGXu 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_737NGX_LTS_LandingLtFixedSw", 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_737NGX_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 NGXu 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_737" 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_737NGX_MCP_annunHDG_SEL, bool =1	
FCU-NAV	NAV	1, 1	“Nav hold” mode is on, in the 737 profile “VOR LOC”.	L:AS_PMDG_737NGX_MCP_annunVOR_LOC, bool =1	
FCU-APR	APR	1, 2	“Approach” mode is on.	L:AS_PMDG_737NGX_MCP_annunAPP, bool =1	
FCU-REV	REV	1, 3	“Back course hold” is on, in the 737 profile “LNAV”.	L:AS_PMDG_737NGX_MCP_annunLNAV, bool =1	
FCU-ALT	ALT	1, 4	“Altitude hold” mode is on, in the 737 profile “VNAV”.	L:AS_PMDG_737NGX_MCP_annunVNAV, bool =1	
FCU-VS	VS	1, 5	“Vertical speed” mode is on, in the 737 profile “LVL CHG”.	L:AS_PMDG_737NGX_MCP_annunLVL_CHG, bool =1	
FCU-IAS	IAS	1, 6	“IAS hold” or “Speed hold” mode is on.	L:AS_PMDG_737NGX_MCP_annunSPEED, bool =1	
GCU-AP	AUTO PILOT	1, 7	Autopilot is switched on, on the 737 profile this is either “CMD A” or “CMD B”.	L:AS_PMDG_737NGX_MCP_annunCMD_A, bool =1 L:AS_PMDG_737NGX_MCP_annunCMD_B, bool =1	OR

Landing Gear Status Lights

The landing gear lights can be red or green, or switched off.

Name	**Byte, Bit**	**Remarks**	**Programming**
LDG-L_GREEN	2, 0	Left main gear is UP.	L:AS_PMDG_737NGX_MAIN_annunGEAR_locked, 0, bool =1
LDG-L_RED	2, 1	Left main gear is IN TRANSIT.	L:AS_PMDG_737NGX_MAIN_annunGEAR_transit, 0, bool =1
LDG-N_GREEN	2, 2	Nose wheel is UP.	L:AS_PMDG_737NGX_MAIN_annunGEAR_locked, 1, bool =1
LDG-N_RED	2, 3	Nose is IN TRANSIT.	L:AS_PMDG_737NGX_MAIN_annunGEAR_transit, 1, bool =1
LDG-R_GREEN	2, 4	Right main gear is UP.	L:AS_PMDG_737NGX_MAIN_annunGEAR_locked, 2, bool =1
LDG-R_RED	2, 5	Right main gear is IN TRANSIT.	L:AS_PMDG_737NGX_MAIN_annunGEAR_transit, 2, bool =1

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	In the 737 profile this is interpreted as a fire warning in either engine.	L:AS_PMDG_737NGX_WARN_annunFIRE_WARN, 0, bool =1 L:AS_PMDG_737NGX_WARN_annunFIRE_WARN, 1, bool =1	OR
ANC-ENG_FIRE	ENGINE FIRE	2, 7	The overheat annunciator.	L:AS_PMDG_737NGX_WARN_annunOVHT_DET, bool =1	
ANC-OIL	LOW OIL PRESSURE	3, 0	The engine annunciator.	L:AS_PMDG_737NGX_WARN_annunENG, bool =1	
ANC-FUEL	LOW FUEL PRESSURE	3, 1	The fuel annunciator.	L:AS_PMDG_737NGX_WARN_annunFUEL, bool =1	
ANC-ANTI_ICE	ANTI ICE	3, 2	The anti-ice annunciator.	L:AS_PMDG_737NGX_WARN_annunANTI_ICE, bool =1	
ANC-STARTER	STARTER ENGAGED	3, 3	The overhead annunciator.	L:AS_PMDG_737NGX_WARN_annunOVERHEAD, bool =1	
ANC-APU	APU	3, 4	The APU annunciator.	L:AS_PMDG_737NGX_WARN_annunAPU, bool =1	
ANC-MSTR_CTN	MASTER CAUTION	3, 5	Master caution annunciator.	L:AS_PMDG_737NGX_WARN_annunMASTER_CAUTION, 0, bool =1 L:AS_PMDG_737NGX_WARN_annunMASTER_CAUTION, 1, bool =1	OR
ANC-VACUUM	VACUUM	3, 6	The air conditioning annunciator.	L:AS_PMDG_737NGX_WARN_annunAIR_COND, bool =1	
ANC-HYD	LOW HYD PRESSURE	3, 7	The hydraulics annunciator.	L:AS_PMDG_737NGX_WARN_annunHYD, bool =1	
ANC-AUX-FUEL	AUX FUEL PUMP	4, 0	(unused)		
ANC-PRK_BRK	PARKING BRAKE	4, 1	The parking brakes are set.	L:AS_PMDG_737NGX_PED_annunParkingBrake, bool =1	
ANC-VOLTS	LOW VOLTS	4, 2	The electrical systems annunciator.	L:AS_PMDG_737NGX_WARN_annunELEC, bool =1	
ANC-DOOR	DOOR	4, 3	The doors annunciator.	L:AS_PMDG_737NGX_WARN_annunDOORS, bool =1	