MRS Communications (2023) 13:416–424
© Crown 2023
https://doi.org/10.1557/s43579-023-00364-z

MRS Communications
Research Letter

Implementation of an edge-fog-cloud computing IoT architecture in aircraft components

Ramona Dogea, and Xiu T. Yan, Department of Design, Manufacture and Engineering Management (DMEM), University of Strathclyde, Glasgow, Scotland
Richard Millar, National Manufacturing Institute Scotland – University of Strathclyde, Glasgow, Scotland

Address all correspondence to Ramona Dogea at **ramona.dogea@strath.ac.uk**

(Received 4 January 2023; accepted 20 April 2023; published online: 8 May 2023)

Abstract

This paper presents a deployment of an IoT architecture in aircraft wings based on a physical asset, three IoT platforms, MATLAB® on a personal computer and ThingSpeak as a cloud. The IoT architecture was designed considering five layers and implemented using a simple wireless sensor network. Temperature, humidity, air quality and air pressure were collected, pre-processed and visualised in real-time to improve the reliability of aircraft components. The results show that embedding sensory capability into wing components can create a smart ecosystem that will support different IoT-enabled services in-flight, and it can also be used for predictive maintenance purposes.

Introduction

In today's industries, the aerospace sector is impacted by Industry 4.0 technologies to support design, maintenance and in-flight structural health.[1] Aircraft manufacturers and operators have been trying to enhance performance through connected systems-to-systems using the Internet of Things (IoT) paradigm. In particular, the Internet of Things for Aerospace (IoTA) allows aircraft systems with in-built computing devices to communicate with other devices through a secure network that works on a centralised and decentralised model.[2]

Traditionally, commercial aircraft systems consist of complex subsystems like wings that generate lift and control systems that provide flight monitoring and control. Nowadays, the maintenance of such systems becomes complex when the aircraft needs to be in service on a 24 × 7 basis, and it requires more resources concerning time and costs. With the latest technologies, real-time health monitoring of these aircraft systems is feasible, and they can be continuously supervised to take preventive actions before a system failure occurs.[3]

Current research is focused on designing fog computing architectures in unmanned aerial vehicles (UAVs) to improve power consumption, latency and efficiency for meeting time constraints when acquiring, processing and storing data at different levels, but commercial aircrafts still remains an unexplored field.[4] In Ref. 5 it is mentioned that the aerospace company Bombardier used the fog computing paradigm for real-time analysis of its aircraft engines' data. In addition, there is no established architecture design considering the paradigm IoT-fog-edge-cloud computing in aircrafts.[6] Therefore, there is a need to continuously explore ways of designing and integrating IoT architectures in aircraft components in order to improve the performance and reliability of aircrafts.

The structural performance of an aircraft can be improved by integrating intelligent sensors, actuators and microprocessors.[7] Considering Industry 4.0 technologies, it is possible to embed IoT sensors into structures that allow the collection of life cycle data. These sensors can be used to move away from a condition-based preventive maintenance system to a data-driven predictive maintenance-based system.[8] In general, a smart asset can be developed by means of[9]: sensors embedded in the physical component to gather data, an IoT device to link the physical object to the Internet and to send gathered data, analysis capabilities for converting data into information and making predictions, and hardware for visualising information.

The RSL10 sensor development kit was used in this project since it is a node-to-cloud IoT platform to develop IoT applications with Bluetooth Low Energy (BLE) system-on-chip (SoC). This platform includes the NOA1305, an ambient light sensor; the BME680 environmental sensor that measures temperature, humidity, pressure and air quality (0: clean air, 500: heavily polluted air); the BHI160, a 3-axis accelerometer and gyroscope sensor; and the BMM150, a magnetometer. The RSL10 is also equipped with a flexible near-field communication (NFC) antenna, and the firmware can be flashed via a 10-pin needle adapter. Figure 1 shows the RSL10 and its main components. This work considered the application of three RSL10 platforms and a small prototype.

Data collection from the RSL10 platform can be achieved through single-board computers (SBC). To integrate the physical assets with the IoT, current researchers are working on designing and deploying edge-fog-cloud architectures.[10] Cloud computing is one of the IoT applications that receives all data given from the sensors and stores it. The concepts of edge computing and fog computing are implemented to prevent massive streaming data from being sent and stored in the cloud.

Sensor	Measurement range
NOA1305	0.165 lux to over 100,000 lux
BME680	Pressure: 300...1100 hPa Humidity 0…100% Temperature: -40…85°C Index for air quality: 0 …500
BHI160	Acceleration ± 16g Gyroscope 2,000 °/s
BMM150	±1300μT (x,y-axis) ±2500μT (z-axis)

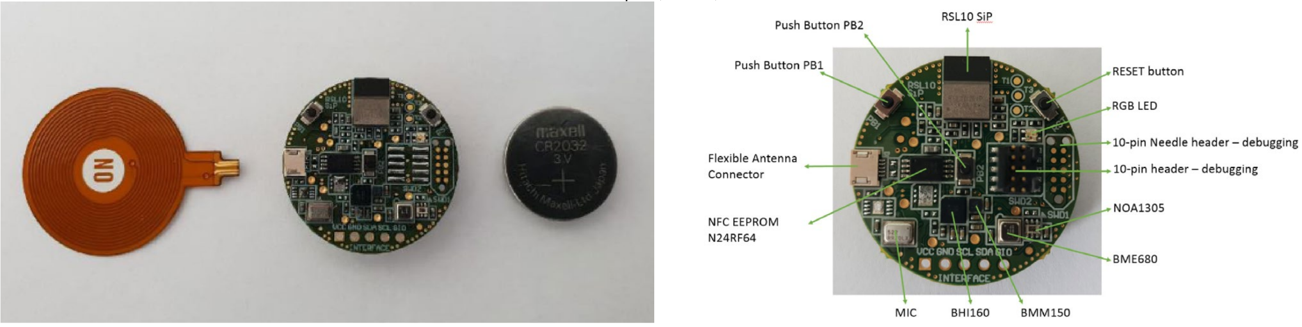


Figure 1. Sensors included in the RSL 10 platform and their measurement range (top). Bottom left: RSL10 platform: NFC antenna (left), platform (middle), battery supply (right). Bottom right: RSL10 platform with main components.

In edge computing, an edge device connects to the sensors and sends data to the cloud, i.e., the edge device serves as a bridge between the physical and virtual layers. Fog computing is an additional layer between the edge and cloud layers that acts as a computing layer and pre-processes data before sending it to the cloud. Thus, the fog layer allows the cloud to receive only the relevant data useful for the specific purpose.[11–13]

In this work, MATLAB® was used as an edge server for data collection, processing and storage instead of sending all collected data to the cloud. In addition, edge computing was considered since data processing helps reduce the amount of data sent to the cloud. While in cloud computing mode data is not pre-processed, in edge computing mode MATLAB® processes input data and sends the most relevant data to the cloud.[14] This research considered MATLAB® as an edge device in the edge computing layer, and in the fog computing layer a MATLAB® script was written to pre-process sensor data before sending it to the cloud. Researchers commonly use MATLAB® to program since it is easy to learn and flexible,[15] and it is usually chosen for client–server communication.[16] The client is a process that connects to a message bus, whilst the server is a process that listens for and accepts connections. Once the connection between both assets has been established, there is no distinction between a client and a server.[17] In this research, MATLAB® was used as a client, while the three RSL10 platforms were the server.

Nowadays, BLE devices are employed to design and implement wireless sensor networks.[18] A wireless sensor network (WSN) can be defined as a network of devices that can transmit the information collected from a monitored application through a wireless connection. Data from each device may be forwarded through multiple nodes, which can be linked to other networks through a gateway. A sensor node can sense, detect events and control local actuators. A WSN can thus remotely monitor physical and environmental magnitudes, like temperature and pressure, and co-operatively pass data through the network to a sink location.[19] A WSN can be divided into different groups consisting of parent and children nodes. While children nodes send data to the parent, the parent sends the information to the sink location within an established sampling rate.[18] Researchers have shown a simple WSN implemented in MATLAB® consisting of three sensors that send collected data to a multiport switch to select it alternatively and send it to a receiver using a specific sample time.[20]

In order to send data to the cloud, the platform ThingSpeak -owned by the MathWorks foundation, the parent enterprise of MATLAB®- was chosen since it is a free IoT platform that allows the collection of real-time data streams, and then their visualisation and analysis. The three main steps that ThingSpeak follows are (a) data collection, (b) data analysis, and (c) acting by causing a response.[21] In this research, the RSL10 platforms were used to collect environmental data, and MATLAB® was applied to process sensory data and send it from the sensors to ThingSpeak. In Ref. 22 the ThingSpeak platform was used to visualise temperature and humidity. Other researchers worked with ThingSpeak to visualise air quality data by plotting graphs of pollutants concentration values.[23,24] The main scope of this work reported here has been to investigate and implement IoT capability on a wing rib prototype used for academic purposes. The purpose of this work was to answer the following research question: *How can an IoT architecture be designed and deployed in an Industry 4.0 ecosystem within an aircraft wing box? How can the interaction between physical and virtual assets be achieved?*

This article includes an introduction, materials and methods, results and discussion, conclusions and further work, and references.

Materials and methods

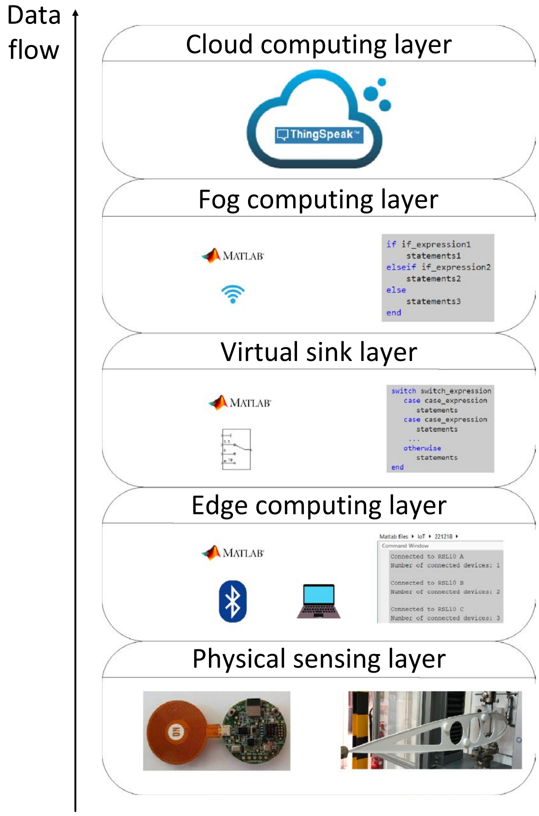
IoT architecture

In order to get the environmental data, an IoT architecture divided into four layers was implemented. The physical sensing layer is on the bottom and consists of the physical objects: wing rib and the three RSL10 IoT platforms. In this work, four magnitudes were collected: temperature, humidity, air quality and air pressure. In the edge computing layer, data from the sensors is collected by MATLAB® installed on a personal computer through a Bluetooth connection. In this application, MATLAB® serves therefore as the IoT gateway between the physical layer and the cloud. Before sending data to the cloud, a MATLAB® script was written for pre-processing the collected data in the fog computing layer. Cloud computing is on the top and it receives data previously pre-processed. In the cloud, data visualisation is done on the IoT platform ThingSpeak. Figure 2 shows this IoT architecture.

Case study

The IoT architecture was implemented using an aircraft wing rib prototype as a physical asset. The sensor locations (SLs) were designed according to the IoT platform RSL10. Integrating sensors into the wing ribs allows the collection of life-cycle data in harsh environmental conditions. Monitoring real-time data can enhance efficiency and improve the performance of the entire aircraft wing. In this case, connectivity means real-time access to critical information, better decision-making and outcomes.

The deployment of the proposed IoT architecture on the wing rib was done by inserting three RSL10 IoT platforms on their respective SLs. This IoT architecture is composed of five layers: the physical sensing layer that contains the three IoT sensor platforms, the edge computing layer corresponding to the personal computer that establishes a connection with the IoT platforms through MATLAB®, a virtual sink layer that selects data alternatively from the sensors to form a WSN, a fog computing layer that pre-processes data before sending it to the ThingSpeak. The data flow through the five layers is shown in Fig. 3.



```
if if_expression1
    statements1
elseif if_expression2
    statements2
else
    statements3
end
```

```
switch switch_expression
   case case_expression
      statements
   case case_expression
      statements
    ...
   otherwise
      statements
end
```

```
Connected to RSL10 A
Number of connected devices: 1

Connected to RSL10 B
Number of connected devices: 2

Connected to RSL10 C
Number of connected devices: 3
```

Figure 2. IoT architecture.

Implementation steps

In order to implement the IoT architecture using the physical assets aircraft wing rib and the RSL 10 platforms, the following steps were realised:

1. Connection establishment between the RSL10 platforms and MATLAB® on the personal computer

In order to set up a connection with the IoT platforms, the MATLAB® function *ble* was used in a *for* loop whose input is the address of each IoT platform. The function *ble* creates three BLE objects, one for each IoT platform. The *try/catch* executes the *ble* function and displays information about whether the connection was established or not. A warning message is displayed if the connection attempt is unsuccessful and the code stops. Algorithm 1 executes the connection between physical sensing and edge computing layers. Once the three IoT platforms get connected, the command window shows the following messages:

```
Connected to RSL10 A               Connected to RSL10 B               Connected to RSL10 C

Number of connected devices: 1     Number of connected devices: 2     Number of connected devices: 3
```

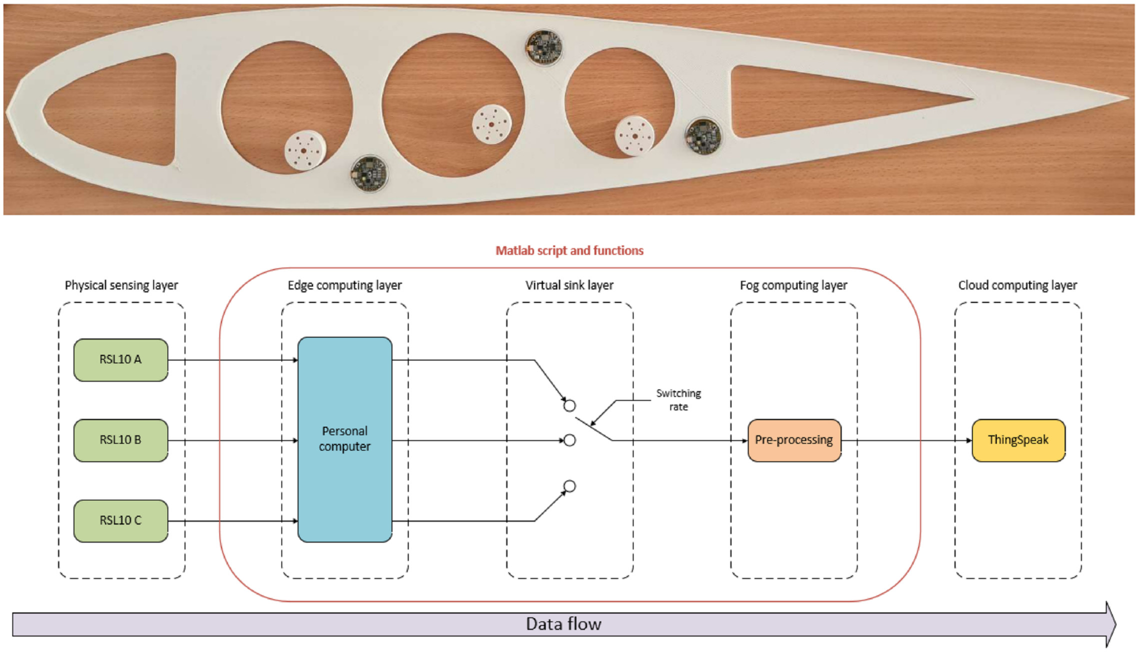


Figure 3. Top: Location of IoT platforms on the wing rib prototype; Bottom: Deployment of the proposed IoT architecture in a wing rib prototype.

2. RSL10 data access with MATLAB®

Algorithm 2 was written to extract the characteristics from the IoT platforms. The MATLAB® function *characteristic* was used to create an object representing a characteristic on each peripheral device b (the BLE object). This function has three inputs: the ble object, the characteristic universally unique identifier (UUID) of the service and the UUID of the characteristic. These UUIDs can be extracted from the ble objects created with Algorithm 1, and they were used to make available the data-receiving and the data-sending ends, named *rxc* and *txc*, respectively. The MATLAB® function *subscribe* was employed to start receiving notifications from the IoT platforms, and to extract data from them, another function named *displayData* was coded. Algorithm 3 was written to request environmental data with a switching rate of 250 ms and a capture time of 3 s. Temperature, humidity, air quality and air pressure, previously calculated in the function *displayData*, are displayed in the command window. In this work, the temperature and timestamps for each IoT platform were stored in the workspace.

3. Data to information conversion

To convert data to information algorithm 4 was coded and it represents the virtual sink where raw data is collected and information is obtained. To display data, in this case, temperature, humidity, air quality and air pressure, the data collected by each platform is read. The functions *char* and *split* were used to convert data for being recognised as requested information.

4. Information pre-processing

Before sending information to the ThingSpeak cloud, a pre-processing algorithm (Algorithm 5) for the temperature was written so that temperatures below 20°C would not be sent to the cloud and the neglected information could be stored in the workspace.

5. Information transfer to the cloud

To transmit the pre-processed data to ThingSpeak, Algorithm 6 was written using the function thingSpeakWrite. Its inputs are the channelID, the data, the WriteKey command, and the channel write API key. In addition, a pause of 15 s between measurements was introduced to avoid the limited update frequency of ThingSpeak, which is dependent on the license type.

6. Data visualisation on cloud

The ThingSpeak channel was previously set up with four field charts to display temperature, humidity, air quality and air pressure. The results will be discussed in the next section.

The algorithms are given in Online Resource as additional information.

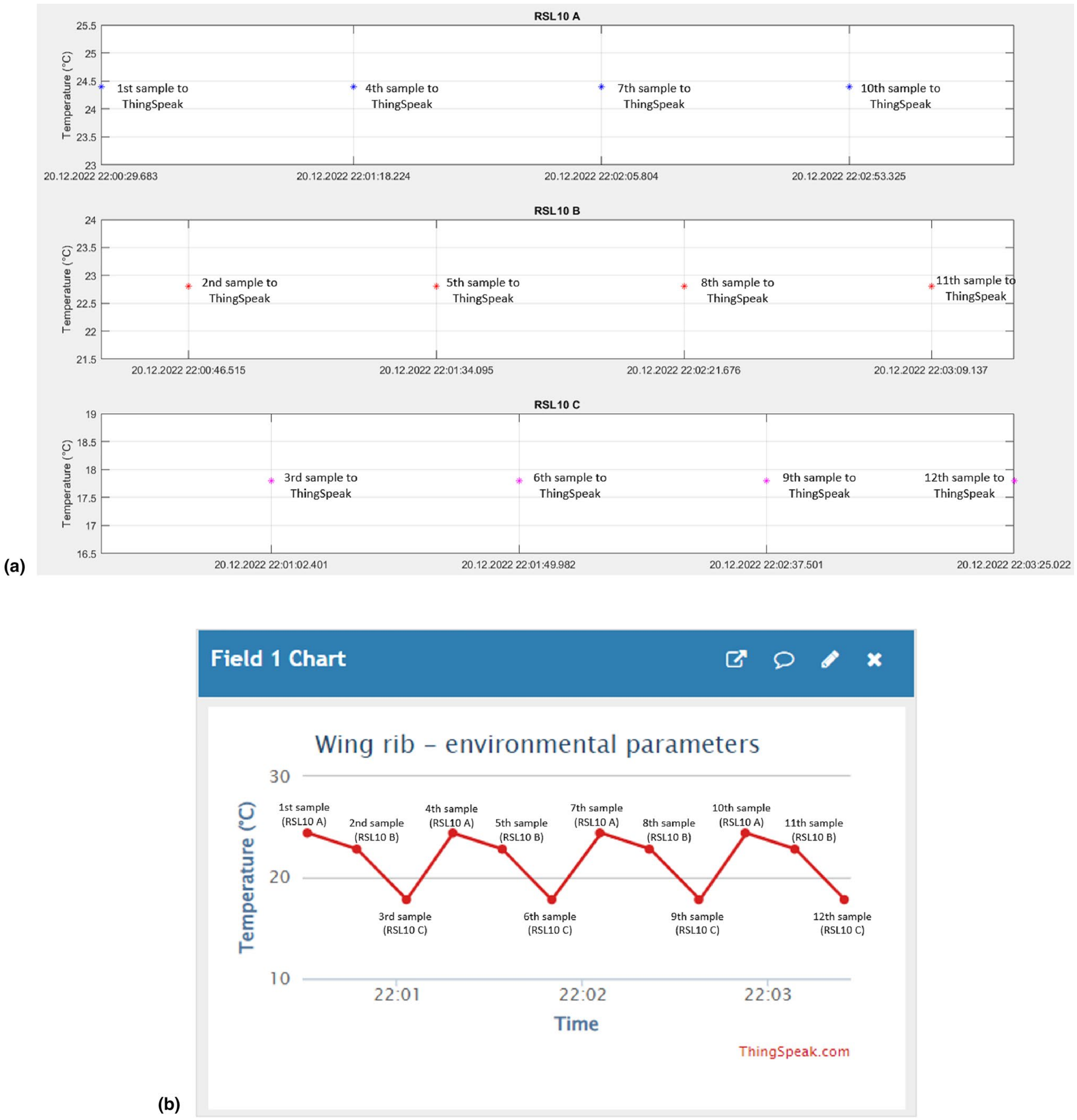


Figure 4. (a) Collected temperature by the IoT platforms; (b) WSN functionality; (c) Collected temperature by the IoT platforms with pre-processed temperature; (d) ThingSpeak visualisation with pre-processed temperature.

Results and discussion

In order to prove the functionality of the WSN, some trials were done by sending the temperature of the three RSL10 platforms to ThingSpeak. The three IoT platforms were placed in three different environmental conditions during these trials. The pre-processing was deactivated in these trials to show how the real-time data was transmitted. Figure 4 shows the temperature from each RSL10 collected through MATLAB®. The samples were sent alternatively to ThingSpeak following the sequence RSL10 A - RSL10 B - RSL10 C. This sequence could also be verified in ThingSpeak. When running the MATLAB® code, the command window displays the data acquisition parameters. For the first three samples, the following information is shown:

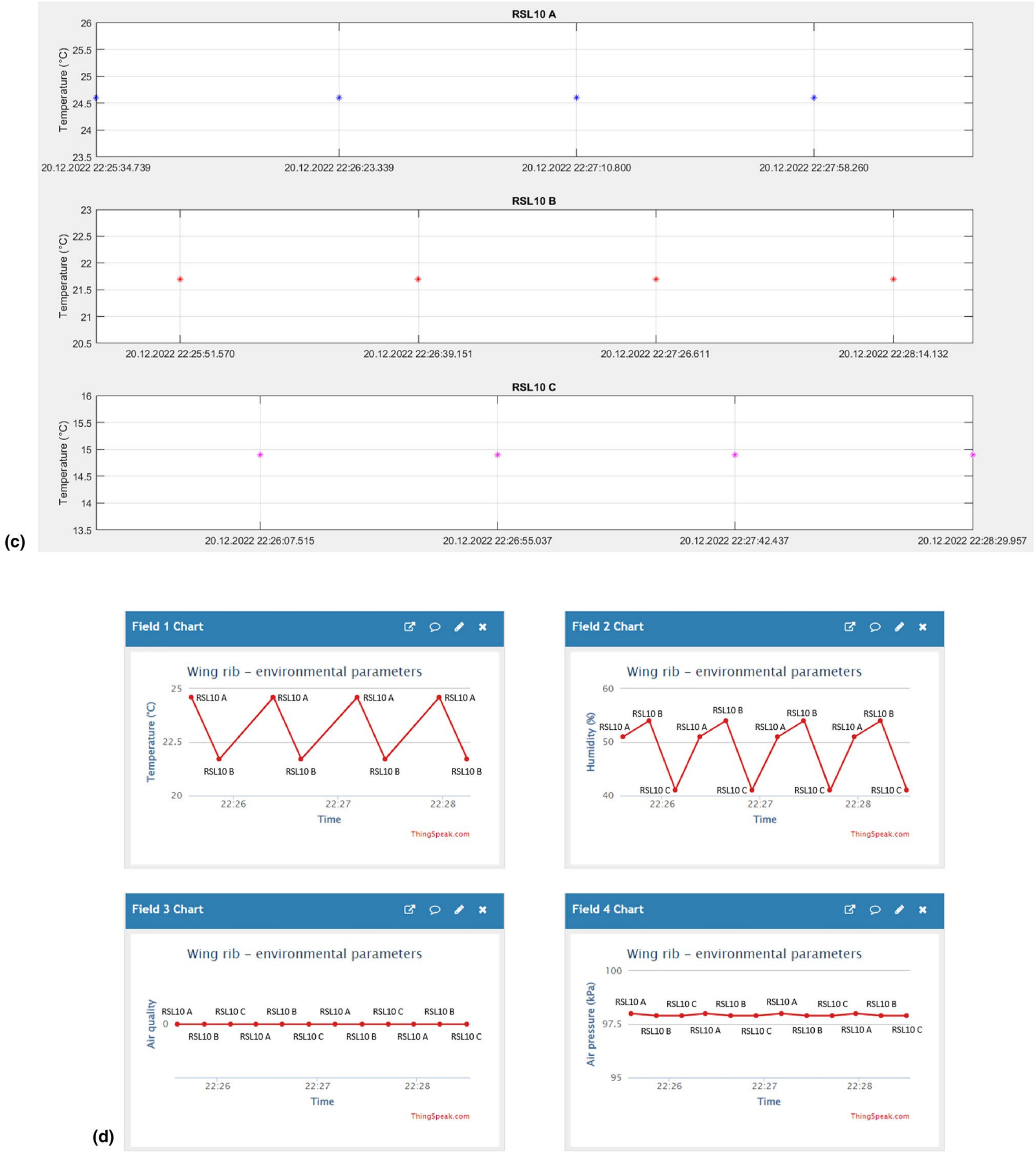
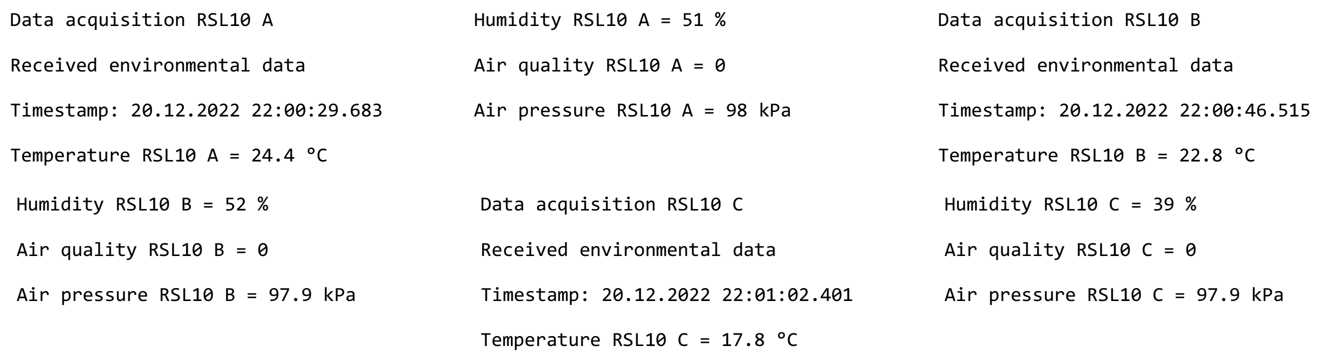


Figure 4. (continued)

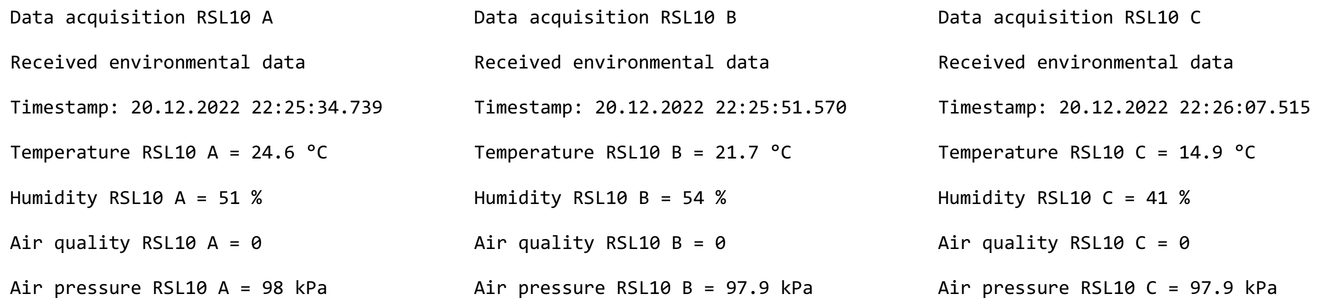


```
Data acquisition RSL10 A
Received environmental data
Timestamp: 20.12.2022 22:00:29.683
Temperature RSL10 A = 24.4 °C
Humidity RSL10 A = 51 %
Air quality RSL10 A = 0
Air pressure RSL10 A = 98 kPa

Data acquisition RSL10 B
Received environmental data
Timestamp: 20.12.2022 22:00:46.515
Temperature RSL10 B = 22.8 °C
Humidity RSL10 B = 52 %
Air quality RSL10 B = 0
Air pressure RSL10 B = 97.9 kPa

Data acquisition RSL10 C
Received environmental data
Timestamp: 20.12.2022 22:01:02.401
Temperature RSL10 C = 17.8 °C
Humidity RSL10 C = 39 %
Air quality RSL10 C = 0
Air pressure RSL10 C = 97.9 kPa
```

Considering the pause of 15 s, the information of each sample is alternatively sent to ThingSpeak. Once the WSN was successfully verified, other trials were performed to check the pre-processing stage by setting in MATLAB® a threshold value for the temperature. In this case, the threshold temperature was set so that only temperature values above 20°C would be transmitted to ThingSpeak. Figure 4 shows that the temperature collected from the RSL10 C was 14.9°C, which was not sent to ThingSpeak. When running the MATLAB® code, the command window displays the data acquisition parameters. For the first three samples, the following information is shown:



```
Data acquisition RSL10 A
Received environmental data
Timestamp: 20.12.2022 22:25:34.739
Temperature RSL10 A = 24.6 °C
Humidity RSL10 A = 51 %
Air quality RSL10 A = 0
Air pressure RSL10 A = 98 kPa

Data acquisition RSL10 B
Received environmental data
Timestamp: 20.12.2022 22:25:51.570
Temperature RSL10 B = 21.7 °C
Humidity RSL10 B = 54 %
Air quality RSL10 B = 0
Air pressure RSL10 B = 97.9 kPa

Data acquisition RSL10 C
Received environmental data
Timestamp: 20.12.2022 22:26:07.515
Temperature RSL10 C = 14.9 °C
Humidity RSL10 C = 41 %
Air quality RSL10 C = 0
Air pressure RSL10 C = 97.9 kPa
```

When applying the pre-processing algorithm, the temperature below the threshold, in this trial 20°C, was not sent to the ThingSpeak cloud. Compared to the case without threshold, the amount of transmitted temperature data to ThingSpeak was reduced by 33%. The pre-processing was not applied for the other environmental parameters, and all real-time information was sent to the cloud.

In order to answer the research questions addressed in the introduction section, the IoT architecture was designed and tested on a wing rib prototype. The interaction between physical and virtual layers was achieved using three IoT platforms and MATLAB®. The transmission of the collected data from the IoT platforms was physically implemented. The pre-processing algorithm was written to filter data before being sent to the ThingSpeak cloud for visualisation.

Conclusions and future work

In order to answer the addressed research questions in this article, an IoT architecture on complex aircraft wing components was designed and implemented to collect real-time environmental data using a wing rib prototype as case study. This implementation considered a physical prototype, three IoT platforms, MATLAB® on a personal computer and ThingSpeak as a cloud. The IoT architecture was designed based on five layers that allow data flow to the cloud and it was also deployed using MATLAB® as a programming tool with a pre-processing algorithm. The results showed that a simple WSN could give a big picture of the environmental conditions within the aircraft wing box. During this work, temperature, humidity, air quality and air pressure were collected and visualised in real-time to improve aircraft components' reliability. The temperature data was pre-processed before being sent to the cloud and the amount of transmitted data was reduced by 33%. In this work, an IoT architecture was designed and tested on a wing rib prototype, the interaction between physical and virtual layers was achieved through the use of three IoT platforms and MATLAB®, the data transmission from the IoT platforms was physically implemented, and a pre-processing algorithm was employed to filter data before being sent to the ThingSpeak cloud for visualisation.

The outcomes of this research confirm that the aircraft wing rib can be integrated into an Industry 4.0 ecosystem by means of IoT technologies that allow the collection of life cycle data in complex aircraft wing components for health monitoring and predictive maintenance purposes. Furthermore, data sent to the cloud helps aircraft manufactures move away from a condition-based preventive maintenance system to a data-driven predictive maintenance-based system. As future work, it is recommended to extend this research by investigating the implementation of machine learning algorithms into aircraft wings for making predictions on the structural performance of the wing box components.

Acknowledgments

The authors acknowledge the support of the University of Strathclyde (Scotland, United Kingdom) and the Universidad Politécnica de Cartagena (Cartagena, Spain) for using its facilities and equipment.

Author contributions

Conceptualisation, RD; methodology, RD; software, RD; validation, RD; formal analysis, XTY, RM and RD; investigation, RD; resources, RD; data curation, RD; writing—original draft preparation, RD; writing—review and editing, XTY and RM; visualisation, XTY and RM; project administration, RD; funding acquisition, XTY and RM. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the University of Strathclyde.

Data availability

All data generated or analysed during this study is included in this published article.

Code availability

The code is available by request.

Declarations

Conflict of interest

The authors declare no conflict of interest.

Open Access

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1557/s43579-023-00364-z.

References

1. S. Aydın, C. Kahraman, Aviation 4.0 revolution. Stud. Syst. Decis. Control **372**, 3–19 (2022). https://doi.org/10.1007/978-3-030-75067-1_1
2. T. Ramalingam, B. Christophe, F.W. Samuel, Assessing the potential of IoT in aerospace, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, (2017). vol. 10595 LNCS. https://doi.org/10.1007/978-3-319-68557-1_11
3. S. Jena, A. Gupta, *Embedded Sensors for Health Monitoring of an Aircraft, in Energy, Environment, and Sustainability* (Springer, Singapore, 2019)
4. M.F. Pinto, A.L.M. Marcato, A.G. Melo, L.M. Honório, C. Urdiales, A framework for analyzing fog-cloud computing cooperation applied to information processing of UAVs. Wirel. Commun. Mob. Comput. **2019**, 1–14 (2019). https://doi.org/10.1155/2019/7497924
5. P. Kansal, D. Sharma, M. Kumar, Introduction to fog data analytics for IoT applications. Stud. Big Data **76**, 19–38 (2020). https://doi.org/10.1007/978-981-15-6044-6_2
6. A. Pati, M. Parhi, B.K. Pattanayak, IoT-fog-edge-cloud computing simulation tools a systematic review. Int. J. Smart Sens. Adhoc Netw. (2022). https://doi.org/10.47893/ijssan.2022.1206
7. Y. Wang, S. Hu, T. Xiong, Y. Huang, L. Qiu, Recent progress in aircraft smart skin for structural health monitoring. Struct. Health Monit. **21**(5), 2453 (2022). https://doi.org/10.1177/14759217211056831
8. C. Liu, H. Vengayil, Y. Lu, X. Xu, A cyber-physical machine tools platform using OPC UA and MTConnect. J. Manuf. Syst. **51**, 61–74 (2019). https://doi.org/10.1016/j.jmsy.2019.04.006
9. J. Rub, H. Bahemia, A review of the literature on smart factory implementation, in *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)* (2019). pp. 1–9. https://doi.org/10.1109/ICE.2019.8792577
10. F. Firouzi, B. Farahani, A. Marinšek, The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT). Inf. Syst. **107**, 101840 (2022). https://doi.org/10.1016/j.is.2021.101840
11. W. Yu et al., A survey on the edge computing for the internet of things. IEEE Access **6**, 6900 (2017). https://doi.org/10.1109/ACCESS.2017.2778504
12. K. Dolui, S.K. Datta, Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing, in *GIoTS 2017 - Global*

Internet of Things Summit, Proceedings (2017). https://doi.org/10.1109/GIOTS.2017.8016213.

13. A. Ometov, O.L. Molua, M. Komarov, J. Nurmi, A survey of security in cloud, edge, and fog computing. Sensors **22**(3), 927 (2022). https://doi.org/10.3390/s22030927
14. K.L. Ketshabetswe, A.M. Zungeru, B. Mtengi, C.K. Lebekwe, S.R.S. Prabaharan, Data compression algorithms for wireless sensor networks: a review and comparison. IEEE Access **9**, 136872 (2021). https://doi.org/10.1109/ACCESS.2021.3116311
15. A. Dhinnesh, Wireless sensor networks and its tools for simulation. GRD J. Glob. Res. Dev. J. Eng. **4**, 45–48 (2019)
16. M.G. Krishnan, A.T. Vijayan, S. Ashok, Real-time integration of industrial robot with MATLAB, in *ICCCE 2021* (Springer, 2022) pp. 123–131
17. M. Woolley, Bluetooth for Linux Developers Study Guide. Version (2021)
18. L. Leonardi, L. Lo Bello, G. Patti, LoRa support for long-range real-time inter-cluster communications over Bluetooth Low Energy industrial networks. Comput. Commun. **192**, 57–65 (2022)
19. S.R. Jondhale, R. Maheswar, J. Lloret, *Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks* (Springer, Cham, 2022)
20. D. Okeke, O.A. Oluwasakin, MATLAB modelling of bridge health monitoring using wireless sensor network (2018)
21. N. Gupta, A. Soni, Y.K. Gond, D. Kumar, An E-healthcare system using IoT and edge computing, in *Machine Intelligence and Smart Systems* (Springer, 2022). pp. 165–177
22. O.G. Hastak, K. Verma, A. Zagdamwar, A. Patil, R. Tembhurnikar, IoT based weather forecasting system using Arduino board. Int. J. Res. Publ. Rev. **3**(4), 2664–2669 (2022)
23. D.S. Katiyar, R. Raj, A.K. Dahiya, "Design and execution of an internet of things based air pollution monitoring device. IEEE Stud. Conf. Eng. Syst. (SCES) **2022**, 1–6 (2022)
24. N. Patel, Y. Bhavsar, S. Mundra, V. Dharia, R. Taware, Detection and monitoring of harmful gases in air using a microcontroller and android app. Lect. Notes Data Eng. Commun. Technol. **92**, 315–324 (2022). https://doi.org/10.1007/978-981-16-6601-8_29

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.