Volume 6 Issue 1 (March 2026)
DOI 10.24191/jaeds.v6i1.168
eISSN 2805-5756

Journal of Applied Engineering Design & Simulation

Observation on Control and Navigation Capability Of a Tail-Less Blended Wing-Body UAV Equipped With 4G Internet Communication

R. E. M. Nasir[1,*], N. E. Abdul Rashid[2], S. Zainurin[3]
[1]Flight Technology and Test Research Group (FTTC), Fac. of Mech. Eng., Universiti Teknologi MARA, 40450 Shah Alam, Malaysia.
[2]Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia.
[3]CAE London Burgess Hill, Innovation Drive, York Road Burgess Hill RH15 9TW, United Kingdom.
[*]corresponding author: rizal524@uitm.edu.my

ABSTRACT

This paper highlights the development of flight navigation system using 4G communication protocol for a transport unmanned aerial vehicle (UAV) based on a tail-less blended wing-body (BWB) UAV. Normal radio-based transceiver of 2.4GHz or 433MHz frequencies are limited, by regulation, to low power that limits their communication range to just merely one to two kilometres. Long-range navigation, while possible is automated mode, cannot be observed in real time due to limited communication distance. The objective of this research is to observe capabilities of a UAV navigation system utilizing 4G Internet communication specifically to the Blended Wing-Body type of configuration that requires sophisticated control and active stabilization. The UAV features six control surfaces with mixing strategy that enables these surfaces to act as elevators, ailerons, rudders and airbrakes. To achieve this, Navio2-Raspberry Pi4 IMU-controller-computer with Internet Protocol (IP) from cellular network is integrated into the BWB UAV in which the former controls propeller speed, four elevons and a pair of split drag flaps. The results show that good navigational accuracies of each waypoint is within 10 metres except for four waypoints which fall outside of aircraft's manoeuvrability envelope, possibly due to manoeuvrability limitation. This successful integration opens promising possibilities for the future development and deployment of similar but larger UAV platforms, with improved efficiency and reliability in various applications.

Keywords: Blended Wing-Body; UAV Flight Control; Flight Navigation; Communication System; Internet of Things

Abbreviations

UAV/S	Unmanned Aerial Vehicle/System
GCS	Ground Control Station
GNSS/ GPS	Global Navigation Satellite System/Global Positioning System
BLOS	Beyond Line-Of-Sight
UTM	UAS traffic management
RTK	Real-Time Kinematics
IMU	Inertial Measurement Unit
SATCOM	Satellite Communication
ESC	Electronic Speed Controller
NAVCOM	Navigation & Communication

1.0 INTRODUCTION

An unmanned aerial vehicle (UAV) is defined as a pilotless aircraft which is flown without a pilot-in command on-board and is either remotely controlled (from ground, another aircraft or space) or programmed to fly automatically, or fully autonomous [1]. A Ground Control Station (GCS) enables UAV operators to communicate and control a UAV and its payloads by either allowing direct control of the UAV (manual) or setting parameters for autonomous operation. Many UAV software platforms are set up as "virtual cockpits" with representations of real instruments and displays that are commonly found in many aircraft, and GCS software that enable direct control of a UAV by an operator. A joystick for the aircraft or payload, a throttle controller, a keyboard, and a mouse are examples of flight control hardware that may be included in the GCS. Map display, instrument overlays, camera payload feeds, flight parameters, and a multitude of other data can all be viewed on the GCS. UAV ground control station system should have the following typical functions:

- Attitude control of the aircraft
- Mission planning, aircraft position monitoring, and map display of routes

Received on 12.02.2026
Accepted on 29.03.2026
Published on 30.03.2026

- Navigation and target positioning. The drone maintains contact with the ground control station through the wireless data link during the mission.
- Communication links with other subsystems.

Telemetry data, commands, and sensor data such as video, images, and flight parameter and engine health measurements may need to be transferred between the UAV and GCS. Some of the most common data transfer methods include analogue and digital radio communication, as well as cellular communications. Depending on transmission power, both have operational ranges that can reach hundreds of kilometres if designed properly. A stable and reliable flight control system (FCS), coupled with accurate and precise flight sensors such as Global Navigation Satellite System's (GNSS) sensors, accelerometers, gyroscopes, magnetic sensors, and pressure sensors can ensure seamless and accurate navigation.

Flight Technology and Test Centre in Universiti Teknologi MARA is currently developing an unmanned aerial vehicle to transport small parcels between cities in Malaysia. Due to its beyond line-of-sight (BLOS) operation, reliable communication between the UAV and its GCS is essential to ensure uninterrupted navigation and successful mission. Due to strict regulatory requirement from the Malaysian Communication and Multimedia Commission (MCMC) that only allows low-powered 433 MHz VHF radio that limits communication distance to less than 2 kilometres, and with vast network of telecommunication broadcasting tower in the country, it is hypothetically possible to device a 4G-Internet-based UAV-GCS communication system specifically for this purpose. The research here focuses on the qualitative functions of such communication system to control a dynamically unstable aircraft with four-elevon and twin split drag flaps setup for longitudinal and lateral-directional control, hence, enables precise navigation within its flight envelope.

1.1 Literature Review

Pavithran et al. [1] demonstrated that UAVs may be used to transport essential goods more safely, more quickly, and with greater energy efficiency. The cargo must fly outside the line of sight during transportation, necessitating the installation of an autonomous Pixhawk flying controller in the model with automatic flight guided by GNSS navigation. Real-time navigation monitoring requires BLOS communication between the UAV and the GCS. Abdalla and Marojevic [2] provided an overview of the current standardization efforts to support networked UAS. In particular, the specifications, proposed architecture, and services that are provided to/by UAVs and RCs as they connect via cellular networks to one another, with UAS traffic management (UTM), and with other users are devised. A pilot guiding an aircraft from within the aircraft is the foundation upon which Air Traffic Management (ATM) is built [3]. On the other hand, the growing demand for the use of unmanned aircraft systems (UAS) and their safe integration into segregated/non-segregated airspace have prompted concerns about the suitability of the current ATM for UAS Traffic Management (UTM). Ali [3] outlined the intended functions of UTM system by defining it and examining its potential communication, navigation, and surveillance (CNS) technologies to support it. A key to this is a well-functioning GCS. The three components that make up a well-functioning GCS are provided by Hong et al. [4] - a portable ground station hardware system, a virtual instrument panel for the display of attitude data and flight paths, and various error alerts. GCS is able to accurately display the remote sensing data and timely issue of the control command for the entirety of the test on the ground and in the air. The system can also be utilised to support a UAV's autonomous cruise mission – a pre-requisite to Intelligent Transportation System (ITS). UAVs are crucial to the operation of ITS. Gupta et al. [5] conducted a survey of recent UAV advancements and their applications in the present-day and presented foreseeable transportation systems. The debate highlights the benefits and problems of integrating UAVs into future ITS. A reliable, BLOS, and reprogrammable on-board flight control and navigation hardware is thus essential.

Benhadhria et al. [6] presented an autonomous navigation UAV built on the Raspberry Pi hardware and Android software. Depending on the circumstances of the assigned task, Android provides a wide selection of applications for the UAV to utilize directly. The proposed system determines the best paths to take, offers autonomous navigation without outside assistance, finds obstructions, and ensures live broadcasting during the operation. The requirements are put to the test through experiments. However, open-source software is becoming increasingly popular. A comprehensive, multi-layered, and open-source flying architecture was suggested by Kakamoukas et al. [7]. The architecture incorporates an integrated flight stack made up of the GCS side stack and the UAV side stack. A communication layer is additionally employed to control communication between the two side stacks. A simulated UAV is used to test the suggested architecture, which aims to be a software stack that would make it easier to construct more advanced UAV ideas like Flying Ad-hoc Networks (FANETs). In most cases, a ground operator in a ground control station (GCS) oversees controlling UAS. This necessitates that GCS be updated at the same pace as UAVs in order to keep up with advancements in technology [8]. In the meantime, the navigation system must also be able to decide the best navigation route. Pugliese et al. [9] examined three different transportation networks that gradually incorporate drone deliveries. They specifically address the issue of parcel delivery without a drone, also known as the vehicle routing problem, and the issue of a fleet of drones performing deliveries beginning at a central depot. An Internet cloud-based system with intelligent yet robust hardware and software may look like an obvious solution. Burke [10] demonstrated an open-source, 4G cloud-

connected, self-flying aircraft that is built on a safe, encrypted flight stack. The avionics section (which includes the flight controller, 4G cellular data modem, and on-board companion Linux computer) ensures GPS positioning, autonomous flight capability without the need for a pilot in close proximity to the UAV, and 4G secure encrypted robust self-healing Internet connectivity for video and telemetry. This is made possible by the convergence of developments in numerous hardware and software systems and disciplines, which are characterized by their inherently long flight times, inherently stable aerodynamic flight in the event of hardware failure, and Internet connectivity for control from virtually anywhere on Earth.

Chiueh et al. introduced a UAV positioning and navigation system utilising Zigbee for communication between UAVs, ground monitoring, and RTK-GPS stations. Addressing limited 4G coverage concerns in flight areas, the system employs Zigbee to transmit flight paths from ground monitoring, while the UAV uses ROS2 for flight guidance and PX4 for control. Precise RTK-GPS data are sent to the UAV via Zigbee from a portable station, achieving positioning accuracy within 10 centimetres and tracking accuracy under 60 centimetres in strong winds [12]. Increasing accuracy is crucial especially for land survey use. Yazid et al. [13] proposed a novel way of generating DTM and DSM by utilising unmanned aerial vehicle (UAV) for different land covers including forest, plantation and developed areas in the tropical region of Malaysia. To determine accuracy, quantitative and qualitative analysis are carried out by means of root mean square error (RMSE) and visual inspection. While transporting good purpose does not require such accurate waypoint-to-waypoint measurement, a good accuracy shall be within 10-metre radius from the intended waypoint. Kusmirek et al. [14] introduced the concept of a mountable device designed to track and assess the flight parameters of quadcopters, independent of the UAV's systems. Specifically, it involves independent monitoring of flight parameters. The device validation involved real flight tests with the IRIS + quadcopter using the Pixhawk control system, whose data are taken as a benchmark for validation. In the context of validating the operational envelope, the results showed that the parameters of real flights fall within the predicted area of the created operational envelope for the IRIS+ drone.

The efficacy of cellular internet signal propagation is critically influenced by the vertical positioning of the transmitting antenna relative to the surrounding environment and receiving devices [15]. This optimal height is a complex function of factors such as path loss, angle and delay characteristics, phase shift, and environmental attenuations caused by obstructions like foliage, concrete, and topographical features [16]. Consequently, determining an optimal base station height involves a trade-off: while higher elevations can mitigate ground-level obstructions and reduce through-crop attenuation, excessive height increases total propagation distance and angular deviation from the antenna's main lobe, both of which negatively impact received signal strength [17]. Furthermore, the altitude of the base station significantly influences propagation models, which are crucial for assessing power density, path loss, and coverage to optimize system performance and network planning [18]. This is particularly evident in urban environments, where the intricate interplay of building density and varying elevation profiles necessitates sophisticated modeling to accurately predict signal propagation and attenuation patterns [19]. The primary contributors to propagation loss, however, remain the propagation environment itself and the distance and direction to the nearest mobile phone tower [20], [21]. Investigations into optimizing cellular network coverage and minimizing interference often involve adjusting parameters such as transmit power and downtilt settings across various sectors to maximize reference signal received power and quality [22].

1.2 Objective

Based on aforementioned discussion, it is possible to actually set up similar cellular network-based IP communication system between the UAV and its GCS for a specific transport-type BWB configuration UAV. The development of UAV-GCS communication using 3G/4G cellular network at Universiti Teknologi MARA is not new and has been developed since 2017 with the first publication dated back in 2019 [11]. In this case, similar programming codes are included in this new research but with the latest IMU-controller hardware and Internet-of-Thing computer, faster 4G/5G internet network and most importantly more complex flight control system (four elevons, two split flaps and a motor against two elevons and a motor in the previous BWB UAV). While the use of Mission Planner suite is maintained for the flight control setup and navigation management, the flight control surface mixing strategy is different. This paper aims to qualitatively test the flight control and navigation of a tail-less BWB UAV with 4G-LTE or 5G cellular internet communication between UAV and its GCS since there is no suitable radio system for beyond line-of-sight (BLOS) range except satellite communication (SATCOM) using X, C or L bands. Operationally, the proposed system can make use of the vast Internet communication network available nationwide allowing for unlimited communication distance at relatively low cost. However, the flight operation is limited to low altitude depending on the location and altitude of the broadcast tower. High altitude operation, in this case, is not possible and one has to resort to a more "global" methods such as SATCOM or Over-the-horizon (OTH) radars. An 'Iron bird' is used as test bench before the system is integrated into the actual UAV. Navio2-Raspberry Pi4 IMU-controller-computer hardware with Internet Protocol (IP) is integrated into the UAV to achieve reliable BLOS communication with GCS enabling real-time flight control and ensuring precise navigation.

2.0 METHODOLOGY

2.1 Flow Chart and Schematics

The development is shown in Figure 1, beginning with the understanding of navigation and control system of UAV, followed by construction of the UAV's 'Iron Bird'. Iron Bird is a term used in the aerospace industry to refer to a ground-based test rig or platform that simulates the physical structure and systems of an aircraft. It is essentially a full-scale mock-up or prototype of an aircraft's structure, including its various mechanical, electrical, and avionics systems. The purpose of an iron bird is to conduct comprehensive testing, analysis, and validation of the aircraft's systems and components before the actual aircraft is built and flown. Iron bird testing allows engineers and technicians to evaluate the interactions and performance of different systems and components in a controlled environment. This testing phase helps identify any potential issues, design flaws, or system conflicts before they become problems in a real aircraft.

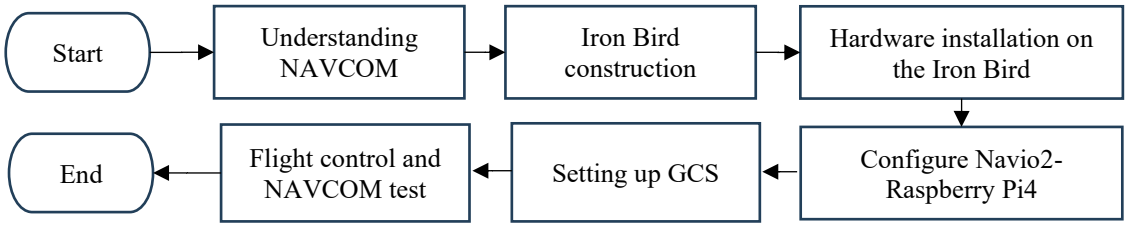

Figure 1. Methodology Flowchart

A comprehensive circuit diagram that serves as a guide for installing the flight control system is developed as shown in Figure 2. To facilitate initial testing and ensure a seamless transition to the final body design, the team opted for a temporary platform made of corrugated cardboard paper as the basis of the Iron Bird. This choice offers advantages of flexibility and cost-effectiveness.

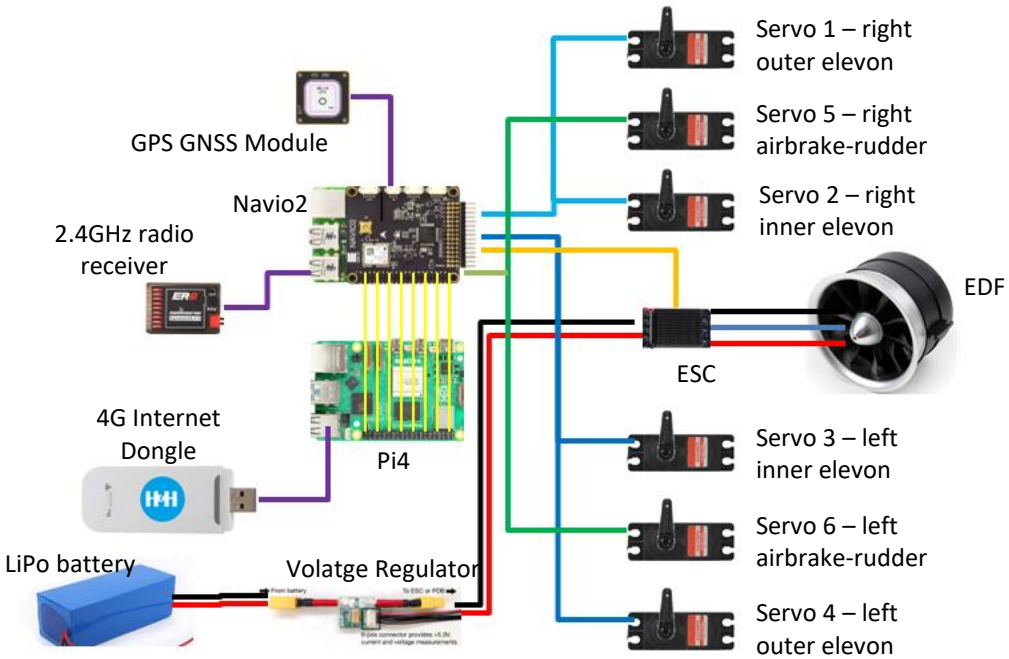

Figure 2. Flight control schematics

Each component, such as the battery, voltage regulator, ESC, and servos, is meticulously labelled to ensure precise and clear instructions during the construction phase. On board hardware consists of six servo motors, a pair of 24V 10Ah Lithium Battery, power module, Navio2-Raspberry Pi4 combo, receiver, 40A ESC, EDF and GPS GNSS Module. Unlike other conventional UAVs, this system does not use separate airbrakes and ailerons on wings, elevators and rudder on tail, but all control surfaces are located on the wing-body (tail-less aircraft) in which there are four elevons that function as aileron and elevator mixture and two split drag flaps that function as rudder and airbrake mixture. This novel flight control surface mix allows for more aerodynamically efficient

performance by reducing the need for separate horizontal and vertical tails. The UAV consists of 4 channel controlling servo and EDF:

- Channel 1 controls right elevons (servo 1 & servo 2)
- Channel 2 controls left elevons (servo 3 & servo 4)
- Channel 3 controls throttle (EDF)
- Channel 4 controls rudders (split drag flaps) (servo 5 & servo 6).

2.2 System Setup

All the hardware components of the Navio2 on the iron bird are shown in Figure 3. The Navio2 flight controller is mounted on Raspberry Pi4 computer and connected to various peripherals, including the GPS module, telemetry radio, ESC and servos. The Navio2 hardware requires a preconfigured Raspbian to run which can be referred in its website at https://docs.emlid.com/navio2/configuring-raspberry-pi/ [23]. NAVIO command interface and the process leading to Mission Planner GCS via Remote.it interface are also shown in Figure 3.

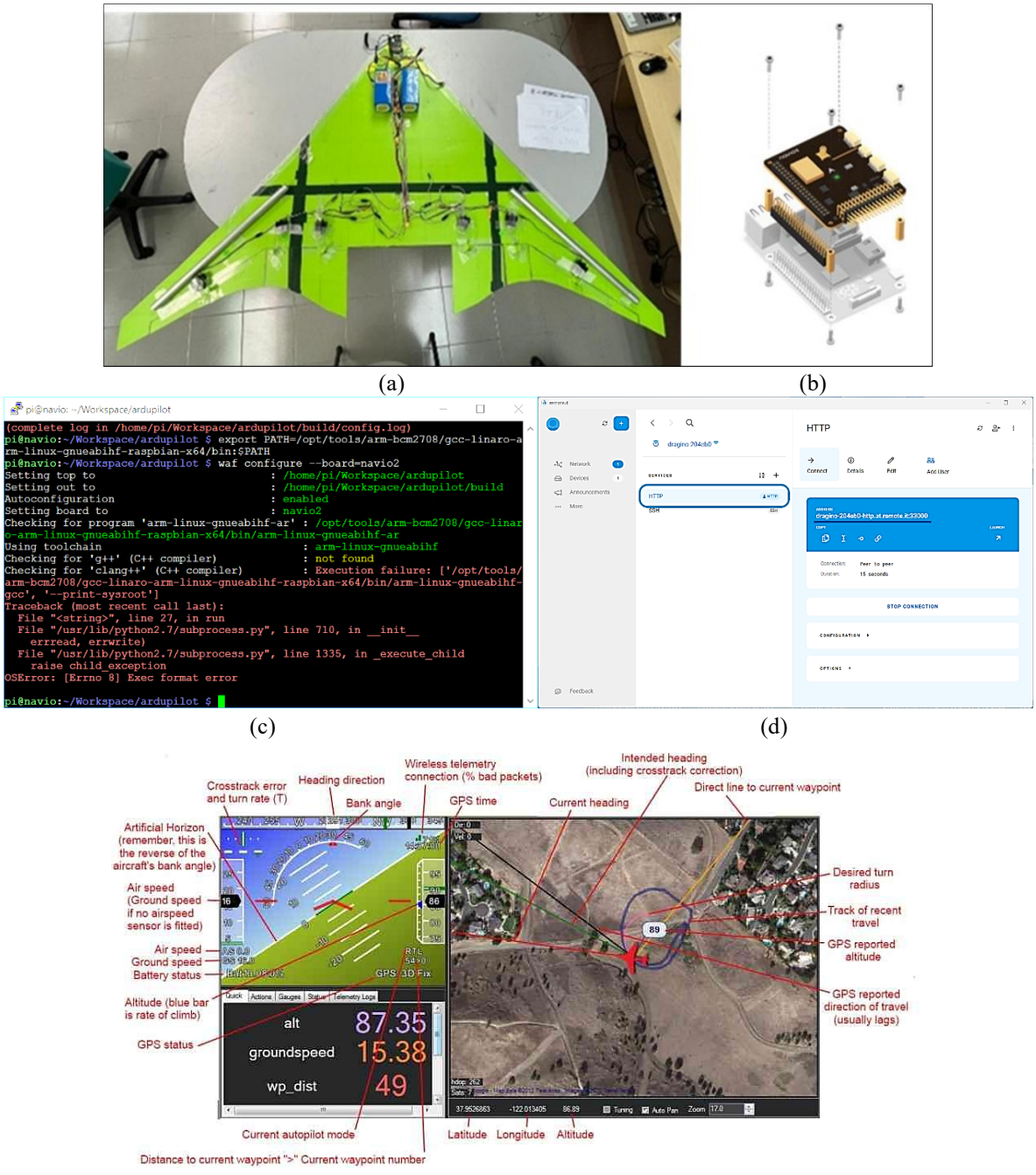

Figure 3. (a) the Iron Bird, (b) the Navio2 IMU/controller mounted on Raspberry Pi4 computer, (c) Navio2 programming interface, (d) Remote.it interface, and (e) Mission Planner GCS suite (ardupilot.org)

Raspberry Pi4 has an internal Wi-Fi module, and Wi-Fi networks can be configured to establish remote accessibility from the Internet. While several methods exist to achieve this, a prevalent approach involves port forwarding. This entails configuring GCS network router to expose a customized port and link it to UAV's IP address. While widely used, port forwarding introduces certain security apprehensions. To address these concerns, an alternative solution known as 'remot.it' is executed. This option furnishes remote entry to Raspberry Pi while also constructing network tunnels to the device's operational services—ranging from HTTP and VNC to SSH. This architecture allows secure access over the internet, circumventing the potential vulnerabilities associated with traditional port forwarding practices. A firewall serves as a partition separating an "internal" network (considered secure and trusted) from an "external" network (deemed unsecure and untrusted). Its primary function is to block undesired and unauthorized communication from entering or leaving the internal network. This design aims to thwart unauthorized interactions, consolidating security to a singular checkpoint.

An account with Remot3.it must first be created and registered. Once registered, the UAV is then connected on-demand, from anywhere, without port forwarding or leaving open ports on router. A "public endpoint" in remote.it refers to a network address that is accessible over the internet. It is a unique address that remote.it assigns to a device or service to enable remote access. Once the NAVIO2-Raspberry Pi4 on board the UAV is connected to the GCS computer via remote.it protocol that establishes BLOS communication (as long as there is Internet service on both), the Mission Planner software is opened to configure the flight control and navigation system which also serve as the GCS software.

On top of the Iron Bird, the core components like gyro meters, accelerometers, magnetic compass, GNSS sensors, and Internet Modem are positioned. These sensors play a crucial role in manipulating the vehicle's attitude. Calibration of these sensors is performed before assigning servo outputs for controlling the elevons – the control surfaces that combine aileron and elevator functions. To power the system, a 24V 10Ah lithium battery is utilized along with a voltage regulator/power module that powers Navio2-Raspberry Pi4 computer, sensors and servos.

The flight control's manual input is transmitted from the GCS to the UAV using a control joystick on the ground through the GCS computer's Internet Network to the modem connected to Raspberry Pi4 via remote.it protocol. These signals are then channelled into the Navio2 flight controller. Upon receiving the command, the controller executes it by regulating the EDF and deflecting the elevons and split drag flaps. These configurations provide the necessary flight dynamics information for the UAV, encompassing crucial parameters like pitch angle, pitch rate, roll angle, roll rate, yaw angle, yaw rate (p, q, r) and speed components (u, v, w). PWM values are utilised to compute the aircraft's movement angles and rates. However, determining its speed requires reliance on GPS data and the controller's internal clock. Furthermore, the flight controller possesses the capability to store all recorded flight dynamics data within its internal storage, facilitated by an SD card. The flight movements can be done by controlling the servos that are assigned respectively. Note that Oe, Re, Rr, Lr, and Le stand for outer elevon, right elevon, right rudder, left rudder and left elevon, respectively. There are two kinds of functionality tests executed and discussed here – 1. Flight control functions in which the control stick is moved to command pitch, roll, yaw and accelerate to the UAV and the motion of control surfaces (hence control servos) is recorded and observed on the Iron Bird. – 2. Navigation functions in which the discussed flight control and navigation systems are installed in a 1:2 scaled-down proof-of-concept (POC) model of the UAV to be flown automatically on a preset path. The actual flight path is recorded and plotted later to be evaluated qualitatively.

2.3 Flight Plan

The reason behind using the half-sized flying model is because the actual prototype is still under construction while it only takes two weeks to print and integrate the 1:2 scale smaller-sized ones. Secondly, since it is cheaper and lighter in terms of mass (at just 1.5 kilograms and 0.25 m^2 wing area, the stall speed is 8.0 m/s at C_{Lmax} = 1.0), the risk of causing injury to people if the small UAV crashed is much lower than the full-sized 15 kilograms MTOW prototype. Figure 4 shows the 1:2-scale POC model flown to test the functionality of the navigation system.

A flight test session was conducted at Politeknik Banting, Selangor (PBS). The test, which was authorized by CAAM (see Attachment), was cleared to be carried out at maximum approved altitude of 400ft AGL and was conducted within Visual Line of Sight (VLOS) of the Remote Pilot and the Visual Observers. The flight test was meticulously planned and executed. The flight plan (see Figure 5) was designed with a specific altitude of 20 meters Above Ground Level (AGL), ensuring a safe and controlled flight within the designated airspace. The total distance covered during the test was 1.33 kilometres, a substantial distance that allows for comprehensive testing of the UAV's capabilities. The flight plan parameters (Table 1) are structured to include two full circuits. This repetition provides an opportunity to assess the UAV's performance under consistent conditions and allows for the detection of any anomalies or deviations in flight patterns. A key feature of the test is the inclusion of a single loiter point. At this point, the UAV is programmed to make two turns, each with a radius of 50 meters.

This manoeuvre tests the UAV's ability to maintain a precise path and direction, crucial for real-world applications where accurate navigation is paramount. The take-off and landing phases of the flight are conducted manually. This approach ensures that the most critical phases of flight are under direct human control, allowing for immediate intervention if necessary. Once airborne, the UAV switches to an automatic mission mode. This mode allows the UAV to follow a pre-programmed flight path at a consistent speed of 16 metres per second, testing its automated navigation capabilities. The flight plan is designed with nine distinct waypoints. These waypoints guid the UAV along its flight path, testing its ability to reach specific locations accurately and efficiently. Each waypoint serves as a checkpoint, ensuring that the UAS correctly follows its programmed course. Overall, the UAV flight test is a comprehensive examination of the system's capabilities, providing valuable data for further refinement and development.

Figure 4. 1:2 scale POC model and navigation functionality test

Figure 5: Flight Plan

Table 1: Flight Plan Parameters

WP	Command	Lat	Lon	Alt	Grad%	Angle	Dist.	AZ
1	Waypoint	2.8257197	101.4668888	20	27.1	15.2	76.3	120
2	Waypoint	2.82605195	101.46690369	20	0.0	0.0	37.0	3
3	Waypoint	2.82689857	101.46692657	20	0.0	0.0	94.2	2
4	Waypoint	2.82680202	101.46798706	20	0.0	0.0	118.3	95
5	Loiter_Turns	2.82576251	101.46800995	20	0.0	0.0	115.6	179
6	Waypoint	2.8252697	101.467967	20	0.0	0.0	55.0	185
7	Do_Jump	-	-	-	-	-	-	-
8	Waypoint	2.8252375	101.4670819	15	-5.1	-2.9	98.5	268
9	Waypoint	2.8252857	101.4663202	10	-5.9	-3.4	84.9	274
10	Land	-	-	-	-	-	-	-

Description of flight plan is given below;

- Waypoint 1: The UAS starts at a latitude of 2.8257197 and a longitude of 101.4668888 at an altitude of 20 meters. The gradient is 27.1%, the angle is 15.2 degrees, the distance is 76.3 meters, and the azimuth is 120 degrees.
- Waypoint 2: The UAS moves to a latitude of 2.82605195 and a longitude of 101.46690369 at the same altitude of 20 meters. The gradient and angle are both 0.0, and the distance to the next waypoint is 37.0 meters. The azimuth is 3 degrees.
- Waypoint 3: The UAS moves to a latitude of 2.82689857 and a longitude of 101.46692657, maintaining the same altitude. The gradient and angle remain 0.0, and the distance to the next waypoint is 94.2 meters. The azimuth is 2 degrees.
- Waypoint 4: The UAS moves to a latitude of 2.82680202 and a longitude of 101.46798706, still at 20 meters altitude. The gradient and angle are 0.0, and the distance to the next waypoint is 118.3 meters. The azimuth is 95 degrees.
- Waypoint 5 - Loiter_Turns: The UAS moves to a latitude of 2.82576251 and a longitude of 101.46800995, maintaining the same altitude, and loiters (or circles) for 2 turns. The gradient and angle are 0.0, and the distance to the next waypoint is 115.6 meters. The azimuth is 179 degrees.
- Waypoint 6: The UAS moves to a latitude of 2.8252697 and a longitude of 101.467967, still at 20 meters altitude. The gradient and angle are 0.0, and the distance to the next waypoint is 55.0 meters. The azimuth is 185 degrees.
- Waypoint 7 - Do_Jump: The UAS is commanded to jump back to a previous waypoint, in this case, waypoint 1.
- Waypoint 8: The UAS moves to a latitude of 2.8252375 and a longitude of 101.4670819, but this time at a lower altitude of 15 meters. The gradient is -5.1%, the angle is -2.9 degrees, the distance to the next waypoint is 98.5 meters, and the azimuth is 268 degrees.
- Waypoint 9: The UAS moves to a latitude of 2.8252857 and a longitude of 101.4663202, further decreasing altitude to 10 meters. The gradient is -5.9%, the angle is -3.4 degrees, the distance to the next waypoint is 84.9 meters, and the azimuth is 274 degrees.
- Waypoint 10 Land: The UAS is commanded to land.

In summary, this flight plan involves the UAS moving between a series of waypoints at different altitudes, performing a loitering maneuver, jumping back to a previous waypoint, and finally landing. The flight plan also includes information on the gradient, angle, distance, and azimuth for each maneuver. The UAV Log Viewer is a tool used to analyse and visualise flight logs from Unmanned Aerial Vehicles (UAVs), It can process data from various types of flight log formats and display the information such as altitude, speed, GPS position, battery status, and many more. It can also plot these parameters over time or against each other to help understand the dynamics of the flight.

3.0 RESULTS AND DISCUSSION

3.1 Ground Test

Figure 6 shows servo channel output values during testing. The values are in Pulse width modulation (PWM). Servo output response greatly with the transmitter during testing. PWM = 900 corresponds to minimum servo position (-90 deg.), meaning that the control surface is deflecting upwards in elevon setup. Meanwhile, PWM=2100 is in its maximum servo position (+90 deg.), meaning that the control surface is deflection downwards. The red line is channel 1 output represents pitch control, green line is channel 2 output represents roll control and blue line is channel 4 output represents yaw control. Channel 1 and Channel 2 output, theoretically,

shall have the same values in elevator mode function but here there are slight differences due to initial aileron trim setup in which one side of elevons must have slightly more deflection angle than the opposite side due to motor torque factor. This is normal for a single-engine setup. The servo control plots here show sound and consistent PWM values reflecting consistent elevons and flaps deflections at any time.

Figure 6: Channel output test

Figure 7(a) shows both outer elevon (Oe), right elevon (Re) and left elevon (Le) deflected upwards while right rudder (Rr) and left rudder (Lr) remain at neutral point. This shows that all elevons are currently functioning as elevators that make the UAV pitch up. In Figure 7(b), the right side elevons go up while the left side elevons go down, indicating the function of aileron which is to cause the UAV to roll or bank to the right. Pictures within Figure 8(a) and (b) show the effect of roll left and roll right command on the iron bird. Longer distance of outer elevons from the centre produces larger rolling moment than the inner elevons deflected at the same angle magnitude; thus, to ensure similar rolling moment effect, the outer elevons have smaller deflection magnitude than their inner counterparts.

Since the UAV is using split drag flap mechanism, only one side of rudder is deflected open to produce yaw because asymmetrical drag is produced between left and right rudder at the moment. Figure 7(c) shows the yaw movement by deflecting the right rudder, while the left rudder remains undeflected. This makes the UAV yaw to the right. Figures 8(c) and (d) show the effect of right and left yaw commands to the left and right split drag flaps. Meanwhile, for air brake function, both left and right rudders are deflected open at the same deflection angle magnitude to increase drag on both sides of the wing (Figure 7(d)).

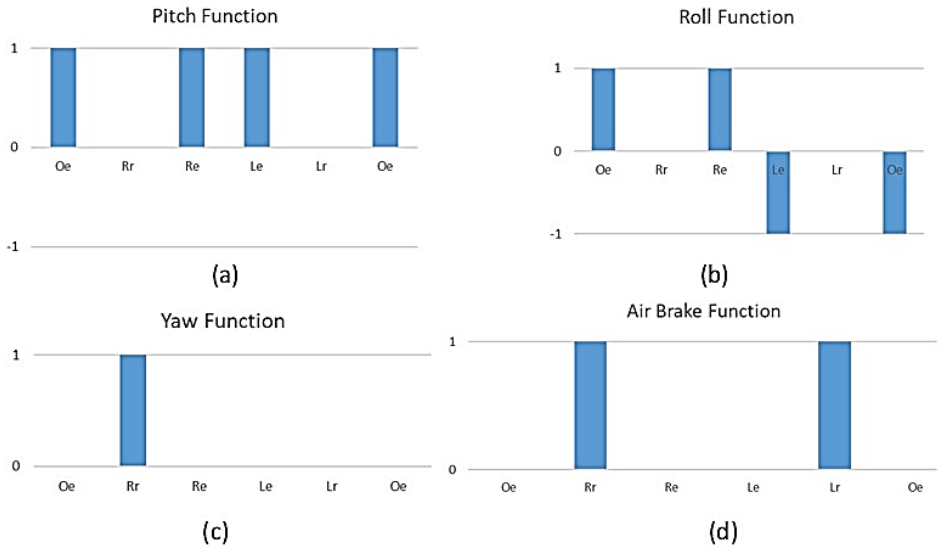

Figure 7: Control surface function test results (command signal)

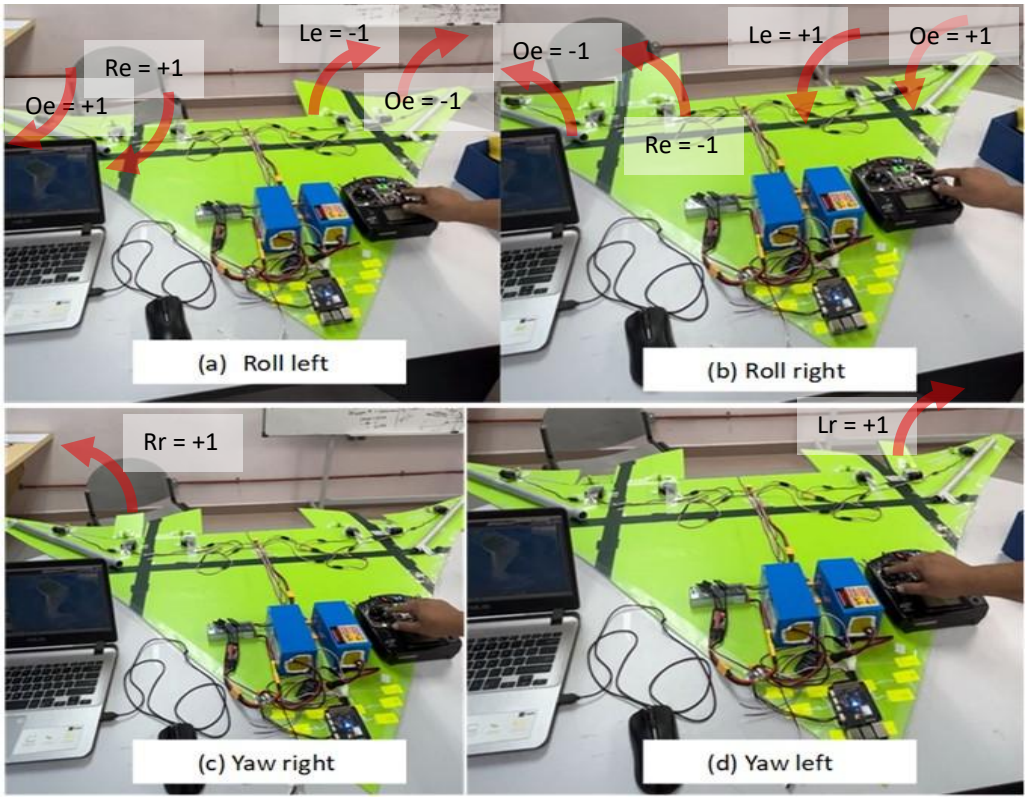

Figure 8: Control surface function test results (deflections on the 'iron bird')

3.2 Flight Test

Flight tests were done by flying the Navio2 with GPS module inside the 1:2 scale POC model UAV following preset waypoints. A total of 19 waypoints covered with 15.4 km of distance travelled at average of 15.6 m/s ground speed. For information, this POC model has a minimum speed (near stall) of 8.0 m/s. The UAV's actual flight path is compared with the planned flight path. Any deviations are noted and analysed. There are four flight modes used in the flight test. Here are the brief descriptions of each flight mode;

- Manual: In this mode, the pilot has direct control over the UAS's movements. The pilot's inputs to the controller are translated directly into changes in the UAS's throttle, pitch, roll, and yaw. This mode requires the most skill to use effectively and is typically used for take-off and landing, or for flying in complex environments.
- Fly By Wire A (FBWA): In this mode, the pilot's inputs to the controller are used to control the UAS's attitude (i.e., its pitch, roll, and yaw) and throttle, but the UAS's onboard flight controller also provides some assistance. For example, it might automatically stabilize the UAS if it starts to roll or pitch too much. This mode is easier to use than manual mode and is typically used for general flying.
- Auto: In this mode, the UAS's onboard flight controller fully controls the UAS's movements based on a pre-programmed flight plan. The flight plan typically includes series of waypoints that the UAS will fly to, along with other commands such as changes in altitude or speed. This mode requires the least skill to use and is typically used for long-distance flights or for carrying out specific tasks (such as aerial surveying).
- Return to Land (RTL): In this mode, the UAS's onboard flight controller automatically flies the UAS back to its take-off point and then lands it. This mode is typically used in emergencies (for example, if the communication link with the UAS is lost) or at the end of a flight. The UAS's flight controller uses GPS to navigate back to the take-off point and may also use other sensors (such as altimeters or rangefinders) to land safely.

Table 2 shows the flight phase duration of selected modes executed in this flight test. Note that manual mode is not executed and serves only to test the effectiveness of control before ethe actual flight test is conducted. Total flight time is 250 seconds or slightly more than 4 minutes. For the given battery capacity, the aircraft should be able to fly slightly more than 5.0 minutes. Maximum flight time is calculated based on average current consumed (based on thrust required/drag on cruising speed) and battery capacity.

Table 2: Flight Phase Duration

Flight Phase	1	2	3	4	Total
Flight Mode	FBWA	AUTO	RTL	FBWA	
Duration	22 s	144 s	34 s	50 s	250 s

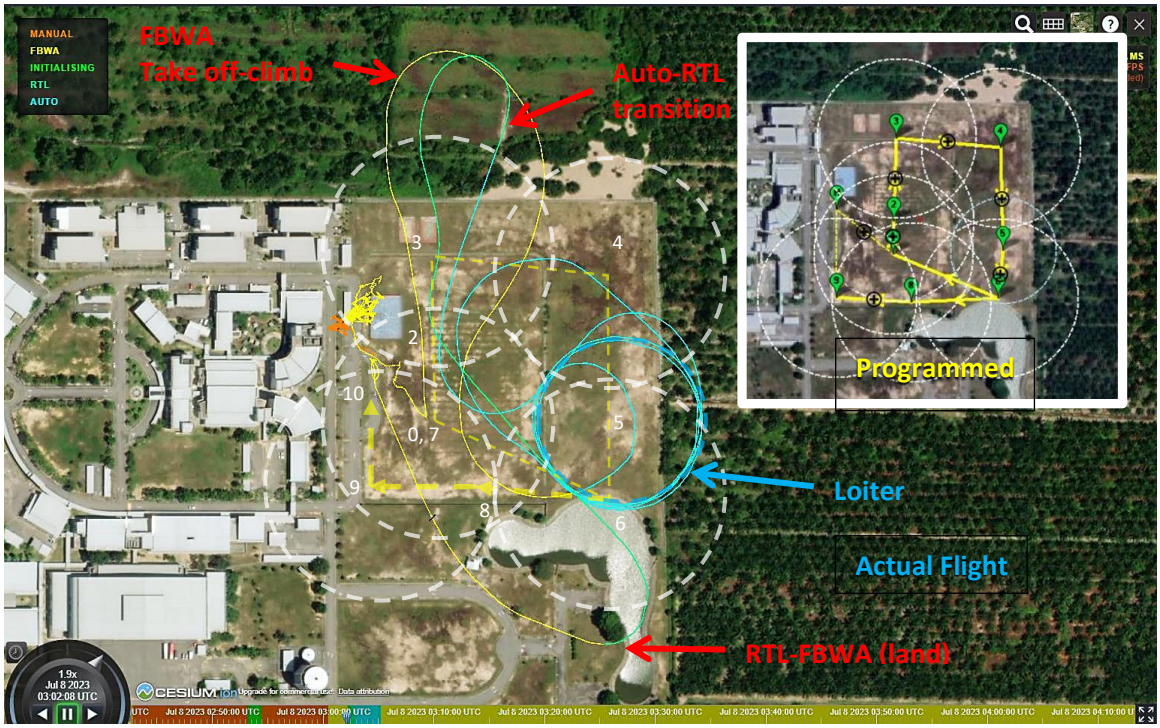

Figure 9: Actual recorded path of the tested UAV

The result of implementing flight navigation, communication, and ground control using Navio2 and Mission Planner on the UAV is fairly accurate (withing GPS accuracy white circle) except at three regions (shown with red arrow) within these waypoints (highlighted in yellow, dashed lines) in which the aircraft veered away from the waypoint due to aircraft manoeuvrability limitation. The yellow-blue-green lines in Fig. 9 indicate the actual path of the POC UAV that the GCS can receive coordinates from the on-board system and locate the UAV during the test in real-time. This is similar, in terms of accuracy of 1 to 10 metre on waypoints, to results obtained in [14] in which the author used IRIS+ and Pixhawk hardware on board of a quadcopter instead of NAVIO2-plus-Raspi 4 setup on BWB UAV here. Ten metre maximum waypoint error is an acceptable number since this system is only equipped with two GNSS (one internal within NAVIO and one external GNSS which are able to read NAVSTAR GPS and GLONASS). There is no RTK system in this case just like in [12] and [13] in which RTK is integrated with Zigbee 4G communication with ROS2 navigation system and Pixhawk control hardware, allowing 10 to 60 cm accuracy even under strong winds. However, on different platform, the flight speed of 20 m/s is similar to the case here. In fact, both [12], [14], and this study have aircraft weight of 1.3 to 1.5 kg. Reliable communication is established, and the navigation system works well within acceptable waypoint accuracy the UAV's mission profile programmed except on four waypoints – initial climb right after taking off at waypoint 3 and 4 where it overshot the former waypoint and skipped the latter, and another overshot at the later phase (final approach to land) at waypoint 6, skipping waypoint 8 and proceeded to land from waypoint 9 onwards. The cause of those are unknown but might possibly be due to manoeuvrability limitation of this aircraft. The distance between those two waypoints were short, the automatic flight command was in transition phase (take off to cruise, and cruise to land), and the being a BWB-type aircraft, the wingspan was longer than its length making tight roll manoeuvre quite a challenging task.

Fig. 10 shows the histograms of altitude and airspeed versus time in which cyan graph represents altitude while magenta plot represents groundspeed. Despite programming constant altitude and airspeed during cruising phase from the ground control station, the altitude plot shows repeated oscillatory motion with smaller, higher frequency oscillation "sitting" on the larger amplitude, lower frequency oscillation. There are two flight modes that appear here – the smaller oscillations represent short-period mode with sub-second period of oscillation that seems to never stop, and the large 10-15 metre amplitude oscillations with approximately 30-second period of oscillation that is also not quite settled. The reason is that tail-less BWB aircraft is inherently dynamically unstable (although

it is statically stable) and its elevons keep moving all the time to make corrections. The centre of gravity position, which is quite close to its neutral point, did not help to stabilize the aircraft at all. The CG was designed so that some manoeuvrability requirements could be achieved; thus, the aircraft will be fairly unstable. This may be remedied with more advanced flight control algorithm or just a little bit of fine tuning on its PID controller.

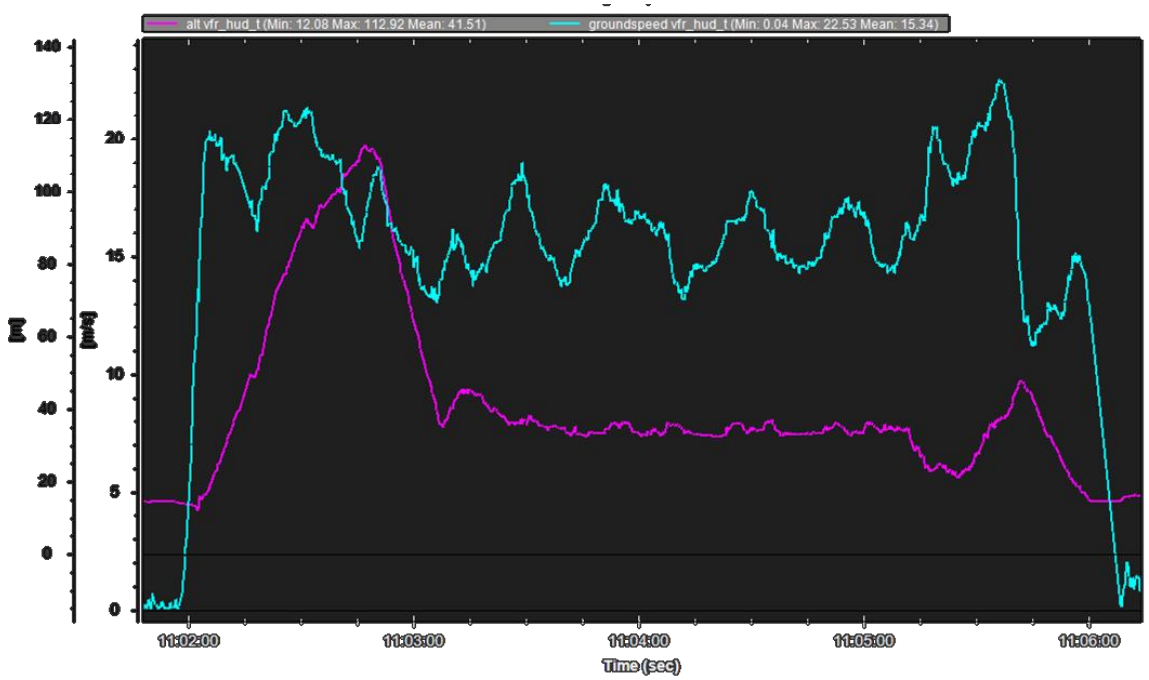

Figure 10: Altitude and groundspeed history

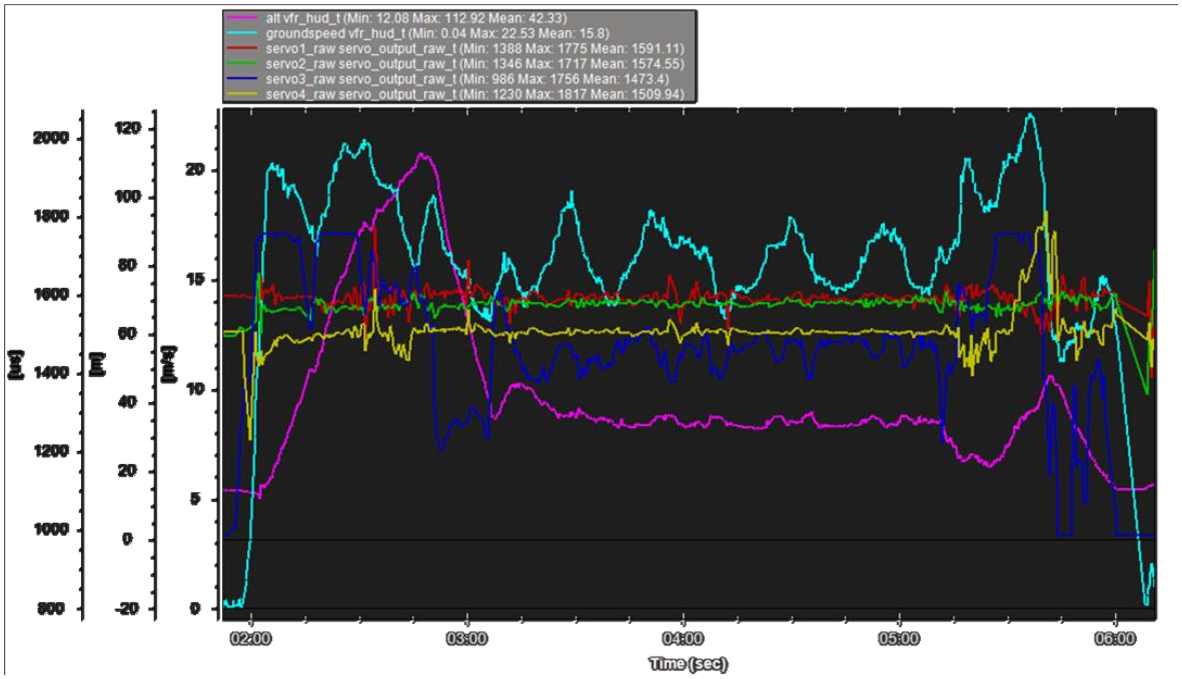

Figure 11: Altitude, groundspeed and servo output (channel 1 – servo 1 &2 right elevons, channel 2 – right servo 3 & 4 left elevons, channel 3 – throttle, channel 4 – servo 5 & 6 drag rudders).

Fig. 11 looks like Fig. 10 but with additional four servo command outputs (in μs) representing motion of servos 1 & 2 (right elevons) from channel 1 (red), servos 3 & 4 (left elevons) from channel 2 (green), throttle from channel 3 (blue) and servos 5 & 6 (split drag rudder) from channel 4 (yellow). Servos 1 and 4 represent outboard elevons of left and right while servos 2 and 3 represent inboard elevons on both sides; thus, Channels 1 and 2 are related to all these four servos and they have quite similar motion trends in terms of amplitude. Channel 3 (throttle, blue), on the other hand, does not only have different trend but has more erratic with large amplitude

oscillations between PWM = 900 to 1700 μs because the throttle may vary widely from zero thrust (PWM = 900) to maximum thrust (PWM = 2100). Channel 4 (yellow) is quite "calm" at around PWM = 1500 μs because the autopilot seldomly uses rudder function. Despite these findings, the flight seemed stable and was handled well by the computer.

4.0 CONCLUSION

As a conclusion, the integration of Navio2-Raspberry Pi4 combo and Mission Planner GCS in the flight control and navigation, BLOS communication via the Internet Protocol of the proposed BWB UAV demonstrated good results. This successful integration opens promising possibilities for the future development and deployment of similar but larger UAV platforms, with improved efficiency and reliability in various applications. For future work, the team will continue refining and optimizing the system. This could involve further enhancing the compatibility between Navio2-Raspberry Pi4 and the Mission Planner suite to streamline their interaction. Additionally, conducting thorough testing across diverse scenarios and environmental conditions will help uncover potential challenges and areas for improvement.

ACKNOWLEDGEMENT

This research is funded by KEPU Grant (600-RMC/KEPU 5/3 (004/2021)) from Research Management Centre (RMC) of Universiti Teknologi MARA and by Prototype Research Grant Scheme (PRGS) from the Ministry of Higher Education of Malaysia (600-RMC/PRGS 5/3 (004/2022)).

AUTHORS CONTRIBUTION

R. E. M. Nasir: Flight Control & Navigation System; Writing Final Draft & Editing; Corresponding Author; N. E. Abdul Rashid: Communication System; S. Zainurin: Aviation Electronics, International Research Collaborator.

DECLARATION OF COMPETING OF INTEREST

The manuscript has not been published elsewhere and is not under consideration by other journals. All authors have approved the review, agree with its submission, and declare no conflict of interest on the manuscript.

REFERENCES

[1] R. Pavithran, V. Lalith, C. Naveen, S. P. Sabari, MST Ajay Kumar, and V. Hariprasad, ″A prototype of Fixed Wing UAV for delivery of Medical Supplies,″ in *International Conference on Mechatronics in Energy and Environment Protection,* Erode, India, 2020, pp. 012015.

[2] A. S. Abdalla, and V. Marojevic, ″Communications standards for unmanned aircraft systems: The 3GPP perspective and research drivers″, *IEEE Communications Standards Magazine,* vol. 5, no. 1, pp. 70-77, 2021.

[3] B. Syd Ali, "Traffic management for drones flying in the city", *International Journal of Critical Infrastructure Protection,* vol. 26, pp. 100310, 2019.

[4] Y. Hong, J. Fang, and Ye Tao. ″Ground control station development for autonomous UAV″, in *International Conference on Intelligent Robotics and Applications*, Hangzhou, China, 2008, Springer Berlin Heidelberg.

[5] A. Gupta, T. Afrin, E. Scully, and N. Yodo, ″Advances of UAVs toward future transportation: The state-of-the-art, challenges, and opportunities″, *Future Transportation*, vol. 1, no. 2,pp. 326-350, 2021.

[6] S. Benhadhria, M. Mansouri, A. Benkhlifa, I. Gharbi, and N. Jlili, ″Vagadrone: Intelligent and fully automatic drone based on raspberry pi and android″, *Applied Sciences,* vol. 11, no. 7, pp. 3153, 2021.

[7] G. Kakamoukas, P. Sarigiannidis, and I. Moscholios, ″High level drone application enabler: An open source architecture″, in *12th International Symposium on Communication Systems, Networks and Digital Signal Processing*, 2020, © IEEE.

[8] A. Mohammed, B. Garba Ibrahim, M. Omuya Momoh, K. Peter Ter, A. O. Adetifa, and D. A. Oluwole, ″Challenges of Ground Control System in Ensuring Safe Flights for Unmanned Aerial Vehicles″, *Journal of Mechatronics and Intelligent Manufacturing (Mekatronika)*, vol. 4, no. 1, pp. 8-19, 2022.

[9] L. D.P. Pugliese, Fr. Guerriero, and G. Macrina, "Using drones for parcels delivery process", *Procedia Manufacturing*, vol. 42, pp. 488-497, 2020.

[10] P. J. Burke, ″A safe, open source, 4G connected self-flying plane with 1 hour flight time and all up weight (AUW)< 300 g: towards a new class of internet enabled UAVs″, *IEEE Access*, vol. 7, pp. 67833-67855, 2019.

[11] S. Zainurin, R. E. M. Nasir, A. B. A. Muta'ali, W. Wisnoe, W. Kuntjoro, "Performance of 4G-LTE Communication and Navigation System in Blended Wing-Body UAS", *International Journal of Recent*

Technology and Engineering, vol. 8, no. 4, pp. 9538-9542, 2019.

[12] H. -L. Chiueh, C. -H. Wu, Z. -D. Xie and H. -W. Xu, "Implementation of UAV Positioning And Navigation System Using Zigbee Communication," *2024 4th International Conference on Electronics, Circuits and Information Engineering (ECIE)*, Hangzhou, China, 2024, pp. 358-362.

[13] Yazid, A.S.M., Wahid, R.A., Nazrin, K.M., Ahmad, A., Nasruddin, A.S., Rozilawati, D., Hamzah, M.A. and Razak, M.Y.A., Terrain mapping from unmanned aerial vehicles. *Journal of Advanced Manufacturing Technology (JAMT)*, *13*(1), 2019. pp.1-16.

[14] S. Kusmirek, V. Socha, T. Malich, L. Socha, K. Hylmar and L. Hanakova, "Dynamic Flight Tracking: Designing System for Multirotor UAVs With Pixhawk Autopilot Data Verification," in *IEEE Access*, vol. 12, pp. 109806-109821, 2024

[15] P. Bondalapati et al., "SuperCell: A Wide-Area Coverage Solution Using High-Gain, High-Order\n Sectorized Antennas on Tall Towers," arXiv (Cornell University), Nov. 2020.

[16] S. Ali, A. Abu-Samah, N. F. Abdullah, and N. L. M. Kamal, "Propagation Modeling of Unmanned Aerial Vehicle (UAV) 5G Wireless Networks in Rural Mountainous Regions Using Ray Tracing," *Drones*, vol. 8, no. 7, Jul. 2024. pp. 334

[17] E. Sie et al., "BYON: Bring Your Own Networks for Digital Agriculture Applications," arXiv (Cornell University), Feb. 2025.

[18] F. A. Almalki and M. C. Angelides, "Deployment of an aerial platform system for rapid restoration of communications links after a disaster: a machine learning approach," *Computing*, vol. 102, no. 4, Nov. 2019. pp. 829.

[19] J. Warczek, J. Kozuba, M. Marcisz, W. Pamuła, and K. Dyl, "Research on Mobile Network Parameters Using Unmanned Aerial Vehicles," *Sensors*, vol. 24, no. 17, Aug. 2024. pp. 5526.

[20] B. Nesa, "Analysis of Signal Strength Variations for an Urban Public University Campus in Bangladesh," *International Journal of Engineering Research*, no. 7, Jul. 2020.

[21] T. Jawhly and R. C. Tiwari, "The special case of Egli and Hata model optimization using least-square approximation method," *SN Applied Sciences*, vol. 2, no. 7, Jun. 2020.

[22] R. M. Dreifuerst et al., "Optimizing Coverage and Capacity in Cellular Networks using Machine Learning," May 2021. pp. 8138

[23] Raspberry Pi configuration, https://docs.emlid.com/navio2/configuring-raspberry-pi/ [accessed on 26 March 2026]