Implementation of a Communication System Between the (IP) Internet Protocol and the Profibus Standard for General Application Using Programmable Development Systems

Cristian Caballero [1], Rodolfo Romero [1], Danilo Alfonso López Sarmiento [2]

[1.] Electronic Enginner, Graduate at Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
[2.] Faculty of Engineering, Full time professor at Universidad Distrital Francisco José de Caldas, Bogotá, Colombia
k4balleroQgmail.com, rrromero71@gmail.com, dalopezs@udistrital.edu.co, ingeniero24@hotmail.com

Abstract: The article shows the development of an interconnection application between IPv4 and Profibus protocols, trying to implement a PA fieldbus type non intrinsically safe and be monitored from the Internet using a web service called ThingSpeak, which serves as the basis for the preparation of the study concerning the protocols set forth at the beginning, appreciating the limitations of the industrial systems and their trends into the future, where they will come to merge with the Ethernet protocol, giving rise to a new generation of standardized fieldbuses.
[Caballero C, Romero R, López DA. **Implementation of a Communication System Between the (IP) Internet Protocol and the Profibus Standard for General Application Using Programmable Development Systems**. *Life Sci J* 2014;11(12):628-642]. (ISSN:1097-8135). http://www.lifesciencesite.com. 122

Keywords: Arduino; Ethernet; IPv4; profibus

1. Introduction

When dealing with industrial electronic systems, the involvement of fieldbuses and the technological developments regarding them, are apparent, therefore following the current trend of improvement and integration of technologies, the need to include the connection to Internet of these elements is imminent, mainly for remote monitoring from almost anywhere on the planet, so we are looking for a prototype that meets the specifications of the Profibus protocol when implementing a PA type plant non-intrinsically safe, that has connectivity to the network using TCP and arduino items besides converters between SPI and RS-485.

The paper summarizes the current status of field buses, jobs that have been advanced as to the communication of the latter with the Internet, the development system offered by Arduino, the methodology, design, integration and testing to achieve communication between the Profibus PA protocol and the Ethernet shield standard.

2. Theoretical framework

By treating electronics at the industrial level, interactions are conducted with some areas of physics, such as pneumatic or hydraulic therefore different groups of companies and institutions have dedicated time and effort to the integration of some devices and sensors to their communication systems, that is how it gave rise to various standards and protocols for many types of applications, obtaining models such as CAN, BitBus, Modbus, WorldFIP, Hart and PROFIBUS, currently used around the world (some are free to use), allowing to further new advances almost anywhere on the planet.

As for Profibus, this is a standard field bus that hosts a wide range of applications in manufacturing, processing and automation, guided by the EN 50-170 standard. Its main purpose is to communicate components from different manufacturers without special adjustments to their interfaces; therefore Profibus can be used for time critical transmission of high-speed data and for large and complex communication tasks [1]. Its versatility is given by these three mutually compatible versions:

PA: Designed for process automation, allowing for the connection of sensors and actuators to a common bus line, even in high-risk areas, it additionally enables the communication of data and energy by using two technologies, the first for areas with explosion hazards as stipulated in the IEC 1158-2 [23] standard and other non-intrinsically safe that uses RS-485.

DP: Optimized for high speed, using simple, low-cost connections, was designed especially for communication between automation control systems with distributed inputs and outputs.

FMS: This was devised as a general solution for communication tasks at the cell level; it supports a wide range of applications, allowing its use in complex and extensive communication tasks as the Internet.

The design of Profibus defines a network of industrial communication from the field level to the plant level. Figure 1 shows the family of this fieldbus applied throughout the CIM (Computer Integrated Manufacturing) pyramid.

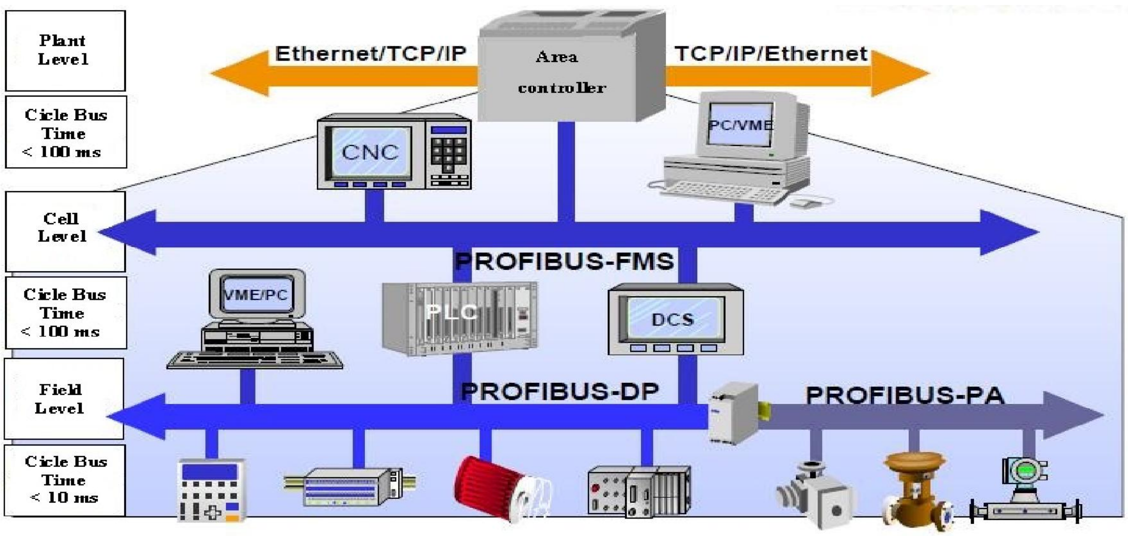

Figure 1. Profibus fieldbus applied to the CIM pyramid [1].

It should be noted that Profibus PA is at the field level where there are transducers, actuators and sensors, which require an absolute determinism with bus cycle times under 10 ms and an operability greater than 99%.

The communication structure operates three Datagram formats, one with fixed-length without data, another with fixed length and a data field of 11 bytes and one with variable length and a data field ranging from 4 bytes to 240 bytes, as shown in figure 2.

Frame Start	Destination Address	Origin Address	Function	CRC	End of Frame

3 bytes

Fixed Length Format without Data Field

Frame Start	Destination Address	Origin Address	Function	Data	CRC	End of Frame

11 bytes

Fixed Length Format without Data Field

Frame Start	Frame Length	Frame Start (2)

Destination Address	Origin Address	Function	Data	CRC	End of Frame

4 a 20 bytes

Variable Length Format without Data Field

Figure 2. Profibus Datagram Formats: with fixed length information field and without data (top), with the length of the information field and fixed data (middle part), and with variable length data field (bottom). [3]

Today Profibus is employed by the Group of the 20 most industrialized countries in the world, which provide a support for the rest of the world [2], being governed under the regulations of the International PROFIBUS Organization (PI), which has over 750 members.

On the other hand when speaking of programmable development systems, Arduino is an electronic platform for prototyping using open source based on hardware and software characterized both by its flexibility and its ease of use. In terms of its structure we can say that it is based on a plate with analog-digital type inputs and outputs, which are programmed in a development environment that is implemented from processing language [4].

Depending on the device, it uses a member of the ATmega [24] processor family, Manufactured by Atmel, which are simple and inexpensive chips, allowing the development of multiple designs.

An Arduino can sense the environment through a variety of sensors and similarly modifies actuators controlling lights, motors, controls, among others. Additionally platform projects can be autonomous or communicate with software running on a computer.

Among the advantages presented by Arduino, the ease of adaptation of the programming code should be highlighted, as long as the device on which it will be deployed has the necessary features. Currently it is common to find a wide variety plates in the market, such as the Duemilanove, Uno, Pro 328 Nano, Bluetooth, LilyPad, Mini, Mega 2560, among others [5].

3. State of the art

Advances to connect various fieldbuses to the Internet are happening slowly; in the case of Profibus the developments are presented as follows:

Evaluation of Delays Induced by Profibus PA Networks [6]: Talks about the solutions to manage active objects and their importance in modern control systems and is particularly necessary in the area of process control. The control systems used here usually consist of PLC based networks, which are optimized on the special temporal and topological requirements of the process. Asset management functions have to be introduced to the systems, without any influence on existing equipment. The paper shows an object oriented solution based on a PROFIBUS PA device. Automatically evaluating a structure of a mapping system and the relevant parameters for the profiles of the proxy devices, different application modules for asset management functions can be implemented acting on these objects.

Integration of TCP/IP and PROFIBUS Protocols [7]*:* Presents the architecture of a solution proposal to incorporate an application support based on the TCP / IP protocol on Profibus networks. The solution provided guarantees the timing requirements and traffic control to support QoS (Quality of Service) for the multimedia applications being implemented in the interface of Windows NT NDIS.

Among the considerations in this paper indicate that the integration of PROFIBUS and TCP / IP must be correctly specified, to provide not only the appropriate QoS for applications that support TCP / IP but also ensure the time requirements of Profibus traffic control. Also the following details should be addressed: being a solution that fits the paradigm of master / slave fieldbus to the symmetrical nature of IP networks, and also the integration should be transparent from the point of view of the application.

Real-time property analysis of single master DP/PA hybrid network control system [8]*:* It is a Profibus PA system that is built from a Profibus DP network, differentiating their transmission medium, the speed of transmission, the data frame format and so on were different question. We take the typical construction of a network outlet with a control system with a hybrid DP / PA master as a research topic, to make a proposal for a method of improving the performance control in the network and create favorable conditions for further analysis.

4. Results

4.1. Results of implementation

Considering that the main purpose of the physical implementation is to generate a system which allows connectivity for a Profibus PA type device without intrinsic safety to the Internet, resulting on the scheme shown in figure 3, which show five relevant blocks, which are the plant, the master, the bus, the slave and the Ethernet card (which will connect to the router to access the World Wide Web).

From this point we performed a series of tests to build and validate the progress to get the Profibus prototype, so it is recommended to have in mind the previous representation to more easily understand how was performed the inclusion sequence of the blocks shown there.

We should give special emphasis to mention that when using LCD devices, these are used as a display mechanism of the local variable (measured and / or transmitted) in each of the bus terminals. For the digital to analog converter (hereafter CAD), it is responsible for capturing the value of the evaluated variable, while a pair of pins (0 and 1) allow for the SPI communication with other elements (such as MAX485), in this case enabling the construction of master and slave (where a pair of Arduino UNO are used), as in the present development. Finally the Arduino Ethernet Shield card (inserted directly on the master) provides connectivity to the Internet via a network cable that connects directly to a router.

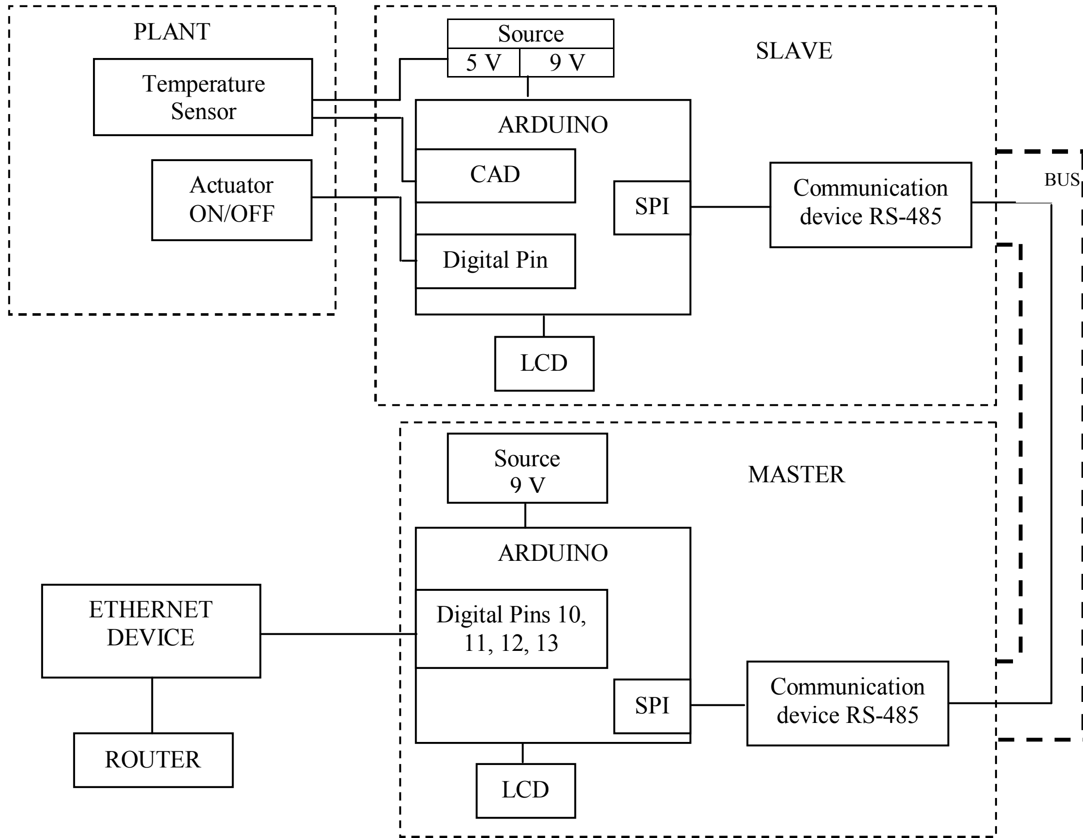

Figure 3. Block diagram of the implemented prototype.

From this point we performed a series of tests to build and validate the progress to get the Profibus prototype, so it is recommended to have in mind the previous representation to more easily understand how was performed the inclusion sequence of the blocks shown there.

We should give special emphasis to mention that when using LCD devices, these are used as a display mechanism of the local variable (measured and / or transmitted) in each of the bus terminals. For the digital to analog converter (hereafter CAD), it is responsible for capturing the value of the evaluated variable, while a pair of pins (0 and 1) allow for the SPI communication with other elements (such as MAX485), in this case enabling the construction of master and slave (where a pair of Arduino UNO are used), as in the present development. Finally the Arduino Ethernet Shield card (inserted directly on the master) provides connectivity to the Internet via a network cable that connects directly to a router. So once given these notions, we have the following partial results:

4.1.1 Serial display. By employing the Arduino UNO after being programmed, the serial visualization produced by the device is used and by using the USB type cable (which energizes it and is later replaced by voltage sources) performs a communication with the computer via this port, that can monitor the behavior of the card (figure 4), taking advantage of a free tool provided by the manufacturer of the software (the serial monitor), that works at 9600 baud and 19200 bps, which are recognized by the computer.

To program the Arduino card is necessary to download the software of the same name (which in this case is the 0022 version), proceed to connect the board and install the device driver for the operating system used (following the manufacturer's instructions) then open the program and follow to the area code setting, initialize the serial port with the instruction "Serial.begin (9600)" where the number represents the target transmission rate, which in this case will be recognized by the computer.

Later the cyclical part is established, which includes the "Serial.println (40)" and "delay (500)," the first shall be responsible for printing the value to the serial port and give a line break, while the latter generates a pause of five hundred milliseconds, and then restart the loop, getting a result like the one in figure 5. Finally the hardware is programed and the visualization tool is enabled, the process can be synthesized in flow diagram shown below.

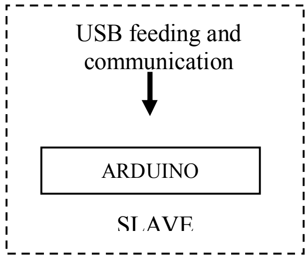

Figure 4. Diagram of the system for visualization.

From this point we performed a series of tests to build and validate the progress to get the Profibus prototype, so it is recommended to have in mind the previous representation to more easily understand how was performed the inclusion sequence of the blocks shown there.

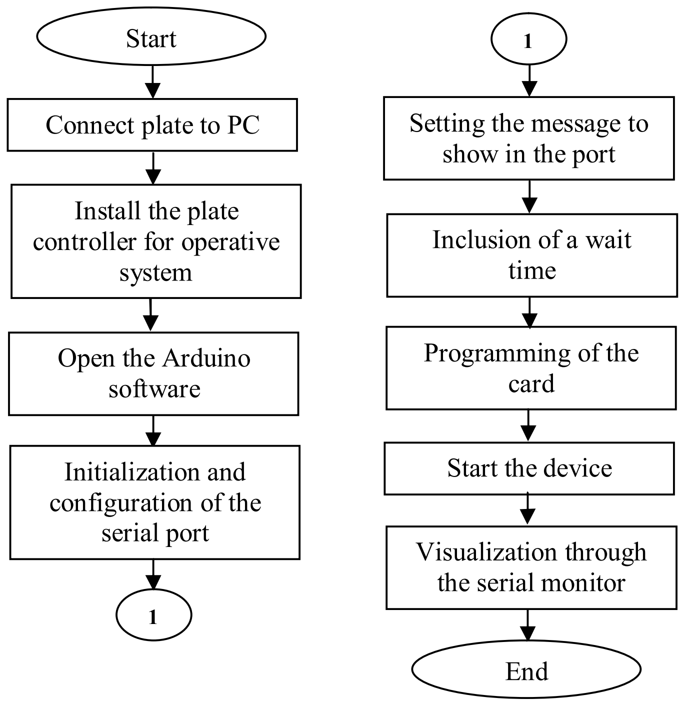

Figure 5. Serial Visualization Flowchart with board UNO.

4.1.2 CAD Reading and LCD visualization. The value of a potentiometer is captured (which temporarily emulates the sensor for its high current), showing on a LCD matrix [20] of 16x2 (figure 6) to supervise it without a computer, which is essential in future tests, because the Profibus protocol has a transmission rate of 31.25 kbps and it is not possible for the serial monitor to work at that speed.

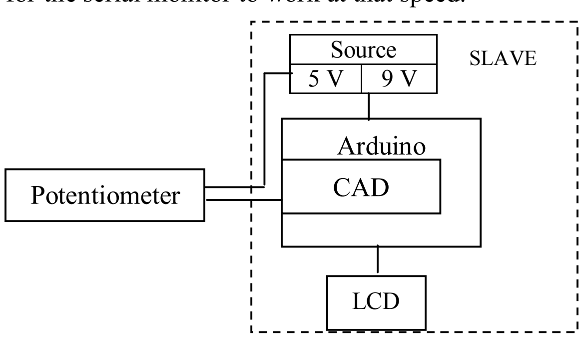

Figure 6. Diagram of device using the CAD and visualization with LCD.

4.1.3 Serial Communication SPI Master-Slave with LCD display. Using serial communication to form a master-slave system, the second captures the data from the potentiometer (through its respective CAD) sending it to the other board, checking their status with LCD displays at each terminal. It is important to note that pins 0 and 1 of the Arduino UNO correspond to serial reception and transmission, respectively (as shown in figure 7, with images based on a simulator called Virtual Breadboard or VBB [9]). Clarifying that for the communication from master to slave the connection is shown with the blue line, while for the opposite operation it is with the red. In figure 8, the mounting block diagram shown above.

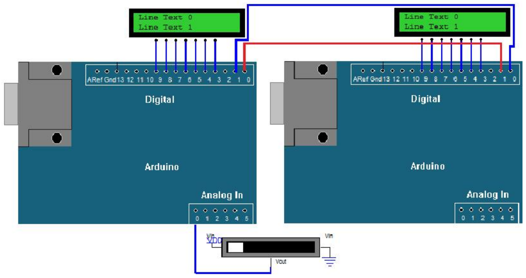

Figure 7. The Arduino device with a potentiometer connected to the slave CAD and LCD display

It should also be noted that the partial developments drove a sequence where data is transmitted, then modifying the code is passed to handle a chain of these and finally after further adjustments for time, bidirectional communication is to finally change the speed specified by the Profibus protocol.

4.1.4 Serial Master-Slave Communication RS-485 with LCD display. Because the Arduino devices handle SPI communication, we use a conversion from this to RS-485 and vice versa, provided the protocol Profibus PA without intrinsic safety specifications of the physical layer (see [10] and [11]) where as specified in the TIA / EIA-485 standard, you must use a shielded twisted-pair 20 to 24 AWG depending on the specifications of current and the line impedance present (typically 100-120 ohms), having to put a termination resistor at the end of the bus to prevent signal reflections.

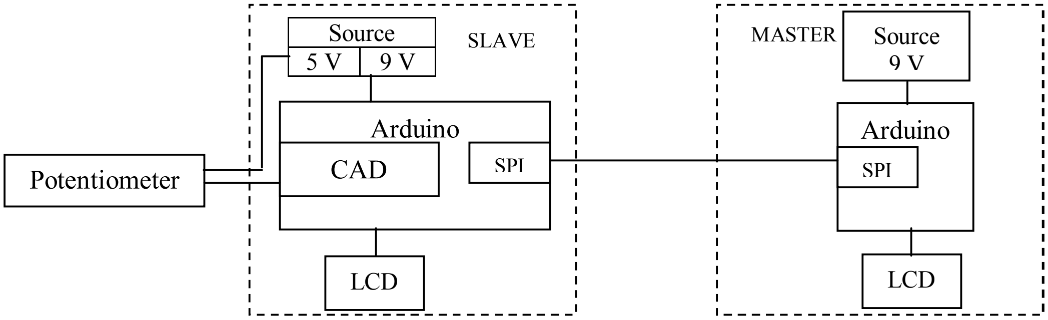

Figure 8. Block diagram of the master slave system connected by SPI.

The Integrated MAX485 of MAXIM (see [12]), which is responsible for making the bidirectional conversion previously stated, alternating the direction of operation by a digital signal at its pin 2 and 3, for an encapsulated DIP / SO as presented in figure 9.

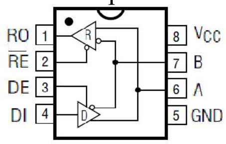

Figure 9. Internal configuration of MAX485 [12].

Similar to the developments with SPI, a bidirectional data chain is reached, all driving the LCD display and varying transmission rates to the speed set by Profibus 31.25 kbps, obtaining a diagram of the physical connectivity way of different modules used (figure 10).

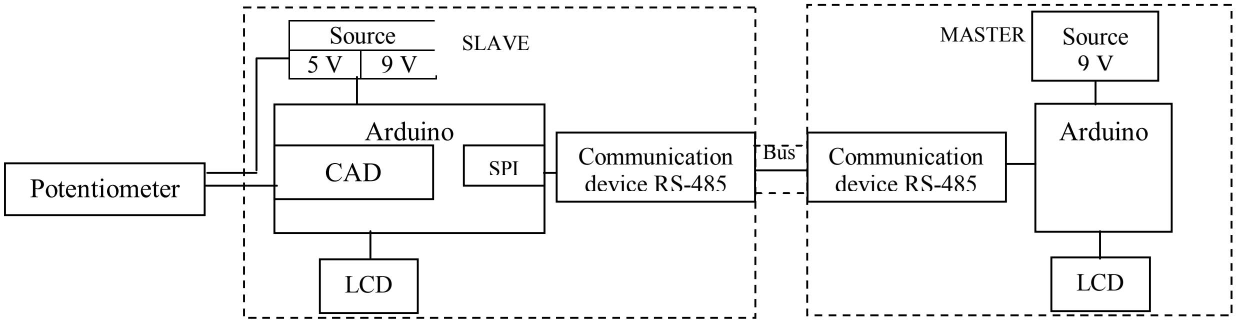

Figure 10. Block diagram of the slave master system using RS-485.

4.1.4 CAD Reading, connection and Ethernet display. Using the Arduino Ethernet Shield SD [25], it gives an expansion of the functionality of the motherboard where it is installed, allowing a connection to the network, by using pins 10 to 13 to communicate between them.

In the scheme of figure 11 a potentiometer to which the voltage value is captured is shown, generating a data, which is then sent by the Ethernet card to the network, in the way that will be explained below.

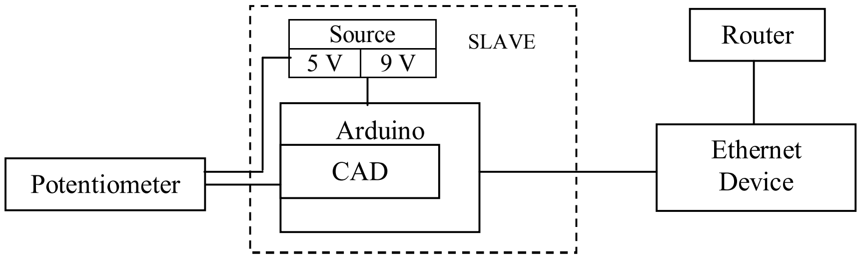

Figure 11. Block diagram of the system to display the Ethernet board.

For this the UNO card is independently programed with a procedure similar to that described in the diagram of fgure 5, however when forming the source code, further initialization must be performed on the Ethernet module using the "Ethernet.begin (MAC , IP, gateway, subnet)" instruction whose values vary depending on the network where the connection is made, then depending on the type of application it can be used as a server or a client, the latter is the one used for the project requirements. Once the card is programmed, it is inserted in the network module and connected to both the power supply and the router, to then display the information via a web browser. The procedure described above is shown in figure 12.

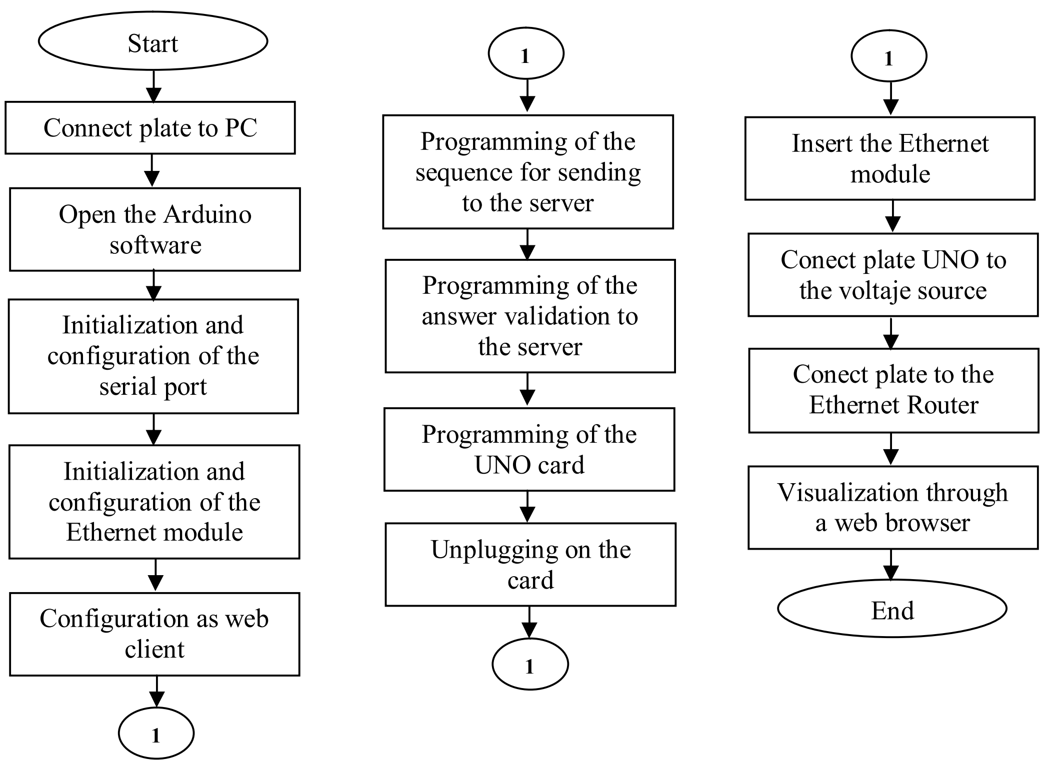

Figure 12. Flowchart displaying the Ethernet board.

4.1.5 CAD reading, connection and ThingSpeak visualization and LCD. By integrating Ethernet Shield SD module we are connected to the Internet for viewing through a web page, including in this last part of the development to use the service ThinkSpeak [13], which receives the information sent by the board, storing it in an account and returning to the user a historical graph (as in figure 22). Significantly, the connection with the page is given in a Web client configuration, routing data to the account through a single key named API KEY, and through the "client.print *(" X-THINGSPEAKAPIKEY* command: "Insert here the API KEY" \ n *")"* Making the ID to access and modify the database. Furthermore we use hardware similar to figure 11, differing only by the inclusion of an LCD, used in the verification of local Arduino information.

4.1.6 Serial Communication SPI Master-Slave with ThingSpeak display and LCD. Merging the Ethernet card with the master-slave system, performing an SPI communication, similar to previous developments, and to adapt the code for the use of the plate, both times managed in coordination applications, obtaining implementation physics as presented in figure 13. It is clear that the block diagram is very similar to figure 7; however the added connection to the Ethernet device is performed on the master.

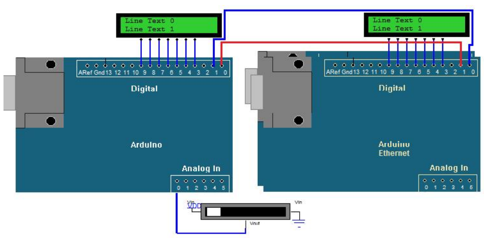

Figure 13. Arduino Device with Ethernet board in the master and SPI communication.

4.1.7 Master-Slave Serial Communication RS-485 with ThingSpeak display and LCD. Devices to the RS-485 communication to the hardware previously implemented (figure 14), with the respective settings code for MAX485 operate, and maintain two-way communication, becoming a model as depicted in figure 3, however are used has no actuator or sensor.

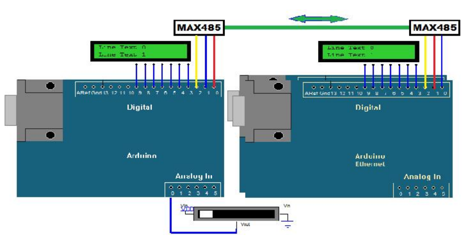

Figure 14. Arduino Device with Ethernet board in the master and RS-485 communication.

4.1.8 Master-Slave Serial Communication RS-485 with ThingSpeak display and LCD using thermal plant. Construction of a thermal power plant (figure 15) is performed and the LM35 [18] was used, which has a linear behavior in the range you want to handle (10 to 60 ° C); plus an actuator (tungsten bulb); to raise the internal temperature of the oven was included.

Figure 15. Top view of the plant.

4.1.9 Final Prototype. The hardware uses a pair of Arduino UNO plates, for information processing and SPI communication (gaining a master-slave type system), plus an equal amount of MAX485, which allows conversion between that standard and the RS-485 (figure 16). Additionally using another Arduino module family, known as "Ethernet Shield SD", which is assembled directly on the UNO plate manually to handle the same pin distribution? On the other hand you have an ON / OFF control to handle an actuator (light bulb) responsible for maintaining the reference temperature (figure 17 and 18).

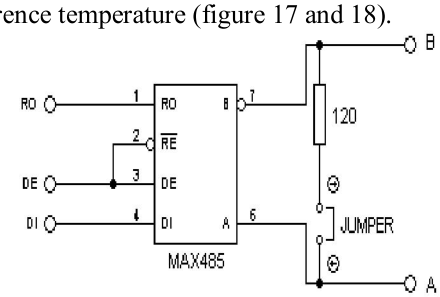

Figure 16. Converter using MAX485.

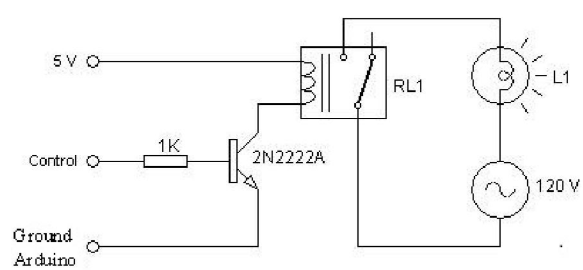

Figure 17. Actuator Control

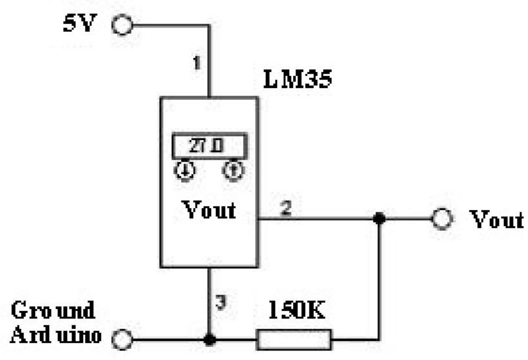

Figure 18. Sensor connection of the circuit.

Finally to obtain the final prototype (figures 19 and 20) according to figure 2, use of a web service (ThingSpeak) is done to monitor the sensor data in time intervals, generating a graph, where the induced disturbances are evident on the ground, and appreciated in detail as the account administrator. However, when visitor access is activated, this only shows a single trend line without further details on the behavior of the sensor (in this case) and is shown in figure 21 and 22.

In figure 20 the final developed argument (to differentiate components includes a numerical nomenclature): 1) The plant; 2) sensor; 3) Arduino master; 4) Arduino slave; 5) LCD teacher; 6) LCD slave; 7) RS-485 master converter; 8) RS-485 slave converter; 9) Actuator Circuit; 10) twisted pair of the bus.

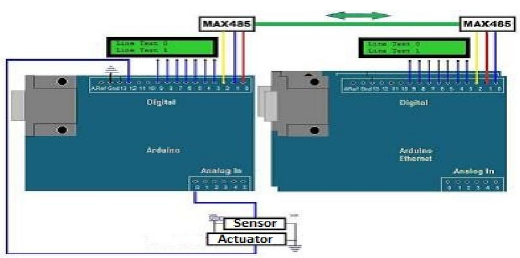

Figure 19. Outline of final prototype.

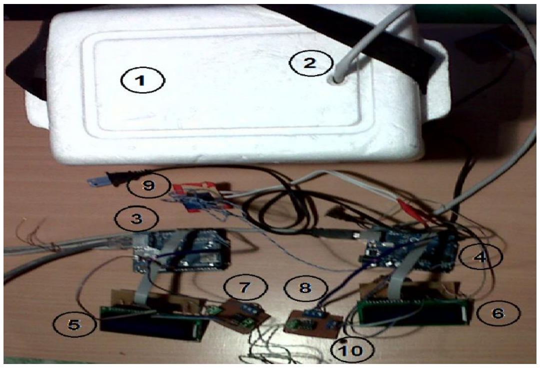

Figure 20. Top view of the hardware and the plant.

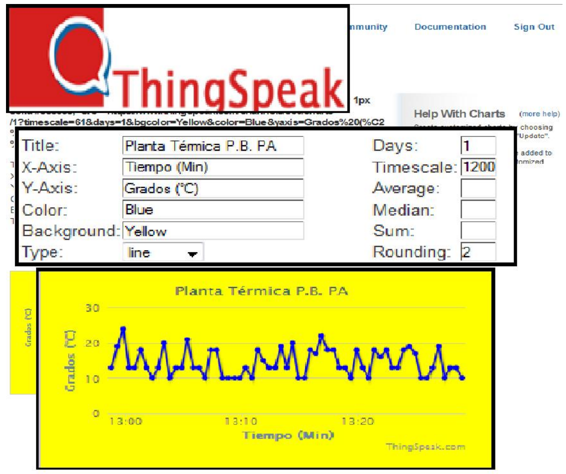

Figure 21. Viewing ThingSpeak account as an administrator.

Figure 22. Viewing ThingSpeak account as a guest.

Figure 23 shows the flow chart where the operation of the slave code is generally embodied as following the structure required by the software is appreciated, this should start by defining the libraries to use (in this case related to the LCD screen), then the connection of pins to the matrix display are specified and then the variables to use are set. Now you enter the configuration zone in which the serial port features are defined, as well as the digital pins (input or output) and finally the pin setting for the LCD. In this instance enters the cyclical part of the code, capturing the CAD data, for display on the LCD and wait for the master's request for a certain time, where it finally responds or the loop restarts.

For the master (figure 24) there is a block diagram similar to the slave flow, although the use of the Ethernet library is included to use the module of the same name. In this part of the configuration is added adequacy of the system to the access point to the network as discussed in one of the previous tests.

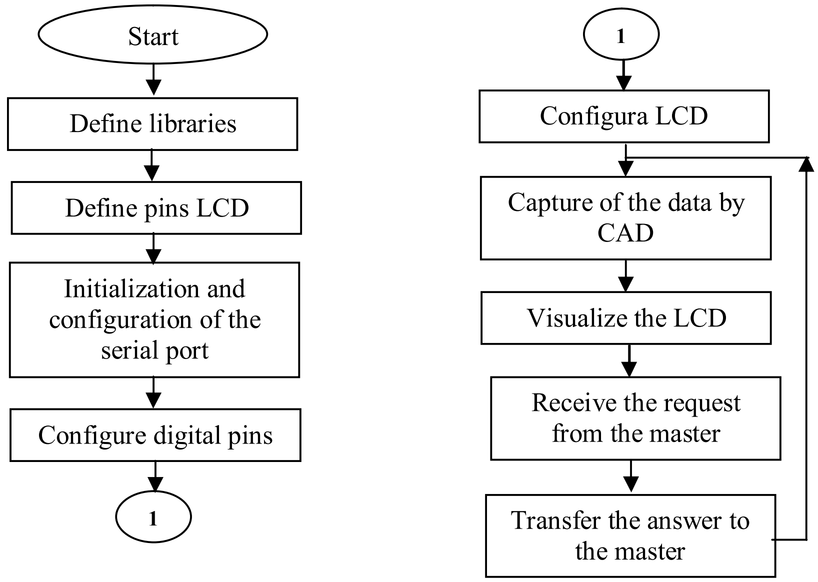

Figure 23. Flowchart of the slave code.

Finally the cyclic section begins by performing the request of data to the master, which, once it responds, the value is displayed locally via an LCD, and then makes a customer interaction as ThingSpeak web server, where the proper steps to make it possible to observe communication information via a web browser is stable, completing the cycle.

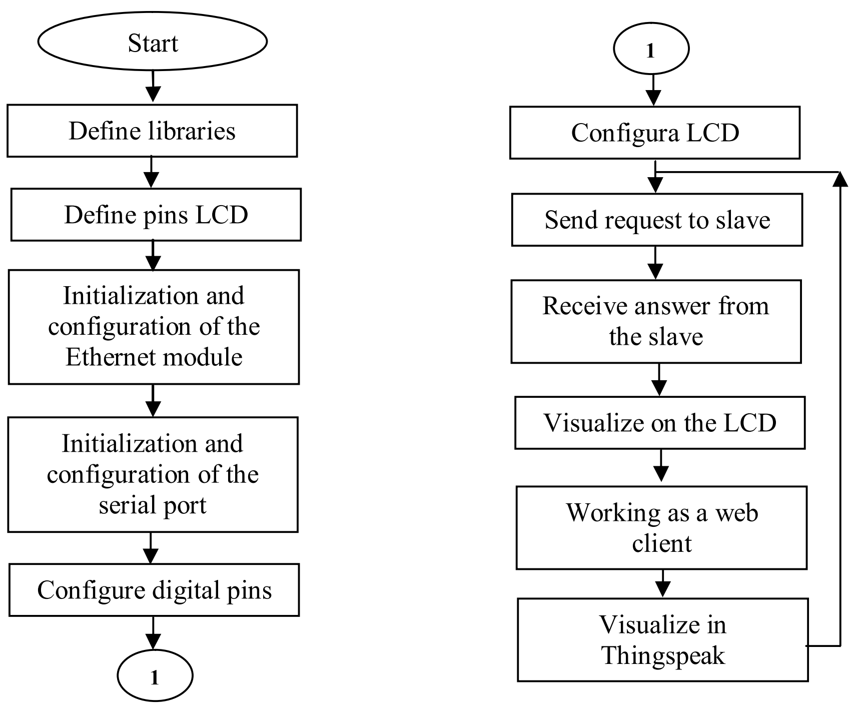

Figure 24. Flowchart of the master code.

4.2 Validation tests

In conducting the tests and the prototype, Arduino UNO was used which is an ATmega328 microcontroller [14] allowing SPI serial communication, where is necessary to establish a desired transfer rate (baud rate) [15], initially locating it at 9600 bps (recognized by the computer) to transmit a constant byte in a cyclic manner, obtaining figure 23, and then increase the rate to 31250 bps (set by the Profibus specification) repeating the process, arriving at the figure 24.

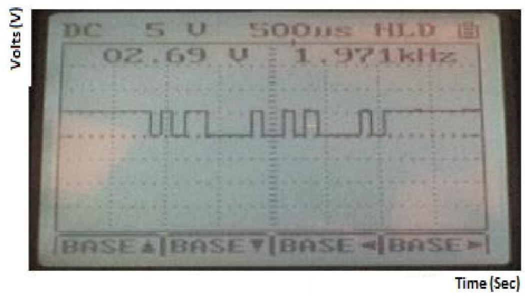

Figure 23. Capturing a byte transmitted by the 9600 bps Arduino UNO with 5 V / div vs. 200 µs/div.

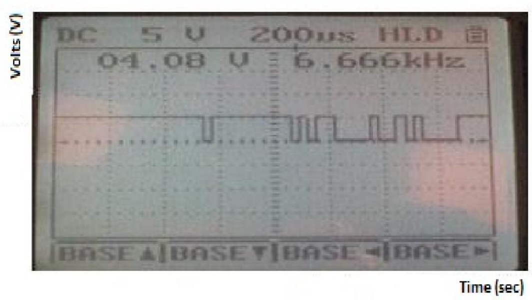

Figure 24. Capturing a transmitted byte by Arduino ONE to 31250 bps with 5 V / div vs. 500 µs/div.

By proving that Arduinos support the necessary speed for development, we proceeded to drive the display serial monitor [19] at 9600 bps (figure 25), to monitor the operation of the devices in some tests, especially when forming the master system (port com10) slave (port com9), previously named.

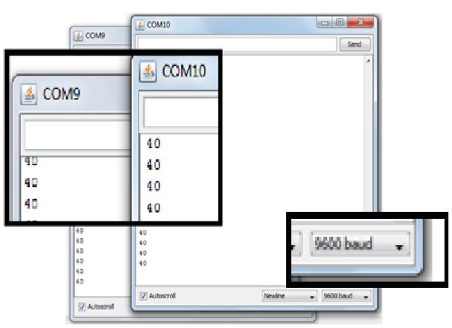

Figure 25. Display serial monitors for an Arduino UNO configuration as master-slave.

When the transfer rate of the serial port is changed to 31250 bps, the speed is not recognized by the computer, so it is essential to have another way to verify proper connection of the system, being here where the LCD displays become important, being able to see through them the proper transmission of data, an example is figure 26 which has a potentiometer whose value is displayed by a 16x2 matrix display through Arduino ONE.

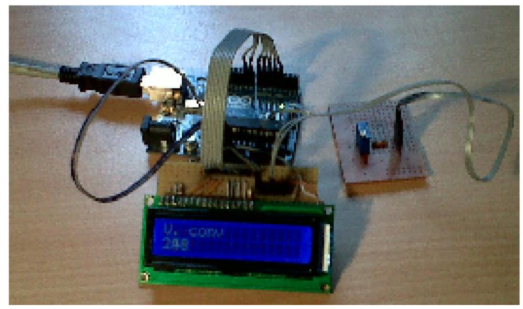

Figure 26. Visualization with LCD of data captured by the CAD.

As for the converter device between SPI and RS-485 we can observe using an oscilloscope the differential signals transmitted by the bus (figure 27) finding on top signal A and at the bottom B which meets the specification for transmission over twisted pair cable, by using the MAX485 (see figure 9).

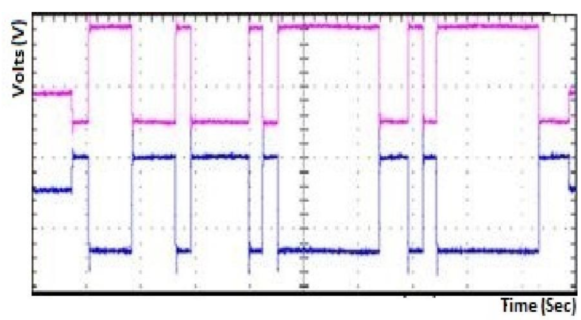

Figure 27. Differential signals on the bus with the MAX485 at 9600 bps, with 2 V / div transmitted vs. 500 µs/div.

To validate the sending of data packets to the master output, we used the software known as WIRESHARK, connecting the equipment with the analyzer, the output of the Arduino Ethernet Shield card and capturing from the frame sent by it to the ThingSpeak page for 305 seconds, figure 28, which clearly shows the sending of a request every 30 seconds as stipulated in the source code of the master, and which ideally is the minimum communication range offered Freely by the server. On the Y axis the number of packets sent per interval is plotted.

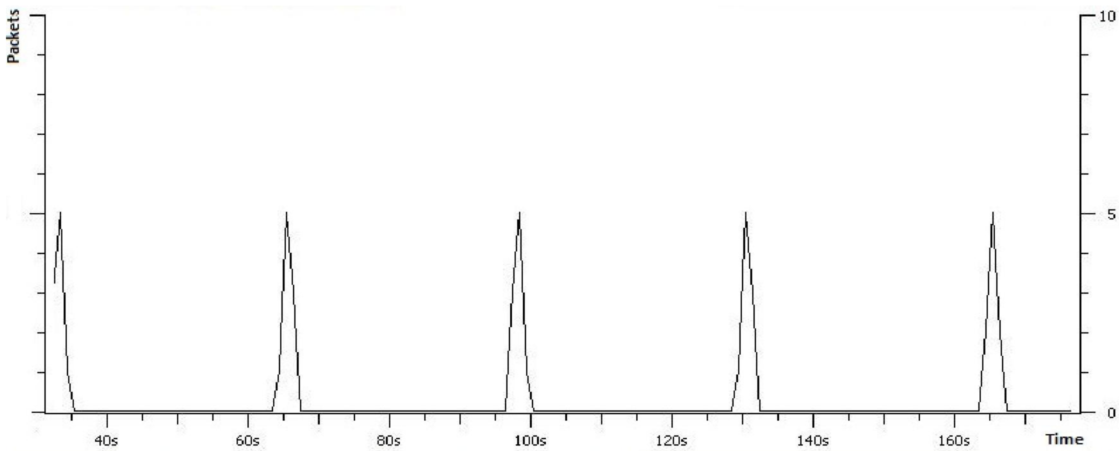

Figure 28. Delivery Graph of scheme against time (seg.) using Wire shark.

Table 1 shows some statistics of the capture graphed in figure 28 which shows that in 305 seconds 103 packets were captured and in each request 10 packets were sent so we can infer that 10 requests were sent, one every 30 seconds.

Table 1. Capture statistics on the Ethernet shield output.

wire shark capture statistics	
Capture filter	Ethernet
Captured packets	103
Time between the first and the last packet	305.651 sec
Average packet length	73.981 bytes
Bytes	7620
Average bytes per second	24.93
Packages by request	10

Sending data from the master and the ethernet shield board, is done using HTTP 1.1 POST type requests placed in its program, therefore the server can take and process the data allowed by its functionality. By averaging the data received by the web service and from keeping the plant in operation for 10 continuous hours, figure 29 is obtained, where an estimated 15.4 uploaded data per hour, i.e. it makes around every 3 minutes 54 seconds a new value is stored there, which may be reduced, depending on the service provider, to about 5 seconds by paying for an account with higher privileges.

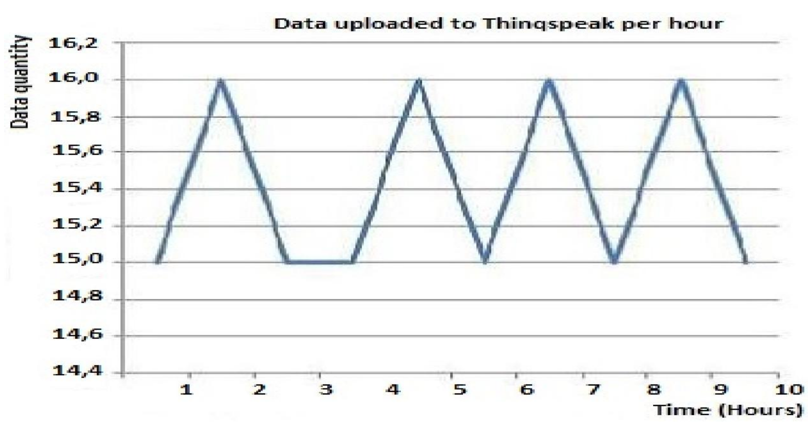

Figure 29. Graph historical record of the amount of data stored per hour.

5. Comparative study between the profibus network and PA and IPv4

To get to this point it was necessary to implement the connection between a sensor on the end of a Profibus PA profile type network with no intrinsic security to web display data after being stored on a server, where each value is plotted against time and produces a database, which can serve as a great way to make decisions about the censed variable. It should be noted that the capture was made by means of Arduino UNO plates in a Master Slave configuration interconnected by a RS-485 bus, obtaining an assembly used as the basis for insight into the characteristics of each element present, as well as their work together and then make a comparison as it is presented in Table 2.

Table 2. Comparison Profibus PA and IPV4

Basic Info	PA Type Profibus	IP
Development of technology	SIEMENS	Research Projects Agency of the Pentagon ARPANET
Year of introduction	1994 -1995	1969
Standard	EN50170 / DIN1945 Part 3 and 4 / IEC 1158-2	IPV4, TCP / IP
Characteristics	ASICs from Siemens and Profichip products from more than 300 vendors	Lots of chips and sellers
Physical Characteristics		
Network Topology	Line, star, ring or tree	Star, Daisy Chain
Hardware	Twisted pair or fiber	Coaxial, par braid, fiber, wireless
Max. number of nodes, devices	127 nodes (124 slaves - 4 segments, 3 repeaters) +3 masters	According to the network topology, but theoretically up to 4,294,967,296 that is the number of addresses
Max. distance	1900 m between segments At 31.25 Mbps	On the scheme there is no limit
Transport mechanism		
Communication method	master / slave, peer to peer	Peer to peer
Properties Transport	DP: 9.6, 19.2, 93.75, 187.5, 500 Kbps, 1.5,3, 6,12 Mbps PA:31.25 Kbps	Conditioned by the available bandwidth
Transfer data size	0-244 bytes	Maximum 65536 bytes, But it varies depending on the type of network
Priority method	Token passing	CSMA / CD and other according to the network
Error check	CRC	CRC
Delivery time frame (In this case)	One frame every 0.5 seconds (programmed in Arduino)	1 frame every 3 seconds (Scheduled HTTP request on the Ethernet card)
Diagnosis	Station, module and channel diagnostics	

It can be said that the development of these technologies has been an initiative of large companies and / or countries, focused on the investigation of the electronics and communications, who conceived different standards for different applications. IP began as a research project of the United States Department of Defense to 1969, while Profibus was initiated by Siemens to unify the field buses that were booming the last decade of the twentieth century. The PA profile was born shortly after DP and FMS, with the main aim of reaching industrial sensors and actuators, which are found in complex and hostile environments.

The two networks are fully normalized within their respective standards that are applied globally; therefore they have a variety of devices. On the fieldbus side there are some like the Asic [21] (chips Siemens) and Profichip [22], which facilitate the implementation of slave nodes with transducers or actuators driving a limited processing capacity, while IP has an extensive standardization of protocols, of which the most representative is known as TCP / IP.

The network topology in the PA model can be linear, star, ring or tree, physical connectivity through fiber optic or twisted pair plus a maximum of 127 nodes divided into 4 segments, which should be a repeater and a master (at least) for each segment as well due to handling is limited to current dependent nodes 32 from the same source.

On the side of the Internet protocol almost any network topology is used as a star or daisy chain [30] (which consists of a sequence of links). Its physical link can be done with coaxial cable, twisted pair and fiber optic, handling up to 4,294,967,296 nodes, which in theory is the maximum number of addresses available in IPv4.

It is noteworthy that the IP address can be of two types: static and dynamic, which depends on the allocation type of the address to the device, either fixed or a new assigned to it every time it connects to the network, while on the Profibus network there is a fixed system for each device on the network that is programmed and connects all devices to make an immediate recognition.

As to the possible mechanisms of transport the PA communication method is master-slave, although in the case of having more than one master peer to peer is used; (which is also a model widely used in IP communications) at a defined transmission speed of 31.25 kbps.

The two networks have a system for error correction (CRC) to determine if the frame is correct or if it should be resent due to data corruption. Additionally, the Profibus PA can have a specific configuration to make a diagnosis of the channel, the station and the module from the master to initialize the network.

The size of the Profibus data transfer is from 0 to 244 bytes and messages with token passing among masters are prioritized, while for the Internet the length can be up to 65536 bytes.

Despite the above, the most important point reached by the comparison is relative to the speed of information transfer via the Internet, since this is subject to the width provided by the network where it connects, and even on the Internet protocol version used since in the fourth a faulty packet could be discard when transiting between its origin and destination, while using the sixth the packet arrives at the target and the upper layers (OSI model) is entrusted with verifying if it is correct or not, so a bit similar to the operation of the fieldbus, which involves the physical level, the link and the application, while on his side the TCP / IP involves the link, network, transport and application layers.

In terms of speed the PA specification has a definite speed defined by the protocol of 31.25 kbps (which is a result of taking a crystal oscillator 1 MHz, the frequency dividing by 32), which is taken as reference the ASI [16] technology, which was widely used at the time of the consolidation of the standard, also one of the main goals of a Profibus is to have a low level bus for field instruments, where the processing power is not very high and requires low transmission speeds, for which the transfer rate was a good solution.

For the Internet protocol the speed depends on the bandwidth being used, so it is important to note that the transfer rate of the network varies depending on factors such as time of day or the number of active users. Additionally we should consider when making a request for a Web page to a server, the request and its response travel over the network through various links until it finds the less crowded route, spending a few milliseconds, so that the result is altered according traffic that existed at that time.

6. Conclusions

We have designed and developed a Profibus PA system that can handle transmission speeds of up to 31.25 kbps (as suggested by the standard).

As for the modular system designed for this Arduino fieldbus, it allows devices configured with the developed scheme to have a simple adaptation, fulfilling one of the main objectives of the generation of the standard more than two decades ago. Unfortunately this family of development boards does not bring integrated information transmission according to the RS-485 standard (the standard Profibus physical environment), but this issue is solvable with relative simplicity by employing an integrated circuit commonly known as MAX485 who is responsible for making conversion to or from SPI, allowing to handle various schemes fieldbuses.

It should be noted that this system can be implemented in various applications as long as the response time monitoring is not critical, since the time display on web services are tested in the order of minutes, which can be a serious limitation when making decisions on plants or systems that require rapid changes.

To finish, it only remains to say that despite being an academic application, the doors are open to future innovations that could use this work as one of its bases, and that future applications will be very numerous, because every day there are new needs dependent of fieldbuses to be properly addressed.

Corresponding Author:
PhD (c). Danilo Alfonso López Sarmiento
Department of Engineering
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia (South America)
E-mail: ingeniero24@hotmail.com

Authors:
Cristian Caballero
Electronic Enginner, graduate at Universidad at Universidad Distrital Francisco José de Caldas. Bogotá (Colombia – South America).

Rodolfo Romero
Electronic Enginner, graduate at Universidad at Universidad Distrital Francisco José de Caldas. Bogotá (Colombia – South America).

Danilo Alfonso López Sarmiento
Electronic Engineer, Master in Telematics, Full Time Professor at Universidad Distrital Francisco José de Caldas, PhD Student in Engineering at Universidad Distrital Francisco José de Caldas. Bogotá (Colombia – South America).

References

1. Profibus Coherent overview and application oriented, http://www.profibus.com, <Accessed: 2/02/2011
2. Profibus (PA / DP / FMS), Polytechnic University of Cartagena, Department of Electronics Technology, Industrial Communications Course. 2010.
3. López J. Profibus, Universidad de Vigo. Department of Systems Engineering and Automation. 2010.
4. Arduino Company, http://www.arduino.cc, <Accessed: 2/06/2011.
5. Arduino Company in Spanish, http://www.arduino.cc/es/, <Accessed: 2/0/2011.
6. Li Q, Jiang J, Rankin D. Evaluation of delays induced by profibus PA networks, Department of Electronic. & Computation Engineering, University of Western Ontario, London, Canada. 2009.
7. Pereira N. Research Group of the Polytechnic Institute of Porto, Portugal; Filipe Pacheco Paulo, Industrial Systems Institute, University of Patras, Greece; Luis Miguel Pinho, Institute for Automation and Communication, Magdeburg, Germany. 2002.

8. Zheng-shi C, Jun-zheng S, Jian-fang S. Real-time property analysis of single master DP/PA hybrid network control system. Coll. of Computation & Electronic, Inf., Maoming University, Maoming, China. 2010.
9. James C. Virtual breadboard, http://www.virtualbreadboard.net/, <Accessed: 12/07/2011.
10. Jan A. Programming and circuits for SPI and RS-485, Serial port complete, Lakeview Research, USA. 2003.
11. [Online], www.neoteo.com/rs485-domotica-al-alcance-de-tu-mano-15810, <Accessed: 3/05/2011.
12. Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers. Maxim Integrated Products, USA. 2003.
13. IoBridge. ThingSpeak Company, https://www.thingspeak.com/, <Accessed: 13/08/2011.
14. ATmega328P, ATmel Corporation, http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf, <Accessed: 24/03/2011.
15. Arduino Company in Spanish. Communication via serial port, http://arduino.cc/es_old/Ejercicios/PuertoSerie, <Accessed: 10/08/2011.
16. Guerrero V. Bus AS-I: Configuration and programming of an AS-i network, http://instrumentacionycontrol.net/Descargas/Descargas/SIEMENS/Comunicaciones/IyCnet_Siemens_01_ASi_S7_300.pdf, <Accessed: 21/01/2011.
17. Quiroga A, Ignacio J. Class 79th automation and control Course, http://www.youtube.com/watch?v=DsPGXfBXfZY, <Accessed: 18/05/2011.
18. LM35 Precision Centigrade Temperature Sensors, National Semiconductor Corporation, 2000.
19. Arduino Company. Arduino Development Environment, http://www.arduino.cc/en/Guide/Environment, <Accessed: 19/03/2011.
20. Datashet Led, http://es.scribd.com/doc/18946526/Datasheet-Lcd-16x2, <Accessed: 20/03/2011.
21. Procentec. SPC4 Protocol ASIC, http://www.procentec.com/products/spc4-profibus-asic.htm, <Accessed: 27/07/2011.
22. profichip Gmb. Profibus-DP Slave Controller with serial Interface, http://www.profichip.com/products/overviewasics/, <Accessed: 2/04/2011.
23. IEC 1158-2, Transmission Technology, http://www.control.aau.dk/~henrik/undervisning/netpro/033iec1158.html, <Accessed: 25/03/2011.
24. ATMEL, Atmel AVR 8 - and 32-bit, http://www.atmel.com/dyn/products/devices.asp?family_id=607, <Accessed: 21/03/2011.
25. Arduino, Ethernet Shield, http://arduino.cc/es/Main/ArduinoEthernetShield, <Accessed: 3/03/2011.
26. Chain D. [Online], http://es.wikipedia.org/wiki/Daisy_chain, <Accessed 20/04/2011.

8/6/2014