

		FADEC	
Mode	Command	Comment	Type
ENGINE 0	FADEC_>FADEC_engines[0].PL_in_dgr	Degrees are all 0 to 100. This will be copied from the "engine controls"	DOUBLE
	FADEC_>FADEC_engines[0].CL_in_pos	0 -> OFF 1 -> START&FEATHER 2 -> MIN 3 -> 900RPM 4 -> MAX	BYTE
	FADEC_>FADEC_engines[0].sensors		
	FADEC_>FADEC_engines[0].out		
	FADEC_>FADEC_engines[0].int_system_mode	0 -> Power OFF 1 -> Feather 2 -> Reverse 3 -> Beta Range 4 -> Flight Range	BYTE
	FADEC_>FADEC_engines[0].Nh_system_mode	0 -> Minimum Power 1 -> Prop RPM by Nh AUTO 2 -> Torque AUTO	BYTE
	FADEC_>FADEC_engines[0].prop_system_mode	0 -> Maintain Pitch 1 -> Constant Speed	BYTE
	FADEC_>FADEC_engines[0].trq_bug		DOUBLE
	FADEC_>FADEC_engines[0].engine_rating_procent		SIGN_CHAR
	FADEC_>FADEC_engines[0].engine_rating_mode	0 -> No Rating 1 -> MTOP 2 -> MCP 3 -> NTOP 4 -> RDC TOP 5 -> MCL 6 -> MCR	BYTE
	FADEC_>FADEC_engines[0].engine_rating_mode_override_requested	0 -> Nothing 1 -> MTOP 3 -> NTOP (reset) 4 -> RDC TOP 5 -> MCL 6 -> MCR 7 -> RDC NP	BYTE
	FADEC_>FADEC_engines[0].Nh_USP	Nh underspeed protection	BOOLEAN
	FADEC_>FADEC_engines[0].Np_USP	Np underspeed protection	BOOLEAN
	FADEC_>FADEC_engines[0].bleed_mode	0 -> OFF 1 -> MIN 2 -> NORM 3 -> MAX	BYTE
	FADEC_>FADEC_engines[0].uptrim	Set by PCU	BOOLEAN
FADEC_>FADEC_engines[0].limits			
FADEC_>FADEC_engines[0].FADEC_electrical_power_available		BOOLEAN	
FADEC_>FADEC_engines[0].FADEC_starter_power_available		BOOLEAN	
FADEC_>FADEC_engines[0].controls.throttle	-0xFFFF TO 0xFFFF. Will be fed to the calibration function	INT	
FADEC_>FADEC_engines[0].controls.prop	-0xFFFF TO 0xFFFF. Will be fed to the calibration function	INT	
ENGINE 1	FADEC_>FADEC_engines[1].PL_in_dgr	Degrees are all 0 to 100. This will be copied from the "engine controls"	DOUBLE
	FADEC_>FADEC_engines[1].CL_in_pos	0 -> OFF 1 -> START&FEATHER 2 -> MIN 3 -> 900RPM 4 -> MAX	BYTE
	FADEC_>FADEC_engines[1].sensors		
	FADEC_>FADEC_engines[1].out		
	FADEC_>FADEC_engines[1].int_system_mode	0 -> Power OFF 1 -> Feather 2 -> Reverse 3 -> Beta Range 4 -> Flight Range	BYTE
	FADEC_>FADEC_engines[1].Nh_system_mode	0 -> Minimum Power 1 -> Prop RPM by Nh AUTO 2 -> Torque AUTO	BYTE
	FADEC_>FADEC_engines[1].prop_system_mode	0 -> Maintain Pitch 1 -> Constant Speed	BYTE
	FADEC_>FADEC_engines[1].trq_bug		DOUBLE
	FADEC_>FADEC_engines[1].engine_rating_procent		SIGN_CHAR
	FADEC_>FADEC_engines[1].engine_rating_mode	0 -> No Rating 1 -> MTOP 2 -> MCP 3 -> NTOP 4 -> RDC TOP 5 -> MCL 6 -> MCR	BYTE
	FADEC_>FADEC_engines[1].engine_rating_mode_override_requested	0 -> Nothing 1 -> MTOP 3 -> NTOP (reset) 4 -> RDC TOP 5 -> MCL 6 -> MCR 7 -> RDC NP	BYTE
	FADEC_>FADEC_engines[1].Nh_USP	Nh underspeed protection	BOOLEAN
	FADEC_>FADEC_engines[1].Np_USP	Np underspeed protection	BOOLEAN
	FADEC_>FADEC_engines[1].bleed_mode	0 -> OFF 1 -> MIN 2 -> NORM 3 -> MAX	BYTE
	FADEC_>FADEC_engines[1].uptrim	Set by PCU	BOOLEAN
FADEC_>FADEC_engines[1].limits			
FADEC_>FADEC_engines[1].FADEC_electrical_power_available		BOOLEAN	
FADEC_>FADEC_engines[1].FADEC_starter_power_available		BOOLEAN	
FADEC_>FADEC_engines[1].controls.throttle	-0xFFFF TO 0xFFFF. Will be fed to the calibration function	INT	
FADEC_>FADEC_engines[1].controls.prop	-0xFFFF TO 0xFFFF. Will be fed to the calibration function	INT	
OUTPUTS	FADEC_>outputs.light_SELECT		BYTE
	FADEC_>outputs.light_START		BYTE
	FADEC_>start_selector	0 1 2	BYTE
O/HO CONTROLS	FADEC_>controls.ignition1		BYTE
	FADEC_>controls.ignition2		BYTE
	FADEC_>controls.start_button_trigger		BYTE
	FADEC_>controls.start_selector_trigger	0 1 2	BYTE
GENERAL FADEC	FADEC_>FADEC_engines[0].FADEC_failure		UNS. LONG
	FADEC_>FADEC_engines[1].FADEC_failure		UNS. LONG
	FADEC_>QTGEngineControl	Will override the usual engine axis	BYTE
GLOBAL	FADEC_>controls.global_engine_rating_mode_override_requested	0 -> Nothing 1 -> MTOP 3 -> NTOP (reset) 4 -> RDC TOP 5 -> MCL 6 -> MCR 7 -> RDC NP	BYTE
FOR INS	FADEC_>FADEC_engines[0].sensors.Nh		DOUBLE
	FADEC_>FADEC_engines[0].sensors.FF		DOUBLE
	FADEC_>FADEC_engines[0].sensors.ITT		DOUBLE
	FADEC_>FADEC_engines[0].sensors.prop_trq	Calculated	DOUBLE
	FADEC_>FADEC_engines[1].sensors.Nh		DOUBLE
	FADEC_>FADEC_engines[1].sensors.FF		DOUBLE
	FADEC_>FADEC_engines[1].sensors.ITT		DOUBLE
	FADEC_>FADEC_engines[1].sensors.prop_trq	Calculated	DOUBLE
	FADEC_>FADEC_engines[0].sensors.prop_rpm		SHORT INT
	FADEC_>FADEC_engines[0].sensors.NL		DOUBLE
	FADEC_>FADEC_engines[1].sensors.prop_rpm		SHORT INT
	FADEC_>FADEC_engines[1].sensors.NL		DOUBLE
	FADEC_>start_button		BYTE
	FADEC_>start_selector	0 -> OFF 1 -> Left 2 -> Right	BYTE

FDE			
Mode	Command	Comment	Type
CONTROLS	fdeData_->position	To control the position from the slewed device. This is the position from slewed device to the FDE. whenever override is 1 Filled by the terrain simulator. meters If this is 1 or 2 sim will be repositioned. 2 is dialog 1 is slew or pause. From the FDE. This is to acknowledge GS commanded by vision TRUE speed by vision From FDE Position will not update Position and orientation will not update All airplane velocities will be held	
	fdeData_->position_loopback		DOUBLE
	fdeData_->reportedAGL_m		BYTE
	fdeData_->commendReposition		BYTE
	fdeData_->commendReposition_confirmation		DOUBLE
	fdeData_->vision_GS		DOUBLE
	fdeData_->vision_TS		DOUBLE
	fdeData_->RealflapsPosition_grad		DOUBLE
	fdeData_->holdPosition		BYTE
	fdeData_->holdPositionAndOrientation		BYTE
	fdeData_->holdVelocities		BYTE
	fdeData_->prop_J [0]		DOUBLE
	fdeData_->prop_J [1]		DOUBLE
	fdeData_->prop_Thrust [0]		DOUBLE
	fdeData_->prop_Thrust [1]		DOUBLE
	fdeData_->engine_Params [0]		
	fdeData_->engine_Params [1]		
	fdeData_->prop_Beta_dgr [0]		DOUBLE
fdeData_->prop_Beta_dgr [1]	DOUBLE		
fdeData_->ForceZeroElevation_FSI		BOOLEAN	
WSA	fdeData_->flightInstrumentData.WDA_rad	Wind Drift Angle	DOUBLE
	fdeData_->transportDelayTestRequested	From FSI	BOOLEAN
	fdeData_->transportDelayTimeMs		DOUBLE
	fdeData_->solar_zenith_angle_rad	Current solar zenith angle, rad. >100 degrees is night. 45 degrees is full day	DOUBLE
	fdeData_->solar_azimuth_angle_rad	Current solar azimuth angle, rad. At the moment wrong!! must be fixed if to be used	DOUBLE
	fdeData_->GS	From FDE	DOUBLE
	fdeData_->FSI_apply_pushback_speed_active		BOOLEAN
	fdeData_->FSI_apply_pushback_speed		DOUBLE
fdeData_->pilot_acceleration_Z	+UP	DOUBLE	

FDR			
Mode	Command	Comment	Type
GENERAL	FDR_>CL_FDR	Caution light FDR	BOOLEAN
	FDR_>CVR_status_led	Out of the status led	BYTE
	FDR_>CVR_erase_button_pressed		BOOLEAN
	FDR_>CVR_test_button_pressed		BOOLEAN
	FDR_>FDR_test_button_pressed		BOOLEAN
	FDR_>ELT_switch	0 -> ARMED 1 -> TEST 2 -> ON	BYTE
	FDR_>ELT_test_indicator		BYTE

Fire			
Mode	Command	Comment	Type
CONTROLS	Fire_>>controls.handle1_pulled		BOOLEAN
	Fire_>>controls.handle2_pulled		BOOLEAN
	Fire_>>controls.ENGINE_1_EXTG_SW	0 -> OFF 1 -> AFT BTL 2 -> FWD BTL	BYTE
	Fire_>>controls.ENGINE_2_EXTG_SW	1 -> OFF 1 -> AFT BTL 2 -> FWD BTL	BYTE
	Fire_>>controls.ENGINE_TEST_DETECTION_SW	0 -> OFF 1 -> ENGINE 1 TEST 2 -> ENGINE 2 TEST	BYTE
	Fire_>>controls.APU_EXTG_SW	0 -> OFF 1 EXTNG	BOOLEAN
	Fire_>>controls.APU_FIRE_TEST_SW	0 -> OFF 1 -> TEST	BOOLEAN
	Fire_>>controls.BAGGAGE_AFT_EXTG_SW	0 -> OFF 1 EXTNG	BOOLEAN
	Fire_>>controls.BAGGAGE_AFT_TEST_SW	0 -> OFF 1 -> TEST 1 2 -> TEST 2	BYTE
	Fire_>>controls.BAGGAGE_FWD_EXTG_SW	0 -> OFF 1 -> EXTNG	BOOLEAN
	Fire_>>controls.BAGGAGE_FWD_TEST_SW	0 -> OFF 1 -> TEST	BOOLEAN
	Fire_>>controls.ENGINE_FIRE_WARN_RESET_SW		BOOLEAN
OUTPUTS	Fire_>>outputs.handle1_light		BOOLEAN
	Fire_>>outputs.handle2_light		BOOLEAN
	Fire_>>outputs.FaultA_L_light		BOOLEAN
	Fire_>>outputs.FaultB_L_light		BOOLEAN
	Fire_>>outputs.FaultA_R_light		BOOLEAN
	Fire_>>outputs.FaultB_R_light		BOOLEAN
	Fire_>>outputs.Engine_1_BottleArmingAft_light		BOOLEAN
	Fire_>>outputs.Engine_1_BottleArmingFwd_light		BOOLEAN
	Fire_>>outputs.Engine_2_BottleArmingAft_light		BOOLEAN
	Fire_>>outputs.Engine_2_BottleArmingFwd_light		BOOLEAN
	Fire_>>outputs.Engine_BottleLow_light		BOOLEAN
	Fire_>>outputs.EngineFire_glareshield_light	On the glareshield	BOOLEAN
	Fire_>>outputs.APU_BottleArm_light		BOOLEAN
	Fire_>>outputs.APU_FIRE_light		BOOLEAN
	Fire_>>outputs.APU_BottleLow_light		BOOLEAN
	Fire_>>outputs.APU_Fault_light		BOOLEAN
	Fire_>>outputs.BaggageAft_VentInlt_light		BOOLEAN
	Fire_>>outputs.BaggageAft_VentOutlt_light		BOOLEAN
	Fire_>>outputs.BaggageAft_Smoke_light		BOOLEAN
	Fire_>>outputs.BaggageAft_Extg_light		BOOLEAN
	Fire_>>outputs.BaggageFwd_Smoke_light		BOOLEAN
	Fire_>>outputs.BaggageFwd_Extg_light		BOOLEAN
	Fire_>>outputs.Baggage_bottle_AftArm_light		BOOLEAN
	Fire_>>outputs.Baggage_bottle_ForwardArm_light		BOOLEAN
	Fire_>>outputs.Baggage_bottle_AftLow_light		BOOLEAN
	Fire_>>outputs.Baggage_bottle_ForwardLow_light		BOOLEAN
	Fire_>>failures.force_engine1_fire	This is something for detectors to detect, if they can	BOOLEAN
	Fire_>>failures.force_engine2_fire	This is something for detectors to detect, if they can	BOOLEAN
	Fire_>>failures.force_aft_baggage_cmpt_fire	This is something for detectors to detect, if they can	BOOLEAN
	Fire_>>failures.force_fwd_baggage_cmpt_fire	This is something for detectors to detect, if they can	BOOLEAN
	Fire_>>failures.force_apu_fire	This is something for detectors to detect, if they can	BOOLEAN
	Fire_>>bottles.engine_FWD_bottle_discharged	1 -> if discharged into engine 1 2 -> if discharged into engine 2	BYTE
Fire_>>bottles.engine_AFT_bottle_discharged	2 -> if discharged into engine 1 2 -> if discharged into engine 2	BYTE	
Fire_>>bottles.fwd_HRD_baggage_comp_bottle_discharged	0 OR 1	BOOLEAN	
Fire_>>bottles.aft_HRD_baggage_comp_bottle_discharged	0 OR 1	BOOLEAN	
Fire_>>bottles.LRD_baggage_comp_bottle_discharged	0 OR 1	BOOLEAN	
Fire_>>bottles.APU_bottle_discharged	0 OR 1	BOOLEAN	
Fire_>>bottles.LRD_baggage_comp_bottle_discharged	0 OR 1	BOOLEAN	
FLOODED	Fire_>>bottles.aft_HRD_baggage_comp_flooded	Will maintain for 30 seconds to cancel the fire	BYTE
	Fire_>>bottles.fwd_HRD_baggage_comp_flooded	Will maintain for 30 seconds to cancel the fire	BYTE
	Fire_>>bottles.engine1_flooded	Will maintain for 30 seconds to cancel the fire Can be 0 OR 1 OR 2 (discharge counts)	BYTE
	Fire_>>bottles.engine2_flooded	Will maintain for 30 seconds to cancel the fire Can be 0 OR 1 OR 2 (discharge counts)	BYTE
	Fire_>>bottles.APU_flooded	Will maintain for 30 seconds to cancel the fire	BYTE

Mode	Command	Right Controls	Comment	Type
	FlightControls->out.brake_r_temp_dgr		Left brake temperature, from 0.0 (ambient), 1.0 (nominal), 1.5 (fire)	FLOAT
	FlightControls->out.brake_l_temp_dgr		Right brake temperature, from 0.0 (ambient), 1.0 (nominal), 1.5 (fire)	FLOAT
	FlightControls->out.rudderTrim_cnd		FROM 1 TO 1	FLAOT
	FlightControls->out.ElevatorTrimCmd		FROM 0 TO 255 for gauge	BYTE
	FlightControls->out.gear_open_ind		G1	BYTE
	FlightControls->out.gear_unsafe_ind		G13	BYTE
	FlightControls->out.gear_locked_ind		G4	BYTE
	FlightControls->out.gearl_open_ind		G11	BYTE
	FlightControls->out.gearl_unsafe_ind		G3	BYTE
	FlightControls->out.gearl_locked_ind		G14	BYTE
	FlightControls->out.gearr_open_ind		G12	BYTE
	FlightControls->out.gearr_unsafe_ind		G2	BYTE
	FlightControls->out.gearr_locked_ind		G15	BYTE
	FlightControls->out.GearLockCh			BYTE
	FlightControls->out.FlapsPos_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.rudderTrim_out_gaug		0 TO 255	BYTE
	FlightControls->out.aileronTrim_out_gaug		1 TO 255	BYTE
	FlightControls->out.GearPos		FROM 0 TO 1	FLOAT
	FlightControls->out.GearPosC		FROM 0 TO 1	FLOAT
	FlightControls->out.GearPosR		FROM 0 TO 1	FLOAT
	FlightControls->out.rudderTrim_cnd		FROM -1 TO 1	FLOAT
	FlightControls->out.aileronTrim_cnd		FROM -1 TO 1	FLOAT
	FlightControls->out.ElevatorTrim_cnd		FROM -1 TO 1	FLOAT
	FlightControls->out.Spoiler_L1_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.Spoiler_L2_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.Spoiler_R1_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.Spoiler_R2_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.Aileron_cnd		FROM 0 TO 1	FLOAT
	FlightControls->out.gear_at_ind			BYTE
	FlightControls->out.gear_at_ind			BYTE
	FlightControls->out.gear_at_ind			BYTE
	FlightControls->out.gear_at_ind			BYTE
	FlightControls->sensors.RudderTrimHandlePos		0 1 -> Left 2 -> Left 3 -> Right 4 -> Right	BYTE
	FlightControls->sensors.AileronTrimHandlePos		0 1 -> Left 2 -> Right	BYTE
	FlightControls->sensors.Brake_handle		From the interface	BYTE
	FlightControls->sensors.Elevator0		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Aileron0		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Rudder0		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.BrakeR0		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.BrakeL0		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Elevator1		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Aileron1		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Rudder1		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.BrakeR1		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.BrakeL1		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Tiler		0 TO 1 OR -1 TO 1	DOUBLE
	FlightControls->sensors.Elevator0_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Aileron0_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Rudder0_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.BrakeR0_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.BrakeL0_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Elevator1_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Aileron1_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Rudder1_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.BrakeR1_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.BrakeL1_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.Tiler_ADC		0 TO 1 OR -1 TO 1	WORD
	FlightControls->sensors.ElevatorDisconnect_switch		Default 0	BYTE
	FlightControls->sensors.AileronDisconnect_switch		Default 0	BYTE
	FlightControls->sensors.ClearWarningSwitch		0 -> OFF 1 -> MUTE 2 -> TEST	BYTE
	FlightControls->sensors.SteeringSwitch		0 -> OFF 1 -> ON	BYTE
	FlightControls->sensors.ControlLockCh		1 if ON	BYTE
	FlightControls->sensors.FlapsInout		0 -> FO 1 -> FS 2 -> F10 3 -> F15 4 -> F4F	BYTE
	FlightControls->sensors.GearHandlePos		0 OR 1	BYTE
	FlightControls->sensors.ElevatorTrimUp			BYTE
	FlightControls->sensors.ElevatorTrimDn			BYTE
	FlightControls->sensors.at_TOP_guard_on		1 if OPEN	BYTE
	FlightControls->sensors.at_TOP_switch_on		1 if ON	BYTE
	FlightControls->sensors.at_BTM_handle_pulled		1 if PULLED	BYTE
	FlightControls->sensors.at_BTM_pos_open		1 if OPEN	BYTE
	FlightControls->sensors.at_BTM_handle_pulled		1 if PULLED	BYTE
	FlightControls->sensors.at_BTM_pump_on		1 if ON	BYTE
	FlightControls->sensors.at_BTM_wtr_switch_on		2 if ON	BYTE
	FlightControls->FCU.gaugedata.elev_trim_off_light			BOOLEAN
	FlightControls->FCU.gaugedata.elev_trim_shutoff_cnd			BOOLEAN
	FlightControls->FCU.gaugedata.rudder_H_PCU_shutoff_button		GAUGE Control	BOOLEAN
	FlightControls->FCU.gaugedata.rudder_L_PCU_shutoff_button		GAUGE Control	BOOLEAN
	FlightControls->FCU.gaugedata.rudder_H_PCU_off_light			BOOLEAN
	FlightControls->FCU.gaugedata.rudder_L_PCU_off_light			BOOLEAN
	FlightControls->FCU.gaugedata.spoiler_1_shutoff_button		GAUGE Control	BOOLEAN
	FlightControls->FCU.gaugedata.spoiler_2_shutoff_button		GAUGE Control	BOOLEAN
	FlightControls->FCU.gaugedata.spoiler_1_off_light			BOOLEAN
	FlightControls->FCU.gaugedata.spoiler_2_off_light			BOOLEAN
	FlightControls->FCU.gaugedata.spoiler_1_flight_switch		0 -> TAXI 1 -> Flight	BOOLEAN
	FlightControls->FCU.gaugedata.roll_inout_adv_light		Spoilers advisory light	BOOLEAN
	FlightControls->FCU.gaugedata.roll_inout_adv_light		Spoilers advisory light	BOOLEAN
	FlightControls->FCU.gaugedata.pitch_CL		Show a pitch trim CL	BOOLEAN
	FlightControls->FCU.gaugedata.rudder_H_PCU_push_light			BOOLEAN
	FlightControls->FCU.gaugedata.rudder_L_PCU_push_light			BOOLEAN
	FlightControls->FCU.gaugedata.amisid_sw		0 -> OFF 1 -> ON 2 -> TEST	BOOLEAN
	FlightControls->FCU.gaugedata.elev_trim_push_light			BOOLEAN
	FlightControls->FCU.gaugedata.elev_trim_shutoff_cnd			BOOLEAN
	FlightControls->FCU.Elevator_ind		FROM 0 TO 255	SIGN_CHAN
	FlightControls->FCU.Elevator1_ind		FROM 0 TO 255	SIGN_CHAN
	FlightControls->FCU.Rudder_ind		FROM 0 TO 255	SIGN_CHAN
	FlightControls->failures.induced_elevator_failure			BYTE
	FlightControls->failures.induced_aileron_failure			BYTE
	FlightControls->failures.induced_aileron1_failure			BYTE
	FlightControls->failures.gear_failure			BYTE
	FlightControls->failures.gear_failure			BYTE
	FlightControls->failures.gear_failureC			BYTE
	FlightControls->HOTofFlightControls		If this variable is 1, QDG will exclusively control the flight controls, except for the autopilot	BYTE
	FlightControls->FF_theoreticforce_aileron		For the FS to check the forces	DOUBLE
	FlightControls->FF_theoreticforce_elevator		For the FS to check the forces	DOUBLE
	FlightControls->FS_elevator_pos_override		for the FS to override the elevator. QDG/FlightControls must be on	DOUBLE
	FlightControls->FS_aileron_pos_override		for the FS to override the elevator. QDG/FlightControls must be on	DOUBLE
	FlightControls->FS_rudder_pos_override		for the FS to override the elevator. QDG/FlightControls must be on	DOUBLE
	FlightControls->FS_flaps_pos_override		for the FS to override the elevator. QDG/FlightControls must be on	DOUBLE
	FlightControls->FS_tiler_override_active		Steering Direct Control	DOUBLE
	FlightControls->FS_tiler_override_value		Steering Direct Control	DOUBLE
	FlightControls->aileronTrim_effect		Steering Direct Control	DOUBLE
	FlightControls->HOT param.TilerInput		0 -> Mouse 1 -> Ailerons 2 -> Spoiler Axis	BOOLEAN

		FMC	
Mode	Command	Comment	Type
FMC-1	FMC->FMC[0].eventState.FMS_screen_blink_trigger	To synchronize blinking	BOOLEAN
	FMC->FMC[0].sensorData.IRS_manualHDG_input	NON 0 if there is a manual input, will be reset to 0 when accepted	WORD
	FMC->FMC[0].interface.vector_in_redraw	1 when display is redrawing by GDI	BOOLEAN
	FMC->FMC[0].nav.FMCToChunk	-1 if none. Otherwise sequential number in the chunk list. This will reset to -1 when Leg changes	INT
	FMC->FMC[0].interface.toARCDU_request_remote_tune	Frequency request to ARCDU. ARCDU will set it to 0 when fulfilled	SHORT INT
	FMC->FMC[0].sensorData.IRS_ControlKnob	0 -> OFF 1 -> ATT 2 -> NAV	BYTE
	FMC->FMC[0].eventState.key_press_input	Input (can be held). Will be set to 0 when processed *	BYTE
	FMC->FMC[0].eventState.POWER_avail	if power is available to this FMC	BYTE
	FMC->FMC[0].fuel.alternate_fuel_lbs	Alternate Fuel	INT
	FMC->FMC[0].fuel.hold_fuel_lbs	Hold Fuel	INT
	FMC->FMC[0].fuel.extra_fuel_lbs	Extra Fuel	INT
	FMC->FMC[0].fuel.fuel_onboard_lbs	This value is updated automatically from within the fuel manager, based on fuel_used_since_confirm_lbs [x]	DOUBLE
FMC->FMC[0].fuel.fuel_onboard_confirmed		BOOLEAN	
FMC-2	FMC->FMC[1].eventState.FMS_screen_blink_trigger	To synchronize blinking	BOOLEAN
	FMC->FMC[1].sensorData.IRS_manualHDG_input	NON 0 if there is a manual input, will be reset to 0 when accepted	WORD
	FMC->FMC[1].interface.vector_in_redraw	1 when display is redrawing by GDI	BOOLEAN
	FMC->FMC[1].nav.FMCToChunk	-1 if none. Otherwise sequential number in the chunk list. This will reset to -1 when Leg changes	INT
	FMC->FMC[1].interface.toARCDU_request_remote_tune	Frequency request to ARCDU. ARCDU will set it to 0 when fulfilled	SHORT INT
	FMC->FMC[1].sensorData.IRS_ControlKnob	0 -> OFF 1 -> ATT 2 -> NAV	BYTE
	FMC->FMC[1].eventState.key_press_input	Input (can be held). Will be set to 0 when processed *	BYTE
	FMC->FMC[1].eventState.POWER_avail	if power is available to this FMC	BYTE
	FMC->FMC[1].fuel.alternate_fuel_lbs	Alternate Fuel	INT
	FMC->FMC[1].fuel.hold_fuel_lbs	Hold Fuel	INT
	FMC->FMC[1].fuel.extra_fuel_lbs	Extra Fuel	INT
	FMC->FMC[1].fuel.fuel_onboard_lbs	This value is updated automatically from within the fuel manager, based on fuel_used_since_confirm_lbs [x]	DOUBLE
FMC->FMC[1].fuel.fuel_onboard_confirmed		BOOLEAN	
GEN	FMC->start_activator	To activate FMCs on startup	BYTE

FMC->FMC[0].eventState.key_press_input

Key Press Constants

FMC_KEY_L1	1
FMC_KEY_L2	2
FMC_KEY_L3	3
FMC_KEY_L4	4
FMC_KEY_L5	5
FMC_KEY_R1	6
FMC_KEY_R2	7
FMC_KEY_R3	8
FMC_KEY_R4	9
FMC_KEY_R5	10
FMC_KEY_DATA	20
FMC_KEY_NAV	21
FMC_KEY_VNAV	22
FMC_KEY_DTO	23
FMC_KEY_LIST	24
FMC_KEY_PREV	25
FMC_KEY_FUEL	30
FMC_KEY_FPL	31
FMC_KEY_PERF	32
FMC_KEY_TUNE	33
FMC_KEY_MENU	34
FMC_KEY_NEXT	35
FMC_KEY_A	100
FMC_KEY_B	101
FMC_KEY_C	102
FMC_KEY_D	103
FMC_KEY_E	104
FMC_KEY_F	105
FMC_KEY_G	106
FMC_KEY_H	107
FMC_KEY_I	108
FMC_KEY_J	109
FMC_KEY_K	110
FMC_KEY_L	111
FMC_KEY_M	112
FMC_KEY_N	113
FMC_KEY_O	114
FMC_KEY_P	115
FMC_KEY_Q	116
FMC_KEY_R	117
FMC_KEY_S	118
FMC_KEY_T	119
FMC_KEY_U	120
FMC_KEY_V	121
FMC_KEY_W	122
FMC_KEY_X	123
FMC_KEY_Y	124
FMC_KEY_Z	125
FMC_KEY_0	200
FMC_KEY_1	201
FMC_KEY_2	202
FMC_KEY_3	203
FMC_KEY_4	204
FMC_KEY_5	205
FMC_KEY_6	206
FMC_KEY_7	207
FMC_KEY_8	208
FMC_KEY_9	209
FMC_KEY_BACK	240
FMC_KEY_MSG	241
FMC_KEY_ONOFF	242
FMC_KEY_PLUSMINUS	243
FMC_KEY_ENTER	244

Fuel			
Mode	Command	Comment	Type
CONTROLS	Fuel_->systems[0].AuxFuelPumpButton	1 when pressed	BOOLEAN
	Fuel_->systems[1].AuxFuelPumpButton	1 when pressed	BOOLEAN
	Fuel_->fuelTransferSwitch	0 -> nothing 1 -> Left 2 -> Right	BYTE
	Fuel_->FuelPanelOpenSwitch	0 -> closed 1 -> Open	BYTE
OUTPUTS	Fuel_->systems[0].AuxFuelPumpOperating	0 or 1. Will no operate if not turned on or failed	BYTE
	Fuel_->systems[1].AuxFuelPumpOperating	0 or 1. Will no operate if not turned on or failed	BYTE
	Fuel_->systems[0].FireShutoffValveOpenLight		BYTE
	Fuel_->systems[0].FireShutoffValveClosedLight		BYTE
	Fuel_->systems[1].FireShutoffValveOpenLight		BYTE
	Fuel_->systems[1].FireShutoffValveClosedLight		BYTE
	Fuel_->systems[0].AuxFuelPumplight	0 or 1. Will no operate if not turned on or failed	BYTE
	Fuel_->systems[1].AuxFuelPumplight	0 or 1. Will no operate if not turned on or failed	BYTE
SYN C	Fuel_->fuel_disbalance	Flag	BYTE
	Fuel_->fuel_to_transfer	This is a command to FDE to transfer fuel	DOUBLE
FUEL TEMP	Fuel_->systems[0].FuelTankTemperature		INT
	Fuel_->systems[1].FuelTankTemperature		INT
	Fuel_->systems[0].FuelQuantity_Kgs	For this tank. Filled by the FDE system	DOUBLE
	Fuel_->systems[1].FuelQuantity_Kgs	For this tank. Filled by the FDE system	DOUBLE

HGS			
Mode	Command	Comment	Type
	HGS_ ->hgdModeIntern	Real HGS mode	BYTE

Hydraulic			
Mode	Command	Comment	Type
OUTPUTS	Hydraulic_->outs_stby_hydr_ON_indicator		BOOLEAN
	Hydraulic_->outs_PTU_Operating_light		BOOLEAN
	Hydraulic_->outs_PTU_Operating	Also serves as a light	BOOLEAN
	Hydraulic_->outs_PKBRK_ACC_PRESS_PSI		INT
	Hydraulic_->outs_N3_MAIN_HYD_PRESS_PSI		INT
	Hydraulic_->outs_N3_MAIN_HYD_PRESS_avail		BOOLEAN
	Hydraulic_->outs_N3_HYD_QTY_PROC		BYTE
	Hydraulic_->outs_N3_FSW_LIGHT		BOOLEAN
	Hydraulic_->outs_N3_FSW_CLOSED_SIGNAL		BOOLEAN
	Hydraulic_->outs_N3_DCMP_TIMER_SEC	For how many seconds DCMP is running	WORD
	Hydraulic_->outs_N3_DCMP_OPERATES		BOOLEAN
	Hydraulic_->outs_N3_ACC_PRESS_PSI	Not indicaetd	INT
	Hydraulic_->outs_N2_reservoir_temp_C		INT
	Hydraulic_->outs_N2_MAIN_HYD_PRESS_PSI		INT
	Hydraulic_->outs_N2_MAIN_HYD_PRESS_avail		BOOLEAN
	Hydraulic_->outs_N2_HYD_QTY_PROC		BYTE
	Hydraulic_->outs_N2_HYD_QTY_avail		BOOLEAN
	Hydraulic_->outs_N2_HYD_ISO_VLV_OPEN		BOOLEAN
	Hydraulic_->outs_N2_FSW_CLOSED_SIGNAL	Signal to the valve to open /close	BOOLEAN
	Hydraulic_->outs_N2_EDP_PRESS_PSI		INT
	Hydraulic_->outs_N1_STBY_HYD_PRESS_PSI		INT
	Hydraulic_->outs_N1_STBY_HYD_PRESS_avail		BOOLEAN
	Hydraulic_->outs_N1_reservoir_temp_C		INT
	Hydraulic_->outs_N1_MAIN_HYD_PRESS_PSI		INT
	Hydraulic_->outs_N1_MAIN_HYD_PRESS_avail		BOOLEAN
	Hydraulic_->outs_N1_HYD_QTY_PROC		BYTE
	Hydraulic_->outs_N1_HYD_QTY_avail		BOOLEAN
	Hydraulic_->outs_N1_HYD_PRIORITY_VLV_OPEN		BOOLEAN
	Hydraulic_->outs_N1_HYD_ISO_VLV_OPEN		BOOLEAN
	Hydraulic_->outs_N1_FSW_CLOSED_SIGNAL	Signal to the valve to open /close	BOOLEAN
Hydraulic_->outs_N1_EDP_PRESS_PSI		INT	
Hydraulic_->outs_Hyd_FSOV_2_open_light	Operated directly by CHydraulicSystem	BOOLEAN	
Hydraulic_->outs_Hyd_FSOV_2_closed_light	Operated directly by CHydraulicSystem	BOOLEAN	
Hydraulic_->outs_Hyd_FSOV_1_open_light	Operated directly by CHydraulicSystem	BOOLEAN	
Hydraulic_->outs_Hyd_FSOV_1_closed_light	Operated directly by CHydraulicSystem	BOOLEAN	
FAILURES	Hydraulic_->failures.resetHydQuantity		BYTE
	Hydraulic_->failures.PTUFail	2 -> INOP	BYTE
	Hydraulic_->failures.N3ReservoirLeak	1 -> LEAK 2 -> HOT	BYTE
	Hydraulic_->failures.N3DCMPFail	1 -> LEAK 2 -> INOP 4 -> HOT 64 -> TIMER FAILURE	BYTE
	Hydraulic_->failures.N2ReservoirLeak	1 -> LEAK 2 -> HOT	BYTE
	Hydraulic_->failures.N2ISOValveFail	2 -> INOP	BYTE
	Hydraulic_->failures.N2EDPFail	2 -> INOP 4 -> HOT	BYTE
	Hydraulic_->failures.N1SPUFail	2 -> INOP 4 -> HOT	BYTE
	Hydraulic_->failures.N1ReservoirLeak	1 -> LEAK 2 -> HOT	BYTE
	Hydraulic_->failures.N1PRIOValveFail	2 -> INOP (closed)	BYTE
	Hydraulic_->failures.N1ISOValveFail	2 -> INOP (closed)	BYTE
Hydraulic_->failures.N1EDPFail	2 -> INOP 4 -> HOT	BYTE	
Hydraulic_->failures.handPumpFail	2 -> INOP	BYTE	
CONTROLS	Hydraulic_->controls_stby_hyd_switch		BOOLEAN
	Hydraulic_->controls_ptu_switch		BOOLEAN
	Hydraulic_->controls_n3_isol_vlv_open_switch		BOOLEAN
	Hydraulic_->controls_alt_ldg_selector_door	FALSE -> OFF	BOOLEAN

Hydraulic			
Mode	Command	Comment	Type
GENERAL	Instructor_->VisionSwitch	0 -> Left 1 -> Center 2 -> Right	BYTE
	Instructor_->reposition	Via change detect	
	Instructor_->Volume	0 TO 9	BYTE
	Instructor_->Visibility_feet		INT
	Instructor_->CloudLayer1_base		INT
	Instructor_->CloudLayer1_top		INT
	Instructor_->CloudLayer1_coverage		INT
	Instructor_->CloudLayer2_base		INT
	Instructor_->CloudLayer2_top		INT
	Instructor_->CloudLayer2_coverage		INT
	Instructor_->WindDirectionDegrees		INT
	Instructor_->WindSpeed		INT
	Instructor_->vision_eyepoint_angle		DOUBLE
	Instructor_->vision_eyepoint_offset		DOUBLE
	Instructor_->vision_direction		BYTE
	Instructor_->flight_freeze	0 -> No Freeze	BYTE
	Instructor_->engineQuickStartRequest		BYTE
	Instructor_->MonthOfTheYear		BYTE
	Instructor_->InstructorWindEnable		BOOLEAN
	Instructor_->temperature_override_c		DOUBLE
	Instructor_->empty_weight_set	Set the empty weight. When this is accepted by FDE, weight will be reset to C	DOUBLE
	Instructor_->set_delta_qnh		DOUBLE
Instructor_->SL_temperature_override_c		DOUBLE	
Instructor_->variationAtAirplane_dgr		DOUBLE	
Instructor_->emptypadding		BYTE	

Maintenance			
Mode	Command	Comment	Type
GENERAL	Maintenance_->times_flightHours		DOUBLE
	Maintenance_->times_engineHours[0]	For Engine 0	DOUBLE
	Maintenance_->times_engineHours[1]	For Engine 1	DOUBLE
	Maintenance_->times_brakesHours[0]	For Brakes 0	DOUBLE
	Maintenance_->times_brakesHours[1]	For Brakes 1	DOUBLE
	Maintenance_->times_landings		UNSIG. INT

NAVDATA			
Mode	Command	Comment	Type
	NavData_->InRange_data_stable	Will be true when array contains the valid indexes and is not being compilec	BOOLEAN

NAV System			
Mode	Command	Comment	Type
	Nav_>navsources_failures	Failures for NAV SOURCE	WORD
MARKER BEACON SOURCE	Nav_>MarkerBeaconSource.OM_from_FS_flag	Input from FS	
	Nav_>MarkerBeaconSource.MM_from_FS_flag	Input from FS	
	Nav_>MarkerBeaconSource.IM_from_FS_flag	Input from FS	
	Nav_>ilsVisionData [0]	Input from FS	
	Nav_>ilsVisionData [1]	Input from FS	
	Nav_>MarkerBeaconSource.OM_receiving	Input from FS	
	Nav_>MarkerBeaconSource.MM_receiving	Input from FS	
	Nav_>MarkerBeaconSource.IM_receiving	Input from FS	
NAV & ADF	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].freq	Active frequencies forNAV1	DOUBLE
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].freq	Active frequencies forNAV2	DOUBLE
	Nav_>NavSources[NavSystem_NAVSOURCE_ADF1].freq	Active frequencies for ADF1	DOUBLE
	Nav_>NavSources[NavSystem_NAVSOURCE_ADF2].freq	Active frequencies for ADF2	DOUBLE
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].gs_avail	Glide slope available flag for NAV1	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].gs_avail	Glide slope available flag for NAV2	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].gs_needle	Glide slope needle positions for NAV1	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].gs_needle	Glide slope needle positions for NAV2	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].needle_avail	VOR/ILS needle available	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].needle_avail	VOR/ILS needle available	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].needle	VOR/ILS needle position -1.0 TO 0.0	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].needle	VOR/ILS needle position -1.0 TO 0.0	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV1].navsource_type	TYPE for NAV1	
	Nav_>NavSources[NavSystem_NAVSOURCE_NAV2].navsource_type	TYPE for NAV2	
TRANSPONDER	Nav_>atcTransponders[0].mode	0 -> STBY 1 -> ON 2 -> ALT 3 -> TEST	BYTE
	Nav_>atcTransponders[0].activeCode		UNS. SHORT
	Nav_>atcTransponders[1].mode	0 -> STBY 1 -> ON 2 -> ALT 3 -> TEST	BYTE
	Nav_>atcTransponders[1].activeCode		UNS. SHORT

Oil			
Mode	Command	Comment	Type
GENERAL	Oil_->failures_OSG_failed[0]		
	Oil_->failures_OSG_failed[1]		
	Oil_->failures_PCUVALVE_fine_failed[0]		
	Oil_->failures_PCUVALVE_fine_failed[1]		
	Oil_->failures_PCUVALVE_coarse_failed[0]		
	Oil_->failures_PCUVALVE_coarse_failed[1]		

		PCU	
Mode	Command	Comment	Type
PCU 0	PCU_->PCUs[0].controls_featherPumpButtonPressed	Set when failed is set. Reset by the feather pump with it processed the command	BOOLEAN
	PCU_->PCUs[0].indications_prop_groundrange_light		BOOLEAN
	PCU_->PCUs[0].Autofeather_fail_flag		BOOLEAN
	PCU_->PCUs[0].Autofeather_armed		BOOLEAN
	PCU_->PCUs[0].Autofeather_remote_arm_prohibit		BOOLEAN
	PCU_->PCUs[0].Autofeather_feather_requested_flag		BOOLEAN
	PCU_->PCUs[0].auxFeatherPumpLight		BYTE
	PCU_->PCUs[0].uptrim_armed		BOOLEAN
	PCU_->PCUs[0].number		0 -> L 1 -> R
	PCU_->PCUs[0].electricalPowerAvail		BOOLEAN
	PCU_->PCUs[0].controls_featherPumpButtonPressed	Set when failed is set. Reset by the feather pump with it processed the command	BOOLEAN
	PCU_->PCUs[0].indications_prop_groundrange_light		BOOLEAN
	PCU_->PCUs[0].Autofeather_fail_flag		BOOLEAN
	PCU_->PCUs[0].Autofeather_armed		BOOLEAN
	PCU_->PCUs[0].Autofeather_remote_arm_prohibit		BOOLEAN
	PCU_->PCUs[0].Autofeather_feather_requested_flag		BOOLEAN
	PCU_->PCUs[0].auxFeatherPumpLight		BYTE
	PCU_->PCUs[0].uptrim_armed		BOOLEAN
	PCU_->PCUs[0].number		0 -> L 1 -> R
	PCU_->PCUs[0].electricalPowerAvail		BOOLEAN
AUTOFEATHER	PCU_->AutofeatherSelectButtonPressed		BOOLEAN
	PCU_->AutofeatherSelectLight		BOOLEAN
	PCU_->AutofeatherArmLight		BOOLEAN
DIVERS	PCU_->OSGTestButtonPressed	TRUE when AF test is performed. This will reset if: CLs are above start&feather, PLs are not in DISC, AF is deselected	BOOLEAN
	PCU_->AF_test_counter		WORD
	PCU_->AF_test_performed		BOOLEAN
	PCU_->AF_test_result		BOOLEAN
	PCU_->OSG1MinuteTimer		To indicate the test abort in 1 minute
	PCU_->OSG_test_performed		OSG test is not selected
	PCU_->OSG_test_result		0 -> Aborted 1 -> OK 2 -> Fail
	PCU_->PCUs[0].outs_real_prop_beta		In radians
	PCU_->PCUs[1].outs_real_prop_beta		In radians

// + CLs are above start&feather
// + PLs are not in DISC
// + AF is deselected

PSEU			
Mode	Command	Comment	Type
GENERAL	PSEU_>controls.to_warn_switch		BYTE
	PSEU_>outputs.WOW		
	PSEU_>sensors.WOW_L	Left gear WOW	BOOLEAN
	PSEU_>sensors.WOW_R	Right gear WOW	BOOLEAN
	PSEU_>sensors.WOW_C	Center gear WOW	BOOLEAN

Scripting			
Mode	Command	Comment	Type
GENERAL	Scripting_->empty_		BYTE
	Scripting_->strOutput		
	Scripting_->sc_scriptLeaderHash	SC Sync	DWORD
	Scripting_->scExchangeVar	Numerical Variable SYNC	

Scripting			
Mode	Command	Comment	Type
GENERAL	Sound_->globalVolume		BYTE
	Sound_->request_switch_sound_extern_hash_	Requested from outside	WORD
	Sound_->request_switch_sound_extern	Requested from outside	WORD
	Sound_->request_switch_pos_extern		
	Sound_->inputs.music_request_operation		BYTE
	Sound_->inputs.cabin_announcement_request_operation		BYTE
	Sound_->inputs.cabin_request_fileName	Relative Path + Filename	CHAR
	Sound_->inputs.cockpit_announcement_request_operation		BYTE
	Sound_->inputs.cockpit_request_fileName	Relative Path + Filename	CHAR
	Sound_->inputs.cockpitops_announcement_request_operation		BYTE
Sound_->inputs.cockpitops_announcement_request_fileName	Relative Path + Filename	CHAR	

INDEX						
System	Remarks	Link		System	Remarks	Link
AFCS	AutoFlight Control System	Go...		HGS	Head-Up Guidance System	Go...
AHRS	Attitude and Heading Reference System	Go...		Hydraulics		Go...
Air		Go...		Instructor System		Go...
Air Data		Go...		NavData System		Go...
Anti Ice		Go...		NAV System		Go...
Audio		Go...		Oil		Go...
Cabin		Go...		PCU	Power Control Unit	Go...
Clocks		Go...		PSEU	Proximity Switch Control Unit	Go...
Crew Alerting		Go...		Scripting		Go...
EFIS	Electronic Flight Instrument System	Go...		Sound		Go...
Electrics		Go...		<p>done by bruno teixeira and oleksiy frolov last update: march 28th 2015 ©majestic software</p>		
FADEC	Full Authority Digital Engine Control	Go...				
FDE	Flight Dynamics Engine	Go...				
FDR	Flight Data Recorder	Go...				
Fire		Go...				
Flight Controls		Go...				
FMC	Flight Management Computer	Go...				
Fuel		Go...				

AFCS - AutoFlight Control System			
Mode	Command	Comment	Type
INPUT	AFCS_->inputs.mode_button	Type AFCS_KEY_NONE	BYTE
	AFCS_->inputs.GA_button	On the throttles	BYTE
	AFCS_->inputs.AP_disengage	On the yoke	BYTE
OUTPUT	AFCS_->outputs.YD_engaged	0 -> Nothing BIT1 -> Left BIT2 -> Right	BYTE
	AFCS_->outputs.HSISEL_side_indicator	0 -> Nothing BIT1 -> Left BIT2 -> Right	BYTE
	AFCS_->outputs.AP_diseng_indicator	Autopilot disengage indicator	BYTE
	AFCS_->outputs.AP_engaged_left_lt	Left Light AP	BYTE
	AFCS_->outputs.AP_engaged_right_lt	Right Light AP	BYTE
	AFCS_->outputs.YD_engaged_left_lt	Yaw Demper Left Light	BYTE
	AFCS_->outputs.YD_engaged_right_lt	Yaw Demper Right Light	BYTE
	AFCS_->outputs.HSI_left_lt	Horizontal Situation Indicator Left Light	BYTE
	AFCS_->outputs.HSI_right_lt	Horizontal Situation Indicator Right Light	BYTE
AFCS_->outputs.FGMMaster	Tell the AutoPilot where to take params (0 -> FGM1 1 -> FGM2)	BYTE	
FLIGHT	AFCS_->FGM[0].ias_to_maintain	By the VS mode	INT
	AFCS_->FGM[1].ias_to_maintain	By the VS mode	INT
	AFCS_->FGM[0].current_vertcmd_pitch_dgr	In degrees	INT
	AFCS_->FGM[1].current_vertcmd_pitch_dgr	In degrees	INT
	AFCS_->FGM[0].current_vert_mode	Exept ALT SEL	BYTE
	AFCS_->FGM[1].current_vert_mode	Exept ALT SEL	BYTE
	AFCS_->FGM[0].current_latcmd_bank_dgr	In degrees	INT
	AFCS_->FGM[1].current_latcmd_bank_dgr	In degrees	INT
	AFCS_->FGM[0].current_vertcmd_vspeed	Vertical Speed required to maintain the VS, ALT GS or VNAV mode (in FPM)	INT
	AFCS_->FGM[1].current_vertcmd_vspeed	Vertical Speed required to maintain the VS, ALT GS or VNAV mode (in FPM)	INT
	AFCS_->FGM[0].current_lat_mode	Current Lateral Mode	BYTE
	AFCS_->FGM[1].current_lat_mode	Current Lateral Mode	BYTE
GUIDANCE	AFCS_->FGM[0].failed	FGM1 is failed	BYTE
	AFCS_->FGM[1].failed	FGM2 is failed	BYTE
	AFCS_->FGM[0].ALTSEL_armed	Can be armed with any other mode	BYTE
	AFCS_->FGM[1].ALTSEL_armed	Can be armed with any other mode	BYTE
MODULE	AFCS_->autopitchDisconnect	Disconnect Automatic Elevator Pitch during Flaps 15 to Flaps 35 transitior	Boolean
	AFCS_->pitchChannelFSIDisconnect	Used by FSI_ to make measurements with AP without pitch trim changes	Boolean
	AFCS_->rollChannelFSIDisconnect	Used by FSI_ to make measurements with AP without roll changes	Boolean
	AFCS_->QTG_AP_limits_force_OFF	Prevent AutoPilot disconnect when the aircraft exceeds the FGM bank and pitch limits	Boolean
	AFCS_->QTG_YD_force_OFF	Disengage the Yaw Damper without disconnecting the autopilot (0 or 1)	Boolean
SHARED COCKPIT	AFCS_->sc_pitchwheel_data	Used for shared cockpit Pitch Wheel synchronisation	FLOAT
	AFCS_->sc_FD_lat_data	Used for shared cockpit master FD data synchronisation	FLOAT
	AFCS_->sc_FD_vert_data	Used for shared cockpit master FD data synchronisation	FLOAT

AFCS_->inputs.mode_button

Key Press Constants

AFCS_KEY_NONE	0
AFCS_KEY_IAS	1
AFCS_KEY_VS	2
AFCS_KEY_VNAV	3
AFCS_KEY_ALT	4
AFCS_KEY_ALTSEL	5
AFCS_KEY_HDG	6
AFCS_KEY_NAV	7
AFCS_KEY_APPR	8
AFCS_KEY_BC	9
AFCS_KEY_STBY	10
AFCS_KEY_AP	11
AFCS_KEY_YD	12
AFCS_KEY_HSISEL	13

AHRS - Attitude and Heading Reference System			
Mode	Command	Comment	Type
CONTROLS	AHRS_>AHRS[0].controls.DGSlewLeft_Switch_pressed		BOOLEAN
	AHRS_>AHRS[1].controls.DGSlewLeft_Switch_pressed		
	AHRS_>AHRS[0].controls.DGSlewRight_Switch_pressed		
	AHRS_>AHRS[1].controls.DGSlewRight_Switch_pressed		
	AHRS_>AHRS[0].controls.AlignSwitchPressed		
	AHRS_>AHRS[1].controls.AlignSwitchPressed		
OUTPUT	AHRS_>AHRS[0].outputs.alignIndicator		BOOLEAN
	AHRS_>AHRS[1].outputs.alignIndicator		BOOLEAN
	AHRS_>AHRS[0].outputs.slaveIndicator		BOOLEAN
	AHRS_>AHRS[1].outputs.slaveIndicator		BOOLEAN
	AHRS_>AHRS[0].outputs.basicIndicator		BOOLEAN
	AHRS_>AHRS[1].outputs.basicIndicator		BOOLEAN
	AHRS_>AHRS[0].outputs.Pitch_rad		DOUBLE
	AHRS_>AHRS[1].outputs.Pitch_rad		DOUBLE
	AHRS_>AHRS[0].outputs.Bank_rad		DOUBLE
	AHRS_>AHRS[1].outputs.Bank_rad		DOUBLE
	AHRS_>AHRS[0].outputs.THDG_rad		DOUBLE
	AHRS_>AHRS[1].outputs.THDG_rad		DOUBLE
	AHRS_>AHRS[0].outputs.accel_Long	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[1].outputs.accel_Long	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[0].outputs.accel_Lat	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[1].outputs.accel_Lat	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[0].outputs.accel_Vert	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[1].outputs.accel_Vert	For the AutoPilot and Speed trend	DOUBLE
	AHRS_>AHRS[0].outputs.vSpeed	In FPM - Feet Per Minute Dumped	DOUBLE
	AHRS_>AHRS[1].outputs.vSpeed	In FPM - Feet Per Minute Dumped	DOUBLE
	AHRS_>AHRS[0].outputs.MHDG_rad		DOUBLE
	AHRS_>AHRS[1].outputs.MHDG_rad		DOUBLE
	AHRS_>AHRS[0].outputs.att_Valid		BOOLEAN
	AHRS_>AHRS[1].outputs.att_Valid		BOOLEAN
AHRS_>AHRS[0].outputs.hdg_Valid		BOOLEAN	
AHRS_>AHRS[1].outputs.hdg_Valid		BOOLEAN	
AHRS_>AHRS[0].outputs.vSpeed_Valid		BOOLEAN	
AHRS_>AHRS[1].outputs.vSpeed_Valid		BOOLEAN	
VARIOUS	AHRS_>AHRS_mag_input_from_FS		DOUBLE
	AHRS_>AHRS[0].ext_failed		BYTE
	AHRS_>AHRS[1].ext_failed		BYTE
	AHRS_>AHRS[0].ext_flux_valve_fail		BYTE
	AHRS_>AHRS[1].ext_flux_valve_fail		BYTE
	AHRS_>AHRS[0].powerOnTestProgress	0 if completed assuming 20 seconds (624). For the debugging 5 seconds (156)	INT
	AHRS_>AHRS[1].powerOnTestProgress	0 if completed assuming 20 seconds (624). For the debugging 5 seconds (156)	INT
	AHRS_>AHRS[0].OperationalTestProgress	0 if completed. 156 initial for 5 seconds	INT
	AHRS_>AHRS[1].OperationalTestProgress	0 if completed. 156 initial for 5 seconds	INT
	AHRS_>AHRS[0].alignmentMode		INT
	AHRS_>AHRS[1].alignmentMode		INT
	AHRS_>AHRS[0].DG_mode		BOOLEAN
	AHRS_>AHRS[1].DG_mode		BOOLEAN
	AHRS_>AHRS[0].SLAVE_mode		BOOLEAN
	AHRS_>AHRS[1].SLAVE_mode		BOOLEAN
	AHRS_>AHRS[0].BASIC_mode		BOOLEAN
AHRS_>AHRS[1].BASIC_mode		BOOLEAN	
SENSORS	AHRS_>AHRS[0].sensors.speed_Roll	In FPS (Feet Per Second)	DOUBLE
	AHRS_>AHRS[0].sensors.speed_Pitch	In FPS (Feet Per Second)	DOUBLE
	AHRS_>AHRS[0].sensors.speed_Yaw	In FPS (Feet Per Second)	DOUBLE
	AHRS_>AHRS[0].sensors		

AIR DATA			
Mode	Command	Comment	Type
ADC	airData_->ADC[0].controls.barometricCorrectionRotaryDelta	Set by the cockpit	S CHAR
	airData_->ADC[0].controls.barometricCorrectionSTDPush	Set by the cockpit	BYTE
	airData_->ADC[0].sensors		
	airData_->ADC[0].outputs		
	airData_->ADC[0].PowerAvailable	If ADC available	BOOLEAN
	airData_->ADC[1].controls.barometricCorrectionRotaryDelta	Set by the cockpit	S CHAR
	airData_->ADC[1].controls.barometricCorrectionSTDPush	Set by the cockpit	BYTE
	airData_->ADC[1].sensors		
	airData_->ADC[1].outputs		
	airData_->ADC[1].PowerAvailable	If ADC available	BOOLEAN
COMMON	airData_->controls_pitotStaticIsolationValveSwitch	Switch in the panel	BOOLEAN
	airData_->output_pitotStaticIsolationValveLightNorm	Switch in the panel	BOOLEAN
	airData_->output_pitotStaticIsolationValveLightIso	Switch in the panel	BOOLEAN
	airData_->controls_Operational_Test_Switch	Side Panels 1 -> ADC1 2 -> ADC2	BOOLEAN
	airData_->output_WarningTone	Will be combined from the ADC individual warning tone requirement:	BOOLEAN
OUTPUT	airData_->ADC[0].outputs.altitude		DOUBLE
	airData_->ADC[0].outputs.IAS		DOUBLE
	airData_->ADC[0].outputs.TAS		DOUBLE
	airData_->ADC[0].ext_failed	External Failure	BOOLEAN
	airData_->ADC[1].ext_failed	External Failure	BOOLEAN
RED COC	airData_->ADC[0].outputs.barometric_correction	Saved in the memory when power off	DOUBLE
	airData_->ADC[1].outputs.barometric_correction	Saved in the memory when power off	DOUBLE

AIR			
Mode	Command	Comment	Type
AIR CONDITIONING	Air_>controls.recirc_sw		BOOLEAN
	Air_>controls.bleed1_sw		BOOLEAN
	Air_>controls.bleed2_sw		BOOLEAN
	Air_>controls.bleed_mode_sw	0 -> NORM 1 -> MIN 2 -> MAX	BYTE
	Air_>controls.cabin_duct_gauge_sw	0 -> CAB DUCT 1 -> FC DUCT 2 -> CABIN	BYTE
	Air_>controls.AC_pack_1_sw	0 -> MAN 1 AUTO 2 -> OFF	BYTE
	Air_>controls.AC_pack_2_sw	0 -> MAN 1 AUTO 2 -> OFF	BYTE
	Air_>controls.cabin_temp_regulator		WORD
Air_>controls.fltcomp_temp_regulator		WORD	
<hr/>			
RESURIZATION	Air_>controls.manual_pressure_inc_pressed	From gauges	BYTE
	Air_>controls.manual_pressure_dec_pressed	From gauges	BYTE
	Air_>controls.fwd_valve_pos_input		WORD
<hr/>			
OUTPUT	Air_>logic.pressurepanel_FAULT_light		BYTE
	Air_>logic.temperatureGaugeOutput	0 to 255 (0 -> 0 255 ->100)	BYTE
	Air_>controls.pressureMode_Switch	0 -> Manual 1 -> Dump 2 -> Auto	BYTE
	Air_>controls.landing_altitude_input	From -2000 to 14000	WORD
<hr/>			
VARIOUS	Air_>controls.emer_outflow_vlv_selector	0 or 1. Position of the emergency outflow valve selector lever	
	Air_>logic.ECUs		
	Air_>logic.selectedLandingAltitude	From -2000 to 14000	INT
	Air_>logic.cabinPressureDiffIndication	In PSI	DOUBLE
	Air_>logic.cabinRateIndication		DOUBLE
	Air_>logic.cabinAltitudeIndication		DOUBLE
Air_>logic.toEFISreducebrightness	EFIS must reduce brightness when operating on one fan	BYTE	

		ANTI ICE	
Mode	Command	Comment	Type
CONTROLS	Antice_-_controls.L_PitotHeat_SW	0 -> OFF 1 -> ON	BYTE
	Antice_-_controls.R_PitotHeat_SW	1 -> OFF 1 -> ON	BYTE
	Antice_-_controls.airframe_mode_select_sw	0 -> OFF 1 -> SLOW 2 -> FAST 3 -> MAN	BYTE
	Antice_-_controls.airframe_manual_select_sw	1 -> OFF 2 -> OFF 3-7 -> POSITIONS 1-5 0 -> POSITION 6	BYTE
	Antice_-_controls.prop_timer_sw	1 -> ON 2 -> TEST	BYTE
	Antice_-_controls.engine_intake_sw_1	0 -> OFF 1 -> ON (push only)	BYTE
	Antice_-_controls.engine_intake_sw_2	1 -> OFF 1 -> ON (push only)	BYTE
	Antice_-_controls.boot_air_sw	0 -> OFF 1 -> ISOL	BYTE
	Antice_-_controls.stby_pitot_static_sw	1 -> OFF 1 -> STDBY	BYTE
	Antice_-_controls.Windshield_heat_sw	0 -> OFF 1 -> WARMUP 2 -> NORM	BYTE
	Antice_-_controls.Wiper_sw	0 -> PARK 1 -> OFF 2 -> LOW 3 -> HIGH	BYTE
	Antice_-_controls.plt_side_wdo_ht_sw	0 -> OFF 1 -> ON	BYTE
	OUTPUT	Antice_-_outputs.boot_inflation_L_t[0]	
Antice_-_outputs.boot_inflation_L_t[1]			BYTE
Antice_-_outputs.boot_inflation_L_t[2]			BYTE
Antice_-_outputs.boot_inflation_L_t[3]			BYTE
Antice_-_outputs.boot_inflation_L_t[4]			BYTE
Antice_-_outputs.boot_inflation_L_t[5]			BYTE
Antice_-_outputs.boot_inflation_L_t[6]			BYTE
Antice_-_outputs.boot_inflation_R_t[0]			BYTE
Antice_-_outputs.boot_inflation_R_t[1]			BYTE
Antice_-_outputs.boot_inflation_R_t[2]			BYTE
Antice_-_outputs.boot_inflation_R_t[3]			BYTE
Antice_-_outputs.boot_inflation_R_t[4]			BYTE
Antice_-_outputs.boot_inflation_R_t[5]			BYTE
Antice_-_outputs.boot_inflation_R_t[6]			BYTE
Antice_-_outputs.engine_intake_L_OPN_It			BYTE
Antice_-_outputs.engine_intake_L_HTR_It			BYTE
Antice_-_outputs.engine_intake_L_CLOSED_It			BYTE
Antice_-_outputs.engine_intake_R_OPN_It			BYTE
Antice_-_outputs.engine_intake_R_HTR_It			BYTE
Antice_-_outputs.engine_intake_R_CLOSED_It			BYTE
Antice_-_outputs.prop_advisory_L_It		Prop Heater left light	BYTE
Antice_-_outputs.prop_advisory_R_It		Prop Heater left light	BYTE
Antice_-_outputs.deicepressure_L_psi			BYTE
Antice_-_outputs.deicepressure_R_psi			BYTE
Antice_-_controls.alt_Wiper_sw		Located on the pilot's panel 0 -> OFF 1 -> ON	BYTE
Antice_-_controls.plt_wdo_demist_lever		0 or 1	BYTE
Antice_-_outputs.icingLevelAmbient		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelWingL		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelWingR		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelEngineIntakeL		0 to 1	DOUBLE
Antice_-_outputs.icingLevelEngineIntakeR		0 to 1	DOUBLE
Antice_-_outputs.icingLevelPropL		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelPropR		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelElevL		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelElevR		0 to 1 IN	DOUBLE
Antice_-_outputs.icingLevelWindshield		0 to 1	DOUBLE
Antice_-_outputs.icingLevelPltSideWdo		0 to 1	DOUBLE
Antice_-_outputs.icingLevelCpltSideWdo		0 to 1	DOUBLE
Antice_-_outputs.enginePrecipitationAccumul_L		0 to 1. Accumulates if door is not open. IN	DOUBLE
Antice_-_outputs.enginePrecipitationAccumul_R		1 to 1. Accumulates if door is not open. IN	DOUBLE
Antice_-_outputs.icingLevelEngineDoorL	0 to 1. Accumulates if door is open but door heater is not on	DOUBLE	
Antice_-_outputs.icingLevelEngineDoorR	0 to 1. Accumulates if door is open but door heater is not on	DOUBLE	
Antice_-_outputs.icingLevelPitotTubeL	0 to 1 IN	DOUBLE	
Antice_-_outputs.icingLevelPitotTubeR	0 to 1 IN	DOUBLE	
Antice_-_outputs.icingLevelPitotTubeSby	0 to 1 IN	DOUBLE	
Antice_-_outputs.icingLevelAlphaSensorL	0 to 1 IN	DOUBLE	
Antice_-_outputs.icingLevelAlphaSensorR	0 to 1 IN	DOUBLE	
Antice_-_outputs.mistLevelAmbient	0 to 255	SHORT INT	
Antice_-_outputs.mistLevelWindshield	0 to 1	DOUBLE	
Antice_-_outputs.WindshieldDemistOperating	Boolean	BYTE	
Antice_-_outputs.mistLevelPilotWdo	0 to 1	DOUBLE	
Antice_-_outputs.PilotWdoDemistOperating	Boolean	BYTE	
Antice_-_outputs.mistLevelCaplotWdo	0 to 1	DOUBLE	
VARIOUS	Antice_-_failures.in_ice_condition	External trigger. If airplane is in the icing conditions (stage 0..4)	BYTE

Mode	Command	AUDIO	Comment	Type
CONTROL	Audio->VHFReceiver[0] Freq			U SHORT
	Audio->VHFReceiver[0] control_tx		1 to transmit	BOOLEAN
	Audio->VHFReceiver[0] control_squash			BOOLEAN
	Audio->VHFReceiver[0] m_flag			BOOLEAN
	Audio->VHFReceiver[1] Freq			U SHORT
	Audio->VHFReceiver[1] control_tx		1 to transmit	BOOLEAN
OTHER	Audio->VHFReceiver[0] MMR_receiver_sensitivity			BYTE
	Audio->VHFReceiver[0] test_On		if TEST is ON	BYTE
	Audio->VHFReceiver[0] unused4			BYTE
	Audio->VHFReceiver[1] MMR_receiver_sensitivity		if TEST is ON	BYTE
	Audio->VHFReceiver[1] test_On			BYTE
AUDIO	Audio->controls.Left_MIC_PTT			BOOLEAN
	Audio->controls.Left_MIC_NPH			BOOLEAN
	Audio->controls.Right_MIC_PTT			BOOLEAN
	Audio->controls.Right_MIC_NPH			BOOLEAN
	Audio->controls.Observer_mic_sel_control		Value of OBSERVER_MIC_SEL_x	BYTE
	Audio->controls.Observer_volume_control		Indices of OBSERVER_VOLUME_x	WORD
	Audio->controls.Observer_INT_RAD_switch		0 -> Nothing 1 -> INT 2 -> RAD	BYTE
	Audio->controls.Observer_BDCM_MARK_switch		0 -> BDCM 1 -> MARK	BYTE
	Audio->controls.Observer_VDR_MLS_switch		0 -> VDR 1 -> MLS	BYTE
	Audio->controls.Observer_volume			BYTE
	Audio->controls.Observer_volume_control[1]			BOOLEAN
	Audio->controls.Observer_volume_control[2]			BOOLEAN
	Audio->controls.Observer_volume_control[3]			BOOLEAN
	Audio->controls.Observer_volume_control[4]			BOOLEAN
	Audio->controls.Observer_volume_control[5]			BOOLEAN
Audio->controls.Observer_volume_control[6]			BOOLEAN	
Audio->controls.Observer_volume_control[7]			BOOLEAN	
Audio->controls.Observer_volume_control[8]			BOOLEAN	
Audio->controls.Observer_volume_control[9]			BOOLEAN	
Audio->controls.Observer_volume_control[10]			BOOLEAN	
Audio->controls.Observer_volume_control[11]			BOOLEAN	
Audio->controls.Observer_volume_control[12]			BOOLEAN	
Audio->controls.Observer_volume_control[13]			BOOLEAN	
SYSTEM	Audio->audioPASystem.controls_SEATBELT_SWITCH_position			BYTE
	Audio->audioPASystem.controls_NOSMOKING_SWITCH_position			BYTE
	Audio->audioPASystem.inputs_LEFT_PA_MODE_SELECTOR_pressed		1 -> PA button 2 -> CHME 3 -> CALL 4 -> EMER button	BYTE
	Audio->audioPASystem.inputs_RIGHT_PA_MODE_SELECTOR_pressed		2 -> PA button 2 -> CHME 3 -> CALL 4 -> EMER button	BYTE
	Audio->audioPASystem.inputs_PA_MODE_SELECTOR_pressed		3 -> PA button 2 -> CHME 3 -> CALL 4 -> EMER button	BYTE
	Audio->audioPASystem.filter_selected_by_crew		0 -> LEFT 1 -> RIGHT 2 -> PA (Used to detect the PA properly)	BYTE
	Audio->audioPASystem.PA_mode		1 -> PA 2 -> CALL 3 -> EMER Internal mode	BYTE
	Audio->audioPASystem.controls_SEATBELT_LIGHT		Passenger Seat Belt Light	BYTE
	Audio->audioPASystem.controls_NOSMOKING_LIGHT		Passenger No Smoking Light	BYTE
	Audio->audioPASystem.tosoundsystem_H_sound_requested		To sound system, to play H sound. Reset to zero by sound system	BYTE
Audio->audioPASystem.tosoundsystem_HL_sound_requested		To sound system, to play HL sound. Reset to zero by sound system	BYTE	
Audio->audioPASystem.tosoundsystem_L_sound_requested		To sound system, to play L sound. Reset to zero by sound system	BYTE	
Audio->audioPASystem.tosoundsystem_PA_sound_HPN_requested		If this particular HL sound to be played in the HPNs. Reset to zero by sound system	BYTE	
Audio->audioPASystem.light_PA			BYTE	
Audio->audioPASystem.light_CALL			BYTE	
Audio->audioPASystem.light_EMER			BYTE	
VHF3	Audio->VHFReceiver[2] Freq			U SHORT
	Audio->VHFReceiver[2] control_tx			BOOLEAN
	Audio->VHFReceiver[2] control_squash			BOOLEAN
	Audio->VHFReceiver[2] m_flag			BOOLEAN
	Audio->entertainment.microphoneSelectorOverrideToPA		Signal to audio logic to override mic selector when captain entertainment announcement is active	BYTE
Audio->entertainmentcockpitDoorPosition		0 -> Closed 100 -> Open	BYTE	
Audio->VHF3Volume_Inp		VHF3 volume from VHF3 panel	WORD	
Audio->VHF3Volume		VHF3 volume from VHF3 panel	BYTE	
Audio->VHF3Volume_Ext			BOOLEAN	
VCF	Audio->shw_intercom_volumes_net[0][0]			
	Audio->shw_intercom_volumes_net[0][1]			
	Audio->shw_intercom_volumes_net[0][2]			
	Audio->shw_intercom_volumes_net[0][3]			
	Audio->shw_intercom_volumes_net[0][4]			
	Audio->shw_intercom_volumes_net[0][5]			
	Audio->shw_intercom_volumes_net[0][6]			
	Audio->shw_intercom_volumes_net[0][7]			
	Audio->shw_intercom_volumes_net[1][0]			
	Audio->shw_intercom_volumes_net[1][1]			
	Audio->shw_intercom_volumes_net[1][2]			
	Audio->shw_intercom_volumes_net[1][3]			
	Audio->shw_intercom_volumes_net[1][4]			
Audio->shw_intercom_volumes_net[1][5]				
Audio->shw_intercom_volumes_net[1][6]				
Audio->shw_intercom_volumes_net[1][7]				
VCF	Audio->shw_intercom_volumes_net[2][0]			
	Audio->shw_intercom_volumes_net[2][1]			
	Audio->shw_intercom_volumes_net[2][2]			
	Audio->shw_intercom_volumes_net[2][3]			
	Audio->shw_intercom_volumes_net[2][4]			
	Audio->shw_intercom_volumes_net[2][5]			
	Audio->shw_intercom_volumes_net[2][6]			
	Audio->shw_intercom_volumes_net[2][7]			
	Audio->shw_intercom_volumes_net[3][0]			
	Audio->shw_intercom_volumes_net[3][1]			
	Audio->shw_intercom_volumes_net[3][2]			
	Audio->shw_intercom_volumes_net[3][3]			
	Audio->shw_intercom_volumes_net[3][4]			
Audio->shw_intercom_volumes_net[3][5]				
Audio->shw_intercom_volumes_net[3][6]				
Audio->shw_intercom_volumes_net[3][7]				
VCF	Audio->shw_intercom_volumes_net[4][0]			
	Audio->shw_intercom_volumes_net[4][1]			
	Audio->shw_intercom_volumes_net[4][2]			
	Audio->shw_intercom_volumes_net[4][3]			
	Audio->shw_intercom_volumes_net[4][4]			
	Audio->shw_intercom_volumes_net[4][5]			
	Audio->shw_intercom_volumes_net[4][6]			
	Audio->shw_intercom_volumes_net[4][7]			
	Audio->shw_intercom_volumes_net[5][0]			
	Audio->shw_intercom_volumes_net[5][1]			
	Audio->shw_intercom_volumes_net[5][2]			
	Audio->shw_intercom_volumes_net[5][3]			
	Audio->shw_intercom_volumes_net[5][4]			
Audio->shw_intercom_volumes_net[5][5]				
Audio->shw_intercom_volumes_net[5][6]				
Audio->shw_intercom_volumes_net[5][7]				
SYSTEM COM	Audio->SBVHF_Mode		0 -> OR 1 -> OR 2	BYTE
	Audio->controls.SBVHF_Control		Gauge Input: 0 -> CHT 1 -> OR 2 -> TEST	BYTE
	Audio->controls.SBVHF_tune_small			S CHAR
	Audio->controls.SBVHF_tune_large		Exchange gauge input: (1) -> 1 (2) 2s to 7s (3) -> 7s	BYTE

CABIN			
Mode	Command	Comment	Type
ANIMATIONS	Cabin_->FC_animation.out_CPT_HAND_POS	FCANM_HANDPOS_XXX	S CHAR
	Cabin_->FC_animation.out_CPT_HEAD_POS	-50 -> Left 50 -> Right	S CHAR
	Cabin_->FC_animation.out_FO_HAND_POS	FCANM_HANDPOS_XXX	S CHAR
	Cabin_->FC_animation.out_FO_HEAD_POS	-50 -> Left 50 -> Right	S CHAR
	Cabin_->pushbackStatusInput	0 -> No Push 1 -> Left 2 -> Right 3 -> Straight	BYTE
	Cabin_->inputs.DoorFWDPAx_open		BYTE
	Cabin_->pushBackInProgress		BOOLEAN
	Cabin_->entertainment_auto_announcements		BYTE
	Cabin_->fms_pushbackinput	0 -> No Push 1 -> Left 2 -> Right 3 -> Straight	BYTE

CLOCKS			
Mode	Command	Comment	Type
CONTROLS	Clocks_>clock[0].controls_.modeSwitchPos	0 -> DATE 1 -> LOC 2 -> GMT 3 -> SET Not zeroed	BYTE
	Clocks_>clock[0].controls_.ET_pressed	Reset by the system	BYTE
	Clocks_>clock[0].controls_.CHR_pressed	Reset by the gauge	BYTE
	Clocks_>clock[1].controls_.modeSwitchPos	0 -> DATE 1 -> LOC 2 -> GMT 3 -> SET Not zeroed	BYTE
	Clocks_>clock[1].controls_.ET_pressed	Reset by the system	BYTE
	Clocks_>clock[1].controls_.CHR_pressed	Reset by the gauge	BYTE
OUTPUTS	Clocks_>clock[0].outputs_.d1_left	Upper screen digits Left 255 -> no digits	BYTE
	Clocks_>clock[0].outputs_.d1_right	Upper screen digits Right 255 -> no digits	BYTE
	Clocks_>clock[0].outputs_.d2_left	Lower screen digits Left 255 -> no digits	BYTE
	Clocks_>clock[0].outputs_.d2_right	Lower screen digits Right 255 -> no digits	BYTE
	Clocks_>clock[0].outputs_.dot_pos	Dot position: 1 -> LOC 2 -> GMT	BYTE
	Clocks_>clock[0].outputs_.arrow_output_sec	0 to 58	BYTE
	Clocks_>clock[0].outputs_.sc_upper	1 if : is shown	BYTE
	Clocks_>clock[0].outputs_.sc_lower	1 if : is shown	BYTE
	Clocks_>clock[1].outputs_.d1_left	Upper screen digits Left 255 -> no digits	BYTE
	Clocks_>clock[1].outputs_.d1_right	Upper screen digits Right 255 -> no digits	BYTE
	Clocks_>clock[1].outputs_.d2_left	Lower screen digits Left 255 -> no digits	BYTE
	Clocks_>clock[1].outputs_.d2_right	Lower screen digits Right 255 -> no digits	BYTE
	Clocks_>clock[1].outputs_.dot_pos	Dot position: 1 -> LOC 2 -> GMT	BYTE
	Clocks_>clock[1].outputs_.arrow_output_sec	0 to 58	BYTE
	Clocks_>clock[1].outputs_.sc_upper	1 if : is shown	BYTE
	Clocks_>clock[1].outputs_.sc_lower	1 if : is shown	BYTE

CREW ALERTING			
CAUTION AND WARNING			
Mode	Command	Comment	Type
CONTROLS	CautionWarning->ins_		
	CautionWarning->outs_		
	CautionWarning->controls_master_caution_reset_button		BOOLEAN
	CautionWarning->controls_master_warning_reset_button		BOOLEAN
<hr/>			
OUTPUTS	CautionWarning->outs_master_caution_light_blinkin	0 -> OFF 1 -> Blinking	BOOLEAN
	CautionWarning->outs_master_warning_light_blinkin	0 -> OFF 1 -> Blinking	BOOLEAN
<hr/>			
EGPWS			
Mode	Command	Comment	Type
CONTROLS	EGPWS->GPWS_pullup_onscreen		BYTE
	EGPWS->GPWS_GS_onscreen		BYTE
	EGPWS->GPWS_FLAPS_10_onscreen		BYTE
	EGPWS->GPWS_FLAPS_15_onscreen		BYTE
	EGPWS->GPWS_FLAPS_35_onscreen		BYTE
	EGPWS->GPWS_GS_cancel_button_pressed		BYTE
	EGPWS->GPWS_test_button_pressed		BYTE
	EGPWS->GPWS_flaps_selector	2 -> 10° 1 -> 15° 0 -> 35°	BYTE
	EGPWS->GPWS_model1_switch	Momentary when 1	BYTE
	EGPWS->GPWS_model1_indicator	Light output. 1 -> STD 2 -> STEEP	BYTE
	EGPWS->GPWS_terrain_inhibit_sw_L	From the gauge. Push -> ON Push -> OFF	BYTE
	EGPWS->GPWS_terrain_inhibit_sw_R	From the gauge. Push -> ON Push -> OFF	BYTE
	EGPWS->GPWS_terrain_inhibit_status_L	This is for the light, and to tell EFIS about inhibiting the terrain	BYTE
	EGPWS->GPWS_terrain_inhibit_status_R	This is for the light, and to tell EFIS about inhibiting the terrain	BYTE
	EGPWS->EGPWS_MFD_Overlays[0].peaksMode		BYTE
	EGPWS->EGPWS_MFD_Overlays[0].terrainMinIndication	To indicate on the MFD	BYTE
	EGPWS->EGPWS_MFD_Overlays[0].terrainMaxIndication	To indicate on the MFD	BYTE
	EGPWS->EGPWS_MFD_Overlays[1].peaksMode		BYTE
EGPWS->EGPWS_MFD_Overlays[1].terrainMinIndication	To indicate on the MFD	BYTE	
EGPWS->EGPWS_MFD_Overlays[1].terrainMaxIndication	To indicate on the MFD	BYTE	
<hr/>			
STALL SYSTEM			
Mode	Command	Comment	Type
CONTROLS	Stall->STALL_computer_TEST_switch	0 -> Middle 1 -> TEST1 2 -> TEST2	BYTE
	Stall->StickPusherOFFSwitch1		BYTE
	Stall->StickPusherOFFSwitch2		BYTE
	Stall->REF_SPEED_switch	0..1	BYTE
INDIC.	Stall->StickPusher_Light		
<hr/>			
SLOW OUTPUT	Stall->SPM [0].AoA	Angle of Attack (radians)	DOUBLE
	Stall->SPM [1].AoA	Angle of Attack (radians)	DOUBLE
	Stall->SPM [0].stick_shaker_signal	< Vstall+10	BOOLEAN
	Stall->SPM [1].stick_shaker_signal	< Vstall+10	BOOLEAN
	Stall->SPM [0].stick_pusher_signal	<=Vstall	BOOLEAN
	Stall->SPM [1].stick_pusher_signal	<=Vstall	BOOLEAN
	Stall->ice_detecting_valid	0 or 1	BOOLEAN
	Stall->ice_detected	0 or 1	BOOLEAN
<hr/>			
TCAS			
Mode	Command	Comment	Type
CONTROLS	TCAS->tcasACForNDCcount		BYTE
	TCAS->tcasVisionPlaneCount		BYTE
	TCAS->tcasVisionPlaneIndex	All entries are valid	DWORD
	TCAS->tcasVisionPlaneIndex_DistSq	Distance is set entirely by the FSX. All entries are set	FLOAT
	TCAS->TCAS_test_active	Is a counter in seconds also	BYTE
	TCAS->TCAS_mode	Of type TCAS_MODE_STBY	BYTE
	TCAS->TCASscenarioinput	Of type TCAS_SCENARIO_NONE	BYTE
<hr/>			
WEATHER			
Mode	Command	Comment	Type
CONTROLS	WXR->controls.mode_cntrl	0 -> OFF 1 -> STBY 2 -> TEST 3 -> ON	BYTE
	WXR->controls.map_mode_cntrl	1 -> WX 2 -> WXA 3 -> GMAP	BYTE
	WXR->controls.stab_off_cntrl	0 -> ON 1 -> OFF	BYTE
	WXR->controls.tilt_cntrl	From 0x0000 to 0xFFFF (Center 0x8000)	WORD
	WXR->controls.gain_cntrl	From 0x0000 to 0xFFFF (Neutral 0x8000)	WORD
	WXR->wxr_mode	0 -> OFF 1 -> STBY 2 -> TEST 3 -> ON	BYTE
	WXR->WXA_requested		BYTE
	WXR->WXR_submode	0 -> WXR 1 -> GMAP	BYTE
	WXR->WXA_requested		BYTE
	WXR->ant_fail	If isn't in the correct position, this will fail	BYTE
	WXR->ant_pos_dgr	From -45.0 to 45.0	DOUBLE
	WXR->ant_motor_dir	0 -> OFF 1 -> LEFT 2 -> RIGHT	BYTE
	WXR->ant_tilt_dgr	From -15.9 to 16.0	DOUBLE
	WXR->ant_gain	From 0.25 to 4.0	DOUBLE
	WXR->ant_stab_on		BYTE
	WXR->TGT_ind		BYTE

???

???