23

Artícle **Journal of Technological Development**

June, 2023 Vol.7 No.19 23-29

# Virtual instrumentation on CAN network for process monitoring Red CAN de instrumentación virtual para monitoreo de procesos

# Instrumentación virtual en red CAN para monitoreo de procesos Red CAN de instrumentación virtual para monitoreo de procesos

SANCHEZ-QUINTAL, Ricardo Jesús*†, UC-RIOS, Carlos Eduardo, DURAN-LUGO, Juan Miguel and LUGO-DEL TORO, Julio Francisco

*Universidad Autónoma de Campeche-Facultad de Ingeniería*

ID 1st Author: *Ricardo Jesús, Sánchez-Quintal* / **ORC ID**: 0009-0003-8437-931X, **CVU CONAHCYT ID**: 786068

ID 1st Co-author: *Carlos Eduardo, Uc-Rios* / **ORC ID**: 0000-0003-1321-019X, **CVU CONAHCYT ID**: 88147

ID 2nd Co-author: *Juan Miguel, Duran-Lugo* / **ORC ID**: 0000-0001-7179-4301, **CVU CONAHCYT ID**: 653065

ID 3rd Co-author: *Julio Francisco, Lugo-Del Toro* / **ORC ID**: 0009-0000-2014-3154, **CVU CONAHCYT ID**: 1299953

**DOI:** 10.35429/JTD.2023.19.7.23.29

Received March 26, 2023; Accepted June 28, 2023

## Abstract

Virtual instrumentation has experienced significant advancements in recent decades, providing efficient and reliable solutions for the connection and communication of electronic devices in various applications. This article explores the most relevant developments in the field, focusing on the integration of Controller Area Network (CAN) networks and microcontrollers in distributed sensor projects. The article also presents a case study in which a CAN network is implemented along with the ESP-Wroom32 microcontroller in a distributed sensor project for alcoholic beverage production. The sensors used in each stage of the process, such as ultrasonic, load, water level, and motion sensors, are described, highlighting the importance of redundancy in monitoring critical parameters to ensure safety and proper process operation.

**CAN, Virtual Instrumentation, LabView, Esp- Wroom**

## Resumen

La instrumentación virtual ha experimentado avances significativos en las últimas décadas, proporcionando soluciones eficientes y confiables para la conexión y comunicación de dispositivos electrónicos en diversas aplicaciones. Este artículo explora los desarrollos más relevantes en el campo, centrándose en la integración de redes de Área de Controlador (CAN) y microcontroladores en proyectos de sensores distribuidos. El artículo también presenta un estudio de caso en el que se implementa una red CAN junto con el microcontrolador ESP-Wroom32 en un proyecto de sensores distribuidos para la producción de bebidas alcohólicas. Se describen los sensores utilizados en cada etapa del proceso, como sensores ultrasónicos, de carga, de nivel de agua y de movimiento, destacando la importancia de la redundancia en la monitorización de parámetros críticos para garantizar la seguridad y el correcto funcionamiento del proceso. Se obtiene la manera adecuada de organizar los datos obtenidos en los sensores, la transmisión de estos paquetes de datos y los criterios lógicos para activar los actuadores del proceso.

**CAN, Instrumentación Virtual, LabView Abstract**

**Citation:** SANCHEZ-QUINTAL, Ricardo Jesús, UC-RIOS, Carlos Eduardo, DURAN-LUGO, Juan Miguel and LUGO-DEL TORO, Julio Francisco. Virtual instrumentation on CAN network for process monitoring Red CAN de instrumentación virtual para monitoreo de procesos. Journal of Technological Development. 2023. 7-19: 23-29

---

* Correspondence to Author (e-mail: rjsanche@uacam.mx)

† Researcher contributing first author

## Introduction

The term virtual instrumentation can be coined from the development of the general purpose interface bus (GPIB) in 1970 by the Hewlett-Packard company, which allowed, through serial communication, to connect electrical devices and measurement equipment; over time it became a worldwide standard, widely used in the electronics and measurement industry. It is estimated that in 2009 there were as many as 10,000 different types of instruments ranging from low-cost voltmeters to high-performance network analysers (Jia, 2009).

The GPIB is a network with the ability to connect up to 15 electronic devices through a 24-wire cable, the system has a main controller in charge of coordinating the communication between the different devices, it also stands out for its communication speed that ranges between 1.5 and 2 megabytes per second.

Nowadays, the diversification of instrumentation and control systems presents us with various options for integrating electronic devices, one of these options is the CAN network (Controller Area Network). CAN (Controller Area Network) contains a specific high-level communication protocol that was developed by BOSCH in the 1980s for use in the automotive industry (Ishak, 2019), but has been adopted in numerous applications outside this industry; it has also been used in virtual instrumentation and real-time control systems.

The use of CAN in virtual instrumentation allows the transmission of information between sensors and actuators in an efficient and reliable manner (Gharavi, 2020), resulting in increased accuracy and speed in data acquisition. It is a high-speed serial network that has been shown to be cost-effective, efficient and very economical. In a very simple way, sensors and actuators can be connected using a twisted pair cable that can reach speeds of 1 MBit/sec with 40 devices simultaneously (H. F. Othman, April 2006).

## Hardware used

Embedded systems design is based on the programming of microprocessors and real-time operating systems. To integrate this knowledge into the project, the ESP-Wroom32 microcontrollers with node function in the network were used.

This microcontroller consists of a CPU (microprocessor) and two wireless communication modules WIFI and Bluetooth. The microprocessor has two processing cores with a processing speed of 240 MHz, which allows it to include real-time operating systems (Technical documents: Espressif Systems, 2023). The processor peripherals facilitate connection to a variety of external interfaces such as:

- SPI
- I2C
- Ethernet
- SD cards
- Touch interfaces

The module contains 4 MB of flash memory and only 38 pins (so as not to increase the size of the module), as shown in Figure 1. This device has become so popular due to its low cost and the possibility of being programmed by a variety of IDE interfaces such as the Arduino.

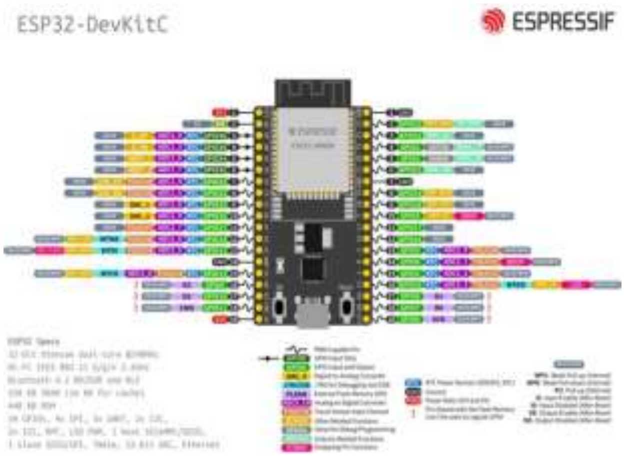

**Figure 1** ESP Wroom32 (ESP32-DevKitC V4 Getting Started Guide, 2023)

Another device used in this project is the mcp2512 CAN controller, which is capable of transmitting and receiving remote source data strings, has reception masks and 6 different filters that are used to reject irrelevant messages, relieving the microprocessor of the computational burden messages, relieving the microprocessor of computational load. This device is connected and configured via the SPI interface.

The logic control includes interrupt pins that are provided to increase the flexibility of the system. It has a general purpose interrupt pin which can be configured to detect received messages, invalid messages, valid message, etc.

The use of this pin is optional, as is the use of the status register, these pins are configured by the SPI interface (Microchip, 2018). The connection of the mcp2515 to the ESP Wroom32 microcontroller is done according to the diagram shown in Figure 2.

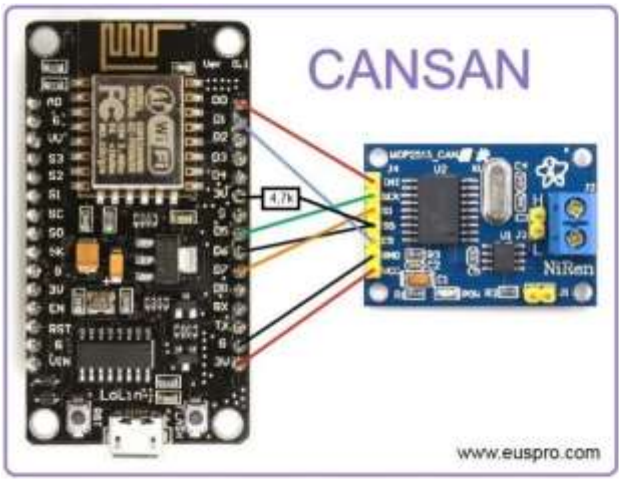

**Figure 2** Connection of the mcp2515 (Arduino, 2023)

For the joint operation of the Wroom32 microcontroller and the Mcp2515 CAN controller, the library published by Cory J. Fowler (Fowler, 2023) is used, thus establishing the sending of data packets from the distributed sensors to the HMI interface via the CAN instrumentation network.

## Digital sensors

There are different ways to obtain I/O data from a sensor, a common device for this is to use a stand-alone node such as the MCP25050 (Esro, 2009), however, for cases where processing of the acquired signals is required, it is necessary to use a microcontroller capable of performing the interaction with the specific sensor.

An example of the above is the case of ultrasonic sensors, where the duration of an input pulse is measured, the duration of an input pulse, and this cannot be done by the MCP25050 stand-alone node, which, although they are CAN nodes with input/output pins, analogue-digital converters, are not capable of processing signals as the microcontroller does.

The sensors implemented in this project are:

- HC-SR04 Short range ultrasonic sensor (Morgan, 2014).
- HX711 Sensor specialised in reading and processing load cells (Al-Mutlaq, 2023).
- HW-038 Water level sensor (Ashari, 2022).
- HCSR501 Infrared Motion Sensor (Wahyuni, 2021)

Most of the libraries for these sensors are directly available in the Arduino IDE application or on the network, this allows for quick implementation of all CAN network elements.

## Methodology

For the implementation of this project it is proposed to use a CAN network, a graphical HMI interface developed in LabView and a CPU that will display the graphical interface board. In order to be able to measure different physical variables, different digital sensors connected to the esp-wroom32 microcontroller will be used. These microcontrollers are networked using the Mcp2515 CAN controller.

For the application of this network in a real case, the final stage of the production process in a distillery is selected (Figure 3). This stage can be subdivided into three main parts:

1. Last Distillation
2. Hydration and resting
3. Bottling and labelling.

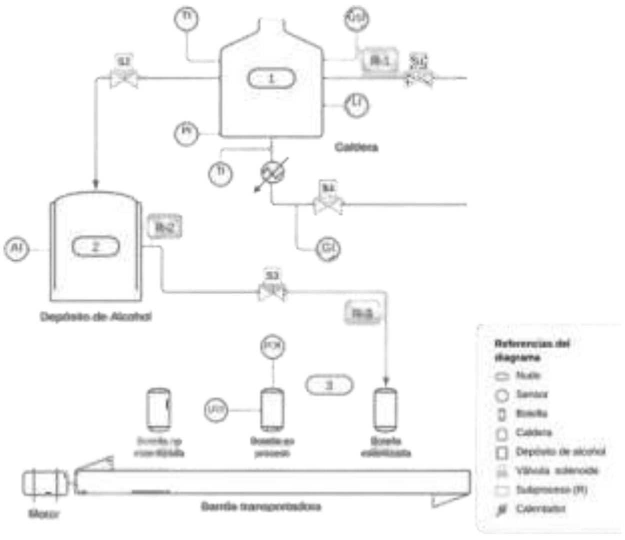

**Figure 3** Proposed process

SANCHEZ-QUINTAL, Ricardo Jesús, UC-RIOS, Carlos Eduardo, DURAN-LUGO, Juan Miguel and LUGO-DEL TORO, Julio Francisco. Virtual instrumentation on CAN network for process monitoring Red CAN de instrumentación virtual para monitoreo de procesos. Journal of Technological Development. 2023.

**Final Distillation**

Due to the fact that the value chain of the production of alcoholic beverages can have more than one distillation process, in this stage a hot infusion is carried out by distillation, with a very slow heating following a traditional One-Shot process in the London Dry, each distillation is carried out in 12 hours, extracting all the essence and achieving the desired aroma and flavour.

This is a critical stage due to the risk it represents and determines the quality of the product, for these reasons it was decided to implement redundancy in the measurement of the liquid level, i.e., it is ensured by three sensors the proper level before sending the command to ignite the boiler. In addition, after sending the ignition command, it is necessary to verify that the boiler is actually ignited the boiler has actually been ignited.

To check for proper liquid level, the information provided by the ultrasonic level sensor (USI), the load cell (PI) at the base of the tank and the level sensor (LI) is taken into account, thus reducing the likelihood of a high-risk event such as ignition without the proper volume of liquid.

A temperature rise by the TI sensor in the boiler chamber as well as the absence of gas provided by the GI sensor must be met to check ignition.

**Hydration and standing**

Once the liquid has been distilled, it is left to settle before hydration and once hydrated, it is left to settle again before bottling. The alcohol sensor (AI) is installed in this section to verify that there are no leaks in the hydration tank.

**Bottling and labelling**

In the last stage of the process, the bottles are disinfected with UV light. An ultraviolet radiation sensor (UVI) is used to verify that the UV light is switched on, which must detect a constant level of UV light and thus guarantee the disinfection process.

Additionally, a proximity sensor (PIR) is used to check the position of the bottles in the packaging and to avoid collisions.

**Network implementation**

The CAN network is a bidirectional network, which in this project was configured so that the nodes only send the information from the sensors to the central node (Node C), given the robustness of the CAN network it is possible to use the central node to send control signals to the actuators in the network, these control signals can depend on the decisions programmed in each node or be directly controlled by the CPU.

The capabilities of the CAN network allow:

1. To know the status of each node, if it is operational or presents an error.

2. The readings of the sensors and the operation of the actuators connected

3. To verify the data traffic on the network, loss of information packets; among others. Although in this network each node has only one sensor, it is possible to connect more than one sensor to each node.

The block structure of the network is shown in Figure 4.

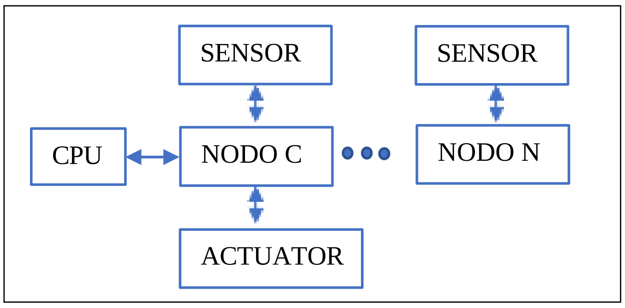

**Figure 4** Communication schem

When implementing this network, we found 3 key points for the good performance of the CAN network in this virtual instrumentation application in the distillery. The first is to organise the information received from the sensors to the central node in an efficient way. The second is the way in which the central node will send the information received from the sensors to the CPU. The third is the implementation of the selection logic for checking the critical variables. In this project, the different ways of performing these processes were analysed and those that showed the most advantage were selected.

## Results

For the development of the HMI interface (Figure 5) where the information obtained from the sensors will be displayed, it was necessary to send the information through the serial port from the central node to the CPU. This information, regardless of the transmission medium, has to be structured in order to be interpreted by the HMI interface.

Three options are found to organize sensor information: generate an own structure or use one of the two most known data organization methods, which are JSON (Mora-Castillo, 2016) and XML (Skogan, 1999). In order to be able to work as a team and reduce development time, the creation of an in-house structure was discarded and the performance of each of the other two options was reviewed: JSON was chosen because it presents better overall performance in speed and consistent use of resources (Nurseitov, 2009). It is a widely used format, for working with database and transmission of information via web (Lv, 2019), in addition to having more ease of structuring, reading and processing data in LabView. This implementation can be seen in Figure 6.

**Figure 5** HMI interface

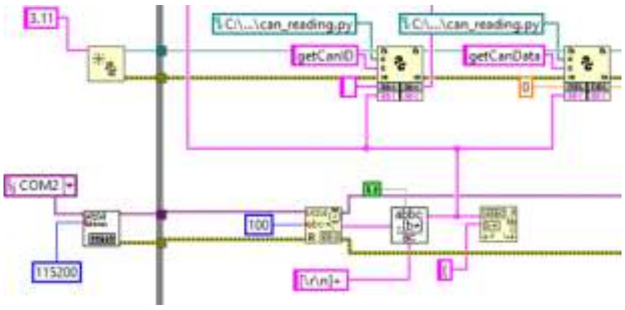

**Figure 6** Reading from the central node

To receive messages through the serial port, a simple algorithm was developed to improve performance for this particular project, seeking compatibility with data packets in JSON format.

This communication consists of:

- Open the VISA serial communication
  - Selecting the corresponding COM port
  - Set the communication speed

- Using a reading node
  - Set the number of characters to be read
  - Eliminate the line breaks between each message

- Extract the necessary information
  - Open a Python node and set the version to use
  - Use function nodes
    - Specify the path to store the code
    - Specify the function name
    - Specify the type of data the function works on

- Sensor selection
  - Connect the output of the Python ID extraction node to the input of the case, to select the sensor with respect to its ID.

- Display message on flags
  - Connect the output of the serial communication after removing the line breaks to the text indicators.

- Display values on gauges
  - Connect the output of the Python value extraction node to the meter, and then evaluate whether the condition is met to give a status indication.

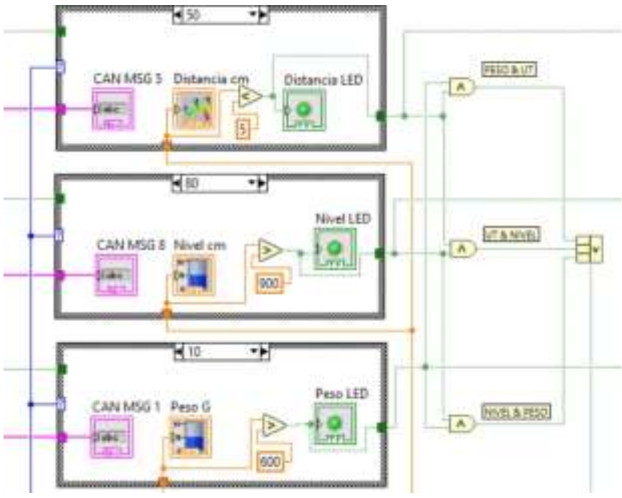

**Figure 7** Sensor redundancy

Using the concept of redundancy (Cueva Tumbaco, 2019) in the sensors, which consists of giving each sensor an importance (weight) that is evaluated in a logical function, if the sensors that work correctly have sufficient weight to declare the system operational, the system enables the output to the actuator.

It is proposed that the sensors have the same importance, because of this the condition is created that two of the three sensors in charge of monitoring the state of the tank must be working correctly and give similar measurements (Figure 7), so that the switch that turns on the boiler is enabled, this switch is shown in the HMI interface (Figure 5).

## Acknowledgement

The authors are grateful for the support and effort of the Universidad Autónoma de Campeche so that their researchers and students can disseminate the research topics in which they are involved.

## Conclusions

In conclusion, advances in virtual instrumentation have enabled the integration of CAN networks and microcontrollers in distributed sensor projects, providing efficient and reliable solutions for real-time data acquisition.

These technologies offer significant potential in a wide range of applications, from the electronics industry to industrial process monitoring and control.

There were no perceptible delays or drops in data, the microprocessors were programmed to turn on an LED in case they lost a packet of data, the tests lasted an average of 4 hours and there was no loss of data. Future tests are expected to apply methodologies to stress and push the network to its limit in order to characterise the robustness of this application.

## References

Al-Mutlaq, S. A. (2023). *Load cell amplifier hx711 breakout hookup guide*. Obtenido de sparkfun: https://learn.sparkfun.com/tutorials/load- cell-amplifier-hx711-breakout-hookupguide/introduction

Cueva Tumbaco, J. E. (2019). Diseño e implementación de un sistema redundante basado en equipos de la marca rockwell automation. *Bachelor's thesis, Espol*.

Esro, M. B. (2009). Controller area network (can) application in security system. *World Academy of Science, Engineering and Technology*, 35.

Fowler, C. J. (19 de marzo de 2023). *MCP_CAN_lib.* Obtenido de GitHub: https://github.com/coryjfowler/MCP_CAN_lib#readme

Gharavi, H. &. (2020). Deterministic intra-vehicle communications: timing and synchronization. *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 1-6. *https://doi.org/10.1109/ICC40277.2020.9148777*

H. F. Othman, Y. R.-A. (Abril 2006). Controller area networks: evolution and aplications. *2nd International Conference on Information Communication Technologies*, vol. 2, pp. 3088-3093.

Jia, J. K. (2009). Design of automated test system based on GPIB. *9th International Conference on Electronic Measurement & Instruments* , 1-943. *https://doi.org/10.1109/ICEMI.2009.52743 84*

Lv, T. Y. (2019). On Massive JSON Data Model and Schema. . *In Journal of Physics: Conference Series*, (Vol. 1302, No. 2, p. 022031). IOP Publishing. *https://doi.org/10.1088/1742-6596/1302/2/022031*

Microchip. (2018). *Stand-Alone CAN Controller with SPI Interface.*

Mora-Castillo, J. A. (2016). Serialización/ deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura. *Revista Tecnología en Marcha*, 29(1) 118-125. *https://doi.org/10.18845/tm.v29i1.2544*

Morgan, E. J. (2014). HC-SR04 ultrasonic sensor. *https://kysungheo.com/.*

Nurseitov, N. P. (2009). Comparison of JSON and XML data interchange formats: a case study. *Caine*, 9, 157-162.

Skogan, D. (1999). UML as a schema language for XML based data interchange. *Proceedings of the 2nd International Conference on The Unified Modeling Language (UML'99).*

*Technical documents: Espressif Systems*. (24 de mayo de 2023). Obtenido de Technical Documents | Espressif Systems: https://www.espressif.com/en/support/documents/technical-documents

Wahyuni, R. R. (2021). Home security alarm using Wemos D1 and HC-SR501 sensor based telegram notification. *Journal of Robotics and Control (JRC),* 2(3), 200-204. *https://doi.org/10.18196/jrc.2378*