

Implementation of a passive device for a Profibus-DP network, based on microcontroller¹

Implementación de un dispositivo pasivo para una red Profibus-DP, basado en microcontrolador¹

Received: 02-02-2015 Accepted: 04-05-2015

Asfur Barandica López²
Carlos Julio Ramírez Borrero³
Melani Gisella Viveros Moreno⁴

Abstract

This paper presents the development of a passive station with communication capability through the Profibus-DP protocol, for general purpose inputs and outputs. The implementation is based on an STM32 microcontroller, and it is fully compatible with the specifications of the physical and data link layers defined for the protocol. A detailed hardware and software architecture of the passive station, as well as the tests performed to verify its operation, are also described in this paper. These tests involve equipment from known industrial manufacturers to demonstrate the interoperability capability of the device.

Keywords: Profibus-DP slave; device level networks; automation networks; industrial communications.

Introduction

Profibus is, without doubt, the communication protocol for industrial machinery with the most number of nodes installed in the world (Spiegel, 2008; Control Engineering, 2007; Lydon, 2008; Carlsson, 2015). In the Andean region, there are many of the automatized facilities with this network and the one that is previewed that will be installed in a near future. A study performed in 2008 found out that Profibus competes with DeviceNet for the buses device market in the industry of the Valle del Cauca, but their offerings are not well-known and exploited (Barandica, 2008), what generates the need to capacitate personal in design, operation and maintenance of industrial communication networks.

It is clear that the deeper the knowledge around the protocol, the higher will be the return obtained from the exploitation from the industrial network and less

¹ Developed by the Perception and Intelligent Systems, Universidad del Valle (Cali-Colombia), that has been developing with Profibus protocol since 2009.

² Colombiano. MSc. Engineering with emphasis in electronics, professor in the Universidad del Valle Cali, Colombia. Perception and intelligent systems research group. e-mail: asfur.barandica@correounivalle.edu.co.

³ Colombiano. Electronic engineer - Postobón S.A. e-mail: carlosjulior@gmail.com.

⁴ Colombiana. Electronic engineer - ZTE Colombia S.A.S. e-mail: melani.viveros@gmail.com.

will be the stop times due to fails, expansion operations and maintenance. For this, the inside of the Perception and Intelligent Systems research groups, projects were promoted to form engineers and develop teams that contribute the industrial competitiveness through the adequate use of these technologies. The design capabilities of the gear in the Profibus protocol implies the dominance of deep knowledge and allow to offer to non-conventional solutions, with good cost/benefit, to the users of Profibus network.

The majority of the related posts with Profibus in Colombia and its neighbor countries limit themselves in describing the interconnection in facilities of foreign manufacturers using the protocol (Córdoba y Sandoval, 2007; Otáñez, 2009; Echeverri y Grisales, 2013). Own implementations have been reported, as it is in the case of Calderón (2004), where apart from being a Profibus slave, he developed a software that makes the times of a network master device; however, there is no trustworthy process in validation, due to that system components are developed by the author; in Cardona y Castañeda (2010), a Profibus slave was built was conveniently validated, although it was supported in two Specific Application Integrated Circuits(SAIC) (UFC100-F1, AMIS-492x0) that reduce to the minimum the work upon the Profibus communication stacks. Recent publishing of developed countries, denote that the addressed topic has relevance and commercial potential (Xu et al., 2011; Yanjun et al., 2007).

A passive dispositive (slave) is presented to the Profibus-DP network, which, different from the already mentioned, was assembled without specific integrated circuits for the protocol, because of this, it was necessary to develop the whole communication stack in base of a digital processing platform in the STM32 microcontroller. Also the validation process was executed to give trustiness to the results.

The article was organized in the next way: The methodology briefly describe the features of the protocol and the minimum specifications of each device with Profibus capacity, followed up with the slave design and its definition, the hardware selection and the software architecture; the tests conducted to the prototype and the outcomes, the conclusions and the work perspective.

Methodology

Initially the Profibus-DP study was addressed and the slave specifications were defined, in order to establish the minimum characteristics of the digital processing platform and the slave hardware in general. A slave implementation was carried out and its connection with a master simulator to begin the basic communication tests. A GSD file was made using the program *GSD Editor5* developed by Profibus *Nutzerorganisation* e.V. For it a GSD file of the Micromaster was taken as an example. The GS files provide a clear and comprehensible description of the devices, specifying the communicative features of a defined format. Finally a slave was added to a Profibus network to conduct ultimate tests of communication.

Up next, a brief description of the protocol is made, before showing the design.

Profibus-DP (Decentralized Peripherals)

Profibus-DP is a communication protocol between two industrial facilities that allow high transfer data rates in a trustworthy way. A Profibus-DP network differentiates in two kinds of gear: the masters or active stations and the slaves or passive stations. The masters send output data to the slave and obtain an answer that includes its input data.

There are two kinds of active stations in DP protocol:

- **DP Maestro class 1:** Refers generally to the gears that perform the central control through the Exchange of information with the distributed teams or slaves.
- **DP Maestro class 2:** Are devices used in programming, configuring or diagnostic. Are parameterized in the commissioning with the goal to specify a configuration of the DP system. This configuration provides with information such as the number of DP devices, the bus station directions and the output and input devices.

The passive devices or slaves are peripheries that generally perform I/O functions, compiling the status of the sensors and giving values to the processors performer.

The communication stack was organized according the OSI reference model for open inter-connection systems. Profibus defines the last layer as physical and of data link. The physic layer in copper twisted pair makes used of the standardized Rs-485, which allows transfer speeds up to 10Mbps and up to 32 stations in a segment; the

transmission is asynchronous with 8 bits capacity of data and parity. Alternatively, the protocol also accepts the use of optical fiber.

The link data layer is in charge of making the link reliable and provides the ways to activate, keep, maintain and deactivate the link. It is also in charge of the detection and error controls. The Profibus frame formats are summarized in the Table 1. A passive station should only

be in capacity to perform three of these formats (2, 3 and 5), due that the other two are used for communication in active station. The transfer data transaction include and emission of a frame request from the master to the slave in which the output data are sent, and answer frame from the slave to the master, in which the input data are given. A full description of the protocol can be found in (Acromag, 2002).

Table 1. Profibus type of frames

Type of frame	Fields used in the frame												
	SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS		
1. Search of new masters	10H				x	x	x					x	16H
2. Data transfer	68H	x	x	68H	x	x	x	x	x	X	x		16H
3. Data transfer	A2H				x	x	x				8x	x	16H
4. Transfer from witness to master	DCH				x	x							16H
5. Short recognition	E5H												

SD: frame start delimiter
 LE: byte de longitud (DA+SA+FC+DSAP+SSAP+DU)
 LEr: repeated length byte (same content from LE)
 DA: destination node address
 SA: source node address
 FC: FC: Frame control byte or function code

DSAP: destiny service access point
 SSAP: source service Access point
 DU: Data unity. 8 bytes with SD=A2H and between 1 and 244 s bytes with SD=68H
 FCS: frame check sequence
 ED: end delimiter

The blank spaces are not used. Meaning that, the frame type 1 frame only has 6 bytes, and the type 5 frame, 1 byte. X means that this field contains a number between the bytes variables (1 to 244). 8x this that this field contains 8 bytes.

Source: The authors

The first version of Profibus was denominated DP-V0. In it the original DP specifying were defined which allow the cyclic data transfer between master and slave, la parameterization and diagnostic data reading, gives a fast failure localization. The DP-V1, version that emerged as an extension of DP-V0, adds the acyclic communication of master/slave data, what constitutes a good complement for the parameter modification, because it allows a parallel transmission process way (Cyclically transmitted) and of the configuration parameters, acyclically transmitted and with less priority; also adds the alarm recognition. The DP-V2 version adds to the prior versions the slave/slave exchange of information, allowing reaching high synchronization speeds upon the running and clocking synchronization.

Slave specifying and hardware requirement

A slave with general porpoise capable of manipulating four inputs and four outputs was specified. With a *watchdog* to guarantee the strength of the design against noise and other environmental variables, and equipped with a serial asynchronous interface destined to the communication in the Profibus network. It was the first develop upon this protocol, it was opt for a relatively slow objective speed of 57.6 kbps.

For a network that works to speeds up to 57.6 kbps, it was determined that an 8 bits microcontroller with a 1MIPS performance could attend without delay the master's requests. In the same way, it was considered appropriate a size of 32 Kbytes of storage memory and at

least 2 kbytes of internal RAM memory for the handling of temporal variables generated during the programs execution. It is essential the presence of no-volatile memory of at least 256 bytes for the storage of node configuration parameters.

In view that no microcontroller include the drive/ receiver for the RS-485 interface, it is essential its external addition; besides, it is always recommended to isolate the electrical signals of the RS-485 bus, with the purpose to protect the node against possible network failures and vice versa.

Fulfilling the physical layers of the Profibus, a DB9 female connector must be used in the slave, to place the RS-485 bus signals in the middle of the transmission.

Hardware architecture of the slave

As a support of the passive unity, the *ET-STM32 Stamp* development board was selected which counts with a 32 bits Cortex™-M3 processor ARM, a 72 MHz of maximum frequency CPU, which offers an exceptional performance in the gearing and (90 MIPS) and an advance system of uninterrupted answers, high speed onboard memory, flash memory for 512 Kbytes, up to 64 Kbytes of STAM and an interesting joint of peripheral units (ADC of 12 bits, 16 bits timers, PWM sub-systems, I2C, SPI, I2S, SDIO, USB, CAN and USART communication interfaces).

Since the card doesn't dispose EEPROM, an EEPROM emulation was made using flash, as it is described in ST Microelectronics, 2009.

In the Figure 1 a diagram is shown of the Profibus slave in blocks. The interface in the Profibus network is composed of two stages: electric isolation and coupling with the environment. The isolation was made by MCT6 optoacoplators for the reception signals (Rx), transmission (Tx) and data transfer habilitation (DE). These devices limit the bit ratio to a maximum of 150 kbps, even though there are other references (6N137, VO2601) which with you can reach 10 Mbps. The coupling stage with the environment convert the signals referred to the earth of the microcontroller to differential signals and adequate the tensions levels and impedances, to the ones defined by the RS-485 specification; for this purpose the el MAX485 driver/receiver was selected. As part of this stage DB9 female connector was also added for the connection to the environment.

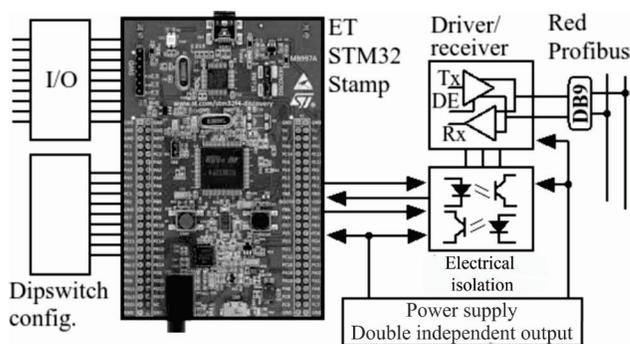


Figure 1. Block diagram of the slave's hardware
Source: The authors

A joint of dipswitches type of switches were placed to manually define the slave's direction and a LED to confirm to the user that the parameterization and configuration processes were being performed successfully; the LED blinks in different speeds depending of the state it finds itself in, until becoming completely on to reach the state of data exchange.

Software architecture

The implementation of the data layer was done through the programming of the microcontroller of the STM32F103 development board. The program was written in C language and compiles the Keil uVision4 development environment, which generates a file with .hex extension. The microcontroller possesses a default bootlader that allows the download of the .hex to the platform through the computer's RS-232 through a software called Flash Loader Demonstrator.

In its most general view, the developed program implements the machine in states that regime the behavior of the Profibus' slave, which is represented in Figure 2 (a).

At the time of turning on, after a reset or after activating the *watchdog* timer, the slave will be in the state *Power_ON/Reset*, during which it makes the initialization for hardware resources and the memory parameters that define the node's behavior. If the address of the node is valid (any value between 0 and 125), the machine will automatically pass to the Parameterization state; if the address is not valid, it will wait for an assignation of address message. In the Parameterization state, the slave will wait for the master telegram, which identifies the station that has been assigned as a master and in operation mode. Once the parameters are received,

the slave goes to a Configuration_I/O state, in which the slave waits for a telegram that specifies an input and output number of bytes that shall be exchanged in each posterior transaction. In the Parameterization and Configuration_I/O states the different microcontroller peripherals will be initialized to answer the demands in what respects to the slave's functionality. Done the configuration, the slave stays in a Data_exchange state, where I/O data can be Exchange or modify the parameters and configuration of the slave.

For some reason the slave takes the initiative to send a message; it should always wait for a request message from a master. The program then remains catching and analyzing the frames that go through the web and executing appropriate actions, according to the current state. The attention to the master's telegrams is done following the flux diagram of Figure 2 (b).

The block Frame_reception captures the received characters by the Synchronic and Asynchronous Communication Interface (SACI) following the same state machine of the Figure 2 (c), that is mainly controlled by the Reception by Interruption (RI) and the received data in the Reception Buffer of the SACI (SBUF). A counter is used to capture the number of bytes expected according to the kind of frame. Errors generate if the end of frame delimiter is not in a expected position or if the repeated length byte does not matches with the announced length. Finished the repetition process of a length, the Valid_frame? Block makes the sequential validation verification (FCS in Chart 1) and determines

if the frame goes directed to the slave, in which case the Frame_Analysis block makes an analysis of the remaining fields: SA, FC, DSAP, SSAP y DU.

After analyzing the received frame, the Answer_Generation block is in charge of building an answering frame to send it to the master. For this, the different frame fields of answer are stored in an Array variable to then be transmitted to the SACI. The Direct Access Memory (DAM) completely controls the direct transmission of data towards the SACI data register, until the DAM counter reaches zero; thus liberating the CPU the management interruption of transmission each time a byte is sent through the frame; in fact, before giving the control to the DAM transmission, the SACI interruptions must be disabled (Hitex, 2009).

Finished the transmission of the frame, the transmission driver must be disabled to allow the use of the communication channel by other network stations and SACI interruptions of general receptions must be enabled.

Tests and results

The slave tests were focused on three aspects: the capability in operating a bus without generating interferences upon other devices, conformity with the specification of the protocol and determination of the operation's max speed.

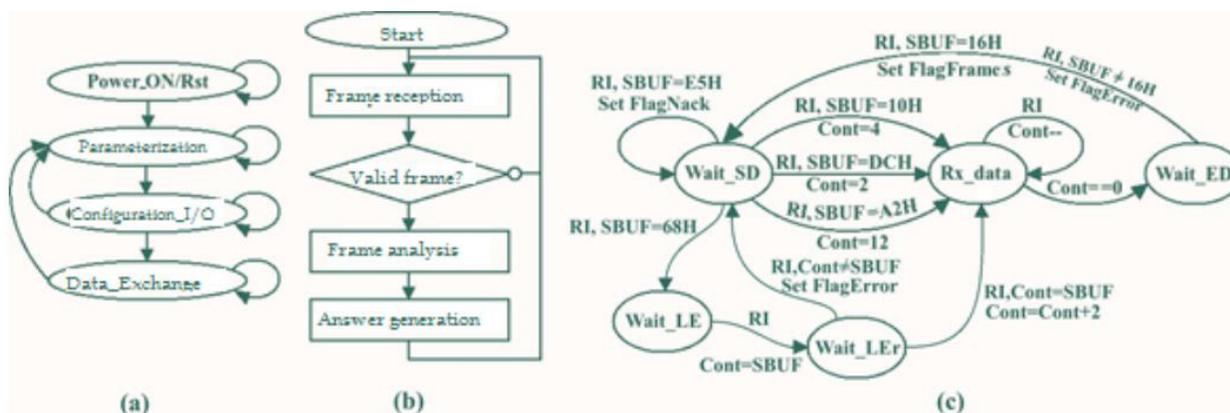


Figure 2. State diagrams and program blocks. Source: The authors

In their early development stages, the slave was tested using Profibus frames from a laptop disposed of USB/RS-485 (USOPTL4- LS from B and B Electronics) converter, that simulated the actions of manipulating the serial ports (232Analyzer) that allow to manually compose the frames and be sent and receive the answers. In later phases, a device conformed by industrial gear and

software of known manufacturers; was configured the Profibus network at 9600 bps with two Siemens Micromaster 440 speed variators and a CP-5611 card, installed in a PCI slot of the PC that sometimes take the roll of network master. The full scheme in shown in Figure 3.

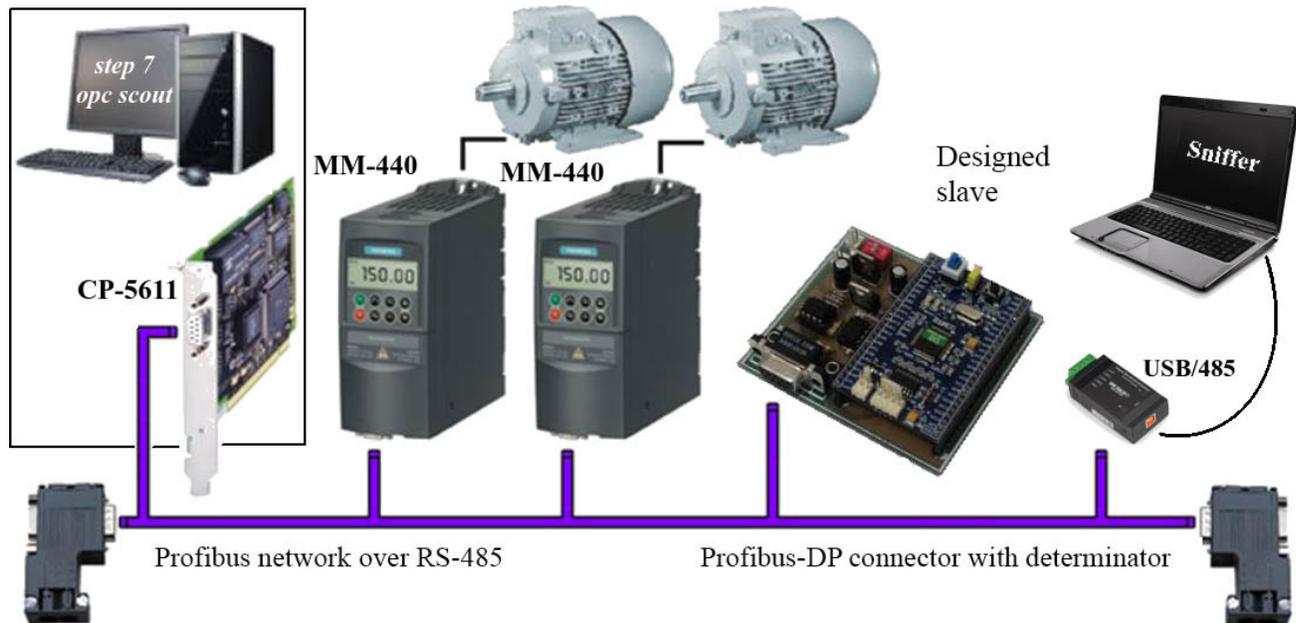


Figure 3. Network schemes for the slave's test.
Source: The authors

With the appliance of Step 7 the network configuration was made (address assignation, speed assignation, communication speed, and OPC server definition). Before adding to the network the slave base don STM32, a GSD file was loaded with the purpose that Step7 would recognize it and easily install it. Once with the configured network, basic connectivity tests were held with the variators, taking in account that the Micromaster 440 use the Profidrive profile. The program 232Analyzer was used in a sniffer way in the network, to capture the traffic and prove the frame sequences for the initialization of the devices in operation of the master. It was verified that the operation of the speed variators was not affected by the presence of the designed slave; hence from the computer the data was exchanged with the variator, through the Siemens' OPC Scout software.

Likewise, the conformity with the specification of the protocol is taken for granted as satisfied, taking in account that the slave surpassed the two initial states until remaining in data Exchange mode. Though the OPC Scout server allowed the visualization of all the

frames, it only delivered the information of the DU field that include; thereupon, sniffer was of great utility, because it captures the complete frames with any delimiters, allowing to verify that no recognizings were presented negative or there were failed attempts of communication coming from the master.

In the operation or data exchange operation, it was possible to read the state of the digital input joint and alter the digital output joint. In the same way, operations were performed with the change of address. These exchanges were performed while the other network gear was actively exchanging data which assures the interoperability of the developed slave. The Figure 4 shows the complete implemented frame in Step7 and the successful communication from OPC Scout.

Finally, the network speed was increased to 19.2 without any changes in its function.

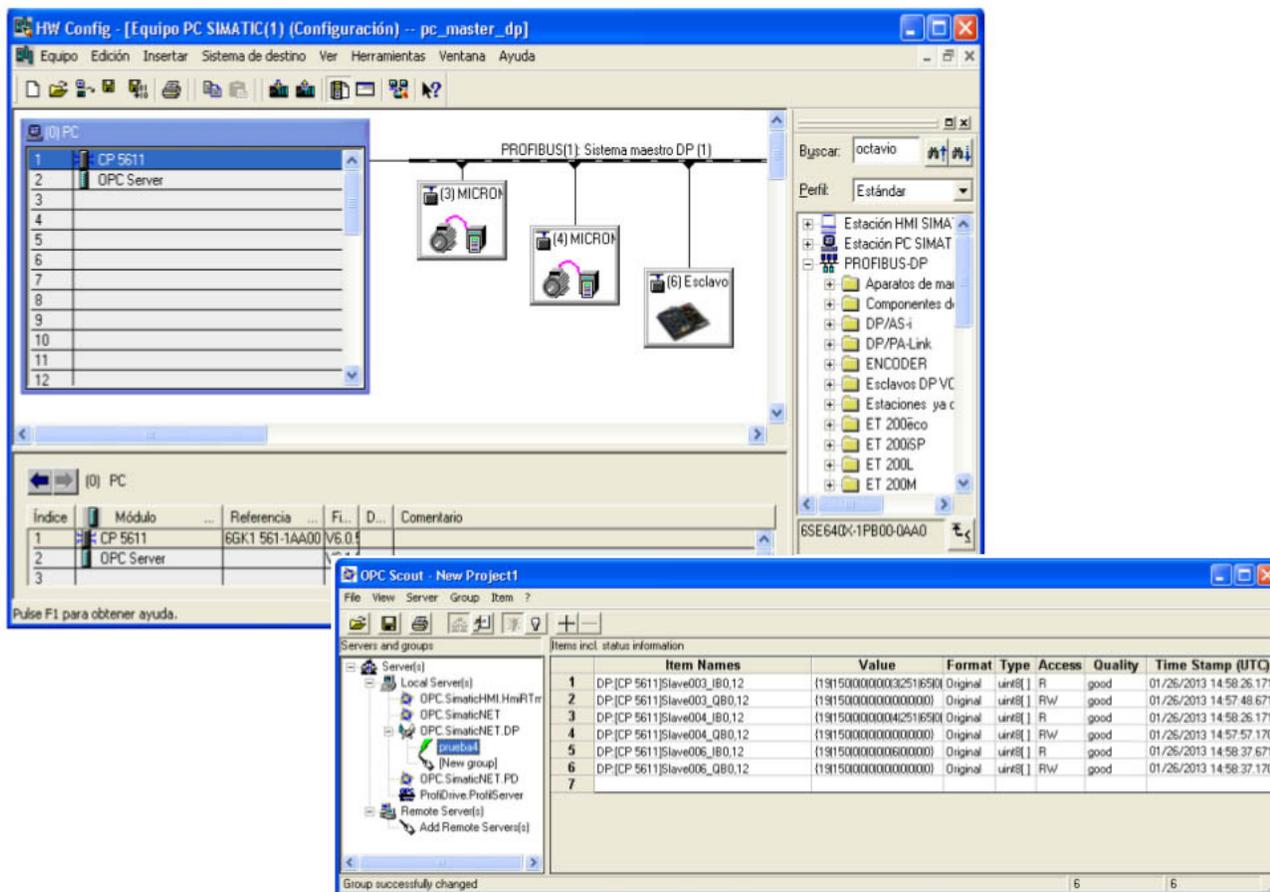


Figura 4. Screenshot showing the network and successful exchanged messages.
Source: The authors

Conclusions

An implementation was made in a passive station for a Profibus-DP network using a STM32 microcontrolled platform. That implies that the communication stack had to be completely developed, different to the implementations that SACI circuits use. Also it was necessary the development of an electronic data sheet type file (GSD file), as an essential condition for the master to make the parameterization and configure the slave. The slave was inserted in a network composed of industrial gear, demonstrating that its capacity to exchange data with the master without interfering with other communication devices in the network. In the network, two slaves were present, a master and a diagnose gear (snnifer). The tests were done successively to 9600 and 19200 bps, without problems appearing because the speed changes.

The slave was allowed to change the address, parameterization, I/O configuration and data Exchange using the defined frames and functions by the specifying of the protocol.

More than the design of a product, this Project shows our capacity to offer the national industries solutions that can make better the productive processes with inferior costs to those imported options offered by the market. With it, it has been demonstrated that the design capacity of Colombian engineering and files have generated and useful tools to develop new implementations and capacitate personal in the area of industrial communications, particularly the Profibus-DP protocol. It is pretended that in a posterior version, the slave include the Freeze and Sync capacities, can exchange analogical data variables and operate at 500 kbps.

References

Acromag Incorporated (2002). *Introduction to Profibus DP. BusWorks™ 900PB Series ProfiBus/RS485 Network I/O Modules Technical Reference, USA.*

- Recuperado de: http://www.acromag.com/sites/default/files/Acromag_Intro_ProfibusDP_698A.pdf.
- Barandica A. (2008). *Una propuesta para la apropiación de la tecnología HART y su transferencia hacia el sector industrial*, (tesis de maestría) Universidad del Valle, Cali, Colombia.
- Calderón Porras E. (2004). Implementación del protocolo Profibus de redes de comunicación industriales para la automatización de sistemas de manufactura, *Revista Colombiana de Tecnologías de Avanzada*, 2(4), 17-24.
- Cardona C.M. y Castañeda A.M. (2010). *Convertidor 4-20mA a Profibus PA*, (tesis de <http://www.automationworld.com/networking-amp-connectivity/pto-step-profinet-promotion>).
- Carlsson T. (2015). *HMS' view of the different networks on the market in 2015*. January 22. Recuperado de: <http://www.anybus.com/readnews.asp?NID=177>.
- Castañeda Verano, A.M. y Cardona Barón, C.M. *Convertidor 4-mA a Profibus PA*, (Tesis de pregrado) Pontificia Universidad Javeriana, Bogotá, Colombia. Recuperado de: <http://repository.javeriana.edu.co/bitstream/10554/7002/1/tesis451.pdf>.
- Control Engineering (2007). *Digital network: With 20 million nodes, Profibus is now mainstream technology*, Control Engineering, Agosto 14. Recuperado de: [http://www.controleng.com/index.php?id=2805&tx_ttnews\[tt_cHash\]=95bae4cd19569a06ea7e406e4cc9aa36](http://www.controleng.com/index.php?id=2805&tx_ttnews[tt_cHash]=95bae4cd19569a06ea7e406e4cc9aa36).
- Córdoba J.E. y Sandoval A.M. (2007). *Implementación de un sistema supervisorio sobre una red industrial Profibus-Ethernet en un caso de estudio*, (tesis de pregrado) Universidad del Cauca, Popayán, Colombia.
- Echeverri S.M. y Grisales G.A. (2013). *Implementación de una red Profibus-DP en un sistema automatizado*, (tesis de pregrado) Universidad Tecnológica de Pereira, Pereira, Colombia. Recuperado de: <http://repositorio.utp.edu.co/dspace/bitstream/11059/4353/1/6298E18.pdf>.
- Hitex (UK) Ltd. (2009). *The insider's guide to the STM32 ARM based microcontrollers*. Coventry, Reino Unido. Recuperado de: <http://www.hitex.com/fileadmin/pdf/insiders-guides/stm32/isg-stm32-v18d-scr.pdf>.
- Lydon B. (2008). PROFIBUS Hits New Highs... over 25 million installed nodes!. *Rev. Automation*. Recuperado de: <http://www.automation.com/library/articles-white-papers/articles-by-bill-lydon/pto-2008-annual-meeting>.
- Otáñez J.P. (2009). *Modernización de dos sierras de corte para el proceso de elaboración de tablas en la planta industrial Aglomerados Cotopaxi S.A.*, (tesis de pregrado) Escuela Politécnica Nacional, Quito, Ecuador. Recuperado de: <http://bibdigital.epn.edu.ec/bitstream/15000/1437/1/CD-2135.pdf>.
- Spiegel R. (Septiembre 2008). PTO to step up Profinet promotion, *Rev. Automation World*. Recuperado de: <http://www.automationworld.com/networking-amp-connectivity/pto-step-profinet-promotion>.
- ST Microelectronics (2009). *Application note AN2594: EEPROM emulation in STM32F10x microcontrollers*. Recuperado de: http://www.st.com/web/en/resource/technical/document/application_note/CD00165693.pdf.
- Xu Hongyan, Wu Guichu, Chen Chong. (2011). *Based on STM32F103 Implement Profibus-DP Slave with High-speed Transmission*, International Conference on Uncertainty Reasoning and Knowledge Engineering, Bali, August 4-7, Vol. 1, 232-234. Recuperado de: <http://www.ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6007805>.
- Yanjun Fang, Jingyu Liu, Bo Xi. (2007). *Development of Multifunctional Controller Based on PROFIBUS-DP*, IEEE International Conference on Automation and Logistics, August 18 - 21, Jinan, China, 144-148. Recuperado de: <http://www.ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4338546>.