Table of Contents:

Section 1 : Introduction and Objectives

In this project assignment, we are provided with a drone to perform different tasks in a post-zombie apocalyptic world. Our main objective of the drone is to defend the sanctuary of Lab E4-03-07, while taking measurements of the outside environment and sending the data back to the scientists residing in the sanctuary. The drone is weaponized and is capable of destroying zombies and other objectives with its laser gun.

Section 2 : Flowcharts of the System

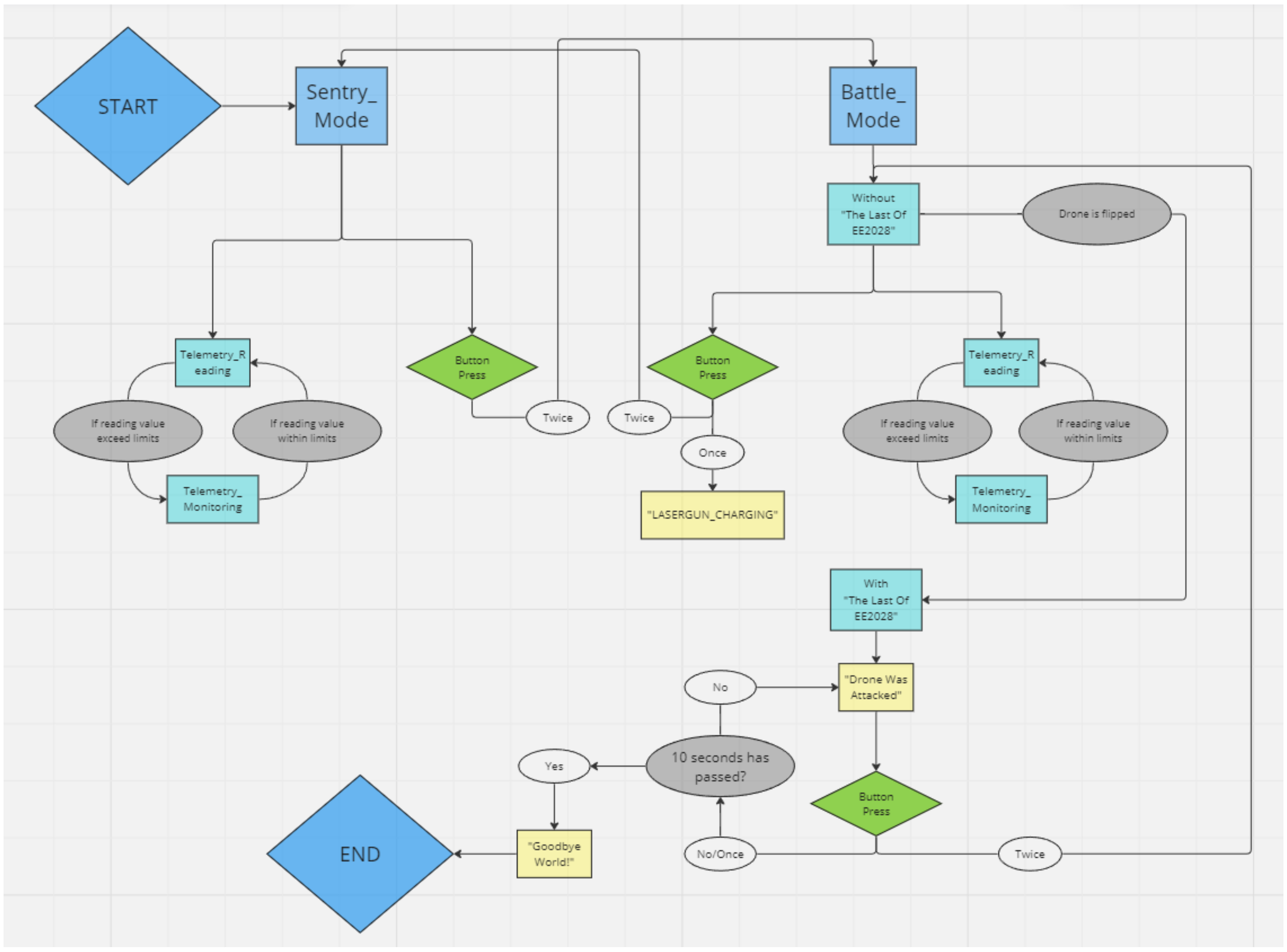


Figure 1: Flowchart of Original Requirements

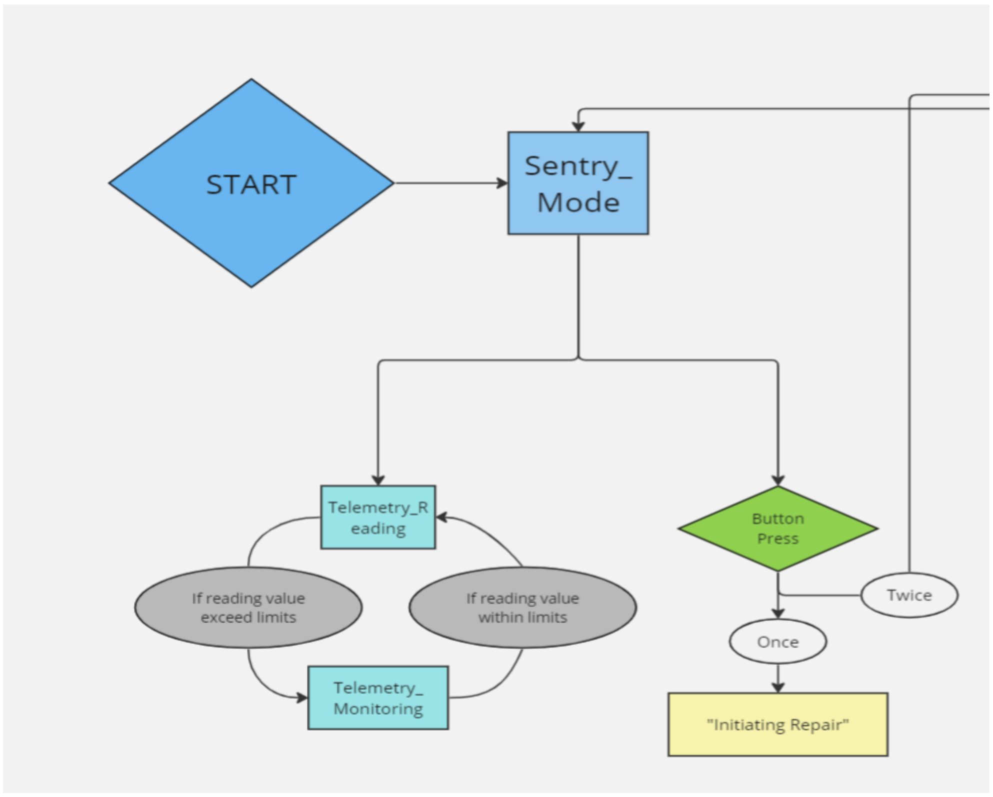


Figure 2: Drone Structural Integrity Repair Logic (Health) (Standby State)

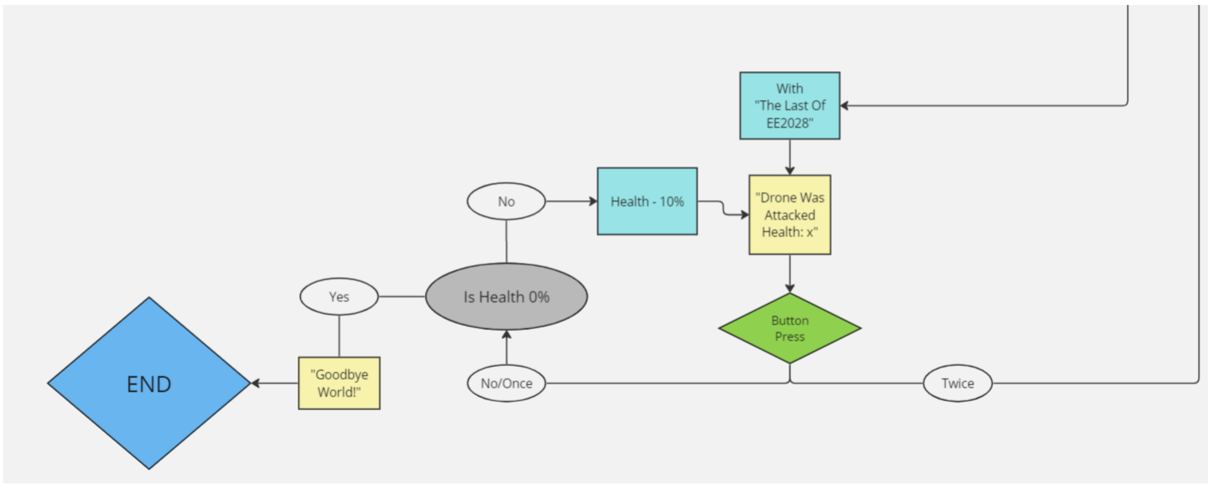


Figure 3: Drone Structural Integrity Depletion Logic (Health) (“The Last of EE2028” State)

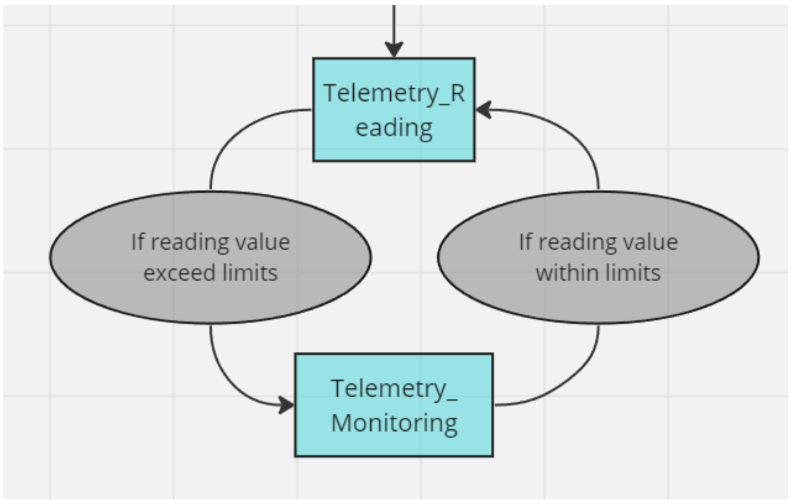


Figure 4: Telemetry_Reading to Telemetry Monitoring logic

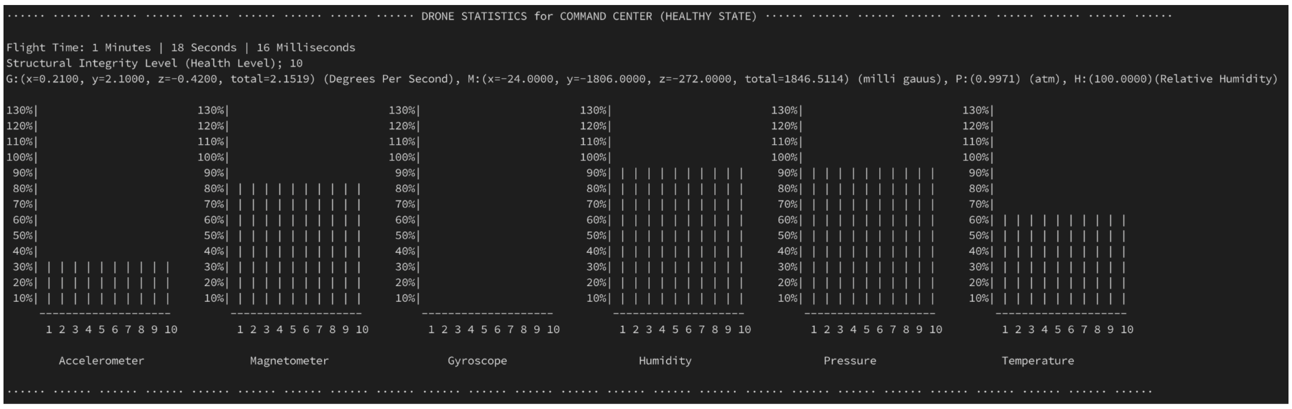
Section 3 : Detailed Explanation of Implementation Steps

We programmed an advanced statistical dashboard with both a current system overview as well as detailed breakdown and visual analysis of the sensor readings over time with graphical visualisations that provides valuable information to the command center personnel as a major enhancement feature encompassing. Furthermore, we have implemented a structural integrity system of the drone with automatic damage during attack and dynamic recuperating systems implemented as another enhancement feature.

3.1 Standby/Battle Mode (No Anomalies Detected By Sensor)

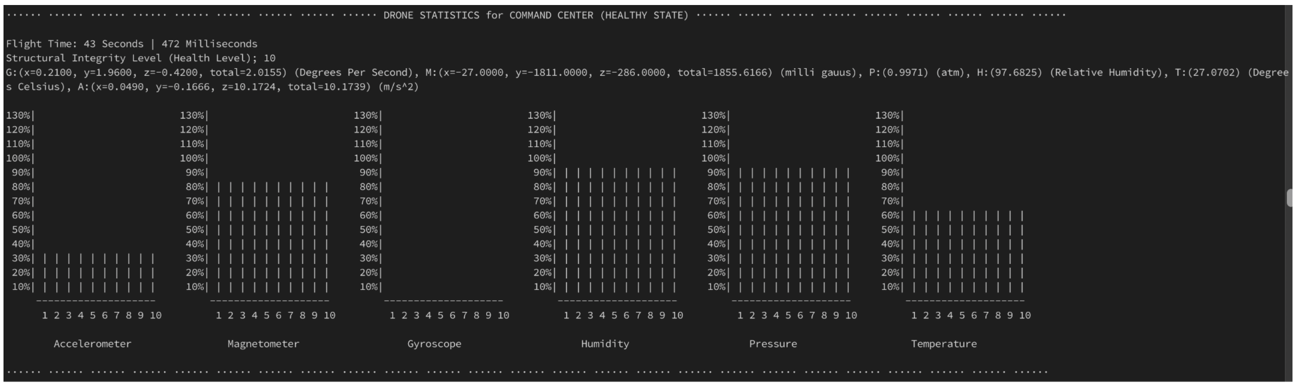
Scenario 1: Drone remains stationary and the conditions in its environment remain constant. So, all the sensors show a constant, unchanging pattern on the plotted graphs in the statistical dashboard over time.

- Standby Mode => Only four sensors gets printed in the console as string
  (Gyroscope, Magnetometer, Pressure, Humidity)



```
······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 1 Minutes | 18 Seconds | 16 Milliseconds
Structural Integrity Level (Health Level); 10
G:(x=0.2100, y=2.1000, z=-0.4200, total=2.1519) (Degrees Per Second), M:(x=-24.0000, y=-1806.0000, z=-272.0000, total=1846.5114) (milli gauus), P:(0.9971) (atm), H:(100.0000)(Relative Humidity)
```

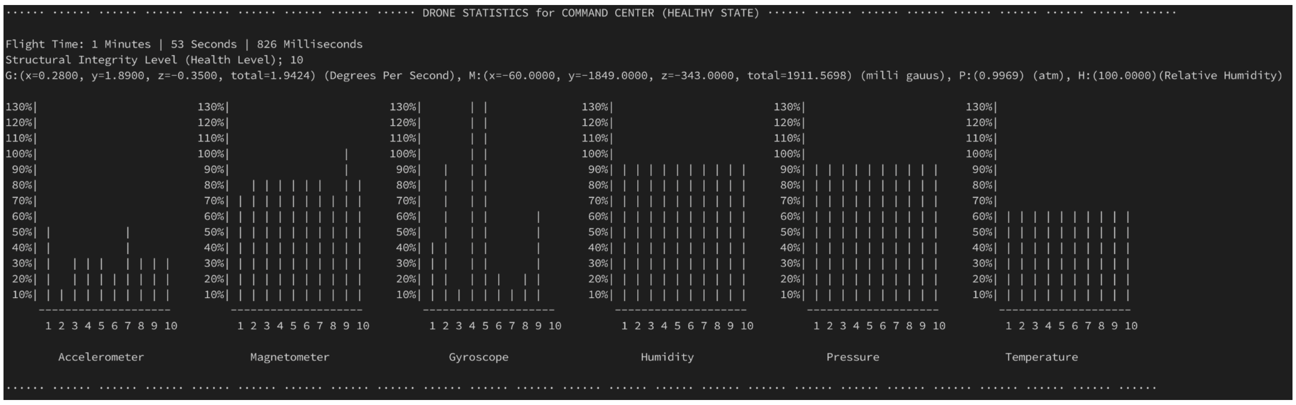
- Battle Mode => All six sensors get communicated to the command center
  (Gyroscope, Magnetometer, Pressure, Humidity, Temperature, Accelerometer)



```
······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 43 Seconds | 472 Milliseconds
Structural Integrity Level (Health Level); 10
G:(x=0.2100, y=1.9600, z=-0.4200, total=2.0155) (Degrees Per Second), M:(x=-27.0000, y=-1811.0000, z=-286.0000, total=1855.6166) (milli gauus), P:(0.9971) (atm), H:(97.6825) (Relative Humidity), T:(27.0702) (Degree
s Celsius), A:(x=0.0490, y=-0.1666, z=10.1724, total=10.1739) (m/s^2)
```

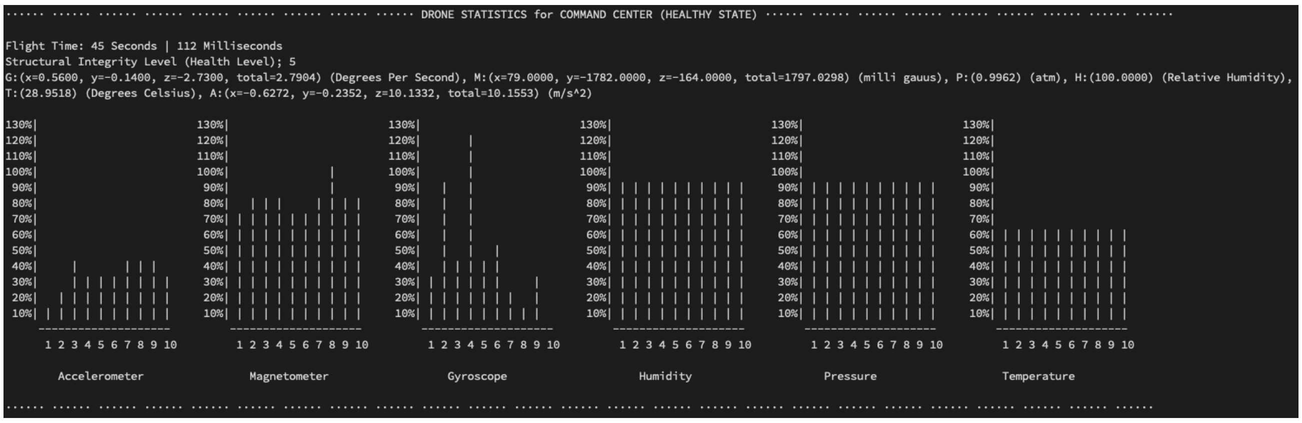
Scenario 2: Drone is in motion or the conditions of its environment are changing over time, some or all the sensors are changing over time as plotted on the graphs in the statistical dashboard.

- Standby Mode => only 4 get communicated to the command center
  (Gyroscope, Magnetometer, Pressure, Humidity)



```
······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 1 Minutes | 53 Seconds | 826 Milliseconds
Structural Integrity Level (Health Level); 10
G:(x=0.2800, y=1.8900, z=-0.3500, total=1.9424) (Degrees Per Second), M:(x=-60.0000, y=-1849.0000, z=-343.0000, total=1911.5698) (milli gauus), P:(0.9969) (atm), H:(100.0000)(Relative Humidity)
```

- Battle Mode => all 6 sensors get communicated to the command center (Gyroscope, Magnetometer, Pressure, Humidity, Temperature, Accelerometer)



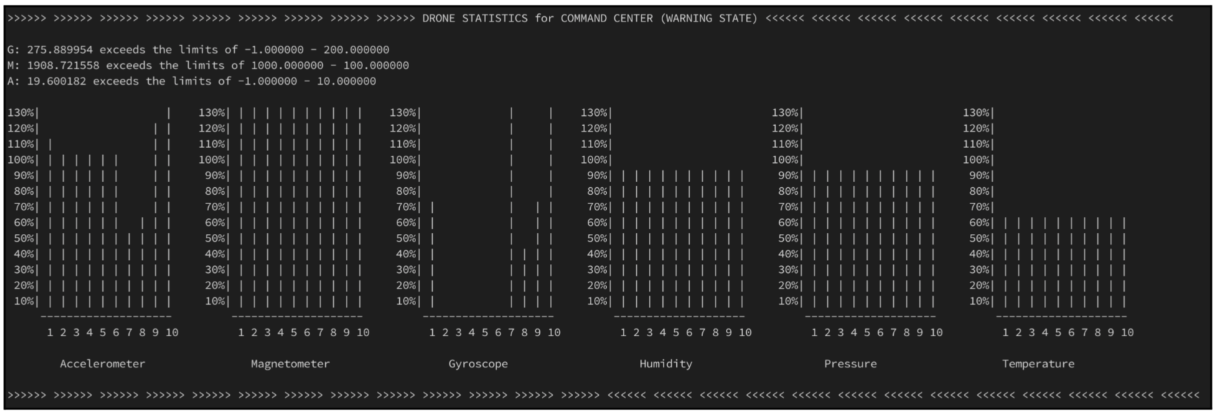
```
······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 45 Seconds | 112 Milliseconds
Structural Integrity Level (Health Level); 5
G:(x=0.5600, y=-0.1400, z=-2.7300, total=2.7904) (Degrees Per Second), M:(x=79.0000, y=-1782.0000, z=-164.0000, total=1797.0298) (milli gauus), P:(0.9962) (atm), H:(100.0000) (Relative Humidity),
T:(28.9518) (Degrees Celsius), A:(x=-0.6272, y=-0.2352, z=10.1332, total=10.1553) (m/s^2)

130%|                         130%|                         130%|                         130%|                         130%|                         130%|
120%|                         120%|                         120%|       |                 120%|                         120%|                         120%|
110%|                         110%|                         110%|       |                 110%|                         110%|                         110%|
100%|                         100%|               |         100%|       |                 100%|                         100%|                         100%|
 90%|                          90%|               |          90%|   |   |                  90%| | | | | | | | | |      90%| | | | | | | | | |      90%|
 80%|                          80%|   | | |     | | | |      80%|   |   |                  80%| | | | | | | | | |      80%| | | | | | | | | |      80%|
 70%|                          70%| | | | | | | | | |      70%|   |   |                  70%| | | | | | | | | |      70%| | | | | | | | | |      70%|
 60%|                          60%| | | | | | | | | |      60%|   |   |                  60%| | | | | | | | | |      60%| | | | | | | | | |      60%| | | | | | | | | |
 50%|                          50%| | | | | | | | | |      50%|   |   |   |              50%| | | | | | | | | |      50%| | | | | | | | | |      50%| | | | | | | | | |
 40%|     |       | | |       40%| | | | | | | | | |      40%|   | | | | |              40%| | | | | | | | | |      40%| | | | | | | | | |      40%| | | | | | | | | |
 30%|     | | | | | | | |      30%| | | | | | | | | |      30%| | | | | | |     |        30%| | | | | | | | | |      30%| | | | | | | | | |      30%| | | | | | | | | |
 20%|   | | | | | | | | |      20%| | | | | | | | | |      20%| | | | | | | |   |        20%| | | | | | | | | |      20%| | | | | | | | | |      20%| | | | | | | | | |
 10%| | | | | | | | | | |      10%| | | | | | | | | |      10%| | | | | | | | | |        10%| | | | | | | | | |      10%| | | | | | | | | |      10%| | | | | | | | | |
     --------------------          --------------------          --------------------          --------------------          --------------------          --------------------

      1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10

        Accelerometer                 Magnetometer                   Gyroscope                     Humidity                     Pressure                    Temperature

······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······
```

Explanation on Sensor Reading Implementation: We began by initialising all the six sensors that we were planning to utilise in our drone (Gyroscope, Magnetometer, Pressure, Humidity, Temperature, Accelerometer). Then, using **BSP_*sensorName*_*readData*()** , we are able to extract the readings using our own function **read*SensorName*()**. For the humidity, magnetometer and temperature sensors, we used the raw values in other parts of our code. For the pressure sensor, we converted raw values to a percentage of atmospheric pressure at sea level. For the accelerometer sensor, we have converted the unit to m/s^2. For the gyroscope sensor, we converted the unit to dps (degrees per second). Then, for all the sensors that provide data in 3 dimensions (x, y, z), which are Accelerometer, Magnetometer and Gyroscope sensors, we processed the data from all 3 axes to calculate the vector magnitude for each.

Explanation on Flight Time Calculation: We initialise a variable **milliseconds** as an integer that takes the value of **HAL_GetTick()** every loop of the main code. Using this variable, we are able to calculate how much time has passed since the start of the code (this tells us the total duration that the drone has been in flight). We perform a series of calculations by using the modulo(%) operator to return the days, hours, minutes, seconds and milliseconds.

Explanation on Graphs Plotting Implementation: We begin by initialising a two-dimensional array with dimensions [10][6] to store 10 sets of readings from all six sensors. When new data is being read from the sensors, we get rid of the oldest reading and add the new set of data into the array. We then read the array when printing to plot the graph. Since we are plotting the graph using UART, we have to print the graphs in strings, row by row for a visual representation of the data. Here, we decided to plot the graphs for all six sensors in every mode to enable the pilot to always have full awareness of the drone's physical environment. For the y-axis of the graph, we converted the readings to a proportion of the maximum threshold and used the percentage to plot. The x-axis is a simple counter.

3.2 Telemetric Sensor Anomaly Detection System



```
>>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> DRONE STATISTICS for COMMAND CENTER (WARNING STATE) <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<<

G: 275.889954 exceeds the limits of -1.000000 - 200.000000
M: 1908.721558 exceeds the limits of 1000.000000 - 100.000000
A: 19.600182 exceeds the limits of -1.000000 - 10.000000

130%|                   |     130%| | | | | | | | | |     130%|             |     |     130%|                         130%|                         130%|
120%|                 | |     120%| | | | | | | | | |     120%|             |     |     120%|                         120%|                         120%|
110%| |               | |     110%| | | | | | | | | |     110%|             |     |     110%|                         110%|                         110%|
100%| | | | | | |     | |     100%| | | | | | | | | |     100%|             |     |     100%|                         100%|                         100%|
 90%| | | | | | |     | |      90%| | | | | | | | | |      90%|             |     |      90%| | | | | | | | | |      90%| | | | | | | | | |      90%|
 80%| | | | | | |     | |      80%| | | | | | | | | |      80%|             |     |      80%| | | | | | | | | |      80%| | | | | | | | | |      80%|
 70%| | | | | | |     | |      70%| | | | | | | | | |      70%| |           |   | |      70%| | | | | | | | | |      70%| | | | | | | | | |      70%|
 60%| | | | | | |   | | |      60%| | | | | | | | | |      60%| |           |   | |      60%| | | | | | | | | |      60%| | | | | | | | | |      60%| | | | | | | | | |
 50%| | | | | | | | | | |      50%| | | | | | | | | |      50%| |           |   | |      50%| | | | | | | | | |      50%| | | | | | | | | |      50%| | | | | | | | | |
 40%| | | | | | | | | | |      40%| | | | | | | | | |      40%| |           | | | |      40%| | | | | | | | | |      40%| | | | | | | | | |      40%| | | | | | | | | |
 30%| | | | | | | | | | |      30%| | | | | | | | | |      30%| |           | | | |      30%| | | | | | | | | |      30%| | | | | | | | | |      30%| | | | | | | | | |
 20%| | | | | | | | | | |      20%| | | | | | | | | |      20%| |           | | | |      20%| | | | | | | | | |      20%| | | | | | | | | |      20%| | | | | | | | | |
 10%| | | | | | | | | | |      10%| | | | | | | | | |      10%| |           | | | |      10%| | | | | | | | | |      10%| | | | | | | | | |      10%| | | | | | | | | |
     --------------------          --------------------          --------------------          --------------------          --------------------          --------------------

      1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10          1 2 3 4 5 6 7 8 9 10

        Accelerometer                 Magnetometer                   Gyroscope                     Humidity                     Pressure                    Temperature

>>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> >>>>>> <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<< <<<<<<
```

Figure 6: Multiple Sensors Exceeding Threshold Values (Warning State)

Implementation of Sensor Anomaly Detection System of our drone consists of continuously monitoring various sensors critical for the drone's safe and effective operation. Sensors include temperature, pressure, humidity, accelerometer, gyroscope and magnetometer. To ensure optimal performance, and avoid dangerous situations, specific threshold values have been set for each sensor.

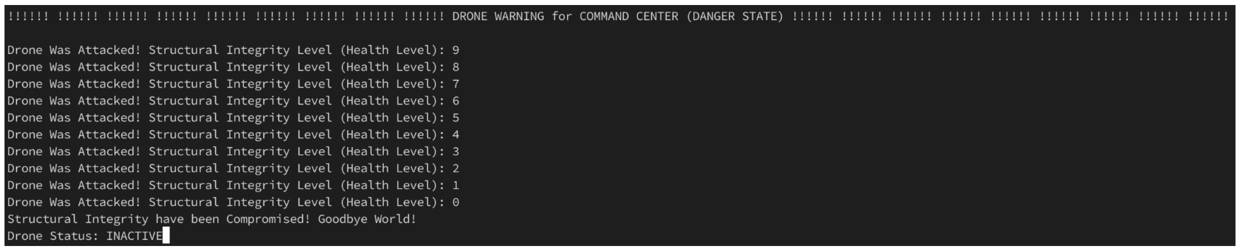
The drone's behaviour is determined based on its operational mode, which can be either Standby Mode or Battle Mode. Standby Mode utilises four sensors, while Battle Mode incorporates six sensors. The code continuously checks if the current sensor readings are within the specified thresholds, simulating normal drone operation.

If any sensor reading falls outside its designated threshold, it raises a flag for potential anomalous behaviour. The code assesses the gyroscope, magnetometer, accelerometer, pressure, humidity, and temperature readings for anomalies. This is essential for maintaining the drone's safe and reliable performance. For temperature, pressure and humidity we utilise their direct values (scaled to appropriate units) to compare with thresholds and for gyroscope, magnetometer and accelerometer we calculate their vector magnitudes (scaled to appropriate units) to compare with their thresholds.

In cases of detected anomalies, the code initiates a warning message to the command centre. The message not only indicates the sensor values that exhibited anomalies but also provides graphical representations of how the values of these sensors have changed over the past 10 seconds. These detailed insights enable the command centre to make informed decisions and take necessary actions to address any emerging issues.

3.3 Drone (Damage and Recuperation) of its Structural Integrity Health

Scenario 1: Drone is attacked and not rescued on time

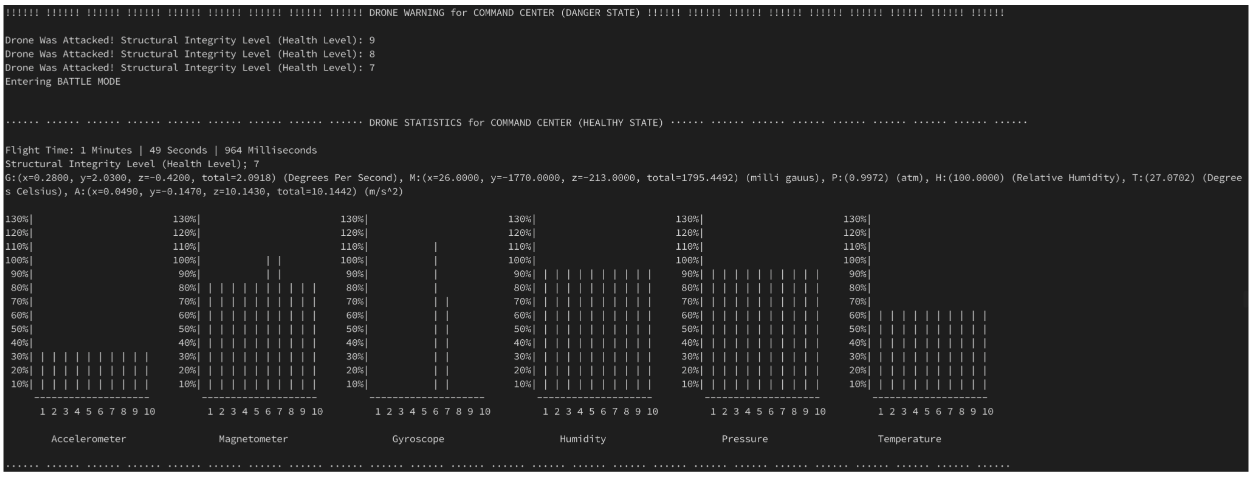


```
!!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! DRONE WARNING for COMMAND CENTER (DANGER STATE) !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!!

Drone Was Attacked! Structural Integrity Level (Health Level): 9
Drone Was Attacked! Structural Integrity Level (Health Level): 8
Drone Was Attacked! Structural Integrity Level (Health Level): 7
Drone Was Attacked! Structural Integrity Level (Health Level): 6
Drone Was Attacked! Structural Integrity Level (Health Level): 5
Drone Was Attacked! Structural Integrity Level (Health Level): 4
Drone Was Attacked! Structural Integrity Level (Health Level): 3
Drone Was Attacked! Structural Integrity Level (Health Level): 2
Drone Was Attacked! Structural Integrity Level (Health Level): 1
Drone Was Attacked! Structural Integrity Level (Health Level): 0
Structural Integrity have been Compromised! Goodbye World!
Drone Status: INACTIVE
```

Figure 7: Drone being attacked without being rescued, resulting in its structural integrity decreasing every second until it is destroyed. Once destroyed, all communication with the drone is terminated.

Drone Under Attack Detection: To detect potential attacks on a drone, we rely on the data from the accelerometer sensor, specifically focusing on readings along the z-axis. When the accelerometer reading along the z-axis drops below -8, we interpret this as an indication that the drone has turned upside down, which then triggers an alarm to notify that the drone is under attack. The rationale behind choosing an accelerometer reading of less than -8 as the threshold for identifying an inverted drone is due to the physics of gravity. In a standard, upright position, the accelerometer should record a force equivalent to 9.81 m/s^2 on the z-axis due to gravity. Conversely, when the drone is turned upside down, the z-axis measurement should register a force equivalent to -9.81 m/s^2 due to gravity. Consequently, an accelerometer reading falling below -8 provides a reliable threshold for recognizing an abnormal orientation that could signify a drone being upside down, thereby indicating an attack on the drone.

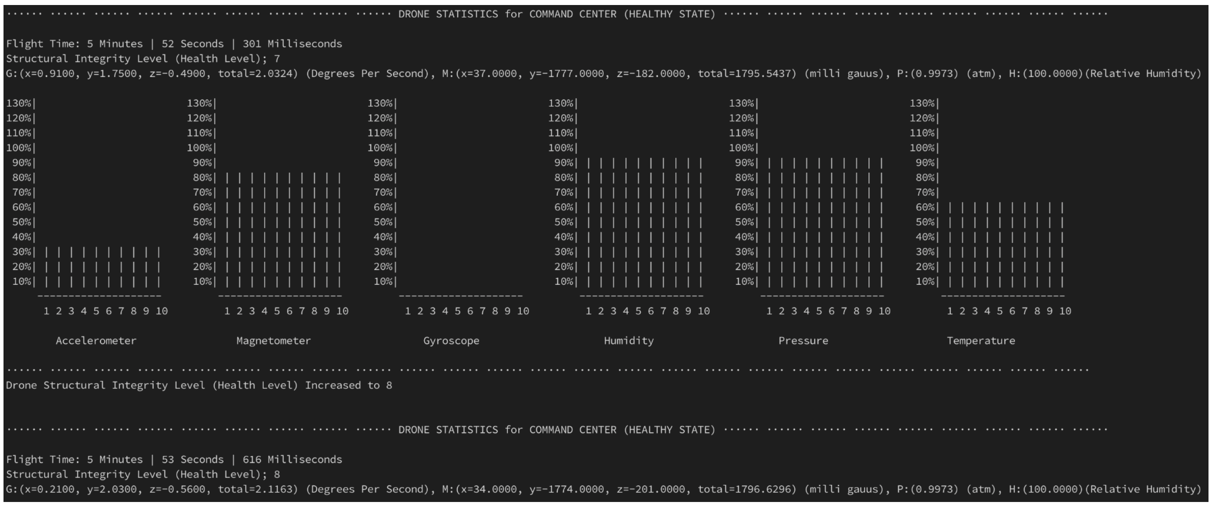
Scenario 2: Drone is attacked and is successfully rescued



```
!!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! DRONE WARNING for COMMAND CENTER (DANGER STATE) !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!! !!!!!!

Drone Was Attacked! Structural Integrity Level (Health Level): 9
Drone Was Attacked! Structural Integrity Level (Health Level): 8
Drone Was Attacked! Structural Integrity Level (Health Level): 7
Entering BATTLE MODE


······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 1 Minutes | 49 Seconds | 964 Milliseconds
Structural Integrity Level (Health Level); 7
G:(x=0.2800, y=2.0300, z=-0.4200, total=2.0918) (Degrees Per Second), M:(x=26.0000, y=-1770.0000, z=-213.0000, total=1795.4492) (milli gauus), P:(0.9972) (atm), H:(100.0000) (Relative Humidity), T:(27.0702) (Degree
s Celsius), A:(x=0.0490, y=-0.1470, z=10.1430, total=10.1442) (m/s^2)

130%|                        130%|                        130%|                        130%|                        130%|                        130%|
120%|                        120%|                        120%|                        120%|                        120%|                        120%|
110%|                        110%|                        110%|           |            110%|                        110%|                        110%|
100%|                        100%|           | |          100%|           |            100%|                        100%|                        100%|
 90%|                         90%|           | |           90%|           |             90%| | | | | | | | | | |     90%| | | | | | | | | | |     90%|
 80%|                         80%| | | | | | | | | | |     80%|           |             80%| | | | | | | | | | |     80%| | | | | | | | | | |     80%|
 70%|                         70%| | | | | | | | | | |     70%|           | |           70%| | | | | | | | | | |     70%| | | | | | | | | | |     70%|
 60%|                         60%| | | | | | | | | | |     60%|           | |           60%| | | | | | | | | | |     60%| | | | | | | | | | |     60%| | | | | | | | | | |
 50%|                         50%| | | | | | | | | | |     50%|           | |           50%| | | | | | | | | | |     50%| | | | | | | | | | |     50%| | | | | | | | | | |
 40%|                         40%| | | | | | | | | | |     40%|           | |           40%| | | | | | | | | | |     40%| | | | | | | | | | |     40%| | | | | | | | | | |
 30%| | | | | | | | | | |     30%| | | | | | | | | | |     30%|           | |           30%| | | | | | | | | | |     30%| | | | | | | | | | |     30%| | | | | | | | | | |
 20%| | | | | | | | | | |     20%| | | | | | | | | | |     20%|           | |           20%| | | | | | | | | | |     20%| | | | | | | | | | |     20%| | | | | | | | | | |
 10%| | | | | | | | | | |     10%| | | | | | | | | | |     10%|           | |           10%| | | | | | | | | | |     10%| | | | | | | | | | |     10%| | | | | | | | | | |
     --------------------         --------------------         --------------------         --------------------         --------------------         --------------------
      1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10

        Accelerometer                Magnetometer                  Gyroscope                    Humidity                   Pressure                   Temperature

······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······
```

Figure 8: Shows the structural integrity of the drone decreasing every second, but the drone is rescued with two button presses and resumes operation in battle mode.

We've designed a Structural Integrity Health System for our drone to ensure its robustness. Initially, the drone operates at full health, rated at 10 out of 10. When engaged in battle mode and under attack, the drone's structural integrity gradually depletes, losing 1/10 of its health per second. If this attack continues until the drone's structural integrity reaches zero, it indicates that the drone is critically damaged and beyond repair. In such a scenario, the drone will terminate communication with the command center and initiate a self-destruct sequence.



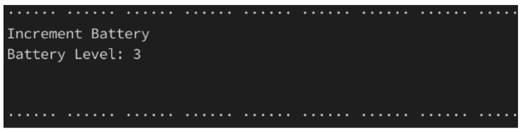
```
······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 5 Minutes | 52 Seconds | 301 Milliseconds
Structural Integrity Level (Health Level); 7
G:(x=0.9100, y=1.7500, z=-0.4900, total=2.0324) (Degrees Per Second), M:(x=37.0000, y=-1777.0000, z=-182.0000, total=1795.5437) (milli gauus), P:(0.9973) (atm), H:(100.0000)(Relative Humidity)

130%|                        130%|                        130%|                        130%|                        130%|                        130%|
120%|                        120%|                        120%|                        120%|                        120%|                        120%|
110%|                        110%|                        110%|                        110%|                        110%|                        110%|
100%|                        100%|                        100%|                        100%|                        100%|                        100%|
 90%|                         90%|                         90%| | | | | | | | | | |     90%| | | | | | | | | | |     90%|
 80%|                         80%| | | | | | | | | | |     80%|                         80%| | | | | | | | | | |     80%| | | | | | | | | | |     80%|
 70%|                         70%| | | | | | | | | | |     70%|                         70%| | | | | | | | | | |     70%| | | | | | | | | | |     70%|
 60%|                         60%| | | | | | | | | | |     60%|                         60%| | | | | | | | | | |     60%| | | | | | | | | | |     60%| | | | | | | | | | |
 50%|                         50%| | | | | | | | | | |     50%|                         50%| | | | | | | | | | |     50%| | | | | | | | | | |     50%| | | | | | | | | | |
 40%|                         40%| | | | | | | | | | |     40%|                         40%| | | | | | | | | | |     40%| | | | | | | | | | |     40%| | | | | | | | | | |
 30%| | | | | | | | | | |     30%| | | | | | | | | | |     30%|                         30%| | | | | | | | | | |     30%| | | | | | | | | | |     30%| | | | | | | | | | |
 20%| | | | | | | | | | |     20%| | | | | | | | | | |     20%|                         20%| | | | | | | | | | |     20%| | | | | | | | | | |     20%| | | | | | | | | | |
 10%| | | | | | | | | | |     10%| | | | | | | | | | |     10%|                         10%| | | | | | | | | | |     10%| | | | | | | | | | |     10%| | | | | | | | | | |
     --------------------         --------------------         --------------------         --------------------         --------------------         --------------------
      1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10         1 2 3 4 5 6 7 8 9 10

        Accelerometer                Magnetometer                  Gyroscope                    Humidity                   Pressure                   Temperature

······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······ ······
Drone Structural Integrity Level (Health Level) Increased to 8


······ ······ ······ ······ ······ ······ ······ ······ ······ DRONE STATISTICS for COMMAND CENTER (HEALTHY STATE) ······ ······ ······ ······ ······ ······ ······ ······ ······

Flight Time: 5 Minutes | 53 Seconds | 616 Milliseconds
Structural Integrity Level (Health Level); 8
G:(x=0.2100, y=2.0300, z=-0.5600, total=2.1163) (Degrees Per Second), M:(x=34.0000, y=-1774.0000, z=-201.0000, total=1796.6296) (milli gauus), P:(0.9973) (atm), H:(100.0000)(Relative Humidity)
```

Figure 9: Shows the structural integrity of the drone being regenerated manually by pressing once on the button in standby mode from health level 7 to health level 8.

To prevent this outcome, while the drone is under attack, it can be rescued by rapidly pressing the button twice within a 0.5-second window. This action will interrupt the attack and regenerate the drone's structural integrity, restoring it to either 5 or its current health level if it exceeds 5 at the time of rescue. If the drone's structural integrity remains below 10 after rescue, it can be manually regenerated step by step by pressing the button during Standby mode.

3.4 Laser Gun Recharging and Shooting

Scenario 1: Laser gun is charged but insufficient battery to shoot

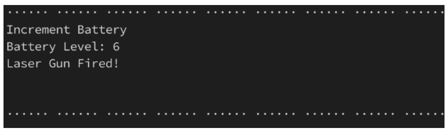


```
...... ...... ...... ...... ...... ...... ...... ...... .....
Increment Battery
Battery Level: 3

...... ...... ...... ...... ...... ...... ...... ...... .....
```

Figure 9: Charging of laser gun

In Battle Mode, upon triggering a button press once within 0.5 seconds, we charge the laser gun with 3 units of energy. Here, the new battery level is 3, and the laser gun does not get fired because each laser gun shot requires exactly 5 units of energy.

Scenario 2: Laser gun is charged and sufficient battery to shoot



```
...... ...... ...... ...... ...... ...... ...... ...... ......
Increment Battery
Battery Level: 6
Laser Gun Fired!

...... ...... ...... ...... ...... ...... ...... ...... ......
```

Figure 10: Charging and shooting of laser gun

If the new battery level is 5 or above, drone will fire the laser gun once, consuming 5 units of energy.

3.5 Drone Obstacle Detection and Automated Shooting

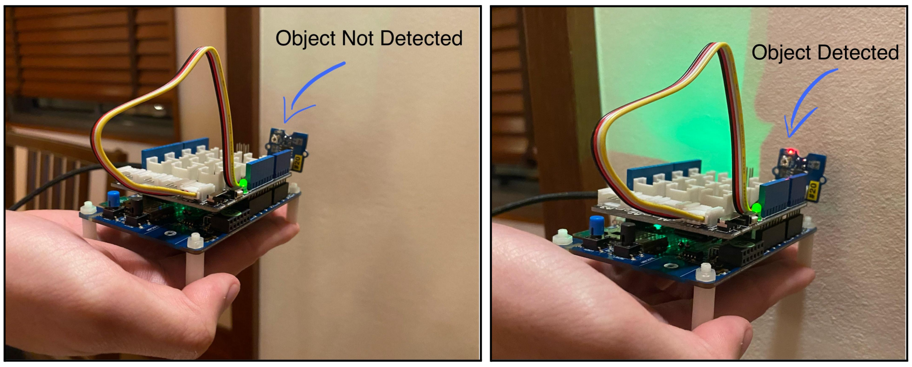


Figure 4 and 5: Red LED light turns on when an object is detected

During normal drone operation, if the Infrared Light Sensor detects an obstacle such as a zombie or obstacles, the drone will automatically charge the laser and fire a shot to clear the obstacle.

Section 4 : Enhancements Listing

Other than the basic requirements, we implemented various additional enhancements to our drone, as listed below.

- We added graphs to showcase sensor data trends over the past 10 seconds. This graph informs the pilots of the conditions of the physical environment that the drone is in, and pilots can use the information to determine if the drone is in danger more effectively. The data is represented in a graphical interface as a percentage with reference to the maximum threshold values, simplifying the process of identifying breached thresholds and ensuring that even complex data is presented in an easily understandable manner.

- We have developed a statistical dashboard which showcases the main overview of general metrics of the drone's operation such as flight time and structural integrity level as well as specific metrics and graphics of the drone's sensors and illustrations of how they vary over time for the command center.
    a. Flight Time - flight time of drone begins counting from the start of the flight in milliseconds and it then gets converted into a user-friendly format of days : hours : minutes : seconds : milliseconds
    b. Structural Integrity Level - showcases the current level of the structural integrity of the drone.
    c. Specific Metrics of the Sensors - showcases the detailed metrics values of each sensor in the adequately adjusted units which allows for detailed analysis, reference and comparison for the personnel of the command center (4 sensors in Standby Mode and 6 sensors in Battle Mode).

   d. Graphics of the Sensors - showcases the trends of the data in the past 10 seconds by allowing the command centre to compare the new values with the previous values. Also provides visual clarity of the environment that the drone is currently while performing the mission

- We implemented a structural integrity variable to track the health of the drone. This is to monitor and assess the overall condition of the drone's physical structure. The structural integrity variable serves as a real-time metric, providing a more realistic experience. By continuously monitoring this variable, we aim to enhance the experience for drone pilots, providing them with as much information about the drone as possible. When the health is too low, the drone pilot can move the drone to a safe place to initiate repair, and continue with the mission.
   a. In Standby mode, we implemented the “repair” feature. With a single button press (within 0.5 seconds), we will repair the structural integrity of the drone by 1 unit.
   b. In Battle mode, when the drone is successfully rescued from the state of “The Last of EE2028”, its structural integrity will be repaired to a maximum of 5 units (if it was lower than 5)

- We added an infrared reflex light sensor to the drone. When the sensor detects an object in front of the drone, if it is an obstacle or an enemy, the drone will charge its laser gun and shoot at it.

Section 5 : Challenges and Solutions Encountered

1. One of the issues faced was during the processing of data received from the sensors. The units of the raw values of the data was unknown to us and we had to locate the information in the respective datasheets. We then had to scale the raw values to ones that we can visualise easily for it to add value to the command centre.
2. Another significant challenge that we have faced was plotting the graphs of all the sensor values over time in the UART console in a horizontal format. We needed to plot all the graphs positioned horizontally next to each other, and make sure everything aligns neatly. Another obstacle was when we were deciding what would be the best y-axis to represent the data in a useful way, and eventually decided it to be a percentage of the maximum threshold values for each corresponding sensor.
3. We have learned how to use button interrupt in practice.

Section 6 : Conclusion

Our project focused on creating a prototype drone for both defensive and scientific purposes in a post-apocalyptic setting. We successfully built a functional drone capable of gathering environmental data using sensors, which is then represented on a comprehensive and user-friendly statistical dashboard.
This dashboard includes a large amount of general information, sensor metrics, and graphical representations of sensor data over time. The data is transmitted to the command center through UART for effective manual and automated monitoring of the drone.

Additionally, we integrated an integrity system to monitor the drone's health, allowing for regeneration following damage from attacks through rescue operations or manual intervention via a button press. Moreover, we implemented an infrared sensor system that enables the drone to detect and eliminate obstacles using laser guns, ensuring safe and obstacle-free operations in the challenging post-apocalyptic environment it was designed to navigate.

In conclusion, completing this assignment has been a valuable learning experience for us. We've delved into various fundamental concepts, such as configuring interrupts for button presses, the methods of sensor initialization and data acquisition, the importance of using the Systick timer for accurate time measurement (as opposed to HAL_Delay()), employing UART for effective communication, and representing data graphically in the console. Lastly, this project has taught us how to construct a complex system with multiple interconnected components by applying the newly learnt skills of Microcontroller Programming into practice while developing our team working and collaboration skills, and also harnessing our creativity to make a user-friendly user interface with as much useful information as possible.