ISSN: 1692-7257 - Volume 1 – Number 45 - 2025

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v1i45.3091

Prototype of an automated system for reading and control of smart meters

Prototipo de sistema automatizado para lectura y control de medidores de energía ingeligentes

Ing. Ronald Stivel Melo Cárdenas[1], **Msc. Luis Alfredo Rodríguez Umaña**[1]
Msc. Javier Eduardo Martínez Baquero[1], **Msc. Nelson Baquero Álvarez**[1]

[1] ***Universidad de los Llanos,*** *Facultad de Ciencias Básicas e Ingeniería, Ingeniería Electrónica, Villavicencio, Meta, Colombia*

Correspondencia: *jmartinez@unillanos.edu.co*

Received: *september 13, 2024.* ***Accepted:*** *december 17, 2024.* ***Published:*** *january 01, 2025.*

How to cite: *R. S. Melo Cárdenas, L. A. Rodríguez Umaña, J. E. Martinez Baquero, and N. Baquero Álvarez, "Prototype of an automated system for reading and control of smart meters", RCTA, vol. 1, no. 45, pp. 57–65, jan. 2025. Recovered from https://ojs.unipamplona.edu.co/index.php/rcta/article/view/3091*

Abstract: This article presents the implementation of a prototype of an automated system for the reading and control of variables in electric energy meters, being a fundamental response to contemporary and future challenges in the energy distribution sector. This approach not only addresses the current demands of the energy landscape, characterized by the need for efficiency and intelligent management, but also projects into the future, anticipating and facing the emerging challenges in the expansion and modernization of electrical infrastructures, optimizing remote reading and accurate control of variables. The results show a significant opportunity to improve operational efficiency and reduce costs in the electricity sector, being the adoption of communication protocols such as DLMS/COSEM and Modbus, supported by the implementation of telemetric systems on platforms such as Raspberry Pi or Orange Pi with Python programming.

Keywords: smart meters, modbus, smart grids, automated system.

Resumen: El presente artículo expone la implementación de un prototipo de sistema automatizado para la lectura y control de variables en medidores de energía eléctrica, siendo una respuesta fundamental ante los desafíos contemporáneos y futuros en el sector de distribución de energía. Este enfoque no sólo aborda las demandas actuales del panorama energético, caracterizadas por la necesidad de eficiencia y gestión inteligente, sino que también se proyecta hacia el futuro, anticipando y enfrentando los desafíos emergentes en la expansión y modernización de las infraestructuras eléctricas, optimizando la lectura remota y el control preciso de variables. Los resultados muestran una oportunidad significativa para mejorar la eficiencia operativa y reducir costos en el sector eléctrico, siendo la adopción de protocolos de comunicación como DLMS/COSEM y Modbus, respaldada por la implementación de sistemas telemétricos en plataformas como Raspberry Pi u Orange Pi con programación en Python.

Palabras clave: medidores inteligentes, modbus, redes eléctricas inteligentes, sistema automatizado.

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

1. INTRODUCTION

In a world increasingly focused on energy efficiency and sustainable resource management, smart meters have become essential elements for monitoring and have become essential elements in the monitoring and control of electrical energy [1], [2]. These devices, used in electricity distribution networks, enable accurate collection and analysis of consumption data, which in turn encourages more responsible energy use [3].

For their operation, these smart meters rely on specific communication protocols, such as DLMS/COSEM o Modbus [4], which control the interaction and exchange of information with other devices and systems. In this context, it has become essential to understand the characteristics, advantages and applications of these protocols to ensure the efficient and secure operation of information in electrical networks. [5], [6], [7], [8]. The fastest growing paradigm in utility distribution and management in the energy sector is the Smart Grid (SG) [9], [10], [11]. In a basic way, a smart grid represents an intelligent utility network that incorporates two-way communication and distributed computing capabilities across the different nodes of the network to improve control, efficiency, reliability and security [12]. SGs being IoT-based communication structures are susceptible to cyber-attacks due to their complex architecture and critical communication systems [13].

This vulnerability calls for thorough research in industry, government and academia to strengthen the security of smart grids. Smart grids leverage information technology for efficient energy delivery, facing significant security threats from weaknesses in communication technology [14]. Understanding the motivations behind attacks on smart grids, which now include both physical and cyber methods to manipulate energy consumption data, is crucial to improving their operational security [15]. As energy requirements increase, new challenges in energy management and demand control arise. Therefore, SGs are developed together with power electronics, sensor and metering technologies, enabling the grid to become more intelligent through communication management and intelligent control [16]. Smart meters, which are key components in smart grids, rely on standardized protocols for their management and control. [17]. Among the most widely used protocols is DLMS/COSEM which is a free and open application layer protocol used for a wide range of measurement applications, including remote meter reading, remote control and value-added services. Another widely used protocol is Modbus, which establishes common rules and formats for the transmission of information between a master device and one or more slave devices. The Modbus protocol is noted for its simplicity and ease of implementation, making it a popular choice in a wide range of industrial applications, such as process control, building automation, energy monitoring and more [18].

The article is structured in four sections, the present introduction which refers to the state of the art. The second section presents the methods and materials in three subsections: requirements, programming language and hardware selection, and firmware design. The third section presents the results and their analysis and finally the fourth section presents the conclusions reached.

2. METHODOLOGY

For the development of the automated system for reading and control of variables in the smart energy meter, the V-model methodology was used as a reference, which is a uniform procedure for the development of both ICT and software products. [19], since the development process proposed must have as its fundamental purpose a quality product that meets the specifications and responds to the expectations and needs of the client and/or end user [20], [21], [22], [23], [24], [25].

The description of the methodology is shown in Fig. 1. In the development process, different phases are established that are essential to guarantee the quality and functionality of the system.

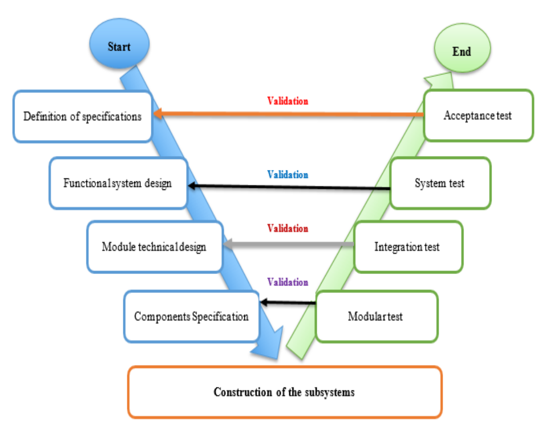

Fig. 1. *Methodology to be used in the development of the prototype*
Source: *Own elaboration*

2.1. Requirements

In this phase of the project, we proceed to the definition and detailed establishment of the fundamental requirements that will guide the development of the prototype, which are classified into four categories: user, functional, quality and implementation requirements. User Requirements focus on the specific needs and preferences of the end users of the system, ensuring an intuitive and satisfactory experience. Functional Requirements detail the specific functions and operations that the system must perform to fulfill its primary purpose. The Quality Requirements establish criteria and standards that will guarantee the efficiency, reliability and security of the system. Finally, the Implementation Requirements address practical and technical considerations essential for the successful execution of the project. This requirements definition process lays the foundation for a structured, results-oriented development aligned with identified key expectations and needs. The requirements established are:

Compatibility with three-phase Indirect Metering Smart Meters (RE1): The system shall be compatible with different models of three-phase smart meters operating by indirect metering, ensuring wide applicability of the Project.

Intuitive User Interface (RE2): An intuitive and user-friendly user interface is required so that users can configure and operate the system without difficulty. This requirement encompasses functional and quality aspects, ensuring that the system is accessible to all users, including those with little technical experience.

Effective Implementation of Communication Protocols (RE3): The system must effectively implement DLMS/COSEM or Modbus communication protocols to ensure interoperability and efficiency in data transmission between system devices.

Ability to Perform Remote Readings and Specific Controls in Real Time (RE4): It is essential for the system to allow remote readings and exercise specific controls over meters in real time, facilitating more efficient and reactive energy management.

Use of Specific Hardware for Telemetry Implementation (RE5): The system must be able to operate with specific hardware such as Raspberry Pi, Orange Pi, ESP32, or Atmega 2560 microcontrollers for the implementation of telemetry functions, which is crucial for data collection and transmission.

Development of Control Modules (RE6): Dedicated control modules should be developed to manage critical functions such as connection, switching and activation of capacitor banks, thus improving the efficiency and resilience of the system.

MQTT Protocol Integration (RE7): Integration of the MQTT protocol for data transmission to the information management platform is essential to ensure efficient and secure communication between the devices and the central system.

Robust Security Mechanisms (RE8): The system must incorporate robust security mechanisms to protect the integrity and privacy of transmitted data, a critical aspect in the current context of cyber security concerns.

Exhaustive Testing (RE9): Prior to final implementation, the system must be thoroughly tested to assess its functionality and performance, ensuring that it meets all established requirements and is' free of defects.

2.2. Programming language and hardware selection

In this critical phase of the project development, the task of comprehensively defining the telemetry system architecture is undertaken, covering both the hardware aspect and the choice of the programming language. The detailed design of the firmware emerges as an essential component to establish an effective communication with the meter and to facilitate the accurate transmission of data to the AOM ENERTEC platform.

The selection of the programming language was the subject of a meticulous pre-selection process that evaluated three outstanding options: Python, C# and C++. A complete description of the most salient features of each language is provided below, addressing aspects such as programming paradigm, syntax, performance, memory management, versatility, as well as the robustness of its community and ecosystem:

Programming Paradigm: Python is known for its flexibility, supporting object-oriented, imperative and functional paradigms. C++, on the other hand, focuses primarily on the object-oriented and imperative paradigm. C# strictly adheres to the object-oriented paradigm, which reflects its design and usage.

Syntax: Python stands out for its clear and concise syntax, which makes it easy to read and write. C++ has a more complex and detailed syntax, which can represent a steeper learning curve. C#, with a syntax similar to Java and C++, offers a well-defined object-oriented structure.

Performance: Python generally operates at a slower speed compared to C++, due to its interpreted nature. C++ is highly regarded for its high performance, as it compiles directly to machine code. C# offers solid performance, although it can be slightly slower than C++ due to its execution in the .NET runtime environment.

Typing: Python uses dynamic typing, which allows more flexibility during runtime. Both C++ and C# employ static typing, which requires data types to be explicitly declared.

Memory Management: Python and C# manage memory automatically, freeing developers from manual memory management. C++, in contrast, requires the programmer to manage memory explicitly, which allows for finer control, but increases complexity.

Versatility: Python is widely used in fields such as web development, data analysis and artificial intelligence. C++ excels in the development of embedded systems and high-performance applications. C# is often chosen for Windows application development, backed by strong support from Microsoft.

Community and Ecosystem: All three languages enjoy large and active communities. Python and C++ have a wide range of applications and libraries available. C# is supported by Microsoft and has a vast array of developer resources.

Typical Applications: Python is used in web development and data analysis; C++ in embedded systems, games and real-time systems; and C# in Windows application.

Platforms Supported: Python and C++ are cross-platform, although C++ is often preferred for native systems. C# is primarily used in Windows environments, although .NET Core has extended its applicability to other platforms.

Simultaneously, the choice of hardware was of equal importance, contemplating four development boards: Orange Pi, Raspberry Pi, ESP32 and Arduino Mega. The following is a detailed ranking that addresses critical aspects such as processing, storage, connectivity, support from the development community and affordability in terms of associated costs:

Architecture: Orange Pi uses ARM Cortex-A7, Raspberry Pi uses ARM Cortex-A72, ESP32 operates with Xtensa LX6, and ATMega 2560 uses AVR.

Processor speed: Orange Pi reaches 1.6 GHz, Raspberry Pi reaches 1.5 GHz, ESP32 operates at up to 240 MHz, and ATMega 2560 runs at 16 MHz.

RAM: Orange Pi has 1 GB DDR3, Raspberry Pi offers versions with 2 GB, 4 GB or 8 GB LPDDR4, ESP32 has 520 KB SRAM, and ATMega 2560 has 8 KB SRAM (of which 2.5 KB is available to the user).

Storage: Orange Pi and Raspberry Pi have microSD card slots, ESP32 can have up to 16 MB Flash, and ATMega 2560 includes 256 KB Flash and 4 KB EEPROM.

Connectivity: Orange Pi provides 10/100 Ethernet and WiFi, Raspberry Pi has Gigabit Ethernet and WiFi, ESP32 includes WiFi and Bluetooth, and ATMega 2560 has no Ethernet, with several models offering options with/without WiFi.

USB Ports: Orange Pi offers 3 USB 2.0, Raspberry Pi has 2 USB 3.0 and 2 USB 2.0, ESP32 has 1 USB 1.1 (OTG) and 1 USB 2.0, and ATMega 2560 has 4 USB 2.0.

GPIO Pins: Orange Pi and Raspberry Pi have 40 GPIO pins, ESP32 has 36, and ATMega 2560 offers 54 GPIO pins, of which 14 are PWM.

Operating System: Orange Pi and Raspberry Pi are Linux compatible (Raspberry Pi also supports Windows 10 IoT Core), ESP32 works with Free RTOS among others, and ATMega 2560 does not have a built-in operating system, being programmed through an IDE.

Community and Support: Orange Pi has a moderate community and support, while Raspberry Pi, ESP32 and ATMega 2560 enjoy a large community and support.

Approximate Price: Orange Pi and Raspberry Pi are moderately priced, varying in the case of Raspberry Pi depending on the model; ESP32 and ATMega 2560 are considered affordable.

These strategic decisions in the system architecture lay the foundation for a coherent and efficient development. The synergy between the selected programming language and hardware will play a key role in the successful implementation of telemetry for the management of three-phase indirect metering smart meters.

2.3. Design of the firmware

In the firmware design phase, a detailed analysis of the information flow was carried out, from the moment it is collected by the meter until it is sent to the AOM platform of Enertec Latinoamérica S.A.S. This process is clearly visualized in the flow diagram presented in Fig. 2. The design of this flow seeks to guarantee efficient, accurate and secure communication, ensuring that the data collected by the meter is optimally transmitted to the corresponding platform. Fig. 2 serves as a graphical representation that illustrates the essential stages and connections in this information flow, facilitating a visual understanding of the system to be implemented.

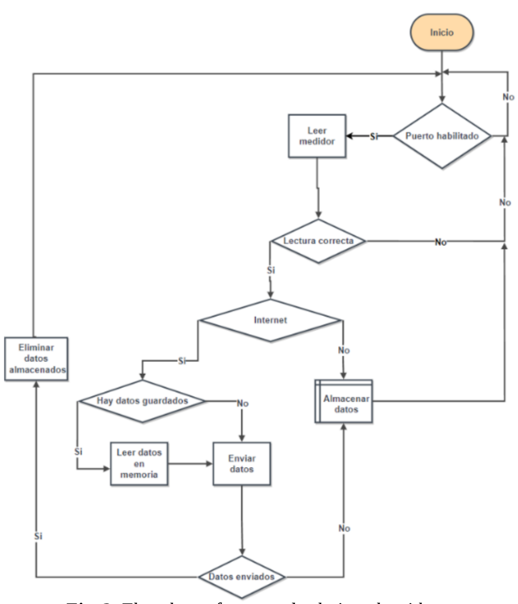

Fig. 2. *Flowchart of proposed solution algorithm.*
Source: *Own elaboration*

In order to confer a systematic behavior to the firmware, it was decided to adopt the use of state machines. This approach brings significant value by providing a logical and visual structure that facilitates the management of states and transitions in the system. In the field of firmware programming and development, the state machine models the behavior of the system by dividing it into discrete states and defining the transitions between them, thus generating a more organized and understandable code.

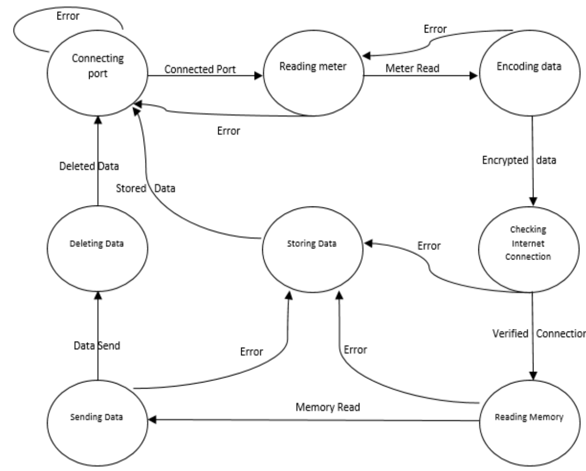

Fig. 3. *Proposed state machine diagram for the solution.*
Source: *Own elaboration*

The main virtue of state machines lies in their ability to their ability to effectively represent complex and dynamic situations. In the telemetry project, each state corresponds to a specific stage of the process, such as connection to the meter, data reading, information encoding, among others, as can be seen in Fig. 3 The transitions between states reflect how the system responds to particular events or conditions.

3. RESULTS

The DLMS/COSEM protocol was chosen as the most appropriate option to carry out the process of reading and and control process of the smart energy meter. This application layer protocol offers an open and free approach, setting standards for various metering applications, including remote reading and remote control. Its flexibility and ability to describe interfaces for various objects, such as voltage and current, plus the fact that it is a bidirectional and end-to-end implementation protocol, together with its high access security, communication channel protection and encryption of data transmitted during all stages, made it a solid choice for the efficient management of this type of smart meters, standing out in these aspects against the Modbus protocol. The standardization of DLMS/COSEM, supported by the DLMS Users Association (DLMS UA) and IEC 62056 standards, ensured interoperability and adoption in the technological context of ENERTEC Latinoamérica S.A.S. The implementation of this protocol facilitated accurate reading and effective control, thus contributing to a more advanced energy management and operational improvement of the services provided by the company.

After evaluating the characteristics of the different development boards described in the materials and methods section it was determined that the Raspberry Pi, supported by a USB to DB9 converter based on the CH-340 Chip which was selected due to low cost, was the appropriate choice to facilitate the remote control and monitoring of the smart power meter. The allocation of points according to the selection criteria is summarized in Table 1.

Each criterion was rated on a scale of 1 to 5, where 1 represents low performance and 5 represents high performance. The Raspberry Pi, especially excelled in computing power, ease of use, connectivity, compatibility, I/O and ports, community and support, as well as cost, which contributed to its selection as the most balanced and efficient choice for the project.

Table 1: *Comparison of development boards*

Criteri	OPi	RasPi	ESP32	ATMega 2560
Computing power	3	5	2	1
Ease of Usability	3	5	4	4
Connectivity	4	5	4	2
Compatibility	3	5	4	3
I/O and Ports	4	5	4	5
Community and Support	3	5	4	4
Price	4	3	5	5
Versatility o projects	3	5	4	4
Total	27	38	31	28

Source: *Own elaboration*

In the selection of the programming language, Python was chosen because of the multiple benefits it offers for development. The evaluation of Python was carried out along with the other shortlisted languages, using predefined criteria where the score The evaluation of Python was carried out along with the other pre-selected languages, using predefined criteria where the score, as in the hardware selection, ranged from 1 (lowest) to 5 (highest). The choice of Python was based on its total score of 30 as seen in Table II, standing out as the most suitable choice compared to C++ and C#. This result highlights the distinctive features of Python, such as its simplicity and code readability, which facilitates both rapid development and program comprehension. In addition, the extensive availability of libraries and frameworks in Python provides valuable tools for data manipulation and implementation of communication protocols, such as DLMS/COSEM.

Table 2: *Comparison of programming languages*

Criteria	Phyton	C++	C#
Ease of learning	4	3	4
Performance	2	5	4
Memory management	4	2	4
Versatility	5	4	4
Community and Ecosystem	5	5	4
Typical applications	5	4	4
Supported platforms	5	4	4
Total	**30**	**27**	**28**

Source: *Own elaboration*

The successful implementation of the prototype designed to meet specific requirements that enable essential telemetry in the control and monitoring of smart power meters is an outstanding breakthrough in the convergence of hardware, firmware, and effective communication. This paper examines the strategic combination of the Raspberry Pi as the core platform, the USB-to-DB9 converter as the key connectivity interface, and the Python programming language in firmware development. The robustness and efficiency of this combination in terms of data transmission and synergistic interaction with the smart meter is addressed in detail.

Fig. 4 complements this analysis by graphically visualizing the flow of information in the project, providing a clear representation of the harmonious integration of key components in the system implementation.

This study highlights the achievements, challenges overcome and significant contributions of the proposed solution, providing valuable insights for future research in the field of smart meter telemetry and management.

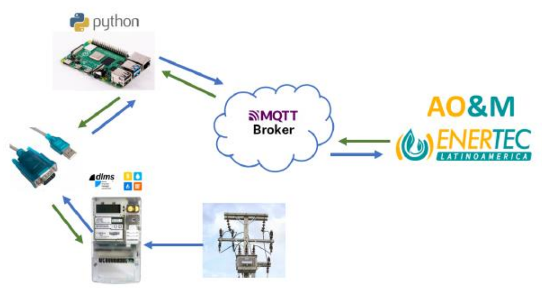

Fig. 4*. Information flow diagram in the project*
Source: *Own elaboration*

The project was successfully implemented in two medium voltage networks, the physical implementation of the prototype is shown in Fig 5 below.

Fig. 5. *Functional prototype installed in Mitú (Vaupés)*
Source: *Own elaboration*

After conducting an evaluation of the automated system, addressing key aspects such as data reading, control responsiveness and remote accessibility, the system demonstrated reliable performance in smart meter data collection, with optimal response times for control operations. In addition, the ability to remotely access and monitor the system has been successfully validated, confirming the effectiveness of the prototype in telemetry environments. Images of the AO&M platform dashboard through which monitoring is performed can be seen in Fig. 6 and Fig. 7 below.

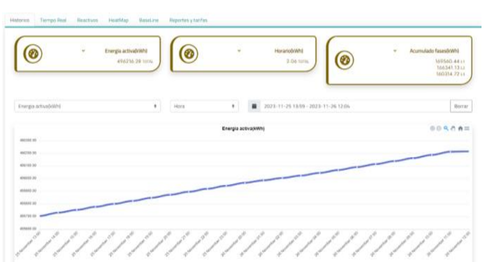

Fig. 6. *Hourly display of the installed system*
Source: *Own elaboration*

Fig. 7. *Consumption in a 3-month period*
Source: *Own elaboration*

4. CONCLUSIONS

The adoption of communication protocols such as DLMS/COSEM and Modbus, supported by the implementation of telemetry systems on platforms such as Raspberry Pi or Orange Pi with Python programming, has proven to be an effective and versatile approach.

This context of automation not only responds to a cyclical need, but also drives the continuous evolution of the electricity sector towards more efficient and sustainable practices, aligned with the changing demands of society and technology.

The limitations associated with manual smart meter management, such as inefficiency, operating costs and human error, underscore the critical need for automated solutions. The successful prototype developed offers an answer to these challenges, providing a solid foundation for the modernization of the electrical infrastructure in Colombia.

The implementation of telemetry technologies at an early stage in Colombia presents significant opportunities to improve operational efficiency and reduce costs in the electricity sector. In addition, remote monitoring capability, accurate data collection and system adaptability to future changes contribute to sustainable development and a comprehensive improvement of energy services in the country. This project lays the foundation for future innovations in energy management, supporting the continuous evolution and modernization of the Colombian electricity sector.

ACKNOWLEDGEMENT

The authors thank to Universidad de los Llanos and Enertec Latinoamérica S.A.S. for their support in the development of this project.

REFERENCES

[1] P. Sospiro, L. Amarnath, V. Di Nardo, G. Talluri, and F. H. Gandoman, “Smart grid in china, EU, and the US: State of implementation,” Energies, vol. 14, no. 18, pp. 1–15, 2021, doi: 10.3390/en14185637.

[2] W. M. Giral Ramírez, H. J. Celedón Flórez, E. Galvis Restrepo, and A. T. Zona Ortiz, “Redes inteligentes en el sistema eléctrico colombiano: Revisión de tema,” Tecnura, vol.

21, no. 53, pp. 119–137, 2017, doi: 10.14483/22487638.12396.
[3] Y. Kabalci, "A survey on smart metering and smart grid communication," Renew. Sustain. Energy Rev., vol. 57, pp. 302–318, 2016, doi: https://doi.org/10.1016/j.rser.2015.12.11.
[4] A. Sahu and A. Goulart, "Implementation of a C-UNB Module for NS-3 and Validation for DLMS-COSEM Application Layer Protocol," in 2019 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR), 2019, pp. 1–6. doi: 10.1109/CQR.2019.8880075.
[5] P. Matoušek, "Analysis of DLMS Protocol," 2017. [Online]. Available: https://www.fit.vut.cz/research/publication-file/11616/TR-DLMS.pdf
[6] L. J. Weith, DLMS / COSEM protocol security evaluation. Eindhoven: Eindhoven University Technology, 2014. [Online]. Available: https://pure.tue.nl/ws/portalfiles/portal/46962657/773263-1.pdf
[7] H. Mendes, I. Medeiros, and N. Neves, "Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework," in 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2018, pp. 179–184. doi: 10.1109/DSN-W.2018.00060.
[8] D. L. Jiménez, J. A. Rea, P. R. Muñoz, G. E. Vizuete, L. J. Latacunga, and C. A. Iza, "Diseño y Construcción de un Medidor de Energía Eléctrica Domiciliar," Rev. Técnica "energía," vol. 20, no. 1, pp. 82–92, 2023, doi: 10.37116/revistaenergia.v20.n1.2023.573.
[9] J. J. Moreno Escobar, O. Morales Matamoros, R. Tejeida Padilla, I. Lina Reyes, and H. Quintana Espinosa, "A Comprehensive Review on Smart Grids: Challenges and Opportunities," Sensors, vol. 21, no. 21. 2021. doi: 10.3390/s21216978.
[10] C. Wietfeld, A. A. Cardenas, H.-H. Chen, P. Popovski, and V. W. S. Wong, "Smart Grids," IEEE Wirel. Commun., vol. 24, no. 2, pp. 8–9, 2017, doi: 10.1109/MWC.2017.7909091.
[11] M. Farmanbar, K. Parham, Ø. Arild, and C. Rong, "A Widespread Review of Smart Grids Towards Smart Cities," Energies, vol. 12, no. 23. 2019. doi: 10.3390/en12234484.
[12] D. D. Vyas and H. N. Pandya, "Advance Metering Infrastructure and DLMS/COSEM Standards for Smart Grid," Int. J. Eng. Res. Technol., vol. 1, no. 10, pp. 1–5, 2012, [Online]. Available: https://www.ijert.org/published-issue-archive
[13] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, "Cyber-security in smart grid: Survey and challenges," Comput. Electr. Eng., vol. 67, pp. 469–482, 2018, doi: https://doi.org/10.1016/j.compeleceng.2018.01.015.
[14] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," Comput. Networks, vol. 169, p. 107094, 2020, doi: https://doi.org/10.1016/j.comnet.2019.107094.
[15] C. Barreto and A. A. Cárdenas, "Impact of the Market Infrastructure on the Security of Smart Grids," IEEE Trans. Ind. Informatics, vol. 15, no. 7, pp. 4342–4351, 2019, doi: 10.1109/TII.2018.2886292.
[16] D. B. Avancini, J. J. P. C. Rodrigues, S. G. B. Martins, R. A. L. Rabêlo, J. Al-Muhtadi, and P. Solic, "Energy meters evolution in smart grids: A review," J. Clean. Prod., vol. 217, pp. 702–715, 2019, doi: https://doi.org/10.1016/j.jclepro.2019.01.229.
[17] J. Zheng, D. W. Gao, and L. Lin, "Smart Meters in Smart Grid: An Overview," in 2013 IEEE Green Technologies Conference (GreenTech), 2013, pp. 57–64. doi: 10.1109/GreenTech.2013.17.
[18] Y. Fang, X. Han, and B. Han, "Research and Implementation of Collision Detection Based on Modbus Protocol," vol. 6, no. 1, pp. 91–96, 2013, [Online]. Available: http://www.jestr.org/index.php?option=com_content&view=article&id=28&Itemid=68
[19] E. García Sánchez, O. Vite Chávez, M. Á. Navarrete Sánchez, and M. Á. García Sánchez, "Metodología para el desarrollo de software multimedia educativo MEDESME," Rev. Investig. Educ. 23, vol. 23, no. Julio-Diciembre, pp. 217–226, 2016.
[20] A. F. Díaz, B. Prieto, J. J. Escobar, and T. Lampert, "Vampire: A smart energy meter for synchronous monitoring in a distributed computer system," J. Parallel Distrib. Comput., vol. 184, p. 104794, 2024, doi: https://doi.org/10.1016/j.jpdc.2023.104794.
[21] J. G. Fierro Mendoza, J. A. Asato España, J. B. Molina Castro, J. G. Delgado Núñez, and E. Noriega Vaca, "PROPUESTA METODOLÓGICA PARA VALIDAR LA FUNCIONALIDAD DE SOFTWARE EN SISTEMAS EMBEBIDOS," Pist. Educ., vol. 38, no. 122, pp. 156–177, 2016, [Online]. Available: https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/689

[22] S. M. Velásquez Restrepo, J. D. Vahos Montoya, M. E. Gómez Adasme, E. J. Restrepo Zapata, A. A. Pino Martinez, and S. Londoño Marín, “Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software,” Rev. CINTEX, vol. 24, no. 2, pp. 13–23, 2019, doi: 10.33131/24222208.334.

[23] M. V. Estrada-Velasco, J. A. Núñez-Villacis, P. R. Saltos-Chávez, and W. C. Cunuhay-Cuchipe, “Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software,” Rev. Científica Dominio las Ciencias, vol. 7, no. 4, pp. 434–447, 2021, doi: 10.23857/dc.v7i4.2429.

[24] B. Molina Montero, H. Vite Ceballos, and J. Dávila Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” Espirales. Rev. Multidiscip. Investig. científica., vol. 2, no. 17, 2018, doi: 10.31876/re.v2i17.269.

[25] J. P. Zumba Gamboa and C. León Arreaga, Cecibel Alexandra., “Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software.,” INNOVA Res. J., vol. 3, no. 10, pp. 20–33, 2018, [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=6777227