FMU Requirements

# ISM Logic

## IN\_FUEL\_TANK\_TEMP

**[FMUREQ-1776]:** FMU shall set the **FUEL\_TANK\_TEMP\_VALID** to Trueand set **FUEL\_TANK\_TEMP** to the value of ***IN\_FUEL\_TANK\_TEMP***if the following evaluates to True:  
  
(  
 (***IN\_FUEL\_TANK\_TEMP\_ACTIVE***is equal to True) AND  
 (  
 ( ***IN\_FUEL\_TANK\_TEMP*** is greater than or equal to *IN\_FUEL\_TANK\_TEMP\_ICD\_MIN) AND* ( ***IN\_FUEL\_TANK\_TEMP*** is less than or equal to *IN\_FUEL\_TANK\_TEMP\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **FUEL\_TANK\_TEMP\_VALID** to False and set **FUEL\_TANK\_TEMP** to the default value of ***IN\_FUEL\_TANK\_TEMP\_MIN*** *(0 Deg C).*

## IN\_PBIT\_STATUS

**[FMUREQ-1780]:** FMU shall set the **PBIT\_STATUS\_VALID** to Trueand set **PBIT\_STATUS** to the value of ***IN\_PBIT\_STATUS***if the following evaluates to True:  
  
(  
 (***IN\_PBIT\_STATUS\_ACTIVE***is equal to True)  
)

**[FMUREQ-1781]:** FMU shall set the **PBIT\_STATUS\_VALID** to Falseand set **PBIT\_STATUS** to False if the following evaluates to True:  
  
(  
 (***IN\_PBIT\_STATUS\_ACTIVE***is equal to False)  
)

## IN\_CTR\_TK\_FUEL\_LEVEL

**[FMUREQ-1782]:** FMU shall set the [**CTR\_TK\_FUEL\_LEVEL\_VALID**](#kix.1rdc4y61wh23)to Trueand set **CTR\_TK\_FUEL\_LEVEL** to the value of ***IN\_CTR\_TK\_FUEL\_LEVEL***if the following evaluates to True:  
  
**(  
 (*IN\_CTR\_TK\_FUEL\_LEVEL\_ACTIVE***is equal to True) AND  
 (  
 ( ***IN\_CTR\_TK\_FUEL\_LEVEL*** is greater than or equal to *IN\_CTR\_TK\_FUEL\_LEVEL\_ICD\_MIN) OR* ( ***IN\_CTR\_TK\_FUEL\_LEVEL*** is less than or equal to *IN\_CTR\_TK\_FUEL\_LEVEL\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set the **CTR\_TK\_FUEL\_LEVEL\_VALID** to False and set **CTR\_TK\_FUEL\_LEVEL** to *IN\_FUEL\_LEVEL\_ICD\_MIN*.

## IN\_RECORDED\_SPEED

**[FMUREQ-1784]:** FMU shall set the **RECORDED\_SPEED\_VALID** to Trueand set **RECORDED\_SPEED** to the value of ***IN\_RECORDED\_SPEED***if the following evaluates to True:  
  
**(  
 (*IN\_RECORDED\_SPEED\_ACTIVE***is equal to True) AND  
 (  
 (***IN\_RECORDED\_SPEED*** is greater than or equal to *IN\_RECORDED\_SPEED\_ICD\_MIN) OR* (***IN\_RECORDED\_SPEED*** is less than or equal to *IN\_RECORDED\_SPEED\_ICD\_MAX)*  )  
)  
  
Otherwise, set **RECORDED\_SPEED\_VALID** to False and set **RECORDED\_SPEED** to *IN\_RECORDED\_SPEED\_ICD\_MIN*.

## IN\_GROUND\_SPEED

**[FMUREQ-1790]:** FMU shall set the **GROUND\_SPEED\_VALID** to Trueand set **GROUND\_SPEED** to the value of ***IN\_GROUND\_SPEED***if the following evaluates to True:

(  
 (***IN\_GROUND\_SPEED\_ACTIVE***is equal to True) AND  
 (  
 (***IN\_GROUND\_SPEED*** is greater than or equal to *IN\_GROUND\_SPEED\_ICD\_MIN) AND* (***IN\_GROUND\_SPEED*** is less than or equal to *IN\_GROUND\_SPEED\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **GROUND\_SPEED\_VALID** to False and set **GROUND\_SPEED** to the default value of *IN\_GROUND\_SPEED\_ICD\_MIN**(0 mph).*

## IN\_BRAKE\_SENSOR\_STATUS

**[FMUREQ-1791]:** FMU shall set **BRAKE\_SENSOR\_STATUS\_VALID** to True and set **BRAKE\_SENSOR\_STATUS** to the value of ***IN\_BRAKE\_SENSOR\_STATUS*** if the following evaluates to True:  
  
(  
 (***IN\_BRAKE\_SENSOR\_STATUS\_ACTIVE***is equal to True)  
)  
  
Otherwise, set **BRAKE\_SENSOR\_STATUS\_VALID** to False and set **BRAKE\_SENSOR\_STATUS** to *BRAKES\_NOT\_APPLIED*.

## IN\_ON\_GROUND

**[FMUREQ-1794]:** FMU shall set **ON\_GROUND\_VALID** to True and set **ON\_GROUND** to the value of ***IN\_ON\_GROUND*** if the following evaluates to True:  
  
(  
 (***IN\_ON\_GROUND\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **ON\_GROUND\_VALID** to False and set **ON\_GROUND** to False.

## IN\_ENGINE\_TEMP\_Sx

**[FMUREQ-1792]:** FMU shall set the **ENGINE\_TEMP\_VALID\_S1** to Trueand set **ENGINE\_TEMP\_S1** to the value of ***IN\_ENGINE\_TEMP\_S1***if the following evaluates to True:  
  
**(  
 (*IN\_ENGINE\_TEMP\_ACTIVE\_S1***is equal to True) AND  
 (  
 (***IN\_ENGINE\_TEMP\_S1*** is greater than or equal to *IN\_ENGINE\_TEMP\_ICD\_MIN) AND* (***IN\_ENGINE\_TEMP\_S1*** is less than or equal to *IN\_ENGINE\_TEMP\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **ENGINE\_TEMP\_VALID\_S1** to False and set **ENGINE\_TEMP\_S1** to the default value of *IN\_ENGINE\_TEMP\_MIN**(0 Deg C).*

**[FMUREQ-1793]:** FMU shall set the **ENGINE\_TEMP\_VALID\_S2** to Trueand set **ENGINE\_TEMP\_S2** to the value of ***IN\_ENGINE\_TEMP\_S2***if the following evaluates to True:  
  
(  
 (***IN\_ENGINE\_TEMP\_ACTIVE\_S2***is equal to True) AND  
 (  
 (***IN\_ENGINE\_TEMP\_S2***is greater than or equal to *IN\_ENGINE\_TEMP\_ICD\_MIN) AND* (***IN\_ENGINE\_TEMP\_S2*** is less than or equal to *IN\_ENGINE\_TEMP\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **ENGINE\_TEMP\_VALID\_S2** to False and set **ENGINE\_TEMP\_S2** to the default value of *IN\_ENGINE\_TEMP\_MIN**(0 Deg C).*

## IN\_FAULT\_IGNORE

[**FMUREQ-1798**]: FMU shall set **FAULT\_IGNORE\_VALID** to True and set **FAULT\_IGNORE** to the value of ***IN\_FAULT\_IGNORE*** if the following logic evaluates to True:  
  
(  
 (***IN\_FAULT\_IGNORE\_ACTIV*E** is equal to True)  
)  
  
Otherwise, set **FAULT\_IGNORE\_VALID** to False and set **FAULT\_IGNORE** to False.

## IN\_FLIGHT\_PHASE

**[FMUREQ-1806]:** FMU software shall set **FLIGHT\_PHASE\_VALID** to True and set **FLIGHT\_PHASE** to the value of ***IN\_FLIGHT\_PHASE*** if the following logic evaluates to True:  
  
(  
 (***IN\_FLIGHT\_PHASE\_ACTIVE*** is equal to True) AND  
 (  
 (***IN\_FLIGHT\_PHASE*** is greater than or equal to *IN\_FLIGHT\_PHASE\_ICD\_MIN*) AND  
 (***IN\_FLIGHT\_PHASE*** is less than or equal to *IN\_FLIGHT\_PHASE\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **FLIGHT\_PHASE\_VALID** to False and maintain the previous value of **FLIGHT\_PHASE**.  
  
Note: Flight phase key  
 1. Pre-flight  
 2. Taxi  
 3. Takeoff  
 4. Climb  
 5. Cruise  
 6. Descent  
 7. Landing  
 8. Post-flight

## IN\_CTR\_TK\_MAN\_SWITCH

**[FMUREQ-1809]:** FMU Software shall set **CTR\_TK\_MAN\_SWITCH\_VALID** to True and set **CTR\_TK\_MAN\_SWITCH** to the value of ***IN\_CTR\_TK\_MAN\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_CTR\_TK\_MAN\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **CTR\_TK\_MAN\_SWITCH\_VALID** to False and maintain the previous value of **CTR\_TK\_MAN\_SWITCH**.

## IN\_CTR\_TK\_PUMP\_x\_SWITCH

**[FMUREQ-1810]:** FMU Software shall set **CTR\_TK\_PUMP\_1\_SWITCH\_VALID** to True and set **CTR\_TK\_PUMP\_1\_SWITCH** to the value of ***IN\_CTR\_TK\_PUMP\_1\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_CTR\_TK\_PUMP\_1\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_1\_SWITCH\_VALID** to False and maintain the previous value of **CTR\_TK\_PUMP\_1\_SWITCH**.

**[FMUREQ-1811]:** FMU Software shall set **CTR\_TK\_PUMP\_2\_SWITCH\_VALID** to True and set **CTR\_TK\_PUMP\_2\_SWITCH** to the value of ***IN\_CTR\_TK\_PUMP\_2\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_CTR\_TK\_PUMP\_2\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_2\_SWITCH\_VALID** to False and maintain the previous value of **CTR\_TK\_PUMP\_2\_SWITCH**.

## IN\_CTR\_TK\_LOW\_LEVEL

**[FMUREQ-1814]** FMU Software shall set **CTR\_TK\_LOW\_LEVEL\_VALID** to True and set **CTR\_TK\_LOW\_LEVEL** to the value of ***IN\_CTR\_TK\_LOW\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_CTR\_TK\_LOW\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **CTR\_TK\_LOW\_LEVEL\_VALID** to False and maintain the previous value of **CTR\_TK\_LOW\_LEVEL**.

## IN\_x\_TK\_LOW\_LEVEL

**[FMUREQ-1815]** FMU Software shall set **L\_TK\_LOW\_LEVEL\_VALID** to True and set **L\_TK\_LOW\_LEVEL** to the value of ***IN\_L\_TK\_LOW\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_TK\_LOW\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_TK\_LOW\_LEVEL\_VALID** to False and maintain the previous value of **L\_TK\_LOW\_LEVEL**.

**[FMUREQ-1816]** FMU Software shall set **R\_TK\_LOW\_LEVEL\_VALID** to True and set **R\_TK\_LOW\_LEVEL** to the value of ***IN\_R\_TK\_LOW\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_TK\_LOW\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_TK\_LOW\_LEVEL\_VALID** to False and maintain the previous value of **R\_TK\_LOW\_LEVEL**.

## IN\_x\_TK\_HI\_LEVEL

**[FMUREQ-1817]** FMU Software shall set **L\_TK\_HI\_LEVEL\_VALID** to True and set **L\_TK\_HI\_LEVEL** to the value of ***IN\_L\_TK\_HI\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_TK\_HI\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_TK\_HI\_LEVEL\_VALID** to False and maintain the previous value of **L\_TK\_HI\_LEVEL**.

**[FMUREQ-1818]** FMU Software shall set **R\_TK\_HI\_LEVEL\_VALID** to True and set **R\_TK\_HI\_LEVEL** to the value of ***IN\_R\_TK\_HI\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_TK\_HI\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_TK\_HI\_LEVEL\_VALID** to False and maintain the previous value of **R\_TK\_HI\_LEVEL**.

**[FMUREQ-1876]** FMU Software shall set **CTR\_TK\_HI\_LEVEL\_VALID** to True and set **CTR\_TK\_HI\_LEVEL** to the value of ***IN\_CTR\_TK\_HI\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_CTR\_TK\_HI\_LEVEL\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **CTR\_TK\_HI\_LEVEL\_VALID** to False and maintain the previous value of **CTR\_TK\_HI\_LEVEL**.

## IN\_x\_TK\_PUMP\_y\_SWITCH

**[FMUREQ-1819]:** FMU Software shall set **L\_TK\_PUMP\_1\_SWITCH\_VALID** to True and set **L\_TK\_PUMP\_1\_SWITCH** to the value of ***IN\_L\_TK\_PUMP\_1\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_TK\_PUMP\_1\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_TK\_PUMP\_1\_SWITCH\_VALID** to False and maintain the previous value of **L\_TK\_PUMP\_1\_SWITCH**.

**[FMUREQ-1820]:** FMU Software shall set **L\_TK\_PUMP\_2\_SWITCH\_VALID** to True and set **L\_TK\_PUMP\_2\_SWITCH** to the value of ***IN\_L\_TK\_PUMP\_2\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_TK\_PUMP\_2\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_TK\_PUMP\_2\_SWITCH\_VALID** to False and maintain the previous value of **L\_TK\_PUMP\_2\_SWITCH**.\

**[FMUREQ-1821]:** FMU Software shall set **R\_TK\_PUMP\_1\_SWITCH\_VALID** to True and set **R\_TK\_PUMP\_1\_SWITCH** to the value of ***IN\_R\_TK\_PUMP\_1\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_TK\_PUMP\_1\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_TK\_PUMP\_1\_SWITCH\_VALID** to False and maintain the previous value of **R\_TK\_PUMP\_1\_SWITCH**.

**[FMUREQ-1822]:** FMU Software shall set **R\_TK\_PUMP\_2\_SWITCH\_VALID** to True and set **R\_TK\_PUMP\_2\_SWITCH** to the value of ***IN\_R\_TK\_PUMP\_2\_SWITCH*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_TK\_PUMP\_2\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_TK\_PUMP\_2\_SWITCH\_VALID** to False and maintain the previous value of **R\_TK\_PUMP\_2\_SWITCH**.

## IN\_XFEED\_VALVE\_SWITCH

**[FMUREQ-1827]** FMU Software shall set **XFEED\_VALVE\_SWITCH\_VALID** to True and set **XFEED\_VALVE\_SWITCH** to the value of ***IN\_XFEED\_VALVE\_SWITCH***if the following logic evaluates to True:  
  
(  
 (***IN\_XFEED\_VALVE\_SWITCH\_ACTIVE***is equal to True)  
)  
  
Otherwise, set **XFEED\_VALVE\_SWITCH\_VALID** to True and maintain the previous value of **XFEED\_VALVE\_SWITCH**.

## IN\_XFEED\_VALVE\_FULLY\_CLOSED

**[FMUREQ-1830]:** FMU Software shall set **XFEED\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **XFEED\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_XFEED\_VALVE\_FULLY\_CLOSED***if the following logic evaluates to True:  
  
(  
 (***IN\_XFEED\_VALVE\_FULLY\_CLOSED****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **XFEED\_VALVE\_FULLY\_CLOSED\_VALID** to True and maintain the previous value of **XFEED\_VALVE\_FULLY\_CLOSED**.

## IN\_XFEED\_VALVE\_FULLY\_OPEN

**[FMUREQ-1831]:** FMU Software shall set **XFEED\_VALVE\_FULLY\_OPEN\_VALID** to True and set **XFEED\_VALVE\_FULLY\_OPEN** to the value of ***IN\_XFEED\_VALVE\_FULLY\_OPEN***if the following logic evaluates to True:  
  
(  
 (***IN\_XFEED\_VALVE\_FULLY\_OPEN****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **XFEED\_VALVE\_FULLY\_OPEN\_VALID** to True and maintain the previous value of **XFEED\_VALVE\_FULLY\_OPEN**.

## IN\_FUEL\_BALANCE\_PB

**[FMUREQ-1832]:** FMU Software shall set **FUEL\_BALANCE\_PB\_VALID** to True and set **FUEL\_BALANCE\_PB** to the value of***IN\_FUEL\_BALANCE\_PB*** if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_BALANCE\_PB\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_PB\_VALID** to False and maintain the previous value of **FUEL\_BALANCE\_PB**.

## IN\_x\_TK\_FUEL\_LEVEL

**[FMUREQ-1836]:** FMU Software shall set **L\_TK\_FUEL\_LEVEL\_VALID** to True and set **L\_TK\_FUEL\_LEVEL** to the value of ***IN\_L\_TK\_FUEL\_LEVEL***if the following logic evaluates to True:  
  
(  
 (***IN\_L\_TK\_FUEL\_LEVEL\_ACTIVE*** is equal to True) AND  
 (  
 (***IN\_L\_TK\_FUEL\_LEVEL*** is greater than or equal to *WING\_TK\_FUEL\_LEVEL\_ICD\_MIN*) AND  
 (***IN\_L\_TK\_FUEL\_LEVEL*** is less than or equal to *WING\_TK\_FUEL\_LEVEL\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **L\_TK\_FUEL\_LEVEL\_VALID** to False and maintain the previous value of **L\_TK\_FUEL\_LEVEL**.

**[FMUREQ-1837]:** FMU Software shall set **R\_TK\_FUEL\_LEVEL\_VALID** to True and set **R\_TK\_FUEL\_LEVEL** to the value of ***IN\_R\_TK\_FUEL\_LEVEL***if the following logic evaluates to True:  
  
(  
 (***IN\_R\_TK\_FUEL\_LEVEL\_ACTIVE*** is equal to True) AND  
 (  
 (***IN\_R\_TK\_FUEL\_LEVEL*** is greater than or equal to *WING\_TK\_FUEL\_LEVEL\_ICD\_MIN*) AND  
 (***IN\_R\_TK\_FUEL\_LEVEL*** is less than or equal to *WING\_TK\_FUEL\_LEVEL\_ICD\_MAX*)  
 )  
)  
  
Otherwise, set **R\_TK\_FUEL\_LEVEL\_VALID** to False and maintain the previous value of **R\_TK\_FUEL\_LEVEL**.

## IN\_ENGx\_RUNNING

**[FMUREQ-1842]:** FMU Software shall set **ENG1\_RUNNING\_VALID** to True and set **ENG1\_RUNNING** to the value of***IN\_ENG1\_RUNNING*** if the following logic evaluates to True:  
  
(  
 (***IN\_ENG1\_RUNNING\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **ENG1\_RUNNING\_VALID** to False and maintain the previous value of **ENG1\_RUNNING**.

**[FMUREQ-1843]:** FMU Software shall set **ENG2\_RUNNING\_VALID** to True and set **ENG2\_RUNNING** to the value of***IN\_ENG2\_RUNNING*** if the following logic evaluates to True:  
  
(  
 (***IN\_ENG2\_RUNNING\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **ENG2\_RUNNING\_VALID** to False and maintain the previous value of **ENG2\_RUNNING**.

## IN\_APU\_RUNNING

**[FMUREQ-1851]:** FMU Software shall set **APU\_RUNNING\_VALID** to True and set **APU\_RUNNING** to the value of***IN\_APU\_RUNNING*** if the following logic evaluates to True:  
  
(  
 (***IN\_APU\_RUNNING\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **APU\_RUNNING\_VALID** to False and maintain the previous value of **APU\_RUNNING**.

## IN\_FUEL\_BALANCE\_VALVE\_x\_FULLY\_OPEN

**[FMUREQ-1844]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN\_VALID** to True and set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN** to the value of ***IN\_FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN\_VALID** to True and maintain the previous value of **FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN**.

**[FMUREQ-1845]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN\_VALID** to True and set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN** to the value of ***IN\_FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN\_VALID** to True and maintain the previous value of **FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN**.

## IN\_FUEL\_BALANCE\_VALVE\_x\_FULLY\_CLOSED

**[FMUREQ-1846]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED\_VALID** to True and set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED** to the value of ***IN\_FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED\_VALID** to True and maintain the previous value of **FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED**.

**[FMUREQ-1847]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED\_VALID** to True and set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED** to the value of ***IN\_FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED****\_****ACTIVE***is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED\_VALID** to True and maintain the previous value of **FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED**.

## IN\_APU\_MASTER\_SWITCH

**[FMUREQ-1852]:** FMU software shall set **APU\_MASTER\_SWITCH\_VALID** to True and set **APU\_MASTER\_SWITCH** to the value of ***IN\_APU\_MASTER\_SWITCH***if the following logic evaluates to True:  
  
(  
 (***IN\_APU\_MASTER\_SWITCH\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **APU\_MASTER\_SWITCH\_VALID** to False and maintain the previous value of **APU\_MASTER\_SWITCH.**

## IN\_APU\_FIRE

**[FMUREQ-1853]:** FMU software shall set **APU\_FIRE\_VALID** to True and set **APU\_FIRE** to the value of ***IN\_APU\_FIRE***if the following logic evaluates to True:  
  
(  
 (***IN\_APU\_FIRE\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **APU\_FIRE\_VALID** to False and maintain the previous value of **APU\_FIRE**.

## IN\_APU\_LP\_VALVE\_FULLY\_OPEN

**[FMUREQ-1854]:** FMU software shall set **APU\_LP\_VALVE\_FULLY\_OPEN\_VALID** to True and set **APU\_LP\_VALVE\_FULLY\_OPEN** to the value of ***IN\_APU\_LP\_VALVE\_FULLY\_OPEN***if the following logic evaluates to True:  
  
(  
 (***IN\_APU\_LP\_VALVE\_FULLY\_OPEN\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **APU\_LP\_VALVE\_FULLY\_OPEN\_VALID** to False and maintain the previous value of **APU\_LP\_VALVE\_FULLY\_OPEN**.

## IN\_APU\_LP\_VALVE\_FULLY\_CLOSED

**[FMUREQ-1855]:** FMU software shall set **APU\_LP\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **APU\_LP\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_APU\_LP\_VALVE\_FULLY\_CLOSED***if the following logic evaluates to True:  
  
(  
 (***IN\_APU\_LP\_VALVE\_FULLY\_CLOSED\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **APU\_LP\_VALVE\_FULLY\_CLOSED\_VALID** to False and maintain the previous value of **APU\_LP\_VALVE\_FULLY\_CLOSED**.

## IN\_x\_LP\_MFLD\_PRESSURE

**[FMUREQ-1859]:** FMU shall set **L\_LP\_MFLD\_PRESSURE\_VALID** to True and set **L\_LP\_MFLD\_PRESSURE** to the value of ***IN\_L\_LP\_MFLD\_PRESSURE*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_LP\_MFLD\_PRESSURE\_ACTIVE*** is equal to True) AND  
 (***IN\_L\_LP\_MFLD\_PRESSURE*** is greater than or equal to *LP\_MFLD\_PRESSURE\_ICD\_MIN*) AND  
 (***IN\_L\_LP\_MFLD\_PRESSURE*** is less than or equal to *LP\_MFLD\_PRESSURE\_ICD\_MAX*)  
)  
  
Otherwise, set **L\_LP\_MFLD\_PRESSURE\_VALID** to False and maintain the previous value of **L\_LP\_MFLD\_PRESSURE**.

**[FMUREQ-1860]:** FMU shall set **R\_LP\_MFLD\_PRESSURE\_VALID** to True and set **R\_LP\_MFLD\_PRESSURE** to the value of ***IN\_R\_LP\_MFLD\_PRESSURE*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_LP\_MFLD\_PRESSURE\_ACTIVE*** is equal to True) AND  
 (***IN\_R\_LP\_MFLD\_PRESSURE*** is greater than or equal to *LP\_MFLD\_PRESSURE\_ICD\_MIN*) AND  
 (***IN\_R\_LP\_MFLD\_PRESSURE*** is less than or equal to *LP\_MFLD\_PRESSURE\_ICD\_MAX*)  
)  
  
Otherwise, set **R\_LP\_MFLD\_PRESSURE\_VALID** to False and maintain the previous value of **R\_LP\_MFLD\_PRESSURE**.

## IN\_x\_REFUEL\_VALVE\_FULLY\_OPEN

**[FMUREQ-1865]:** FMU software shall set **L\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to True and set **L\_REFUEL\_VALVE\_FULLY\_OPEN** to the value of ***IN\_L\_REFUEL\_VALVE\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_REFUEL\_VALVE\_FULLY\_OPEN\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to False and maintain the previous value of **L\_REFUEL\_VALVE\_FULLY\_OPEN**.

**[FMUREQ-1866]:** FMU software shall set **R\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to True and set **R\_REFUEL\_VALVE\_FULLY\_OPEN** to the value of ***IN\_R\_REFUEL\_VALVE\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_REFUEL\_VALVE\_FULLY\_OPEN\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to False and maintain the previous value of **R\_REFUEL\_VALVE\_FULLY\_OPEN**.

**[FMUREQ-1867]:** FMU software shall set **C\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to True and set **C\_REFUEL\_VALVE\_FULLY\_OPEN** to the value of ***IN\_C\_REFUEL\_VALVE\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_C\_REFUEL\_VALVE\_FULLY\_OPEN\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **C\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** to False and maintain the previous value of **C\_REFUEL\_VALVE\_FULLY\_OPEN**.

## IN\_x\_REFUEL\_VALVE\_FULLY\_CLOSED

**[FMUREQ-1868]:** FMU software shall set **L\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **L\_REFUEL\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_L\_REFUEL\_VALVE\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_L\_REFUEL\_VALVE\_FULLY\_CLOSED\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to False and maintain the previous value of **L\_REFUEL\_VALVE\_FULLY\_CLOSED**.

**[FMUREQ-1869]:** FMU software shall set **R\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **R\_REFUEL\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_R\_REFUEL\_VALVE\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_R\_REFUEL\_VALVE\_FULLY\_CLOSED\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **R\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to False and maintain the previous value of **R\_REFUEL\_VALVE\_FULLY\_CLOSED**.

**[FMUREQ-1870]:** FMU software shall set **C\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **C\_REFUEL\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_C\_REFUEL\_VALVE\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_C\_REFUEL\_VALVE\_FULLY\_CLOSED\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **C\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to False and maintain the previous value of **C\_REFUEL\_VALVE\_FULLY\_CLOSED**.

## IN\_x\_TK\_REFUEL\_SWITCH

**[FMUREQ-1871]:** FMU software shall set **TK\_REFUEL\_SW\_VALID** to True and set the following signals:  
  
**L\_TK\_REFUEL\_SW** to ***IN\_L\_TK\_REFUEL\_SW*R\_TK\_REFUEL\_SW** to ***IN\_R\_TK\_REFUEL\_SW*C\_TK\_REFUEL\_SW** to ***IN\_C\_TK\_REFUEL\_SW***if the following logic evaluates to True:  
  
(  
 (***IN\_TK\_REFUEL\_SW\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **L\_TK\_REFUEL\_SW\_VALID** to False and maintain the previous value of **L\_TK\_REFUEL\_SW**.

## IN\_FUEL\_XFER\_MODE

**[FMUREQ-1872]:** FMU software shall set **FUEL\_XFER\_MODE\_VALID** to True and set **FUEL\_XFER\_MODE** to the value of ***IN\_FUEL\_XFER\_MODE***if the following logic evaluates to True:  
  
(  
 (***IN\_FUEL\_XFER\_MODE\_ACTIVE*** is equal to True) AND  
 (  
 (***IN\_FUEL\_XFER\_MODE*** is equal to XM*\_OFF*) OR  
 (***IN\_FUEL\_XFER\_MODE*** is equal to XM*\_MAN*) OR  
 (***IN\_FUEL\_XFER\_MODE*** is equal to XM*\_AUTO*) OR  
 (***IN\_FUEL\_XFER\_MODE*** is equal to XM*\_DEFUEL*)  
 )  
)  
  
Otherwise, set **FUEL\_XFER\_MODE\_VALID** to False and set **FUEL\_XFER\_MODE** to *XM\_OFF*.

## IN\_DEFUEL\_VALVE\_FULLY\_OPEN

**[FMUREQ-1873]:** FMU software shall set **DEFUEL\_VALVE\_FULLY\_OPEN\_VALID** to True and set **DEFUEL\_VALVE\_FULLY\_OPEN** to the value of ***IN\_DEFUEL\_VALVE\_FULLY\_OPEN*** if the following logic evaluates to True:  
  
(  
 (***IN\_DEFUEL\_VALVE\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **DEFUEL\_VALVE\_FULLY\_OPEN\_VALID** to False and maintain the previous value of **DEFUEL\_VALVE\_FULLY\_OPEN**.

## IN\_DEFUEL\_VALVE\_FULLY\_CLOSED

**[FMUREQ-1874]:** FMU software shall set **DEFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to True and set **DEFUEL\_VALVE\_FULLY\_CLOSED** to the value of ***IN\_DEFUEL\_VALVE\_FULLY\_CLOSED*** if the following logic evaluates to True:  
  
(  
 (***IN\_DEFUEL\_VALVE\_ACTIVE*** is equal to True)  
)  
  
Otherwise, set **DEFUEL\_VALVE\_FULLY\_CLOSED\_VALID** to False and maintain the previous value of **DEFUEL\_VALVE\_FULLY\_CLOSED**.

## IN\_DESIRED\_FUEL\_LEVEL

**[FMUREQ-1875]:** FMU software shall set **DESIRED\_FUEL\_LEVEL\_VALID** to True and set **DESIRED\_FUEL\_LEVEL** to the value of ***IN\_DESIRED\_FUEL\_LEVEL*** if the following logic evaluates to True:  
  
(  
 (***IN\_DESIRED\_FUEL\_LEVEL\_ACTIVE*** is equal to True) AND  
 (***IN\_DESIRED\_FUEL\_LEVEL***is greater than or equal to *IN\_FUEL\_LEVEL\_ICD\_MIN*) AND  
 (***IN\_DESIRED\_FUEL\_LEVEL***is less than or equal to *IN\_FUEL\_LEVEL\_ICD\_MAX*)  
)  
  
Otherwise, set **DESIRED\_FUEL\_LEVEL\_VALID** to False and maintain the previous value of **DESIRED\_FUEL\_LEVEL**.

# **Monitoring Logic**

## FUEL\_TANK\_OVHT\_LATCHED

**[FMUREQ-1777]:** FMU shall set the **FUEL\_TANK\_OVHT\_LATCHED** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_TANK\_OVHT\_RESET** is equal to False) AND  
 (  
 (**FUEL\_TANK\_TEMP** is greater than *FUEL\_TANK\_TEMP\_OVHT\_TH\_SET* ) AND  
 (**FUEL\_TANK\_TEMP\_VALID** is equal to True)  
 ) is equal to True unchanged for greater than *FUEL\_TANK\_TEMP\_OVHT\_SET\_DELAY*)  
  
otherwise, maintain the previous value of **FUEL\_TANK\_OVHT\_LATCHED** if the following logic evaluates to True:  
  
(  
 (**FUEL\_TANK\_OVHT\_RESET** is equal to False)  
)

**[FMUREQ-1778]:** FMU shall set the **FUEL\_TANK\_OVHT\_LATCHED** to False if the following logic evaluates to True:  
  
(  
 (**FUEL\_TANK\_OVHT\_RESET** is equal to True)  
)

**[FMUREQ-1779]:** FMU shall set the **FUEL\_TANK\_OVHT\_RESET** to True if the following logic evaluates to True:  
  
(  
 (  
 (**FUEL\_TANK\_TEMP** is less than *FUEL\_TANK\_TEMP\_OVHT\_TH\_RESET* ) AND  
 (**FUEL\_TANK\_TEMP\_VALID** is equal to True)  
 ) is equal to True unchanged for greater than *FUEL\_TANK\_TEMP\_OVHT\_RESET\_DELAY*)  
  
otherwise, set **FUEL\_TANK\_OVHT\_RESET** to False.

## FUEL\_LEVEL\_XDCR\_FAULT

**[****FMUREQ-1783]:** FMU shall set the **FUEL\_LEVEL\_XCDR\_FAULT** to Trueif the following evaluates to True:  
  
(  
 ([**CTR\_TK\_FUEL\_LEVEL\_VALID**](#kix.qfpgzigo428u)is equal to False)  
)  
  
Otherwise, set the **FUEL\_LEVEL\_XCDR\_FAULT** to False.

## AUTO\_FAULT\_DETECT\_ENABLED

**[FMUREQ-1799]:** FMU shall set **AUTO\_FAULT\_DETECT\_ENABLED** to Trueif the following evaluates to True:  
  
(  
 (  
 (**PBIT\_STATUS**is equal to True) AND   
 **(PBIT\_STATUS\_VALID**is equal to True)  
 ) AND  
 (  
 (**RECORDED\_SPEED**is greater than *IN\_FLIGHT\_TH*) AND  
 (**RECORDED\_SPEED\_VALID**is equal to True)  
 ) is True for greater than *STABLE\_FLIGHT\_DELAY AND* (  
 (**ON\_GROUND** is equal to False) AND  
 (**ON\_GROUND\_VALID** is equal to True)  
 )  
)  
  
Otherwise, set **AUTO\_FAULT\_DETECT\_ENABLED** to False.

## FAULT\_DETECT\_ENABLED

**[FMUREQ-1785]:** FMU shall set **FAULT\_DETECT\_ENABLED** to True if the following evaluates to True:  
  
(  
 (**AUTO\_FAULT\_DETECT\_ENABLED** is set to True) AND  
 (**FAULT\_IGNORE** is set to False) AND  
 (**FAULT\_IGNORE\_VALID** is set to True)  
) is True for greater than *FAULT\_DETECT\_CONFIRM\_DELAY*Otherwise, set **AUTO\_FAULT\_DETECT\_ENABLED** to False.

## FUEL\_LEVEL\_DETECT\_VALUE\_1

**[FMUREQ-1786]:** FMU shall set the **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1** to **CTR\_TK\_FUEL\_LEVEL** and set **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1\_VALID** to Trueif the following evaluates to True:  
  
(  
 (  
 (**CTR\_TK\_FUEL\_LEVEL\_VALID** is equal to True)AND  
 (**FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2\_VALID** is equal to False**)** ) is equal to True for greater than *LEAK\_DETECTION\_DELAY* AND  
 (**FAULT\_DETECT\_ENABLED** is equal to True)  
)  
  
Otherwise, maintain the previous value **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1** and **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1\_VALID** to False.  
  
**Note: FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1** default value *FUEL\_LEVEL\_MIN = 0*

## FUEL\_LEVEL\_DETECT\_VALUE\_2

**[FMUREQ-1787]:** FMU shall set the **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2** to **CTR\_TK\_FUEL\_LEVEL** and set **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2\_VALID** to Trueif the following evaluates to True:  
  
(  
 (  
 (**CTR\_TK\_FUEL\_LEVEL\_VALID** is equal to True) AND  
 (**FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1\_VALID** is equal to True)  
 )equal to Truefor greater than *LEAK\_DETECTION\_DELAY* AND  
 (**FAULT\_DETECT\_ENABLED** is equal to True)  
)  
  
Otherwise, maintain the previous value of **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2** and **FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2\_VALID** to False.  
**Note: FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2** default value *FUEL\_LEVEL\_MAX = 1000*

## FUEL\_LEVEL\_LEAK

**[FMUREQ-1788]:** FMU shall set the **FUEL\_LEVEL\_LEAK** to Trueif the following evaluates to True:  
  
(  
 ((**FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_2 - FUEL\_LEVEL\_LEAK\_DETECT\_VALUE\_1**)is greater than *LEAK\_DETECT\_THRESHOLD*) AND  
 (**XFFAULT\_DETECT\_ENABLED** is equal to True for greater than *LEAK\_DETECTION\_DELAY2*)  
)  
  
Otherwise, set **FUEL\_LEVEL\_LEAK** to False.

## AC\_NO\_GO

**[FMUREQ-1789]:** FMU shall set the **AC\_NO\_GO** to Trueif the following evaluates to True:  
  
(  
 (**FUEL\_TANK\_OVHT\_LATCHED** is equal to True) OR  
 (**FUEL\_LEVEL\_LEAK** is equal to True**)** OR (**FUEL\_LEVEL\_XDCR\_FAULT** is equal to True) OR  
 (**OVERSPEED\_FAULT** is equal to True)  
)  
  
Otherwise, set **AC\_NO\_GO** to False

## OVERSPEED\_FAULT

**[FMUREQ-1795]:** FMU shall set **OVERSPEED\_FAULT** to True if the following evaluates to True:  
  
(  
 (**BRAKE\_SENSOR\_STATUS** is equal to BRAKES\_APPLIED) AND  
 (**BRAKE\_SENSOR\_STATUS\_VALID** is equal to True) AND  
 (  
 (**RECORDED\_SPEED** is greater than *IN\_FLIGHT\_TH*) AND  
 (**RECORDED\_SPEED\_VALID** is True)  
 ) is equal to True unchanged for greater than *OVERSPEED\_FAULT\_DELAY* AND  
 (**FAULT\_DETECT\_ENABLED** is equal to True) AND  
 (**ON\_GROUND** is equal to True) AND  
 (**ON\_GROUND\_VALID** is equal to True)  
)  
  
Otherwise, set **OVERSPEED\_FAULT** to False.

## AC\_IN\_FLIGHT

**[FMUREQ-1796]:** FMU software shall set **AC\_IN\_FLIGHT** to True if the following evaluates to True:  
  
(  
 (**ON\_GROUND** is equal to False) OR  
 (**ON\_GROUND\_VALID** is equal to True)  
)  
  
Otherwise, set **AC\_IN\_FLIGHT** to False.

## AC\_ON\_GROUND

**[FMUREQ-1797]:** FMU software shall set **AC\_ON\_GROUND** to True if the following evaluates to True:  
  
(  
 (**ON\_GROUND** is equal to True) OR  
 (**ON\_GROUND\_VALID** is equal to True)  
)  
  
Otherwise, set **AC\_ON\_GROUND** to False.

## ISM\_FAULT

**[FMUREQ-1801]:** FMU software shall set **ISM\_FAULT** to True if the following logic evaluates to True:  
  
(  
 (**BRAKE\_SENSOR\_STATUS\_VALID** is equal to False) OR  
 (**ENGINE\_TEMP\_VALID** is equal to False) OR  
 (**FAULT\_IGNORE\_VALID** is equal to False) OR  
 (**CTR\_TK\_FUEL\_LEVEL\_VALID** is equal to False) OR  
 (**FUEL\_TANK\_TEMP\_VALID** is equal to False) OR  
 (**GROUND\_SPEED\_VALID** is equal to False) OR  
 (**ON\_GROUND\_VALID** is equal to False) OR  
 (**PBIT\_STATUS\_VALID** is equal to False) OR  
 (**RECORDED\_SPEED\_VALID** is equal to False)  
) is True for greater than *ISM\_FAULT\_CONFIRM\_DELAY*Otherwise, set **ISM\_FAULT** to False.

## FUEL\_WING\_TK\_LEVEL\_DIFF

**[FMUREQ-1835]:** FMU Software shall set **FUEL\_WING\_TK\_LEVEL\_DIFF** to (ABS(**L\_TK\_FUEL\_LEVEL** - **R\_TK\_FUEL\_LEVEL**)) if the following logic evaluates to True:  
  
(  
 (**L\_TK\_FUEL\_LEVEL\_VALID** is True) AND  
 (**R\_TK\_FUEL\_LEVEL\_VALID** is True)  
)  
  
Otherwise, maintain the previous value of **FUEL\_WING\_TK\_LEVEL\_DIFF**.

## FUEL\_BALANCE\_INHIBIT

**[FMUREQ-1838]:** FMU Software shall set **FUEL\_BALANCE\_INHIBIT** to True if the following logic evaluates to True:  
  
(  
 (**L\_TK\_FUEL\_LEVEL\_VALID** is equal to False) OR  
 (**R\_TK\_FUEL\_LEVEL\_VALID** is equal to False) OR  
 ([REFUELING] is equal to True) OR   
 ([DEFUELING] is equal to True) OR  
 (  
 (**AC\_ON\_GROUND** is equal to True) AND  
 (  
 (  
 (**ENG1\_RUNNING** True) AND  
 (**ENG1\_RUNNING\_VALID** True)  
 ) OR  
 (  
 (**ENG2\_RUNNING** True) AND  
 (**ENG2\_RUNNING\_VALID** True)  
 ) OR  
 (  
 (**APU\_RUNNING** True) AND  
 (**APU\_RUNNING\_VALID** True)  
 )  
 )  
 ) OR  
 (  
 (**XFEED\_VALVE\_FULLY\_CLOSED** is equal to False) AND  
 (**XFEED\_VALVE\_FULLY\_CLOSED\_VALID** is equal to True)  
 )  
)  
  
Otherwise, set **FUEL\_BALANCE\_INHIBIT** to False.

## FUEL\_BALANCE\_COMPLETE

**[FMUREQ-1840]:** FMU Software shall set **FUEL\_BALANCE\_COMPLETE** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_WING\_TK\_LEVEL\_DIFF** is less than or equal to *FUEL\_BALANCE\_COMPLETE\_TH*)  
)  
  
Otherwise, set **FUEL\_BALANCE\_COMPLETE** to False.

## x\_LP\_MFLD\_LOW\_PRESSURE

**[FMUREQ-1861]:** FMU Software shall set **L\_LP\_MFLD\_LOW\_PRESSURE** to True if the following logic evaluates to True:  
  
(  
 (**L\_LP\_MFLD\_LOW\_PRESSURE\_RESET** is equal to False) AND  
 (  
 (**L\_LP\_MFLD\_PRESSURE** is less than or equal to *LP\_LOW\_PRESSURE\_TH*) AND  
 (**L\_LP\_MFLD\_PRESSURE\_VALID** is equal to True)  
 ) is equal to True for greater than *LP\_LOW\_PRESSURE\_DELAY*  
)  
  
Otherwise, maintain the previous value of **L\_LP\_MFLD\_LOW\_PRESSURE** if the following logic evaluates to True:  
  
(  
 (**L\_LP\_MFLD\_LOW\_PRESSURE\_RESET** is equal to False)  
)  
  
Otherwise, set **L\_LP\_MFLD\_LOW\_PRESSURE** to False.  
  
**[FMUREQ-1862]:** FMU Software shall set **L\_LP\_MFLD\_LOW\_PRESSURE** to True if the following logic evaluates to True:  
  
(  
 (**L\_LP\_MFLD\_PRESSURE** is greater than or equal to *LP\_LOW\_PRESSURE\_RESET\_TH*) AND  
 (**L\_LP\_MFLD\_PRESSURE\_VALID** is equal to True)  
) is equal to True for greater than *LP\_LOW\_PRESSURE\_DELAY*  
  
Otherwise, set **L\_LP\_MFLD\_LOW\_PRESSURE** to False

**[FMUREQ-1863]:** FMU Software shall set **R\_LP\_MFLD\_LOW\_PRESSURE** to True if the following logic evaluates to True:  
  
(  
 (**R\_LP\_MFLD\_LOW\_PRESSURE\_RESET** is equal to False) AND  
 (  
 (**R\_LP\_MFLD\_PRESSURE** is less than or equal to *LP\_LOW\_PRESSURE\_TH*) AND  
 (**R\_LP\_MFLD\_PRESSURE\_VALID** is equal to True)  
 ) is equal to True for greater than *LP\_LOW\_PRESSURE\_DELAY*  
)  
  
Otherwise, maintain the previous value of **R\_LP\_MFLD\_LOW\_PRESSURE** if the following logic evaluates to True:  
  
(  
 (**R\_LP\_MFLD\_LOW\_PRESSURE\_RESET** is equal to False)  
)  
  
Otherwise, set **R\_LP\_MFLD\_LOW\_PRESSURE** to False.  
  
**[FMUREQ-1864]:** FMU Software shall set **R\_LP\_MFLD\_LOW\_PRESSURE** to True if the following logic evaluates to True:  
  
(  
 (**R\_LP\_MFLD\_PRESSURE** is greater than or equal to *LP\_LOW\_PRESSURE\_RESET\_TH*) AND  
 (**R\_LP\_MFLD\_PRESSURE\_VALID** is equal to True)  
) is equal to True for greater than *LP\_LOW\_PRESSURE\_DELAY*  
  
Otherwise, set **R\_LP\_MFLD\_LOW\_PRESSURE** to False.

## AUTO\_REFUEL\_COMPLETE

**[FMUREQ-1883]:** FMU Software shall set **AUTO\_REFUEL\_COMPLETE** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**FUEL\_XFER\_MODE\_VALID** is equal to True) AND  
 (**DESIRED\_FUEL\_LEVEL** is greater than or equal to **FUEL\_ON\_BOARD**) AND  
 (**DESIRED\_FUEL\_LEVEL\_VALID** is equal to True) AND  
 (**FUEL\_ON\_BOARD\_VALID** is equal to True)  
)  
  
Otherwise, set **AUTO\_REFUEL\_COMPLETE** to False

# Control Logic

## Center Tank Pumps Enabled

**[FMUREQ-1807]:** FMU Software shall set **CTR\_TK\_PUMP\_1\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (  
 (**CTR\_TK\_MAN\_SWITCH** is equal to True) AND  
 (**CTR\_TK\_MAN\_SWITCH\_VALID** is equal to True) AND  
 (**CTR\_TK\_PUMP\_1\_SWITCH** is equal to True) AND  
 (**CTR\_TK\_PUMP\_1\_SWITCH\_VALID** is equal to True)  
 ) OR  
 (  
 (**CTR\_TK\_MAN\_SWITCH** is equal to False) AND  
 (**CTR\_TK\_MAN\_SWITCH\_VALID** is equal to True) AND  
 (**CTR\_TK\_PUMP\_1\_AUTO\_STOP** is equal to False)  
 )  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_1\_ENABLE** to False.

**[FMUREQ-1808]:** FMU Software shall set **CTR\_TK\_PUMP\_2\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (  
 (**CTR\_TK\_MAN\_SWITCH** is equal to True) AND  
 (**CTR\_TK\_MAN\_SWITCH\_VALID** is equal to True) AND  
 (**CTR\_TK\_PUMP\_2\_SWITCH** is equal to True) AND  
 (**CTR\_TK\_PUMP\_2\_SWITCH\_VALID** is equal to True)  
 ) OR  
 (  
 (**CTR\_TK\_MAN\_SWITCH** is equal to False) AND  
 (**CTR\_TK\_MAN\_SWITCH\_VALID** is equal to True) AND  
 (**CTR\_TK\_PUMP\_2\_AUTO\_STOP** is equal to False)  
 )  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_2\_ENABLE** to False.

## Wing Tank Pumps Enabled

**[FMUREQ-1823]:** FMU Software shall set **L\_TK\_PUMP\_1\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (**L\_TK\_PUMP\_1\_SWITCH** is equal to True) AND  
 (**L\_TK\_PUMP\_1\_SWITCH\_VALID** is equal to True)  
)  
  
Otherwise, set **L\_TK\_PUMP\_1\_ENABLE** to False.

**[FMUREQ-1824]:** FMU Software shall set **L\_TK\_PUMP\_2\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (**L\_TK\_PUMP\_2\_SWITCH** is equal to True) AND  
 (**L\_TK\_PUMP\_2\_SWITCH\_VALID** is equal to True)  
)  
  
Otherwise, set **L\_TK\_PUMP\_2\_ENABLE** to False.

**[FMUREQ-1825]:** FMU Software shall set **R\_TK\_PUMP\_1\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (**R\_TK\_PUMP\_1\_SWITCH** is equal to True) AND  
 (**R\_TK\_PUMP\_1\_SWITCH\_VALID** is equal to True)  
)  
  
Otherwise, set **R\_TK\_PUMP\_1\_ENABLE** to False.

**[FMUREQ-1826]:** FMU Software shall set **R\_TK\_PUMP\_2\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (**R\_TK\_PUMP\_2\_SWITCH** is equal to True) AND  
 (**R\_TK\_PUMP\_2\_SWITCH\_VALID** is equal to True)  
)  
  
Otherwise, set **R\_TK\_PUMP\_2\_ENABLE** to False.

## CTR\_TK\_PUMP\_x\_AUTO\_STOP

**[FMUREQ-1812]:** FMU Software shall set **CTR\_TK\_PUMP\_1\_AUTO\_STOP** to True if the following logic evaluates to True:  
  
(  
 (  
 (**FLIGHT\_PHASE** is equal to *FLIGHT\_PHASE\_TAKEOFF*) AND  
 (**FLIGHT\_PHASE\_VALID** is equal to True)  
 ) OR  
 (  
 (**CTR\_TK\_LOW\_LVL** is equal to True) AND  
 (**CTR\_TK\_LOW\_LVL\_VALID** is equal to True)  
 ) is equal to True for greater than *CTR\_TK\_LO\_LVL\_DELAY* OR  
 (  
 (**L\_TK\_LOW\_LVL** is equal to True) AND  
 (**L\_TK\_LOW\_LVL\_VALID** is equal to True)  
 )  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_1\_AUTO\_STOP** to False.

**[FMUREQ-1813]:** FMU Software shall set **CTR\_TK\_PUMP\_2\_AUTO\_STOP** to True if the following logic evaluates to True:  
  
(  
 (  
 (**FLIGHT\_PHASE** is equal to *FLIGHT\_PHASE\_TAKEOFF*) AND  
 (**FLIGHT\_PHASE\_VALID** is equal to True)  
 ) OR  
 (  
 (**CTR\_TK\_LOW\_LVL** is equal to True) AND  
 (**CTR\_TK\_LOW\_LVL\_VALID** is equal to True)  
 ) is equal to True for greater than *CTR\_TK\_LO\_LVL\_DELAY* OR  
 (  
 (**R\_TK\_LOW\_LVL** is equal to True) AND  
 (**R\_TK\_LOW\_LVL\_VALID** is equal to True)  
 )  
)  
  
Otherwise, set **CTR\_TK\_PUMP\_2\_AUTO\_STOP** to False.

## XFEED\_VALVE\_OPEN\_CMD

**[FMUREQ-1828]** FMU Software shall set **XFEED\_VALVE\_OPEN\_CMD** to True if the following logic evaluates to True:

(  
 (**XFEED\_VALVE\_SWITCH** is equal to True) AND  
 (**XFEED\_VALVE\_SWITCH\_VALID** is equal to True) AND  
 (**XFEED\_VALVE\_FULLY\_OPEN** is False) AND  
 (**XFEED\_VALVE\_FULLY\_OPEN\_VALID** is True)  
)  
  
Otherwise, set **XFEED\_VALVE\_OPEN\_CMD** to False.

## XFEED\_VALVE\_CLOSE\_CMD

**[FMUREQ-1829]** FMU Software shall set **XFEED\_VALVE\_CLOSE\_CMD** to True if the following logic evaluates to True:

(  
 (**XFEED\_VALVE\_SWITCH** is equal to False) AND  
 (**XFEED\_VALVE\_SWITCH\_VALID** is equal to True) AND  
 (**XFEED\_VALVE\_FULLY\_CLOSED** is False) AND  
 (**XFEED\_VALVE\_FULLY\_CLOSED\_VALID** is True)  
)  
  
Otherwise, set **XFEED\_VALVE\_CLOSE\_CMD** to False.

## FUEL\_BALANCE\_ENABLE

**[FMUREQ-1833]:** FMU Software shall set **FUEL\_BALANCE\_ENABLE** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_ENABLE\_RESET** is equal to False) AND  
 (**FUEL\_BALANCE\_PB** transitions from False to True) AND  
 (**FUEL\_BALANCE\_PB\_VALID** is equal to True) AND  
 (Previous value of **FUEL\_BALANCE\_ENABLE** is equal to False) AND  
 (**FUEL\_BALANCE\_INHIBIT** is equal to False)  
)  
  
Otherwise, maintain the previous value of **FUEL\_BALANCE\_ENABLE** if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_ENABLE\_RESET** is equal to False)  
)

**[FMUREQ-1834]:** FMU Software shall set **FUEL\_BALANCE\_ENABLE\_RESET** to True if the following logic evaluates to True:  
  
(  
 (  
 (**FUEL\_BALANCE\_PB** transitions from False to True) AND  
 (**FUEL\_BALANCE\_PB\_VALID** is equal to True) AND  
 (Previous value of **FUEL\_BALANCE\_ENABLE** is equal to True)  
 ) OR  
 (**FUEL\_BALANCE\_INHIBIT** is equal to True) OR  
 (**FUEL\_BALANCE\_COMPLETE** is equal to True for greater than *FUEL\_BALANCE\_COMPLETE\_DELAY*)  
)  
  
Otherwise, set **FUEL\_BALANCE\_ENABLE\_RESET** to False.

## FUEL\_BALANCE\_VALVE\_x\_OPEN\_CMD

**[FMUREQ-1839]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_L\_OPEN\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_COMPLETE** is equal to False) AND  
 (**FUEL\_BALANCE\_ENABLE** is equal to True) AND  
 (**L\_TK\_FUEL\_LEVEL** is greater than **R\_TK\_FUEL\_LEVEL**) AND  
 (**FUEL\_BALANCE\_VALVE\_R\_OPEN** is equal to False) AND  
 (**FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN** is equal to False) AND  
 (**FUEL\_BALANCE\_VALVE\_L\_FULLY\_OPEN\_VALID** is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_L\_OPEN\_CMD** to False.

**[FMUREQ-1841]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_R\_OPEN\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_COMPLETE** is equal to False) AND  
 (**FUEL\_BALANCE\_ENABLE** is equal to True) AND  
 (**R\_TK\_FUEL\_LEVEL** is greater than **L\_TK\_FUEL\_LEVEL**) AND  
 (**FUEL\_BALANCE\_VALVE\_L\_OPEN** is equal to False)  
 (**FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN** is equal to False) AND  
 (**FUEL\_BALANCE\_VALVE\_R\_FULLY\_OPEN\_VALID** is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_R\_OPEN\_CMD** to False.

## FUEL\_BALANCE\_VALVE\_x\_CLOSE\_CMD

**[FMUREQ-1848]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_L\_CLOSE\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_COMPLETE** is equal to True) AND  
 (**FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED** is equal to False) AND  
 (**FUEL\_BALANCE\_VALVE\_L\_FULLY\_CLOSED\_VALID** is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_L\_CLOSE\_CMD** to False.

**[FMUREQ-1849]:** FMU Software shall set **FUEL\_BALANCE\_VALVE\_R\_CLOSE\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_BALANCE\_COMPLETE** is equal to True) AND  
 (**FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED** is equal to False) AND  
 (**FUEL\_BALANCE\_VALVE\_R\_FULLY\_CLOSED\_VALID** is equal to True)  
)  
  
Otherwise, set **FUEL\_BALANCE\_VALVE\_R\_CLOSE\_CMD** to False.

## APU\_LP\_VALVE\_OPEN\_CMD

**[FMUREQ-1856]:** FMU Software shall set **APU\_LP\_VALVE\_OPEN\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**APU\_MASTER\_SW** is equal to True) AND  
 (**APU\_MASTER\_SW\_VALID** is equal to True) AND  
 (**APU\_FIRE** is equal to False) AND  
 (**APU\_FIRE\_VALID** is equal to True) AND  
 (**APU\_LP\_VALVE\_FULLY\_OPEN** is equal to False) AND  
 (**APU\_LP\_VALVE\_FULLY\_OPEN\_VALID** is equal to True)  
)  
  
Otherwise, set **APU\_LP\_VALVE\_OPEN\_CMD** to False.

## APU\_LP\_VALVE\_CLOSE\_CMD

**[FMUREQ-1857]:** FMU Software shall set **APU\_LP\_VALVE\_CLOSE\_CMD** to True if the following logic evaluates to True:  
  
(  
 (  
 (  
 (**APU\_MASTER\_SW** is equal to False) AND  
 (**APU\_MASTER\_SW\_VALID** is equal to True) AND  
 ) OR  
 (  
 (**APU\_FIRE** is equal to True) AND  
 (**APU\_FIRE\_VALID** is equal to True) AND  
 )  
 ) AND  
 (**APU\_LP\_VALVE\_FULLY\_CLOSED** is equal to False) AND  
 (**APU\_LP\_VALVE\_FULLY\_CLOSED\_VALID** is equal to True)  
)  
  
Otherwise, set **APU\_LP\_VALVE\_OPEN\_CMD** to False.

## APU\_ELEC\_PUMP\_ENABLE

**[FMUREQ-1858]:** FMU software shall set **APU\_ELEC\_PUMP\_ENABLE** to True if the following evaluates to True:  
  
(  
 (**APU\_MASTER\_SW** is equal to True) AND  
 (**APU\_MASTER\_SW\_VALID** is equal to True) AND  
 (**APU\_FIRE** is equal to False) AND  
 (**APU\_FIRE\_VALID** is equal to True) AND  
 (**L\_LP\_MFLD\_LOW\_PRESSURE** is equal to True)  
)  
  
Otherwise, set **APU\_ELEC\_PUMP\_ENABLE** to False.

## x\_REFUEL\_VALVE\_OPEN\_CMD

**[FMUREQ-1877]:** FMU Software shall set **L\_REFUEL\_VALVE\_OPEN\_CMD** if the following logic evaluates to True:  
  
(  
 (**L\_REFUEL\_VALVE\_CLOSE\_CMD** is False) AND  
 (**L\_REFUEL\_VALVE\_FULLY\_OPEN** is False) AND  
 (**L\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** is True) AND  
 (**L\_TK\_REFUEL\_SW** is True) AND  
 (**TK\_REFUEL\_SW\_VALID** is True) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True) AND  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_MAN) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**L\_TK\_HI\_LEVEL** is equal to False) AND  
 (**L\_TK\_HI\_LEVEL\_VALID** is equal to True) AND  
 (**AUTO\_REFUEL\_COMPLETE** is equal to False)  
 )  
 )  
)  
  
Otherwise, set **L\_REFUEL\_VALVE\_OPEN\_CMD** to False.

**[FMUREQ-1879]:** FMU Software shall set **R\_REFUEL\_VALVE\_OPEN\_CMD** if the following logic evaluates to True:  
  
(  
 (**R\_REFUEL\_VALVE\_CLOSE\_CMD** is False) AND  
 (**R\_REFUEL\_VALVE\_FULLY\_OPEN** is False) AND  
 (**R\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** is True) AND  
 (**R\_TK\_REFUEL\_SW** is True) AND  
 (**TK\_REFUEL\_SW\_VALID** is True) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True) AND  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_MAN) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**R\_TK\_HI\_LEVEL** is equal to False) AND  
 (**R\_TK\_HI\_LEVEL\_VALID** is equal to True) AND  
 (**AUTO\_REFUEL\_COMPLETE** is equal to False)  
 )  
 )  
)  
  
Otherwise, set **R\_REFUEL\_VALVE\_OPEN\_CMD** to False.

**[FMUREQ-1881]:** FMU Software shall set **CTR\_REFUEL\_VALVE\_OPEN\_CMD** if the following logic evaluates to True:  
  
(  
 (**CTR\_REFUEL\_VALVE\_CLOSE\_CMD** is False) AND  
 (**CTR\_REFUEL\_VALVE\_FULLY\_OPEN** is False) AND  
 (**CTR\_REFUEL\_VALVE\_FULLY\_OPEN\_VALID** is True) AND  
 (**CTR\_TK\_REFUEL\_SW** is True) AND  
 (**TK\_REFUEL\_SW\_VALID** is True) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True) AND  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_MAN) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**CTR\_TK\_HI\_LEVEL** is equal to False) AND  
 (**CTR\_TK\_HI\_LEVEL\_VALID** is equal to True) AND  
 (**AUTO\_REFUEL\_COMPLETE** is equal to False)  
 )  
 )  
)  
  
Otherwise, set **CTR\_REFUEL\_VALVE\_OPEN\_CMD** to False.

## x\_REFUEL\_VALVE\_CLOSE\_CMD

**[FMUREQ-1878]:** FMU Software shall set **L\_REFUEL\_VALVE\_CLOSE\_CMD** if the following logic evaluates to True:  
  
(  
 (**L\_REFUEL\_VALVE\_FULLY\_CLOSED** is False) AND  
 (**L\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** is True) AND  
 (  
 (  
 (**L\_TK\_REFUEL\_SW** is False) AND  
 (**TK\_REFUEL\_SW\_VALID** is True)  
 ) OR  
 (  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_OFF) OR  
 (**FUEL\_XFER\_MODE** is equal to XM\_DEFUEL)  
 ) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 ) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 (**L\_TK\_HI\_LEVEL** is equal to True) AND  
 (**L\_TK\_HI\_LEVEL\_VALID** is equal to True)  
 ) OR  
 (**AUTO\_REFUEL\_COMPLETE** is equal to True)  
 )  
)  
  
Otherwise, set **L\_REFUEL\_VALVE\_CLOSE\_CMD** to False.

**[FMUREQ-1880]:** FMU Software shall set **R\_REFUEL\_VALVE\_CLOSE\_CMD** if the following logic evaluates to True:  
  
(  
 (**R\_REFUEL\_VALVE\_FULLY\_CLOSED** is False) AND  
 (**R\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** is True) AND  
 (  
 (  
 (**R\_TK\_REFUEL\_SW** is False) AND  
 (**TK\_REFUEL\_SW\_VALID** is True)  
 ) OR  
 (  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_OFF) OR  
 (**FUEL\_XFER\_MODE** is equal to XM\_DEFUEL)  
 ) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 ) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 (**R\_TK\_HI\_LEVEL** is equal to True) AND  
 (**R\_TK\_HI\_LEVEL\_VALID** is equal to True)  
 ) OR  
 (**AUTO\_REFUEL\_COMPLETE** is equal to True)  
 )  
)  
  
Otherwise, set **R\_REFUEL\_VALVE\_CLOSE\_CMD** to False.

**[FMUREQ-1882]:** FMU Software shall set **CTR\_REFUEL\_VALVE\_CLOSE\_CMD** if the following logic evaluates to True:  
  
(  
 (**CTR\_REFUEL\_VALVE\_FULLY\_CLOSED** is False) AND  
 (**CTR\_REFUEL\_VALVE\_FULLY\_CLOSED\_VALID** is True) AND  
 (  
 (  
 (**CTR\_TK\_REFUEL\_SW** is False) AND  
 (**TK\_REFUEL\_SW\_VALID** is True)  
 ) OR  
 (  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_OFF) OR  
 (**FUEL\_XFER\_MODE** is equal to XM\_DEFUEL)  
 ) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 ) OR  
 (  
 (**FUEL\_XFER\_MODE** is equal to XM\_AUTO) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True)  
 (**CTR\_TK\_HI\_LEVEL** is equal to True) AND  
 (**CTR\_TK\_HI\_LEVEL\_VALID** is equal to True)  
 ) OR  
 (**AUTO\_REFUEL\_COMPLETE** is equal to True)  
 )  
)  
  
Otherwise, set **CTR\_REFUEL\_VALVE\_CLOSE\_CMD** to False.

## DEFUEL\_VALVE\_OPEN\_CMD

**[FMUREQ-1884]:** FMU Software shall set **DEFUEL\_VALVE\_OPEN\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_XFER\_MODE** is equal to XM\_DEFUEL) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True) AND  
 (**DEFUEL\_VALVE\_FULLY\_OPEN** is equal to False) AND  
 (**DEFUEL\_VALVE\_FULLY\_OPEN\_VALID** is equal to True)  
)  
  
Otherwise, set **DEFUEL\_VALVE\_OPEN\_CMD** to False.

## DEFUEL\_VALVE\_CLOSE\_CMD

**[FMUREQ-1885]:** FMU Software shall set **DEFUEL\_VALVE\_CLOSE\_CMD** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_XFER\_MODE** is not equal to XM\_DEFUEL) AND  
 (**FUEL\_XFER\_MODE\_VALID** is True) AND  
 (**DEFUEL\_VALVE\_FULLY\_CLOSED** is equal to False) AND  
 (**DEFUEL\_VALVE\_FULLY\_CLOSED\_VALID** is equal to True)  
)  
  
Otherwise, set **DEFUEL\_VALVE\_CLOSE\_CMD** to False.

## FUEL\_ON\_BOARD

**[FMUREQ-1800]:** FMU Software shall set **FUEL\_ON\_BOARD** to the value of (**L\_TK\_FUEL\_LEVEL** + **CTR\_TK\_FUEL\_LEVEL**+ **R\_TK\_FUEL\_LEVEL**) and set **FUEL\_ON\_BOARD\_VALID** to True if the following logic evaluates to True:  
  
(  
 (**L\_TK\_FUEL\_LEVEL\_VALID** is equal to True) AND  
 (**CTR\_TK\_FUEL\_LEVEL\_VALID** is equal to True) AND  
 (**R\_TK\_FUEL\_LEVEL\_VALID** is equal to True) AND  
)  
  
Otherwise, set **FUEL\_ON\_BOARD** to the value of -1 and set **FUEL\_ON\_BOARD\_VALID** to False.

# Cockpit Display Logic

## LOW\_FUEL\_DISPLAY

**[FMUREQ-1805]:** FMU Software shall set **LOW\_FUEL\_DISPLAY** to the value of

## FMU\_FAULT\_DISPLAY

**[FMUREQ-xxxx]:** FMU software shall set **FMU\_FAULT\_DISPLAY** to **FMU\_FAULT\_RPT** and set **SCREEN\_UPDATE** to True if the following logic evaluates to True:  
  
(  
 (**FMU\_FAULT\_RPT** is not equal to 0)  
)  
  
Otherwise, set **FMU\_FAULT\_DISPLAY** to 0 and set **SCREEN\_UPDATE** to False.

## FMU\_FAULT\_RPT

**[FMUREQ-xxxx]:** FMU software shall set **FMU\_FAULT\_RPT** to 0 if the following logic evaluates to True:  
  
(  
 (**FUEL\_LEVEL\_XDCR\_FAULT** is equal to False) AND  
 (**OVERSPEED\_FAULT** is equal to False) AND  
 (**ISM\_FAULT** is equal to False)  
)

**[FMUREQ-xxxx]:** FMU software shall set **FMU\_FAULT\_RPT** to 1 if the following logic evaluates to True:  
  
(  
 (**FUEL\_LEVEL\_XDCR\_FAULT** is equal to True)  
)

**[FMUREQ-xxxx]:** FMU software shall set **FMU\_FAULT\_RPT** to 2 if the following logic evaluates to True:  
(  
 (**OVERSPEED\_FAULT** is equal to True)  
)

**[FMUREQ-xxxx]:** FMU software shall set **FMU\_FAULT\_RPT** to 3 if the following logic evaluates to True:  
  
(  
 (**ISM\_FAULT** is equal to True)  
)

## FUEL\_IMBALANCE\_DISPLAY

**[FMUREQ-1850]:** FMU software shall set **FUEL\_IMBALANCE\_DISPLAY** to True if the following logic evaluates to True:  
  
(  
 (**FUEL\_WING\_TK\_LEVEL\_DIFF** is greater than *FUEL\_IMBALANCE\_DISPLAY\_TH*)  
) is equal to True for greater than *FUEL\_IMBALANCE\_DISPLAY\_DELAY*  
  
Otherwise, set **FUEL\_IMBALANCE\_DISPLAY** to False.