Article

Not peer-reviewed version

Design and Validation of an IoT-Based Blood Pressure Monitoring System for Rabbits

Carlos Exequiel Garay , Gonzalo Nicolás Mansilla , Rossana Elena Madrid , Agustina González Colombres , Susana Josefina Jerez *

Posted Date: 4 March 2026

doi: 10.20944/preprints202603.0296.v1

Keywords: telemedicine; internet of things (IoT); preclinical models; edge computing; cloud computing; 6G

 doi:10.20944/preprints202603.0296.v1

Article

Design and Validation of an IoT-Based Blood Pressure Monitoring System for Rabbits

Carlos Exequiel Garay [1], Gonzalo Nicolás Mansilla [1], Rossana Elena Madrid [2,3], Agustina González Colombres [3] and Susana Josefina Jerez [3,*]

[1] CIASUR (Centro de Investigación de Atmósfera Superior y Radiopropagación) - Universidad Tecnológica Nacional (UTN) - Facultad Regional Tucumán (FRT), Rivadavia 1050, C.P. (4000) Tucumán, Argentina

[2] Laboratorio de Medios e Interfases (LAMEIN), Departamento de Bioingeniería (DBI), FACET, Universidad Nacional de Tucumán

[3] Instituto Superior de Investigaciones Biológicas (INSIBIO), Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), C.P. (4000) Tucumán, Argentina

* Correspondence: sjerez@herrera.unt.edu.ar

Abstract

Telemedicine, driven by the Internet of Things (IoT) and next-generation mobile networks, is essential for managing cardiovascular diseases, where hypertension remains the primary risk factor. In preclinical research, rabbits are superior biological models compared to rodents due to their human-like lipid metabolism. However, conventional blood pressure monitoring in this species is hindered by significant limitations: existing systems are non-portable, lack real-time capabilities, and often necessitate terminal procedures (euthanasia). To address these challenges, this study presents a portable, minimally invasive monitoring system utilizing a pressure transducer in the central auricular artery. The device integrates IoT technology for digital signal processing and seamless wireless data transmission to cloud platforms. This development enables continuous, real-time hemodynamic tracking throughout the experimental period without requiring permanent tethering to desktop hardware. By reducing invasiveness and enhancing data mobility, this system provides a robust framework for the preclinical evaluation of antihypertensive agents and cardiovascular mechanisms, bridging the gap between edge computing and remote clinical diagnostics.

Keywords: telemedicine; internet of things (IoT); preclinical models; edge computing; cloud computing; 6G

---

1. Introduction

In rabbit production systems (commercial farms and laboratory animal facilities), continuous supervision is often limited, whereas rapid microenvironmental fluctuations (e.g., temperature, humidity, ventilation) and early changes associated with health status can progress into stress, disease, and losses if not detected promptly. In high-density settings, manual inspection is intermittent and does not ensure early detection. Therefore, an IoT-based approach, distributed sensors and/or non-intrusive perception, connectivity to transmit small periodic updates, and cloud services for logging and alerts, is key to enabling continuous, scalable monitoring and improving veterinary and husbandry decision-making under operational constraints related to maintenance and connectivity availability [1]. For these reasons, it is essential to explore efficient wireless connectivity alternatives.

The Internet of Things (IoT) is one such alternative. It refers to the digital interconnection of objects, people, and even animals to the internet. The concept was introduced by Kevin Ashton in 1999 with the invention of Radio Frequency Identification (RFID) [2]. Currently, there are 20 billion objects connected to the internet, and by 2034, it is estimated that this number will reach 40.6 billion. Therefore, it is important to consider the limitations of these devices in terms of storage, processing, computing, and security capabilities [3].

When IoT technologies are applied to healthcare, the cost and efficiency of medical services can be improved and expanded by automating tasks that were previously performed by humans [4]. To fulfill this objective, it is necessary to implement wireless technologies that are efficient in both energy consumption and data transmission. These technologies must meet various requirements such as reliability, interoperability, energy efficiency, low-latency response, mobility, security, among others.

Most animal health monitoring systems are implemented as wearable devices that use low-bandwidth communication technologies to transmit data to the cloud. Several studies have addressed species monitoring from different technological perspectives. Tomaszewski and Kołakowski proposed a 5G mobile network-based approach for monitoring fish in aquaculture environments [5]. Meanwhile, [6] presents large-scale IoT network architectures oriented toward applications in agriculture, forestry, crop production, and livestock farming.

6G technology envisions seamless and ubiquitous connectivity, with coverage extending to underwater, aquatic, terrestrial, and aerial environments, thereby overcoming the limitations inherent in 5G networks. Moreover, it is estimated that 6G will be capable of supporting up to 10^7 connected devices per square kilometer [7], enabling comprehensive and real-time monitoring of various aspects related to animal health and their environments without encountering scalability issues.

Wearable devices for animal health monitoring enable the measurement of physiological parameters such as blood pressure, heart rate, respiratory rate, and other vital signs, thereby assisting caregivers in the early detection and prevention of serious diseases [8]. Bello and Passaglia presented a wireless sensor designed to measure intraocular pressure in conscious animals [9]. Tested in mice, the device transmits data via Bluetooth to a computer for storage and visualization. Chen et al. proposed a system consisting of a telemetry sensor for measuring body temperature in rabbits, which transmits the measurements to a computer for storage and analysis [10].

Monitoring and controlling blood pressure in animals is essential for the early detection of cardiovascular diseases and for monitoring medical and experimental treatment. In rabbits, blood pressure is a key indicator of health and well-being, as they are prone to cardiovascular conditions [11]. The reference technique (gold standard) for measuring blood pressure in rabbits is direct measurement using a pressure transducer, which involves carotid artery catheterization followed by euthanasia. The most commonly used indirect methods today include Doppler, oscillometric, and plethysmographic techniques [12–15]. However, none of these methods allow for continuous blood pressure monitoring; they only provide pressure values at a specific moment in time [16]. Among these, only the Doppler technique has demonstrated some correlation with the direct method. Therefore, the development of a continuous, non-invasive direct blood pressure monitoring system based on IoT could offer an innovative and effective solution. Given the aforementioned considerations, it is crucial to investigate the implementation and performance of IoT in the field of animal health for the early detection of chronic diseases.

This study contributes by presenting and explaining an IoT model, structured in four layers, designed for monitoring blood pressure in rabbits: the cloud layer, the edge layer, the physical layer, and the network layer [17]. This technological development enables the monitoring of pressure changes in an experimental animal (specifically a rabbit) over the course of the experimental period and facilitates the evaluation of various therapeutic interventions, allowing for preclinical assessment of antihypertensive agents.

2. Materials and Methods

In the four-layer system developed , the cloud layer is represented in this case, by ThingSpeak, an open-source platform for the Internet of Things (IoT) that enables the collection, storage, analysis, and visualization of data from sensors connected to the internet [18]. The edge layer is where user-level computational activities are enabled. In this case, it is where the Fast Fourier Transform (FFT) of the rabbit's physiological parameters is performed (see Figure 1).

 doi:10.20944/preprints202603.0296.v1

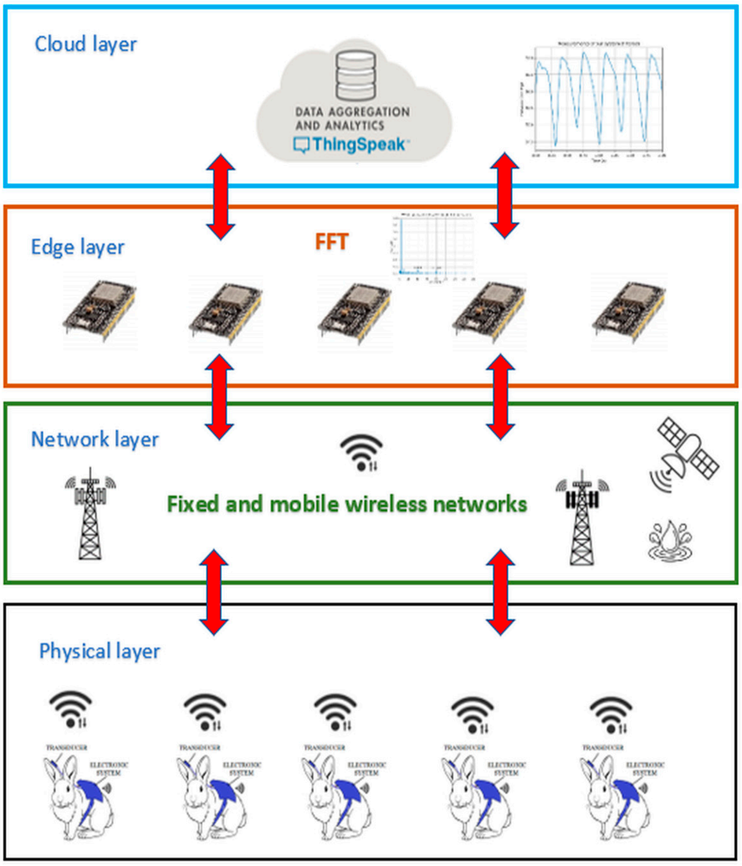

Figure 1. Layers of the system.

The physical layer is responsible for activating the rabbits' blood pressure sensor to collect data. This data is then made available for processing and for generating information about the animals' health status. The network layer provides the necessary infrastructure to enable data communication and overall connectivity. This layer includes technologies such as Wi-Fi networks (when coverage is available), 5G mobile wireless networks, and, in the near future, 6G networks, which facilitate reliable, remote, high-speed internet access and the ability to coordinate and manage network complexity [17].

2.1. System Architecture

The billions of devices being connected to the internet, such as wearables with limited hardware capabilities, are not able to store the data they generate. One approach to address this limitation is to transfer the data to the cloud for storage and processing. However, due to the massive number of devices and their limited processing power and bandwidth, cloud-based latency can be significantly high. To mitigate this issue, fog or edge computing provides the necessary capabilities to offer an effective solution.

Fog Computing is a "system-level architecture that distributes resources and services of computing, storage, control, and networking anywhere along the continuum from the Cloud to Things" [19]. Fog computing is essential for emerging IoT applications (such as industrial automation, transportation, medical devices, etc.) that demand real-time or predictable latency. Due to its geographic distribution, fog computing is well-suited for locally processing information generated by IoT nodes, enabling applications that require low latency [20].

Cloud services include data processing, storage, and analysis, as well as security, privacy, alert and notification services, device management, content management, user services, and more [21]. In the present case, the cloud service is implemented using ThingSpeak [18].

2.2. System Development

The blood pressure monitoring system consisted of a vest adapted to the rabbit's body (Figure 2a), with a pocket to house the pressure transducer and another pocket containing the electronic system (Figure 2b). The blood pressure sensor, in this case the Edwards Px260 pressure transducer, consists of a matrix of strain gauges arranged in a Wheatstone bridge configuration. The output voltage varies according to the pressure applied to the transducer, resulting in a voltage change at the two output terminals of the Wheatstone bridge. Figure 42c shows a schematic of the developed system positioned on the rabbit

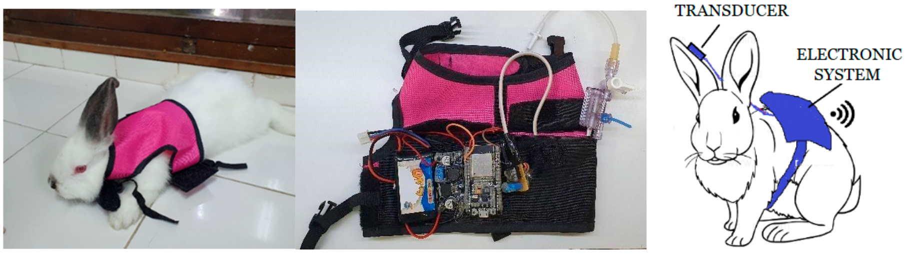

Figure 2. (a) Backpack installed on one of the rabbits that houses the measurement system; (b) Electronics of the measuring system; (c) The system placed on the rabbit.

The sensor output is amplified using the INA122 integrated circuit, a precision instrumentation amplifier, to elevate the voltage level to a range detectable by the analog-to-digital conversion (ADC) module of the ESP32 microcontroller. The amplification circuit follows the configuration recommended in the INA122 datasheet [22] for amplifying signals from a Wheatstone bridge, with the addition of a capacitor connected between the output of the IC and ground. The inclusion of this capacitor, recommended by the ESP32 manufacturer, serves to reduce noise that may be picked up by the ADC module [23].

Figure 3 presents a block diagram of the implemented system.

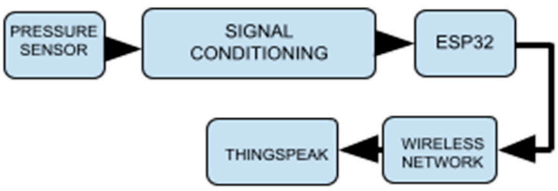

Figure 3. Block diagram of the system.

The circuit of the implemented instrumentation amplifier is shown in the following Figure:

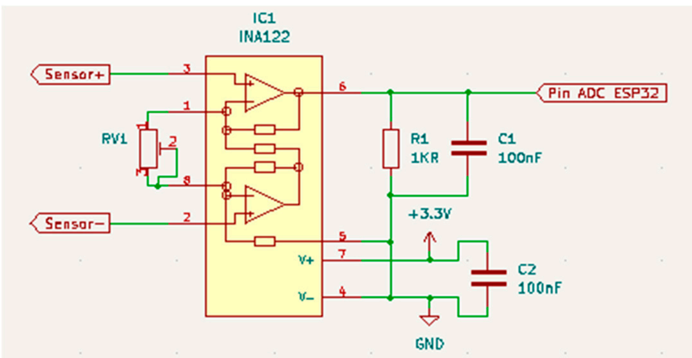

Figure 4. Instrumentation amplifier circuit.

The ESP32 microcontroller captures the signal from the amplifier at a sampling frequency of 1000 Hz, acquiring approximately 10,000 samples to obtain several cycles of the rabbit's heartbeat (3–6 Hz). Finally, the samples are transmitted to the ThingSpeak platform via a wireless connection, where the necessary procedures are performed to present the acquired data in a simple format for the relevant personnel. From this dataset, the systolic and diastolic pressure values are extracted, and the following formula is applied to determine the mean arterial pressure (MAP):

$$MAP=(2 \cdot DBP+SBP)/3 \qquad (1)$$

where DBP is the diastolic blood pressure and SBP is the systolic one.

Once these results are obtained, they are transmitted via the internet to the ThingSpeak platform for subsequent analysis by the end user. The ESP32 microcontroller is programmed to perform this procedure at various time intervals, starting from 1 minute and extending as needed. In this case, the microcontroller is configured to execute the measurement routine either every 12 hours or every 10 minutes, depending on the specific requirements for monitoring the rabbit's behavior.

To calculate and measure time intervals, the microcontroller relies on its internal clock. However, due to fluctuations in its operating frequency, temporal estimation errors are introduced. These errors accumulate over time, potentially causing significant deviations between the user-defined wait time and the time estimated by the device. To compensate for this drift, the system periodically (every 1 hour) queries an NTP (Network Time Protocol) time server as a correction and synchronization mechanism. A time server is a computer with access to a reference clock, used to synchronize the date and time of devices connected to a network [24], while NTP is the network protocol used to transmit this information [25]. This ensures that measurements are performed at the pre-established times.

Moreover, since the ESP32 remains idle for most of the time while waiting for these intervals to elapse, the device's SLEEP modes were utilized to increase energy efficiency by reducing current consumption. SLEEP modes are strategies used to minimize the energy consumption of a microcontroller when it is not actively performing tasks. In this work, the DEEP-SLEEP mode of the ESP32 is utilized, which powers down the processor, RAM, and all internal modules of the device, except for those necessary to enable system wake-up.

Exit from this state occurs through an external event or an interrupt generated by a module that remains active and functions as the system's "wake-up" trigger. Specifically, the RTC (Real Time Clock) module is employed. In addition to its primary role of maintaining time and date, the RTC includes a configurable internal timer. This timer allows the setting of a wait interval after which a countdown begins; once it reaches zero, an event is generated that wakes the ESP32 from DEEP-SLEEP mode and restarts program execution from the beginning.

Additionally, the RTC includes a small backup RAM that remains powered during DEEP-SLEEP mode. This feature is particularly useful, as it allows the storage of relevant information prior to entering this mode, preventing the loss of variables due to the power-down of the main RAM. In this way, it is possible to preserve the necessary data for the correct execution of the program upon waking, without being affected by the limitations inherent to the DEEP-SLEEP mode [26].

Finally, the acquired samples are packaged and transmitted to a cloud service for subsequent storage and processing.

2.3. Rabbits Preparation

For this study, four male hybrid rabbits (a cross between Californian and New Zealand breeds), weighing between 2 kg and 3 kg, were used. The animals were obtained from the rabbit production unit of the Faculty of Agronomy and Animal Science at the National University of Tucumán (UNT) and housed in the rabbit vivarium of INSIBIO. The vivarium is a facility that houses animals with defined genetic and microbiological quality, used for research purposes. The protocols were unanimously approved by the Institutional Committee for the Care and Use of Laboratory Animals (CICUAL) of UNT under research protocol No. 076/2023, dated June 2, 2023.

 doi:10.20944/preprints202603.0296.v1

Prior to the rabbits' housing into the quarantine room, a thorough cleaning and disinfection of cages, floors, and surfaces was carried out. Upon arrival at the vivarium, the rabbits were placed individually in metal cages, equipped and integrated into a micro-ventilated rack system that complies with international standards on animal welfare and biosafety. The animals remained under quarantine for 15 days to rule out any pathologies, stabilize immune function, and perform internal parasite examinations.

During this period, detailed records were kept on: date of entry, previous illnesses, treatments, other assessments, body weight, behavior, among other factors. The animals were housed under a 12-hour light/12-hour dark cycle, controlled by a mechanical timer. Ambient temperature was maintained at 24±1°C using air conditioning and monitored with a wall-mounted thermometer.

Daily maintenance included changing bedding, removing soiled wood shavings with feces and urine, disinfecting trays, and refilling with clean shavings. Each day, the rabbits were weighed, their general condition assessed, and their water and food intake measured [27].

2.4. Experimental Procedures

The experiments were conducted over a one-month period. Initially, the performance of the proposal system was assessed. Subsequently, the system was validated by comparing the obtained data with those from the reference system: direct blood pressure measurement.

The experimental subjects were divided into two groups, each consisting of two rabbits. In the first group, the performance of the proposal system was assessed by placing the pressure transducer in the central auricular artery of the rabbit's ear (Figure 5). The catheterization was optimized using an Abbocath 24G catheter, which remained in place throughout the experimental period. Rabbits were immobilized using a restraint device specifically designed at INSIBIO-CONICET for performing experimental procedures. During the 15 days prior to the procedure, the animals were placed in this device daily at the same time to acclimate them and reduce stress, which could otherwise alter the blood pressure values.

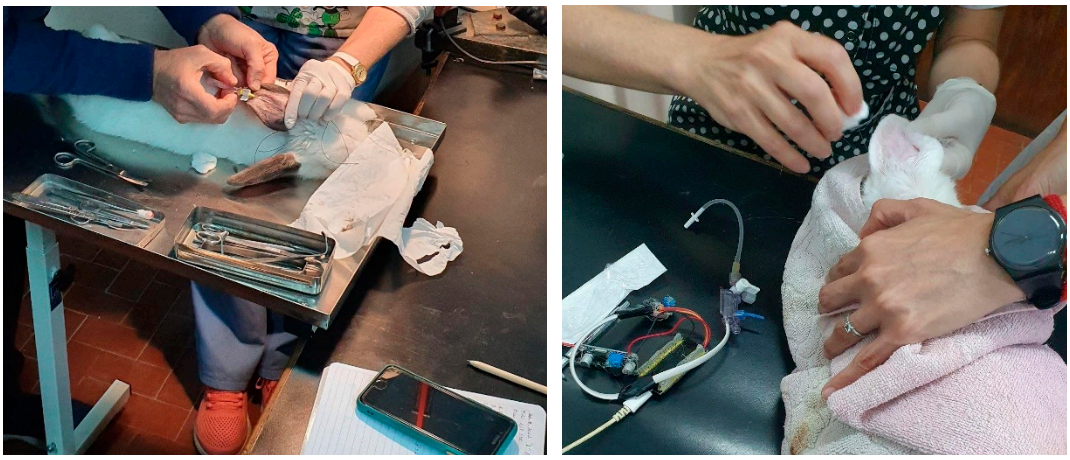

Figure 5. Procedure for placing an intraarterial catheter placement for the proposed system.

The procedure was as follows: after visually identifying the auricular artery, the inner surface of the ear was shaved, disinfected with 5% povidone-iodine (Pervinox), and locally anesthetized using 7% lidocaine gel. The catheter was then inserted and secured with hypoallergenic adhesive tape. The animal's behavior in response to the foreign object was monitored every four hours during the first 48 hours. If the catheter remained viable after this period, the blood pressure measurement was conducted.

To perform the measurement, the rabbit was placed in the immobilization device, the catheter was connected to the pressure transducer, and the arterial pressure signal was captured and

 doi:10.20944/preprints202603.0296.v1

transmitted to the cloud for approximately 15 minutes (the exact duration varied depending on the electronic variables involved). This procedure was carried out on two rabbits from the first group.

In the second group, composed of two additional rabbits, the data obtained from the first group were validated by comparing them with the gold standard blood pressure measurement technique, applied to the carotid artery.

The procedure was as follows: Animals were fasted for 8 hours, weighed, and their vital signs were monitored. Pre-anesthesia was administered subcutaneously using diazepam at a dose of 5 mg/kg. Anesthesia was then induced via a subcutaneous injection of ketamine (50 mg/kg) [28]. The neck was shaved in the region of the carotid artery. Subsequently, 7% lidocaine was injected subcutaneously prior to the first skin incision. Using a scalpel, an incision was made at the level of the carotid artery, which runs parallel to the trachea. The skin, muscle, and fat layers were dissected until the artery was located. Surrounding fat and nerves were cleared, and a small incision was made in the artery to insert the catheter. The catheter was secured with sutures, and a temporary clamp was applied to momentarily obstruct blood flow..

Simultaneously, the same procedure as that used for the first group was applied: catheterization of the auricular artery was performed, and the pressure transducer (Edwards Px260) was connected to transmit data to the cloud.

Finally, the catheter placed in the carotid artery was connected to a pressure transducer (Gould, Germany), the arterial clamp was removed, and blood pressure readings were acquired using the BIOPAC MP100 data acquisition and analysis system, model MP100. This system was specifically designed for measuring biological signals and features a 12-bit resolution D/A converter [29]. The system is connected to a PC, as shown in Figure 9. In this way, simultaneous blood pressure values were obtained from both the auricular and carotid arteries for comparative analysis. All animals were euthanized by exsanguination at the end of each experiment.

3. Results

3.1. System Calibration

To calibrate the system, the pressure transducer is connected to an external hydraulic system, which generates a pressure pulse with an amplitude of 15 mmHg. The transducer is calibrated in this way by measuring its response to the pulse. Figure 6 represents the transducer response.

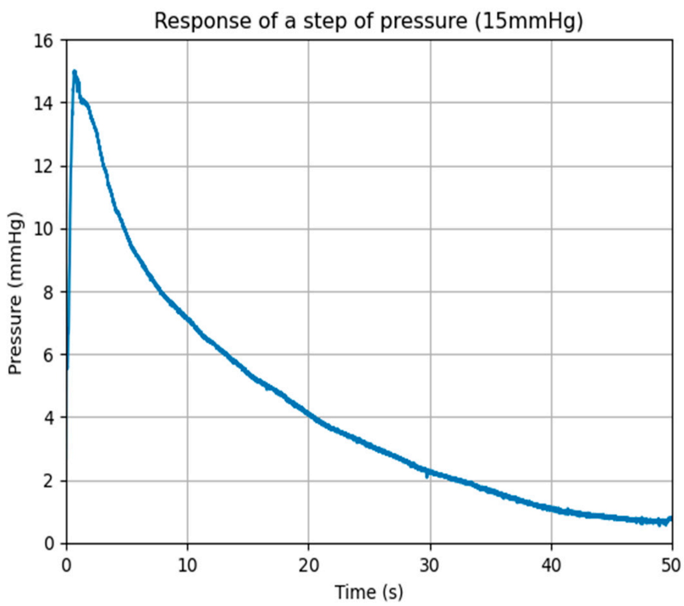

Figure 6. System response to a pressure pulse of 15 mmHg.

3.2. Data Acquisition and Processing

To maintain independence between the microcontroller's firmware and the sensor and amplification stage, the value transmitted to the cloud is the raw ADC count. Conversion and calibration coefficients are then applied in the cloud, resulting in a modular firmware design that is not tied to a specific sensor and/or amplifier.

A common source of external noise in devices that operate with transducers emitting signals in the millivolt range is interference from the electrical power grid [30,31]. Figure 7 shows the FFT of the rabbit's blood pressure signal captured by the ADC module, where peaks can be observed at the power line frequency and its first harmonic—50 Hz and 100 Hz in this case.

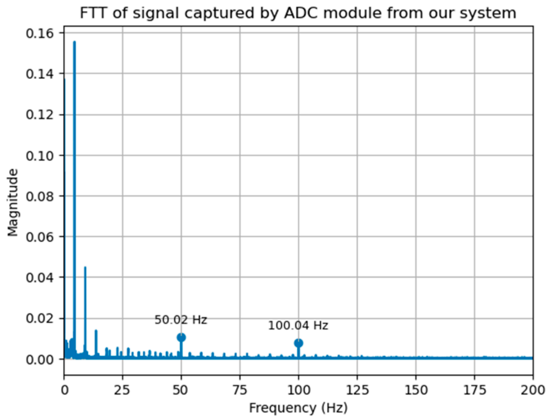

Figure 7. FTT of the rabbit's blood pressure signal captured by ADC module of the developed system.

Since the frequency range of interest is lower than that of the power grid, a filter can be applied to reduce the noise caused by this interference. A digital filter offers the advantage of being immune to noise, as it is implemented via software or digital circuits [32]. Therefore, it is not affected by electrical noise from the power grid.

The implemented filter is an IIR Butterworth filter, selected for its maximally flat response in the passband [33]. To eliminate signal distortion caused by transients typically introduced by filtering, the "forward-backward filtering" technique was applied [34].

The ThingSpeak platform sends, receives, and stores data in character format. For example, to send the value 10 to ThingSpeak, it must first be converted into a character string, which is then transmitted over the internet using an application protocol such as HTTP or MQTT [18]. ThingSpeak organizes incoming data using channels, each of which can store up to 8 fields of data. Following this structure, storing 10,000 samples would require 1,250 channels. However, the academic and standard licenses of ThingSpeak only allow the creation of up to 250 channels, meaning that five licenses would be needed. Additionally, the firmware would have to be programmed with 1,250 channel identifiers along with their respective write keys in order to transmit the data.

To reduce the number of channels required without decreasing the volume of data sent, a data-packing strategy was devised. This method allows multiple data points to be encoded into a single field, thereby drastically reducing the number of channels required. ThingSpeak stores the numerical data in each field as character strings, with a maximum length of 255 characters. This feature allows multiple samples to be grouped within a single field, which can later be unpacked in the cloud to retrieve the original dataset. Since the maximum value produced by the ADC module is 4095, each sample is represented as a 4-character string. These strings are concatenated sequentially up to a total length of 252 characters—without exceeding the platform's field limit—allowing 63 samples per field and a total of 504 samples per channel.

Using this method, 10,080 values can be stored using only 20 channels, significantly reducing the number of channels required while preserving the existing workflow and functionality of ThingSpeak. Subsequently, to reconstruct the original dataset, the reverse process is applied. First,

 doi:10.20944/preprints202603.0296.v1

the character string stored in each field is retrieved and split into 4-character substrings. Each substring is then converted back into its numerical format, and the resulting values are sequentially grouped to recover the originally acquired numerical dataset (Figure 8).

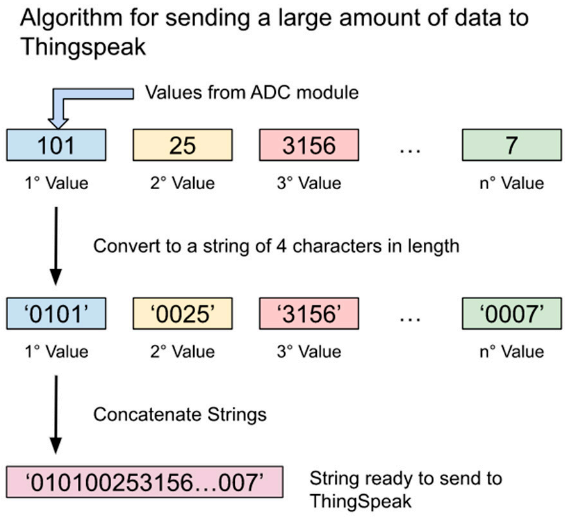

Figure 8. Algorithm for sending a large amount of data to Thingspeak.

The "format" of the data sent to ThingSpeak is not natively recognized by the platform; therefore, it is necessary to implement a script or program to interpret the data and display it graphically. Within each channel, ThingSpeak provides a space to execute MATLAB code, which can be used to create "interactive windows" such as line plots, bar charts, histograms, etc., using the data sent to the platform. Through this functionality, it is possible to develop a script that retrieves the character strings from the used channels to reconstruct the original sequence of values. In this way, the recovered data can be filtered, graphically represented, or processed to extract parameters of interest, such as systolic and diastolic pressure, from the signal captured by the device.

To improve the overall system performance, part of the processing was moved to the edge (ESP32 microcontroller), while another part remains in the cloud (ThingSpeak). The edge, which is responsible for managing the pressure sensor operation, is also where the Fast Fourier Transform (FFT) is performed on the data. This allows only the necessary information for visualizing the frequency spectrum of the captured signal to be transmitted to the cloud. Since it is not essential to capture all signal details for frequency spectrum analysis—and considering that the device measures biological signals, which are typically low-frequency—a reduction in the sampling rate to 250 samples per second was applied. This ensures capture of the signal's frequency spectrum, as well as the 50 Hz and 100 Hz noise components from the power grid (50 Hz in this case and its first harmonic).

Furthermore, because the FFT of the captured signal is consistent regardless of the system's application, this processing can be performed on the device without compromising firmware modularity. The FFT results are transmitted to ThingSpeak in the same manner as the raw signal values, with the difference that each data string begins with a 1 to indicate a negative value or a 0 for a positive value. The amplitude values are limited between -100 and 100, and only one decimal digit is included. This results in 1 character for the sign indicator, 3 characters for the integer part, and 1 character for the decimal part—so each FFT value occupies 5 characters.

The data are later adjusted in the cloud according to the system's calibration coefficients, and the corresponding graph is generated. Although there are algorithms available for data compression—both lossy and lossless—that reduce the number of bytes required and allow for more data to be packed into a single transmission [35–37], their implementation typically requires prior knowledge of the data to be compressed. In the case of lossy algorithms, it is necessary to determine whether the algorithm can reconstruct the original signal in a way that allows for equivalent analysis

 doi:10.20944/preprints202603.0296.v1

as when using a reference instrument. For lossless algorithms, it is important to assess whether the nature of the data justifies the additional processing required, both at the cloud and edge levels, and whether the reduction in data size warrants the extra computational effort.

Although reference instrument data were available for analysis and for determining the most efficient compression algorithm, the decision was made to proceed as previously described. This approach was chosen in order to test and evaluate the performance of the sensor adaptation circuit in a real-world setting, particularly given the difficulty of repeating the experimental procedure.

4. Discussion

To evaluate the performance of the developed blood pressure monitoring system, measurements obtained using the system were compared with those acquired simultaneously through the direct method (gold standard).

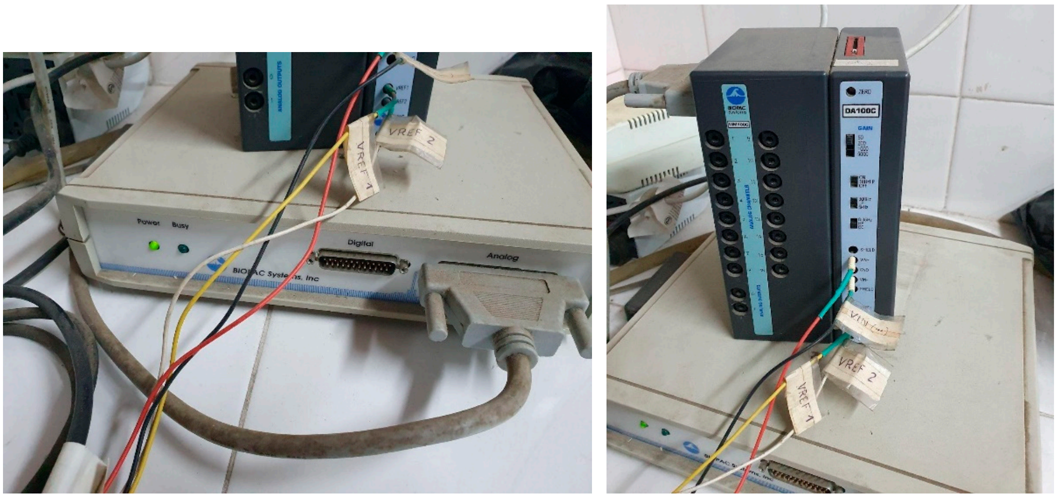

Figure 9. Reference device BIOPAC model MP100.

The following Figure shows the pressure signal obtained with the reference BIOPAC system (Figure 10a), while Figure 10b shows the pressure signal obtained with the proposed system:

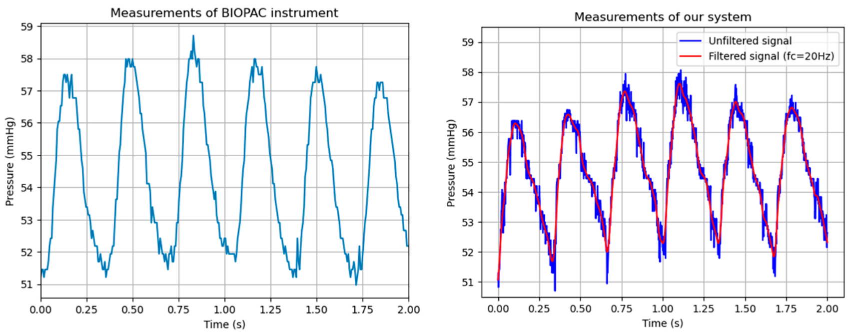

Figure 10. (a) Pressure signal obtained with the BIOPAC instrument; (b) Pressure signal obtained with the proposed system (Original in blue, and filtered in red).

Beat-to-beat metrics were extracted from 11 recordings, each containing 10,000 samples (10-second windows at 1 kHz). Systolic blood pressure (SBP) was defined as the local maximum of each cardiac cycle, and diastolic blood pressure (DBP) as the minimum point between consecutive systolic peaks. Mean arterial pressure (MAP) was calculated per beat using equation (1).A total of **230** cardiac cycles were detected (median of **29** beats per recording; range: **26–32**). Considering all beats, the SBP, DBP, and MAP obtained values are shown in Table 1 compared to the reference values obtained with

BIOPAC. MAP showed the smallest bias (ΔMAP ≈ **−0.91**; ~**−1.60%**), suggesting that the proposed system reliably captures the mean pressure component, which is less sensitive to waveform distortions and noise. In contrast, the largest discrepancy was observed for SBP (≈ **−5.34**; ~**−8.58%**), consistent with the higher sensitivity of systolic peak estimation to high-frequency attenuation, transducer coupling variability, and local-maximum detection differences. Overall, the combined results support the robustness and reproducibility of the proposed approach, particularly for MAP tracking under comparable experimental conditions.

Table 1. Validation of the proposed system. and session-level stability.

Metric	BIOPAC PA (mean ± SD)	Proposed system (all beats, mean ± SD)	Relative error (%)	Proposed system (session mean ± SD, N=10)	Range (min–max)
Detected beats (n)	230	230	—	—	—
SBP	62.24 ± 1.46	56.90 ± 2.07	-8.58	56.71 ± 1.74	54.14–58.41
DBP	54.00 ± 1.04	55.31 ± 2.21	+2.43	55.10 ± 1.78	52.00–56.72
MAP	56.75 ± 1.17	55.84 ± 2.11	-1.60	55.64 ± 1.72	52.72–57.28

Abbreviations: SBP, systolic blood pressure; DBP, diastolic blood pressure; MAP, mean arterial pressure. SBP, DBP, and MAP are reported as mean ± SD. Beat-level comparison uses BIOPAC arterial pressure (PA) as reference. Absolute error is defined as Δ = (Proposed − BIOPAC); relative error (%) = 100·Δ/BIOPAC. Session-level statistics correspond to per-recording averages across N = 10 recordings (range: min–max).

5. Conclusions

An IoT-based blood pressure monitoring system has been developed for rabbits, enabling real-time, beat-to-beat estimation of mean arterial pressure (MAP), systolic blood pressure (SBP), and diastolic blood pressure (DBP). Compared to the direct measurement of blood pressure (gold standard) reference, the proposed system showed relative errors of **1.60%** for MAP, **8.58%** for SBP, and **2.43%** for DBP.

The IoT-based blood pressure monitoring system offers several advantages over conventional blood pressure measurement methods in rabbits.

First, the device is portable and minimally invasive, reducing animal stress and avoiding the need for terminal euthanasia by exsanguination at the end of each experiment.This allows for monitoring disease progression throughout the course of treatment and evaluating its effectiveness.

Second, the IoT platform enables remote, real-time access to blood pressure data, facilitating continuous health monitoring of the rabbits. Third, the methodological approach adopted in this work can be adapted for blood pressure monitoring in other animal species as well as in humans.

This experience also contributed to the development of a foundational database that will support future improvements, such as the implementation of data compression algorithms, which have been shown to reduce data size and network traffic, as well as enhance energy efficiency at the edge [21,38].

Acknowledgments: The team gratefully acknowledges Dr. Eduardo Gaspar Álvarez Villamil from CERELA – CONICET for his assistance with the surgical procedures performed on the experimental rabbits. This work was funded by CIASUR (UTN–FRT) and institucional funds from the Instituto Superior de Investigaciones Biológicas (INSIBIO), CONICET.

References

1. Han, Y.; Wei, T.; Chen, Z.; Wang, H.; Wang, L.; Li, C.; Mei, X.; Kuang, L.; Gong, J. LIVEMOS-G: A High Throughput Gantry Monitoring System with Multi-Source Imaging and Environmental Sensing for Large-Scale Commercial Rabbit Farming. Animals 2025, 15(21), 3177. https://doi.org/10.3390/ani15213177

2. Ashton, K. That ‘Internet of Things’ Thing. RFID J. 2009, 22, 97–114.
3. Vailshery, L.S. Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2021, with Forecasts from 2022 to 2034; Statista: Hamburg, Germany, 2022; Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 15 February 2026).
4. Yan, H.; Xu, L.D.; Bi, Z.; Pang, Z.; Zhang, J.; Chen, Y. An Emerging Technology – Wearable Wireless Sensor Networks with Applications in Human Health Condition Monitoring. Journal of Management Analytics 2015, 2, 121–137, doi:10.1080/23270012.2015.1029550.
5. Tomaszewski, L.; Kołakowski, R. Mobile Services for Smart Agriculture and Forestry, Biodiversity Monitoring, and Water Management: Challenges for 5G/6G Networks. Telecom 2023, 4, 67–99, doi:10.3390/telecom4010006.
6. 5G Americas. Mobile Communications Beyond 2020—The Evolution of 5G Towards the Next G; 5G Americas: Bellevue, WA, USA, 2020; Available online: https://www.5gamericas.org/wp-content/uploads/2020/12/Future-Networks-2020-InDesign-PDF.pdf (accessed on 16 February 2026).
7. Giordani, M.; Polese, M.; Mezzavilla, M.; Rangan, S.; Zorzi, M. Towards 6G Networks: Use Cases and Technologies 2020.
8. Karthick, G.S.; Sridhar, M.; Pankajavalli, P.B. Internet of Things in Animal Healthcare (IoTAH): Review of Recent Advancements in Architecture, Sensing Technologies and Real-Time Monitoring. SN COMPUT. SCI. 2020, 1, 301, doi:10.1007/s42979-020-00310-z.
9. Bello, S.A.; Passaglia, C.L. A Wireless Pressure Sensor for Continuous Monitoring of Intraocular Pressure in Conscious Animals. Ann Biomed Eng 2017, 45, 2592–2604, doi:10.1007/s10439-017-1896-3.
10. Chen, Y.; Niimi, M.; Zhang, L.; Tang, X.; Lu, J.; Fan, J. A Simple Telemetry Sensor System for Monitoring Body Temperature in Rabbits—A Brief Report. Animals 2023, 13, 1677, doi:10.3390/ani13101677.
11. Pariaut, R. Cardiovascular Physiology and Diseases of the Rabbit. Veterinary Clinics of North America: Exotic Animal Practice 2009, 12, 135–144, doi:10.1016/j.cvex.2008.08.004.
12. Williams, P.B.; Schapiro, H.; Yeiser, P.E. Noninvasive Blood Pressure Determination in the Rabbit with a Doppler Ultrasound Probe. Proceedings of the Society for Experimental Biology and Medicine 1979, 161, 417–420, doi:10.3181/00379727-161-40564.
13. Jekl, V.; Agudelo, C.F.; Hauptman, K. Cardiology in Rodents, Rabbits, and Small Exotic Mammals—Diagnostic Workup. Veterinary Clinics: Exotic Animal Practice 2022, 25, 503–524, doi:10.1016/j.cvex.2022.01.010.
14. Calero Rodriguez, A.; Van Zeeland, Y.Ra.; Schoemaker, N.J.; De Grauw, J.C. Agreement between Invasive and Oscillometric Arterial Blood Pressure Measurement Using a High-Definition Oscillometric Device in Normotensive New Zealand White Rabbits Using Two Different Anaesthetic Protocols. Veterinary Anaesthesia and Analgesia 2021, 48, 679–687, doi:10.1016/j.vaa.2021.03.016.
15. Barter LS, Epstein SE. Comparison of Doppler, Oscillometric, Auricular and Carotid Arterial Blood Pressure Measurements in Isoflurane Anesthetized New Zealand White Rabbits. Vet Anaesth Analg. 2014 Jul;41(4):393-7. Doi: 10.1111/Vaa.12131. Epub 2014 Feb 27. PMID: 24571422.
16. González-González, E.; González-Alonso-Alegre, E.; Montesinos-Barceló, A.; Caro-Vadillo, A. Non-invasive Blood Pressure Measurement in Conscious Rabbits: A Comparison of Doppler Ultrasonic and Oscillometric Devices. Veterinary Record 2024, 195, e4399, doi:10.1002/vetr.4399.
17. Akhigbe, B.I.; Munir, K.; Akinade, O.; Akanbi, L.; Oyedele, L.O. IoT Technologies for Livestock Management: A Review of Present Status, Opportunities, and Future Trends. BDCC 2021, 5, 10, doi:10.3390/bdcc5010010.
18. The MathWorks, Inc. Learn More About ThingSpeak; The MathWorks, Inc.: Natick, MA, USA, 2026; Available online: https://thingspeak.mathworks.com/pages/learn_more (accessed on 16 February 2026).
19. OpenFog, C. Openfog Reference Architecture for Fog Computing. URL: https://www. openfogconsortium. org/ra 2017.
20. Markakis, E.; Mastorakis, G.; Mavromoustakis, C.X.; Pallis, E. Cloud and Fog Computing in 5G Mobile Networks: Emerging Advances and Applications; Institution of Engineering and Technology, 2017; ISBN 1-78561-083-X.

21. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach. Future Generation Computer Systems 2018, 78, 641–658, doi:10.1016/j.future.2017.02.014.
22. Texas Instruments. INA122 Datasheet. Available online: https://www.ti.com/lit/ds/symlink/ina122.pdf (accessed on 16 February 2026).
23. Espressif Systems. ADC API Reference (ESP-IDF v4.4). Available online: https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html (accessed on 16 February 2026)
24. Harrenstien, K. Time Server; RFC 738; Internet Engineering Task Force: Fremont, CA, USA, 1977. Disponible en línea: https://www.rfc-editor.org/rfc/rfc738.html (accedido el 16 de febrero de 2026).
25. Mills, D.; Martin, J.; Burbank, J.; Kasch, W. Network Time Protocol Version 4: Protocol and Algorithms Specification; RFC 5905; Internet Engineering Task Force: Fremont, CA, USA, 2010. Disponible en línea: https://www.rfc-editor.org/rfc/rfc5905 (accedido el 16 de febrero de 2026).
26. Espressif Systems. Sleep Modes API Reference. Available online: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep_modes.html (accessed on 16 February 2026).
27. INSIBIO. Apoyo a la Investigación—Conejos. Available online: http://insibio.org.ar/apoyo-a-la-investigacion/conejos/ (accessed on 16 February 2026).
28. Comolli, J.; d'Ovidio, D.; Adami, C.; Schnellbacher, R. Technological Advances in Exotic Pet Anesthesia and Analgesia. Veterinary Clinics: Exotic Animal Practice 2019, 22, 419–439.
29. BIOPAC Systems, Inc. AcqKnowledge Software Guide. Available online: https://www.biopac.com/wp-content/uploads/acqknowledge_software_guide.pdf (accessed on 16 February 2026).
30. Limaye, H.; Deshmukh, V.V. ECG Noise Sources and Various Noise Removal Techniques: A Survey. International Journal of Application or Innovation in Engineering & Management 2016, 5, 86–92.
31. Mewett, D.T.; Nazeran, H.; Reynolds, K.J. Removing Power Line Noise from Recorded EMG.; IEEE, 2001; Vol. 3, pp. 2190–2193.
32. Tompkins, W.J. Biomedical Digital Signal Processing; Prentice Hall New Jersey, 1993; Vol. 237
33. Lyons, R.G. Understanding Digital Signal Processing; 3rd ed.; Prentice Hall: Upper Saddle River, NJ, 2011; ISBN 978-0-13-702741-5.
34. Gustafsson, F. Determining the Initial States in Forward-Backward Filtering. IEEE Transactions on signal processing 2002, 44, 988–992.
35. De Oliveira, M.A.; Da Rocha, A.M.; Puntel, F.E.; Cavalheiro, G.G.H. Time Series Compression for IoT: A Systematic Literature Review. Wireless Communications and Mobile Computing 2023, 2023, 1–23, doi:10.1155/2023/5025255.
36. Smith, S.W. The Scientist and Engineer's Guide to Digital Signal Processing; 2nd edition.; California Technical Pub.: San Diego (Calif.), 1999; ISBN 978-0-9660176-7-0.
37. Chiarot, G.; Silvestri, C. Time Series Compression Survey. ACM Computing Surveys 2023, 55, 1–32.
38. Azar, J.; Makhoul, A.; Barhamgi, M.; Couturier, R. An Energy Efficient IoT Data Compression Approach for Edge Machine Learning. Future Generation Computer Systems 2019, 96, 168–175, doi:10.1016/j.future.2019.02.005.