# **Part 1: Use the problem-solving process to propose a solution to the challenge given.**

## **Q 1) The Problem at hand**

The problem here requires me to create a system to monitor an aircraft landing gear, ensuring that all three gears are fully extended and aligned in the correct position when the gear down button is pressed before the landing. When the gear down button is pressed, the monitoring system should activate a green LED signal if all three landing gears correctly extend and align with the aircraft’s fuselage, i.e., align at 0 degrees. But if any of the three gears fails to extend correctly or misaligns with the aircraft’s fuselage, a red LED signal should be displayed. The system should also alert these signals for 10 seconds when the gear down switch is activated, informing the pilots whether the landing gear system is ready for landing or not. The system design should also use digital and analog sensors for processing. Digital sensors can be used to detect the landing gear positions, while analog sensors are used to check the alignment of gears. These analog signals are then converted to digital form for processing.

To design this monitoring system, we assume that the gear sensors provide 1 signal (HIGH) if all the gears are successfully extended and 0 signal (LOW) if at least one gear retracts. We also assume that the alignment sensor is analog, which represents the degree of alignment and is converted to digital, which provides 1 signal (HIGH) if the gear perfectly aligns, i.e., gear is 0 degrees and 0 signal (LOW) if the gear is > 0 degrees. The gear sensors system works when the gear down switch is activated, i.e., 1 signal (HIGH) and no LEDs should be displayed when the gear down switch is deactivated or switched off, i.e., 0 signal (LOW). The system should also light the signal for 10 seconds so that the pilot is informed whether the landing gears are successfully activated or not. Unfortunately, if the landing gear malfunctions, then the airplane lift is calculated and presented to the pilot.

## **Q 2) Describing inputs and outputs**

We can break down the main system into two main subsystems i.e., landing gear status and gear alignment with aircraft’s fuselage status. The inputs include the status of landing gears, the status of gear alignment, and the status of the gear-down switch. The inputs will be stored using a proper data structure like a dictionary, and the system will display the result in the form of an LED.

### **Inputs (Structure):**

1. **Gear Down Switch:**

Gear Down Switch is one of the inputs of type digital, denoted as “gearDownSwitchStatus”. The pilot activates this switch to lower the landing gear in preparation for landing. Here, the “gearDownSwitchStatus” stores the status of the gear-down switch. The value is either 1 (HIGH) if activated or 0 (LOW) if deactivated.

Structure: “gearDownSwitchStatus”: 1 or 0

Output: 1 (HIGH) if activated or 0 (LOW) if deactivated

1. **Landing Gear Sensors** (3 sensors: Gear 1, Gear 2, Gear 3):

Landing Gear Sensors is a digital input type with three sensors i.e., Gear 1, Gear 2 and Gear 3. This input stores the status of all three gear sensors in a nested dictionary, with each sensor representing the value 1 if it is fully extended and 0 if it is retracted.

Structure:

# Dictionary for the three landing gear sensors

“landingGearSensors”: {

“gear1”: 1 or 0,

“gear2”: 1 or 0,

“gear3”: 1 or 0

}

Output: 1 (HIGH) if all the gears extend successfully, 0 (LOW) if one of the gears retracts.

1. **Gear Alignment Sensor:**

This gear alignment sensor continuously checks if the landing gears have successfully aligned with the aircraft fuselage. This input is of analog type and needs to be converted into digital before being stored. It stores either 1 (HIGH) if all gears are aligned at 0° degrees (correct position) or 0 (LOW) if one of the gears is aligned other than 0° degrees (misaligned).

Structure: “alignmentStatus”: 0-90

Outputs: 1 (HIGH) if the angle is 0° degrees, 0 (LOW) if the angle is other than 0° degrees.

1. **In case the landing gear fails to land, a calculation of the lift of the aircraft is needed:**

If the landing gear fails to land, we need to calculate the lift of the airplane for the reliability of the fuselage. For this, we need to have inputs such as the velocity of the aircraft, air density affected by altitude, wing area of the aircraft, and coefficient of lift.

**Overall input:**

“gearDownSwitchStatus”: 1 or 0, # if 1 then activated else deactivated

“landingGearSensors”: { # Nested dictionary for the three landing gear sensors

“gear1”: 1 or 0, # Gear 1: 1 if fully extended, 0 if retracted

“gear2”: 1 or 0, # Gear 2: 1 if fully extended, 0 if retracted

“gear3”: 1 or 0, # Gear 3: 1 if fully extended, 0 if retracted

},

“alignmentStatus”: 0-90 # Alignment angle between 0° – 90° degrees

### **Outputs:**

Green LED or Red LED

The outputs are Green LED or Red LED. Green LED is displayed when all the gears are extended and aligned correctly, whereas Red LED is displayed if any one of the gears is not extended fully or when one of the gears is misaligned. In case the gear-down switch is deactivated, no action is to be taken.

## **Q 3) Hand-written calculations**

The hand-written calculations include:

### Boolean Logic Calculations

Creating Truth Table:

We create a table listing all possible combinations of sensor inputs and their corresponding LED outputs for the logic circuit of the landing gear system.

Table 1: Hand-written truth table calculation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gear Down Switch | Gear1 (Nose Gear) | Gear2 (Left Main Gear) | Gear3 (Right Main Gear) | Alignment | Green LED | Red LED |
| 0 | X | X | X | X | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |

If the gear-down switch is deactivated, i.e., 0 (LOW), then no action is performed, which will be denoted by “X” in the below truth table, and both the LEDs are off, i.e., 0 (LOW). The gear-down switch is turned off, so both the LEDs are off, i.e., 0 (LOW), irrespective of the other inputs, so other inputs are placed as “X”.

If all three landing gears are extended correctly and the alignment is correct, then the green LED turns on, i.e., 1 (HIGH).

If any of the gears do not extend correctly or the alignment of any of the gears is incorrect, then a red LED turns on, i.e., 1 (HIGH).

Deriving Boolean expressions:

We can derive the following Boolean expressions from the above truth table for the Green and Red LED.

***Green LED:*** The green LED will light up only when all three gears are extended and aligned correctly. Also, the gear-down switch must be on.

Green LED = GearDown **⋅** NoseGear **⋅** LeftMainGear **⋅** RightMainGear **⋅** Alignment

***Red LED:*** The red LED lights up if any gear fails to extend or the alignment is incorrect (not 0° degrees). Also, the gear-down switch must be on.

Red LED = GearDown **⋅**

### Lift Calculation

The problem requires calculating the airplane’s lift in case the landing fails due to a malfunction of any one of the gears. To calculate the lift of airplane, we use the following formula expressions:

*L = ½ ρV2SrefCL*

Where:

L = Lift force,

ρ = Air density,

V = Velocity of the aircraft,

Sref = Wing area,

CL = Coefficient of lift

For example, the parameters for aircraft F-117A, according to assumptions, is given below:

Weight = 52,500 pounds,

Velocity (V) = 400 knots (convert to m/s)

Sref = 1140 square foot (convert to square meter)

The steps for aircraft lift calculation are given below:

**a)** Convert Units in International System of Units (SI):

We need to convert the units of the input values, such as velocity and wing area, into standard units.

*Velocity (V):* We convert the units of velocity, which are given by knots, to meters per second. The assumed velocity is 400 knots.

1 knot = 0.51444 m/s

So, V = 400 X 0.51444 = 205.776 m/s

*Wing Area (Sref):* We convert the units of wing surface area, which are given by square foot, to square meters. The assumed wing area is 1140 square feet.

1 square foot = 0.092903 square meters

So, Sref = 1140 X 0.092903 = 105.91 square meters

**b)** We get air density from standard atmosphere tables in the assignment. The assumed altitude of the aircraft is 30,000 feet, so the air density (ρ), according to the standard atmosphere tables, is 0.000889 slug/ft3.

Conversion:

*Air Density(*ρ*):* We convert the units of air density given by slug per cubic foot to kg per cubic meter. The assumed air density is 0.000889

1 slug per cubic foot = 515.379 kg per cubic meter

So, ρ = 0.000889 \*515.379 = 0.458

**c)** We also get the coefficient of lift CL, which depends on the angle of attack (pitch angle). The assignment also mentioned that if the pitch angle is greater than 16° degrees, the plane will stall. From the provided assignment, the following table can be used as a guide for the coefficient of lift at different angles:

Table 2: The coefficient of lift with respective to the angle of attack

|  |  |
| --- | --- |
| **The angle of Attack (**°**)** | **Coefficient of Lift CL** |
| 0° | 0.2 |
| 6° | 0.6 |
| 8° | 0.8 |
| 10° | 1.0 |
| 12° | 1.1 |

Now, after all the required inputs are obtained, we perform the lift calculation with 5 different coefficients of lift or in 5 different pitch angles, which are given below:

**At 0° (CL = 0.2):**

L = 1/2 ​× (0.000889 \*515.379) × (205.776)2 × 105.91 × 0.2

L = 0.229 × 42350.4 × 105.91 × 0.2 = 205428.154 Newtons (Kg.m/s2)

**At 6° (CL = 0.6):**

L = 1/2 ​× (0.000889 \*515.379) × (205.776)2 × 105.91 × 0.6

L = 0.229 × 42350.4 × 105.91 × 0.6 = 616284.46 Newtons

**At 8° (CL = 0.8):**

L = 1/2 ​× (0.000889 \*515.379) × (205.776)2 × 105.91 × 0.8

L = 0.229 × 42350.4 × 105.91 × 0.8 = 821712.61 Newtons

**At 10° (CL = 1.0):**

L = 1/2 ​× (0.000889 \*515.379) × (205.776)2 × 105.91 × 1.0

L = 0.229 × 42350.4 × 105.91 × 1.0 = 1027140.77 Newtons

**At 12° (CL = 1.1):**

L = 1/2 ​× (0.000889 \*515.379) × (205.776)2 × 105.91 × 1.1

L = 0.229 × 42350.4 × 105.91 × 1.1 = 1129854.85 Newtons

In this way, we can calculate the lift of aircraft at five different angles of attack (pitch angles).

## **Q 4) Creating a list of all the tasks required to provide a suitable working solution.**

### **Designing logic of the circuit**

1. **Truth Table**

Table 3: Truth table of the system

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gear Down Switch | Gear1 (Nose Gear) | Gear2 (Left Main Gear) | Gear3 (Right Main Gear) | Alignment | Green LED | Red LED |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | X | X | X | X | 0 | 0 |

The gear-down switch is turned off i.e., 0 (LOW), so both the LEDs are off, i.e., 0 (LOW), irrespective of the other inputs, so other inputs are placed as “X”.

When the gear-down switch is on i.e., 1 (HIGH), the LED status depends on the status of the landing gears and their alignments, as shown below:

* The Green LED turns on only when all gears are extended, and the alignment is correct.
* The Red LED turns on if any gear is not extended or the alignment is incorrect (i.e., not 0° degrees)

1. **Boolean Expression from the Truth Table**

We can derive the Boolean expressions for each LED from the above truth table.

Green LED: Since the green LED turns on when all the gears are extended, and the alignment is correct, the Boolean expression can be represented as follows:

Green LED = GearDown **⋅** NoseGear **⋅** LeftMainGear **⋅** RightMainGear **⋅** Alignment

Where **(⋅)** is AND operator

* Red LED: Since the red LED turns on if any of the gears are retracted and misaligned, the Boolean expression can be represented as follows:

1. **Visualising the circuit based on the Boolean expression above**

The circuit can be visualised as follows:

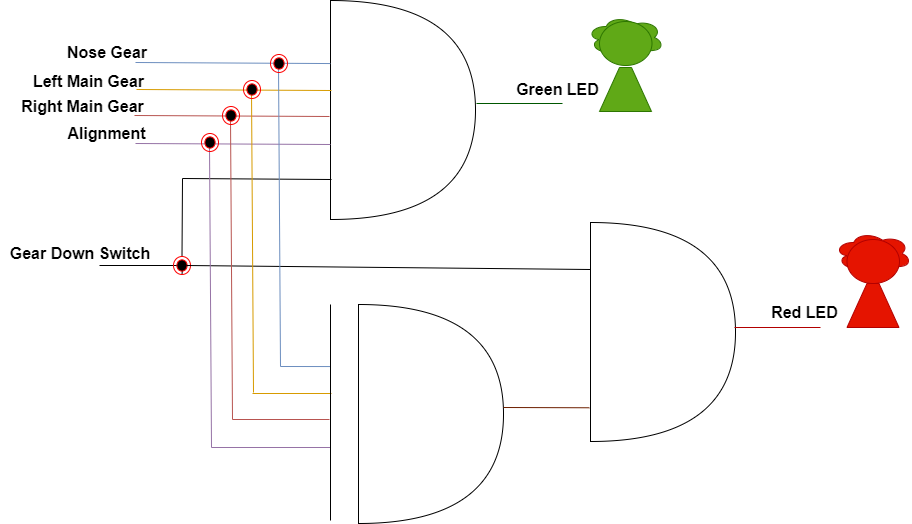


Figure 1: A logic circuit based on the Boolean expression

*Source:* Created using draw.io and own understanding

### **Creating a log to store aircraft landing conditions (its black box)**

We will design a system log (black box) that stores the aircraft landing information. This log will record all the essential data, such as user inputs, gear positions, alignment angle, aircraft lift value and LED statuses. We store all this data in a binary system for safety and protection. We will also be using a Python dictionary to store each value as a key-value pair and bin() Python function as this function returns the binary version of a specified integer, and the result will always start with prefix 0b ([W3Schools, n.d.](#_ENREF_1)).

Example of Dictionary Structure:

blackBoxLog = {  
 “gearDownSwitch”: bin(gear\_down\_status),

“landingGearPositionStatus”:{

“noseGear”: bin(nose\_gear\_status),

“leftMainGear”: bin(left\_main\_gear\_status),

“rightMainGear”: bin(right\_main\_gear\_status)

},

“alignmentAngle”: bin(alignment\_angle),

“lift”: bin(lift\_value),

“ledStatus”:{

“greenLed”: bin(green\_led\_status),

“redLed”: bin(red\_led\_status)

}  
}

Let us assume values of:

noseGear: 1, leftMainGear: 0, rightMainGear: 1, alignmentAngle: 15° degrees, lift: 243662, greenLed: 0, redLed: 1

Now, we store the value as follows:

blackBoxLog = {

"gearPositions": {

"noseGear": "0001", # Nose Gear is extended (binary 1)

"leftMainGear": "0000", # Left Main Gear is retracted (binary 0)

"rightMainGear": "0001" # Right Main Gear is extended (binary 1)

},

"alignmentAngle": "1111", # Alignment angle is 15 degrees (binary 1111)

"lift": "111011011110011110", # Lift is 243662 (binary representation)

"ledStatus": {

"greenLED": "0000", # Green LED is off

"redLED": "0001" # Red LED is on

}

}

### **Converting the Directional Sensor Data to Binary**

We now convert the analog data received from the directional sensor, which measures the alignment angle of landing gears, into its binary equivalent. As per the assignment, the angle of the sensor ranges from 0° to 90° and needs to be represented as binary numbers ranging from 0 to 9.

Firstly, we map the analog value (0° to 90°) to its corresponding discrete scale i.e., 0 to 9. For this, we divide the range from 0 ° to 90° degrees into 10 intervals, each having a single corresponding value of 0 to 9.

Table 4: Mapped value from directional sensor data

|  |  |
| --- | --- |
| Degree Range (°) | Mapped Value (0-9) |
| 0° - 9° | 0 |
| 10° - 19° | 1 |
| 20° - 29° | 2 |
| 30° - 39° | 3 |
| 40° - 49° | 4 |
| 50° - 59° | 5 |
| 60° - 69° | 6 |
| 70° - 79° | 7 |
| 80° - 89° | 8 |
| 90° | 9 |

Then, we convert the mapped value to their corresponding binary equivalent. The binary representations of values from 0 to 9 are given below:

Table 5: Corresponding binary equivalent of the mapped directional sensor data

|  |  |
| --- | --- |
| Mapped Value (0-9) | Binary Equivalent |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

### **Calculating the lift of the airplane and incorporating it into the black box log**

For this, we first calculate the lift of the airplane using the formula provided in the assignment appendix.

*L = ½ ρV2SrefCL*

Then, we discretize the lift value into intervals and convert it to binary for storing in a black box log. Let us consider the following intervals:

Table : Discretised lift range value and its corresponding binary equivalent

|  |  |  |
| --- | --- | --- |
| Lift Value Range (Newtons) | Discretize Value (0-9) | Binary Equivalent |
| 0 to 100,000 | 0 | 0000 |
| 100,001 to 200,000 | 1 | 0001 |
| 200,001 to 300,000 | 2 | 0010 |
| 300,001 to 400,000 | 3 | 0011 |
| 400,001 to 500,000 | 4 | 0100 |
| 500,001 to 600,000 | 5 | 0101 |
| 600,001 to 700,000 | 6 | 0110 |
| 700,001 to 800,000 | 7 | 0111 |
| 800,001 to 900,000 | 8 | 1000 |
| Greater than 900,000 | 9 | 1001 |

For example,

Let’s assume the value of air density = 0.364 kg/m3, velocity = 205.776 m/s (400 knots converted to m/s), Sref = 105.91 m2 (converted from square feet) and CL = 1.0 (for a 10° angle of attack).

L = 1/2 ​× 0.364 × (205.776)2 × 105.91 × 1.0

L = 0.182 × 42350.4 × 105.91 × 1.0 = 812209.4 Newton

Since the lift value is a large number i.e., 812209.4 Newtons, and the black box log only stores binary data, we need to discretise the lift value according to the table given above (*See:* Table 6: Discretised lift range value and its corresponding binary equivalent). Hence, the lift value 812209.4 Newtons will fall in 8th interval (800,001 to 900,000), so the discretised value is 8

The binary equivalent of 8 is 1000. So, we store 1000 as the lift value in the black box log.

## **Q 5) Flowchart**

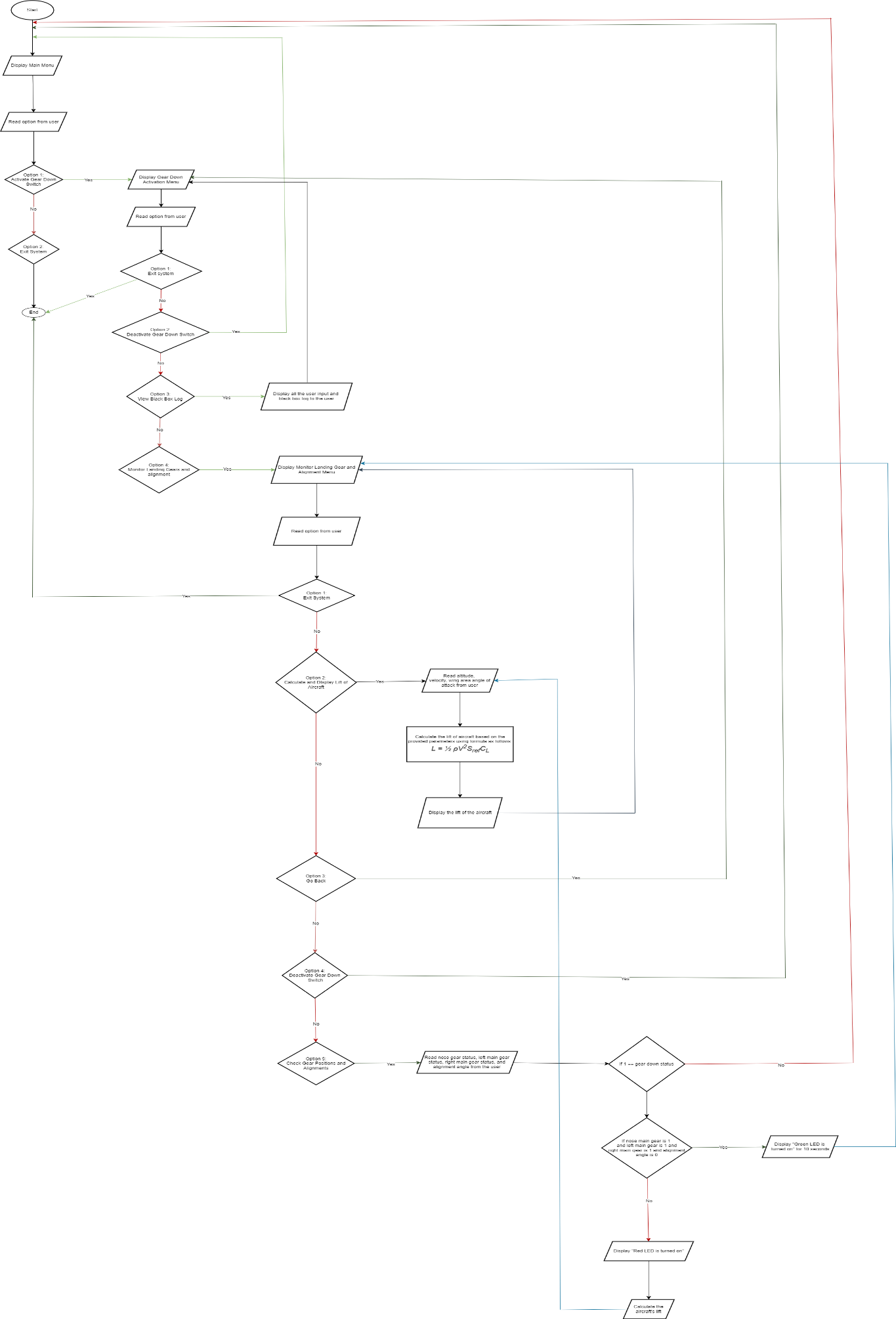


Figure : Flowchart design to create the monitoring system

The above figure (*See:* Figure 2: Flowchart design to create the monitoring system) demonstrates the flow diagram to create this monitoring system. The diagram uses different symbols, which is given in the table below (*See*: Table 7: Flow diagram symbols)

Table : Flow diagram symbols

|  |  |  |
| --- | --- | --- |
| Symbols | Symbol Name | Descriptions |
|  | Start/End symbol | It represents the start and the end of the process |
|  | Process symbol | It represents a process or calculation |
|  | Arrow symbol | It indicates the flow between each step |
|  | Decision symbol | It represents conditional operation i.e., true or false (or yes/no) |
|  | Input/Output symbol | It is used for input and output |

## **Q 6) Pseudocode**

Step 1: **Start**

* 1. Display “Main Menu”
  2. Read user input
  3. **If option 1: Activate Gear Down Switch, then**
     1. Display “Gear Down Switch Activated”
     2. Display “**Gear Down Activation Menu**”
     3. Read user input
     4. **If Option 1: Exit system, then**
        1. Jump to step 2
     5. **If Option 2: Deactivate Gear Down Switch, then**
        1. Jump to step 1.1.
     6. **If Option 3: View Black Box Log, then**
        1. Display all the user input and black box log to the user
        2. Jump to step 1.3.2.
     7. **If Option 4: Monitor Landing Gears and alignment, then**
        1. Display “**Monitor Landing Gear and Alignment Menu”**
        2. Read option from user
        3. **If Option 1: Exit system**
           1. Display “Existing the System”
           2. Jump to step 2
        4. **If Option 2: Calculate and Display Lift of Aircraft, then**
           1. Read altitude, velocity, wing area angle of attack from user
           2. Calculate the lift of aircraft based on the provided parameters using formula as follows: *L = ½ ρV2SrefCL*
           3. Display the lift of the aircraft
           4. Jump to step 1.3.7.1.
        5. **If Option 3: Go Back, then**
           1. Jump to step 1.3.2.
        6. **If Option 4: Deactivate Gear Down Switch, then**
           1. Jump to step 1.1.
        7. **If Option 5: Check Gear Positions and Alignments, then**
           1. Read nose gear status, left main gear status, right main gear status, and alignment angle from the user
           2. **If gear down status is 1, then**

**If nose main gear is 1 and left main gear is 1 and right main gear is 1 and alignment angle is 0, then**

Display **“Green LED is turned on”** for10 second

Jump to step 1.3.7.1

**Else**

Display “Red LED is turned on”

Calculate the aircraft’s lift

Jump to step 1.3.7.4.1

Display the result for 10 seconds

* + - * 1. **Else**

Jump to step 1.1.

* 1. **If option 2: Exit System**
     1. Display “**Existing the system**”
     2. Jump to step 2

Step 2: **End**

## Q 7) Test cases:

**Test1:**

Choose Option 1) Check Gear Positions and Alignment Monitor landing gear and alignment menu

Inputs:

Nose gear status = 1

Left main gear status = 1

Right main gear status = 1

Alignment angle = 0

Expected Output: Green LED turned on

**Test2:**

Choose Option 1) Check Gear Positions and Alignment Monitor landing gear and alignment menu

Inputs:

Nose gear status = 1

Left main gear status = 1

Right main gear status = 0

Alignment angle = 0

Expected Output as per handwritten calculation: Red LED is turned on

Result: Red LED is turned on

Calculating aircraft’s lift

Inputs:

Altitude in feet = 30000

Velocity in knots = 400

Wing area in square feet = 1140

Angle of attack = 0

Expected Output as per handwritten calculation = 205428.154 Newtons

Result: 205471.94 Newtons (due to decimal value round-offs)

**Test3:**

Choose Option 1) Check Gear Positions and Alignment Monitor landing gear and alignment menu

Inputs:

Nose gear status = 1

Left main gear status = 1

Right main gear status = 1

Alignment angle = 50

Expected Output as per handwritten calculation: Red LED is turned on

Result: Red LED is turned on

Calculating aircraft’s lift

Inputs:

Altitude in feet = 30000

Velocity in knots = 400

Wing area in square feet = 1140

Angle of attack = 6

Expected Output as per handwritten calculation = 616284.46 Newtons

Result: 616415.81 Newtons (due to decimal value round-offs)

# **Part 2: Coding**

The coding has been attached under the name: “Part 2 (coding)”

The coding for AI-GEN has been attached under the name: “Part 3 (AI Generated coding)”

# **Part 3: Comparison between AI-GEN code and my code**

I used an AI-gen called ChatGPT to generate the code for part 3. ChatGPT provided me with code with a simpler structure but lacks in many aspects. The main difference between the code I developed and the one ChatGPT generated is complexity and efficiency. ChatGPT focuses mostly on simple tasks like checking the landing gear status, alignment and calculating the lift, whereas my code is more complex as it includes features like global state management and black box logging, which fulfils the assignment requirements and makes my code more suitable for the real-world systems. The code generated by ChatGPT makes the user to convert the units themselves and then enter them, whereas my code asks the user to input as per the requirement and converts the data itself. My code also handles and logs the error properly and has proper documentation within the code, as I have used docstring to document the usage of each function.

Looking at the speed of development, the code generated by ChatGPT is faster to produce because it only focuses on immediate outputs and does not focus on error handling and state management. Even though my code takes a long time to develop, it has fulfilled all the requirements that the assignments have asked for, such as converting analog signals to digital signals, storing the system’s state, handling errors, unit conversion, and so on. My code development also needs more testing to ensure the system functions appropriately.

In terms of efficiency, the ChatGPT-generated code is more efficient than my code because I have used some time delays, state management, logging and complex programming. However, the design of my code can still handle a more robust and realistic system. So, my code is slower than the code generated by ChatGPT, but it is more reliable when the sensor data are constantly changing and need to be processed and stored correctly. Overall, my code is more aligned with the problem description provided in the assignment, which includes sensor conversion, unit conversion, black box log storage, error handling and state management, making it more suitable. The ChatGPT-generated code sure has simpler use cases, but it fails to fulfill a number of important requirements.

# **REFERENCES**

OPENAI. 2024. *ChatGPT (Sept 20 version)* [Online]. Available: https://chatgpt.com/share/66ed1711-5768-800a-a9cc-e367893ca704 [Accessed 19 September 2024].

W3SCHOOLS. n.d. *Python bin() Function* [Online]. Available: <https://www.w3schools.com/python/ref_func_bin.asp> [Accessed 18 September 2024].