The Development of a Node-RED-Based Dashboard for Real-Time Monitoring and Control of Air Data Test Set (ADTS) Based on IoT

Rival Elfais Prayogo, Simon Siregar *, Ema, Fahmi Pratama Khair
and Aufa Nur Faiz Yudhantoro

Department of Computer Engineering, School of Applied Sciences, Telkom University, Bandung, West Java, 40257, Indonesia

***Corresponding Author E-mail:** simon.siregar@tass.telkomuniversity.ac.id

Received: Jul 04, 2025; **Revised:** Sep 11, 2025; **Accepted:** Sep 22, 2025

Abstract

Aviation accidents are often caused by incorrect airspeed readings due to inaccurate pitot-static sensors. This study develops an IoT-based Air Data Test Set (ADTS) for real-time airspeed calibration, utilizing an ESP32, MS5803 sensor, MQTT protocol, and Node-RED interface. Unlike previous studies limited to local monitoring, Node-RED in this study functions as an interactive control hub as well as an integrated visualization platform with automatic logging to MySQL, thereby enhancing the system's reliability and accessibility. Experiments were conducted over a range of 20–140 knots with five repetitions per data point, resulting in communication latencies of 200–500 ms and high accuracy, with pressure-to-speed conversion errors ranging from 0.09% to 14%. The largest deviation occurred at low speeds (40 knots, −14%), whereas at speeds above 70 knots, errors remained below ±1%. With features such as remote control, real-time monitoring, and automatic logging, this system provides a practical calibration tool for laboratories and educational purposes, while also laying the groundwork for further development under more realistic flight conditions.

Keywords: Air Data Test Set, Pitot-static, Node-RED, ESP32, MS5803

1. Introduction

Aviation safety remains a top priority in the global aviation industry. Numerous incidents and accidents have been linked to instrument discrepancies or sensor malfunctions in aircraft systems [1]. Human factors, particularly pilot attention distribution, along with the reliability and accuracy of instruments such as the airspeed indicator, altimeter, and vertical speed indicator, play a critical role in overall flight safety [2], [3]. One key component is the pitot-static sensor, which measures dynamic and static pressures to generate essential flight data, including airspeed and altitude [4]. Given its critical role, the pitot-static sensor must be regularly tested and calibrated to ensure reliability in supporting pilot decision-making [1].

The Air Data Test Set (ADTS) is a dedicated testing instrument designed to evaluate and calibrate aircraft flight instruments by simulating air pressure under various altitude and airspeed conditions [5]. ADTS is commonly used to verify the performance of three primary instruments airspeed indicator, altimeter, and vertical speed indicator through precise control of static and dynamic pressures in a controlled environment [5].

However, conventional ADTS systems still rely on manual data acquisition and are limited to local environments. To address these limitations, this study proposes the development of an Internet of Things (IoT)-based system that enables real-time monitoring and control [6], utilizing the Node-RED platform, Mosquitto MQTT protocol, and a MySQL database [7],[8].

The proposed system is designed as an embedded system using the ESP32 microcontroller as the central controller, integrated with the MS5803 pressure sensor for wireless data acquisition and transmission [9],[10]. The measured pressures are calibrated to derive airspeed and altitude parameters that closely approximate commercial ADTS performance [11]. Node-RED provides a graphical flow-based programming interface for data processing and system control [12], while real-time communication is managed through the Mosquitto MQTT broker [13]. Experimental data are stored in a MySQL database to support historical analysis [14].

Through the integration of ESP32, MS5803, MQTT, Node-RED, and MySQL, this system is capable of simulating air pressure conditions similar to commercial ADTS, with a responsive real-time monitoring platform. Previous studies have emphasized the importance of IoT applications in avionics [15], yet few have integrated all components into a single functional and customizable ADTS platform. Therefore, this study contributes by developing an IoT-based ADTS monitoring dashboard, which offers advantages in real-time pressure and vacuum monitoring while supporting data integration for instrument performance analysis [16]. The objective of this research is to design and develop an ESP32- and MS5803-based ADTS for simulating air pressure in pitot-static instruments, as well as implementing an IoT platform using Node-RED, Mosquitto MQTT, and MySQL for efficient and integrated system monitoring and control.

DOI: 10.55003/ETH.420405 ISSN: 2985-1688 (Online)

2. Materials and Methods

The Air Data Test Set (ADTS) system presented in this study represents a technological advancement that integrates hardware components with Internet of Things (IoT) technology, as an evolution from previous hardware-based systems that relied solely on pneumatic mechanisms. The main advantage of this IoT-based system lies in its ability to monitor and display pressure readings (in mbar) and vacuum levels (in inHg), as well as to control parameters required to achieve target values on pitot-static indicators, including the Airspeed Indicator (ASI), Altimeter, and Vertical Speed Indicator (VSI). This aligns with Kabashkin et al [17], which explains that IoT sensors are strategically deployed throughout the aircraft to collect real-time data on various parameters, such as engine performance, structural integrity, and environmental conditions, thereby providing a comprehensive overview of the aircraft's operational status.

The primary focus of this system development is the implementation of airspeed indicator testing, enabling precise control of the input values for the ASI according to specific testing requirements. In addition, the system is equipped with a data logging feature that facilitates the analysis and evaluation of the airspeed indicator's sensor performance. With the capability for remote control and efficient data recording, the system plays a vital role in ensuring the optimal performance of aircraft instrumentation.

2.1 System Flow Design

The Internet of Things (IoT)-based Air Data Test Set (ADTS), as shown in **Figure 1**, begins with pressure and vacuum measurement using the MS5803 sensor. The acquired data are processed by the ESP32 microcontroller and transmitted via the Mosquitto Broker using the Message Queuing Telemetry Transport (MQTT) protocol, which is widely adopted in IoT applications due to its lightweight design [18]. The transmitted data are then distributed to Node-RED for real-time visualization through web-based gauges, charts, and graphs. In addition to visualization, Node-RED also controls actuators such as pumps and solenoid valves based on the received data and stores the results in a MySQL database [19]. This architecture ensures efficient integration between hardware and digital monitoring systems while supporting systematic data logging and further analysis.

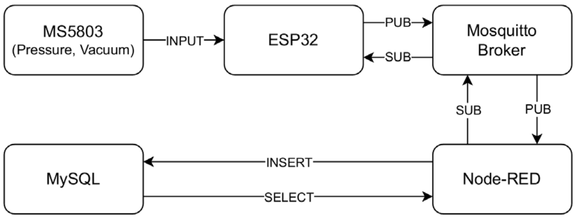


Figure 1 System Flow Diagram of the IoT-based ADTS

2.2 Hardware Components

Figure 2 presents the flow diagram of the Air Data Test Set (ADTS) system, which utilizes the ESP32 microcontroller as the central data processor. ESP32 is a highly integrated chip designed for IoT applications and capable of processing sensor data and controlling peripheral devices [20]. In this diagram, the MS5803 sensor measures air pressure, which is then processed by the ESP32 to control the pump and solenoid valve that regulate pressure and vacuum levels within the system.

The system is also equipped with a pressure gauge and vacuum gauge, which are used to monitor test results and ensure accurate pressure adjustments. The measured total pressure and static pressure generated by the system are subsequently directed to the airspeed indicator, altimeter, and vertical speed indicator to perform instrument calibration for aircraft.

This component diagram demonstrates how each part of the system works in an integrated manner to ensure efficiency and accuracy in the testing and calibration of aircraft flight instruments.

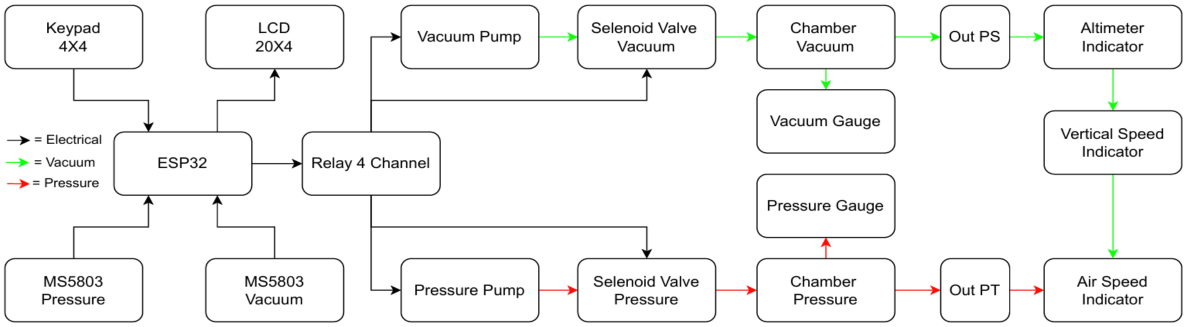


Figure 2 Flow Diagram of the Hardware Components in the ADTS

The Air Data Test Set (ADTS) shown in **Figure 3** consists of several integrated components designed to test the pitot-static indicators of an aircraft. The figure illustrates the integration of hardware and pneumatic systems, where sensors, pumps, and solenoid valves are combined to regulate pressure and vacuum according to the testing requirements.

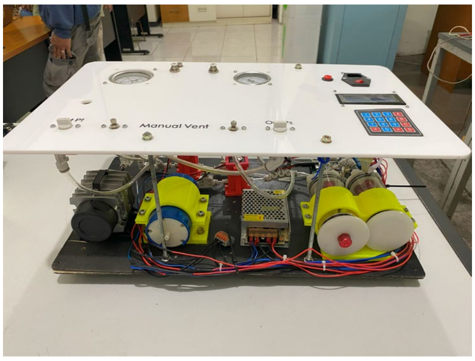


Figure 3 Integration of hardware and pneumatic components in the ADTS prototype

2.3 Data Acquisition and Storage Logic

In this system, the MQTT protocol is used as a real-time data communication method between the ESP32 sensor device and a processing interface such as Node-RED. MQTT supports asynchronous communication based on the publish/subscribe model, which is efficient and requires low bandwidth [21]. This protocol enables low-latency and high-efficiency data transmission, making it particularly well-suited for IoT systems that require continuous and integrated communication [18]. The transmitted parameters include:

1) Pressure data (esp32/pressure)
2) Vacuum data (esp32/vacuum)
3) Target airspeed in knots (esp32/set_knots)
4) Relay status (esp32/relay_status)

Node-RED is configured to subscribe to these topics. The systematic process from data reception to storage is outlined in the following pseudocode:

Algorithm 1 presents the pseudocode of the real-time data processing workflow, starting from sensor acquisition via MQTT to storage in the MySQL database. Each data packet is verified, and if valid, the pressure is converted into airspeed (knots) using the *convertPressureToKnots()* function, after which the system status is determined based on the relay condition. The labels “Start” and “End” mark the testing cycle to support instrument documentation and calibration. This approach ensures data integrity and enables systematic tracking of system performance. Previous studies have shown that MQTT is efficient and reliable for IoT-based real-time monitoring and is compatible with interfaces such as Node-RED [22].

Algorithm 1 The Pseudocode of MQTT Data Logger with relay status

```
1. Initialize:
  1.1 Set MQTT topics:
    - topic_pressure = "esp32/pressure"
    - topic_vacuum = "esp32/vacuum"
    - topic_set_knots = "esp32/set_knots"
    - topic_relaystatus = "esp32/relay_status"
  1.2 Initialize previous relay status:
    - prevRelayStatus = None
2. Loop (continuously):
```

Algorithm 1 The Pseudocode of MQTT Data Logger with relay status (cont.)

```
  2.1 Retrieve data from MQTT topics:
    - pressure, vacuum,set_knots, relay_status
  2.2 If any data is None:
    - Print error and skip loop
  2.3 Convert pressure to knots:
    - Knots = convertPressureToKnots(pressure)
  2.4 Determine targetKnots:
    - If setKnots is provided:
      - targetKnots = setKnots
    - Else:
      - targetKnots = Knots
  2.5 Check relay status change:
    - If relayStatus != prevRelayStatus:
      - If relayStatus == "OFF":
        - status = "End"
      - Else:
 - status = "Start"
  2.6 Insert data into database:
    - InsertDataToDatabase (pressure, vacuum, airspeed, altimeter, verticalspeed, status)
  2.7 Update previous relay status:
    - prevRelayStatus = relayStatus
```

2.4 Dashboard Development

The system utilizes a Node-RED dashboard as the main interface for real-time monitoring and control of the Air Data Test Set (ADTS). Pressure and vacuum sensor data are displayed through interactive gauges, along with visualizations of airspeed, altimeter, and vertical speed indicator. Control buttons are also provided to adjust airspeed and pressure. This approach offers an intuitive and responsive interface with immediate visual feedback, enhancing operational efficiency. As reported in previous studies, Node-RED serves as a flexible and reliable platform for IoT sensor data integration and real-time flow-based applications [19].

Figure 4 illustrates the data processing flow, starting from sensor data reception via MQTT from the ESP32, then processed using logic functions before being visualized on the dashboard. The main features in this flow diagram include:

1) Validation and manual input of airspeed (knots) via SET KNOTS and RESET buttons, which are published to the MQTT broker.

2) Relay ON/OFF control, which is linked to output devices for simulating loads or triggering external components.

3) Data storage into a MySQL database through the Insert Database node, limited to test states labeled as “start” and “end” to prevent redundant data logging.

4) Clear Table and Refresh Table functions for immediate data deletion and table reload from the dashboard interface.

This flow structure ensures that user commands are immediately translated into system actions, enabling real-time interaction and supporting efficient test execution

Figure 5 presents the Node-RED dashboard interface for monitoring and controlling the Air Data Test Set (ADTS). The dashboard employs digital analog-style gauges that replicate aircraft instruments, thereby enhancing data readability and accuracy for operators. The main features include:

1) Pressure & Vacuum Gauges: Displaying total pressure and vacuum in real time.

2) Airspeed Indicator (ASI), Altimeter, VSI: Simulating aircraft instruments with pointer needles and safety zones (green: normal, yellow: caution, red: critical).

3) Sensor Graph: Visualizing continuous fluctuations of pressure and vacuum over time.

4) Test Table: Presenting complete test data with timestamps and status, equipped with a CLEAR ALL DATA button.

5) Airspeed Control Panel: Allowing airspeed (knots) input with SET and RESET buttons, as well as input validity indicators.

The aircraft-style layout facilitates intuitive interaction for technicians and operators, minimizes reading errors, and accelerates decision-making during testing and calibration.

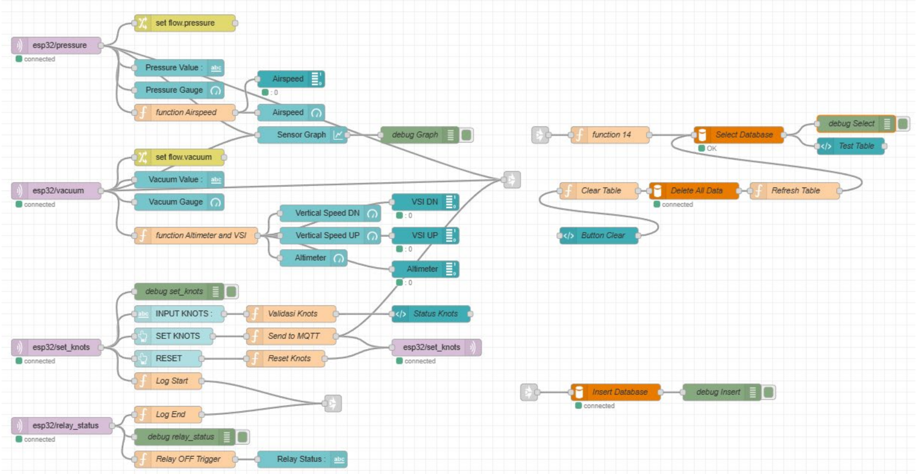
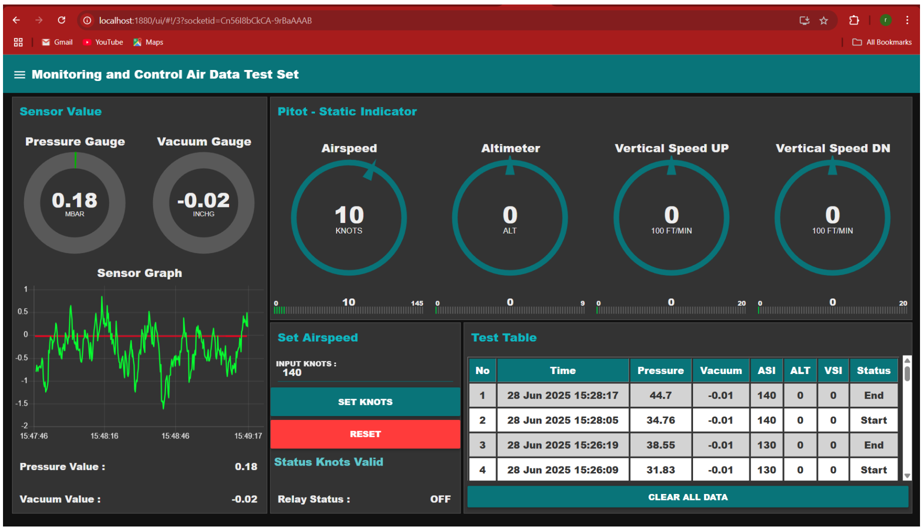


Figure 4 Data Processing Flow Diagram on the Node-RED Dashboard



Monitoring and Control Air Data Test Set

Sensor Value

Pressure Gauge 0.18

Vacuum Gauge -0.02

Sensor Graph

Pressure Value : 0.18

Vacuum Value : -0.02

Pitot - Static Indicator

Airspeed 10

Altimeter 0

Vertical Speed UP 0

Vertical Speed DN 0

Set Airspeed

INPUT KNOTS : 140

Status Knots Valid

Relay Status : OFF

Test Table

No	Time	Pressure	Vacuum	ASI	ALT	VSI	Status
1	28 Jun 2025 15:28:17	44.7	-0.01	140	0	0	End
2	28 Jun 2025 15:28:05	34.76	-0.01	140	0	0	Start
3	28 Jun 2025 15:26:19	38.55	-0.01	130	0	0	End
4	28 Jun 2025 15:26:09	31.83	-0.01	130	0	0	Start

Figure 5 Node-RED Dashboard Interface for ADTS Monitoring and Control

2.5 Algorithm of System Operation

The conversion of pressure into airspeed can essentially be explained by the pitot-static equation derived from Bernoulli's principle [23]. This relationship is given in Eq. (1):

$$V = \sqrt{\frac{2(P_t - P_s)}{\rho}} \qquad (1)$$

Where P_t is the total pressure (Pa), P_s is the static pressure (Pa), and ρ\rhop is the air density (kg/m^3). The air density is calculated from the local pressure and temperature using the ideal gas equation, as shown in Eq. (2):

$$\rho = \frac{P_s}{RT}, \; R = 287.05 \text{ J/(kg·K)} \qquad (2)$$

All pressures are expressed in Pascal (Pa), with the conversion $1 \text{ mbar} = 100 \text{ Pa}$, irspeed is reported in knots, with the conversion 1 knot = 0.514444 m/s. In general, the aeronautical literature adopts the standard ISA air density of $\rho = 1.225 \text{ kg/m}^3$ (sea level, 15°C), 1013.25 hPa) [24]. However, since the experiments were conducted in Bandung (±750 m above sea level), this study employed the local air density calculated from barometric pressure and temperature at the time of testing, namely $\rho \approx 1.14\ \text{kg/m}^3$ (Bandung, ±750 m a.s.l.). Accordingly, the theoretical Bernoulli values used in the accuracy analysis represent the actual conditions at the test site.

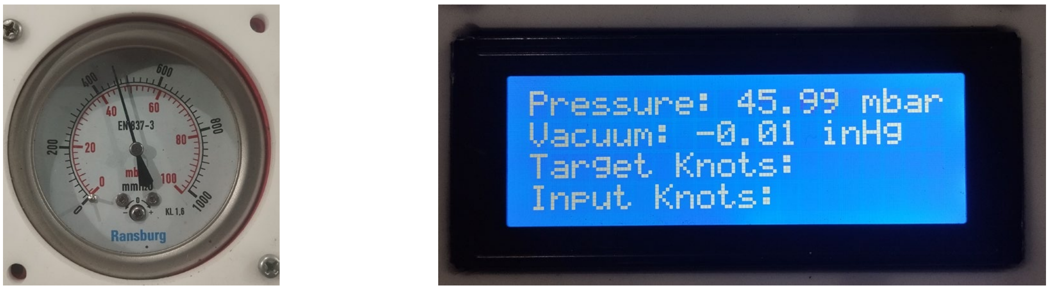
However, the developed system does not implement Bernoulli's equation directly. Instead, it utilizes an empirical calibration curve obtained from the reference instrument, with the conversion performed through linear interpolation between the calibration points, as expressed in Eq. (3):

$$V = V_1 + \frac{(q-q_1)}{(q_2-q_1)}(V_2 - V_1) \tag{3}$$

Where q is the measured pressure (mbar), q_1 and q_2 are the nearest calibration pressures, and V_1 and V_2 re the corresponding airspeeds (knots) at those calibration points. With this approach, the *convertPressureToKnots()* function consistently follows the empirical calibration curve, while the theoretical Bernoulli values are used only as a baseline for accuracy analysis.

2.6 Sensor Accuracy Evaluation

To ensure measurement accuracy, the readings of the MS5803 sensor were compared with a reference instrument, namely the Ransburg EM837-3 analog pressure gauge. The comparison is presented in **Figure 6**, where **Figure 6(a)** shows the analog gauge reading of approximately 45.00 mbar, while **Figure 6(b)** displays the sensor reading on the LCD of 45.99 mbar.



Pressure: 45.99 mbar
Vacuum: -0.01 inHg
Target Knots:
Input Knots:

(a) **(b)**

Figure 6 Comparison of pressure readings between (a) Ransburg EM837-3 analog pressure gauge and (b) MS5803 sensor on the LCD display.

Table 1 shows that the difference between the sensor and the reference instrument ranges from +0.13 to +0.99 mbar. This deviation is relatively small at lower pressures but increases at higher pressures. Nevertheless, the values are still acceptable for monitoring applications; however, further precision validation requires calibration against a traceable pressure standard.

Table 1 Comparison of Sensor and Gauge Measurement Results

Reference Instrument (mbar)	MS5803 Sensor (mbar)	Difference (mbar)
10.00	10.13	+0.13
20.00	20.24	+0.24
30.00	30.35	+0.35
40.00	40.23	+0.23
45.00	45.99	+0.99

2.7 Database Design and Logging

The system utilizes MySQL as the primary database to record testing results. MySQL is an open-source relational database management system that allows users to create, manage, and manipulate data efficiently [25]. The main table used is sensor_data, which stores key data such as pressure, vacuum, airspeed, altimeter, vertical speed, status, and timestamps. Data are recorded only at two critical moments in each testing cycle: when the test begins ("start") and when it ends ("end"). This approach is designed to minimize data redundancy and ensure efficient storage while supporting further data analysis

Figure 7 illustrates the structure of the sensor_data table within the MySQL database. This table records essential flight parameter data at the beginning and end of each ADTS test session, enabling efficient and structured logging. The columns in the table are described as follows:

1) (id (int (11)): Primary key, auto-increment for each entry.

2) pressure (float): Total (pitot) pressure measured by the sensor.

3) vacuum (float): Static pressure or vacuum for differential calculations.

4) airspeed (int (11)): Airspeed in knots, derived from pressure differential.

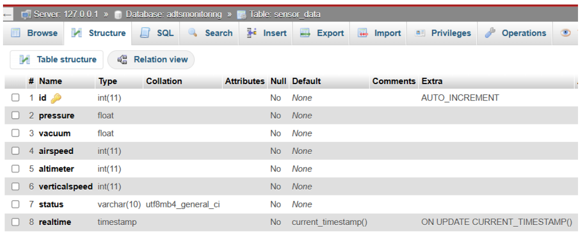
5) altimeter (int (11)): Estimated altitude in feet or meters from atmospheric pressure.

6) verticalspeed (int (11)): Rate of altitude change (vertical speed).

7) status (varchar (10)): Test phase indicator, e.g., "start" or "end."

8) realtime (timestamp): Automatic log of entry time, updated on modification.

This database structure supports a systematic calibration and monitoring process for air data measurement systems. By focusing data logging on the critical start and end points of each test, the system reduces storage overhead while maintaining accuracy and traceability of the recorded data



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	pressure	float			No	*None*		
3	vacuum	float			No	*None*		
4	airspeed	int(11)			No	*None*		
5	altimeter	int(11)			No	*None*		
6	verticalspeed	int(11)			No	*None*		
7	status	varchar(10)	utf8mb4_general_ci		No	*None*		
8	realtime	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Figure 7 MySQL database structure

The pseudocode in **Algorithm 2** is designed to receive sensor data via the MQTT protocol and process it in Node-RED. The process begins with the reception of validated airspeed values (knots), which are then stored with the status *start*. Subsequently, the pressure, vacuum, and airspeed data are used to control the pump or relay in order to reach the target pressure. Node-RED monitors the relay status until the condition *off* is reached, indicating that the target pressure has been achieved. At this stage, the final data, including pressure and vacuum values, are stored with the status *end*. The entire dataset, recorded only at the *start* and *end* points of the testing cycle, is then stored in the MySQL database. This approach ensures storage efficiency, data consistency, and supports more structured data processing within the IoT-based ADTS system.

Algorithm 2 Test data Processing and Storage in MySQL Database

```
1. Step 1: Receive Knots input
1.1 If InputKnots is valid:
    - status = "start"
    - knots = InputKnots
    - first_pressure = ReadPressureFromSensor()
    - first_vacuum = ReadVacuumFromSensor()
2. Step 2: Read pressure, vacuum, and control pump/relay
2.1 pressure = ReadPressureFromSensor()
2.2 vacuum = ReadVacuumFromSensor()
2.3 ControlPumpRelay(target_pressure)
3. Step 3: Check relay status
3.1 If Relay OFF:
    - last_pressure = ReadPressureFromSensor()
    - last_vacuum = ReadVacuumFromSensor()
    - status = "end"
4. Step 4: Send data to MySQL
4.1 SendDataToMySQL(start_data, end_data)
    - start_data = (airspeed, altimeter, verticalspeed,
    first_pressure, first_vacuum, status="start")
    - end_data = (last_pressure, last_vacuum,
    airspeed, altimeter, verticalspeed, status="end")
```

2.8 Implementation of the Air Data Test Set System on the Instrument Panel

In this implementation, the Air Data Test Set (ADTS) is integrated into an instrument panel, which serves as a learning platform for understanding the functions of flight instruments, including those related to the pitot-static indicators.

Figure 8 presents the flow diagram of the ADTS system on the instrument panel. The diagram illustrates the pressure flow from the ADTS to the airspeed indicator to simulate real flight conditions. In addition, the system is connected to the Node-RED platform via the MQTT protocol using the Mosquitto Broker, enabling real-time monitoring and control through a digital interface.

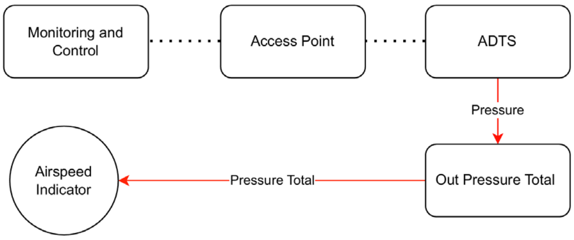


Figure 8 System Flow Diagram of ADTS on the Instrument Panel

The physical implementation is shown in **Figure 9**. In this configuration, the ADTS directly supplies the total pressure to the airspeed indicator to replicate the operational characteristics of the pitot-static instrument. The system is fully integrated with Node-RED via the MQTT protocol, with a laptop serving as the main interface for monitoring and control. This implementation demonstrates the integration of hardware with a real-time IoT-based monitoring system, while also providing an effective training environment for simulating actual flight conditions.

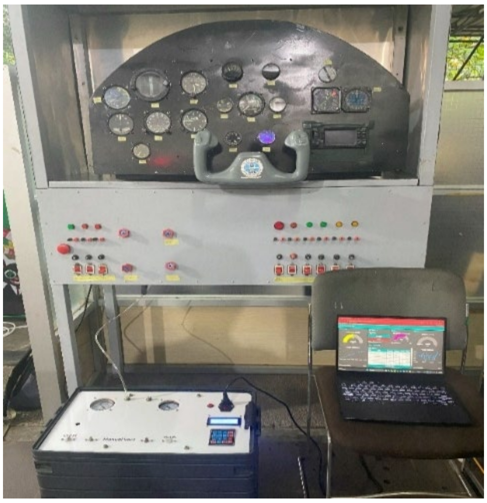


Figure 9 Air Data Test Set Implementation with Instrument Panel

3. Results and Discussion

3.1 Sensor Stability and Accuracy

The stability and accuracy evaluation of the sensors was carried out to ensure the reliability of pressure and vacuum measurements in the Air Data Test Set (ADTS). This analysis was based on real-time sensor data collected during testing.

Figure 10 shows the sensor readings of pressure and vacuum. The green curve represents pressure in millibars (mbar), while the red curve indicates vacuum in inches of mercury (inHg). The pressure sensor exhibited small fluctuations at the beginning of the test, ranging from −0.05 to 0.05 mbar, which indicates stable performance with a low noise level. In contrast, the vacuum sensor showed a relatively flat pattern. This is due to the use of the inHg unit, which has a larger scale compared to mbar (1 inHg ≈ 33.8639 mbar), making small variations in pressure less noticeable in the graph [26].

Overall, both sensors demonstrated sufficient stability and accuracy to support the simulation of pressure and vacuum measurements in the ADTS. This stability ensures that the system is capable of representing flight conditions with a good level of reliability.

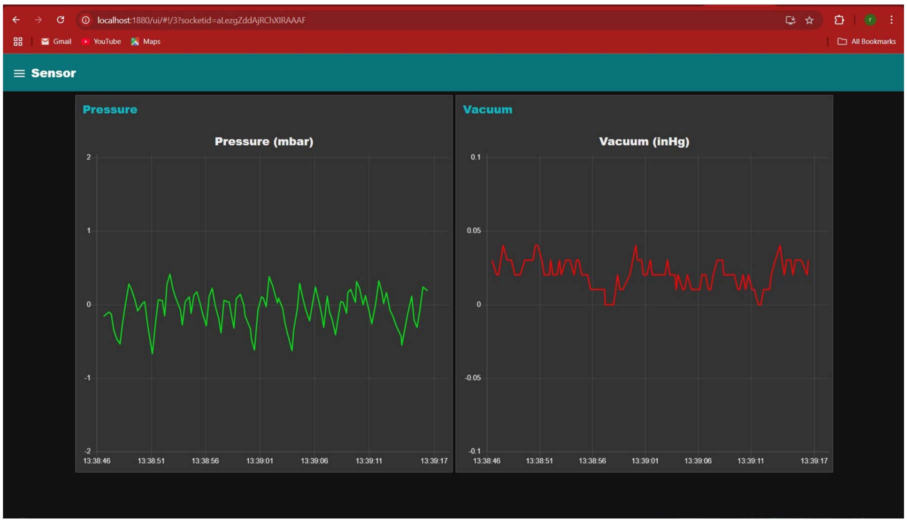
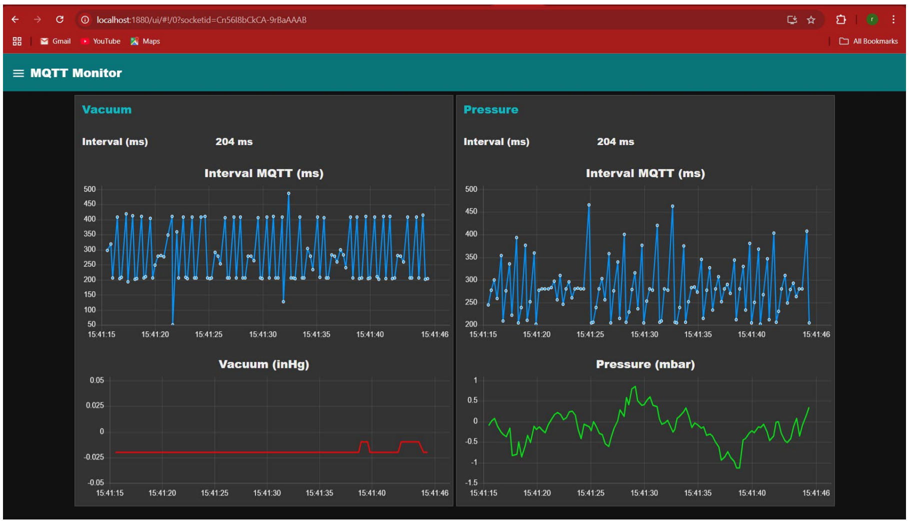

localhost:1880/ui/#!/3?socketid=aLezgZddAjRChXIRAAAF

Figure 10 Graph of Initial Pressure and Vacuum Sensor Values

3.2 Data Transmission and Dashboard Performance

In this section, a performance test of data transmission using the MQTT protocol was conducted to evaluate the stability and latency of data delivery from the ESP32 device to the Node-RED dashboard. The testing focused on two main aspects: the conformity of the transmission interval with the sampling configuration (100 ms) and the variation of end-to-end latency experienced during the data transmission process.

Figure 11 shows the variation of the actual latency, which ranged from 200 ms to 500 ms, higher than the configured transmission interval of 100 ms. This indicates the presence of jitter, mainly influenced by wireless network conditions such as signal interference and traffic congestion. These results are consistent with the findings of [22], which reported that although MQTT is designed for low-latency communication, unstable wireless network performance can cause significant variations in transmission time.


localhost:1880/ui/#!/0?socketid=Cn56I8bCkCA-9rBaAAAB

Interval (ms) 204 ms

Interval (ms) 204 ms

Figure 11 Graph of Data Transmission Intervals to MQTT

The end-to-end latency in this system can be broken down into three main components:

1) Device (ESP32): Sensor acquisition and loop processing, including a 100 ms sampling delay.

2) MQTT broker: Message reception and routing process, relatively small (≈ 1–5 ms).

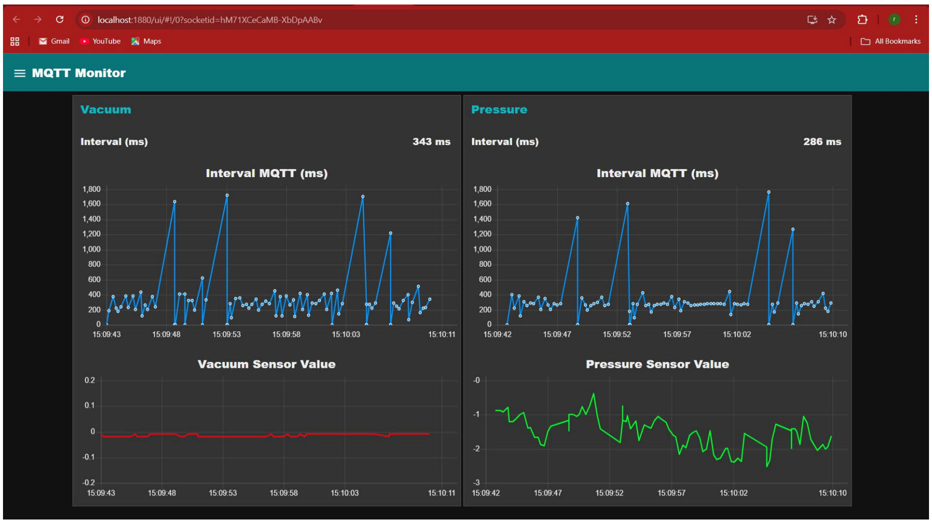
3) Dashboard (Node-RED + browser): Data processing and graphical rendering pipeline, which adds variations of up to several tens of milliseconds.

Although the recorded latency is higher than the configured transmission interval, the average value of approximately 200 ms is still adequate for real-time monitoring. The variations are mainly caused by Wi-Fi communication jitter and dashboard rendering overhead; however, data updates remain responsive, ensuring that both analog and digital indicators appear visually synchronized [16].

Thus, the latency variation of up to 500 ms is mainly attributed to Wi-Fi network jitter and dashboard visualization overhead, while the average latency of around 200 ms remains adequate for real-time monitoring applications

3.3 System Performance Evaluation Under Various Conditions

To ensure reliability, the system was tested under various operational conditions that represent real-world scenarios. The first test was conducted to evaluate latency at a distance of 20 m, with the device placed outdoors and the access point located indoors. The results showed that the system was able to transmit pressure and vacuum sensor readings via MQTT with an average latency of 200–300 ms, despite the presence of wall obstructions. However, occasional latency spikes of up to 1.7 seconds occurred, as shown in **Figure 12**, caused by signal blockage that delayed the publication of sensor data to the web broker.



Interval (ms) 343 ms

Interval (ms) 286 ms

Figure 12 Graph of MQTT Communication Interval and Vacuum and Pressure Sensor Values

Based on **Table 2**, the system performance remained stable and did not show significant differences compared to the test with unobstructed signals, as shown in **Figure 11**, which had an average latency of 200–500 ms without notable spikes. Although an outdoor test recorded latency spikes of up to 1.7 seconds, this did not reduce the overall system performance but only caused temporary delays in real-time data updates on the browser.

In addition, a field test was conducted at different heights under the assumption that the static pressure (Ps) was equal to the atmospheric pressure. The access point was placed on the first floor of the Faculty of Applied Science Building, Telkom University, Bandung (±768 m above sea level), with testing carried out on the first and fourth floors to evaluate the effect of altitude on system performance. The results showed that the difference in height did not have a significant impact on either target achievement or data accuracy. The latency observed during testing also did not affect system performance, as shown in **Table 3.** Although the initial sensor readings indicated variations due to atmospheric differences between the first and fourth floors, these did not affect the measurement outcomes.

Overall, the test results confirm that the system is capable of operating consistently in achieving measurement targets under various environmental conditions, including differences in distance, physical obstructions, and altitude variations

Table 2 Pressure Sensor Performance Against Target in Stable and Unstable Networks

Knots	Target (mbar)	Measured Sensor Value	
		Indoor	Outdoor
20	1	1.14	0.83
30	2	2.16	1.64
40	3	2.94	2.76
50	5	5.05	5.1
60	7	6.64	7.39
70	10	10.18	10.06
80	15	14.85	14.78
90	20	19.91	19.91
100	25	25.22	25.24
110	29	28.84	29.12
120	34	33.72	35.72
130	39	39.28	41.28
140	45	44.86	47.86

Table 3 Comparison of Pressure Readings at Different Altitudes Based on Atmospheric Pressure

Knots	Target (mbar)	First Floor		Fourth Floor	
		Initial Reading (mbar)	Measured Value (mbar)	Initial Reading (mbar)	Measured Value (mbar)
20	1	-0.63	1.14	-2.79	0.82
30	2	-0.52	2.16	-2.07	2.38
40	3	-0.32	2.94	-1.94	3
50	5	-0.66	5.05	-2.08	4.75
60	7	-0.90	6.64	-2.35	7.17
70	10	-0.19	10.18	-2.13	10.45
80	15	-0.03	14.85	-2.18	15.39
90	20	-0.57	19.91	-2.52	19.5
100	25	-0.69	25.22	-2.07	24.84
110	29	-0.32	28.84	-2.28	29.4
120	34	-0.40	33.95	-1.99	34.04
130	39	-0.09	39.23	-2.98	38.85
140	45	-0.12	45.04	-2.26	45.03

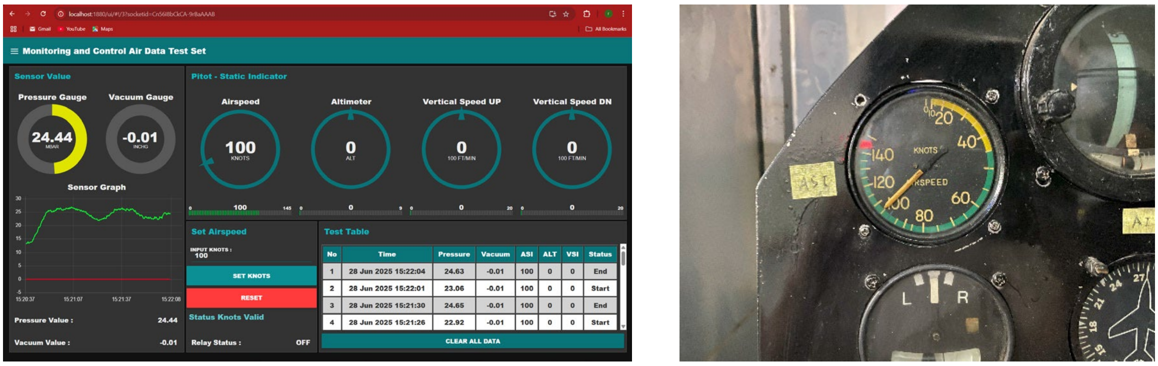
3.4 Input Knots and Target Pressure Evaluation

The Air Data Test Set (ADTS) system testing was focused on the airspeed indicator (ASI), since the vacuum system was not functioning optimally and the altimeter and vertical speed indicator components were not yet available in the hardware configuration. Consequently, the evaluation was limited to the dynamic pressure generated by the pump, which was converted into airspeed (knot) through an empirical calibration method.

This method was selected because the system lacked a stable static pressure regulation, making theoretical approaches less reliable. Empirical calibration was considered more suitable under these conditions, as it provides a simpler implementation while still being able to adjust pressure values to match the target airspeed [27].

The objective of this test was to evaluate the consistency and accuracy of the pressure generated in relation to the target airspeed (knot) on the ASI. The input airspeed values were entered through the monitoring dashboard, which simultaneously displayed the digital airspeed indicator to monitor changes in real time. This feature enabled users to verify whether the system successfully reached the target value once the corresponding pressure was achieved.

Figure 13 shows the system conditions during testing, including the visual representation of the airspeed indicator. During the test, the user entered a value of 100 knots via the dashboard, which was then converted into the target pressure (mbar) using the empirical equation. The system controlled the pump and valves to achieve this target, while the actual pressure was recorded by the sensor and displayed in real time on both the dashboard and the airspeed indicator.



Pressure Value : 24.44

Vacuum Value : -0.01

INPUT KNOTS : 100

Relay Status : OFF

No	Time	Pressure	Vacuum	ASI	ALT	VSI	Status
1	28 Jun 2025 15:22:04	24.63	-0.01	100	0	0	End
2	28 Jun 2025 15:22:01	23.06	-0.01	100	0	0	Start
3	28 Jun 2025 15:21:30	24.65	-0.01	100	0	0	End
4	28 Jun 2025 15:21:26	22.92	-0.01	100	0	0	Start

(a) **(b)**

Figure 13 Testing at 100 Knots (a) Dashboard Display (b) Output on Airspeed Indicator

3.5 Accuracy Analysis

This subsection discusses the system's accuracy in converting airspeed (knot) into dynamic pressure (mbar) based on experimental test results. The evaluation was carried out by comparing the target pressure from calibration with the actual pressure achieved by the system in five test runs, as summarized in **Table 4** and **Table 5**. Empirical calibration was performed because the static pressure (Ps) was assumed to be equal to the ambient atmospheric pressure; therefore, the testing focused on the airspeed indicator (ASI). It should be noted that Ps follows the local atmospheric pressure; if the device is relocated to a different site, Ps will adjust to the atmospheric conditions at that location, while the calculations remain based on total pressure (Pt).

Table 4 Comparison of Pressure Values Between Test Results and Targets at Airspeed

Knots	Target (mbar)	Pressure Reached (mbar)				
		Testing 1	Testing 2	Testing 3	Testing 4	Testing 5
20	1	1.14	1.05	1.08	1.04	1.10
30	2	2.16	1.96	2.17	2.05	2.09
40	3	2.94	3.10	2.90	3.03	2.58
50	5	5.05	5.12	5.09	5.11	4.51
60	7	6.64	7.07	6.90	6.56	6.53
70	10	10.18	10.15	10.05	10.11	9.82
80	15	14.85	15.25	14.93	14.82	14.62
90	20	19.91	19.96	20.08	20.10	20.18
100	25	25.22	25.10	25.12	24.65	24.63
110	29	28.84	28.97	29.04	29.10	28.64
120	34	33.95	34.21	33.98	34.27	33.79
130	39	39.23	39.13	39.14	39.09	38.55
140	45	45.04	45.20	44.70	44.62	44.70

Table 5 Pressure Accuracy Evaluation Based on Airspeed Indicator Test Results

Knots	Target (mbar)	Value Error									
		Testing 1		Testing 2		Testing 3		Testing 4		Testing 5	
		mbar	%	mbar	%	mbar	%	mbar	%	mbar	%
20	1	0.14	14	0.05	5	0.08	8	0.04	4	0.1	10
30	2	0.16	8	-0.04	-2	0.17	8.5	0.05	2.5	0.09	4.5
40	3	-0.06	-2	0.1	3.33	-0.1	-3.33	0.03	1	-0.42	-14
50	5	0.05	1	0.12	2.4	0.09	1.8	0.11	2.2	-0.49	-9.8
60	7	-0.36	-5.14	0.07	1	-0.1	-1.43	-0.44	-6.29	-0.47	-6.71
70	10	0.18	1.8	0.15	1.5	0.05	0.5	0.11	1.1	-0.18	-1.8
80	15	-0.15	-1	0.25	1.67	-0.07	-0.47	-0.18	-1.2	-0.38	-2.53
90	20	-0.09	-0.45	-0.04	-0.2	0.08	0.4	0.1	0.5	0.18	0.9
100	25	0.22	0.88	0.1	0.4	0.12	0.48	-0.35	-1.4	-0.37	-1.48
110	29	-0.16	-0.55	-0.03	-0.1	0.04	0.14	0.1	0.34	-0.36	-1.24
120	34	-0.05	-0.15	0.21	0.62	-0.02	-0.06	0.27	0.79	-0.21	-0.62
130	39	0.23	0.59	0.13	0.33	0.14	0.36	0.09	0.23	-0.45	-1.15
140	45	0.04	0.09	0.2	0.44	-0.3	-0.67	-0.38	-0.84	-0.3	-0.67

This subsection discusses the system's accuracy in converting airspeed (knot) into dynamic pressure (mbar) based on experimental test results. The evaluation was carried out by comparing the target pressure from calibration with the actual pressure achieved by the system in five test runs, as summarized in **Table 4** and **Table 5**. Empirical calibration was performed because the static pressure (Ps) was assumed to be equal to the ambient atmospheric pressure; therefore, the testing focused on the airspeed indicator (ASI). It should be noted that Ps follows the local atmospheric pressure; if the device is relocated to a different site, Ps will adjust to the atmospheric conditions at that location, while the calculations remain based on total pressure (Pt).

The results in **Table 4** show that the actual pressure generally approximates the target values, although deviations occur at low speeds. For example, at 40 knots, the recorded pressure was 2.58 mbar compared to the target of 3.00 mbar. **Table 5** presents the absolute and percentage error values, where the largest error occurred at low speed (around −0.42 mbar or −14%). This is attributed to the system's sensitivity to small pressure changes, making fluctuations more significant in the lower range

In contrast, at medium to high airspeeds (70 to 140 knots), the error values were relatively small and stable. This indicates that the system is more accurate and consistent in converting airspeed to pressure within this range. For instance, at 100 knots, the maximum error was within ±1.48%, demonstrating a very small deviation from the target value. The percentage error is calculated using Eq. (4) as follows:

$$\text{Error (\%)} = \left(\frac{\text{Error (mbar)}}{\text{Target (mbar)}}\right) \times 100\% \qquad (4)$$

Overall, the conversion error ranges from 0.09% to 14% **Table 5**. The highest value was recorded at 40 knots, amounting to −0.42 mbar (−14%), due to the system's sensitivity to low pressure. However, above 70 knots, the error decreases significantly, averaging only ±0.1% to ±1%, with a maximum deviation of ±0.67% at 140 knots.

Figure 14 illustrates the trend of decreasing error with increasing speed. The relatively large fluctuations in the 20–60 knot range gradually decline and stabilize after 70 knots, indicating that the system operates more optimally at medium to high speeds. In general, the measurement error rate falls within the range of 1–5% across most of the speed spectrum (60–140 knots). The larger deviations at lower speeds are mainly caused by the limited accuracy of the MS5803 sensor in matching the target values, as well as the stability of the pressure generated by the pump. These findings demonstrate that the system provides good accuracy in converting airspeed into pressure at medium to high speeds, although improvements are still required in the low-speed range.

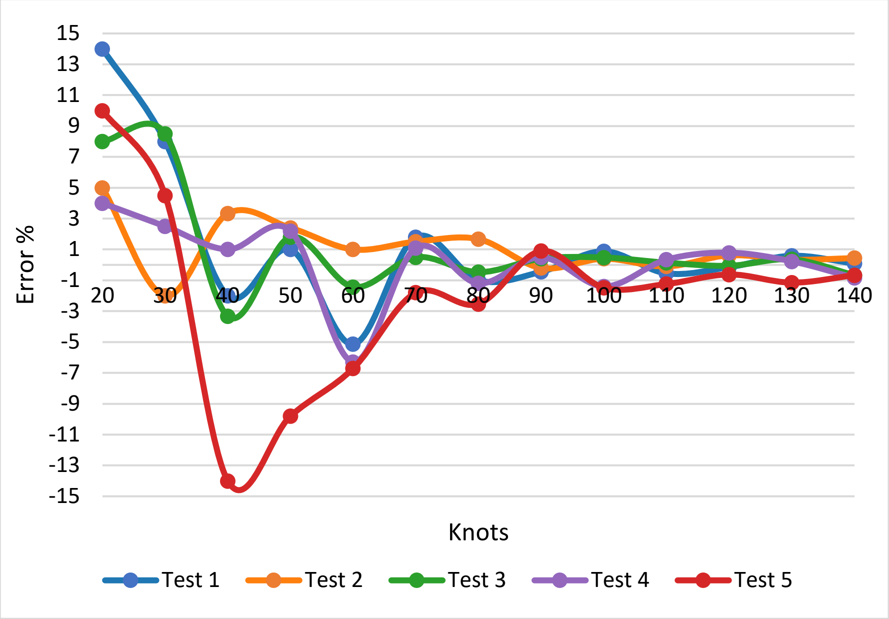


Figure 14 Error Percentage Graph on Achieved Targets

Table 6 presents a comparison of the target values (calibration curve), the measured values (average of five tests), and the theoretical Bernoulli values. The theoretical values were calculated using Bernoulli's Eq. (1) with the local air density in Bandung $\rho \approx 1.14\ \mathrm{kg/m^3}$. The results indicate that the measured values are generally very close to the target values, with small deviations ranging from 0.02 to 0.26 mbar, whereas the differences from the theoretical Bernoulli values are much larger, ranging from 0.48 to 14.91 mbar. These discrepancies are attributed to the assumption of Ps = atmospheric pressure, the limitations of the pressure sensor, and the pneumatic non-linearity effects in the system. Therefore, the evaluation of system accuracy is primarily based on its proximity to the calibration curve, while the theoretical Bernoulli values are used only as a baseline for analyzing systematic deviations.

Table 6 Measured vs. Target vs. Theoretical Values at Each Test Point

Knots	Target (mbar)	Measured (mbar)	Theoretical (mbar)	Measured Deviation vs Target (mbar)	Measured Deviation vs Theoretical (mbar)
20	1.00	1.08	0.60	+0.08	+0.48
30	2.00	2.09	1.35	+0.09	+0.74
40	3.00	2.91	2.41	−0.09	+0.50
50	5.00	4.98	3.76	−0.02	+1.22
60	7.00	6.74	5.42	−0.26	+1.32
70	10.00	10.06	7.40	+0.06	+2.66
80	15.00	14.89	9.68	−0.11	+5.21
90	20.00	20.05	12.28	+0.05	+7.77
100	25.00	24.94	15.19	−0.06	+9.75
110	29.00	28.92	18.41	−0.08	+10.51
120	34.00	34.04	21.94	+0.04	+12.10
130	39.00	39.03	25.78	+0.03	+13.25
140	45.00	44.85	29.94	−0.15	+14.91

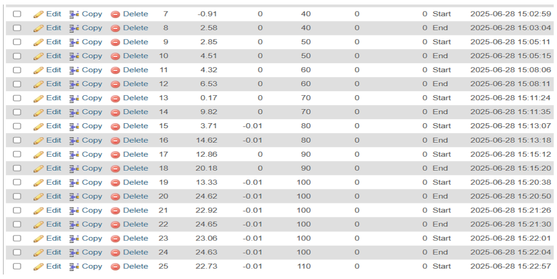
3.6 Data Logging Efficiency

The database structure was designed to be simple but efficient by using a single main table named sensor_data. This table stores measurement results in the form of pressure, vacuum, airspeed, altimeter, vertical speed, operational status (start/end), and a timestamp (realtime) that is automatically updated each time data is recorded. This design allows systematic data logging while also supporting the performance analysis of ADTS under various test conditions. **Table 7** shows the Entity-Relationship (ER) diagram of the sensor_data table, which displays the columns id, pressure, vacuum, airspeed, altimeter, vertical speed, status, and realtime. With this structure, the system can handle real-time data storage to improve monitoring capabilities during testing.

Table 7 Entity–Relationship (ER) Diagram of the sensor_data Table

sensor_data		
PK	id pressure vacuum airspeed altimeter verticalspeed status realtime	int double double int int int varchar timestamp

Figure 15 displays the data stored in the sensor_data table. This view shows both the structure and the content of the table, which records test results at the initial (start) and final (end) conditions. Each entry captures sensor readings, including pressure, vacuum, airspeed, altimeter, and vertical speed.



□	Edit	Copy	Delete	7	-0.91	0	40	0	0	Start	2025-06-28 15:02:59
□	Edit	Copy	Delete	8	2.58	0	40	0	0	End	2025-06-28 15:03:04
□	Edit	Copy	Delete	9	2.85	0	50	0	0	Start	2025-06-28 15:05:11
□	Edit	Copy	Delete	10	4.51	0	50	0	0	End	2025-06-28 15:05:15
□	Edit	Copy	Delete	11	4.32	0	60	0	0	Start	2025-06-28 15:08:06
□	Edit	Copy	Delete	12	6.53	0	60	0	0	End	2025-06-28 15:08:11
□	Edit	Copy	Delete	13	0.17	0	70	0	0	Start	2025-06-28 15:11:24
□	Edit	Copy	Delete	14	9.82	0	70	0	0	End	2025-06-28 15:11:35
□	Edit	Copy	Delete	15	3.71	-0.01	80	0	0	Start	2025-06-28 15:13:07
□	Edit	Copy	Delete	16	14.62	-0.01	80	0	0	End	2025-06-28 15:13:18
□	Edit	Copy	Delete	17	12.86	0	90	0	0	Start	2025-06-28 15:15:12
□	Edit	Copy	Delete	18	20.18	0	90	0	0	End	2025-06-28 15:15:20
□	Edit	Copy	Delete	19	13.33	-0.01	100	0	0	Start	2025-06-28 15:20:38
□	Edit	Copy	Delete	20	24.62	-0.01	100	0	0	End	2025-06-28 15:20:50
□	Edit	Copy	Delete	21	22.92	-0.01	100	0	0	Start	2025-06-28 15:21:26
□	Edit	Copy	Delete	22	24.65	-0.01	100	0	0	End	2025-06-28 15:21:30
□	Edit	Copy	Delete	23	23.06	-0.01	100	0	0	Start	2025-06-28 15:22:01
□	Edit	Copy	Delete	24	24.63	-0.01	100	0	0	End	2025-06-28 15:22:04
□	Edit	Copy	Delete	25	22.73	-0.01	110	0	0	Start	2025-06-28 15:22:57

Figure 15 The display of data storage based on the sensor_data table

In this view, the values for the altimeter and vertical speed are still unavailable, indicated as Not Available (NA) or zero (0). This is because the test results presented in **Figure 15** focus only on the airspeed indicator, without involving measurements of the altimeter or vertical speed.

3.7 Discussion

The developed ADTS demonstrates significant advantages over manual systems, particularly through real-time monitoring on the Node-RED dashboard **Figure 13**. The dashboard displays sensor data, including pressure, vacuum, and airspeed (in knots) converted from pressure. These values are directly reflected on both the analog Airspeed Indicator (ASI) and the digital gauge, designed to resemble the original instrument. The visual consistency between the two indicators demonstrates the success of empirical calibration in aligning output pressure with the target airspeed, while also enhancing monitoring flexibility through the digital interface.

The test results indicate that the ADTS successfully converts pressure into airspeed and can be remotely controlled, enhancing operational flexibility and enabling real-time status updates. Latency fluctuations were recorded between 200–500 ms, exceeding the 100 ms configuration, primarily due to Wi-Fi communication jitter and dashboard rendering overhead. Despite these latency variations, the system is still capable of displaying pressure and airspeed data synchronously on the dashboard, supporting real-time interaction between the hardware and the user interface [28].

Although the system performs well, it has limitations because it relies solely on pressure data without considering vacuum separately, which can affect accuracy. Static pressure (Ps) is not directly measured but assumed equal to ambient ground pressure. This assumption prevents the system from fully representing the differential calculation (Pt–Ps) that forms the basis of pitot-static instruments. As a result, larger deviations occur at low speeds, while errors at medium to high speeds remain below 1%

Furthermore, the pressure sensors used in this study have not been calibrated against a traceable pressure standard, making the measurements relative and influenced by environmental conditions, including air pressure variations at the test site. Calibration was performed empirically by comparing device outputs with a standard airspeed indicator, representing a key limitation to be addressed in future development

Measured values are closer to the empirical calibration curve than to theoretical Bernoulli values, particularly at high speeds. This is due to the use of empirical calibration with linear interpolation, the assumption Ps = atmospheric pressure, sensor resolution limitations, and pneumatic effects such as dead volume and valve characteristics. Therefore, system accuracy is more appropriately evaluated against the calibration curve, with Bernoulli values used only as a baseline for analysis

Testing in this study was limited to static ground conditions. Rapid air pressure changes experienced in actual flight, such as during climb, descent, or turbulence, were not replicated. Consequently, system validation does not fully reflect operational field scenarios, forming the basis for further development as described in the Future Work section.

Accuracy analysis in **Figure 14** shows that the ADTS system exhibits errors ranging from 1–5% across most speed ranges, with the largest deviation reaching 14% at low speeds **Table 3**. These errors are related to sensor limitations and pump pressure stability. In real flight conditions, such deviations could have serious safety implications, as inaccurate airspeed readings may affect pilot decision-making, particularly during critical phases such as takeoff and landing.

In addition to the main tests, supplementary evaluations were conducted to assess system reliability under various environmental conditions. Results confirmed that the system operates stably across different environments. Average communication latency remained low during both indoor and outdoor tests, although occasional spikes up to 1.7 seconds occurred due to signal obstacles. These spikes did not degrade system performance but caused temporary delays in dashboard updates. From an accuracy perspective, pressure sensor readings remained close to calibration targets even under unstable network conditions. Furthermore, height variation tests indicated that atmospheric pressure differences between floors only affected initial readings, without significantly impacting final measurement results. Thus, the system proved reliable in maintaining measurement accuracy and consistency across various operational scenarios.

4. Conclusion

This study successfully developed an IoT-based Air Data Test Set (ADTS) that supports automatic and real-time calibration of airspeed indicators. The system integrates the ESP32 microcontroller, MS5803 pressure sensor, MQTT protocol, and Node-RED dashboard, enabling dynamic pressure control based on airspeed input and providing digital visualization that mimics aircraft instruments.

Experimental results showed that the system was able to achieve the target pressure with an error range of 0.09% to 14%, with high accuracy at 70–140 knots where the error remained below 1%. The recorded MQTT transmission latency ranged between 200–500 ms and occasionally reached 1.7 seconds; however, this did not affect the overall system performance, only causing minor delays in dashboard updates. Similarly, variations in atmospheric pressure due to differences in test altitude only influenced the initial sensor readings but did not affect the accuracy of the final results.

The conversion of pressure to airspeed was not carried out using the theoretical Bernoulli equation but instead through empirical calibration with linear interpolation between calibration points, ensuring that the output values remained consistent with the reference instrument. With its fully integrated architecture covering automatic pressure control, digital input handling, real-time visualization, and selective data logging based on test status (“start” and “end”) the system proved to be stable, reliable, and efficient, making it a viable alternative solution for pitot-static calibration in educational and laboratory environments.

4.1 Future Work

Future development will focus on improving the accuracy and representativeness of the system. First, the system will be expanded through the integration of static pressure sensors and precision differential pressure sensors, enabling measurements to cover all aspects of the pitot-static system and extending functionality to the altimeter and vertical speed indicator.

Second, calibration will no longer be purely empirical but will use static pressure (Ps) from a traceable Air Data Test Set (ADTS) standard. Additionally, mathematical approaches based on Bernoulli’s equation and curve fitting methods will be explored to enhance the accuracy of pressure-to-speed conversion. Performance comparisons between the embedded-system-based ADTS and standard ADTS will also be conducted.

Third, to minimize measurement errors, future research will implement periodic sensor calibration using standard reference instruments, utilize high-precision sensors, and apply signal processing algorithms such as moving average and Kalman filter. Closed-loop control mechanisms for pumps and valves will also be developed to maintain target pressure stability.

5. References

[1] A. Golparvar and M. K. Yapici, “Analysis of Pitot tube Airflow Velocity Sensor Behavior in Blockage Situations,” in *2020 IEEE SENSORS*, Rotterdam, Netherlands, Oct. 25–28, 2020, pp. 1–3, doi: 10.1109/SENSORS47125.2020.9278790.

[2] H. Jin, Z. Hu, K. Li, M. Chu, G. Zou and G. Yu, “Study on How Expert and Novice Pilots Can Distribute Their Visual Attention to Improve Flight Performance,” *IEEE Access*, vol. 9, pp. 44757–44769, 2021, doi: 10.1109/ACCESS.2021.3066880.

[3] L. Stuber, S. L. Jeger, R. Zufferey and D. Floreano, “Miniature Multihole Airflow Sensor for Lightweight Aircraft Over Wide Speed and Angular Range,” *IEEE Robotics and Automation Letters*, vol. 10, no. 10, pp. 10722–10728, 2025, doi: 10.1109/LRA.2025.3604704.

[4] H. Tabanlı and K. B. Yüceil, “Wind Tunnel Tests for a Pitot-Static Probe Designed To Measure Aircraft Speed and Altitude at Subsonic Compressible and Transonic Regimes,” *Journal of Aeronautics Space Technologies*, vol. 14, no. 2, pp. 145–153, 2021.

[5] J. Shi, Y. Cao, L. Chen and Y. Yang, “Remote-controlled air data test system in Iron-Bird,” in *CSAA/IET International Conference on Aircraft Utility Systems (AUS 2022)*, Nanchang, China, Aug. 17–20, 2022, pp. 575–579, doi: 10.1049/icp.2022.1593.

[6] A. Medina-Pérez, D. Sánchez-Rodríguez and I. Alonso-González, “An Internet of Thing Architecture Based on Message Queuing Telemetry Transport Protocol and Node-RED: A Case Study for Monitoring Radon gas,” *Smart Cities*, vol. 4, no. 2, pp. 803–818, 2021, doi: 10.3390/smartcities4020041.

[7] S. Chanthakit and C. Rattanapoka, "MQTT Based Air Quality Monitoring System using Node MCU and Node-RED," in *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, Nakhonpathom, Thailand, Jul. 11–13, 2018, pp. 1–5, doi: 10.1109/ICT-ISPC.2018.8523891.

[8] K. Anam, D. N. Rofi and R. Meiyanti, "Monitoring System for Temperature and Humidity Sensors in the Production Room Using Node-Red as the Backend and Grafana as the Frontend," *Journal of Systems Engineering and Information Technology (JOSEIT)*, vol. 2, no. 2, pp. 59–76, 2023, doi: 10.29207/joseit.v2i2.5222.

[9] T. White, A. Fonnegra, and O. Hazbon, "Development of a low cost and low weight small UAS flight test instrumentation system," in *2024 Regional Student Conferences, Morgantown*, WV, USA, Apr. 12–13, 2024, Art. no. AIAA 2024-85725, doi: 10.2514/6.2024-85725.

[10] T. A. Vergote, "A Wireless Solar-Powered Pore Pressure Sensor for Monitoring of Hydraulically Dredged Sediments Used in Land Reclamations," in *7th International Symposium on Sediment Management*, Lille, France, Jul. 7–10, 2021, pp. 1–6.

[11] D. Dorsel, S. Staacks, H. Heinke and C. Stampfer, "Using a Smartphone Pressure Sensor as Pitot Tube Speedometer," *The Physics Teacher*, vol. 60, no. 4, pp. 273–275, 2022, doi: 10.1119/5.0025899

[12] K. Dwarakesh, R. Rathika and A. S. Suriya, "210 Mw Turbo Generator's Hydrogen Gas Cooling System Online Monitoring And Controlling Using Node Red Flow Based IoT," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6s2, pp. 82–87, 2019, doi: 10.35940/ijeat.F1018.0886S219.

[13] W. He, M. J. A. Baig and M. T. Iqbal, "An Open-Source Supervisory Control and Data Acquisition Architecture for Photovoltaic System Monitoring Using ESP32, Banana Pi M4, and Node-RED," *Energies*, vol. 17, no. 10 2024, Art. no. 2295, doi: 10.3390/en17102295.

[14] S. Rautmare and D. M. Bhalerao, "SQL & NoSQL Database Study for Internet of Things," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 5, pp. 9069–9073, May 2016.

[15] S. Vishnu Karthik, V. Akshaya and P. Sriramalakshmi, "IoT based Predictive Maintenance of Electrical Machines in Aircraft," in *2021 7th International Conference on Electrical Energy Systems (ICEES)*, Chennai, India, Feb. 11–13, 2021, pp. 569–575. doi: 10.1109/ICEES51510.2021.9383669.

[16] A. Angadi, R. Dias and M. Umair Bagali, "An Aircraft Health Monitoring System using IOT," *Indian Journal of Science and Technology*, vol. 9, no. 33, 2016, doi: 10.17485/ijst/2016/v9i33/95625.

[17] I. Kabashkin and L. Shoshin, "Artificial Intelligence of Things as New Paradigm in Aviation Health Monitoring Systems," *future Internet*, vol. 16, no. 8, 2024, Art. no. 276, doi: 10.3390/fi16080276.

[18] K. Lima, T. D. Oyetoyan, R. Heldal and W. Hasselbring, "Evaluation of MQTT Bridge Architectures in a Cross-Organizational Context," in *2025 IEEE 22nd International Conference on Software Architecture (ICSA) Jan. 2025*, Odense, Denmark, Mar. 31–4, 2025, pp. 243–254, doi: 10.1109/ICSA65012.2025.00032.

[19] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform," in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, Mar. 21–23, 2018, pp. 1–5, doi: 10.1109/INFOTEH.2018.8345544.

[20] M. Babiuch, P. Foltýnek and P. Smutný, "Using the ESP32 Microcontroller for Data Processing," in *2019 20th International Carpathian Control Conference (ICCC)*, Krakow-Wieliczka, Poland, May 26–29, 2019, pp. 1–6, doi: 10.1109/CarpathianCC.2019.8765944.

[21] V. Thirupathi and K. Sagar, "Implementation of Home Automation System using MQTT Protocol and ESP32," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 2C2, pp. 111–113, 2018.

[22] S. Pawar, N. Panigrahi, A. P. Jyothi, M. Lokhande, D. Godse and D. B. Jadhav, "Evaluation of Delay Parameter of MQTT Protocol," *International Journal of Engineering Trends and Technology*, vol. 71, no. 3, pp. 227–235, 2022, doi: 10.14445/22315381/IJETT-V71I3P223.

[23] Q. Zhang, Y. Xu, X. Wang, Z. Yu and T. Deng, "Real-Time Wind Field Estimation and Pitot Tube Calibration Using an Extended Kalman Filter," *mathematics*, vol. 9, no. 6, 2021, Art. no. 646, doi: 10.3390/math9060646.

[24] N. Daidzic, "Efficient General Computational Method for Estimation of Standard Atmosphere Parameters," *International Journal of Aviation, Aeronautics, and Aerospace*, vol. 2, no. 1, 2015, Art. no. 3, doi: 10.15394/ijaaa.2015.1053.

[25] J. Wahyudi, M. Asbari, I. Sasono, T. Pramono and D. Novitasari, "Database Management in MYSQL," *Jurnal Pendidikan*, vol. 6, no. 2, pp. 2413–2417, 2022.

[26] *InHg to mbar conversion table*, SensorsONE Ltd., Jul. 1, 2025. [Online]. Available: https://www.sensorsone.com/inhg-to-mbar-conversion-table/

[27] Z. Feng, L. Zheng and B. Ren, "In-situ validation of embedded physics-based calibration in low-cost particulate matter sensor for urban air quality monitoring," *Urban Climate*, vol. 59, 2025, Art. no. 102289, doi: 10.1016/j.uclim.2025.102289.

[28] A. Aljubairy, W. E. Zhang, A. Shemshadi, A. Mahmood and Q. Z. Sheng, "A system for effectively predicting flight delays based on IoT data," *Computing*, vol. 102, no. 9, pp. 2025–2048, 2020, doi: 10.1007/s00607-020-00794-w.