Architecture Analysis, Cloud-Based Using MQTT Protocol for Braille Literacy

Josimar dos Santos, Gilton José Ferreira da Silva and Admilson de Ribamar Lima

September 18, 2023

Architecture Analysis Cloud-Based Using MQTT Protocol for Braille Literacy

Josimar dos Santos
Program for Graduate Studies in Computer Science (PROCC)
Federal University of Sergipe (UFS)
Aracaju-SE, Brazil 49100-000
Email: josimarsts@academico.ufs.br

Gilton José Ferreira da Silva
Program for Graduate Studies in Computer Science (PROCC)
Federal University of Sergipe (UFS)
Aracaju-SE, Brazil 49100-000
Email: gilton@dcomp.ufs.br

Admilson de Ribamar Lima
Program for Graduate Studies in Computer Science (PROCC)
Federal University of Sergipe (UFS)
Aracaju-SE, Brazil 49100-000
Email: admilson@ufs.br

***Abstract*—The use of assistive technologies and social inclusion is becoming increasingly important in both traditional education and active learning methodologies. In innovative education, technologies can play a crucial role in teaching people with disabilities (PWD), offering new perspectives in learning. In this context, this work is specifically aimed at individuals with visual impairments, specifically those with total loss of vision, whether congenital or acquired, or even for individuals interested in learning Braille.**

The learning needs in this case are based on sensory perception, such as touch, taste, smell, and hearing. Therefore, the objective of this study is to present a cloud-based architecture that integrates a tactile reading device and Braille display, which is simple, low-cost, and uses a lightweight messaging protocol like the *Message Queuing Telemetry Transport* (MQTT), widely used in Internet of Things (IoT) architectures.

In this architecture, the entire process of converting text into Braille occurs in the cloud. The used device has voice commands issued, received, and processed through an Application Programming Interface (API), commonly known as API, which performs the conversion of text to Braille. Then, the device prepares to display the points in high or low relief in a Braille cell, representing each character of the text. In this way, the visually impaired person can read the character through touch sensitivity.

This architecture could provide a practical and accessible solution for learning Braille and for reading by visually impaired individuals, enabling greater inclusion and active participation in the educational process.

I. Introduction

The Information and Communication Technologies (ICT) have become consolidated through the processes of globalization, bringing significant changes and characteristics to the perspective of consumerism that permeate digital and social media. ICTs are present in our daily lives, with many of these technologies already integrated into humanity. We use them often without even realizing the level of evolution we have reached, constantly seeking to create, improve, or correct processes and connections, sending or receiving data, a subject for the next topic. Kenski (2007) states that “technologies are as old as the human species.” Hence, it is essential to reflect on the relationship between education and technology. About the formation of human society with technology [...] [1]. This excessive connectivity allows devices to also communicate, interact, and work together to achieve a specific objective or make decisions, as in Internet of Things (IoT) sensor networks. Consequently, with the development process, many devices, tools, and computational systems become increasingly capable of being ubiquitous, with dynamic and virtual connections that define the characteristics of Cloud Computing [2].

In this case, how to obtain a basic and low-cost architecture that allows conversion to the Braille system and obtain an interpretable output?

According to the research by [3], one of the highest rates among disabilities – visual, auditory, motor, and mental or intellectual – is visual impairment, with 20.1% in the age range between 15 and 64 years. This rate represents all individuals who declared having some level of difficulty in seeing. However, this study will be inclined to favor people with total loss of vision, but educators and people with sight interested in learning Braille can also benefit.

This study is organized as follows: firstly, aspects related to people with disabilities (PWDs) are discussed, with a specific focus on visual impairment. In addition, topics such as the Internet of Things, MQTT protocol, cloud computing, 3D modeling, maker culture, as well as their implications in the project using Arduino to compose the device, are addressed.

Subsequently, the processes of the research methodology adopted in this study are presented. Then, the proposed architecture is exposed, highlighting its main elements and operation. The results obtained from the application of this architecture are then discussed.

The final considerations of the study are presented, addressing the main insights, conclusions, and possible directions for future work. Finally, the bibliographic references that underpinned and substantiated the study are listed. This structure ensures a comprehensive and systematic approach to the topic, providing a detailed and well-founded analysis of the project at hand.

II. Methodology

According to the statements by [4], the methodology based on systematic mapping involves planning research with the intention of mapping all literature of empirical and non-empirical studies in a specific thematic area that can be submitted to accounting, selection, qualification, and finally, data extraction to not only answer the main question but also derived questions. In the following sequence, the authors relate systematic mapping to five process steps in the following order: defining research questions; Conducting Primary Studies Research; Screening articles based on inclusion/exclusion criteria; article classification; data extraction and aggregation.

In this chapter, the processes used to return the results, designated through a systematic mapping conducted between September 2021 and December 2021, are gathered.

A. Internet of Things

For [5], IoT is defined as: "[. . .] a system of cooperation between connected smart devices." These devices can be anything, as long as they can connect to the internet, with the purpose of sending and receiving data using the cloud and protocols to compose an "ecosystem." Defining actions, displaying data, performing tasks. It is with this intention that IoT fits into the architecture of this project. Having a terminal that can receive data, process it, and display it in a way that the user can interpret is the idea we need to "display" the conversion of points into Braille, that is, high and low-relief points so that the user can learn or "read" what is being "displayed."

B. Cloud Computing

The proposal of cloud computing enables exploring services and obtaining availability, mobility, flexibility, security, sharing, and low infrastructure cost, as reported by [6].

[7] addresses how the infrastructure facilitates the scalability of these resources, and maintenance can be outsourced. "[. . .] Resources such as processing and storage can be contracted/reserved according to demand." [7].

C. Modeling, 3D Printing, and Maker Culture

3D printing has proven to be a revolutionary technology that has positively impacted the lives of its users. By offering the possibility of building prototypes and customized objects to meet specific needs, either individually or in collaboration with others, it has opened new creative and practical perspectives [8].

The journey of 3D printing began in 1980 with Chuck Hull, a pioneer in creating stereolithography technology. This process uses laser heating of liquid elements that solidify to form the desired object. The construction with a 3D printer occurs by adding fragments, layer by layer, until the object is completed [8].

Among the various technologies available for building 3D objects, the use of plastic filament stands out, where the filament is melted and injected at specific positions to construct the object in the printer. The general process of creating 3D parts can be summarized in two steps: software modeling and printing of the developed piece.

Currently, there are various accessible software options for 3D modeling, with *AutoDesk® Tinkercad* being an example recommended for beginners due to its usability and free availability. In this type of modeling, the approach is based on adding and modifying positive and negative three-dimensional shapes, generating or "destroying" content in the physical world [8].

In the context of this project, 3D modeling and printing are fundamental to creating a final product, even if it is a prototype, with a more appealing presentation. The combination of modeling and 3D printing technology allows the development of a *case* structure to protect wires, microcontrollers, and used parts, as well as to build the parts that make up the mechanics of the Braille relief points, for example.

The modernization of the market and accessibility to new technological products, such as the 3D printer, have driven the maker culture. This movement encompasses people who seek to learn concepts from various areas, such as design, carpentry, and technology, through practical experiences. The maker culture encourages individuals to build new products, disassemble objects to understand their manufacturing processes, and thus satisfy their creative and functional needs [8].

An important aspect of the maker movement is the promotion of creative learning through experiences in maker spaces, where people create prototypes and develop new products that meet their specific needs or those of a certain audience [8].

In the context of product prototyping in the maker culture, we can understand that Arduino is a technology widely used to build prototypes for electronic and computational devices. It is important to note that the Arduino technology was developed for rapid prototyping of products for computational physics. In the development of the Arduino technology, we must be clear that it is characterized by three aspects: 1. A software: which is a platform for program development. 2. A legal aspect: which consists of using the technology without worrying about copyright issues. 3. A hardware: boards with electronic components. There are different types of boards when it comes to hardware, just as there are different software options for product development [8].

In summary, 3D printing has played a significant role in promoting creativity and people's ability to materialize their ideas and needs. The emerging maker culture strengthens this movement, driving creative learning and encouraging the development of personalized and innovative solutions. With technology constantly evolving, we can expect 3D printing to continue to have a positive impact on society, empowering people to turn their ideas into reality.

D. Braille System

In order to understand how the conversion process will be performed and displayed on the Braille display, we need to comprehend the Braille system itself. [9] explain that the formation of a three-by-two (3 rows and 2 columns) matrix of raised dots constitutes a Braille cell, as shown in Figure 1. Different combinations of raised or non-raised dots form representations of letters, numbers, and symbols for tactile mapping. Furthermore, when Braille cells are arranged side by side, they form words [9]. The Braille alphabet can be seen in Figure 2.

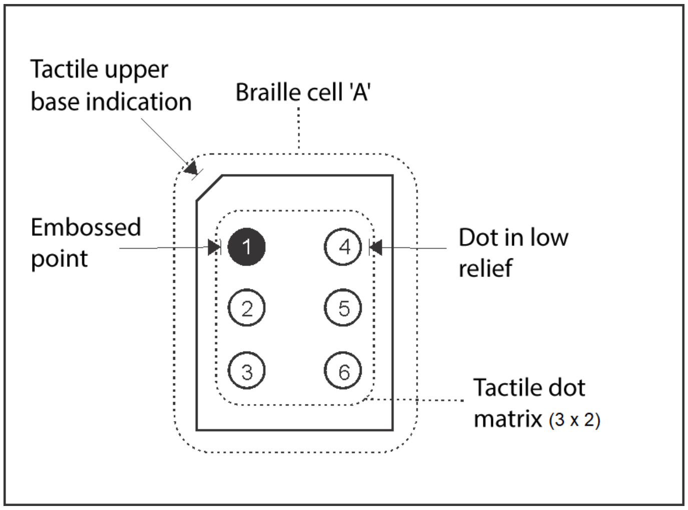

Figure 1. Braille Cell – Representation of the letter A – Authors (2021)

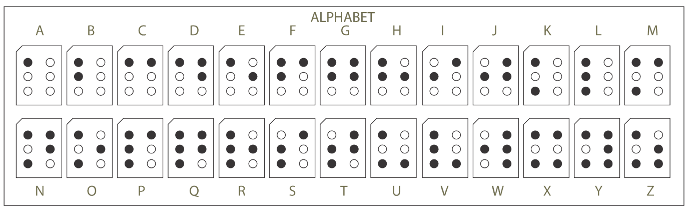

Figure 2. Braille Alphabet – Authors (2021)

E. Structural Concept

The work by [10] aims to transform the inclusion of visually impaired individuals in social discussion forums where transmission is done through speech. With the availability of three main components – Voice to Text, Text to Braille, and Text to Voice - and the ability to save, listen, and delete, according to the presented architecture, the stored file can be converted to Braille and printed.

The study conducted by [9] discusses available technologies that assist visually impaired individuals, ranging from smartphones to tactile Braille input devices and applications. However, the authors note that despite existing innovations, some tutors still use boards with raised or indented points to provide access to the Braille code, requiring more time to teach each student individually and specialized tutor training. [9] propose mass teaching of the Braille code, making the teaching process flexible, fast, and easy to use. The Microcontroller-based Actuator and Cortex-M4 with an Embedded System, the Braille Cell, is a project that allows the tutor to teach multiple students at once.

F. Discussion about the Works

For a better understanding, we have classified subsections that direct the data extracted from each selected study and their respective authors.

1) Architectures and Applicability: According to [10], the application is aimed at tutors/presenters who assist visually impaired individuals in their forums. This system contains the following main functionalities: it recognizes speech and converts it to text during execution, and displays the text on the screen. Once the text is shown on the screen, the user may be able to save the text, listen to the text, review the text, or they can convert this text to Braille. If the text is converted to Braille, the user can print it, as shown in Figures 3 and 4.

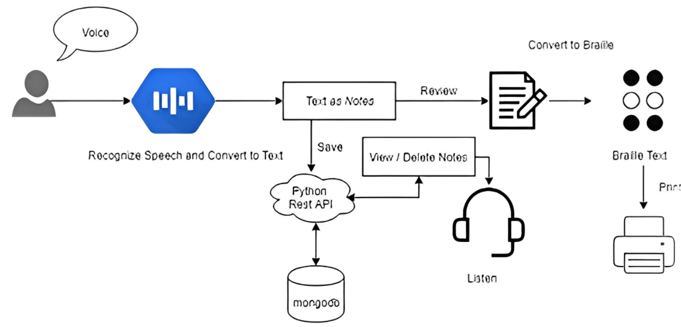

Figure 3. Architecture – Voice o Braille - [10]

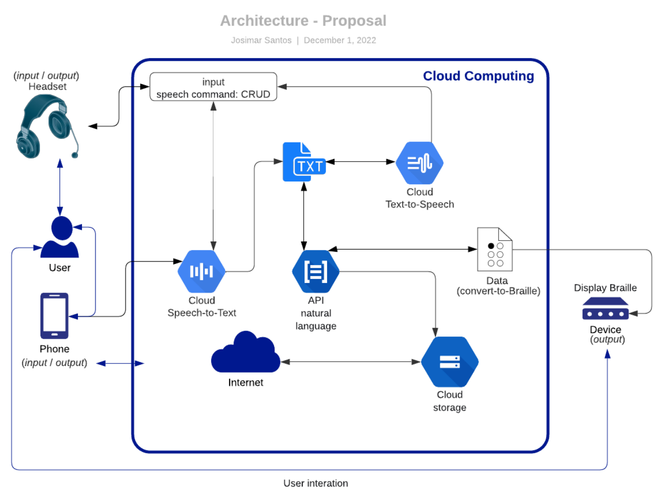

Figure 4. Architecture – Proposal – Authors - 2021

[9] mentions the usage of their project: The teacher will hold the student's hand and guide their finger on the matrix of raised dots. With the help of instructions and tactile interface, the student gradually learns the Braille code for the alphabet. Therefore, a single tutor can only teach one student at a time, and the tutor needs specialized training to teach Braille codes. This teaching method consumes a considerable amount of time when teaching a group of students, as each student needs to be taught individually. Hence, a concept of mass teaching system is proposed in this article to minimize the time required for teaching. This system has wireless features to avoid the use of cables so that students are not restricted to sitting in any specific position and at a particular distance. The tutor can type the character on the touchscreen keyboard, and this character is wirelessly transmitted to the entire student panel.

In the work of [10], beforehand, their proposal introduces the possibility of storage but not sharing of these Braille notations. Another aspect is that although it allows saving and listening to the saved text, we did not identify the possibility of editing or correcting the stored notations in MongoDB; if there is an error, the only available option is deletion. Another storage possibility, to adapt a modeling based on the studies of [10], could be tested in relation to the studies of [6], which report on low-latency and flexible cloud storage due to serverless architecture.

III. Architecture Proposal

In the work of [11], the development of an affordable Braille cell for the local context is discussed. In the initial iterations, the control system kept the solenoids active indefinitely to represent the Braille letters, but this caused excessive energy dissipation in the form of heat, resulting in high energy consumption and risk of damage. To solve this problem, the solenoids were programmed to be activated for only four seconds, allowing the interpretation of symbols and keeping them inactive for the remaining time.

The project faced difficulties in choosing the solenoids, as options with desirable characteristics were scarce in Mexico. In future iterations, hardware components that allow the miniaturization of the Braille cell and reduce energy consumption are analyzed.

In conclusion, the study demonstrated that it is possible to create an affordable Braille cell for the local context. The next steps involve reducing the size of the prototype and conducting tests with users to validate its usability, usefulness, and effectiveness. The work was partially funded by the Human-Computer Interaction Laboratory (IHCLab) of the School of Telematics at the University of Colima, Mexico. There was no need for ethical review for this research.

The study by [12] developed an assistive technology called PINDOTS, with a low-cost and easy-to-use Braille device for students with visual impairment and special education teachers. The technology consists of a six-dot Braille cell and six buttons for basic notation writing. A mobile application was created for teachers, allowing them to send exercises to the Braille device through a wireless connection. The device is capable of spelling three-letter words and has the option to emboss and record Braille dots. To improve the technology, they suggest increasing the number of spelled letters, adding contractions, including numbers, and testing with different types of visual impairments.

The studies by [10] and their architecture are very promising. The architecture of this work provides a vision that guides the structure of the Voice o Braille project, applying some differentials, such as:

Output: Developing a Braille display with the idea of [9], to read characters on Arduino and use it as an output terminal, provides more cost-effectiveness compared to the Braille printer in the author's architecture [10]. As protection for the device, a box, as well as the tactile points, can be 3D printed.

Cloud: Brokers for MQTT publish/subscribe, as well as storage and execution tools for the conversion API, are deployed in the cloud.

The first two works were included after the research and during the experiment, and it is important to cite them as a way to complement the content and add knowledge.

The next steps contextualize and specify how the third-party libraries found and tested in the architecture can assist in each process of the architecture.

The architecture will provide instructions through voice commands, where an "assistant" will speak the process and request commands for decision-making. The user will respond with the desired command to continue the flow. The conversion from voice to text will be done using the SpeechRecognition module [13] in version 3.8.1. The gTTS library [14] in version 2.3.0 will perform the text-to-speech conversion, with the help of the PyAudio library [15] in version 0.2.12. A Python API in version 3.9 will manage a text file with the extension ".txt", enabling resuming the file later to continue, asking the assistant to "read" the text in the file, and also allowing the user to edit parts of the text or delete it. Among these functions, it will also be possible to use the command "convert" to convert the text into a format that will be interpreted by the Braille display device. The API was tested and executed locally using Flask [16], where the results were verified as shown in Figures 5 and 6.

Figure 5. Architecture – API: text conversion screen – authors - 2022

In turn, this Braille display device will be structured with Arduino, specifically an ESP32, servomotors, pushbuttons, among other components for simulation. The display will "show" only one character at a time, and the control to move to the next character can be managed by buttons on the Braille display. For this purpose, we tested platforms like Tinkercad[1], where we did not obtain availability for the ESP32

[1]https://www.tinkercad.com/dashboard

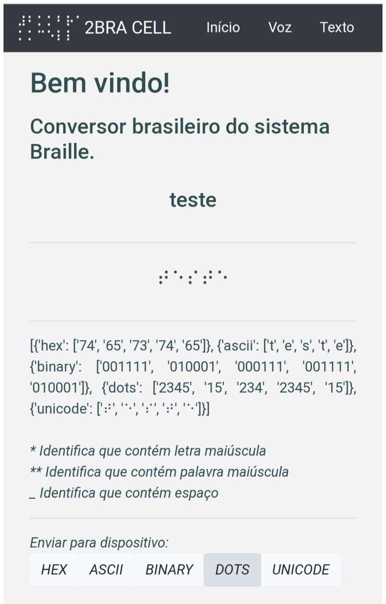

Figure 6. Architecture – API: converted text – authors - 2022

or simulation functionalities with internet communication due to security reasons.

IV. Preliminary Results

We obtained satisfactory results for initial tests with the Wokwi platform[2], where we managed to structure the device with the necessary components. However, some delays were noticed while using the libraries for both speech-to-text and text-to-speech conversions.

During the execution of pre-defined voice texts from the "assistant," if the audio was being generated for the first time, there was a longer delay for the "assistant to speak." This delay occurs because the strings or phrases mentioned to the user are directly executed in the functions of the libraries used. To overcome this problem, it is possible to store these audios after the first execution, as they are reusable, and play them each time they are called. However, this requires sufficient storage space for these files.

For voice capture from the user, even with the noise reduction applied in the library, some words could not be identified, or the capture was not successful. The words or phrases stored in text files on the server can be converted later and sent in JSON format over the internet to a broker, which can be published by Mosquitto[3]. The communication was tested through local simulation and the functionalities provided by the Wokwi platform.

One problem encountered regarding the return message in MQTT is that, since the API prepares a JSON with the text of the file requested by the user, we do not know the size of the text that will be converted. As a result, MQTT does not support sending messages above 268,435,455 bytes, approximately 260 MB. Therefore, a solution to use MQTT would be to limit the number of characters stored in each text file. The JSON is sent and received by the Arduino simulator, which handles the message received via MQTT and manages each character, separating them into sets of pins that will move the servomotors representing each Braille dot, as shown in Figure 7.

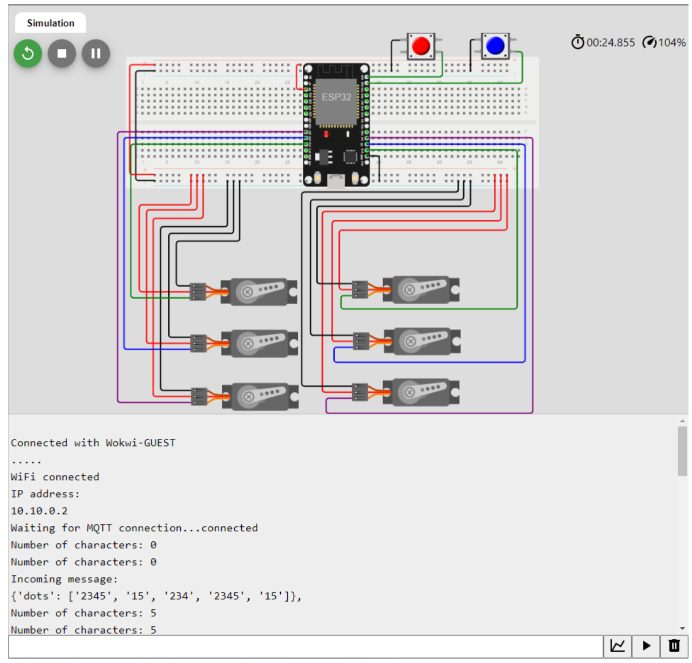

Figure 7. Architecture – Arduino ESP32 Simulator – authors - 2022

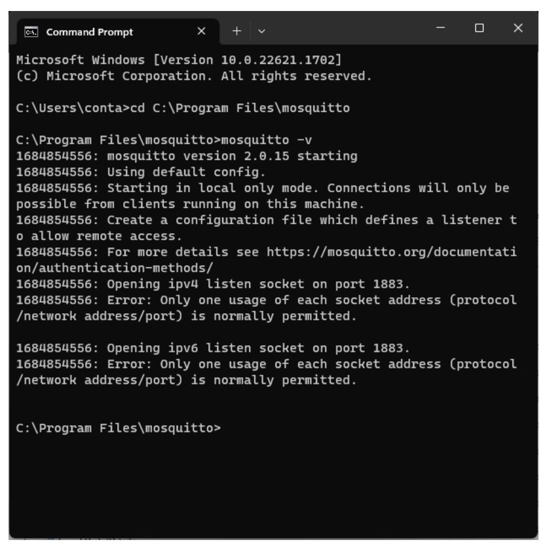

Figure 8. Mosquitto - version 2.0.15 starting

For communication between the server and the Braille dis-

[2]https://docs.wokwi.com/pt-BR/
[3]https://mosquitto.org/

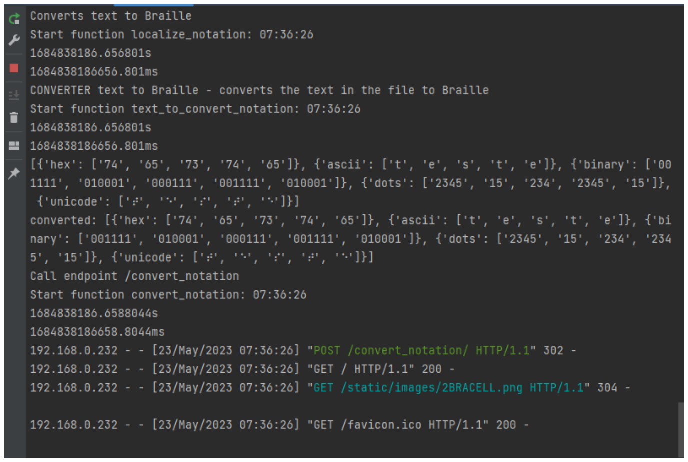

Figure 9. Architecture – API: send JSON data as a message MQTT to ESP32 simulator – authors - 2022

play, the Wokwi platform, link to the project in development[4], allowed local simulation and the use of available features, such as the use of the Mosquitto broker[5] (Figure 8) for MQTT communication. However, a problem was identified regarding the size of the message in MQTT since the API prepares a JSON (JavaScript Object Notation) (Figure 9 and Figure 6), which composes a data structure with the text of the file requested by the user, whose size can exceed the limit supported by MQTT. To solve this issue, we plan to limit the number of characters stored in each text file, ensuring that the messages stay within the supported limit.

The word or phrase is converted to Braille into a structure that will allow future work in various aspects. The database where the letters, numbers, and characters are converted through their respective representations is provided by a ss-braille.csv file[6] available on the Kaggle platform[7]. The file has been carefully updated to meet the Braille conversion needs, such as starting with uppercase letters, as well as words for writing in Brazilian Portuguese, e.g., c and ç. An example below shows the structure and types of conversion for the word “test”, sent by the API to the display via MQTT. The **hexadecimal** or hex represents the characters in hexadecimal according to

the ASCII table[8]. The **ascii** represents the characters according to the ASCII table[9]. The **binary** sends a set of 6 bits between 0 and 1 for each character of the word, which can be represented by 0 (zero) when the dot is down and 1 when it is raised. The **dots** represent a set of 6 numbers from 1 to 6 that indicate which dot will be raised. For example, if the set includes "12," dots 1 and 2 will be raised, while the numbers "345," absent from the set, represent the dots down.

[4]https://wokwi.com/projects/348875560981627476
[5]https://mosquitto.org/
[6]https://www.kaggle.com/datasets/josimarsts/ssbraille?select=ssbraille.csv
[7]https://www.kaggle.com/
[8]https://www.ascii-code.com/
[9]https://www.ascii-code.com/

The **unicode** represents each character in their respective Braille dots used for display. Regarding display, the article is directed towards an architectural solution that can benefit visually impaired individuals as well as educators and interested parties. So, in addition to enabling communication with the API through voice commands, the architecture provides an interface for conversion. The research project aims to integrate Cloud architecture and the MQTT protocol with IoT Arduino devices, along with the use of Braille for information display. The primary focus is not on commercializing the results but rather on promoting accessibility and inclusion for visually impaired individuals in the context of the Internet of Things. The purpose is to develop a technological solution that allows visually impaired individuals to interact more independently with IoT devices and services, providing them with access to relevant information and improving their quality of life. The research aims to contribute to the advancement of knowledge and offer a practical application that benefits society, especially those with special accessibility needs.

In the scenario where Arduino is simulated through Wokwi, the processing of the MQTT message return is handled by the Arduino simulator itself. It takes care of each character present in the received text and then separates them into groups of pins that, in turn, are responsible for controlling the servomotors. These servomotors represent the dots in a Braille cell, as shown in Figure 7.

Despite the challenges faced, the Wokwi platform provides a solid foundation for testing and adjusting the “Voice o Braille” project, allowing the visualization and simulation of the system’s operation before implementing it in a real environment. This approach is essential to identify and solve problems, as well as to improve user interaction, ensuring that the final solution is efficient and effective in including visually impaired individuals in social and educational forums.

V. Considerations

This work presents an architectural proposal to promote Braille literacy for visually impaired individuals. The architecture uses the MQTT protocol for communication between devices and enables interaction through both visual and tactile interfaces, with voice commands. The system is cloud-based, facilitating scalability and remote access. The Braille display device is structured with Arduino and components, and manufacturing is done with a 3D printer. The cloud implementation allows for data sharing and storage, with the possibility of future improvements, such as a specific converter for Brazilian Portuguese. The proposal aims to promote social and educational inclusion, democratizing access to knowledge and making education more accessible and equitable.

VI. Conclusion

The research presented an architecture that can benefit Braille literacy for visually impaired individuals. The related works on the context of the Braille system are recent, and there are still many discoveries that can be made regarding the developed studies. During the research, the study by Santiago and Bengtson (2020) presented a proposal for a Braille display that contextualizes the main idea of the architecture proposed in this work. However, the authors used a system installed on physical machines and suggested, as future work, the expansion of new possibilities for communication and data sending to the display.

The current project aims to offer MQTT solutions for a Braille display, following the Braille System's standards. This implementation is performed through a specific API developed for this architecture. The API's source code is available on GitHub, allowing other programmers to collaborate and improve the project. In addition, the architecture allows for the creation of other converters according to the specific needs of each project.

The central objective is to provide an MQTT solution for the Braille display that is compatible with Braille System standards, thus ensuring an appropriate experience for users who depend on this reading method. By making the source code available on GitHub, the project becomes open and collaborative, encouraging the participation of other programmers who can contribute with new ideas and improvements.

Another crucial point is the flexibility offered by the architecture, allowing other converters to be implemented according to the specific needs of each project. This makes the solution more versatile and adaptable to different contexts and specific requirements.

In conclusion, the proposed architecture seeks to meet the demands of MQTT communication for the Braille display, following the guidelines of the Braille System and promoting collaboration from the developers' community to continuously improve the project. By implementing new converters according to each case's specific needs, it is hoped that the solution can be widely applicable in different scenarios and contribute to a more inclusive and accessible experience for visually impaired individuals.

Although the MQTT protocol presented limitations regarding the message size, it can still be used as long as it does not exceed this limit. The proposed architecture needs to be analyzed, and depending on its usage, the cloud implementation may vary. However, as cloud computing favors scalability, it is possible to start with free or basic plans and expand as needed. As an additional benefit, the architecture allows tutors/teachers to dispense the need for prior knowledge in Braille and can enable the massive use of exercises for various students, instead of individualized classes, even if the teacher is not familiar with the Braille system. The architecture will enable visual and tactile interaction through the device and voice commands. An implementation of a specific converter in Braille for words in Brazilian Portuguese can be added to the architecture, considering that some letters and accents make the conversion particular in this country. A performance comparison between other protocols, in addition to MQTT, is also considered, following the same architecture suggested in this project to solve the issue of the message size limit/JSON sending. This comparative study would evaluate the performance of these two protocols in terms of message transmission efficiency and data processing in IoT applications, considering the message/JSON size restriction. This could provide valuable insights for selecting the most suitable protocol in specific scenarios, considering each application's performance and data transmission capacity requirements. The implementation codes will be improved and made available later.

References

[1] A. B. Possato and P. O. Monteiro, "Docentes de tecnologia da informação e comunicação," *Trabalho & educação*, vol. 29, no. 1, 2020.

[2] I. Machado Junior and J. P. Vece, "Contribuições da computação em nuvem como ferramenta pedagógica na educação superior," *The Journal of Engineering and Exact Sciences*, vol. 2, no. 3, pp. 92–106, 2016.

[3] C. D. . IBGE, "Censo demográfico 2010," https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf, jul 2010, (1) As pessoas incluídas em mais de um tipo de deficiência foram contadas apenas uma vez. (2) Inclusive as pessoas sem declaração dessas deficiências. (3) Inclusive a população sem qualquer tipo de deficiência.

[4] B. A. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research - A participant-observer case study," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 638–651, 2011. [Online]. Available: https://doi.org/10.1016/j.infsof.2010.12.011

[5] V. C. Lins and A. M. De Morais, "Simulação e avaliação de desempenho de uma blockchain para aplicações iot," *Gestão.org*, vol. 19, no. 2, pp. 169–183, 2021.

[6] C. G. Calancea, C.-M. Miluţ, L. Alboaie, and A. Iftene, "iassistme - adaptable assistant for persons with eye disabilities," *Procedia Computer Science*, vol. 159, pp. 145–154, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091931347X

[7] M. A. Spohn and F. D. S. Bonetti, "Análise de desempenho de uma arquitetura de fog computing para internet of things via estudos de caso," *Revista Brasileira de Computação Aplicada.*, vol. 11, no. 3, pp. 72–87, 2019.

[8] G. Santiago, C. Bengtson, D. Pino, C. Pendenza, and J. Santos, "Universidade federal de são carlos - ufscar," https://dcomp.ufscar.br/wp-content/uploads/2016/05/DComp-TR-002.pdf, 2020.

[9] S. Gandhi, B. Thakker, and S. Jha, "Braille cell actuator based teaching system for visually impaired students," in *Braille cell actuator based teaching system for visually impaired students*. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 1381–1385, cited By 3; Conference of 1st IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 ; Conference Date: 20 May 2016 Through 21 May 2016; Conference Code:125896. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015099644&doi=10.1109%2fRTEICT.2016.7808057&partnerID=40&md5=bd3c9373d89ba72173025f1571ee4839

[10] J. Pradeepkandhasamy, A. Priya, and P. Chellappan, "Voice o braille," *Materials Today: Proceedings*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785320407710

[11] O. I. Ramos-García, A. A. Vuelvas-Alvarado, N. A. Osorio-Pérez, M. Ruiz-Torres, F. Estrada-González, L. S. Gaytan-Lugo, S. B. Fajardo-Flores, and P. C. Santana-Mancilla, "An iot braille display towards assisting visually impaired students in mexico," *Engineering Proceedings*, vol. 27, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2673-4591/27/1/11

[12] D. Martillano, A. Chowdhury, J. Dellosa, A. Murcia, and R. Mangoma, "Pindots: An assistive six-dot braille cell keying device on basic notation writing for visually impaired students with iot technology," in *Proceedings of the 2018 2nd International Conference on education and e-learning*, ser. ICEEL 2018. ACM, 2018, pp. 41–47.

[13] Anthony Zhang (Uberi), *SpeechRecognition v.3.8,1*, , 2014. [Online]. Available: https://github.com/Uberi/speech_recognition/blob/master/reference/library-reference.rst

[14] Pierre Nicolas Durette and Contributors, *gTTS v. 2.3.0*, , 2014. [Online]. Available: http://gtts.readthedocs.org/

[15] Hubert Pham, *PyAudio 0.2.12*, , 2006. [Online]. Available: https://pypi.org/project/PyAudio/

[16] Armin Ronacher, *Flask 2.2.2*, Pallets, 2010. [Online]. Available: https://pypi.org/project/Flask/