Flight Safety System

Members

- **BULIGA THEODOR IOAN**

  theodor.buliga@alumnos.upm.es

- **CELAYA RODRIGUEZ JUAN ANTONIO**

  juan.celaya.rodriguez@alumnos.upm.es

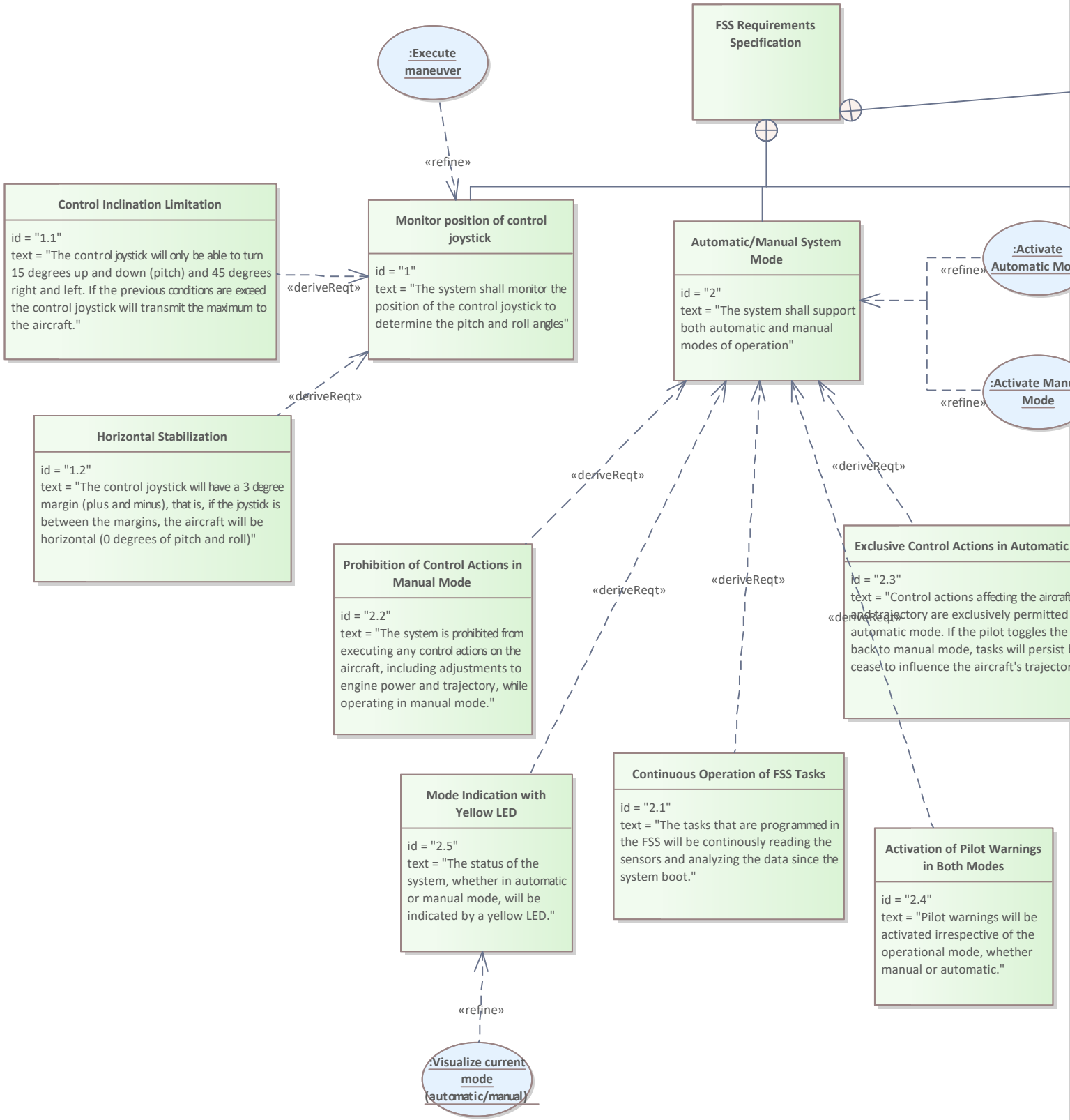
- **OROBIYI-TUDURACHE GERARD-CONSTANTIN**

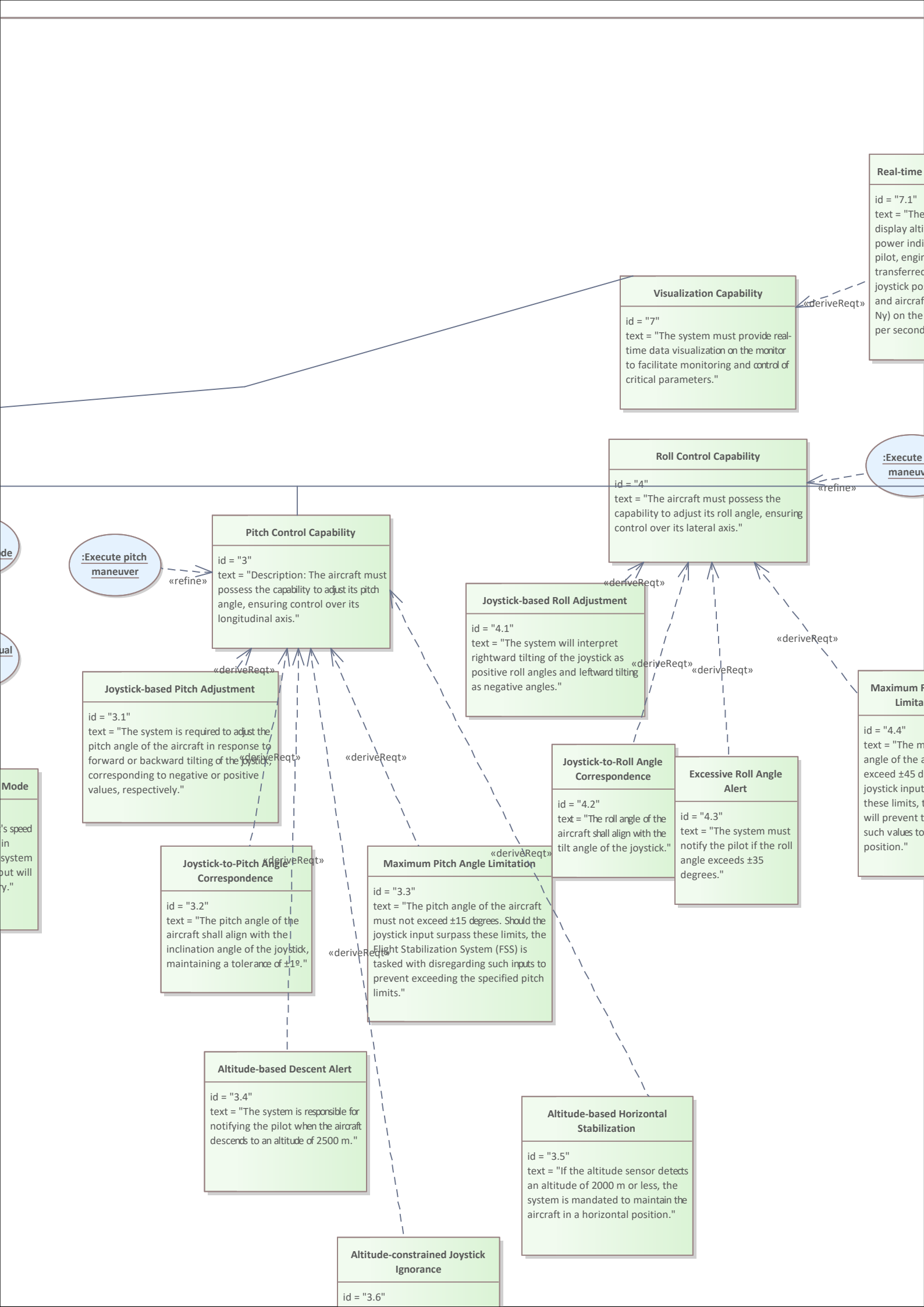
  gerard-constantin.o@alumnos.upm.es

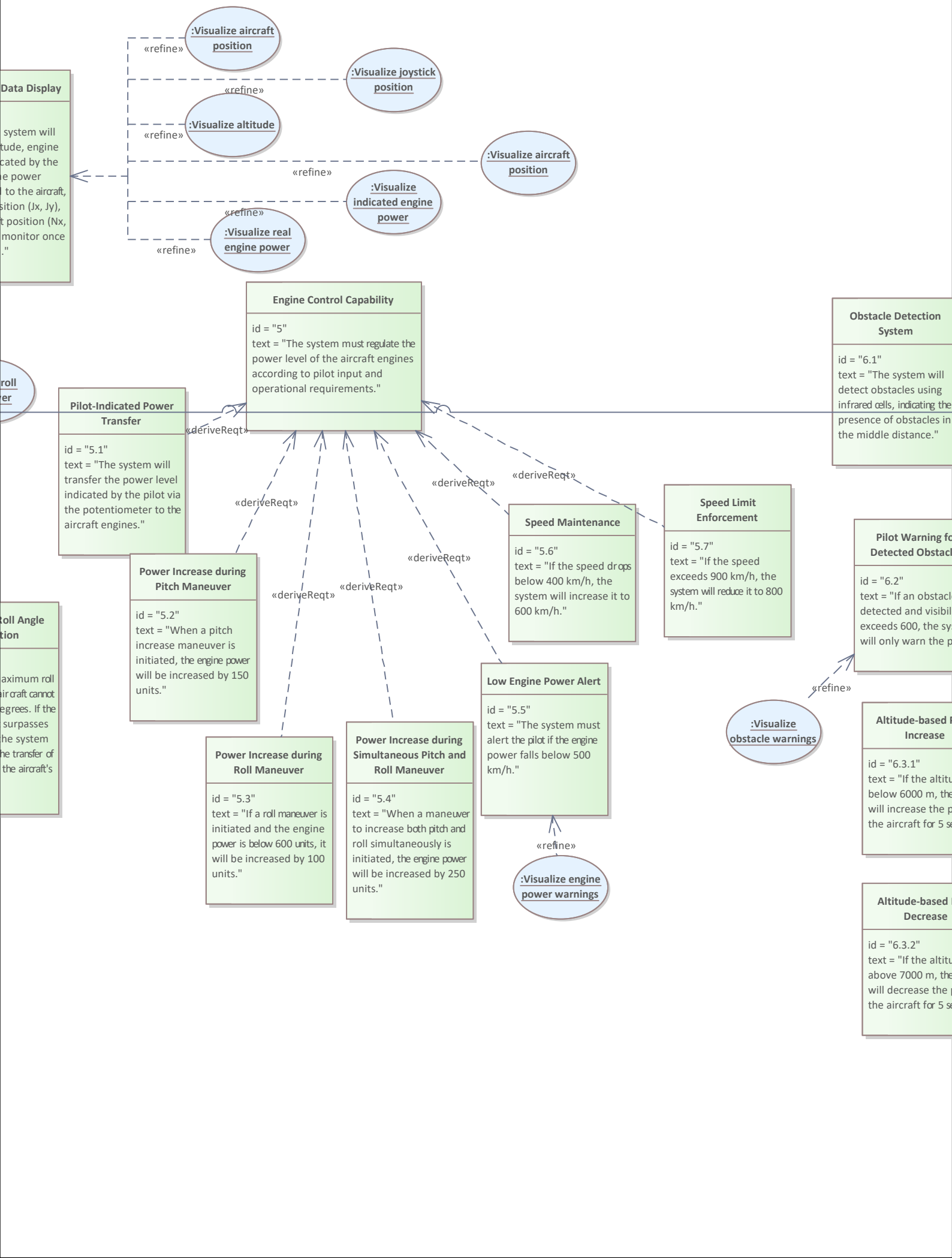
Group opinion

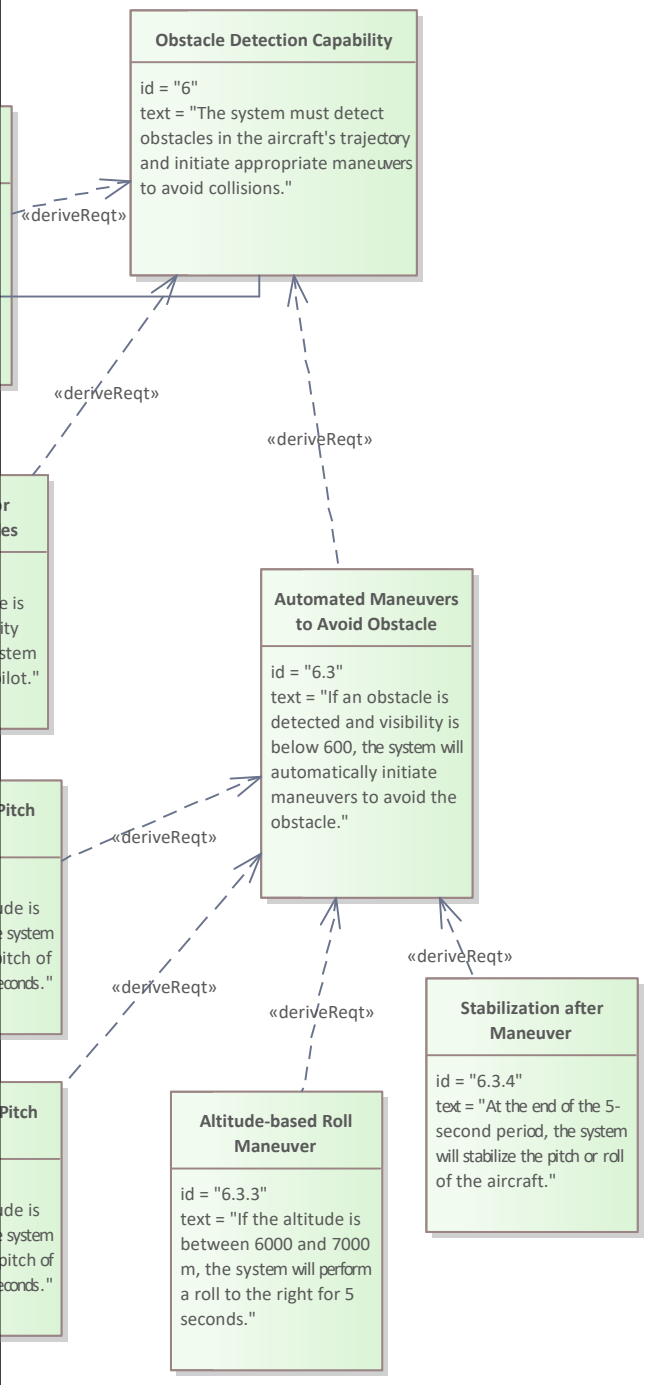
This project helped us develop a better understanding of embedded systems by creating the diagrams specific to the system and implementing the code necessary for it to run and fulfill all the requirements. The group project also developed our teamwork skills, at least one member of the group taking part in each task.

Diagrams

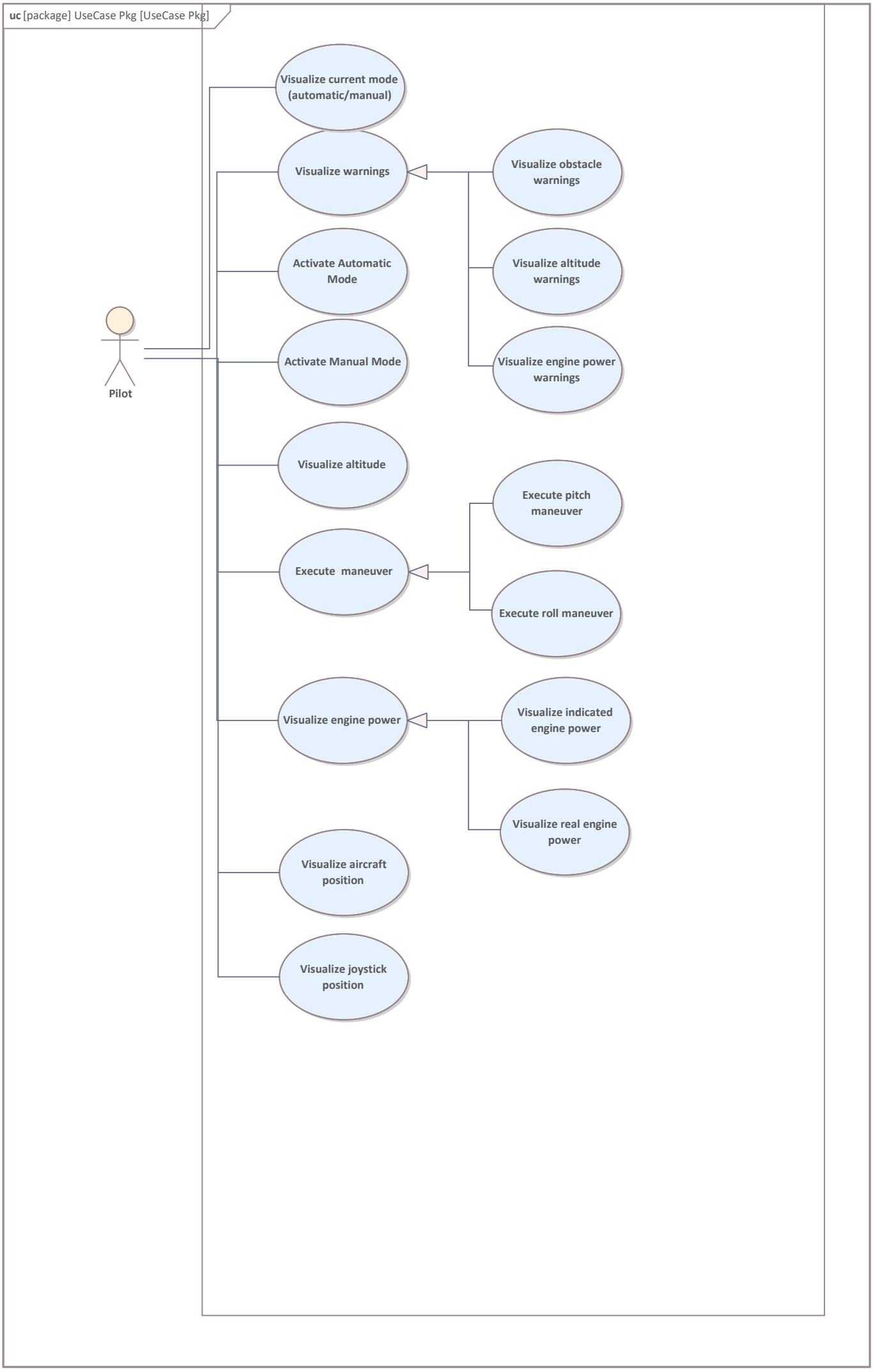

FSS Requirements Specification
:Execute maneuver
«refine»
Control Inclination Limitation
id = "1.1"
text = "The control joystick will only be able to turn 15 degrees up and down (pitch) and 45 degrees right and left. If the previous conditions are exceed the control joystick will transmit the maximum to the aircraft."
«deriveReqt»
Monitor position of control joystick
id = "1"
text = "The system shall monitor the position of the control joystick to determine the pitch and roll angles"
Automatic/Manual System Mode
id = "2"
text = "The system shall support both automatic and manual modes of operation"
«refine»
:Activate Automatic Mo
«refine»
:Activate Manu Mode
«deriveReqt»
Horizontal Stabilization
id = "1.2"
text = "The control joystick will have a 3 degree margin (plus and minus), that is, if the joystick is between the margins, the aircraft will be horizontal (0 degrees of pitch and roll)"
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
Prohibition of Control Actions in Manual Mode
id = "2.2"
text = "The system is prohibited from executing any control actions on the aircraft, including adjustments to engine power and trajectory, while operating in manual mode."
Exclusive Control Actions in Automatic
id = "2.3"
text = "Control actions affecting the aircraft and trajectory are exclusively permitted automatic mode. If the pilot toggles the back to manual mode, tasks will persist cease to influence the aircraft's trajector
Mode Indication with Yellow LED
id = "2.5"
text = "The status of the system, whether in automatic or manual mode, will be indicated by a yellow LED."
Continuous Operation of FSS Tasks
id = "2.1"
text = "The tasks that are programmed in the FSS will be continously reading the sensors and analyzing the data since the system boot."
Activation of Pilot Warnings in Both Modes
id = "2.4"
text = "Pilot warnings will be activated irrespective of the operational mode, whether manual or automatic."
«refine»
:Visualize current mode (automatic/manual)

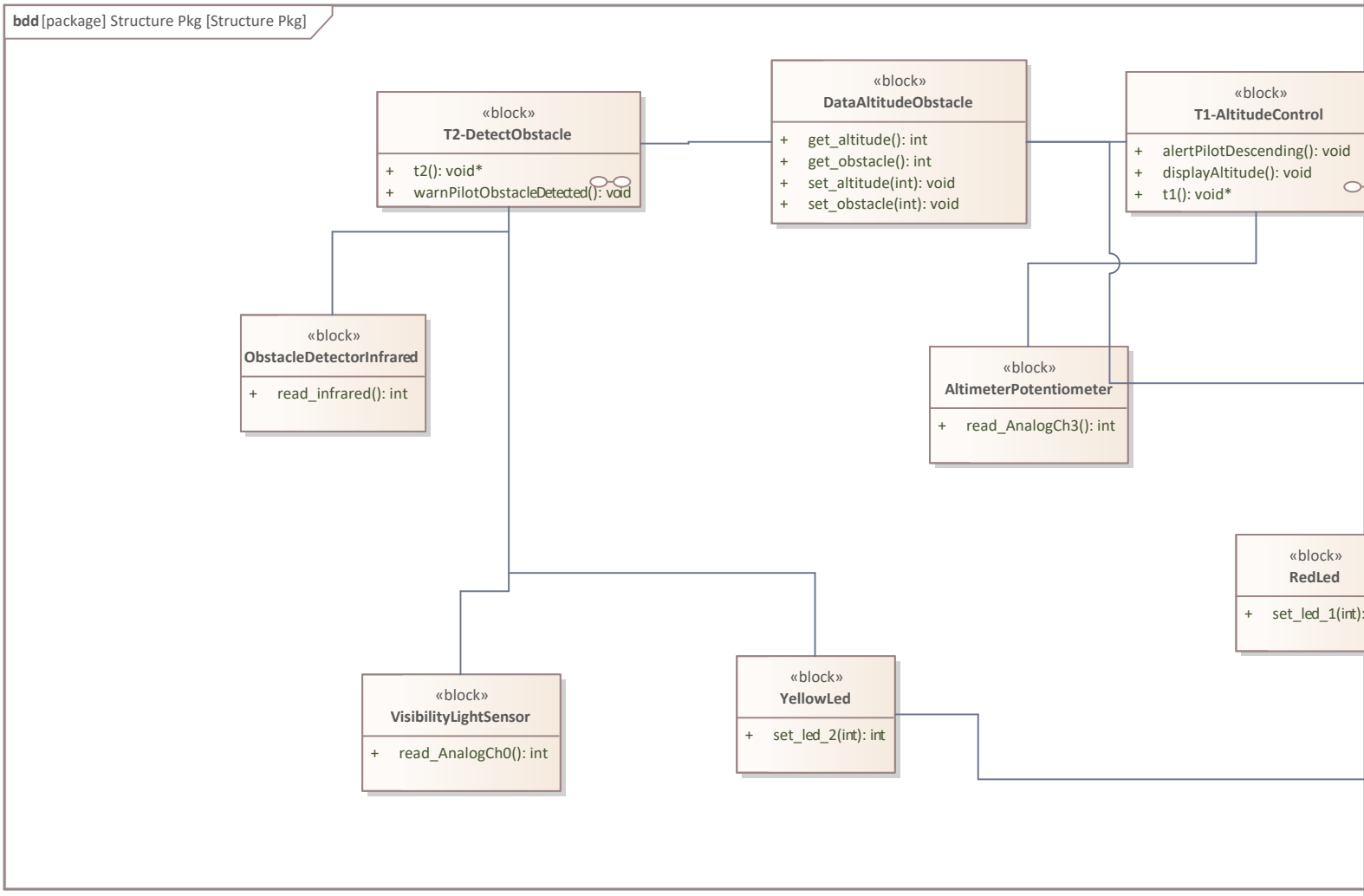

Visualization Capability
id = "7"
text = "The system must provide real-time data visualization on the monitor to facilitate monitoring and control of critical parameters."
«deriveReqt»
Real-time
id = "7.1"
text = "The
display alti
power indi
pilot, engir
transferred
joystick po
and aircraf
Ny) on the
per second
Roll Control Capability
id = "4"
text = "The aircraft must possess the capability to adjust its roll angle, ensuring control over its lateral axis."
«refine»
:Execute
maneuv
Pitch Control Capability
id = "3"
text = "Description: The aircraft must possess the capability to adjust its pitch angle, ensuring control over its longitudinal axis."
:Execute pitch maneuver
«refine»
Joystick-based Roll Adjustment
id = "4.1"
text = "The system will interpret rightward tilting of the joystick as positive roll angles and leftward tilting as negative angles."
«deriveReqt»
Joystick-based Pitch Adjustment
id = "3.1"
text = "The system is required to adjust the pitch angle of the aircraft in response to forward or backward tilting of the joystick, corresponding to negative or positive values, respectively."
Joystick-to-Roll Angle Correspondence
id = "4.2"
text = "The roll angle of the aircraft shall align with the tilt angle of the joystick."
Excessive Roll Angle Alert
id = "4.3"
text = "The system must notify the pilot if the roll angle exceeds ±35 degrees."
Maximum
Limita
id = "4.4"
text = "The m
angle of the a
exceed ±45 d
joystick input
these limits, t
will prevent t
such values to
position."
Joystick-to-Pitch Angle Correspondence
id = "3.2"
text = "The pitch angle of the aircraft shall align with the inclination angle of the joystick, maintaining a tolerance of ±1º."
Maximum Pitch Angle Limitation
id = "3.3"
text = "The pitch angle of the aircraft must not exceed ±15 degrees. Should the joystick input surpass these limits, the Flight Stabilization System (FSS) is tasked with disregarding such inputs to prevent exceeding the specified pitch limits."
Altitude-based Descent Alert
id = "3.4"
text = "The system is responsible for notifying the pilot when the aircraft descends to an altitude of 2500 m."
Altitude-based Horizontal Stabilization
id = "3.5"
text = "If the altitude sensor detects an altitude of 2000 m or less, the system is mandated to maintain the aircraft in a horizontal position."
Altitude-constrained Joystick Ignorance
id = "3.6"

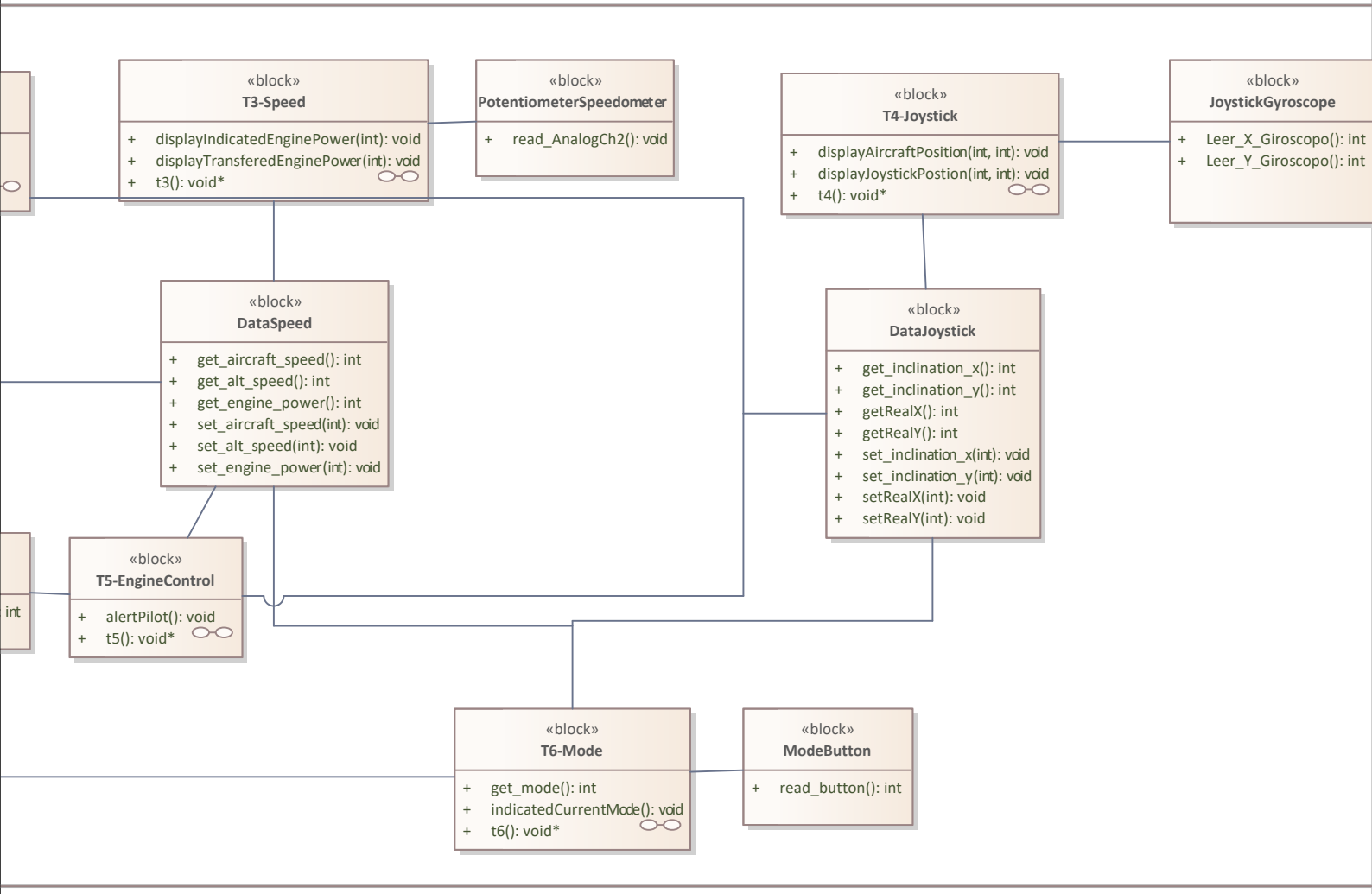

:Visualize aircraft position
«refine»
:Visualize joystick position
«refine»
:Visualize altitude
«refine»
:Visualize aircraft position
«refine»
:Visualize indicated engine power
«refine»
:Visualize real engine power
«refine»
Data Display
system will
tude, engine
cated by the
e power
to the aircraft,
sition (Jx, Jy),
t position (Nx,
monitor once
."
Engine Control Capability
id = "5"
text = "The system must regulate the power level of the aircraft engines according to pilot input and operational requirements."
Obstacle Detection System
id = "6.1"
text = "The system will detect obstacles using infrared cells, indicating the presence of obstacles in the middle distance."
Pilot-Indicated Power Transfer
id = "5.1"
text = "The system will transfer the power level indicated by the pilot via the potentiometer to the aircraft engines."
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
Speed Maintenance
id = "5.6"
text = "If the speed drops below 400 km/h, the system will increase it to 600 km/h."
Speed Limit Enforcement
id = "5.7"
text = "If the speed exceeds 900 km/h, the system will reduce it to 800 km/h."
Pilot Warning fo
Detected Obstacl
id = "6.2"
text = "If an obstacl
detected and visibil
exceeds 600, the sy
will only warn the p
Power Increase during Pitch Maneuver
id = "5.2"
text = "When a pitch increase maneuver is initiated, the engine power will be increased by 150 units."
Low Engine Power Alert
id = "5.5"
text = "The system must alert the pilot if the engine power falls below 500 km/h."
«refine»
:Visualize obstacle warnings
Altitude-based
Increase
id = "6.3.1"
text = "If the altitu
below 6000 m, the
will increase the p
the aircraft for 5 s
Power Increase during Roll Maneuver
id = "5.3"
text = "If a roll maneuver is initiated and the engine power is below 600 units, it will be increased by 100 units."
Power Increase during Simultaneous Pitch and Roll Maneuver
id = "5.4"
text = "When a maneuver to increase both pitch and roll simultaneously is initiated, the engine power will be increased by 250 units."
«refine»
:Visualize engine power warnings
Altitude-based
Decrease
id = "6.3.2"
text = "If the altitu
above 7000 m, the
will decrease the p
the aircraft for 5 s
roll
er
Roll Angle
tion
aximum roll
ircraft cannot
egrees. If the
surpasses
he system
he transfer of
the aircraft's

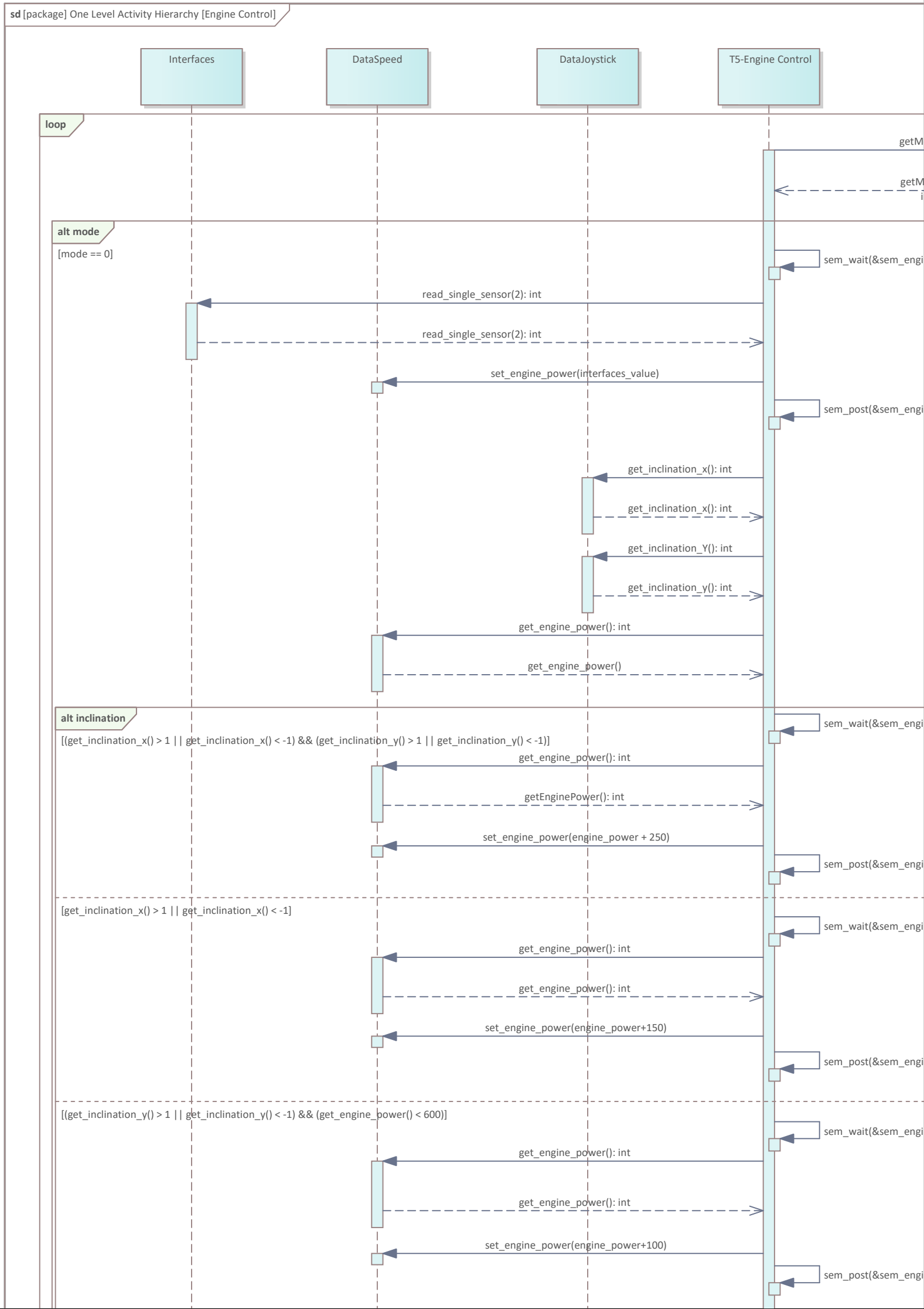

Obstacle Detection Capability
id = "6"
text = "The system must detect obstacles in the aircraft's trajectory and initiate appropriate maneuvers to avoid collisions."
«deriveReqt»
«deriveReqt»
«deriveReqt»
Automated Maneuvers to Avoid Obstacle
id = "6.3"
text = "If an obstacle is detected and visibility is below 600, the system will automatically initiate maneuvers to avoid the obstacle."
«deriveReqt»
«deriveReqt»
«deriveReqt»
«deriveReqt»
Stabilization after Maneuver
id = "6.3.4"
text = "At the end of the 5-second period, the system will stabilize the pitch or roll of the aircraft."
Altitude-based Roll Maneuver
id = "6.3.3"
text = "If the altitude is between 6000 and 7000 m, the system will perform a roll to the right for 5 seconds."

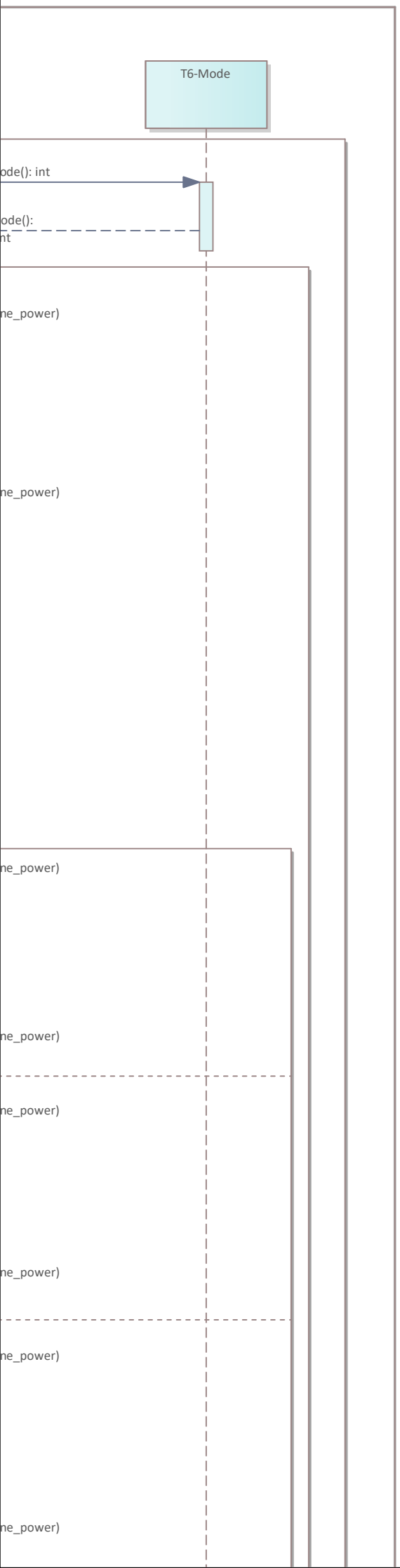
text = "When the altitude is 2000 m or lower and the pilot attempts further descent, the system should disregard joystick inputs."

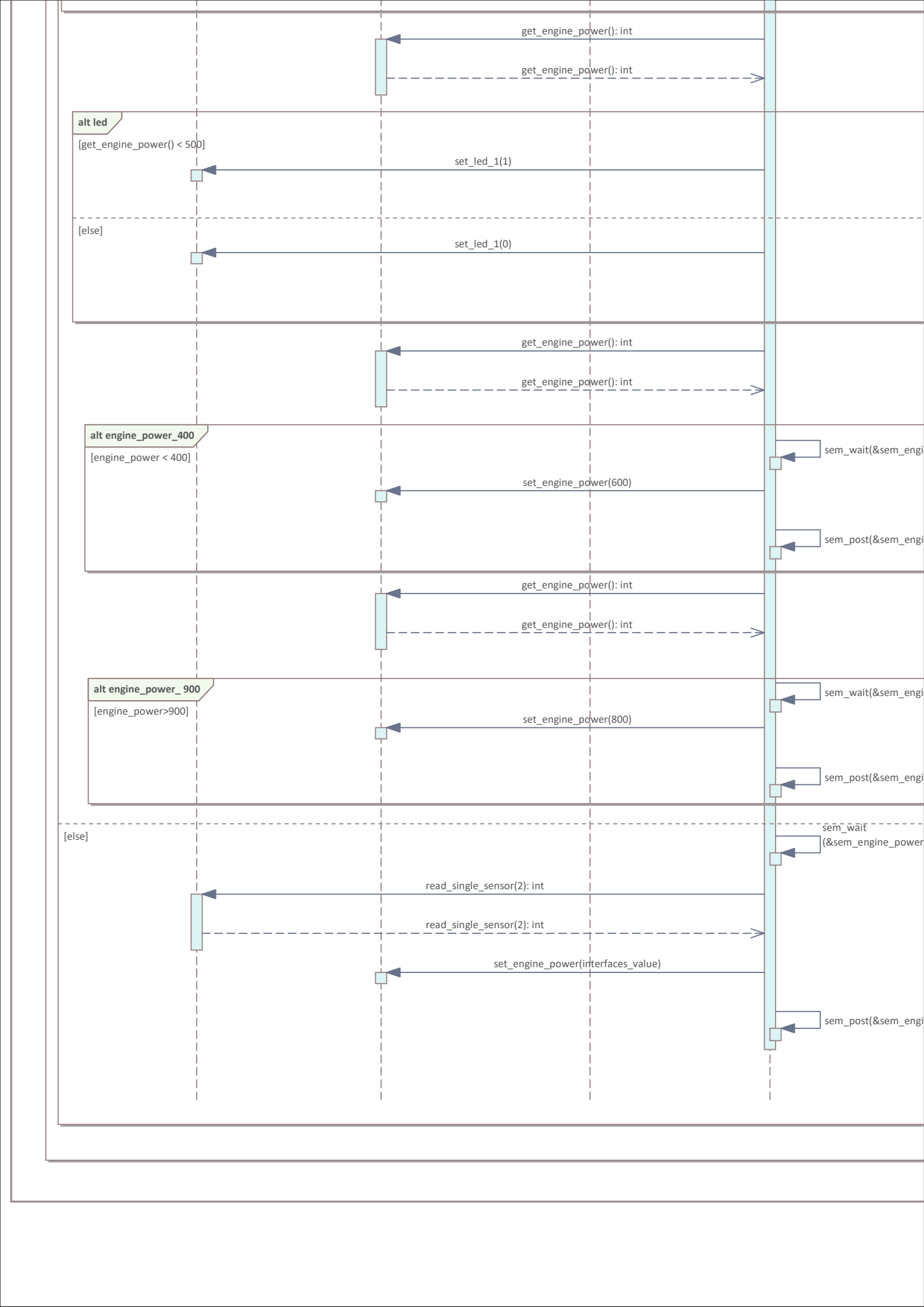

uc [package] UseCase Pkg [UseCase Pkg]
Visualize current mode (automatic/manual)
Visualize warnings
Visualize obstacle warnings
Activate Automatic Mode
Visualize altitude warnings
Pilot
Activate Manual Mode
Visualize engine power warnings
Visualize altitude
Execute pitch maneuver
Execute maneuver
Execute roll maneuver
Visualize engine power
Visualize indicated engine power
Visualize real engine power
Visualize aircraft position
Visualize joystick position

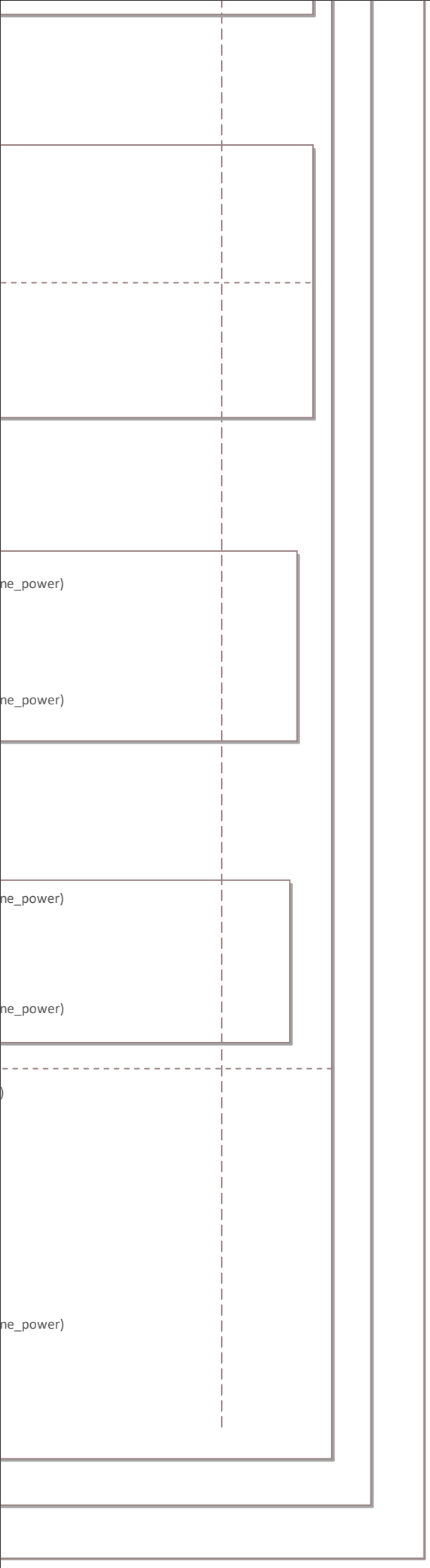

bdd [package] Structure Pkg [Structure Pkg]
«block»
T2-DetectObstacle
+ t2(): void*
+ warnPilotObstacleDetected(): void
«block»
DataAltitudeObstacle
+ get_altitude(): int
+ get_obstacle(): int
+ set_altitude(int): void
+ set_obstacle(int): void
«block»
T1-AltitudeControl
+ alertPilotDescending(): void
+ displayAltitude(): void
+ t1(): void*
«block»
ObstacleDetectorInfrared
+ read_infrared(): int
«block»
AltimeterPotentiometer
+ read_AnalogCh3(): int
«block»
RedLed
+ set_led_1(int):
«block»
VisibilityLightSensor
+ read_AnalogCh0(): int
«block»
YellowLed
+ set_led_2(int): int

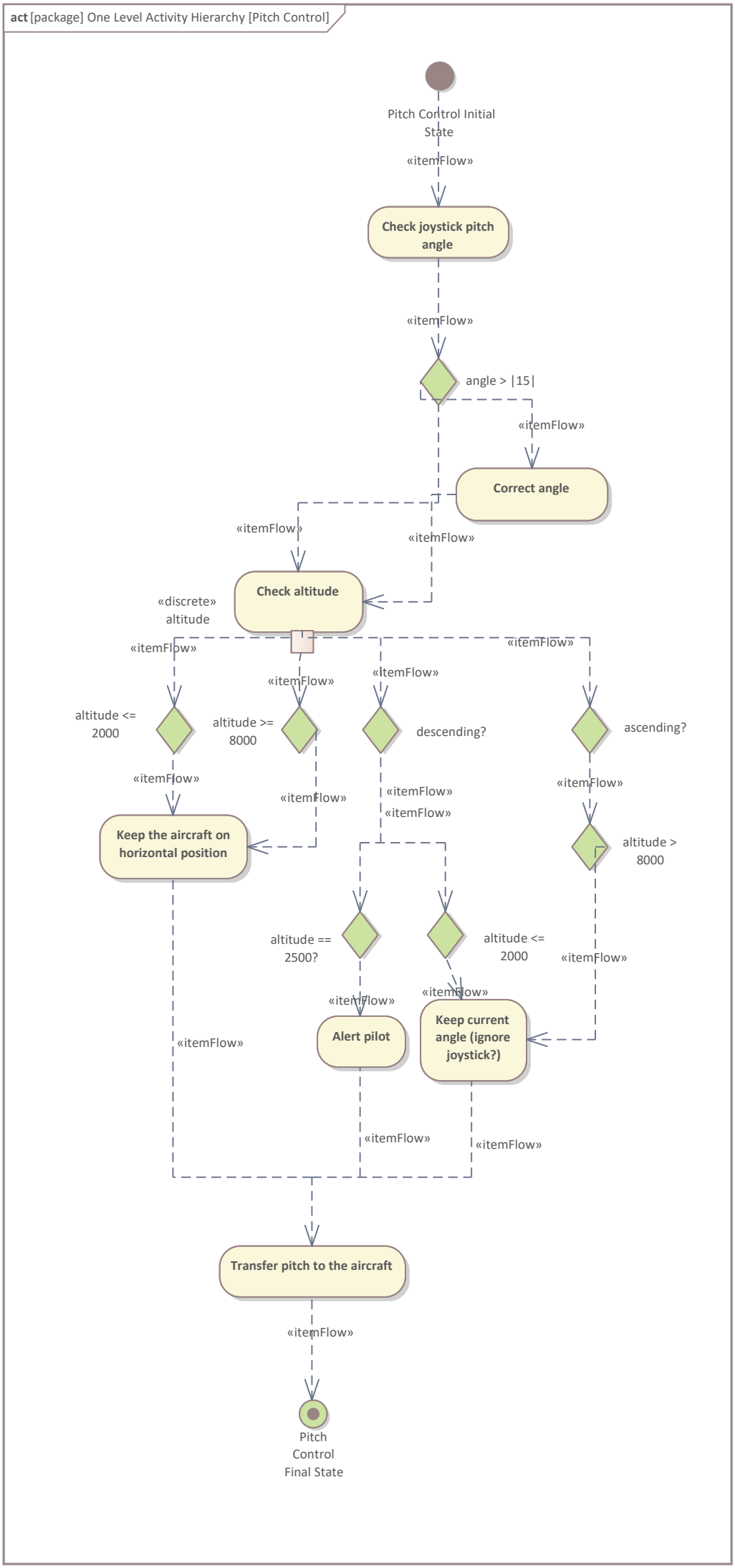

«block»
T3-Speed
+ displayIndicatedEnginePower(int): void
+ displayTransferedEnginePower(int): void
+ t3(): void*
«block»
PotentiometerSpeedometer
+ read_AnalogCh2(): void
«block»
T4-Joystick
+ displayAircraftPosition(int, int): void
+ displayJoystickPostion(int, int): void
+ t4(): void*
«block»
JoystickGyroscope
+ Leer_X_Giroscopo(): int
+ Leer_Y_Giroscopo(): int
«block»
DataSpeed
+ get_aircraft_speed(): int
+ get_alt_speed(): int
+ get_engine_power(): int
+ set_aircraft_speed(int): void
+ set_alt_speed(int): void
+ set_engine_power(int): void
«block»
DataJoystick
+ get_inclination_x(): int
+ get_inclination_y(): int
+ getRealX(): int
+ getRealY(): int
+ set_inclination_x(int): void
+ set_inclination_y(int): void
+ setRealX(int): void
+ setRealY(int): void
«block»
T5-EngineControl
+ alertPilot(): void
+ t5(): void*
«block»
T6-Mode
+ get_mode(): int
+ indicatedCurrentMode(): void
+ t6(): void*
«block»
ModeButton
+ read_button(): int


sd [package] One Level Activity Hierarchy [Engine Control]
Interfaces
DataSpeed
DataJoystick
T5-Engine Control
loop
getM
getM
alt mode
[mode == 0]
sem_wait(&sem_engi
read_single_sensor(2): int
read_single_sensor(2): int
set_engine_power(interfaces_value)
sem_post(&sem_engi
get_inclination_x(): int
get_inclination_x(): int
get_inclination_Y(): int
get_inclination_y(): int
get_engine_power(): int
get_engine_power()
alt inclination
sem_wait(&sem_engi
[(get_inclination_x() > 1 || get_inclination_x() < -1) && (get_inclination_y() > 1 || get_inclination_y() < -1)]
get_engine_power(): int
getEnginePower(): int
set_engine_power(engine_power + 250)
sem_post(&sem_engi
[get_inclination_x() > 1 || get_inclination_x() < -1]
sem_wait(&sem_engi
get_engine_power(): int
get_engine_power(): int
set_engine_power(engine_power+150)
sem_post(&sem_engi
[(get_inclination_y() > 1 || get_inclination_y() < -1) && (get_engine_power() < 600)]
sem_wait(&sem_engi
get_engine_power(): int
get_engine_power(): int
set_engine_power(engine_power+100)
sem_post(&sem_engi


T6-Mode
ode(): int
ode():
nt
ne_power)
ne_power)
ne_power)
ne_power)
ne_power)
ne_power)
ne_power)
ne_power)


get_engine_power(): int
get_engine_power(): int
alt led
[get_engine_power() < 500]
set_led_1(1)
[else]
set_led_1(0)
get_engine_power(): int
get_engine_power(): int
alt engine_power_400
[engine_power < 400]
sem_wait(&sem_engi
set_engine_power(600)
sem_post(&sem_engi
get_engine_power(): int
get_engine_power(): int
alt engine_power_ 900
[engine_power>900]
sem_wait(&sem_engi
set_engine_power(800)
sem_post(&sem_engi
[else]
sem_wait
(&sem_engine_power
read_single_sensor(2): int
read_single_sensor(2): int
set_engine_power(interfaces_value)
sem_post(&sem_engi


ne_power)
ne_power)
ne_power)
ne_power)
)
ne_power)


act [package] One Level Activity Hierarchy [Pitch Control]
Pitch Control Initial State
«itemFlow»
Check joystick pitch angle
«itemFlow»
angle > |15|
«itemFlow»
Correct angle
«itemFlow»
«itemFlow»
Check altitude
«discrete» altitude
«itemFlow»
«itemFlow»
«itemFlow»
«itemFlow»
altitude <= 2000
altitude >= 8000
descending?
ascending?
«itemFlow»
«itemFlow»
«itemFlow»
«itemFlow»
«itemFlow»
Keep the aircraft on horizontal position
altitude > 8000
altitude == 2500?
altitude <= 2000
«itemFlow»
«itemFlow»
«itemFlow»
Alert pilot
Keep current angle (ignore joystick?)
«itemFlow»
«itemFlow»
«itemFlow»
Transfer pitch to the aircraft
«itemFlow»
Pitch Control Final State