IIoT Architecture Using LoRa Technology, LoRaWAN and OPC UA Protocols

Angelo Guimarães Silva
Federal University of Minas Gerais
6627 Ave. Antônio Carlos, Belo Horizonte, MG 31270–901, Brazil
angeloguimas@ufmg.br

Hugo César Coelho Michel
Electronic Engineering Department
Federal University of Minas Gerais
6627 Ave. Antônio Carlos, Belo Horizonte, MG 31270–901, Brazil
hugomichel@ufmg.br

Abstract

This paper presents the integration of a wireless long-range sensor network (LoRa), using the LoRaWAN protocol, to an OPC TCP network with OPC UA standard. This proposal allows the creation of cyber-physical spaces where all the elements can communicate in a safe, adaptable, and scalable way offering an IIoT (Industrial Internet of Things) solution - useful for Industry 4.0. Due to the specificities of each protocol (LoRa, LoRaWAN and OPC UA), developments in hardware, firmware, and software are presented for the purpose to obtain an integrated solution capable of providing data from a LoRaWAN network directly to an HMI, using the OPC UA standard. The work also presents details of the embedded electronic system, developed for the LoRa sensor nodes, including the specification of the components used and signal conditioning circuits required. Furthermore, this work also presents the implementation stages of the LoRaWAN - OPC UA Gateway, based on the open-source operational system, OpenWrt, one of the main components necessary for the proposed integration working. Suggestions for programming the OPC UA server, embedded in the OpenWrt OS, are detailed. Tests of the developed LoRa sensor nodes, as well as the integration with a SCADA operator interface are presented, demonstrating that the proposal integration is feasible and can be applied to increase I/O points monitored by an operator, without the need for intervention in the installed automation system.

Keywords
IoT, OPC UA, LoRaWAN, OpenWrt and Industry 4.0.

1. Introduction

The use of technologies that aim at the concept of IoT (Internet of Things) is a vast field to be explored, bringing viability to Industry 4.0. The concept of IoT was originally employed by Kevin Ashton in 1999, when he proposed the idea of different objects feeling the world around them through networks (Rangel 2014). At the present time, the IoT can be better interpreted as digital interconnections between objects, assets and systems that were previously isolated, but that are now able to connect and exchange information with each other. In the industrial context, a new term derived from IoT emerges: the Industrial Internet of Things (IIoT), which takes into account the specific needs of the industrial environment in the use of solutions based on the concept of IoT, such as security, availability, non-intrusive data collection in the industrial process and compatibility of communication protocols.

There are several protocols and communication technologies based on the Internet of Things, including the LoRaWAN (Long Range Wide Area Network) protocol and the LoRa (Long Range) radio transmission technology. LoRa technology has gained prominence among developers for bringing together in a single, small and inexpensive hardware, interesting features for sensor networks applied to IoT and IIoT, such as long range, low power consumption and high noise immunity (Compology 2017), (Purche 2016). The LoRaWAN protocol is maintained by the LoRa Alliance and defines in its network structure an architecture in star topology protected by 128-bit encryption, allowing the communication of end devices with a network central concentrator point, the gateway (LoRa Alliance 2015).

With regard to the Industrial Internet of Things, the OPC UA protocol has been considered by many experts to be the ideal protocol for Industry 4.0 as it is light, flexible and, above all, platform independent. The OPC UA is a standard protected by the IEC (International Electrotechnical Commission) and is part of the roadmap of the Association of Electrical, Electronic and Information Technologies (Verband der Elektrotechnik Elektronik Informationstechnik eV - VDE) as a standard protocol of Industry 4.0 (Schroeder 2016).

1.1. Objectives

Separately, LoRaWAN and OPC UA are, respectively, ideal stardart for IoT and industrial conectiviy aplications. The present work shows that the integration of these protocols, together with the radio transmission tecnology LoRa, can be a wireless alternative, viable and promising for apply of IIoT concept, since it provides low cost solutions focused on the new context of industry 4.0. This paper presents a development of a LoRaWAN / OPC UA gateway as a strategy to reach the mentioned between both two protocols integration.

2. LoRa, LoRaWAN and OPCUA technologies and their advantages for IIoT

2.1. LoRa technology: main features

LoRa technology makes use of the unlicensed frequency bands, named like this because they are part of the ISM (Industrial Scientific and Medical) bands, an internationally disseminated standard, aimed at the development of industrial, scientific and medical applications. It is important to note that the LoRa Alliance defines regional parameters worldwide, so that LoRaWAN networks can operate without violating regulatory standards regarding the radio spectrum anywhere in the globe (LoRA Alliance 2018).

The LoRa radio transmission technology, together with the LoRaWAN protocol, stands out for the modulation used and the mode of operation, capable of ensuring a good relationship between energy consumption and transmission range, besides the high noise immunity, ideal for industrial environments. LoRa offers a modulation technique derived from CHIRP (Compressed High Intensity Radar Pulse) modulation, which consists of the brief increase (up-chirp) or decrease (down-chirp) of the frequency over time, representing the bits of the packet in a digital radio transmission. This technology is not recent and having been used for a long time for militaries applications that demand security (Semtech 2015).

The frequency variation in a LoRa modulation over the time have a preamble at the beginning of the package, of an up-chirp sequence followed by a down-chirp sequence, this preamble is automatically added by the hardware. In (TALLA et al 2017), a frequency analysis of a LoRaWAN frame is presented with details of Preamble, sync (up and down)-chirp plus Header and Payload with error check. The decoding of the package involves the use of three more factors (Semtech 2015) and (LoRA Alliance 2018):

- ***Band width* (BW)**: Are accepted the values of 125 kHz, 250 kHz and 500 kHz.
- ***Code rate* (CR)**: Are accepted the values of 4/5, 4/6, 4/7 and 4/8. Defines the bits rate that are used on header and payload, the other bits are the CRC (Cyclic Redundancy Check).
- ***Spread factor* (SF):** Are accepted the valors of 6, 7, 8, 9, 10, 11 and 12. These values define subdivision on modulation to create the data presentations named "LoRa symbol".

The characteristics of the LoRa modulation presented drastically reduce the risk of data collision, since it is necessary that two or more packets arrive at the receiver simultaneously, with the same frequency, the same spread factor and the same bandwidth.

2.2. LoRaWAN Protocol: main features and architecture

LoRaWAN is a wireless protocol for LPWAN (Low Power Wide Area Networks) networks, developed for use with LoRa transceptors. The signal range can exceed 12km, although much factors can influence on this range, as obstacles, antenna quality, beyond the network configurations.

LoRa technology acts on the first layer of the OSI (Open System Interconnection) model, the physical layer. The LoRaWAN protocol acts on the second layer, MAC (Media Access Control), or link layer of the OSI model (LORA ALLIANCE, 2015). The other layers of the OSI model, i.e. (3) Network, (4) Transport, (5) Session and (6) Presentation, are not required by the LoRa and LoRaWAN protocol stack. The explanation for this is in the fact that in a LoRaWAN network it concentrates the data in a gateway, responsible for receiving data from the LoRa network and converting it into data in the TCP/IP protocol stack, occupying the (1) Ethernet/IEEE layers 802.3, (2) MAC, (3) IP and (4) TCP/UDP.

The LoRaWAN protocol defines characteristics of network end nodes according three classifications: ClassA, Class B or Class C (LoRa Alliance 2015):

- **Class A**: standarts class with assyncronous transmission and the lowest consumption.
- **Class B**: nodes synchronized with the network by scheduling reception windows.
- **Class C**: nodes receptions windows open indefinitely, closed only during transmission.

It is still important to say that the network nodes and the server have in their records the following keys, used to add security to the packets and address the nodes (LoRa Alliance 2017):

- **Device Address (DevAddr)**: unique key, responsible for identifying the node in the network.
- **Device Extended Unique Identifier (DevEUI)**: used to identify, globally, the final nodes in the process of joining the network using the OTAA (Over the Air Activation) method.
- **Application Session Key (AppSKey)**: key present in each node, used to encrypt the payload in the LoRaWAN package with the 128-bit AES-CTR algorithm.
- **Network Session Key (NwkSKey)**: used to verify the integrity of the packet on the network.
- **Application Key (AppKey)**: auxiliary key used in the OTAA method to generate AppSKey and NwkSKey, which are essential keys to encrypt the LoRaWAN package.
- **Application Extended Unique Identifier (AppEUI)**: global key used to identify the destination application for messages sent by the nodes. This key is present on the sensor node and on the server. The AppEUI is necessary to enter in the network using the OTAA method.

2.3. OPC protocol for diverse architectures

The OPC protocol (Open Platform Communications) aims to standardize communication and interoperability between the elements of the factory floor and other levels of the automation pyramid, by proposing an organizational data structure (OPC Foundation 2017) and (Mahnke et al. 2009), normally accessed over a network using the TCP/IP protocol stack.

OPC UA has been rated as the ideal protocol for applications of the Industry 4.0 concept, as it allows full integration between the various levels of a factory with minimal modifications to the Control and Automation Architecture (CAA) (OPC Foundation 2017), (Mahnke et al. 2009). The fact that it is an open and platform-independent protocol reinforces this preference. The OPC UA unified architecture concept provides for a mutual interaction between clients and servers through the interconnectivity between both. Thus, the same CAA can have, at the same time, several servers and clients. One client can connect to multiple servers and just one server can connect to multiple clients. It is also possible to have a hybrid client and server instance, useful to facilitate the interface between different systems (OPC Foundation 2017), (Mahnke et al. 2009).

2.3.1. OPC UA standard: connectivity

By using a client/server architecture, the OPC UA standard also implements network discovery methods, thereby facilitating clients to locate servers and then establish the session between them. According to Mahnke et al. (2009) the connection can occur in two forms:

- **OPC TCP**: native communication protocol in OPC UA, makes use of sockets to encapsulate all data in a binary form in a TCP frame. It is the most efficient and fastest protocol.
- **SOAP HTTP(S)**: SOAP (Simple Object Access Protocol) is a communication protocol structured based on XML (Extensible Markup Language). The transmission medium can be HTTP (Hypertext Transfer Protocol) or HTTPS (Hyper Text Transfer Protocol Secure).

2.3.2. OPC UA standard: security

Because it has three layers of security, which can be implemented jointly or separately, the OPC UA proves to be a very secure protocol. To understand security in the OPC UA standard, one can subdivide the process into three steps for an application with a known connection procedure (Mahnke et al. 2009):

- **Step 1:** consists of using security certificates between clients and servers;
- **Step 2:** possibility of using user authentication mechanisms on the server;
- **Step 3:** the possibility of creating a session with encrypted data on the network.

In modern industry, treating information only as a numerical value has been shown an insufficient strategy to deal with the immense amount of data (Commserver 2010). In this perspective, the OPC UA protocol instituted a new way of abstracting and modeling the elements in the network, in the form of objects, i.e. instances endowed with variables, method and events:

- **Variables:** represent a value and its data type. It is possible to read and write values.
- **Methods:** are the ways that clients must interact with servers, by execution of some task. Therefore, a customer must invoke a method made available by an object.
- **Events:** they can be of any nature, they signal events of objects for customers.

The set of these objects, together with their components, constitute the Address Space, i.e. a region of memory on the server used by clients to access the functionalities that the objects provide. All objects in Address Space are flagged as nodes, which are ways of organizing objects as instances modeled by two characteristics (OPC Foundation 2017), (Mahnke et al. 2009) and (Unified Automation 2020):

- **Attributes**: are the elements that describe the characteristics of the nodes. These elements are fundamentally id, name, description, data type and indicator, which describe, respectively, the numeric address of the node, the identification name, a brief description of the function of the node, the type of data provided and the signaling that the node was instantiated or not.
- **Reference**: reports how a node is hierarchically linked to other nodes.

In Figure 1, it is possible to see how *Address Space* is related to the connection between client and server. It shows an overview of the internal interfaces on the client and the server, through which the data travels in an OPC UA communication.

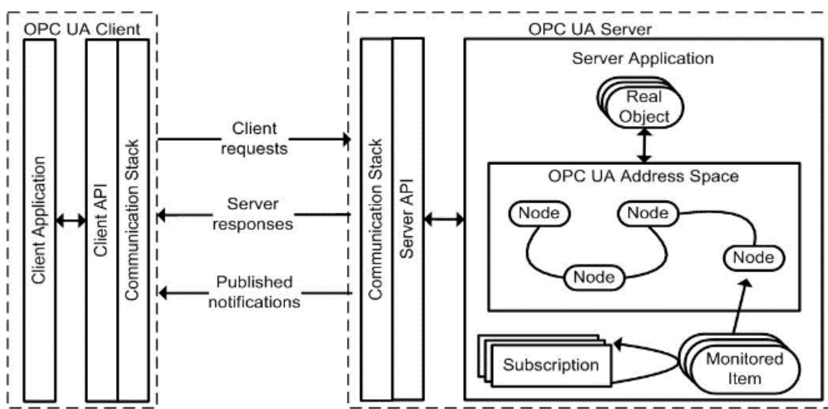


Figure 1. OPC UA client and server connection. Fonte: (Imtiaz and Jarperneite 2013).

3. Architecture for Sensor Nodes and Gateway LoRaWAN/OPC UA

3.1. Architecture for the Proposed Solution

The Figure 2 presents the proposal architecture aiming at integrating the characteristics of long range, low consumption, low collision rate and high noise immunity of LoRa and LoRaWAN technology, with the object model and connectivity of the OPC UA protocol. The sensor nodes, developed in this work, represents the physical layer (OSI model) and they are integrated into the environment to be monitored. The collected data travels initially through the LoRaWAN network to an OPC UA server, where OPC UA clients can connect to collect data from the nodes. In the access control layer, the LoRaWAN protocol predominates, depending directly on the mutual action between the sensor nodes and the gateway. In it, the nodes build the LoRaWAN package and transmit it on the wireless network modulated in LoRa. The gateway acts by receiving this packet in order to encapsulate it in a UDP/IP packet destined for external networks.

On this way, a software modification must be implemented on the original gateway firmware to allow the received LoRaWAN package to be duplicated and directed to the OPC UA server application, which must be running locally. Finally, the application layer is represented by the different destinations that the data can take according to the IoT service configured at the gateway and by the OPC UA client.

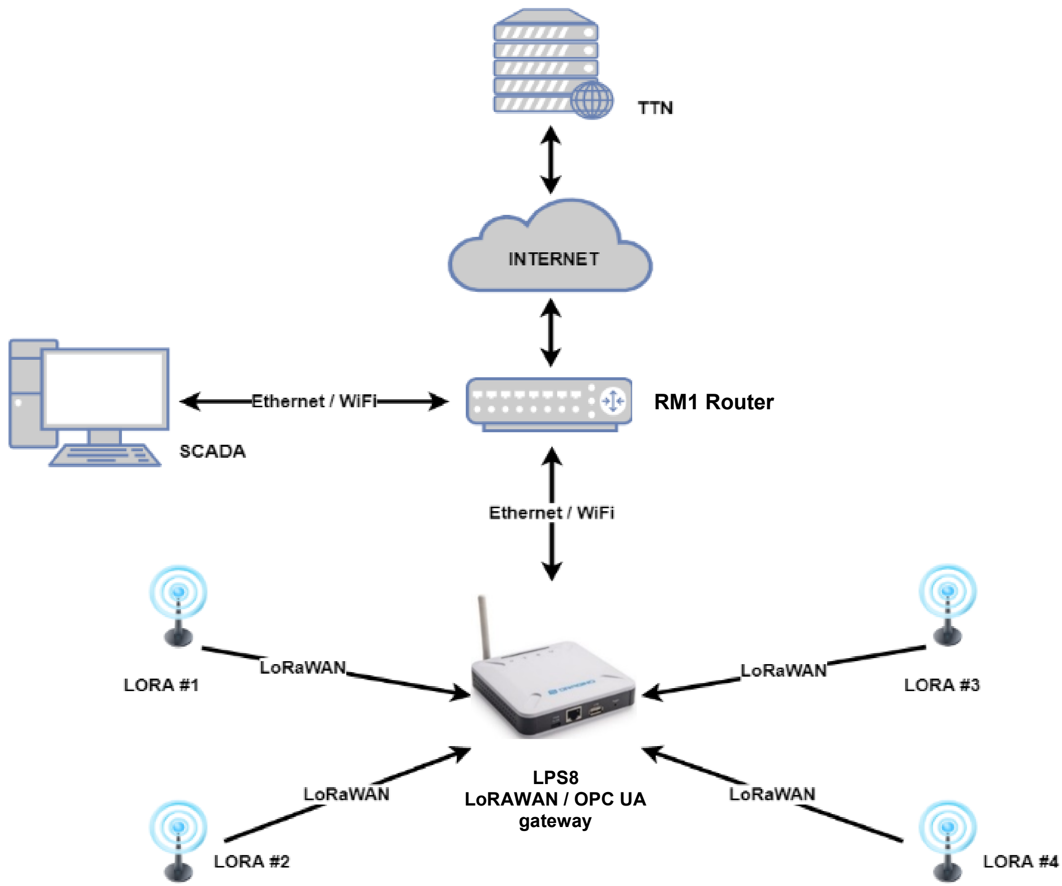


Figure 2. Architecture solution and the principal components around LoRaWAN / OPC UA gateway.

The Figure 2 shows the main elements that make up the entire proposed solution, with emphasis on the equipment: the RM1 Router, the station running a SCADA interface, IoT services in the cloud; the LPS8 gateway (LoRaWAN / OPC UA), whose firmware is modified and, finally, the LoRa sensor nodes, whose firmware and hardware, respectively, are all developed in this work.

3.2. Conception of the LoRaWan / OPC UA Gateway

It is important that the gateway is able to make the LoRaWAN payload content available on an OPC UA server. To achieve this goal, it is necessary to make modifications to the original software that runs embedded in the gateway and, with that, to obtain the desired integration of the two protocols.

The process of routing a packet at a gateway involves the growth of the initial packet due to the addition of new headers that enable the correct traffic of information as it passes through new networks. This process is possible thanks to connections and conversions of information both at the software and hardware levels that occur in this device (Tanenbaum 2002). The gateway used is the Dragino LPS8 915MHz model manufactured by Dragino Technology (Dragino 2019).

The operating system (OS) running on LPS8 is OpenWRT 18.06, i.e. an independent and flexible OS kernel GNU / Linux for embedded systems, especially for routers (Wich 2020). The operating system in LPS8, in addition to executing the routing algorithms mentioned above, also allows the execution of software written in C language, used in the development of the software capable of integrating the LoRaWAN and OPC UA protocols.

3.3. Conception of Sensor Nodes

The nodes are the points where the measurements of the plant actively take place, the values read in the measurements are subsequently treated and compressed in 4 bytes, to then be sent to the service configured at the gateway and in parallel sent to the developed OPC UA server. For testing, four Print Circuit Boards (PCBs) are developed for the nodes. In each PCB there is a µC (µC) from the Microchip Technology with the function of executing the LoRaWAN protocol stack, manipulating the I/O peripherals (Input/Output), the ADC (Analog to Digital Converter) and the SPI bus, used for communication with the SX1276 transceiver, manufactured by Semtech.

The µC chosen for the project is the PIC18LF**26**K22, containing 64 kB of internal memory, enough to receive the LoRaWAN protocol. In the application examples of the LoRaWAN ™ library (Microchip 2017) only µC models PIC18LF**46**K22 are used, however it is possible to use other models listed at the end of the document Release Notes for MPLAB® Code ConfiguratorLibrary LoRaWAN (Microchip 2016).

One of the contributions of this work is that, until now, the model PIC18LF**46**K22 is the only one validated by Microchip Technology to for LoRaWAN protocol. However, the PIC18LF**26**K22 is used, presenting satisfactory results and, therefore, meeting the proposed objectives.

4. Architecture Components Development

4.1. OPC UA Server Development

The OPC UA server is developed based on the open source **Open62541** library, which has the advantage of being able to be ported to distributions of different operating systems such as Microsoft Windows, Linux, QNX and Android (OPEN62541 2014). **Open62541** has functions that allow a value to be made available as an accessible variable on a server, so it is necessary that this value be obtained from some source. Therefore, by means of a program written in C language, it is possible to obtain values from different sources, e.g. a file, or a memory location, or a variable shared with another application, etc.

The proposed integration makes use of data present in the LoRaWAN packages, which travel through the `lora_pkt_fwd` application, running at the gateway. In this case, it is necessary to develop a strategy to send this data to the OPC UA server application. The native `lora_pkt_fwd` program in LPS8 is part of the lora-gateway package developed by Semtech. This package includes programs used as a starting point for the development of a multichannel gateway. The `lora_pkt_fwd` program acts to forward LoRaWAN packages.

With the code made available, it is possible to modify it to allow integration with the OPC UA server. Therefore, the modification to be made consists of copying the LoRaWAN package received in the application `lora_pkt_fwd`, before it is sent to the standard server via UDP/IP. The copy of the package is saved in a local file that can be accessed concurrently by the server application.

The OPC UA server should repeat, cyclically, the reading of the file shared with the application `lora_pkt_fwd`. In this way, whenever the current packet differs from the last packet read, the variable assigned to the address present in the current packet must be updated. The sender's address must be present at the beginning of the package, after the MAC Header field, and the payload, at the end of the package, protected by AES-CTR encryption. Finally, the integration developed still depends on a "**configuration file**" with the following fields to know the details of each node to be listed on the server:

- **Address**: Unique 4-byte address written in hexadecimal form in the first column.
- **Description**: A description of the node, up to 20 characters, must be in the second column.
- **AppKey**: Key used by the server to decrypt the package payload that is marked with the address in the first column. It should be in the third column.

The Figure 3 presents the code flow chart for the application that gives rise to the LoRaWAN - OPC UA gateway. It is possible to observe two threads used to execute the OPC UA server algorithm. On the left, the `main()` thread, where: (1) the loading of the settings used on the server through the **configuration file** occurs; (2) checking the existence of the shared file; (3) the creation of a second thread and (4) the call to execute the OPC UA server. On the right, th `read_data_to_nodes()` thread performs the continuous reading of the shared file, followed by the identification of the class A node address, which must have its respective variables updated on the server.

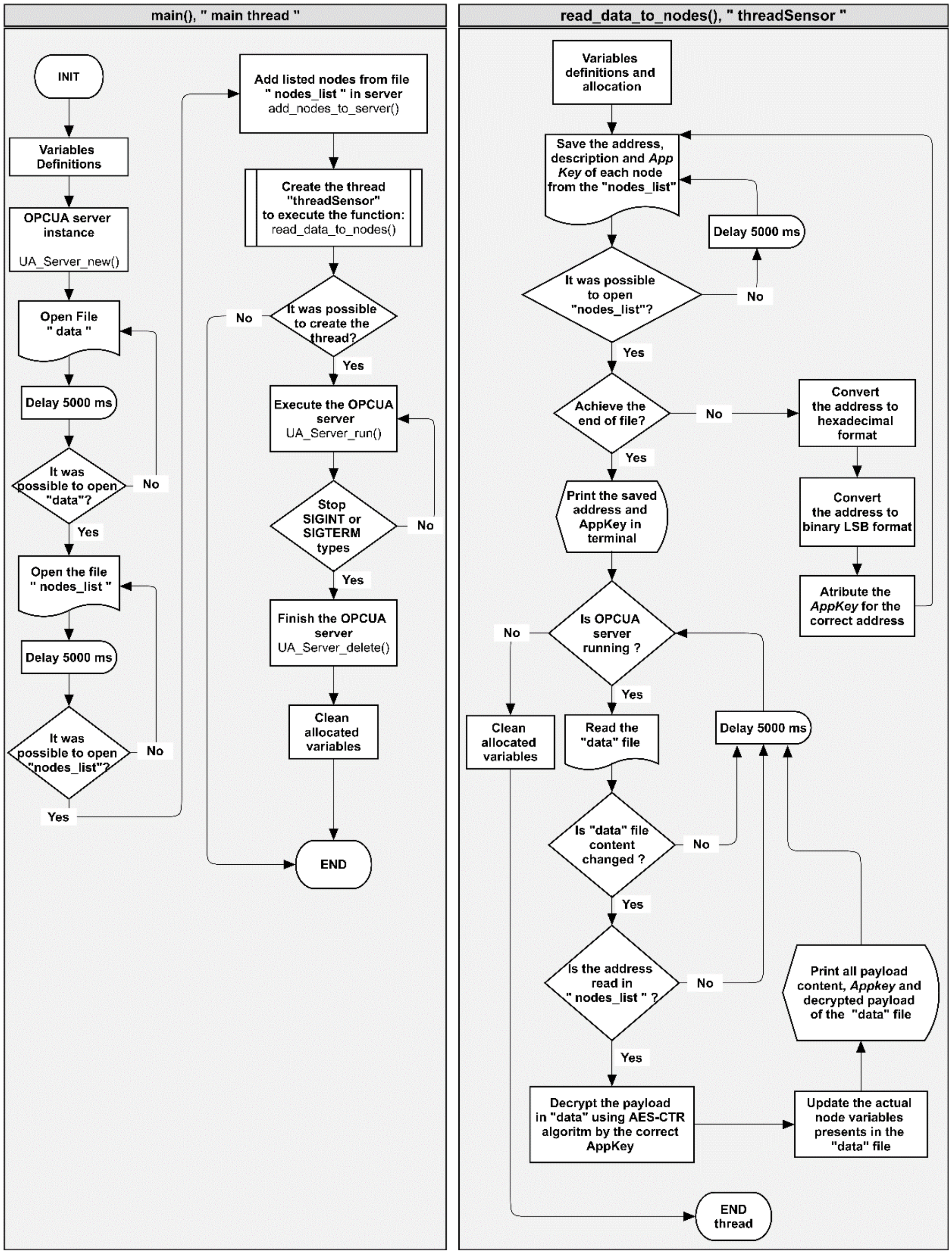


Figure 3. LoRaWAN / OPC UA gateway application - flowchart code.

4.2. Development of the Firmware of the Sensor Nodes

The document entitled “LoRaWAN ™ Library Plug-in for MPLAB® Code Configurator: User’s Guide” (Microchip 2017), details the steps required to use the LoRaWAN ™ Library plugin. This plugin facilitates the process of configuring the LoRaWAN library on the Microchip µCs.

The µC firmware present on the nodes must sequentially send the reading of the sensors connected to the node at 60-second intervals and, afterwards, two reception “windows” are opened. After transmission and possible data reception, the node must go into sleep mode, state in which the µC turns off the LoRa transceiver, its peripherals and its internal oscillator.

In the part destined to send data, two different values are sent in a fixed payload of 4 bytes, the first two bytes being used to send the identification of the current message, obtained through the `LORAWAN_GetUplinkCounter()` function. The remaining two bytes are used to send the reading value of the sensor or any other information of interest. To put the µC into sleep mode, use the `SLEEP()` function, available in the programming environment of MPLAB® X IDE v5.15, software used for programming the µC firmware.

4.3. Integration with a SCADA Interface

To check the data from the OPC UA server, the following OPC UA clients were used:

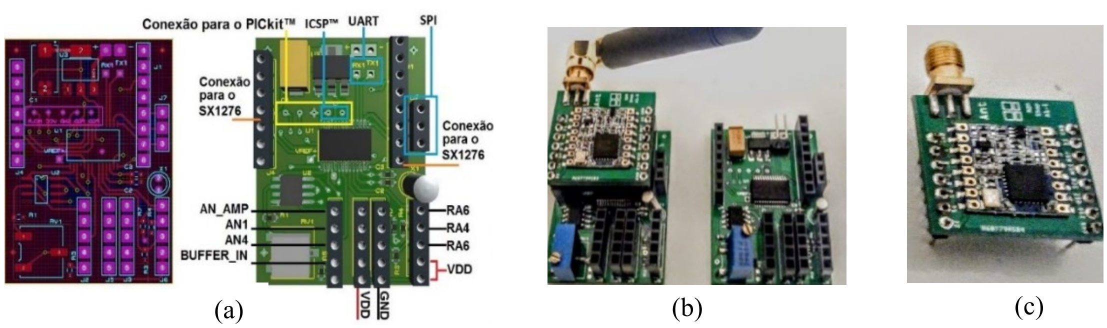
- **Prosys OPC UA Client**: Client for Android (release used, 7.1.1).
- **UaExpert**: Client for Ubuntu (release used, 18.04).

After verifying the data received by the mentioned OPC clients, an industry-oriented application is used, e.g. a SCADA interface (Supervisory Control and Data Acquisition), in order to test the features provided by the OPC UA server. Thus, an interface was developed in the Indusoft Web Studio 8.1 software, from the manufacturer AVEVA / Wonderware, which has a native OPC UA client.

Sections 5.1 and 5.2, presents the results of these tests, where it is possible to observe one of the most important characteristics in the OPC UA, i.e. its interoperability in different environments through TCP/IP.

4.4. Assembly and Programming of Transmitting Nodes

The electronic project should be a platform of a first module of reduced size, where the µC PIC18LF**26**K22 is located, a power supply and an I/O interface. A second module, containing the LoRa radio, is designed to connect to the first. The items a and b of the Figure 4 show the PCBs in the development stage, as well as after manufacture. The item c of the Figure 4 shows the LoRa radio module using the SX1276 transceiver.



(a) (b) (c)

Figure 4. LoRaWAN node project, manufacturing and mounting.

It is important to highlight the presence of the ground plane in the project, whose function is to create a path of less impedance for the return of current in the circuit and, thus, help to reduce problems arising from current pulses and electromagnetic interference.

4.5. Compiling and Running the OPC UA Server

The main contribution that this work presents is the execution of an OPC UA server on a LoRaWAN gateway, using the Linux OpenWRT distribution natively. To achieve this goal, it is necessary to make use of cross compile methods so that it is possible to obtain a program written in C language compatible with the architecture of the processor used in the LPS8, the Atheros AR933.

An agile way of doing the cross compile in the OpenWRT environment is through the SDK (Software Development Kit) for Linux environment, made available by the OpenWRT maintainer project (Bursi 2020). This SDK automates the entire process of compiling and creating packages. After compiling and installing the package, the OPC UA server is created as a Linux OpenWRT OS software and testing occurs by executing the **lorawan_opc_ua** command, i.e. the application name, as shown in Figure 5.

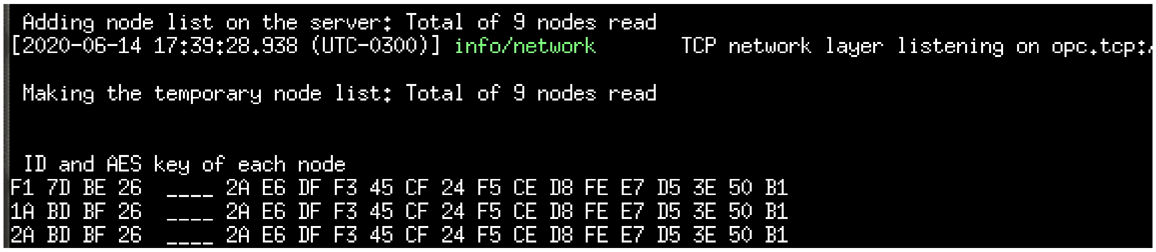


Figure 5. LoRaWAN / OPC UA gateway application running test in the terminal.

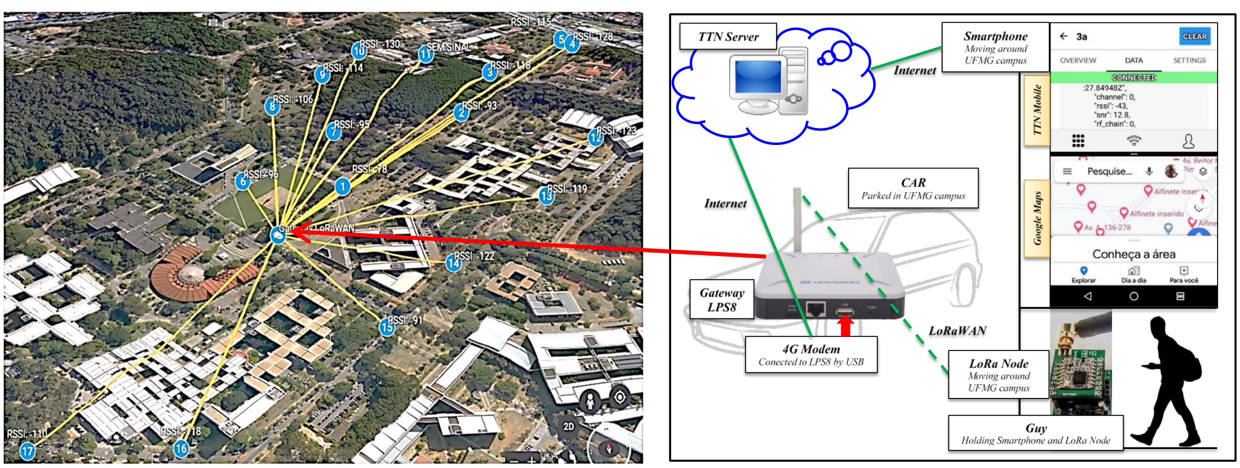
The Figure 5 is important because it shows part of the output of the OPC UA server program running. In the lines, it is possible to observe the establishment of the section and the addition of the nodes to the server. At the end, there is a summary of the nodes with their respective AppKeys. Such messages displayed on the terminal are very useful for debugging and verifying the operation of the application. Once the operation of the LoRaWAN network has been confirmed through the gateway's logfile, the execution of the OPC UA server through the connection with OPC UA clients is verified.

5. Results and Discussion

5.1. Range Test

The range tests were carried out on the Pampulha campus of the Federal University of Minas Gerais. The chosen location offers a variety of open and closed areas, as well as a variety of reliefs that can be used to test the transmission range in different circumstances.

For this test, the gateway is in a fixed position inside a car with an external antenna attached to the roof. In this way, moving around the UFMG campus, several measurements of the signal were made, as well as the registration of the geographical position of the node. The item a of the Figure 6 is possible to visualize all 17 measured points geographically spread in UFMG campus. The item b of the Figure 6 shows a schematic of the Range Test.



(a) (b)

Figure 6. Range test realized in UFMG campus.

The measurements made with a direct line of sight to the gateway, on flat ground and without obstacles, showed intensities of up to -115 RSSI for a distance of 781 m. The lowest signal occurs at 560m from the gateway, this measurement has the disadvantage of being behind obstacles such as trees and uneven relief. However, there is a relatively close concrete building, so at this point the intensity was -130. Table 1 presents some interestings results of the Range Test, relating the terrain conditions, distance and signal strength.

Table 1. Signal power, distance and test conditions compare.

Measured Point	Signal Power (RSSI)	Distance (m)	Test conditions
1	-78	125	Open field, flat terrain
5	-115	781	Open field, flat terrain
10	-130	560	Dense vegetation between gateway and node, uneven terrain
12	-123	463	Several concrete buildings between gateway and node, uneven terrain

5.2. OPC UA Network and Connection to the SCADA Interface via OPC UA

After checking the efficiency of the LoRaWAN network, the communication test on an OPC UA network is performed by connecting two OPC UA clients listed in item on the same network as the server developed and running on the LPS8 gateway. Figure 7 shows the OPC UA server running in the moment when data is received from two different nodes.

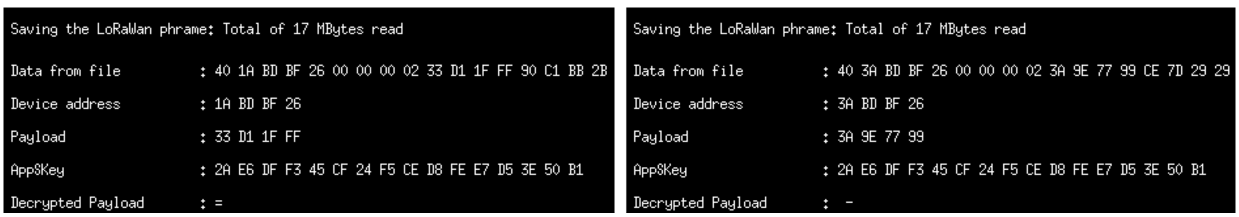


Figure 7. LoRaWAN / OPC UA gateway application running and receiving data.

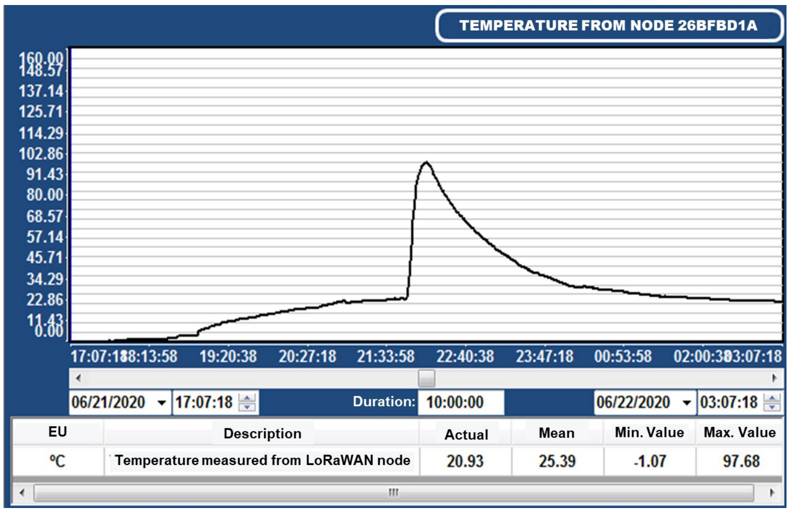
The images in Figure 7 are generated through an SSH terminal session. Details of the process of reading the shared file and the identification that each node sends to the gateway are shown in Figure 7, showing a connectivity and integration capacity of the developed OPC UA gateway, reading and decrypting the payload of multiples nodes. From top to bottom, it is possible to verify the receipt in sequence of data from different nodes through the ***Device address***, while the field ***Data from file*** exposes the raw data read from the shared file, where it is possible to extract the payload (Payload field in the Figure 7 through a decryption operation using the ***AppSKey***. Finally, by looking closely at Figure 7, it is possible to perceive the bytes present in the shared file and thereby identify all fields referring to the MAC layer.

The SCADA interface developed is presented in item a of the Figure 8 and aims to demonstrate the feasibility of integrating LoRa sensor nodes, via the LoRaWAN network, with the level of industrial operation, ie using a SCADA system communicating via OPC directly with the instrumentation level (first hierarchical level of automation). Note that this configuration allows the addition of new I/O points without having to intervene in the architecture of the existing automation system, usually expansions adding I/O modules in controllers, which implies logical reconfigurations, an increase in the number of electrical panels, etc. In other words, this architecture allows the expansion of low criticality measurement points for the process, but useful for several analyzes, with a minimum of local interventions.

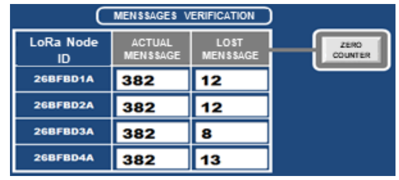
In view of the developed interface, Figure 8, item a presents the measurement for ten hours of the temperature of a portion of pure water (Figure 8, item c); the graph is updated with each new receipt of data coming from node 26BFBD1A. In item a of the graph in Figure 8 it is possible to identify several moments during the temperature measurement, in a volume of 500 mL of water, initially frozen, until reaching the ambient temperature of 29 ºC, where the system is placed under forced heating until reaching about 97º C, cooling naturally.

In order to demonstrate the efficiency of the implemented solution, item b of Figure 8 presents a record of lost packets in the OPC UA network, where four LoRa nodes (26BFBD1A to 26BFBD4A) are connected and exchanging messages with the SCADA system. The sequence of messages received using the first two bytes of the LoRaWAN payload makes it possible to identify the number of messages lost from each node, by comparing the number of the current message with the number of the last message received. Note that the worst loss rate was 3.4% and the best, 2.1%. It is worth mentioning that these messages can be lost for two reasons: packet loss during transmission or problems with the software integration.

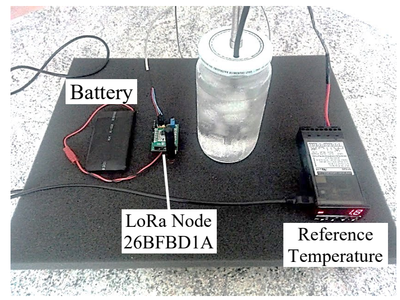
In the LoRaWAN network, packets can be lost due to collisions, this possibility is less likely to occur, since one or more packets must be on the same frequency (Coding Rate and Band Width) at the same time. Losses are more likely to occur due to failures in the execution of the LoRaWAN and OPC UA integration software, embedded in the gateway. A problem that is still under investigation.



(a)



(b)



(c)

Figure 8. (a) Hardware and SCADA interface test bench.

6. Conclusion

This work sought to address all the necessary steps to demonstrate the feasibility of integrating the LoRaWAN and OPC UA protocols embedded in a network device to then be used as another communication option for industrial environments, within the context of Industry 4.0 and IIoT technologies.

The results obtained in the tests of the LoRaWAN network demonstrate the promising characteristics of the protocol in terms of range, cost and ease of implementation. The results obtained in the range test show that the technology can be used in large industrial parks where the diversified structure of tanks, pipes and concrete buildings do not represent major impediments to signal collection. By using appropriate antennas, in addition to ensuring the correct location of the gateway nodes and sensors, it is possible to eliminate several obstacles to the signal, thus contributing to the creation of a cyber-physical space.

With regard to OPC UA, the present work demonstrated one of its main characteristics, i.e. the independence of platforms, through the compilation and execution of an OPC UA server embedded on the same hardware of the LoRAWAN gateway, contributing to the reduction of costs in infrastructure. Software licensing is free due to the choice of Open62541 open source libraries and the OpenWRT system, also open source and present in several network infrastructure devices, constantly being developed by the developer community.

The LoRaWAN and OPC UA protocols worked properly and efficiently, implemented separately. The integration between both also took place successfully and, in order to demonstrate the feasibility of the integration, temperature measurements were carried out, which is a quantity common to the industrial environment, focusing on the representation of these data in a SCADA interface that includes the software for supervision and operation in use for decades across the industry. Presenting this data at the SCADA level reliably validates the proposed integration.

References

Bursi, A., OpenWrt. Using the SDK, 2020. In: https://openwrt.org/docs/guide-developer/using_the_sdk. Accessed in: 10 October 2020.

Commserver, UA System concepts, 2010. In: https://cas.eu/index.php. Accessed in: 01 August 2019.

Compology, A case study: Peninsula Sanitary Service Inc. (PSSI) implements container monitoring sensors. Compology Waste & Tech, 2017.

Dragino, LPS8 LoRaWAN gateway user manual 1.0.2. Dragino Technology, 2019.

Imtiaz, J., Jasperneite, J., CAPRI: Common automation protocol architecture and real-time interface. Lengo: Institut Industrial IT der Hochschule Ostwestfalen-Lippe, 2013.

LoRa Alliance, LoRa Alliance Technical Committee Regional Parameters Workgroup. LoRaWAN 1.0.3 regional parameters, 2018.

LoRa Alliance, LoRaWAN: What is it? A technical overview of LoRa® and Lo-RaWAN™, 2015.

LoRa Alliance, LoRaWAN™ backend interfaces 1.0 specification, 2017.

Mahnke, W., Leitner, S. H.; Damm, M., OPC unified architecture. Berlin, Springer, 2009.

Microchip, LoRaWAN™ library plug-in for MPLAB® code configurator user's guide, 2017.

Microchip, Release notes for MPLAB® code configurator library LoRaWAN 01.10.00_beta, 2016.

Pursche, Fabian. Container 4.0: smart transport on the high seas. Bosch Connec-tedWorld Blog. Berlin, 5 September 2016. In: https://blog.bosch-si.com/industry40/container-4-0-smart-transport-high-seas/. Accessed in: 27 November 2018.

OPC Foundation, OPC unified architecture: interoperability for industrie 4.0 and the internet of things, 2017.

Open62541. Open62541 is an open source C (C99) implementation of OPC UA licensed under the mozilla public License v2.0, 2019. In: https://open62541.org/. Accessed in: 11 April 2020.

Rangel, R, Internet das coisas, nova revolução da conectividade. *Revista Inovação em Pauta, Rio de Janeiro*, n. 18, p. 4-7, 2014.

Schroeder, W., Germany's industry 4.0 strategy: rhine capitalism in the age of digitalisation. London: Friedrich-Ebert-Stiftumg, 2016.

Semtech Corporation. AN1200.22: LoRa modulation basics - Applaction Note, 2015.

Tanenbaum, A. S., Computer Networks. 5$^{th.}$ Edition. Pearson Education, Inc., 2011.

Unified Automation. Address Space Concepts. *Embedded OPC UA stack documentation: Unified Automation*, 2020. Disponível em: https://documentation.unified-automation.com/uasdkdotnet/2.5.2/html/index.html. Accessed in: 26 January 2020.

Wich, J. P., OpenWrt. Welcome to the OpenWrt Project, 2020. Disponível em: https://openwrt.org/. Acesso em: 12 April 2020.

7. Biographies

Angelo Guimarães is Automation Analyst at Concert Cloud, graduated in Control and Automation Engineering from the Federal University of Minas Gerais. Self-taught researcher in IIoT, industrial networks and LoRaWAN. He has experience in digital and analog electronics, C programming, Java, Python, product engineering, instrumentation, microcontrolled systems, telemetry, 3D modeling, mechanical design and assistive technology.

Hugo Michel is Adjunct Professor in the Department of Electronic Engineering at the UFMG School of Engineering, teaching disciplines in the areas of Automation and Industrial Instrumentation. He earned B.S. in Electrical Industrial Engineering from the Federal Center for Technological Education of Minas Gerais and Masters in Electrical Engineering (concentration area: Signals and Systems, research line: Process Control and Automation System) by the Federal University of Minas Gerais (2010). Its main area of expertise involves works related to instrumentation and automation in industry, sensor networks without industrial and home automation.