ICIC Express Letters
Part B: Applications
Volume **11**, Number **7**, July **2020**

ICIC International ©2020 ISSN 2185-2766
pp. 623–630

STUDY AND IMPLEMENTATION OF A REAL INDUSTRIAL APPLICATION INVOLVING CONCEPTS OF INDUSTRY 4.0

Adriene Zilda Corrêa Magalhães[1], Alexandre Baratella Lugli[1]
and Tales Cleber Pimenta[2]

[1]Department of Industrial Automation
National Institute of Telecommunications (INATEL)
CEP 37.540-000, Santa Rita do Sapucaí, MG, Brazil
adrienemagalhaes@gea.inatel.br; baratella@inatel.br

[2]Institute of Engineering System and Information Technology
Federal University of Itajubá (UNIFEI)
CEP 37.500-903, Itajubá, MG, Brazil
tales@unifei.edu.br

Received January 2020; accepted April 2020

Abstract. *The project is an action research that is based on the engineering method, applied to industrial automation. Its proposal is to study and implement an industrial application involving concepts of Industry 4.0. A practical application was developed to control a cooling system. A Programmable Logic Controller (PLC) performs the temperature control. The data measured by the controller can be viewed through the Human Machine Interface (HMI), by the supervisor developed in a supervisory system and in the mobile application. Despite the setbacks and difficulties, the project presented a satisfactory result.*
Keywords: Application, Communication protocols, Industry 4.0, PROFINET

1. **Introduction.** The first industrial revolution took place in the late 17th century, especially the steam engine. Since then, more and more revolutions have occurred. Currently, it is the fourth industrial revolution, marked by the convergence of digital, physical and biological technologies [1].

With the spread of the Internet, access to information and the way people relate to it have been revolutionized. This enabled the fourth industrial revolution, also called Industry 4.0, to bring together intelligent machines, advanced computational analysis and collaborative work among connected people to drive profound change and bring operational efficiency to diverse industrial sectors: manufacturing, transportation, energy and health [1,2].

Consumers are looking for high quality products, especially in the food area, as it is a basic necessity for life. Milk is one of the foods that are used in the different stages of life, as they form a group of foods of great nutritional value, since they are considerable sources of proteins with high biological value, as well as containing vitamins and minerals [3,4]. The processing of milk and its derivatives at industry level requires the correct cooling, because the temperature of milk at the time of milking is very favorable to the multiplication of germs and therefore detrimental to the manufacture of derivatives. The cooling of milk is a very effective measure with respect to the containment of acidification caused by bacteria [3,4].

Industry 4.0 uses industrial networks to link different wired and wireless communication network technologies [2]. Industrial networks aim to define standards for standardization

DOI: 10.24507/icicelb.11.07.623

of protocols [5]. Therefore, devices from different manufacturers can communicate with each other [5].

The purpose of this paper is to demonstrate an industrial application that reads a temperature sensor and monitors a milk cooling plant, controlling an actuator through Industry 4.0 concepts, in order to assist the producer in milk production and qualification.

2. **PROFINET Protocol.** Industrial Ethernet emerged to unify the way it sends data from the fieldbus level [6], therefore creating protocols that companies should follow for a satisfactory operation of their application.

The PROFINET protocol is the open Industrial Ethernet standard of PROFIBUS & PROFINET International (PI) for automation. It is specified in IEC 61158 and IEC 61784 standards. It originated from Industrial Ethernet and uses Transmission Control Protocol/Internet Protocol (TCP/IP) and Information Technology (IT) standards. It also allows integration with other field networks [6,7].

There are three types of distinct data access contained in only one protocol. These are TCP/IP, Real-Time (RT) and Isochronous Real-Time (IRT). TCP/IP is used for parameterization, configuration and diagnostics. Real-Time is used for cyclic transmission of process data, alarm and event signaling. Isochronous Real-Time uses cyclic and synchronized process data transmission when state accuracy is required [6,7].

The network topology influences several critical points, such as flexibility, speed and security. The PROFINET network is considered efficient because it integrates the various ways to connect components, supports switches and works with redundancy. It can work with several topologies, such as inline, components are interconnected; star, requires a central switch to which field devices are connected; in a ring, a communication line is closed to form a ring for cable redundancy; and, in tree, all the mentioned topologies are combined [7,8].

Devices connected to the network can be differentiated by categories. The IO controller is responsible for exchanging input and output signals with field devices, IO devices are the field devices and supervisor that allow the user to view data on a computer, Personal Computer (PC), or a Human Machine Interface (HMI), for startup or diagnostic purposes [8].

In a PROFINET network, the number of nodes can be determined according to the controller. The amount of information that can be transmitted in the same telegram is 1.440 bytes. A telegram provides information about the validity of the data, the state of redundancy and the state of diagnostics. It is based on a 100 Mbps full-duplex Ethernet network. Therefore, it is possible to use high transmission rates over the entire network [8,9].

3. **Practical Application.** The purpose of this paper is to study and implement an industrial application. It proceeds from reading a temperature sensor, and monitoring it, and controlling the speed of a motor through a frequency inverter. To initiate the motor, it is needed to activate an ON/OFF switch or press the direction via the HMI (Human Machine Interface). The HMI displays the temperature and motor speed graph. The data may also be accessed through an application, which provides the concepts of Industry 4.0.

For the development of this work, devices were used that were able to perform the communication by the PROFINET protocol. The TIA PORTAL computational application (configuration software) connects the master (PLC – Programmed Logic Controller) to the external read and actuation drivers. Figure 1 illustrates the diagram of the application performed.

The system operates through the temperature received by the sensor: the higher the temperature, the higher its frequency. In HMI, in addition to displaying temperature and

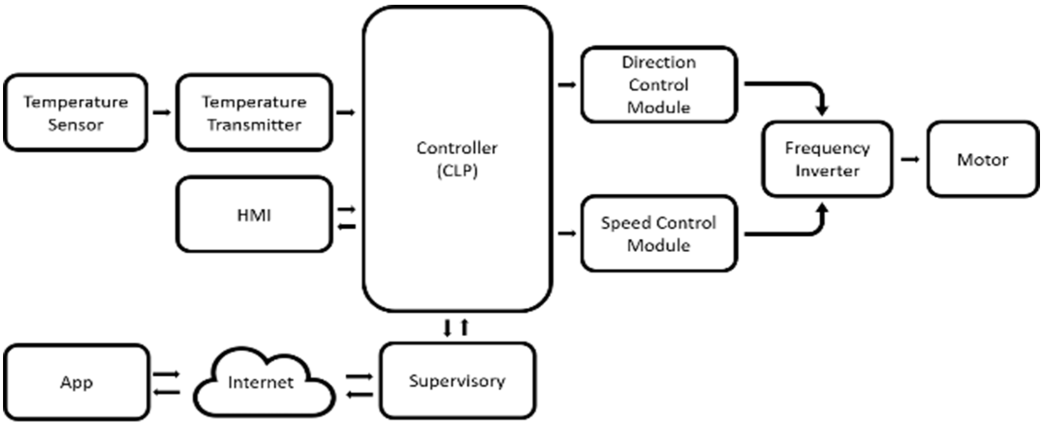

FIGURE 1. Industrial application diagram

frequency, it is possible to select the direction of motor rotation. The application was projected in *Elipse Mobile* and it is applied for remote system supervision.

3.1. **Configuring the PROFINET network.** The SIMATIC S7-1200 PLC is configured as an IO controller to perform the master role by establishing commands on its peripherals. It is needed to configure a PROFINET network assigning individual addresses to the input and output devices. For this purpose, the IP address is employed to include a network address for the device and a subnet mask indicating which parts of the IP address will be used as a network address or as a device address. The addressing of the other devices is illustrated in Table 1.

TABLE 1. Address table

Equipment	*IP Address*	*Subnet Mask*
S7-1200 PLC	192.168.0.70	255.255.255.0
Switch	192.168.0.1	255.255.255.0
HMI KTP400	192.168.0.2	255.255.255.0
Siemens module	192.168.0.3	255.255.255.0
B&R module	192.168.0.4	255.255.255.0

3.2. **Topology.** The applied topology was a mixture of the star and switch-centric topology of the input and output devices. Figure 2 illustrates the representation of the assembled network in application.

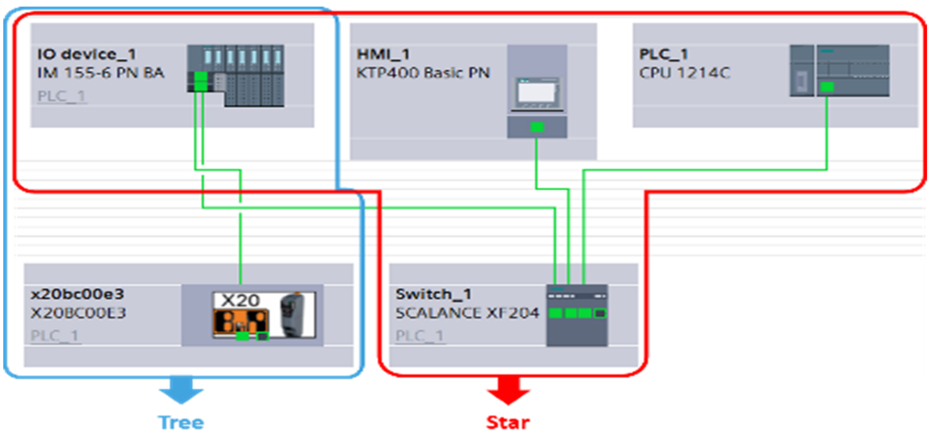

FIGURE 2. PROFINET network topology

3.3. **Application program.** In the controller is embedded the logic program that will act on the devices. For this, variables are created to store data on the input and output. They are illustrated in Table 2.

TABLE 2. Variable table

Name	*Type*	*Variable*
Clockwise_On	Bool	M0.6
Anticlockwise_On	Bool	M0.7
Temperature	Real	MD100
Frequency	Real	MD104
Sensor	Real	IW78
Frequency_Inverter_Output	Int	QW65
Clockwise	Bool	Q2.0
Anticlockwise	Bool	Q2.1

Ladder programming language is used for project implementation. In the configuration software, the analog input port is modified to receive a standard 4-20mA signal from the transmitter, and the analog output port from -10 to 10V, in which it will be transmitted to the frequency inverter to the control motor speed. The signal that the controller delivers to the sensor variable is between 0 and 32767. This value must be converted to a known temperature value. The temperature transmitter has been set from 0 to 100°C. Therefore, to convert it, a division block is inserted in order to parameterize the signal. Figure 3 illustrates the expression to perform the conversion and the program to convert from the input value to temperature.

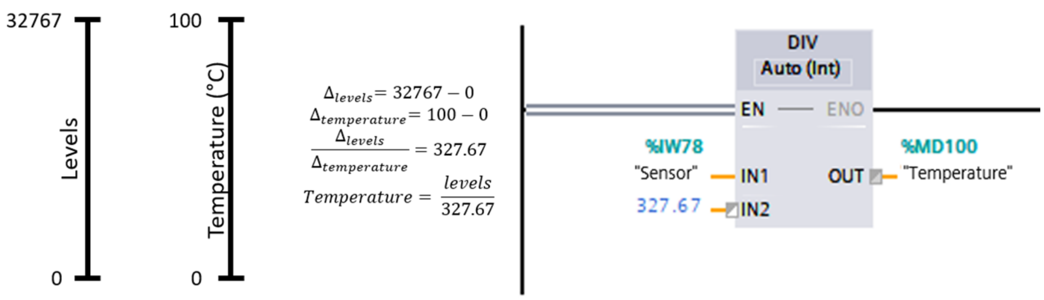

FIGURE 3. Conversion levels to temperature

The frequency inverter signal originates from the temperature signal. Therefore, to visualize a known signal, it is needed to transform the signal from 0 to 32767 (PCL signal) into 0 to 60Hz (frequency inverter signal), where it is converted through the expression illustrated in Figure 4.

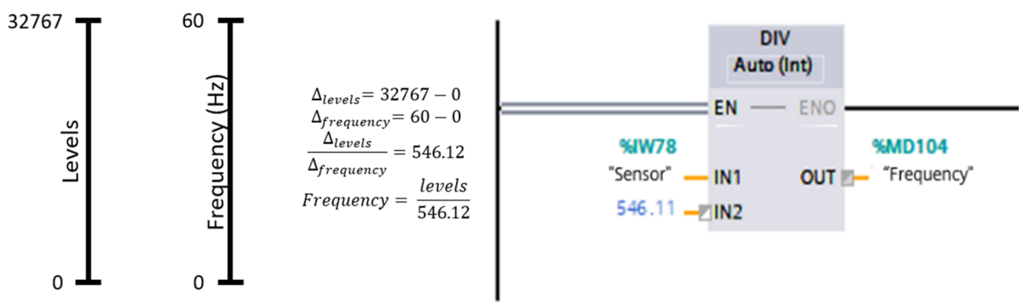

FIGURE 4. Conversion levels to frequency

The frequency signal used to control the motor is 0 to 60Hz, or a voltage 0 to 10V at the inverter input. This voltage can be delivered from the analog output, sending levels from 0 to 32767. Figure 5 illustrates the line of code that sends the input values to the frequency inverter.

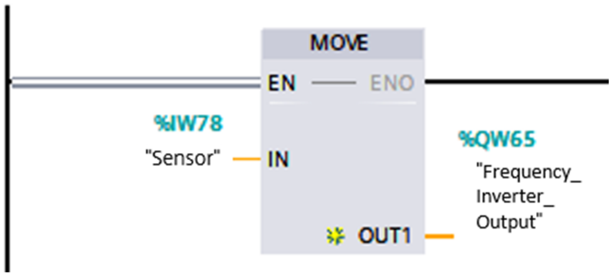

FIGURE 5. Value to inverter frequency

The motor direction rotation is chosen by a command given at the HMI. If the clockwise/counterclockwise button is pressed, the state on the digital port is changed to high level. Figure 6 illustrates the direction of rotation control lines.

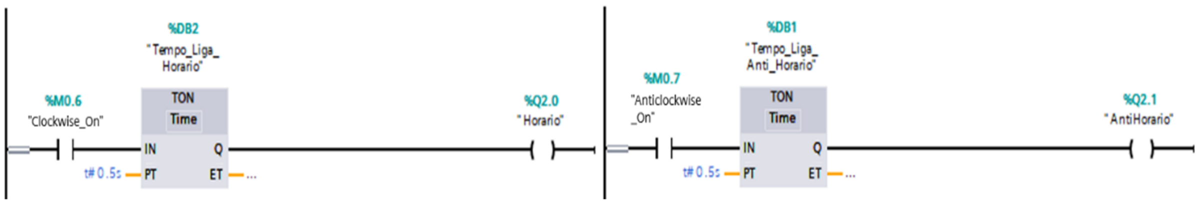

FIGURE 6. Motor direction drive lines

3.4. **Temperature conversion.** Some parameters must be configured to allow the automation system to access the supervisory and to integrate the configuration software to the Eclipse E3 software. The supervisory system must have communicated to a block of data that will have access to the variable to be supervised. When accessing configuration software in controller programming, a new data block is added.

Two blocks were inserted, one for transmitting temperature and one for frequency. For the supervisory to access the data block, the property "*Optimized block access*" must be deselected in the attributes tab of the data block property. In the controller, you must configure the device protection by enabling the options "*Full access (no protection)*" and "*Allow access with Put/GET communication from remote partner*".

In the blocks, floating type variables were created to store the temperature and frequency data to be supervised. To transmit the values, it was inserted the following command lines illustrated in Figure 7.

FIGURE 7. Communication lines for supervisory

3.5. **Communication application.** Communication between controller and *Elipse E3* software is done using the *Multiprotocol Driver* (M-Prot) and the *ISOTCP* protocol directly on the Ethernet port.

The driver has several settings that can be modified to meet the project requirement. In the MProt tab, the network type and rack number are changed, in the setup tab the physical layer for Ethernet, and in the Ethernet tab the protocol type and the physical address of the controller. Subsequently, two communications tags that store the temperature and the frequency were created, and they are illustrated in Figure 8.

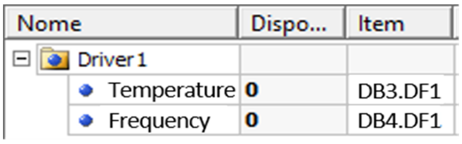

FIGURE 8. Tags configuration

3.6. **Mobile platform.** The application created is the visualization of temperature and frequency data and can be remotely accessed.

The installation of *Elipse Mobile* is divided into two modules: *Server* (for the application server) and *Client* (for viewing via smartphones, tablets or browser). *Server* is responsible for managing services, connections and ports, and *Client* operates as a visualization and application management and editing module [10].

After installing the *Server* module on a local computer, a browser will open, where the application will be hosted.

Elipse Mobile allows the creation of multiple applications within the same installation, but only one execution at a time is allowed. In the *Applications* it is to view the available applications and manage them [10].

The connection to *Elipse E3* can be made through a native connection present in Mobile, providing a better response because the connection is not through *Windows DCOM* [10].

On the home page, the objects that make up the operator interface with the system will be inserted. In the application created, two displays were inserted, an object for displaying values. To associate the temperature and frequency values, the tag property must be set with the *Elipse E3* connection values. Figure 9 illustrates the application developed in this paper as *Server* and as *Client*.

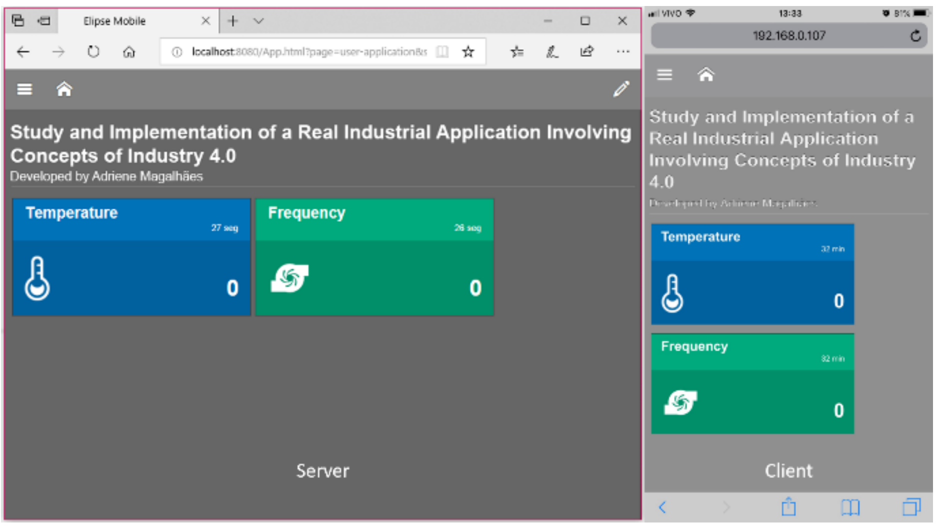

FIGURE 9. Server and client screen

The application is hosted on a virtual server named local host. *Elipse Mobile Server* for default access port is 8080. It was used the remote access with the IPv4 address, which consists of an Internet protocol in its fourth version.

The command prompt is used to locate the address. Entering the command "*ipconfig*" various settings of the computer will be displayed. On the wireless network adapter, there will be the IPv4 address, illustrated in Figure 10.

```
Wireless Network Adapter:

   Connection-specific DNS Suffix . . . . . . . . . :
   Local link IPv6 Address  . . . . . . . . . . . . : fe80::e9cc:9357:b8e1:88e7%13
   IPv4 Address . . . . . . . . . . . . . . . . . . : 192.168.0.107
   Subnet Mask  . . . . . . . . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . . . . . . . .  : 192.168.0.1
```

Figure 10. Connection network validation

This IP address can be hosted on a dynamic Domain Name System (DNS) provider to have a website hosted on the Internet.

4. **Results.** Assemblies and components were performed with the development kits and they are illustrated in Figure 11.

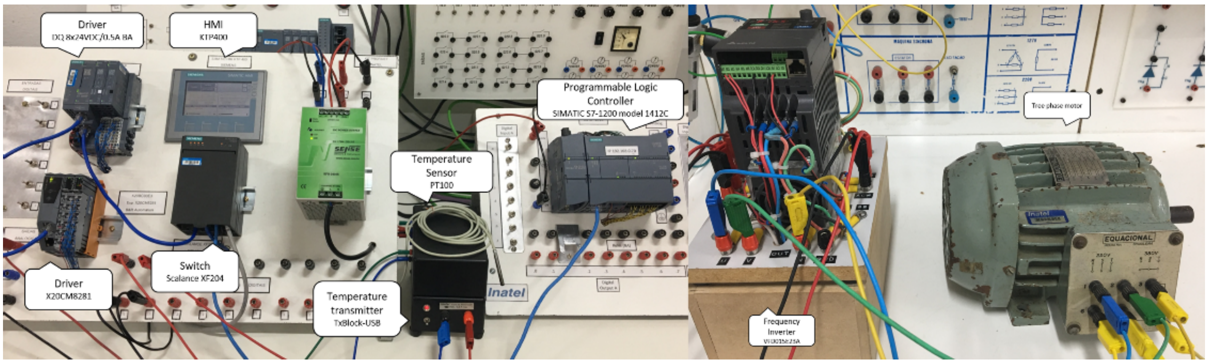

Figure 11. Application assembly

After assemblies, tests were made to see if the system was working properly. The sensor was heated to check the frequency and temperature variation by the supervisor and the application. The result can be illustrated in Figure 12 and Figure 13. The system has a good accuracy in the process variables (temperature and frequency), with error less than 0.1%. There is still a network delay between wireless system and process system (around 1 second) that was not measured in this paper.

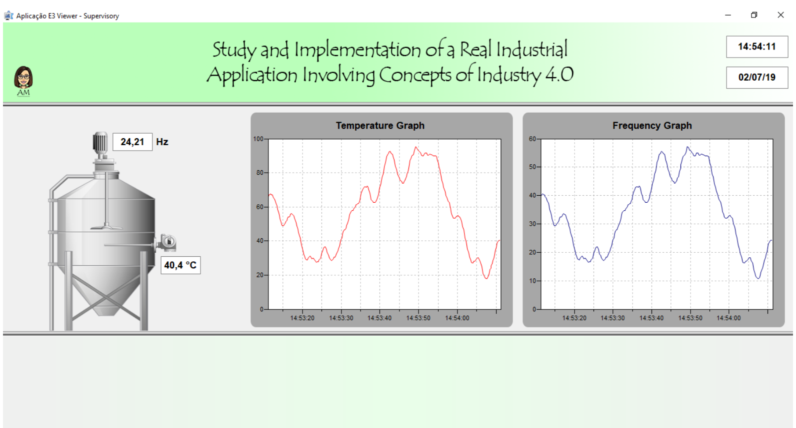

Figure 12. Supervisory project view

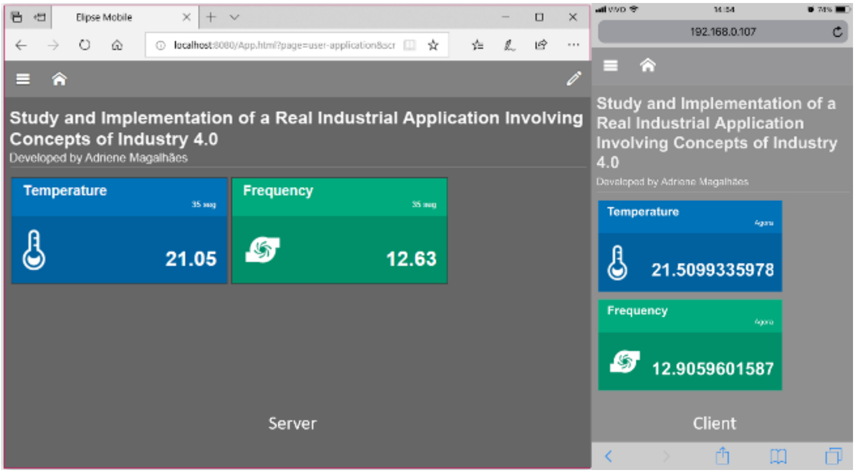

FIGURE 13. Mobile supervisory view

5. **Conclusions.** Industry 4.0 aims to produce efficient methods and quality of life. Among several concepts used, it appears that a security protocol is necessary, because any data that goes the wrong way can harm the company.

When performing the assembly, it was found that according to the temperature of the sensor, the graphs were plotted correctly in both the HMI and the supervisory, and can be viewed in the mobile application.

To perform the test between the supervisory and the practical application, it was needed that the controller was connected to the local network. At the command prompt, the command "Ping" was sent, an application that uses the ICMP protocol to test connectivity between devices. If the response is returned, it means the application is on the network, so the supervisor will receive the data correctly.

The motor controlled by the frequency inverter varied its speed proportionally to the temperature.

An improvement of the project would be to acquire a license for *Elipse Mobile*, as in the free version only two communication tags are allowed. Another analysis that may be performed in future work would be the temporal analysis of communication latency of the application with the controller that was not addressed in this work.

REFERENCES

[1] V. Perasso, *What Is the 4th Industrial Revolution and How It Should Affect Our Lives*, BBC News Brazil, https://www.bbc.com/portuguese/geral-37658309, 2016.
[2] J. Rizzo, *Learn What Industry 4.0 Is and Discover the Opportunities It Generates*, http://bit.ly/saibaoqueeind, 2016.
[3] K. S. Venturini, M. F. Sarcinelli and L. C. Silva, *Milk Processing*, Technical Bulletin of the Federal University of Espírito Santo, Espiríto Santo, 2017.
[4] O. M. S. Amancio et al., The importance of milk consumption in the current Brazilian nutritional scenario, *Brazilian Food and Nutrition Society*, pp.13-14, 2015.
[5] A. B. Lugli and M. M. D. Santos, *Industrial Networks: Evolution, Motivation and Functioning*, Inatel.br, https://www.inatel.br/biblioteca/todo-docman/producao-cientifica/artigos-cientificos/2011/8794-redes-industriais-evolucao-motivacao-e-funcionamento/file, 2011.
[6] X. Wu et al, On the wireless extension of PROFINET networks, *Proc. of the 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, p.5, 2019.
[7] M. Popp, *Communication with PROFINET*, 1st Edition, Translated by A. B. Lugli, PI Brazil, São Paulo, 2018.
[8] J. D. Decotignie, Ethernet based real-time and industrial communications, *Proceedings of the IEEE*, vol.93, no.6, pp.1102-1117, 2005.
[9] A. B. Lugli and M. M. D. Santos, *Industrial Networks for Industrial Automation AS-I, PROFIBUS and PROFINET*, 1st Edition, EditoraÉrica, São Paulo SP, Brazil, 2010.
[10] *Elipsemobile* – Overview, GitHub, https://github.com/elipsemobile, 2019.