Development of a customisable hardware data acquisition system using standardized protocols.

C. Martínez-Ruedas(1), Lucio Ciabattoni(2), Gabriele Comodi(2), J.M. Flores-Arias, Senior Member IEEE(1), F.J. Bellido-Outeiriño, Senior Member IEEE(1)
(1)R&D Group 'Industrial Electronic Instrumentation', Universidad de Córdoba, {z42maruc; jmflores; fjbellido}@uco.es; (2) Marche Polytechnic University, Department of Industrial Engineering and Mathematical Sciences (DIISM) {l.ciabattoni; g.comodi}@staff.univpm.it

2023 IEEE 13th International Conference on Consumer Electronics - Berlin (ICCE-Berlin) | 979-8-3503-2415-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCE-BERLIN58801.2023.10471191

Abstract-- The article presents the design of a customisable IoT system capable of transmitting data collected by sensors to a main (remote) headend, using the LoRAWAN protocol. The designed prototype allows the configuration of the inputs and outputs of the system, in order to adapt to the specific needs of each application.

The device is mainly composed of the Sodaq Explorer development board, to which a series of hardware interfaces have been incorporated. These interfaces have been designed to allow the connection of a wide variety of devices of different nature. The configuration of the different inputs/outputs is carried out through the microcontroller's serial port, using the Modbus RTU protocol. In this way, the system is not closed, but can be used in different environments and scenarios.

In order to increase the interoperability of the remote, the data, in addition to being sent by LoRaWAN, is exposed through the ModBus RTU protocol. Exposure via LoRAWAN allows access through The Things Network platform, while publishing via ModBus extends the scope to industry-standard protocols. This feature extends the range of operation of the system and allows users to monitor and analyse data in real time, in order to detect possible anomalies and make informed decisions.

***Index Terms*— LoRaWAN, IoT, Customizable hardware, Sodaq explorer, Modbus.**

I. Introduction

In recent years, the Internet of Things (IoT) paradigm has been increasingly present in different areas such as: (i) smart homes, to control lighting, air conditioning and household appliances [1]; (ii) smart cities, acting on traffic management, waste collection and energy supply [2]; (iii) industry, supervising and optimising production processes [3], [4]; (iv) healthcare, monitoring the condition of patients [5]; and (v) agriculture, contributing to the precise management of crops and irrigation, among others [6].

This decentralisation and flexibility offered by IoT systems should allow the technology to be adapted to different environments that increasingly demand product customisation. Therefore, for data acquisition systems there is a wide variety of devices and sensors from different manufacturers. This can lead to a lack of standardised interfaces or to closed systems that hinder interoperability between devices. On the other hand, the lack of standardisation and security [7], [8] can be an obstacle to the implementation of these systems, as end users may find it difficult to select, integrate and manage different IoT solutions. Standardisation would help facilitate compatibility and collaboration between different devices and vendors.

The need for interoperability, security and reliability of IoT systems has led to the emergence of organisations and protocols for standardisation. Among the most relevant is the use of LPWAN (Low Power Wide Area Networks) [9]. These wireless networks are characterised by transmitting data over long distances and with low power consumption, but sacrificing the amount of data they are capable of transmitting. Within these networks we find the LoRaWAN (Long Range Wide Area Network) protocol, which is developed on LoRa (Long Range) radio modulation. This protocol can provide connectivity in rural areas without the need for a large amount of network infrastructure [10]. One of its main strengths is its long range, which allows it to cover large areas. In addition, its low power consumption makes it ideal for applications running on long battery life. It is also highly scalable, which makes it an ideal choice for systems that require the addition of new devices in the future.

However, IoT systems, particularly in the agricultural sector, are still in the process of development. These systems present several issues such as the need for generic interfaces, lack of standardisation, communication requirements and low power consumption. The European Union has detected this problem and has initiated a project to address it, called DEMETER. The main objective of this project is to realise a large-scale deployment of farmer-centric, IoT-based interoperable platforms. To this end, 20 pilot projects have been championed in 18 countries. DEMETER aims to demonstrate the potential in interoperability by extending advanced IoT standards to a global agricultural information model [11].

Therefore, the main objective of the paper is the development of a customisable IoT system that exposes data via LoRaWAN and ModBus RTU protocols. To this end, a prototype has been developed, based on the Sodaq Explored development board, which is equipped with a series of hardware interfaces, designed to facilitate the connection of a wide range of devices with different characteristics. The system has been designed in such a way that its inputs and outputs can be configured through the ModBus RTU protocol, with the aim of adapting to new associated sensor needs.

The work is structured as follows. Section II describes the material and methods used to implement the system. Section III provides the results and some technical details for its configuration. Finally, Section IV includes some concluding remarks.

II. MATERIALS AND METHODS

The prototype has been designed to have two modes of operation, a configuration mode and an operation mode. The configuration mode allows the different inputs and outputs of the device to be set up to adapt them to the requirements of the sensors. The hardware interface will dynamically adapt the electrical signals between the sensors and the board based on the configuration set.

This configuration can be carried out from a PC, with serial communication to the board, or via Bluetooth through a mobile device. In both cases, the ModBus RTU protocol will be used through the Simply ModBus program [12] for the configuration of the signals. Table 1 shows the different registers (holding register) for the configuration.

Table 1.ModBus registers and values for device configuration (H.R.: Holding Register / R.S.: Range Select / b: bits / D.C.: Duty cycle /N.P: No. of pulses)

H.R.	INPUT/ OUTPUT	VALUES (16 Bits)	MODES
10	DI1	bit 0: Enable	
11	DI2	bit 0: Enable	
12	DI3	bit 0: Enable	
13	CI	bit 0: Enable bits 1-15: N.P.	
14	AI 4-20 mA	bit 0: Enable	
15	AI 4-20 mA	bit 0: Enable	
16	AI: 0-10V 0-5V 0-2,56V 0-1,28v	bit 0: Enable bits 1-4: R.S.	0101= 0 to 2.5xVref 0110= 0 to 1.25xVref 0111= 0 to 0.625xVref 1111= 0 to 0.3125xVref
17	Dendrometer	bit 0: Enable bits 1-2: Gain bits 3-7: Reads bits 7-15: Offset	b 2-1: 00 -Gain 128 01-Gain 64 10-Gain 32 b 7-3: Max 30 Reads
18	DO1	bit 0: Enable	
19	DO2	bit 0: Enable	
20	DO3	bit 0: Enable	
21	PWM	bit 0: Enable bits 1-7: D. C.	D.C.: 0-100
22	AO 4-20 mA	bit 0: Enable bits 1-3: R.S.	b 3-1:101 4-20 mA 110 0-20 mA 111 0-24 mA
23	AO 4-20 mA	bit 0: Enable bits 1-3: R.S.	b 3-1: 101 4-20 mA 110 0-20 mA 111 0-24 mA
24	AO 0-10 V	bit 0: Enable bits 1-3: R.S.	b 3-1: 000 0-5 V 001 0-10 V
25	AO 0-10 V	bit 0: Enable bits 1-3: R.S.	b 3-1: 000 0-5 V 001 0-10 V

In the operating mode, the device takes the data from the previously enabled and adapted sensors and transmits them via LoRAWAN to the TTN platform (The Things Netwok). Additionally, in order to extend the scope of the device to the industrial sector, it exposes the data via the ModBUS RTU protocol. Figure 1 shows the system architecture.

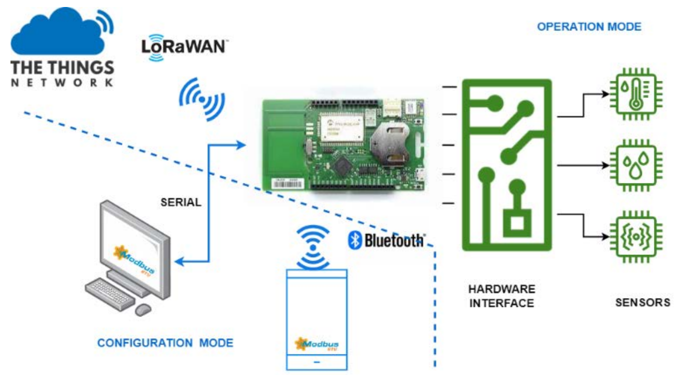


Fig 1. System architecture

For the hardware implementation, the SODAQ EXPLORER development board [13] has been used. The Sodaq Explorer is based on Microchip's 32-bit ATSAMD21G18A microcontroller, which uses the ARM Cortex-M0+ architecture, it has Microchip's RN2483 LoRa module on board, as well as Bluetooth Low Energy (BLE) connectivity through Microchip's RN4871 module. Programming has been done through the Platformio IDE [14] integrating the following libraries: Sodaq_RN2483.h for LoraWAN [15]; Sodaq_wdt.h, for the Watch Dog Timer [16]; FlashStorage_SAMD. h for storing and retrieving user data via emulated EEPROM memory [17]; ArduinoModbus.h, for reading and writing ModBus [18]; ArduinoRS485.h for sending and receiving data via RS485 [19]; RN487x_BLE.h for managing Bluetooth communications [20].

For the development of the hardware interfaces, use was made of the KiCad 7.0 program [21] software simulations were carried out with Multisim 14.0 [22] as well as laboratory implementations. Figure 2 shows the PINOUT of the microprocessor for the development of the different inputs/outputs.

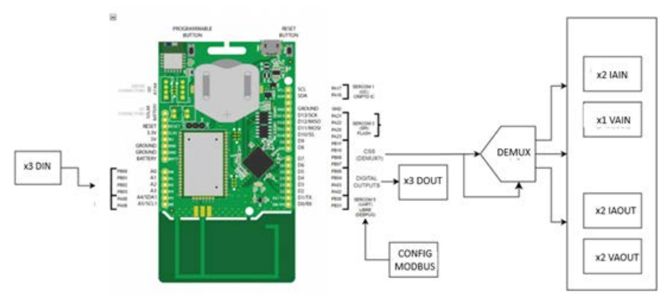


Fig 2 PINOUT of the microprocessor to implement the inputs/outputs

III. EXPERIMENTAL RESULTS

A prototype has been designed and developed with different hardware interfaces to implement the following configurable inputs/outputs:

1. Inputs

1.1. Digital Inputs

- 3 digital inputs, these inputs must be adapted to the logic level of the micro, i.e. 3.3 V for the HIGH state and 0V for the LOW state. Generic inputs compatible with 2-wire and 3-wire digital sensors with PNP, NPN and TTL output. The supply voltages of the sensors must be optoisolated and can be: +24VDC, +12VDC, +5VDC, +3.3VDC.

- 1 counter input, capable of reading fast pulses.

1.2. Analogue Inputs

Regarding the analogue inputs, 4 programmable ADCs communicating via SPI will be used, so that the ADC integrated in the microcontroller, which can only operate from 0 to ~3.3V, is not limited. These ADCs extend the read range from -12 to 12V. The requirements to be met by the inputs are:

- 2 inputs at 4-20 mA (live zero) to detect sensor breakage in case it delivers 0 mA.
- 1 default voltage input at 0-10 V. Which can be configurable to (0-2.5; 0-1.25; 0-0.625;0-0.3125 V).
- 1 input to read millivolts from strain gauge type sensors.

2. Outputs

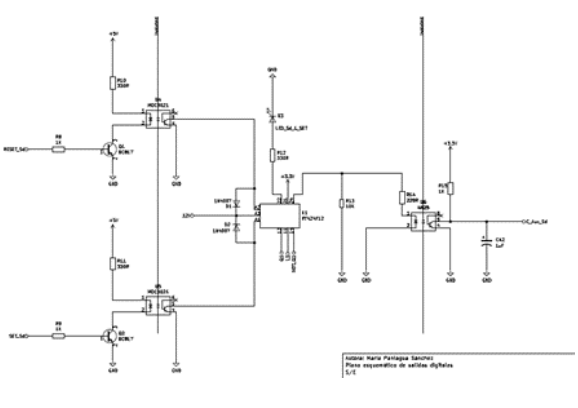
2.1. Digital Outputs

- 3 relay outputs, so that these are isolated from the device, to avoid continuous consumption and seeking maximum energy efficiency, it was defined that these outputs would use a latching relay. In this way, it is not necessary to continuously supply power to the relay, only when a change of state is desired.
- 1 PWM output, so that, for example, pavement lighting can be controlled by specifying the duty cycle.

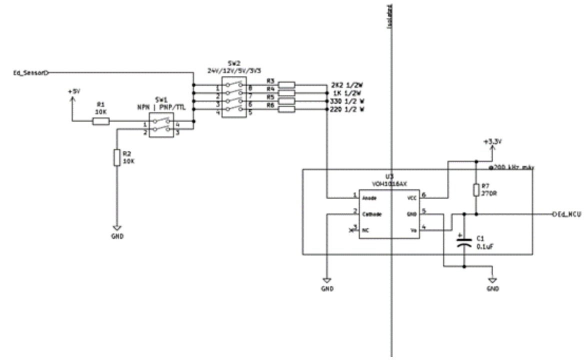
2.2. Analogue Outputs

- 2 default outputs at 4 - 20 mA. These outputs can be configured to different values (4-20; 0-20 and 0;24 mA).
- 2 outputs at 0 - 10 V. These outputs can be configurable (0-5; 0-10 V).

As an example, the implementation of the digital inputs/outputs will be shown. Figure 3 shows the schematics for the hardware interfaces of the digital signals, namely Figure 3a shows the digital outputs, while Figure 3b shows the inputs.



(a)



(b)

Fig 3: schematic diagrams of the digital hardware interfaces (a) Outputs;(b) Inputs

To enable these inputs/outputs, the respective ModBUS registers must be written. Figure 4 shows an example of the configuration through the Simply ModBus program.

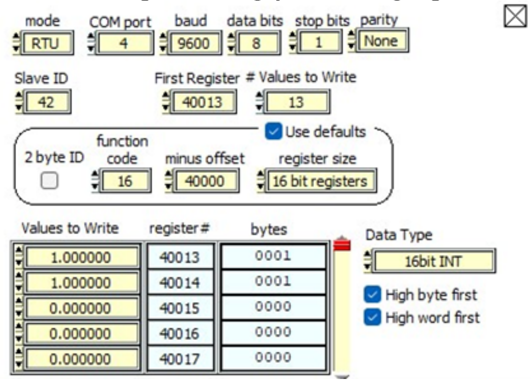
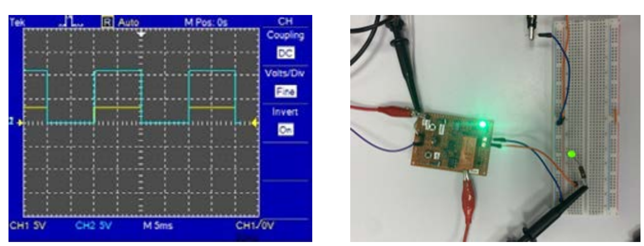


Fig 4.Enable 2 digital outputs by writing ModBus registers

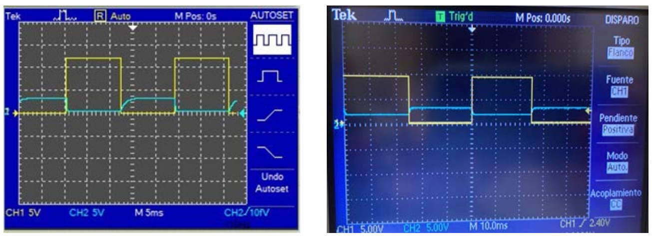
On the other hand, Figure 5a shows the simulation of the hardware interface of a digital output with Multisim, where it can be seen that channel 1 (yellow) would be equivalent to the control signal coming from the digital output port of the microcontroller, and that of channel 2 (blue) would be equivalent to the output signal of the exciter stage of the peripheral line. As can be observed, the electrical levels correspond to those calculated, thus fulfilling the requirements of the electronic design, the yellow trace from 0 to 3.3V and the red trace from 0 to +12V, the latter signal being the one that excites the inductor of the bistable relay considered. Figure 5b shows the implementation of this hardware interface.



(a) (b)

Fig 5.(a) Simulación de salida digital de 12 V con multisim; (b) Implementación de interfaz hardware de la salida digital.

Regarding the digital inputs, Figure 6a shows the simulation of a 12 V input simulated through Multisim, while Figure 6b shows the real simulation in the laboratory. In both cases channel 1 (yellow) shows the signal coming from the sensor while channel 2 (blue) shows the input signal to the microcontroller. As can be seen in both cases, the input signal coming from the digital sensor is a signal from 0 to +12V, while the signal obtained through our digital input interface corresponds to a signal from 0 to +3.3V, logical levels interpretable by the digital input port of a microcontroller system.



(a) (b)

Fig 6. Implementation of the hardware interface for a digital output to REL

IV. Conclusions and Future Work

The result is a customisable IoT device through the ModBus RTU protocol (via serial communication or Bluetooth) that allows the configuration of the inputs and outputs of the system, through the programmable hardware interfaces, with the aim of adapting to the specific needs of each application. In order to increase the interoperability of the remote, the data, in addition to being sent by LoRaWAN, is exposed through the ModBus RTU protocol. Exposure via LoRAWAN allows access through The Things Network platform, while publishing via ModBus extends the scope to industry-standard protocols. This feature extends the range of operation of the system and allows users to monitor and analyse data in real time, in order to detect possible anomalies and make informed decisions.

The hardware interfaces have been designed, developed and tested independently. Therefore, as future work, the implementation of the different interfaces on a single board is proposed, with the aim of developing a marketable prototype. After this, tests will be carried out in real environments capturing data in olive farms within the Smart-o-Live project in which this development is framed.

V. ACKNOWLEDGEMENTS

This research is supported by the Spanish Ministry of Science and Research in the framework of the project Smart-o-Live MIG-20211025 of the "Centro para el Desarrollo Tecnológico Industrial (CDTI)" in collaboration with the company DEUSER.

References

[1] D. Mourtzis, "Simulation in the design and operation of manufacturing systems: state of the art and new trends," *Int J Prod Res*, vol. 58, no. 7, pp. 1927–1949, Apr. 2020, doi: 10.1080/00207543.2019.1636321.

[2] R. H. Filho, D. C. B. de Sousa, W. A. de Brito, J. L. M. de S. Chaves, E. L. Sá, and V. P. de A. Ribeiro, "Increasing Data Availability for Solid Waste Collection Using an IoT Platform based on LoRaWAN and Blockchain," *Procedia Comput Sci*, vol. 220, pp. 119–126, Jan. 2023, doi: 10.1016/J.PROCS.2023.03.018.

[3] A. Anbalagan and C. F. Moreno-Garcia, "An IoT based industry 4.0 architecture for integration of design and manufacturing systems," *Mater Today Proc*, vol. 46, pp. 7135–7142, Jan. 2021, doi: 10.1016/J.MATPR.2020.11.196.

[4] A. G. Mohapatra *et al.*, "An Industry 4.0 implementation of a condition monitoring system and IoT-enabled predictive maintenance scheme for diesel generators," *Alexandria Engineering Journal*, vol. 76, pp. 525–541, Aug. 2023, doi: 10.1016/J.AEJ.2023.06.026.

[5] G. Zhang and N. J. Navimipour, "A comprehensive and systematic review of the IoT-based medical management systems: Applications, techniques, trends and open issues," *Sustain Cities Soc*, vol. 82, p. 103914, Jul. 2022, doi: 10.1016/J.SCS.2022.103914.

[6] D. A. Gzar, A. M. Mahmood, and M. K. A. Al-Adilee, "Recent trends of smart agricultural systems based on Internet of Things technology: A survey," *Computers and Electrical Engineering*, vol. 104, p. 108453, Dec. 2022, doi: 10.1016/J.COMPELECENG.2022.108453.

[7] I. Nadir, H. Mahmood, and G. Asadullah, "A taxonomy of IoT firmware security and principal firmware analysis techniques," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100552, Sep. 2022, doi: 10.1016/J.IJCIP.2022.100552.

[8] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres, and I. Kantzavelou, "IoT: Communication protocols and security threats," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 1–13, Jan. 2023, doi: 10.1016/J.IOTCPS.2022.12.003.

[9] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/J.ICTE.2017.12.005.

[10] "IoT en la agricultura: soluciones inteligentes para la agricultura basadas en IoT | Arrow.com." https://www.arrow.com/es-mx/research-and-events/articles/iot-in-agriculture-iot-based-smart-agriculture-solutions (accessed May 25, 2023).

[11] "Building an Interoperable, Data-Driven, Innovative and Sustainable European Agri-Food Sector | DEMETER | Project | Fact sheet | H2020 | CORDIS | European Commission." https://cordis.europa.eu/project/id/857202 (accessed Jun. 10, 2023).

[12] "Modbus - Supplier information." https://modbus.org/viewcompany.php?id=196 (accessed Jul. 12, 2023).

[13] "Overview - SODAQ Support pages." https://support.sodaq.com/Boards/ExpLoRer/#overview (accessed Jun. 06, 2023).

[14] "A professional collaborative platform for embedded development · PlatformIO." https://platformio.org/ (accessed Jul. 12, 2023).

[15] “Sodaq_RN2483 - Arduino Libraries.” https://www.arduinolibraries.info/libraries/sodaq_rn2483 (accessed Jul. 12, 2023).
[16] “GitHub - SodaqMoja/Sodaq_wdt: An Arduino library for the watchdog.” https://github.com/SodaqMoja/Sodaq_wdt (accessed Jul. 12, 2023).
[17] “GitHub - khoih-prog/FlashStorage_SAMD: The FlashStorage_SAMD library provides a convenient way to store and retrieve user’s data using the non-volatile flash memory.” https://github.com/khoih-prog/FlashStorage_SAMD (accessed Jul. 12, 2023).
[18] “GitHub - arduino-libraries/ArduinoModbus.” https://github.com/arduino-libraries/ArduinoModbus (accessed Jul. 12, 2023).
[19] “GitHub - arduino-libraries/ArduinoRS485.” https://github.com/arduino-libraries/ArduinoRS485 (accessed Jul. 12, 2023).
[20] “Microchip_RN487x/src/RN487x_BLE.h at master · SodaqMoja/Microchip_RN487x · GitHub.” https://github.com/SodaqMoja/Microchip_RN487x/blob/master/src/RN487x_BLE.h (accessed Jul. 12, 2023).
[21] “Version 7.0.0 Released | KiCad EDA.” https://www.kicad.org/blog/2023/02/Version-7.0.0-Released/ (accessed Jul. 12, 2023).
[22] “Multisim Live Online Circuit Simulator.” https://www.multisim.com/ (accessed Jul. 12, 2023).