SENSOR DIAGNOSTIC HART OVERLAY 4-20mA

Master's thesis of:

Richard Ochoa Hidalgo

FACULTY OF SCIENCE AND TECHNOLOGY

2011

SUMMARY

The purpose of this thesis is to provide a better understanding of the HART Communication Protocol; its capabilities to enhance communication with smart field devices and the functionalities that a monitoring and asset management system can draw on through the integration of HART field device's data.

To this end, an integration approach to enable access to HART field devices is presented in this study. The approach provides a seamless manner to connect the devices to higher-level communication systems which supply the communication path for accessing the HART devices and data from remote locations.

In addition to the HART communication protocol, Profibus DP and Ethernet (TCP/IP) protocols were analysed in order to determine the method of integration and select the appropriate components that would meet nowadays industry standards.

In this report, an overview of the aforementioned protocols is given; followed by a description of the integration methods, and a depiction of the hardware and software components which constitute the framework of the experimental trials.

The experimental work comprised the use of level, pressure and temperature sensors with HART capabilities. These devices were connected to HART-enabled signal conditioners (I/O modules) for transmission of HART data into a Profibus DP communication gateway; which incorporates the data into Profibus telegrams. An Ethernet-Profibus gateway was then utilised to embed the telegrams in Ethernet (TCP/IP) messages and in that way enable access to the HART sensors data through a simple Ethernet network.

Furthermore, this study carries out three integration tests where the software applications PACTware, AMS Suite Intelligent Device Manager and TH OPC Server DP are examined.

These applications, based on different standard technologies to handle the HART field device’s data, make use of the presented integration approach and provide access to HART data to operators that wish to monitor and manage HART devices from remote locations.

ACKNOWLEDGMENTS

I would like to thank my parents Roman and Graciela, and my sisters Gaby, Ma. de los Angeles and Ita for all the love, understanding and support provided me during all my years of study away from home.

Special thanks to my Norwegian friends Sidsel Rogde and Olav Gramstad for the unconditional support given to me during my years of living in Norway.

Deep gratitude to Håkon Vidar Straume at National Oilwell Varco for the suggested thesis topic and for all the help received while writing this thesis.

To all my dear friends, who encouraged me through the years to fulfil my studies.

TABLE OF CONTENTS

1 ABBREVIATIONS

Abbreviation	Description
AMS	Asset Management System
Bps	Bits per second
CPU	Central Processor Unit
DCS	Distributed Control System
DP	
DRM	Delayed Response Mechanism
DTM	
EDD / DD	Electronic Device Description / Device Description
EDDL	Electronic Device Description Language
FDT	Field Device Tool
FSK	Frequency Shift Keying
HART	Highway Addressable Remote Transducer
HCF	Hart Communication Foundation
HHT	HART Handheld Terminal
HSA	Highest Profibus Address
ID	Identifier
IEC	International Electrotechnical Commission
IS	Intrinsic Safety
ISO	International Standards Organization
LSB	Least Significant Bit
mA	milliampere
MPI	Multi-Point Interface
ms	milliseconds
MS	Master Slave
MSB	Most Significant Bit
OPC	Object-Linking and Embedding [OLE] for Process Control
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PLC	Programmable Logic Controller
PV	Primary Variable
QV	Quaternary Variable
RBF	Receive Buffer Full
RTS	Request-To-Send
SCADA	Supervisory Control and Data Acquisition
SV	Secondary Variable
TCP / IP	Transmission Control Protocol / Internet Protocol
TV	Tertiary Variable
UART	Universal Asynchronous Receiver/Transmitter

2 INTRODUCTION

Nowadays requirements to achieve greater reliability, lower maintenance costs and reduced number of downtime periods in oil & gas plants and installations promotes continuous improvement efforts regarding integration of intelligent field devices with HART-enabled capabilities.

Such instrumentation utilises the industry standard 4-20 mA analog signal to convey a measurement value, in addition to the digital communication signal enabled by the HART Communication Protocol. This enhancement of the analog signal provides supplementary process measurements as well as device status, configuration options and diagnostics information.

National Oilwell Varco (NOV) provides drilling solutions that include field-proven equipment integrated into systems which aim to maximize safety and efficiency. One of these systems is the NOV's Smart Drilling Instrumentation System (Sdi), which among other components, comprises field instrumentation such as level-, pressure-, and temperature sensors with HART capabilities.

The use of these field devices gives rise to challenges related to access, monitoring and maintenance due to they are, in the first place, located in hazardous areas and then hooked onto Remote I/O modules in marshalling cabinets placed in safe areas. To connect to every and each of the Remote I/O modules, which act as an interface to the field sensors in the Sdi system, is a today's practice when it comes to collect diagnostic sensor data. In this context, failure events are for instance not detected before the measurement values are not longer available on the Sdi monitor system causing, in the worst case, postponements in drilling operations.

Hence, it is the NOV's aim to supply a state-of-the-art solution that would enable remote access of information contained in the HART field devices, allow device configuration and provide status monitoring with predictive diagnostic capabilities.

Based on these requirements, this thesis carries out an analysis of the industry standard communication protocols and components available for remote communication with HART field devices.

Furthermore, it presents an integration approach which describes the connection of the communication components, as well as how the communication protocols - HART, Profibus DP and Industrial Ethernet- are employed in order to achieve the integration of the HART data into software applications.

Subsequently, an evaluation of three software applications, able to integrate the HART field device's data in a comprehensive way, is performed.

The selected software applications for this evaluation make use of different standard technologies – FDT/DTM, EDD and OPC- to communicate with the HART devices and display the information provided by them, in addition to provide configuration and diagnostics functionalities.

In this way, this thesis attempt to present different solutions which could be implemented in NOV's system to access the HART sensors remotely and make it possible to configure and monitor the HART field devices.

This study has therefore been organized in various parts, as it follows:

- Chapters 3 and 4 give a detailed description of the industry standard protocols HART and Profibus DP. Additionally, a brief explanation about the Ethernet protocol is given in chapter 5.

- A theoretical description of HART data and system integration is encountered in chapter 6. In this chapter the different integration levels, which form the base of the experimental part in this study, are defined.

- In chapter 7 the HART field devices, communication components and software utilized in this study are described.

- Chapter 8 presents the configuration of the system with the different components - HART devices, PLC, Remote I/O modules and communication gateways- employed to carry out the integration of HART sensors, as well as it gives a description of their functions.

- The evaluation of the software applications is carried out through three HART data integration tests described in chapter 9. To conclude, the results of the tests are presented in chapter 10.

3 THE HART PROTOCOL

3.1 Introduction

The HART protocol was developed in the mid-1980s by Rosemount Inc. for use with a range of smart measuring instruments. The protocol was later published for free use by anyone, and in 1993, the registered trademark and all rights in the protocol were transferred to the HART Communication Foundation (HCF). However, the protocol remains open and free for all to use without royalties. The HART specifications have been improved and extended over the years (see appendix 1), always under the control of its users through the HCF, with efforts to preserve full compatibility with existing products. [1]

The description “smart” for a field device was first used in the sense of “intelligent” to describe any device that included a microprocessor. Typically, this would imply extra functionality beyond that previously provided.

The majority of process automation equipment utilizes a milliampere (mA) analog signal for field communication. In most applications the signal varies within a range of 4-20 mA in proportion to the process variable being represented. The HART (Highway Addressable Remote Transducer) Protocol is an industry standard protocol for sending and receiving digital information across analog wires between field devices and control and monitoring systems. It preserves the traditional 4-20mA signal, and provides simultaneous transmission of digital communication signals on the same wiring. Thus enabling a bi-directional communication with smart instruments without disturbing the 4-20 mA analog signal. In that way primary process variables and control signal information is carried by the 4-20 mA, while additional process measurements, device configuration and parameter information, calibration, and diagnostics information is accessible through the HART protocol.

3.2 HART devices and networks

Devices which support the HART protocol are divided into two groups: master (host) and slave (field) devices. Master devices are typically a DCS, a PLC, a computer based central control or monitoring system, as well as handheld terminals. On the other hand, HART slave devices include sensors, transmitters a various actuators that respond to commands from the primary or secondary host. The variety ranges from two wire and four-wire devices to intrinsically safe versions for operation in hazardous areas.

Basically the HART data is superimposed on the 4-20 mA current loop making use of the frequency shift keying (FSK) principle[1], via a FSK modem integrated in field devices. This enables devices to communicate digitally using the HART protocol, while analog signal transmission takes place at the same time.

Hart devices have the capability to operate in one of two network configurations: point-to point connection or multidrop mode. Using the polling address structure of the HART protocol, the polling address of the field devices will vary in a range of 0 to 15. [2, 3, 4]

3.2.1 Point-to-point

This is the most common use of HART technology. In this mode, the HART master device is connected to exactly one HART field device. This communication capability allows devices to be configured and set-up for specific applications. The 4-20mA signal carries one process (primary) variable, while additional data showing status, secondary variables, configuration parameters and preventive maintenance information are transferred digitally using the HART Protocol.

This connection alternative requires that the device address of the field device to be always set to zero since the HART master uses this address to establish communication. The analog 4-20 mA signal can be used for control in this point-to-point mode due to the digital signal is superimposed on the analog signal and does not interrupt or interfere or interfere with it. [2, 3]

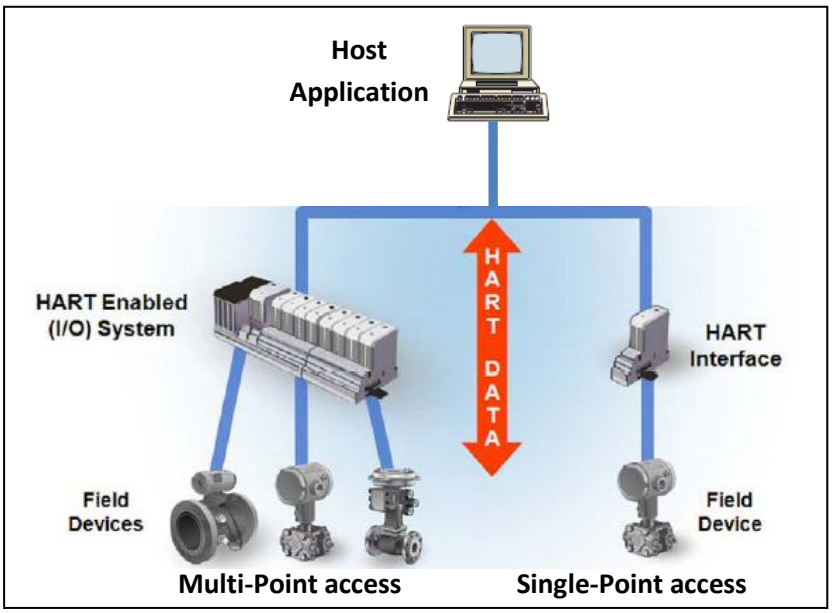


Figure 1 Point-to-point communication [2]

[1] See glossary.

3.2.2 Multidrop

In multidrop operation all process values are transmitted entirely digital, i.e. only via the HART protocol. The communication protocol enables the capability to connect several two-wire measurement devices in a (typical) multidrop network configuration. Depending on the protocol revision, HART 5 or HART 7, can up to 15 or 62 devices be connected in parallel to a single wire pair. In addition, properly connected two-wire loop powered and four-wire active-source devices can be used in the same network.

All the devices are supplied from one voltage source and with a constant current consumption (usually 4 mA). I.e. the analog current loop does not change, and it ceases to have any meaning relative to the process due to the analog signal that enters the master represents the sum of all the analog signals belonging to the devices in the network.

The host distinguishes the field devices by their preset polling addresses (or tag numbers) that must be unique in a range of 1-15 (potentially of 1-63). This address can be set by sending a special command to the devices. Standard HART commands are used to communicate with field instruments to determine process variables or device parameter information.

It can take several seconds before the HART signal, superimposed on the analog signal, follows the corresponding analog value. Process variables (normally transmitted on the 4-20 mA analog signal) are therefore much faster than that of the more accurate digital HART value.

The communication rate of the HART protocol in multidrop networks is perceived as too slow when it comes to monitoring and control of essential processes (see ch. 3.19). Standard HART commands are employed when it comes to communicate with field devices in order to determine process variables or device parameter information. Using HART protocol an average of 2-3 digital updates per second are expected, giving a typical cycle time of 500 ms to read information on a single variable from a HART device. Hence for a network of 15 devices, a total of approximately 7.5 seconds is needed to scan and read the primary variables from all devices. As the data field will usually contain values for up to four variables rather than just one, reading information from multivariable instruments may take longer. [2, 3, 4]

Significant reduction in the amount of wiring is among the advantages that multidrop network brings. In addition, more than 15 devices are possible to connect in a multidrop network when the devices are individually powered and isolated.

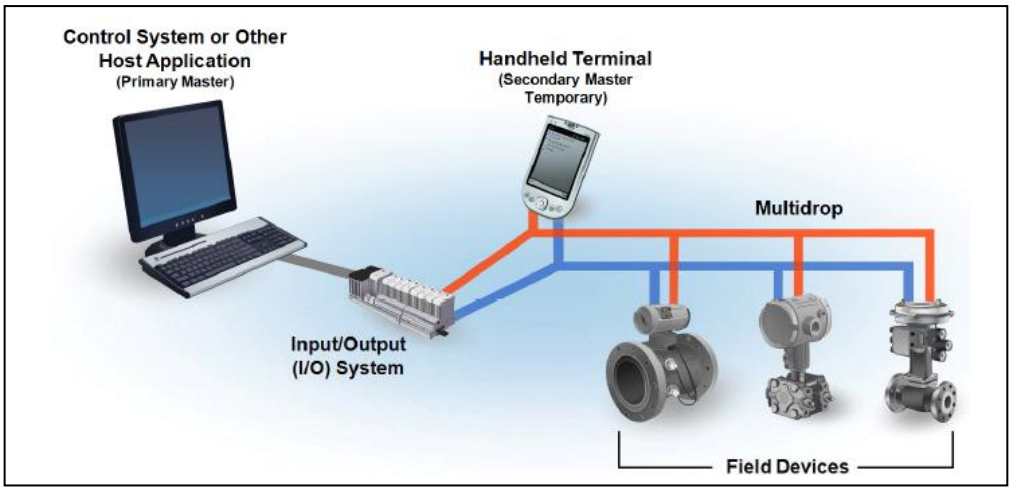


Figure 2 Multidrop connection [2]

3.3 HART Protocol and the OSI-model

The HART protocol is built on the Open Systems Interconnection (OSI) Model, developed by the International Standards Organization (ISO).

The OSI model describes the structure and elements needed for a communications system, sub-dividing it into layers where a collection of similar functions provides and request service from the layer above and below, respectively.

A simplified OSI-model is used by the HART Protocol. It implements layers 1,2,3,4 and 7 of the OSI 7-layer protocol model. The current HART Protocol is revision 7.2, where the "7" denotes the major revision level and the "2" denotes the minor revision level [5]. The following table (1) describes the OSI model layers and their functions:

Layer		Function	HART
7	Application	Formatting of data	Commands and data formats
6	Presentation	Conversion of data format	
5	Session	Management of communication dialog	
4	Transport	End-to-end error recovery	
3	Network	Switching and routing for network connections	
2	Data Link	Message format and media access	Message format and transaction rules
1	Physical	Physical connections and signals	FSK signal

Table 1 The OSI 7-layer protocol model

The information forming a message passes down through the layers in one device, along the wire, then up through the corresponding layers in the other device, as depicted in figure 3. In case of a single local network, with master-slave operation based on single standalone

transactions and without automatic retries for lost or corrupted messages, layers 3 to 6 are either not necessary or are much reduced. This leads to the outstanding simplicity of HART compared with other fieldbus protocols. [1]

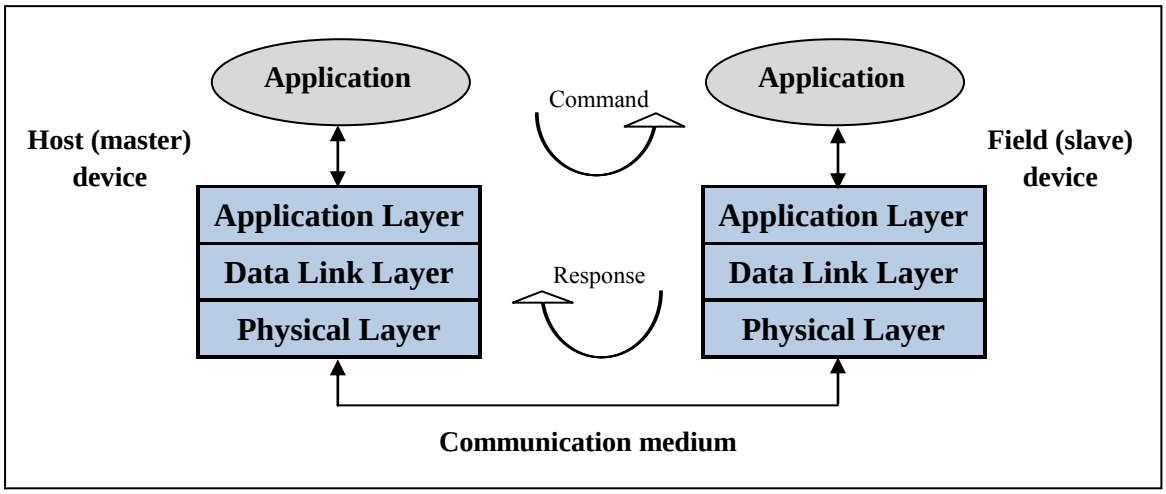


Figure 3 HART and the OSI model [1]

3.4 HART Communication Layers

Layer 1 is the HART **Physical Layer** which defines the physical and electrical specifications of the relationship between a HART device and the transmission medium (cable). Modulation is among the major functions and services performed by this layer. The HART Physical Layer makes use of the Bell 202[2] Frequency Shift Keying (FSK) principle to superimpose digital communication signals at a low level on the top of the 4-20 mA signal, and communicates at 1200 bits per second. The signal frequencies representing bit values of 0 and 1 are 2200 and 1200Hz respectively. The pulse amplitude of ± 0.5 mA averages out to zero as not to falsify the measured signal. [5]

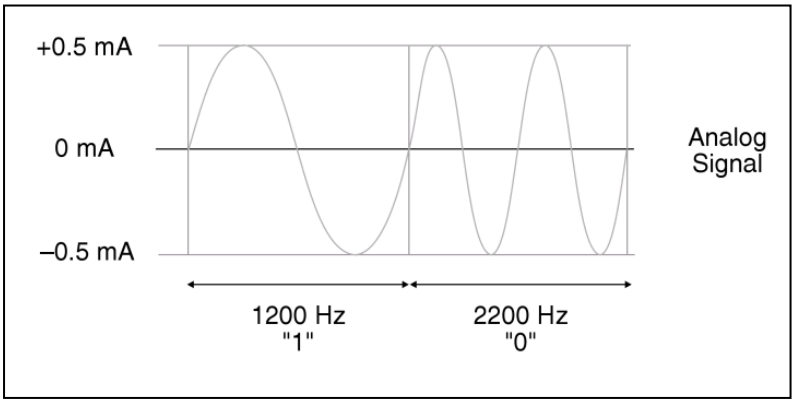


Figure 4 The HART FSK signal

[2] See glossary.

The two frequencies form sine waves which are superimposed on the direct current (dc) analog signal cables in order to provide simultaneous analog and digital communications. Because the average value of the FSK signal is always zero, the 4-20 mA analog signal is not affected. [2]

While the Physical Layer sends the current of bit, the HART **Data Link Layer** (Layer 2) provides the functional and procedural means that will ensure reliable sending of data. It is in this layer where the HART protocol technical format (structure?) is specified. The user can define here the communication Mode, either “Master-Slave” Mode where a field device only replies when it is spoken to, or “Burst Mode” which allows a single slave device to continuously broadcast a standard HART reply message. [2]

The **Network Layer** (Layer 3) provides routing, end-to-end security, and transport services, i.e. the operational and methodology means of transferring variable length data sequences from a source host to a destination host on separate networks. Furthermore, this layer manages "sessions" for end-to-end communication with correspondent devices.

The **Transport Layer** (Layer 4) provides and ensures successful direct transfer of data between end users. Typical examples of this layer are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Layer 7 is the HART **Application Layer** that defines the commands, responses, data types and status reporting supported by the Protocol. The commands make it possible for a system, connected to HART devices, to both collect and send data to the field devices. This can for example be collection of temperature measurements from a transmitter, or write in a name ("Tag") to the transmitter that can be used for device identification. Software applications implement a communication component that makes it possible to both the application layer and the user interacts directly with the application. [5, 6]

In the Application layer, the public commands of the protocol are divided into three major groups [2]:

1. Universal Commands - all HART compatible devices must recognize and support these commands.
2. Common Practice Commands - provide functions common to many, but not all HART communication devices.

3. Device Specific Commands - provide functions that are unique to each field device and are specified by the device manufacturer

A fourth one called Device Family Commands may be defined. It provides a set of standardized functions for instruments with particular measurement types, allowing full generic access without using device-specific commands.

3.5 Physical Layer

The Physical Layer connects to a medium such as a cooper or optical cable, which serves all of the communicating systems. It describes how HART circuit/loops can be build up, which type of cable could be employed as well as its longitude limits, and which signal components take part of the HART circuit/loops. This includes layouts of pins, voltages, impedances, hubs, repeaters, network adapters, etc. The layer also describes the communication signal, and the frequencies used for the digital communication.

In addition to interfacing to the network cable, the HART Physical Layer performs 4 basic functions [5]: modulating an outgoing message, demodulating an incoming message, turning on carrier for an outgoing message, detecting carrier for an incoming message.

3.5.1 Coding

HART is mainly a master/slave protocol. The data transmission between the masters and the field devices (slaves) is physically realized by superimposing an encoded digital signal on the 4 to 20 mA current loop. Since the coding has no mean values, an analog signal transmission taking place at the same time is not affected. This enables the HART protocol to include the existing simplex channel transmitting the current signal and an additional half-duplex channel for communication in both directions.

The bit transmission layer defines an asynchronous half-duplex interface which operates on the analog current signal line. To encode the bits, the FSK method is used and the two digital values, 1 and 0, have assigned frequencies of 1200 and 2200 Hz respectively.

HART utilises FSK with a data rate of 1200 bit per second, which is a very low rate compared with field buses. In HART both a logical 1 and a logical 0 operate in an equal time interval in order to obtain a correct data rate. Thus a logical 1 is represented by a single cycle of 1200 Hz, while a logical 0 is represented by almost two cycles of 2200 Hz. [3, 4, 8]

3.5.2 Digital and Analogue frequency

The analogue signal, where the digital signal (information) is superimposed on, has usually a frequency between 0 and 20 Hz. Thus the frequency to the digital signal lies much higher than the analogue's. Ideally and for that reason none of the signals (digital and analogue) will affect each other. Filters (high- and low-pass) are used in order to separate the two communication channels. The bandwidth occupied by the HART signal is conventionally stated as 950 Hz to 2500 Hz (see figure 5).

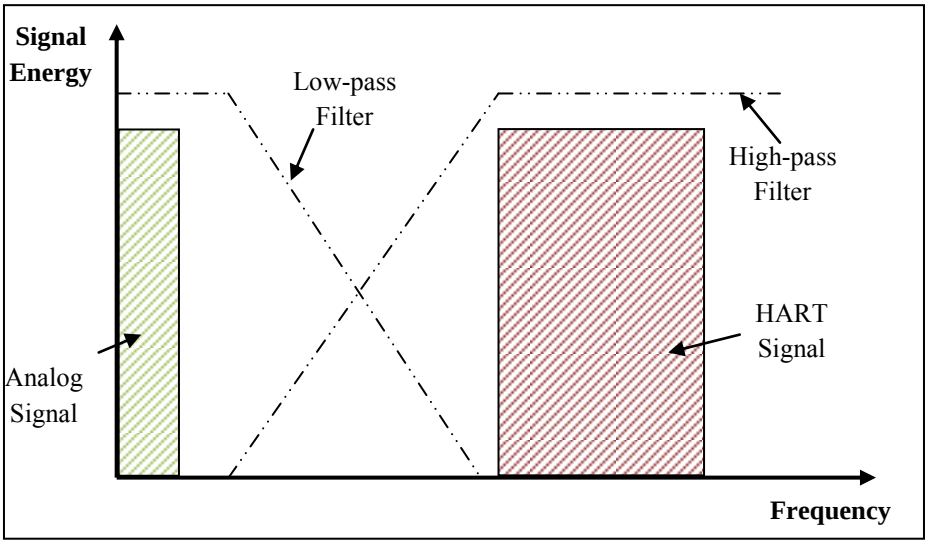


Figure 5 Analog and HART (digital) signal frequencies [1]

HART Communication between two or more devices can only function properly when all the communication participants have the capacity to interpret the HART sine wave signals correctly. Field devices, with inputs and outputs, in the current loop which are not specified only for the 4 to 20 mA technology, can impede or prevent the transmission of the data. Since the input and output resistances change with the signal frequency, such devices are likely to short-circuit the higher frequency HART signals (1200 to 2200 Hz).

3.6 Data Link Layer

This layer defines a request-response protocol. Any communication activity is initiated by the host communication device (master), which is either a control station or an operating device, and a (slave) field device only reply when it is spoken to. In other words the Data Link layer describes how the elements in a HART network communicate with each other. The layer gives a description of the messages' structure that are to be sent, and how they are identified by the right receiver (slave devices). It makes it possible to detect and correct errors that may occur in the Physical Layer. [2, 4, 8]

3.7 Master - Slave Protocol

HART is a protocol for Master/Slave communication that caters for up to two master devices. The primary devices - generally a DCS, PLC - send HART messages with commands, communicating with the field devices which receive the command messages. The command specifies which information the master is looking for to get from the slave. Primary master modules connected with the slave devices, constitute a HART circuit/loop. The secondary master devices – for instance a PC/laptop or a HART Handheld Terminal (HHT) - can be hooked up to the HART circuit/loop or directly to the field device. The two masters would have different addresses, 1 and 0 respectively. The use of handheld terminals is quite common within maintenance personnel of HART field devices, in order to collect data and to set the instrument parameters. [3, 4]

HART field instruments- the slaves- respond when they have received a command message from the master. Since HART is a half-duplex master-slave protocol, the master and the slave cannot send a message at the same time, but one of them has to wait while the other sends. Communication is initiated every time the master makes a request, and there is no communication between the different slaves or masters. Hence instant communication occurs just between one master and one slave.

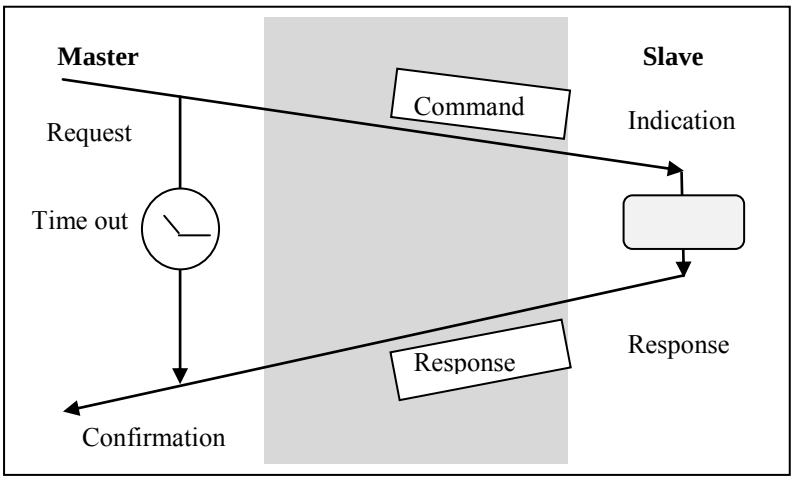


Figure 6 Master - Slave / Request – Response [3]

In case there are several devices in the network, only one of them can send a respond to the master at a given time. The emitted message by the master contains the address to the slave which the message is intended to. The slave sends right back a response to the message the master sent, showing that the slave received it, and the response contains the information the master requested.

The reply message contains also the address to the master it is aimed to. This will prevent an unintended master to get a response, in the event that two masters are active in a network. Once the data exchange between the control station and the field device is complete, the master will pause for a fixed time period before sending another command, allowing the other master to break in. The two masters observe a fixed time frame when taking turns communicating with the slave devices. [3, 4, 8]

3.8 Communication Modes

The HART protocol provides two modes for communicating information to/from smart field instruments and central control or monitoring equipment.

The simplest and most common form is the **Request-Response** (master-slave) mode communication simultaneous with the 4-20 mA analog signal, where a master telegram is directly followed by a response or acknowledgement telegram from the slave. This mode allows digital information to be updated approximately twice or three times per second in the master, without interrupting the analog signal.

A HART message from a master and its corresponding response from a slave are called a transaction. The two bursts of carrier during a transaction are illustrated in figure 7. The master is responsible for controlling the message's transaction. If it does not receive any reply from the slave after an expected time (RT1), the master will soon after send the same request again (RT2 – the time a master waits before sending a new request). The message is dropped after a couple of attempts without getting a response from the slave device, due to a fault in the slave or in the communication loop has most likely arisen.

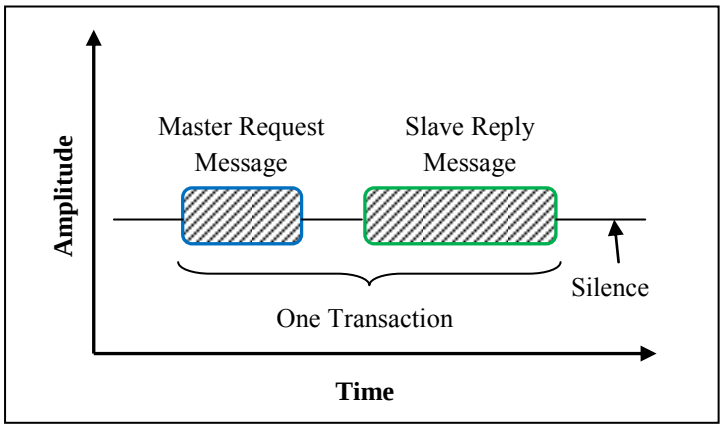


Figure 7 Carrier bursts during HART transaction

Some HART devices support the optional **Burst** communication mode. It allows a single slave device to continuously broadcast a selected standard HART reply message such as a primary variable or other. In this way the master is relieved from having to send repeated command requests. A single field device cyclically sends message telegrams with short 75-ms breaks, which can alternately be read by the primary as well as the secondary master. While usually two transactions per second are possible, Burst mode enables faster communication up to 3-4 data updates per second (will vary with the chosen command). The host receives the message at the higher rate until it instructs the device to stop bursting.

In a network it is only one slave at a time that can be in "burst" mode. This communication mode is therefore more relevant for single slave device networks, where the user wishes fast and continuous updates of particular important equipment data. [2, 4, 7, 8]

3.9 HART Character Structure (Character Coding)

A HART message consists of a series of bytes, that go through an UART[3] (universal asynchronous receiver/transmitter) before they are sent from the HART-modules. HART messages are coded as a series of 8-bit characters or “bytes”. These are transmitted serially using the UART function that serializes each byte, adding a start and stop bit (2 bits), and a parity bit (1 bit). Hence it converts each transmitted byte into an 11 bit serial character.

The original byte that contains the message data becomes the data bit, the start and stop bits are used for synchronization and the parity bit is part of the HART error detection. Thus these bits are used to identify every character in the receiver UART, and to check if an error in the character has occurred during the transportation from the sender to the receiver.

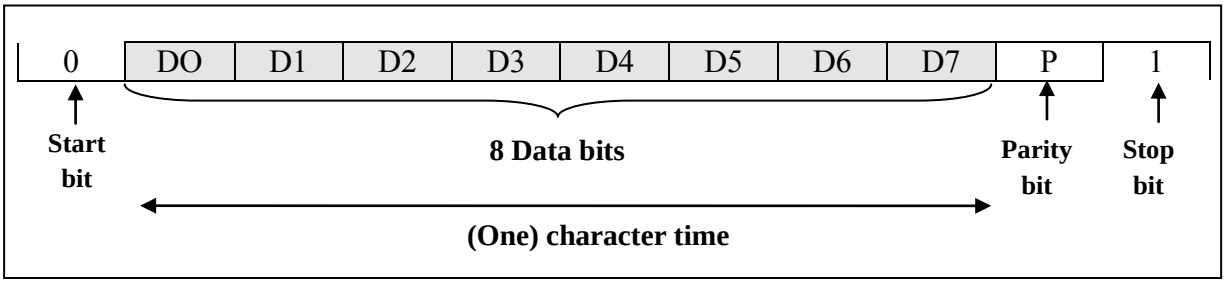


Figure 8 Bit structure of each HART character

The start bit to each single character is a logical 0, while the stop bit is a logical 1. The other bits in the character vary between 0 and 1, depending on the character.

[3] See glossary.

In HART communication, odd-numbers parity is used for error detection purposes. The 8 data bit determines the value of the parity bit. It is in the sender's UART where the value (0 or 1) is set. Furthermore, the number of logical 1's in the 8 bits is summed up in order to check that the sum of them is an odd number. If this is the case, then a logical 0 is set to the parity bit. Otherwise it is set to 1, so that the sum of all 1's in the character becomes an odd number. Consequently, each single character sent from the UART consists of a number of logical 1's that constitute an odd number.

The parities to every character is again checked in the receiver's UART, where an error in the transmission over the network is easily detected if the sum of 1's in the character is no longer an odd number. Noise and other disturbances could then be the cause of such errors. [7]

3.10 HART telegram structure and elements

Each command or reply is a message, varying in length from 10 to 12 bytes to typically 20 or 30 bytes. The structure of a HART telegram is depicted in fig. 9. Each individual byte is sent as 11-bit UART character equipped with a start, parity and a stop bit. HART provides two telegram formats, long and short format, which use different forms of addressing. On the other hand, the HART message structure[4] is equal for long and short format. [3, 4, 8]

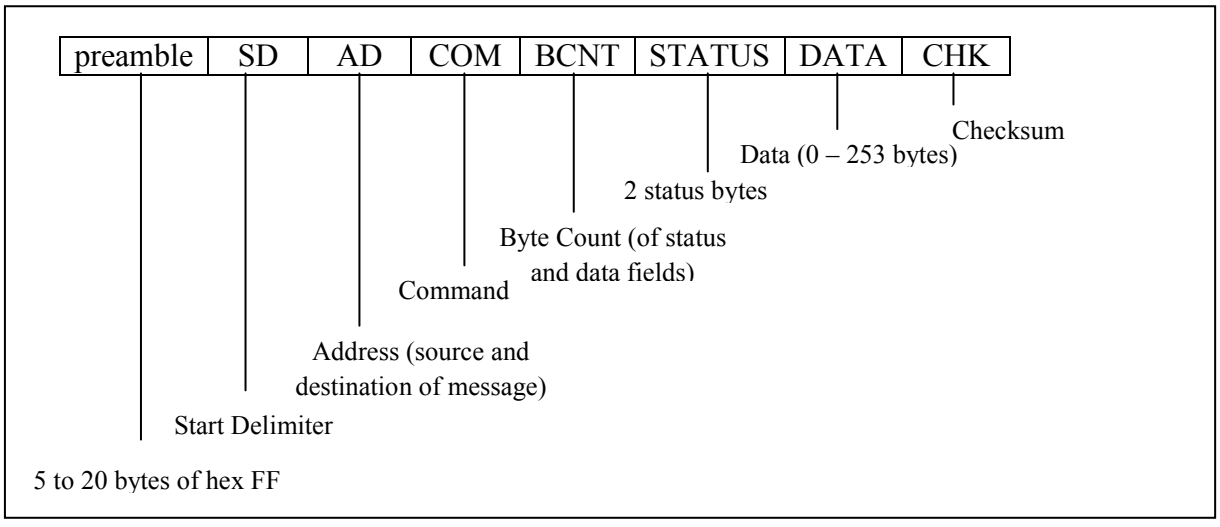


Figure 9 Telegram structure[1]

[4] The succeeding chapters, 3.10.1-3.10.9, provides a quite detailed description of the HART message structure. In this context, the reader is not "required" to go through them to understand how the HART protocol works.

3.10.1 Preamble

This element consists of 5 to 20 characters. All bytes in these characters are set to the logical value 1, in such a way that each character represents the hexadecimal number FF. All the characters will therefore represent a constant sinus signal.

The preamble characters synchronize the signals of the participants, as well as it is employed to detect the start of a message. A message receiver often needs some time to be synchronized with the signal frequency, and the incoming character stream. As well as it could need some time to turn the message stream through the modem after it has sent a message.

The receiving processor expects a sequence of 3 contiguous bytes: preamble, preamble, start delimiter. Thus, at least two good preamble characters must be received and they must be those that immediately precede the start delimiter. Since HART requires a transmission of a minimum of 5 preamble characters, there can be a loss of up to 3 characters during the synchronization at the receiver or by other occurrences, without damaging the message. If a message is received and the receiver sees only one preamble character, then the message will be lost because the receiver module does not manage to detect the start of the message. [1, 3, 4, 8]

Due to variation in number of preamble characters the different slaves need for being synchronized, the first message a master sends always contains 20 preamble characters. This is the maximum number allowed. A longer preamble means slower communication, therefore they should be avoided. To reduce the number of preamble characters needed for every message that is sent; a master can send either command 0 or command 11 to the slaves. The response from the slaves consists of the number of preamble characters the slave wants to receive. Given that the master module particularly demands a specific number of preamble characters, command 59 can be used to tell the slaves. Nowadays slave devices are designed so that they need only a 5 byte preamble (Five preamble characters utilises if no specification is given).

3.10.2 Start Delimiter

The start byte uses for recognizing a message's start. It indicates which participant is sending (master, slave, slave in burst mode) and whether the short frame or the long frame format is used. In case the message has the long format, the start byte will contain the hexadecimal value 82 when it is sent by a master. On the other hand, if the message is sent by a slave the

value is 86 while the value 81 indicates that it is a "Burst" message coming from a slave. The short format utilises other values.

When a Hart device waits for a response from a slave that makes use of the long address format, it will first wait for at least 2 "Preamble" characters and afterwards for a character with one of the named values in the previous paragraph. The HART device will just then know at it is a start of a message.

3.10.3 Address

This group of bytes contains both the master's address and the slave's address to a message. There are two frame formats, the long and the short frame format. This allows more participants to be integrated, while achieving more safety in case of incorrect addressing during transmission failures. The difference between them is the length of the slave's address. [3, 4]

Each HART field instrument must have a unique address. A message sent by a master contains the address of the target slave. When a telegram/message is sent on the HART network, the field device that recognizes the address as its own will read this message and send back a response.

It is just possible to make use of 2 master modules in every HART network. The address field of the short frame format contains one byte with one bit serving to distinguish the two masters and one bit to indicate burst-mode telegrams. The address's most significant bit (MSB) uses to identify the sender. If the value is a logical 1, the message is to or from a primary master. Thus, a logical 0 corresponds to a secondary master. [1, 4, 8]

Furthermore, the address's second most significant bit describes whether the response comes from a slave in "burst" modus or not. A logical 1 indicates that the slave is in "burst" modus.

The short frame format consists of only one byte. The last four bites are used for the addressing of the field devices (addresses 0 to 15). The 4-bit address could be set to any value from o to 15 using HART commands. If a master changed the address of a HART field instrument, it would have to use the new address from then on when talking to that particular instrument.

In messages with long addresses consisting of 5 bytes, the address area is on the other hand big enough such that all field devices (slaves) get their own permanent (unique) address.

Thus, quite practical from the user's point of view given that the slaves can be relocated within the network and yet be identified using the permanent (unique) address.

The address field of the long frame format contains five bytes; hence the field device is identified using 38 bits. The addresses of the field devices are set by the individual device manufacturer. The 38 bits are divided into three groups: the manufacturer ID code (byte 1), the device type code (byte 2) and the device-unique identifier (byte 9 to 11). [2, 4]

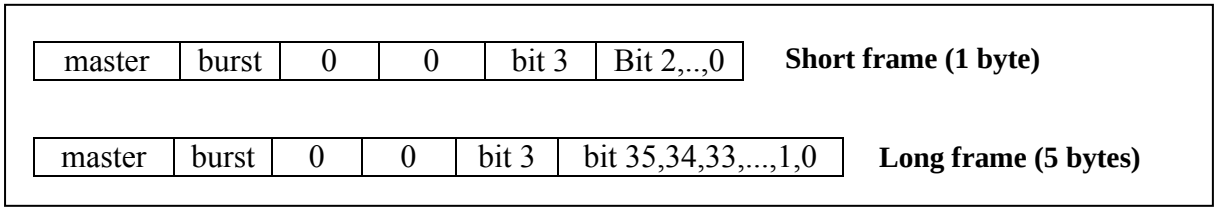


Figure 10 HART address format [4]

Early HART devices used only a 4 bit address (designed based on HART revision 4). Nowadays devices, utilising HART revision 5 or later, use a combination of the 4-bit address and the 38-bit address. The short address format is also known as the polling address, and it is used to quickly determine the field device's long address. The HART protocol rules state that HART Command 0 may be sent using the short address. This is done by the master right after a HART communication has started up. The Command 0 requests the field device to send a response that contains the long 38-bit address. [7]

3.10.4 Unique Identifier

Each HART device has a Unique Identifier, which is a 38-bit number (address) formed by the Manufacturer ID Code (6 bits), Device Type code (8 bits) and individual Device Identification code (24 bits). A unique address is encoded in each field device at the time of manufacture. This address is later used by a HART master to successfully communicate with a field device, ensuring that HART messages are received and acted upon only the intended device.

Unfortunately, the MSB (2 bits) of the Manufacturer Code are omitted due to lack of space, so there are potentially four devices with the same Unique ID. To reduce the minute chance of these devices coming together in a real HART loop, the HCF has made different rules for the allocation of Device Type Codes by manufacturers with Manufacturer Codes starting 00, 01, 10 and 11. [1, 2, 3]

3.10.5 Command

The command byte comprises an integer value in the range 0 to 253 (hex FD) that describes what a master requests a slave to respond. It encodes the master commands of three classes: Universal, Common-practice and Device-specific commands. The significance of these commands depends on the descriptions in the application layer 7. A command send by a master may enter information or new parameter values into the slave. The Slave reply message contains the same command value found in the request it is being answered. [1, 3, 4]

3.10.6 Byte count

This single-byte integer indicates the number of bytes left in the message, depending on the sum of the status and the data bytes (the checksum byte is not included in this count). This is the only way the receiver clearly identifies the telegram and the checksum, i.e. when the message is complete. [1, 4]

3.10.7 Status

The two status bytes are only used in reply messages from slaves. They contain bit-coded information such as possible communication errors and the operational state of the field device. This can e.g. be parity error in the slave's UART, the command is not implemented by the slave or that the device is busy. When the field device operates properly, both status bytes are set to logical zero. [1, 3, 4]

3.10.8 Data

The number of data bytes per telegram can vary from 0 to 25. These bytes comprise the data sent in a telegram. This can be information that a slave sends as response to a command, or information that follows a command sent to a slave. Unsigned integers, floating-point numbers or ASCII-coded character strings can be used for transmitting the data. The data format is determined by the command byte, however, not all commands or responses contain data. [1, 4]

Twenty-five data bytes or less were used by all Universal and Common Practice commands until HART rev. 5. On the other hand, HART rev. 6 specifies commands with up to 33 data bytes, and some devices utilizes even much longer messages in device-specific commands.

Since the byte count is itself just a single byte, the data field could never be more than 255 bytes (including the 2 status bytes). In the event longer data fields are used, transaction timing is then adversely affected. [1, 3, 4]

3.10.9 Checksum

The checksum byte contains the longitudinal parity of all the bytes which precede it in the telegram (message). The starting point is the "start" character. Through the rest of the telegram it is used an Exclusive OR (XOR) on the following bytes, together with the result of the previous XOR computation. This constitutes a Checksum byte at the end.

The byte is placed at the end of every message, and it is used by the receiver to control if there are errors in the messages. In combination with the parity bit attached to each individual byte, it creates a fairly good control over the communication. [1, 3, 4]

3.11 Error Detection on different levels

On the lower levels, the UART and the longitudinal parity check reliably detect any single burst of up to three corrupted bits in the transmitted telegram (Hamming distance HD = 4), with an excellent chance of detecting longer or multiple bursts.

Errors occurring on higher levels, such as HART commands that cannot be recognized and device failures are indicated by the slave device upon each transaction using the status bytes. By polling at regular intervals, the master devices is able to know the state of all connected communication participants and to react as requested by the user or the operating program. [1, 4]

3.12 Monitoring the HART network

Arbitration is employed to resolve access to the HART network. While Slaves access the network as quickly as attainable, Masters depend upon arbitration which is built on monitoring of network traffic and implementation of *timers*. Masters arbitrate by observing who sent the last transmission (a Slave or another Master) and by using timers to delay their own transmissions.

Any Master must be in condition to monitor the data transmission on a HART network, before it can start sending messages. To ensure communication between Masters and Slaves without disturbing any transaction, Masters monitor all network traffic.

In case there are two masters connected to the network, both will monitor any network activity in order identify the moment they can send messages to the slaves. Reply messages sent by a slave will thus be read by both Masters, but only the one the response is addressed to will receive and process the response. Yet the other master will monitor the transaction to take notice when the network is free to be used.

Masters use their timers to make sure it is free to use the network, and to be able to share it to send messages. Moreover, timers are used to determine the number of times a message has been sent without the slave receiving a correct response. This is done to avoid a master sending the same message repeated times, when there is probably an error on the slave or the network that causes an incomplete transaction.

Timers are logical elements that count down from a specified time interval. When the time is counted down to 0, the timers send a notification. These elements are usually used for synchronization of signals. The timers constitute dead time when no device is communicating and therefore contribute to "overhead" in HART communication. [1, 4, 7, 8]

3.13 Monitoring the network transactions

In pursuance of determining when a communication transaction may be initiated, a Master needs to be aware of when a message starts, stops, or is present. A combination of *carrier detect*, UART status indications and monitoring message content is required to achieve this. Monitoring of the network transactions is done by both the Master and the Slave. In that way they are able to send and receive messages.

A carrier detect indicates whether the incoming signal has a bigger amplitude than the lowest threshold value of a HART signal. The receiver would know then that if a carrier of acceptable amplitude is present, a signal that could contain a message exists. A carrier detect that turns up in a HART device instructs the device to examine its UART output and status.

In the UART status, a receive buffer full (RBF) indication will occur once each received character. The presence of a message on the network is thus determined by the combination of carrier detect and the RBF indications. Ideally, these indications would occur at a constant rate of one every 9.17 ms and the last one would correspond to the checksum character. This is the time it takes to send a character in HART communication. However, the RBF's don't necessarily indicate the end of a message or the start of another.

The transition from one message to another can only be identified by monitoring message content. A 3-character start delimiter indicates the start of a message. The sequence of these characters has such a structure that makes it very improbable that such a sequence could show up other places in a message.

The sequence consists of 2 characters with "preamble" bytes, and a start character. A HART message concludes with a "checksum" byte. If this is observed by the HART device, then it knows the message is close to its end.

According to the HART protocol rules, gaps between characters in a message are not permitted. It is nevertheless impossible that gaps between characters in a message appear. They occur when a Slave is not capable to keep up with the 1200 bps data rate. In the time of a gap the carrier is present but no information is being sent.

Most of the HART devices have timers which have values higher than 18 ms, which is the least time it takes between 2 RBF indications assuming that a gap size on the order of 1 character time would occur. Provided that device timers are longer than 2 character times, gaps will have no effect except to slow down communication.

In case the HART indications no longer occur, a HART device can interpret this as the message is through. [7, 8]

3.14 Synchronization

A master device must be synchronized to be able to send messages. When first connected to a HART network, the device is "unsynchronized". It becomes "synchronized" when it has been monitoring bus activity and has recognized the type and end of a previous message. Loss of synchronization takes place if the device's processor must briefly stop monitoring network traffic in order to perform other tasks, even if the master device was already "synchronized".

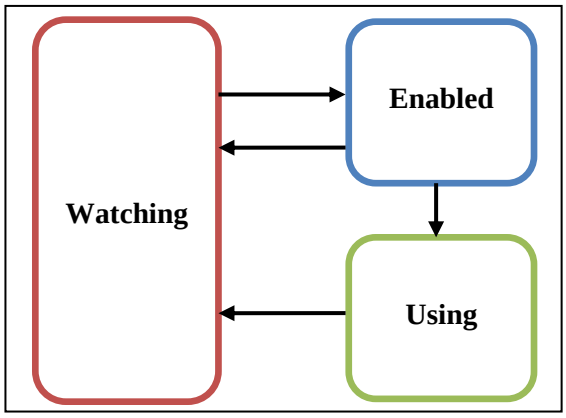
The master (bus) will also become "unsynchronized" if there is no network traffic after a length of time called RT1 since the last performed message transaction (e.g. if there is no response to a command). Message errors would also prevent a master device from knowing what occurs, causing loss of synchronization.

In case a master is new to the network, it must first wait a time RT1 before it becomes synchronized and starts to use the network. Assuming that two masters are present in the network and both are synchronized, then they will make use of the network at intervals. If a

given Master sees and recognizes a transaction of another Master with a Slave before RT1, then it is instantly synchronized. [7, 8]

3.15 Operational states

A Master device can be in three different states: *monitoring, enabled and using*. As mentioned earlier, during "monitoring" the master examine the communication on the network to determine the time it can make use of the network to send a message. This occurs while it waits to become synchronized.



When the master device is in the "enabled" state, it is able to send a message if intended. The master is in the "using" state when it has sent a message and waits for a response from that slave device the message was addressed to.

Figure 11 Operational states of a master device [8]

Master devices alternate between the named states while they are active and communicate on a HART network. The duration a master is on each state depends whether there is another master in the network and if the slave it is sending messages to replies.

A Slave device in a request-response mode, in which the field device is polled to provide dynamic measured values or instrument data, would always be in "wait" state. A Slave device will only reply when it receives a command message addressed to it, otherwise the message is rejected / declined/ discarded. A reply from the slave device acknowledges that the command has been received and may contain data requested by the master. The response is sent out before a time TT0 is over.

If the slave device is switched into “burst mode” when it receives a new message, then the message will be processed and a response sent out before the device continues to broadcast “burst” responses. In case the address of the received message is incorrect, the timer BT used in burst mode is set to RT1(0), so that the master waiting for a response gets clocked out. This is done to prevent that a master interprets a “burst” response as a response to another message. [1, 7, 8]

3.16 Token passing

HART communication protocol is token passing with each message, implying the passing of a token to another device. Token passing of multiple devices on a HART network can be seen as an implied token passing system where permission to transmit next is given by the communication carried in the start character and the master address bits of each message. Masters and Burst-Mode Slave take turns sharing the network. The token passing rules depend on whether a bursting slave device is present, as shown in appendix 2.

An important observation to make is that the slave addressed by a master may or may not be the same slave that is in burst mode.

The bursting slave gets an opportunity to send its burst message after every master-slave transaction. Two masters are supported using token passing to provide bus arbitration. Just like with normal master-slave transactions, the timing rules must be followed and if a device misses its turn, the token is considered as “lost”.

Token recovery is fast, if there is just one master present, it can determine that the other is absent and take over after only 75 ms. Likewise, if no master come up following a burst message a burst slave can broadcast its next burst message after only 75 ms. Note that loss of token is different as the bus becoming unsynchronized with a relatively long recovery time of 300 ms or more. [1, 8]

3.17 Timing rules

Due to the fact that collisions between messages or other communication errors occur when a Master and Slave devices communicate, it is decisive that every element that transmit messages in a HART loop has a fixed time for when and how fast a message or response will be sent/routed. The various masters and slaves have thus prescribed rules for response times and when the (internal) timer shall start and stop. Thus, timers are employed in the logic in order to steer/rule the transmission of messages or responses.

To prevent that a Master stays and wait infinitely for a response to a message, the Slaves have a maximum time it must answer within. This time is called TT0 (256 ms), and it is equal for all type of slaves. In case a slave doesn't manage to send a response within this period, the master will consider this as an unsuccessful transaction. TT0 is deliberately made quite large to accommodate less capable hardware and software that is likely to be found in a Slave. TT0

is set to that time it takes to send 28 characters on 11 bits in HART communication with a data rate of 1200 bits per second.

As specified previously, HART Master devices monitor the communication in the loop; detecting when a reply requested by a master is acquired. This is used by a primary and secondary master to be simultaneously active on the/a same network.

Two masters (if they are present) take turns at communicating with the slave devices. After each transaction is completed, one of the masters should pause for a short time RT2 (75 ms) before sending another command to allow a chance for the other master to break in if it attempts. I.e. after a master has received a message, it will start a "timer" with value RT2; which is equal for both the primary and secondary Master. Note that if three masters are connected and active, the timing rules fail and communication may be severely disrupted.

A slave in “burst” mode will repeatedly sends a data message. Between each of these messages the slave will wait a time defined as BT in order to let the master send a message to the slave between every reply message the slave sends while in “burst” mode operation. BT’s value is the same as RT2. Under these circumstances BT is the time it takes to send 8 characters with 11 bits HART communication protocol. [1, 7, 8]

The different “timer” values are often expressed as the time it takes to send a certain number of characters in HART communication. Table 2 review the most important Data Link Timers for the HART protocol.

Data Link Timer		Characters	Meaning
STO Slave Time out – **TT0**	256 ms	28	The slave must begin its response within this time.
RT1 Link Quiet (primary) – **RT1 (0)** RT1 Link Quiet (secondary) – **RT1 (1)**	305 ms 380 ms	33 41	An unsynchronized master waits for this time before transmitting. This ensures that no ongoing transaction will be interrupted. Different values for the primary and secondary masters ensure that the primary master has first access if both are connected simultaneously.
RT2 Link Grant – **RT2**	75 ms	8	A master waits this time after a response to itself, to allow another master to take turn if it wishes.
Burst mode time - **BT**	75 ms	8	The time a slave in “burst” mode waits between each response it broadcasts.

HOLD	20 ms	2	A master (or a bursting slave) must start its next command within this time after its access entitlement begins.

Table 2 HART Timers

3.18 Delayed response mechanism (DRM)

The DRM mechanism, introduced in HART 6, allows a slave device to indicate that it is unable to respond fully within the allowed time. The use of this added option to the HART protocol is however limited to bridge devices such as multiplexers or I/O systems; where there is often a speed difference and thus a delay in getting information from the field device.

Specific response codes report the initiation, continued existence or failure of the delay activity, so that a master can know how to proceed. Thus a slave implementing the DRM must provide a buffer to retain information regarding this operation. Bridging devices should rather maintain more than two buffers, so they can handle requests to each of its HART I/O channels.

Even while a DR is in progress for a write command, slave devices must always respond to read commands. In case a device runs out of buffers, it simply uses the normal "busy[5]" status in response to the master. [1]

3.19 Performance data of HART transmission

The bit data rate and the number of bits per telegram define the time required to transmit a telegram. The length of the telegram varies depending on the message length and the message format.

HART protocol uses FSK with a data rate of 1200 bps. Not all HART commands or responses contain data. In earlier HART revisions, 25 data bytes or less were used by all Universal and Common Practice commands. However HART revision 6 specifies commands with up to 33 data bytes.

Regarding HART as an asynchronous half-duplex protocol with 11 bits per character, 9.167 ms would be the required time to transmit a single character. The following example gives a brief into the transmission time of a HART telegram.

[5] If a HART field device is unable to respond fully to a command within the allowed time, it may reply that it is "busy"

Example

Consider a message using a short frame format and containing 25 characters. The following data applies to a HART transaction:

	Data
Time per bit	1 bits / 1200 s 0.83 ms
Bytes per telegram	25 message characters + 10 control characters
Telegram size	35 characters 11 bits 385 bits
Transaction time	35 characters 9.167 ms 0.32 s
User data rate	25 message characters (8 bits / 385 bits) 52 %
Time per user data byte	0.32 s / 25 bytes 13 ms

Table 3 HART transaction time

An average of 500 ms is accounted for per HART transaction, i.e. to read information on a single variable from a HART device, including additional maintenance and synchronization times. Consequently 2-3 HART transactions are expected to be carried out per second, showing that HART communication is not suitable for transmitting time-critical data. [1, 4]

3.20 Application Layer

Layer 7 - the Application Layer- of the OSI protocol reference model, provides the user with network capable applications. The communication procedures of HART master devices and operating programs are based on HART commands defined in this Layer. The HART command Set provides uniform and consistent communication for all field devices.

The public commands of the protocol are classified according to their function into commands for master devices and for field devices, defining four major groups: Universal commands, Common-Practice commands, Device-specific commands, and Device family commands.

These pre-defined commands enable HART master devices (HMD) to give instructions to a field device or send messages. In that way, actual values and parameters can be transmitted as well as various services for start-up and diagnostics performed. An immediate response is sent by the field devices in terms of an acknowledgement telegram which contains the requested status reports and/or the data of the field device.

Reply messages will always include two Status bytes, reporting any outgoing communication errors, the status of the received command and the operational state of the slave device.

The function of a HART command can be categorized as:

- **Read**: the field device responds with requested data; does not change the field device in any way.
- **Write:** sends a new value to be stored in the field device, e.g. measurement ranges and tag parameter.
- **Command:** instructs the field device to perform some action, which may involve writing to memory.

Data types used in these commands include integers, floats, alphanumeric, enumerated, bit and date formats. The command byte contains an integer in the range of 0 to 253 (hex FD), representing one of the HART commands. A command byte of 31 (hex 1F) indicates the presence of an extended (device family) command. [1, 2]

3.20.1 Universal Commands

These commands provide functions which must be implemented in all field devices. They provide access to information useful in normal operations such as read PV and units. Universal Commands are in the range of 0 to 30. Appendix 3 summarizes their functions. [1, 2, 3, 4]

Example of data structure for HART Command 9: Appendix 4

3.20.2 Common-Practice Commands

"Common-Practice" commands are in the range of 32 to 121. They provide functions implemented by many, but not necessarily all, HART communication devices. Functions such as read measured variables, set parameters are among others. Appendix 5 summarizes the command functions commonly used in the vast majority of devices including even the simplest devices available. [1, 2, 3, 4]

Commands in the range 122 to 126 are defined as "non-public". They are commonly used by manufacturers to enter device-specific information during assembly, e.g. the device identification number, which will never be altered by users or for direct memory read and write commands. Example of data structure for HART Command 35 and 109: Appendix 6. [1]

3.20.3 Device-specific Commands

These commands provide functions which are more or less unique to a particular field device. “Device-specific” commands are in the range 128 to 253. In HART revision 4 and earlier, device-specific commands always included the Device Type Code as the first byte of the data field to ensure that a command never reached an incompatible device. From HART revision 5 and on the use of Unique Identifiers guarantees that the host has fully identified the field device before any other command can be sent. Appendix 7 shows some examples of device-specific commands. [1, 2, 4]

3.20.4 Device family commands

Introduced in HART 6, it provides a set of standardized functions for instruments with particular measurement types (with specific transducer types), allowing full generic access without using device-specific commands. Furthermore, Device family commands offer improved interoperability between devices from different manufacturers, without using individual Device Descriptions (DD).

Device family commands are "extended commands" of 16 bits. These commands are in the range 1024 to 33791. The command set for each family, such as the "temperature device family" provides a consistent definition of data and configuration procedures.

The temperature device family describes any temperature measurement made using traditional copper, nickel or platinum resistance sensors (RTDs) or thermocouples.

Although the family specification includes special features, many are optional and may not be supported in a particular device. Appendix 8 lists some of the commands specified by this family. [1]

3.21 Data

A further part of layer 7 corresponds to the different kinds of information that HART field devices contain, as well as the relationships between this data and beyond the devices. These are described as follows [1]:

- **HART variables –** include all stored HART-accessible data items which are capable of being changed, either by HART command or by changing process conditions. Some of these data items are accessible via specific HART commands.

- **Device variables –** are the set of floating point numerical data items chosen by the device manufacturer to represent the process measurements. These are readable in groups of up to four by HART commands #9 and #33.

- **Dynamic variables –** are the set up of up to four data items representing measurements: the primary, secondary, tertiary and quaternary variables, abbreviated as PV, SV, TV and QV. The PV is only read by HART command #1, while command #3 reads all four and the current output or input in mA.

- **Configuration parameters –** are a considerable quantity of configuration data to set up the field device for a particular application. Many are floating point variables, such as the end-of-range values for which the analog output is to be 4 mA and 20 mA. Others will be integers representing e.g. sensor type or engineering units to be used. Some parameters are written and read by common practice HART commands; others may use device-specific commands.

- **Device information –** are the few data items which serve to identify the HART field device, such as manufacturer, device type and device revision levels. Others like alphanumeric tag or date are available for the user to set in relation to the particular application. The identification information forms parts of the response to HART command #0, while the user application-related information is accessible through universal commands #12, #13, #17, #18, #20 and #22.

- **Data types –** the HART protocol allows the following representations of data: Integer, Floating point, Alphanumeric, Enumerated, Bit and Date.

3.22 Establishing Communication with a HART Device

When first connecting to a field device, a HART master must access to the address of the field device in order to communicate successfully with it. To achieve this, a master can learn the address of a slave device by issuing Command #0 or Command #11. These and several other HART commands (e.g. #21, #73, #74) are available for initial communication, causing the slave device to respond with its address. [1, 2, 8]

- ***Command #0 – "Read Unique Identifier"*** is the most widely used method in order to launch communication with a slave device. It enables a master to learn the address of each slave device without user interaction.

Since HART revision 5, Command #0 is the only command which is recognized in the older short frame format, using the polling address 0 to 15 (0 to 63 in HART 6) as the device address. I.e. it can be used whether or not a field device has its tag set.

When individual polling addresses have been configured before installation, command #0 can also be used to scan all possible polling addresses in multidrop applications. If the unique ID of a device is already known, command #0 can also be used in the long frame format to obtain further device information [1, 2].

- ***Command #11 – "Read Unique Identifier associated with tag"*** requires that the device has already been configured with a short tag. Applicable in the event there are more than 15 devices in the network or if the network devices were not configured with unique polling addresses.

  Command #11 puts the short tag in the command's data field to specify which field device should respond. This is a long frame format command, thus the address field is set to the broadcast address of 38 zeros. Only the device with a matching tag will respond. [1, 2]

For bridge or I/O systems, *Commands #74 – "read I/O system capabilities"* returns the structure (up to 3 levels) and (maximum) number of sub-devices attached. *Command #75 – "Poll sub-device"* can then be used to poll individual sub-devices to find their unique IDs. Since it may be that not all possible devices are connected, response code 9 is used to indicate that no device has been found at a specified address. [1]

All the above commands (except #74) return the same identity data. Table 3 summarizes some of this information. Refer to Appendix 9 for the complete list of information returned.

Byte	Content
1	Manufacturer identification code
2	Device type code
4	Universal command revision (same has HART major rev. nr.)
6	Software revision
7	Hardware rev. (5 bits) and physical signaling code (3 bits)
9-11	Device ID number
13*	Maximum number of device variables

Table 4 Device Information

From the information returned in bytes 1, 2 and 9-11, the device's 38- bit Unique ID can be constructed. Once the Unique ID is known, the host will continue communication with the field device using the standard long frame address format. This applies only for devices of HART rev. 5 or above.

Knowing which HART revision is implemented in this field device, the host can anticipate which universal commands it should understand. [1]

4 PROFIBUS DP COMMUNICATION PROTOCOL

The Profibus DP (Decentralized Peripherals) is a master-slave protocol, optimized for high-speed and efficient transmission of user data. This Profibus protocol is designed specifically for communication between automation systems and decentralized field devices[6].

Profibus devices communicate using the Profibus DP protocol, which allows cyclical and acyclical communication and specifies rules for this. Profibus DP allows central controllers, such as PLC devices, Master devices or active stations (nodes) to control the bus and transfer messages without a remote request.

In fact, the data exchange for Profibus DP is generally cyclic in nature. The master cyclically prompt the connected slaves (passive communication nodes: field devices, I/Os, drives) to exchange data. The polled slave answers with a response message. A request message contains the output data, and the corresponding response message contains the input data.

Each node is polled cyclically updating the status and data associates with the node. A bus cycle comes to an end once all connected slaves have been polled in order. However, the bus cycle time should be shorter than the program cycle time of the controller, which for many applications is approximately 10ms. Besides the cyclic user data transmission, Profibus DP provides powerful functions for diagnostics.

In addition, a master can take the initiative to access data of a slave device in read or write mode acyclically. There can also be more than one master in a Profibus system. In that case, the access authorization passes from the active master to the next master (token-passing principle). [9, 10, 11, 12]

4.1 Profibus DP Layers

Profibus is based on international standards and oriented towards the OSI Layer model (ISO 7498). Profibus DP uses layer 1 (Physical layer), layer 2 (Data Link layer), and layer 7 (Application layer). This lean architecture ensures high-speed data transmission.

Layer 1 defines the physical transmission. RS-485 is the transmission technology most frequently used by Profibus. With RS-485 standard, layer 1 of Profibus implements symmetrical data transmission.

[6] Also referred as distributed peripherals at the field level (distributed I/O devices)

User Program			Application profiles
7	Application Layer		Profibus DP Protocol (DP-V0, DP-V1, DP-V2)
6	Presentation Layer		Not used
5	Session Layer		
4	Transport Layer		
3	Network Layer		
2	Data Link Layer		Fieldbus Data Link (FDL): Master Slave principle Token principle
1	Physical Layer		Transmission technology
OSI Layer Model			**OSI Implementation at Profibus**

Table 5 OSI Layer Model – Profibus DP

Transmission speeds of 9.6 Kbit/s to 12 Mbit/s are available, and the selected baud rate applies to all devices connected to the bus (segment). The RS-485 transmission procedure used for Profibus is based on semi-duplex, asynchronous, gap-free synchronization, where data is transmitted in an 11-bit character frame in NRZ code (Non Return to Zero).

When RS-485 transmission technology is used, all field devices are typically connected in a line structure with up to 32 nodes (master and slaves) in one segment. The beginning and end of each segment are provided with active bus termination, which must be supplied with power continuously.

Another physical medium is fiber optic, which greatly extends the bus length at high transmission speeds. Fiber optic cables permit transmission distances of up to 15 km between the stations of a Profibus system. [9, 10, 11, 12]

Layer 2 defines bus access control, data security and processing of transmission protocols and telegrams. With Profibus, Layer 2 is called the FDL Layer (Fieldbus Data Link).

The bus access control determines when a station may transmit on the bus and Profibus supports two mechanisms, namely, a decentralized token passing procedure for communication between the active nodes (master), and a centralized master-slave procedure (polling) for communication between the active and the passive nodes.

When an active node (bus station) has the token, it takes over the master function on the bus to communicate with both passive and active nodes. The exchange of messages on the bus is organized by means of node addressing. Each Profibus node is given an address which must be unique throughout the entire bus system. The maximum usable address range within a bus

system lies between addresses 0 and 126. This means that the bus system can have a maximum of 127 nodes (bus stations). In addition, address 127 is used and reserved as the broadcast address, which all slave devices on a network recognize. [11]

Token passing is used for communication between multiple masters on the bus. It involves the passing of software token between masters, in ascending order of their bus addresses. Thus, the active nodes connected to the Profibus network form a logical token ring. For reasons of security and data consistency, a slave can never participate in cyclic data exchange with more than one DP master simultaneously. If there are multiple masters on one bus, different slaves may be assigned to different masters. [10, 11, 13]

The polling method, on the other hand, is used by a master that currently has the token (right to send) to communicate (address) with its associated slave devices (passive stations). The master can send messages to the slaves or fetch messages from the slaves. The typical standard Profibus DP bus configuration is based on this bus access procedure. [10, 11]

Layer 2 telegram formats (see figure 12) provide a high degree of transmission security. The call telegrams have a Hamming Distance of HD=4, which means that up to three simultaneously distorted bits can be detected in the data telegram. This is achieved by selecting special start and end identifiers for the telegrams, by using gap-free synchronization, and by using a parity bit and a control byte (IEC 870-5-1 standard). [10, 11]

The Fieldbus Data Link layer allows multiple-point transmission with Broadcast and Multicast Communication. With Broadcast, an active station sends a message to all other stations (masters and slaves). With Multicast, an active station sends a message to a group of stations. Receipt of the data is not acknowledged for either of these communications. [11]

4.2 Profibus DP Character Format

All Profibus characters are comprised of 11 bits (1 start bit + 8 data bits + 1 even parity bit + 1 stop bit). This character frame applies to all data (character) bytes, comprising also the telegram header bytes. Profibus DP exchanges data in NRZ code, which means that the shape of the signal during the transition from binary "0" to "1" does not change while the bits are being transmitted. In case nothing is being transmitted, the idle state potential on the line is "1". A start bit causes the line to go to "0". [11]

During the transmission of messages on Profibus serial networks, each character (or data byte) is sent in the order of LSB to MSB. While for word transfer (for more than 1 byte) the high byte is transmitted first and followed by the low byte.

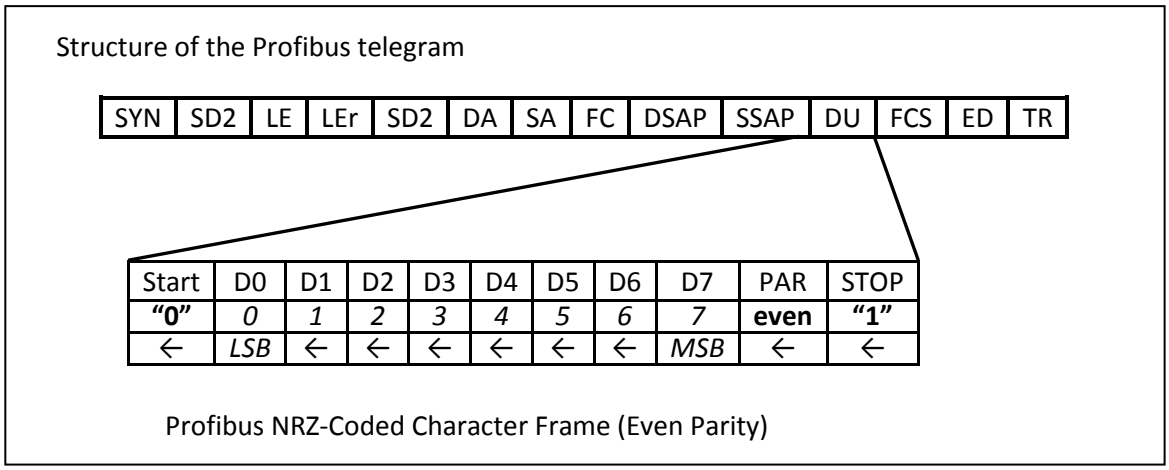


Figure 12 Profibus telegram structure [17]

4.3 Profibus Telegram Structure

Profibus telegrams may contain up to 256 bytes: up to 244 bytes of data per node per message, and 11 bytes of overhead (referred as the telegram header). Except for Data_Exchange telegrams which have 9 bytes of header information (DSAP and SSAP bytes are dropped), all telegram headers are 11 bytes.

Since 11 bytes of overhead for a single message is a lot for a single message (containing small amounts of data), large amounts of transferred data makes Profibus more efficient. An idle state of at least 33Tbits (sync-time in bit time) must be present before every request telegram to be sent. Furthermore, all data is transferred without gaps between individual characters and any master-slave data exchanges are handled in the telegram using Service Access Points (SAP). Profibus DP uses SAP's 54 to 62, plus the default SAP (Data_Exchange) - see Appendix 10. [13, 14]

Abbr.	Nr. of bytes	Description
SYN	33 bits	Bus idle state
SD2	1 byte	Start Delimiter 2; distinguishes between different telegram formats
LE	1 byte	Byte length. Indicates the length of the information field in telegrams with variable length (DAT + DA + SA + FC + DSAP + SSAP)
LEr	1 byte	Byte length repeated
DA	1 byte	Destination Address byte. Indicates where the message goes to
SA	1 byte	Source Address byte. Indicates where the message came from. The address of the sending station

FC	1 byte	Function Code (FC=Type/Priority of this message). Used to identify the type of telegram, such as request, acknowledgement, or response telegrams. Contain details about the priority of the message
DSAP	1 byte	Destination Service Access Point (COM port of receiver). The destination station uses this to determine which service is to be executed
SSAP	1 byte	Source Service Access Point (COM port of sender)
DU	1 to 32 bytes/ 1-244 bytes	Data bits. Contain the useful information of the telegram.
START	1 bit	Start bit
D7...D0	8 bits	Data bits
PAR	1 bit	Parity bit
STOP	1 bit	Stop bit
FCS	1 byte	Frame Checking Sequence. Contains a telegram check sum which is obtained adding all telegram elements without using a carry bit
ED	1 byte	End Delimiter. Indicates the end of the telegram
TR	8 bits	Minimum delay (8 bit timing)

Table 6 Profibus DP Telegram Header Abbreviations and Frame Bytes

Layer 7 provides the application services to the user. These services make an efficient and open data transfer possible between the application programs and layer 2. The Application layer of Profibus consists of the FMs interface (Fieldbus Message Specification) and the LLI interface (Lower Layer Interface).

From the viewpoint of an application process, the communication system is a service provider offering communication services, known as the FMS services. The LLI conducts the data flow control and connection monitoring as well as the mapping of the FMS services onto layer 2, with due consideration of the various types of devices (master or slave). [10, 11]

4.4 Communication with Profibus DP

In order to fulfil the requirements of different areas of use, the functions of the Profibus DP communication protocol are distributed over three performance levels.

- **DP-V0**, which supplies the basic functions of the communication protocol; particularly cyclical communication and device-, module- and channel- specific diagnostics for quick fault localization.
- **DP-V1**, augments functions for acyclical communication to DP-V0. Functions such as parameterization, operation, monitoring and alarm handling. Furthermore, DP-V1 enables online access to bus nodes via engineering tools for this aim. See figure 13.

- **DP-V2,** encloses supplementary functions as extensions of DP-V1, primarily functions which are required for drive control. These consist of functions for communication between slaves, cycle synchronization and time stamping.

Field devices for process automation are generally slaves of performance level DP-V1 and they can thus communicate acyclically to set device parameters. [12, 13]

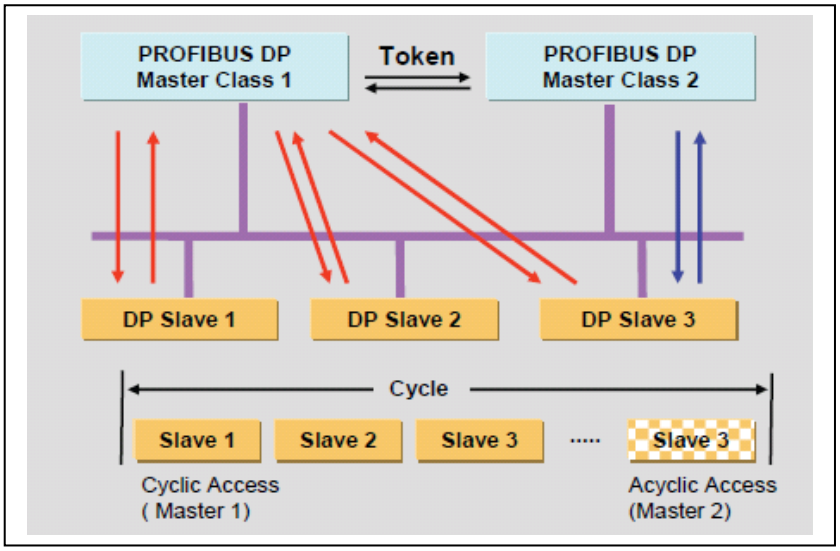


Figure 13 Cyclical and acyclical communication with Profibus DP-V1 [12]

4.5 Type of Bus devices

Based on their functions, Profibus distinguishes between the following types of device:

4.5.1 Profibus DP Master (class 1) - DPM1

A Profibus DP master of class 1 handles the cyclic exchange of input and output user data with its associated DP slaves. A DP master can transmit parameter sets to the DP slaves, read the diagnostic information of the DP slave and use control commands to inform the DP slaves of its operational status.

DP masters can address individual slaves or to specified groups of DP slaves (multicast), or can broadcast a telegram to all connected slaves. The Slaves will return a response to all telegrams addressed to them individually, but do not respond to broadcast or multi-cast telegrams from a master device.

DPM1 devices are often integrated in a memory-programmable controller (such as a PLC) or an automation station of the process control system. [11, 12, 15]

4.5.2 Profibus DP Master (class 2) – DPM2

Profibus DP masters of class 2 are generally employed for device configuration, maintenance and diagnostic purposes, as well as to set device parameters. In this way, the associated exchange of data takes place if necessary and therefore acyclic communication is required.

Class 2 DP masters read the input and output data of DP slaves at the same time as data communication with the DPM1 takes place. They also read current configuration data of DP slaves, and are allowed to assign a new bus address to a DP slave (provided that the slave supports this).

A DPM2 device is not to be permanently connected to the bus system. Thus, for example, a configuration tool based on a DPM2 can write the required parameter sets to Profibus slaves and can be removed from the bus afterwards.

Like a DPM1, a DPM2 is a device with active bus access, but unlike a DPM1, a DPM2 does not use this bus access for process control due to the lack of cyclic communication functionality. Devices of this type are usually part of an engineering station used for device configuration. [11, 12, 13]

4.5.3 Profibus DP Slave

Profibus slaves are usually field devices (I/O devices - transmitters/sensors[7], actuators[8], valves, measuring transducers) which acquire process variables or actively influence the process. Slave devices do not have bus access rights and they can only acknowledge received messages, or send response messages to the master upon request, forming a passive station on the network.

It is important to note that all Profibus slaves have the same priority, and all network communication originates from the master. Besides, a DP slave only exchanges user data with the DP master that was responsible for loading its parameters and configuring it.

Modular and compact slave devices can be distinguished. A modular device encompasses a head station containing the fieldbus interface and a number of slots into which various modules can be inserted. Compact devices are equivalent to a modular device with exactly one permanently installed module, i.e. they have a fixed set of input and output data. [11, 12, 13, 14]

[7] Sensors collect state and process data and provide the master with this information.
[8] Actuators receive input information from the master and actively influence the process.

4.6 Data transfer between DPM1 and DP slaves

Layer 2 provides the application layer with SDR (Send and Request Data with reply) and SDN (Send Data with No acknowledge) communication services. The access of the application layer to these forms of communication is granted via the so-called Service Access Points (SAP).

With the SRD service the master receives a reply from the slave within a defined time span, which consists either of an acknowledgement or of the requested data.

The SDN service sends data to a whole group of slaves. However, a master-controlled bus assignment for slave replies is not possible in this case so that SDN telegrams remain unacknowledged.

Figure 14 shows the telegram sequence between a DP Master class 1 and a DP slave. [15]

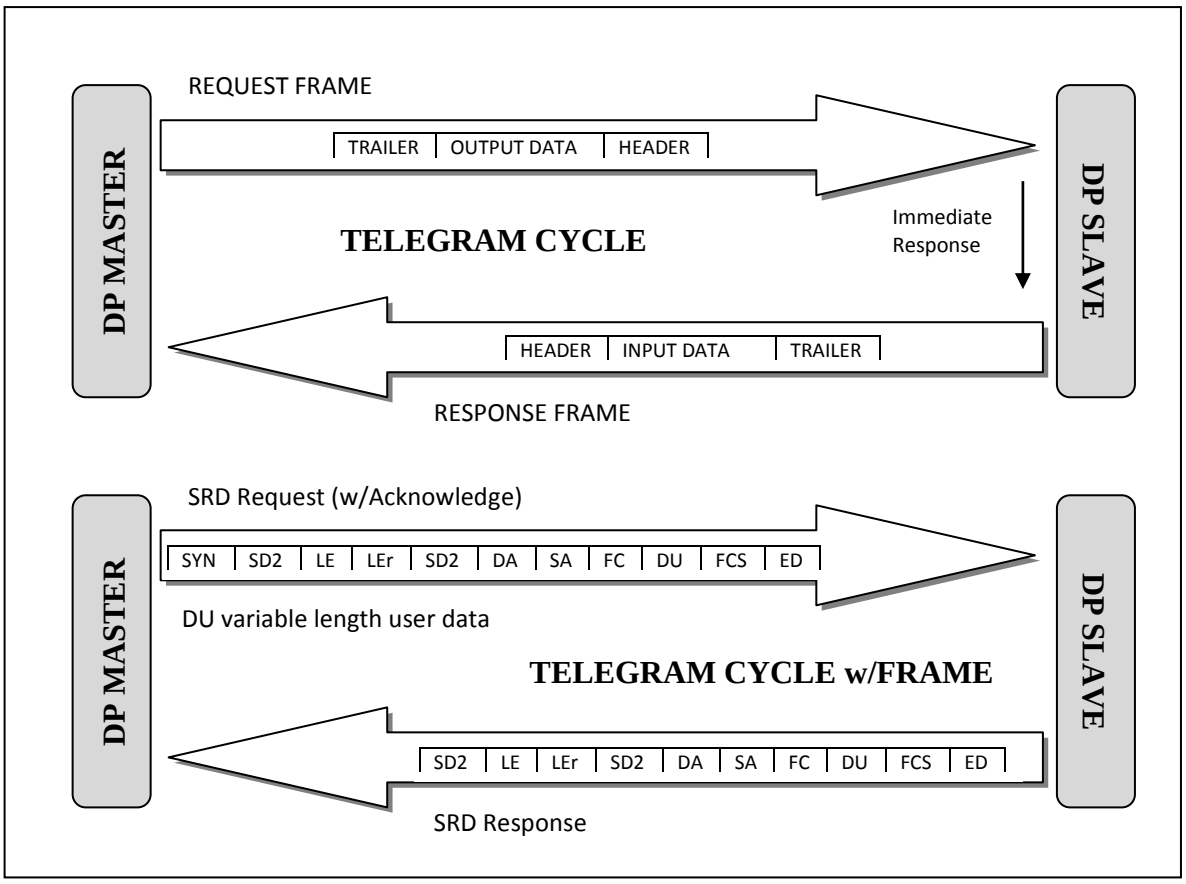


Figure 14 User data exchange for Profibus DP.

4.7 Cyclical communication and Profibus diagnostics

Once the configuration has been loaded on the DPM1 device, the master establishes the cyclic data exchange of the inputs and outputs with the DP slaves assigned[9] to it (*MS0 channel*). First the parameters set in the configuration, for instance master address, watchdog time and ID number, are sent to the slave (*parameterization*) and checked (*configuration*).

Since the ID number is unique for each device type, cyclical communication will only take place if the ID number from the configuration tallies with the ID number saved in the slave. Furthermore, cyclical communication can only be established if the modules which are physically present tally with those set in the configuration or if the device can comply with the configuration received.

The DP slave, in turn, checks the configuration internally. Later the DPM1 device polls the diagnostics, anticipating that the DP slave is ready to respond to the check. Hence, whenever a DPM1 device wants to establish cyclic data exchange with a DP slave, it first reads the requested diagnostics data (after parameter assignment and configuration).

The slave reports invalid parameter or configuration data to the master through corresponding errors in the Profibus standard diagnostics. In case the parameter and configuration data is valid, the master will initiate cyclical communication with the slave device.

Profibus diagnostics comprise both Profibus-specific standard diagnostics and advanced diagnostics[10]. With DPV1 the device-specific diagnostics were further refined with the addition of alarm messages and status messages. Any changes to device-specific diagnostics data are reported by a slave in the response message during cyclical communication; the master will respond accordingly in the next bus cycle by polling the diagnostics data, rather than the process data, of the slave concerned. [12, 13]

4.8 Acyclical communication and parameter addressing

An essential part of the acyclical data exchange process is the writing or reading of device parameters (e.g., units of measured value, linearization curve, etc.) based on a master-issued request. Such device parameters can then be used by a centralized operator tool to configure a field device, thereby adapting it to the specific tasks.

[9] Refers to a DP slave that can only engage in cyclical data exchange with one DPM1
[10] Contains device-specific diagnostics data relating, e.g., to measuring or setting procedures.

For this acyclic communication, there are two different channels, entitled as the MS1 and MS2 channels. In this context, a connection between a master and a slave (MS1 link) via the MS1 channel (MS1 connection) is only possible if cyclical data exchange is taking place between the master and slave. Because a slave is only able to exchange data cyclically with only one master at a time, a slave can only have up to one MS1 link.

On the other hand, a slave can establish multiple connections to more than one master simultaneously via the MS2 channel, as long as it is not engaged in cyclic communication. However, the MS2 connection has to be established explicitly by the master.

Unlike for cyclical communication, a complex configuration based on the device master file is not required for acyclical communication. Knowledge of the address of the device concerned is usually enough to establish an MS2 link on the master side. The master that initiates the data transmission over the MS2 channel is referred to as a master class 2 (DPM2).

Device parameters are addressed in a slave device by means of the specification of the slot and index. The "slot" (values from 0 to 254) is a slot on a modular device. The "index" (0 to 254) is the address of a parameter within the slot concerned. [12, 13]

A DPM1 will automatically detect the presence of a DPM2. Once the DPM1 has completed its polling cycle, it will pass a token to the DPM2 granting it temporary access to the bus. Hence, the DPM2 is able to exchange data with all the slaves within a specific period of time called the token hold-time (or half-time). During this time, the DP slave will stop its normal data exchange with its DPM1.

The DPM2 may then proceed to read data or diagnostic information from any of the slaves. Once completed its cycle, it will pass the token back to the DPM1. In that way, asynchronous data is fitted into the time gaps between the synchronous telegrams and the deterministic[11] behaviour of Profibus maintained because the DPM2 can only use the time allotted to it via the gap time specified. [16]

Since there usually is not enough time during the gap to complete a full data exchange, this process of data retrieval by the DPM2 may continue over several cycles.

[11] The *determinism* of a system refers to the ability to precisely predict the behavior of the system over time.

4.9 Profibus performance

As stated earlier, Profibus uses a polling mechanism between DPM1 device and DP slave. The reaction time of a Profibus system primarily depends on the reaction time in which a slave can respond, the selected transmission rate (baud rate), the minimum slave interval and the agreed net data length. [17]

The following characteristics regarding data transmission using Profibus DP can be entitled:

- ♣ Transmission rate 1.5 Mbaud (Mbit/s) is commonly selected.
- ♣ A Profibus telegram may contain up to 255 bytes. Up to 244 bytes of input/output data per message plus 11 bytes of overhead (telegram header). All telegram headers are 11 bytes, except for Data_Exchange telegrams which have 9 bytes of header information (DSAP and SSAP bytes are dropped).
- ♣ High efficient transmission when large amounts of data must be transferred, since the output data is sent and the input data is received in a single telegram cycle.
- ♣ Approximate time for an information cycle depends on the number of input data bytes (from slave device) and output data bytes (to slave device). Additionally the time for 1 telegram cycle (not including data) in bits (or microseconds).

Due to the determinism of Profibus, it is possible to calculate a reliable system reaction time. This reaction time remains unchanged even if the Profibus system receives many I/O signal changes at some point in time. This includes the presence of a DPM2 performing diagnostics on a slave device while it is communicating with its DPM1, because the DPM2 will not be allowed to use more than the specified gap time within the bus cycle[12]. Regarding the calculation of the reaction time of a Profibus system, the following terms are defined [17]:

Term	Definition
T_{SYN}	Bus time-out time for the synchronization, refers to the minimum time a slave must remain in the idle state before it can accept another request.
T_{SDR}	Slave reaction time; refers to the time it takes a slave to respond a message.
T_{IDI}	Initiator Idle time; refers to the slave delay of the initiator of a request (master).
T_{MC}	Time of one telegram cycle (not including data) results from the addition of the bus times ($T_{SDR} + T_{SYN} + T_{IDI}$) and the telegram header

[12]Refer to Acyclical communication and parameter addressing chapter.

Min_Slave_Interval	Minimum Slave interval; refers to the minimum time which has to elapse between two subsequent requests to the same slave.

Table 7 Profibus terms

The following example gives a simplified description of the reaction time calculation of a Profibus system:

One master and five slaves are connected via Profibus DP. 10 bytes of output data and 20 bytes of input data are to be transferred per slave at 1.5 Mbaud. One character is made up 11 bits.

The minimum interval for one information cycle results from the addition of the bus times and the telegram header. In the data exchange, the header comprises 9 bytes. [17]

Assuming that the bus time-out for the synchronization is 33 bits, T_{IDI} is 36 bits (at 1.5 Mbaud) and T_{SDR} = 30 bits (typical value for an ASIC[13]), the following calculations can be performed:

	Calculations
1 bit at 1.5 Mbaud	—— 0.67µs 670ns
1 character = 11 bits	1 character requires 11 670 ns 7.33µs
T_{MC} / bit	Telegram header + bus times (2 length header(bytes) 11 ——) + (T_{SDR} + T_{SYN} + T_{IDI}) (2 9 11) + (30 + 33 + 36) 300 bits or 300 bits 670 — 201 µs
Approx. Time for an information cycle (including data)	201 µs + 30 bytes (net data) 7.33µs 420.9µs per slave or approx. 2.1 ms for 5 slave stations

Table 8 Profibus performance

The resulting approx. time for an information cycle indicates the time where the SRD service sends data from the outputs and receives data from the inputs in one telegram cycle.

[13] Refer to glossary.

5 INDUSTRIAL ETHERNET

This chapter provides a very general introduction to the Industrial Ethernet (TCP/IP) communication protocol. A detailed description of this protocol and its structure is not part of the scope of work for this study. However, communication components that implement data transmission via Ethernet (TCP/IP) are employed to enable access to HART field devices.

Industrial Ethernet (TCP/IP)

The various industrial communication protocols specify mechanisms to embed their protocols in the Ethernet standard (IEEE 802.3), which addresses only the lower layers of communications networks and not the meaning of the data it transports.

The protocol adopted in Industrial Ethernet is TCP/IP. This standard protocol for data transmission consists of two approaches in order to support the application layer of the corresponding industrial protocol.

First, the industrial protocol is simply encapsulated in the TCP/IP. In that way the protocol controls the transmission and thus the actual data transfer, allowing the shortest development time for defining industrial protocol transportation over TCP/IP. Furthermore, TCP provides guaranteed delivery of the messages, thus adding protocol overhead in the form of acknowledgements to every message. The second approach, the Internet Protocol (IP), is required to unambiguously address a computer in a network.

The Internet Protocol (IP) provides the addressing and routing of data packets from the transmitter up to the receiver over a network. Hence, each station that communicates with another one is identified by an unequivocal IP address.

The data packets with IP are called datagrams. The IP is a connectionless service with an Unreliable Datagram Service. This means that each datagram is delivered in isolation and with IP alone; where there neither connections nor logical circuits. Besides there is no guarantee that the datagram will ever arrive at its destination, or arrive duplicated; due to neither the correctness of the data nor observation of the sequence, completeness and unambiguity of the datagrams is checked at the IP level.

On the other hand, the Transmission Control Protocol (TCP) is a connection oriented, end-to end, reliable protocol. In this way, upper application layers need to implement mechanisms aimed at increasing reliability, since the TCP demands acknowledgement (within specified

time-out periods) of all data sent and in the event such acknowledgements fail to arrive, TCP will re-send the data. Hence, the unreliable nature of lower level protocols such as IP is overcome.

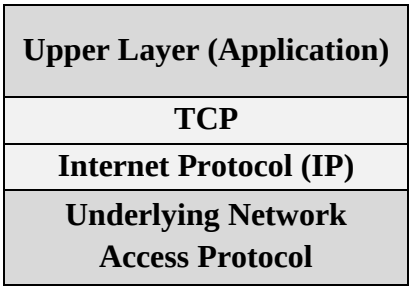


Figure 15 TCP/IP

An application passes data to the TCP module for eventual transmission onto the local network, and delivery to a destination host. Consequently, the TCP module, in turn, calls upon the underlying protocol (IP) that packages the unit of TCP data into a datagram. Then, this layer passes the datagram to the network access protocol for encapsulation in a physical frame, which is transmitted onto the network media.

In this way, a simplification of a system network hierarchy can be carried out using Ethernet (TCP/IP); for instance, at the device level in hazardous areas the devices generally uses specialized fieldbus networks. Adding an Ethernet gateway would then make it possible to connect these special subnetworks into a larger Ethernet network, and thus achieving a single network technology. [3, 18, 19, 20]

6 HART DATA AND SYSTEM INTEGRATION

6.1 HART Communication closes the "Information Gap"

The bi-directional communication of the HART protocol fully exploits the potential of intelligent field devices. The protocol extends the 4-20mA standard to enhance communication with smart devices; enabling extensive additional data such as secondary variables, range information, device status and diagnostics (preventive maintenance information).

The inherent intelligence of HART-enabled devices allows them to perform internal diagnostic checks and communicate information regarding their status continuously. Regardless of manufacture, these devices contain 35-40 data items of rich information for improving plant operations and managing assets.

Integration with control systems allows both the analog and digital communication signals to be used for multi-variable process data and real-time detection to any problems impacting the device or the integrity of the 4-20mA current loop. Standard HART commands provide the access to real-time data in HART devices with device status information being part of every response packet from the device. Furthermore, up to 136 device-specific diagnostic parameters can be accessed with a single HART command. [21]

6.2 Integration of HART Data

There are several strategies to integrate HART data and leverage the intelligence in smart field devices, these include [2, 3]:

- **Point-to-Point** – The most common use of HART technology. It allows HART-enabled devices to be configured and set-up for specific applications.
- **HART-to-Analog** – Integrate HART data with an existing analog control system by converting digital data into 4-20mA signals. For this purpose signal extractors are used.
- **HART-plus-Analog** – Keeps analog control but provides better device access. HART multiplexer can replace existing I/O termination panels, making it easy to communicate with HART devices. The analog control signal continues on to the control system, while the HART data is lifted and sent to a device /asset management system providing diagnostics information 24/7.

- **Full HART integration** – Provides continuous communication between field device and control system. This requires HART-enabled Field or Remote I/O system. This solution provides the opportunity to move from a schedule-based maintenance strategy to a predictive-based maintenance strategy.
- **HART-to-Plant-Network** – This strategy passes HART data into plant Ethernet networks by using a HART OPC Server software tool. The data can then go to OPC-compliant applications anywhere in the plant.

The implementation of these data integration capabilities can save time and money, provide valuable diagnostic information, and allow field device data to be used to improve the efficiency of plant operations and manage plant assets.

Management of HART devices can be extensively automated, using the capabilities of the device to monitor device operational values, as well as for remote re-ranging and calibration. Some HART field devices store historical information in the form of trend logs and statistical calculations (e.g., high and low values and averages). These data can be uploaded into a software application for further processing or record keeping. Particularly in complex devices requiring regular maintenance, instrument management using HART can significantly reduce maintenance costs. [1, 2, 5]

6.3 Level of Integration

There are typically three levels the connection can be done to the control or asset management system for communicating with HART field devices: I/O level, Bus level and System Interface and Data Level.

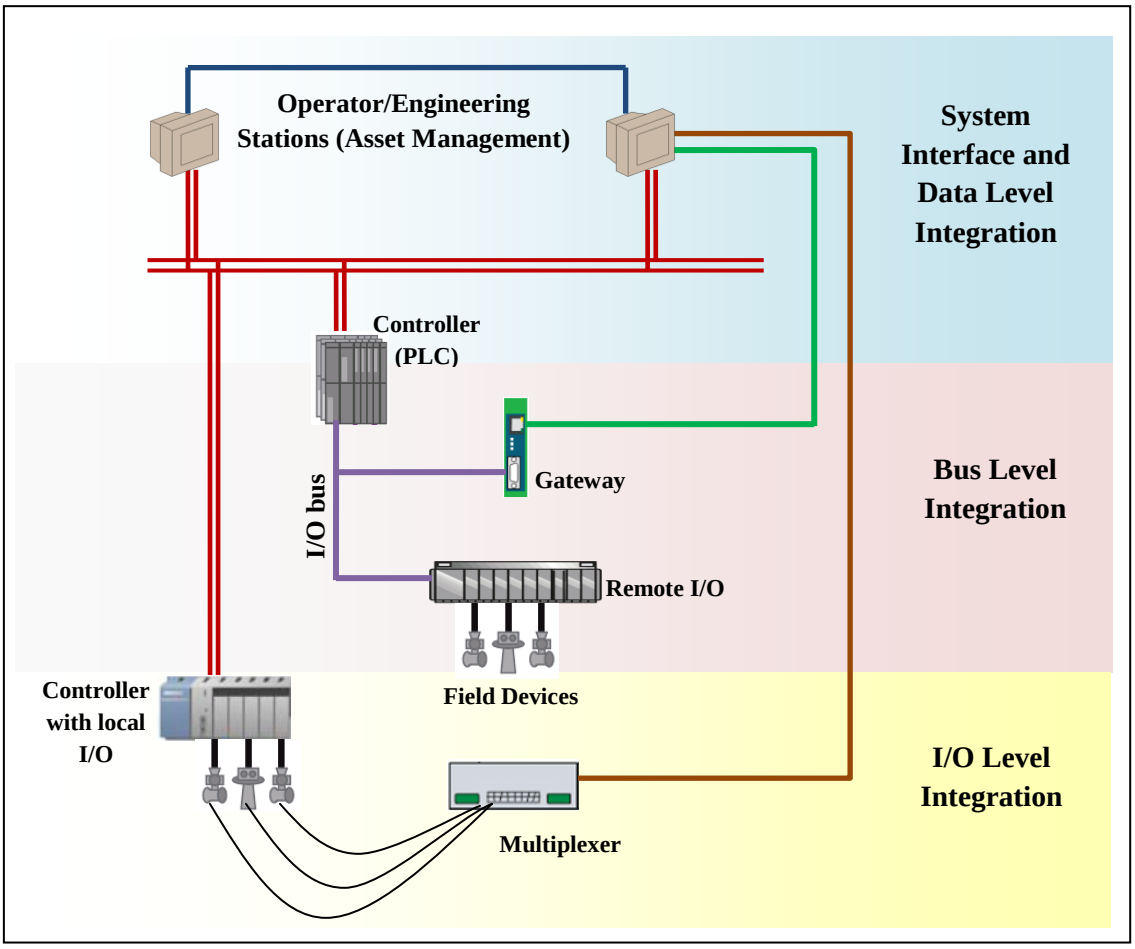


Figure 16 Levels of Integration

6.3.1 I/O level Integration

6.3.1.1 HART bridging devices

The HART protocol supports access to field instruments though ***bridging devices*** such as multiplexers and I/O systems, providing commands to identify and communicate with field equipment beyond them.

A multiplexer or I/O system is addressable with its own Unique ID. Commands #74 and #75 are defined to allow a host device to determine the structure of up to three levels of hierarchy: I/O card number, input channel number and device polling address; and to identify each sub-device. [1]

Many HART-compatible I/O subsystems have multiple analog channels on each I/O card. Suppliers choose whether to provide one HART interface per channel or to share one HART interface among several channels. The numbers of shared channels per HART interface impacts the frequency of data updates from a HART field device and the HART functionality that is supported. For the best performance and flexibility, one HART interface should be dedicated to each I/O channel. [2]

A programmable logic controller (PLC) with local – analog/digital - I/O card can also be used to communicate with HART devices, sending messages with commands and collecting response data from the field device through the I/O card. In HART communication, the analog I/O card acts as a HART master, while the field device as a slave.

The analog measurement signal is usually employed by DCS and PLC hosts for fastest response, while HART communication is used to confirm the status of the field device, or to configure operating parameters such as the analog signal range. Conventional HART-enabled DCS or PLC systems incorporate an internal multiplexer for HART communication, which results in longer scan cycles for those measurements collected in this way. In that case, they may well use HART only to read auxiliary measurements better suited to slower scanning. However, the response time of analog input cards depends in general on the scan rate of the bus. [1]

HART-compatible Multiplexers are ideal when it is desired to interface with a large number of field devices, connecting a number of HART loops to a single host-side connection (some HART multiplexers support up to 32 devices; others can support up to 256 devices). In that way, a multiplexer act as both master (to the individual HART loops) and slave (to the host-side connection).

Multiplexers can be modular and are capable of supporting both point-to-point and all-digital (multidrop) HART communication modes. Usually, a speed and/or protocol conversion is included allowing faster communication from the host. Communication between a multiplexer and a host application depends on the multiplexer capabilities (e.g. RS485, TCP/IP Ethernet). [1, 2]

The scanning strategy implemented in the multiplexer determines the quickness of the response from a field device to the ultimate host. In general, the throughput of a multiplexer

depends critically on its design: whether multidropping is used, how many channels share a HART modem, and what local scanning and buffer storage are provided. [1]

Modular process instrumentation makes use of **Remote I/O modules**, which are signal conditioning components. They make use of digital transmission to interface the plant level of sensors with process control system (eventually monitoring/maintenance system), acting as an interface between signals from hazardous areas (Ex areas) and safe areas (non-Ex areas) via a range of standard busses, e.g. HART, Profibus DP.

Some types of remote I/Os provide built-in HART communication feature, enabling communication with HART-capable field instruments. This has the advantage that parameter data and diagnosis requests can be generated in the control or monitoring/maintenance system and converted into HART commands for the field instrument via the remote I/O. Furthermore, remote I/O systems provide:

- ♣ Simple sensor wiring
- ♣ Simple fault diagnosis in the field
- ♣ Only one communication cable to the remote I/O (e.g. HART, Profibus DP)

Hence, the remote I/Os function as a central access point with which it is possible to address all the HART devices from the process control system using embedded communication technologies. [22, 23]

6.3.1.2 HART I/O for Multidrop support

The analog 4-20 mA signal is no longer required in applications where the measured variables are read by digital communication. This enables the possibility to connect multiple field devices in parallel to a single pair of wired and to communicate with each one in turn to read its measurement or other data. [11]

As stated previously, one HART interface should be dedicated to each I/O channel in order to reach the best performance and flexibility. However, systems that share only one HART interface among several I/O channels may not support multidrop networks, due to the effective update rate of a multiplexed interface is slow enough that the performance of multiplexed multidrop networks would not be practical. [2]

6.3.1.3 HART I/O for Burst mode support

Burst mode is an optional implementation in a field device. Receiving burst mode messages is optional in a host as well. Burst mode is used in order to increase the sampling rate. The field device delivers an updated measured value three or four times per second without needing to be polled. However, this data transfer method is only worthwhile in point-to-point topologies. [2, 21]

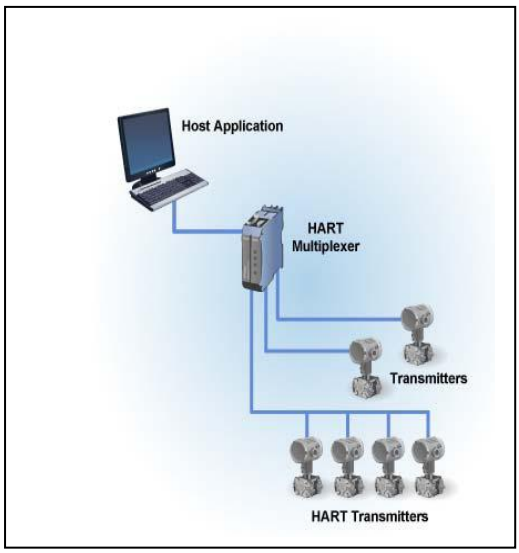
Again, in order to take full advantage of burst mode, the I/O system should have one HART interface for each channel. If the HART interface is shared by more than one channel, messages sent by the field device may not be detected by e.g. the control system. [2]

6.3.2 System integration of HART-compatible multiplexers

HART multiplexers may provide only input to the control or monitoring system, or it may be employed to provide HART Communication to an instrument management system in parallel with the continued use of conventional analog signals to and from a control system. [1]

6.3.2.1 HART Multiplexer as the primary I/O system

In this case, HART multiplexers can be used as the primary I/O front end for a HART-based control or monitoring system (fig. 17). Usually, a PC acts as the host, providing the human-machine interface (HMI) and performing other high-level functions through HART interface applications. The HART multiplexer is used to "strip off" the digital HART message. It continuously monitors the field devices, reports the current readings and instrument status to the host, and passes HART commands from the host computer to the field devices. [2]



As an I/O system, the multiplexer can include IS barriers and other filtering capabilities and provide services to the field device, such as galvanic isolation or power. For this type of installation, no additional terminations or space are required, leading to considerable savings in wiring cost.

Figure 17 HART Multiplexer as the Primary I/O System [2]

6.3.2.2 Parallel monitoring with a HART Multiplexer

This second case retains the high speed of the analog signals for measurement and control, while allowing full use of the capabilities of a HART field device. This is done by adding a HART multiplexer to the network in order to gain access to the digital HART signal.

Hence, the use of a multiplexer enables a supervisory computer to monitor diagnostics and device status, access configuration information, and read any additional process inputs or calculations not provided by the 4-20mA signal. [1, 2]

A multiplexer wired in parallel with the field wiring is commonly used when the control system wiring is already in place. This approach is ideal for retrofit applications, where it is desired to take advantage of the HART capabilities of existing instrumentation, without disturbing a legacy control system.

The multiplexer can also act as a gateway to convert the HART messages to a higher-speed field protocol such as PROFIBUS, or Ethernet. [2]

6.4 Bus Level Integration

6.4.1 Higher-level communication systems

The HART communication protocol provides a combination of field-tested analog measured-value transmission and simultaneous digital communication with bi-directional, acyclic transmission, making it possible to transfer diagnostic, maintenance and process information from field devices to higher-level systems. To be able to connect a HART communication system with other communication systems, gateways with standard communication protocols are used. They convert the respective protocols of the networks to be coupled. In most cases, when complex communication tasks must be solved, fieldbus systems (IEC 61158) would be preferred above HART. [4]

6.4.2 HART Bus Communications

6.4.2.1 HART over Profibus

The Profibus DP protocol, that already offered a high-speed communications channel for cyclic data exchange between the control system and the sensors in the field, was later extended to manage asynchronous information fitted into the time gaps between synchronous telegrams.

Thereupon the extension, called Profibus DPV1, provides asynchronous services that allow the exchange of parameter and configuration data between DP master (class 1, 2) and DP slave. Furthermore, the extended protocol makes it possible to transmit HART telegrams via the system bus.

In this way, whereas synchronous telegrams belonging to one participant are always of the same structure and length, the data exchange of the DPV1 is characterized by a fixed buffer area into which data telegrams that are important for the parameterization and configuring can be inserted, when required.

The entirely compatibility of the Profibus DPV1 and Profibus DP ensures that DPV1 field devices function with an existing Profibus DP as well as an existing slave functions with a DPV1 master. Some control systems or PLCs features Profibus DPV1 master cards, not others. When HART is not supported from the DCS or PLC operator station, a Profibus master class 2 can be employed, in addition to software packages to offer HART communications. [22, 24]

6.4.2.2 Profibus application profiles

The various bus nodes (field devices, controllers, engineering stations, and operator control and monitoring stations) of different manufacturers that are linked in a system, for communication purposes, via bus protocols must be consistent in terms of their basic communication functions and services.

This is achieved through the use of Profibus application profiles which are specified for standard data exchange between field devices on the user level, guaranteeing interoperability in the data exchange between field devices from the diverse manufacturers. [12]

6.4.2.3 Integration profile - HART on Profibus

Endorsed by both the HART Communication Foundation and the Profibus User Organization, the HART on Profibus profile defines the integration of HART field devices in Profibus systems.

The mapping of the client-master server model of HART on Profibus is enabled through the implementation of the Profibus profile in the master and slave above the application layer (layer 7).

This specification standardizes the communication between client applications in HART devices via Profibus DP-V1. The HART client application is integrated in a Profibus master and the HART master in a Profibus slave, as depicted on figure 18, through which the latter serves as multiplexer and takes over communication with the HART devices.

A communication channel which works independently of the MS1 and MS2 connections has been defined for the transmission of HART messages. To use this communication channel each client must request a handle from the HART master device (HMD). The communication objects are then addressed via slot numbers and indexes. [25]

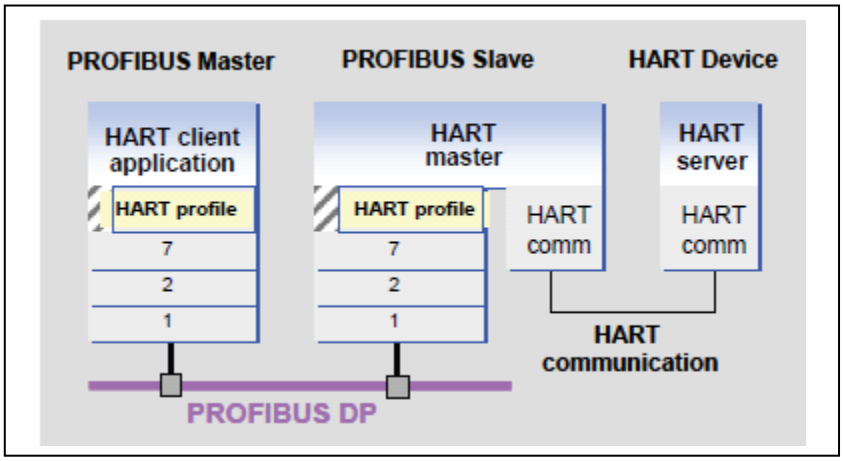


Figure 18 HART on Profibus mapping [12]

The communication between HART client and HART master device is done via acyclic write and read services of Profibus DP-V1. The Profibus profile allows a HART master device to support multiple clients. At the same time several HART clients can access the same channel on one HMD.

The Profibus DP profile for HART provides capabilities for HART-enabled I/O systems. Basically, the I/O is only required to “pass through” HART messages and the I/O does not need to utilize any HART Application Layer capabilities. In that way, the I/O does not know the difference between Universal, Common-Practice, Device-specific or Device family commands.

Profibus restricts the maximum length for MS1 and MS2 data field (w/o slot and index) to 240 bytes, while the HART protocol limits the data of a HART message to 255 bytes. However, if the HART header (9/11 bytes) and the profile control bytes (2 bytes) are included, the HART message size can be 268 bytes in total and does not include the preamble required by the HART FSK Physical Layer.

The HART telegrams are incorporated into the data field of Profibus DP, limiting the max. Length of the HART PDU and the profile commands to be 230 bytes or less. In the event a HART command response is longer than 230 byte the HMD will send back a negative response. Nevertheless, the minimum length for the HART response buffer must be 45 bytes to be able to transfer HART Universal Commands.

Moreover, the HART on Profibus profile standardizes the nomenclature used with HART-capable Profibus DP slaves, their configuration, and operation. HART field devices can be connected to HART master devices to Profibus via modules, e.g. communication gateways, employing the GSD of the component. These are typically implemented as a DPM2 and as a HART master device.

Additionally, the Profibus profile can offer detailed configuration of complex HART devices using an EDD or the FDT/DTM technology. [12, 25, 26]

6.4.2.4 Throughput and latency of a point-to-point connection

Profibus DP and HART protocol requirements are the main factors which affect latency and throughput in such a system. Assuming that the Profibus connection is fast enough to meet the HART Data Link Layer requirements, throughput and latency can be calculated (using Universal Command #1) based on the following table [26]:

Master request	128.3 ms
Slave response	174.2 ms
Link Grant Time	75 ms
Total Time	377.5 ms
Transactions per second	2.65 tps

Table 9 Performance of a point-to-point connection

The actual throughput can be somewhat less. E.g., the slave device may not begin its response immediately and some HART commands are longer.

In this case, it should also be noted that the main purpose of Profibus DP is the fast transmission of cyclic process data. Consequently, there is only a small window for MS1 and MS2 services in each cycle, which at the end affects the latency and throughput. Besides, communication through PC-based clients can have delays of 20 ms to more than 100 ms, giving a significant impact in the system performance.

The HART communication is considered “unsynchronized” if Profibus cyclic communication or latency through the PC client delays the transmission of the request to the HMD; given that the HART Data Link Layer specifies that a master must begin its request within 20ms. In that event, the master must wait 2 times the Link Quiet Time (RT1) before HART communication can continue. This results in as much as 3 times slower communication. [25, 26]

Master request	128.3 ms
Slave response	174.2 ms
Link Grant Time	75 ms
2 Link Quiet	760 ms
Total Time	1137.5 ms
Transactions per second	0.88 tps

Table 10 Affect of missing the hold time window

6.4.3 HART over Ethernet

The nowadays trend in process control instrumentation is increasingly toward using Industrial Ethernet as the networking bus to integrate field instruments into the DCS and PLC environments. Ethernet technology provides a standard well known networking protocol, and allows easy and standardized connection to input and output signals. Furthermore, Industrial Ethernet offers technical improvements over legacy I/O instrumentation and considerable savings allowing wider area networks beyond traditional cable lengths.

Due to HART protocol is based on serial and 4-20 mA technologies, HART devices cannot take advantage of the Ethernet architecture without a gateway device. Similar to the Profibus class 2 master principle a stand-alone HART master can be connected to the Ethernet network. [22]

6.5 System Interface and Data Level Integration

There are several ways to enable the communication interface between field devices and systems. Standard technologies such as EDD, FDT and OPC provide environments to access field instrument's data independent from the communication protocol and the software environment of either the device or the host system.

6.5.1 HART Device Integration

6.5.1.1 Device Description Language (DDL) and Device Descriptions (DDs)

The definition of DDL is included in the HART protocol specifications, being this protocol the first one to implement Electronic Device Description Language (EDDL) as its standard. EDDL (IEC 61804-2) is the only technology endorsed by the HART Communication Foundation (HCF) for configuration of HART devices.

DDL includes descriptions of accessible variables, the instrument's command set, and operating procedures such as calibration. It also includes a description of a menu structure and displays, which a host device can use for a human operator, forming an additional "user layer" on the top of the OSI protocol reference model.

Device Descriptions (DDs), created by using DDL, provide the information needed by a host application or control system to properly access and display device information located in intelligent field devices. Written in a readable text format, DDs define the data available from the field device and how to read, write and display it.

Instrument manufacturers create a Device Description (DD) describing the capabilities of their device. Host applications, control systems and maintenance systems use the DD to access and interpret the instrument's data.

The DDs allow access to all HART commands: Universal commands, Common Practice commands and Device Specific commands. In that way, the DD aware host systems can use the data and display it as the user desires.

The master device does not read the device description as readable text in DDL syntax, but as short, binary-coded DD data record specially generated by the DDL encoder. For devices with sufficient storage capacity, this short form opens up the possibility to store the device description already in the firmware of the field device. During the parameterization phase it can be read by the corresponding master device. Similar to the presentation of HTML that is

independent of an operating system and browser, DDs eliminate the need for host suppliers to develop and support custom interfaces and drivers. [1, 2, 3, 5]

6.5.1.2 FDT

Field Device Tool (FDT) technology is one of the important creations that have come about in response to the requirements for field device integration in systems aiming to support efficient engineering and maintenance. The FDT concept standardizes the communication interface between field devices and systems, providing independency from the communication protocol and the software environment of either the device or the host system. Hence, FDT allows any device to be accessed from any host through any protocol.

To operate a field device, the user always needs the suitable device driver, which is developed and provided by the corresponding field device manufacturer. The FDT standard defines the interface specifications which are necessary for the creation of these device drivers, also referred to as Device Type Managers (DTM).

Thereupon, FDT is a client/server architecture, where DTMs act as servers for device information and functionality. Such DTM can range from a simple graphical user interface (GUI) for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes. Consequently, FDT programming technology allows the flexibility required with regard to graphical and interactive functions, adding expanded capabilities to DDs/EDDs.

Another element of the FDT technology is the FDT interface. This is the specification that describes the standardized data exchange between field devices and engineering systems (for instance control systems or asset management tools). I.e. all data exchanged between the frame application and the device DTM goes via the common FDT/DTM interface.

The FDT concept consists of a Frame Application, where the various DTMs can be integrated. A FDT frame application is thus the container for DTMs, and allows logical point-to-point communication between the DTMs and the field devices according to the network topology (routing).

In general, DTMs are classified into two categories: Device DTMs and Communication DTMs. Devices DTMs encompass all instrument-specific data and functions for the adjustment of a slave device. This includes all the basic functions such as graphic display functions, interactive adjustment options and simulation and diagnostics options.

While the Device DTMs represents the whole logic and parameters of a device, the Communication DTMs represent communication components such as PC communication cards, gateways, remote I/Os, multiplexers, etc. These DTMs are required for communication with a field device via point-to-point connection or also via any network structure. In this way, the DTMs offer a unified structure for accessing device parameters, configuring and operating the devices, and diagnosis problems.

The FDT Frame Application (host system) is a software program that implements Device DTMs and Communication DTMs. A single FDT frame application supports communication protocols such as HART, Profibus, and others. In this way, it provides functionality to manage data, to communicate with the device (performing network configuration) and to embed DTMs. [27, 28, 29]

An example of communication architecture comprising both a HART and Profibus device is depicted on figure 19.

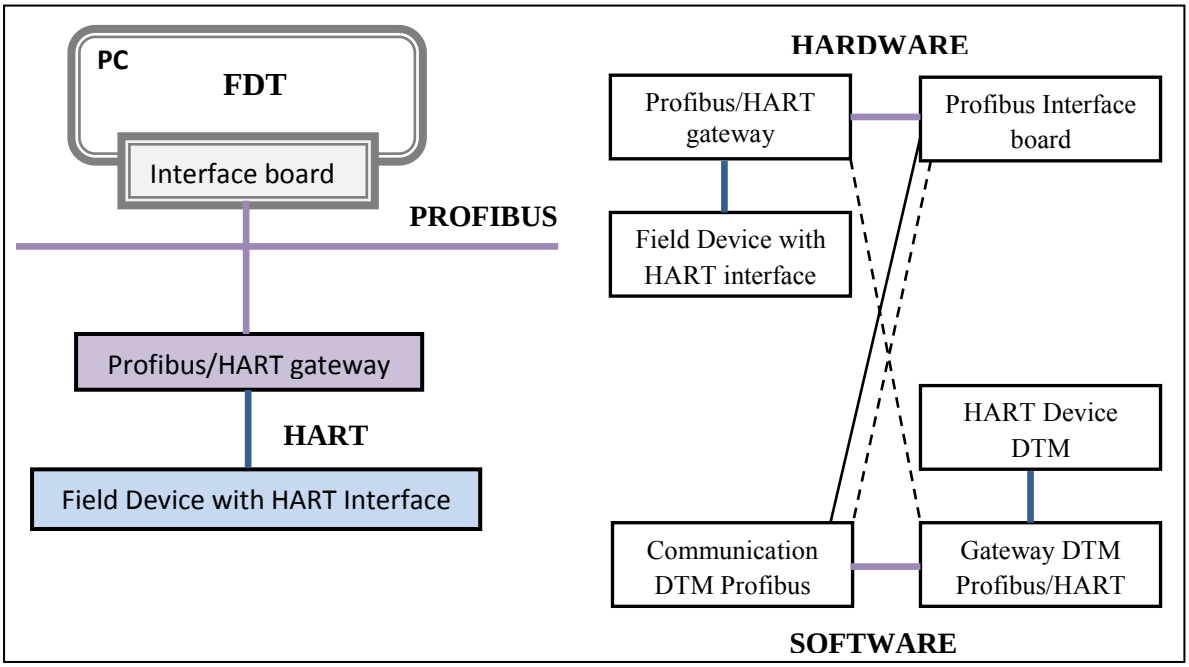


Figure 19 Example of communication architecture and mirror-image hierarchy of DTMs[14,15]

Field devices, communication gateways and communication devices are supported by the FDT specification, having a significant focus on field devices equipped with a communication interface.

[14] The dotted lines indicate a logical assignment between a piece of hardware and a DTM.
[15] The line drawn through the middle represents a physical coupling implemented through operating system mechanisms (outside the scope of the FDT specification).

As shown on figure 19, a HART device is connected to a HART bus system, and a gateway, which acts as a Profibus device on the Profibus bus system and as a HART master on the HART bus system. The Profibus is then connected via a Profibus interface board to a PC running an FDT environment, where the various kinds of DTMs apply: Field Device (Device DTM), Gateway (Gateway DTM), Communication connection device (Communication DTM).

Communication on the hardware side takes place on the basis of the standardized communication protocols such as Profibus (DPV1). Similarly, FDT defines "software" protocols that connect the DTMs to each other in terms of communication technology. [27]

6.5.2 Profibus Device Integration

A remarkable advantage of Profibus technology is the openness for integration of field devices from different manufacturers. Device integration is usually carried out by mapping the various functions of the device to the operator software, and optimized through consistent data management over the life cycle of the system, with equal data structures for all devices. [12]

According to the complexity of the field device, EDDL-GSD and FDT-DTM standards can be used in conjunction with Profibus.

The device **General Station Description (GSD)** provides all required information for cyclical communication with the Profibus master and for the configuration of the Profibus network, so that the master is able to exchange input and output data with a field device.

In form of a text-based description[16], the GSD is sufficient for the cyclic exchange of measured values and manipulated variables between field device and the engineering system. [12, 13]

However, the GSD is not sufficient to describe the application-specific functions and parameters of complex field devices. This requires a language, such as **EDDL** able to carry out parameterization, service, maintenance and diagnostics of devices.

[16] In Profibus, the GSD file is in ASCII format.

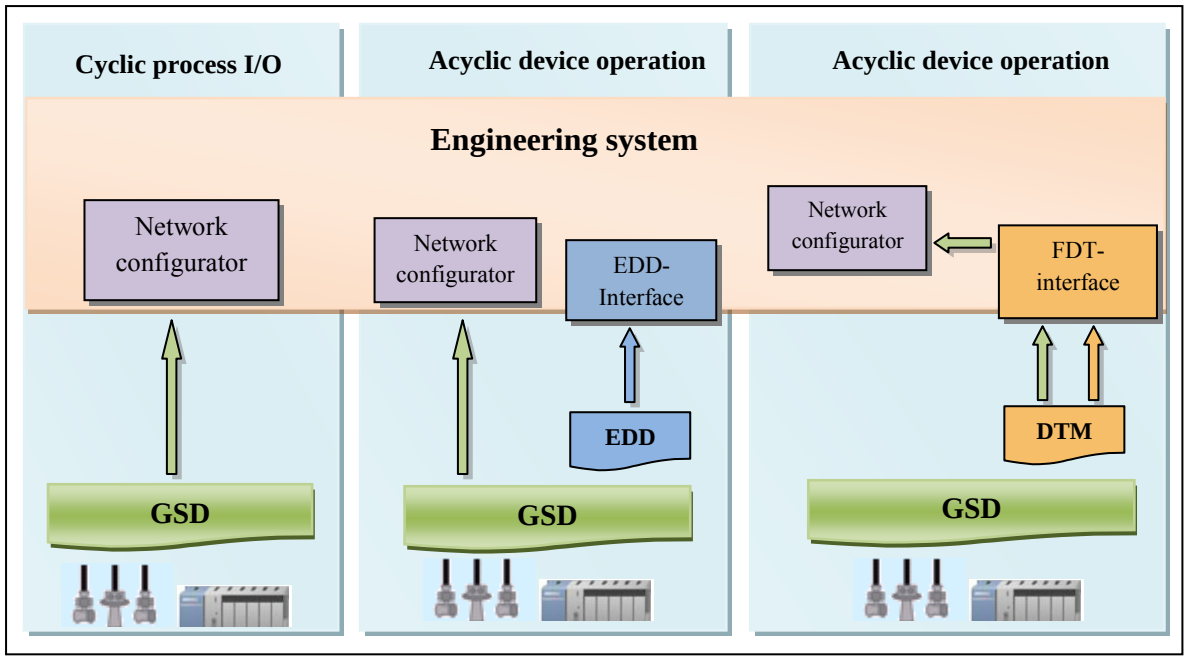


Figure 20 GSD, EDD and FDT/DTM standards

As stated previously, the EDD is a textual device description independent of engineering and control systems that provides the description of the device functions that are communicated acyclically. Developed by the device manufacturer, an EDD is a file used in conjunction with GSD, as depicted on figure 20. It provides the basis for processing and displaying device data on the EDD interpreter which is the open interface between the operator program with data for visualization. [28]

In contrast with GSD and EDD technologies, the **FDT/DTM technology** is a software-based device integration method, and is optimal for highly complex field devices.

The DTM that accompanies the field device is a software component and communicates with the engineering system via the FDT interface[17]. [27]

[17] This is explained in detail in chapter 6.5.1.2.

6.5.3 OPC

OPC is a series of standards specifications for data exchange in industrial automation. In accordance with the different requirements within industrial applications, three major OPC specifications have been developed: OPC Data Access (DA), Alarm & Events (AE), and Historical Data Access (HDA). These are called "OPC Classic", and are based on Microsoft COM and DCOM technology which describes how applications should exchange data on a Microsoft platform.

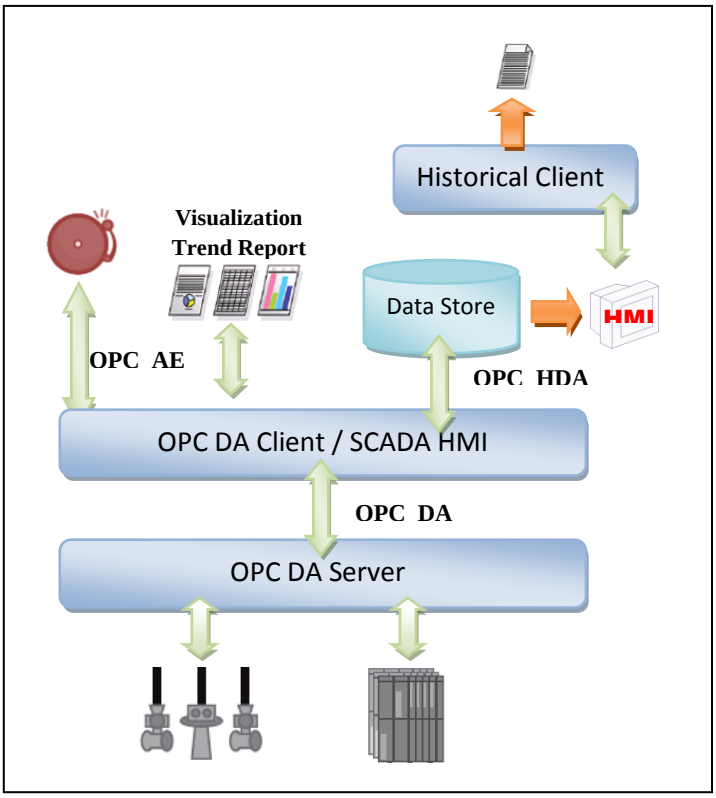


Figure 21 OPC Classic Architecture[18]

OPC applies a client-server approach for the information exchange. An OPC server encapsulates the source of process information like a device and makes the information available through its interface.

Access to the OPC server is accomplished with standardized methods as specified in the OPC standards. This means that an OPC client can connect to the OPC server and access and consume the offered data. Nevertheless, applications consuming and providing data can be both client and server.

[18] Source: www.opcfoundation.org

An OPC server is not required to run on a dedicated sever machine. Furthermore, several OPC servers might run on the same machine, as well as OPC clients and servers can well be running on the same machine.

The OPC standards are maintained by the OPC Foundation. The foundation has defined several software interfaces to standardize the information flow from the process level to the management level. The primary use cases are interfaces for industrial automation applications such as SCADA and HMI systems to consume data from devices and to provide current and historical data and events for management applications. [30, 31]

OPC DA

The OPC Data Access provides the interfaces for data acquisition in support of a vertical architecture[19]. The OPC DA interface enables the functionality for reading, writing, and monitoring of variables containing current process data (from various networked devices).

The main application case is to transfer real-time data from PLCs, DCSs, and other control devices to HMIs and other display clients.

An OPC DA server contains a flat or structured list of OPC items, which usually correspond to tags or I/O points. The OPC DA clients specifically select the variables (OPC items) they want to read, write, or monitor in the server. Connection to the server is established by creating and OPCServer object. The server object allows OPC clients to browse the OPC server and find the available items and their properties like data type and access rights.

In order to access the data, the OPC client creates an OPCGroup object in the server. Such objects contain OPC items with identical settings. Once they are added to a group, the items can be read or written by the client. [30, 31]

Different ways of subscribing to OPC items are available, e.g., asynchronous or cyclical. Yet, the preferred way for the cyclic reading of data by the client is monitoring the value changes in the server.

OPC supplies real-time data that may not always be accessible. This is e.g. in the event the communication to a device gets temporarily interrupted. The Classic OPC technology deals with this issue by providing OPC items a timestamp and a flag for data quality for the delivered data. [30, 31]

[19] Serve data from a device to a client application on a higher level computer.

OPC AE

The OPC Alarms & Events interface enables the reception of alarm and event notifications, which can be considered to be another type of data since they are a valuable component of the information architecture outlined in the OPC DA specification.

Whereas event notifications inform the client about the occurrence of an event, changes of a condition in the process are notified to the client by means of alarm notifications. [30, 31]

For the purpose of receiving notifications, the OPC AE client connects to the server, subscribes for notifications, and then receives all notifications triggered in the server. Similar to the OPC DA, the OPC client establishes connection by first creating an OPCEventServer object in the AE server and then by generating an OPCEventSubscription it receives the event messages.

On the other hand, there is no explicit request for specific information. All process events are supplied and can be limited by the client setting a certain filter criteria. [31]

OPC HDA

In contrast to OPC DA that give access to real-time, continually changing data, the OPC Historical Data Access specification enables access to historical data. Additionally, the OPC HDA provides methods for inserting, replacing, and deleting data in the history database.

Reading of historical data is carried out in three different ways. The first mechanism reads raw data from the historical archive, where the client defines one or more variables and the time domain it wants to read from the server. The second mechanism reads values of defined variables for specified timestamps. The third one calculates values from data in the history database for the specified time domain for the defined variables. [30, 31]

7 TEST FIELD EQUIPMENT

This section provides a brief but comprehensive description of the components that are needed and going to be part of the general test setup system (see chapter 8). This includes four HART field devices, one PLC, five Remote I/O modules and transmission cables.

7.1 HART Field Devices

HART was originally developed for use with measurement devices ("transmitters"). HART field devices are widely used in industrial automation for purposes such as measuring of process variables like flow, level, temperature and pressure. HART technology-equipped field devices provide simultaneous analog and digital communications capability, achieving best resolution and accuracy of the measured data.

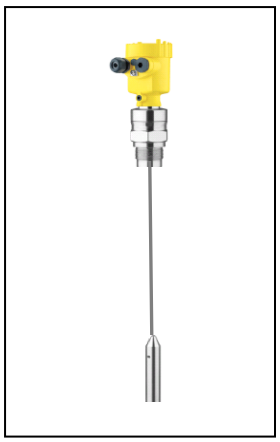
Furthermore, the built-in HART technology enables valuable status and diagnostic information from the field devices, as well as remote configuration. The latest gives great user benefits, making it possible to access to the intelligent HART devices from a safe, remote location, eliminating the need for routine visits to hazardous areas, for instance. This is then performed via DCS or Asset Management systems.

A wide range of process measurements are available from many different manufacturers, e.g. VEGA Grieshaber KG which is a leading supplier of level and pressure instrumentation.

Vega measurement technology covers a broad scope of applications. In the following section three types of sensors will be briefly described.

7.1.1 VEGA sensors

7.1.1.1 Vegawell 52



Vegawell 52 is used for level and gauge measurement in deep wells, ballast tanks as well as atmospherically open vessels. The sensor element of the sensor is the CERTEC measuring cell with rugged ceramic diaphragm. The hydrostatic pressure causes a capacitance change in the measuring cell via the ceramic diaphragm. This change is converted into an appropriate output signal (4-20 mA). Additionally, a resistance thermometer (PT100) is integrated in the transducer. [32]

Figure 22 Vegawell 52

Technical data

Voltage supply	Two-wire electronics 4-20 mA/HART for power supply and measured value transmission over the same cable.
Input variable	Measured variable : Level
Output variable	Output signal 4-20mA/HART HART Output variables: -Primary: pressure -Secondary: temperature Step response time: min 200 ms
Failure signal current output (adjustable)	mA-value unchanged 20.5 mA, >22 mA, < 3.6 mA
Max. output current	22 mA
Measuring range	0 – 2.5 bar -50...+100 °C
Measuring accuracy	± 3 mm

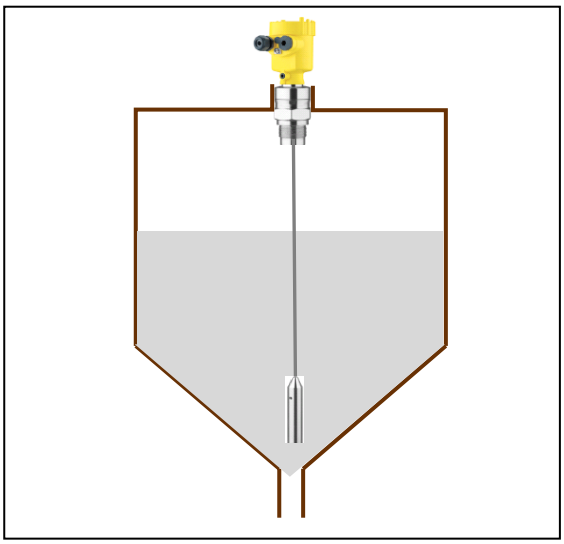
Table 11 Vegawell 52 technical information

Sensor available parameters

Parameter (over 4-20 mA HART)	
Sensor Tag	Linearisation curve
Serial number	Scaling
Version number	Echo curve
Date of manufacture	Damping
Range	Sensor Status
Failure Mode	Sensor Type
Min/Max current	Sensor Element Type/Probe length
Process Temperature	SIL qualified sensor
Process fitting/material	HART Device Status
Message (Failure, “Maintenance requirement” or “Out of Specification”)	Min/Max measured value
Unit of measurement	Begin/End of measurement range
Measured value presentation	Pressure Type
Medium	Device Address
Sensitivity	

Table 12 Summary of the available parameter over HART

7.1.1.2 Vegaflex 61



Vegaflex 61 is a level sensor for continuous level measurement. It has been designed for industrial use in all areas of process technology, particularly suitable for liquids and heavy solids**.**

The sensor's measuring principle is high frequency microwave pulses which are coupled onto a cable or rod and guided along the probe. Upon reaching the product surface, the microwave pulses are reflected. The running time is evaluated by the instrument and outputted as distance. [33]

Figure 23 Vegaflex 61 sensor in a vessel with conical bottom.

Technical data

Voltage supply	Two-wire electronics 4-20 mA/HART for power supply and measured value transmission over the same cable.
Input variable	Measured variable: Level of liquids and solids
Output variable	Output signal 4-20mA/HART HART Output variables: -Primary: distance to level -Secondary: distance to level - scaled Cycle time: min. 1 second (depending on the parameter setting)
Failure signal current output (adjustable)	mA-value unchanged 20.5 mA, >22 mA, < 3.6 mA
Max. output current	22 mA
Measuring range	Rod version: up to 6 meters Cable version: Up to 60 meters
Measuring accuracy	± 3 mm

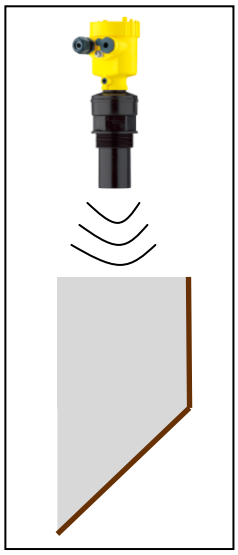
Table 13 Vegaflex 61 technical information

Sensor available parameters

Parameter (over 4-20 mA HART)	
Sensor Tag	Sensitivity
Serial number	Linearisation curve
Version number	Scaling
Date of manufacture	Echo curve
Range	Damping
Calibration date	Sensor Status
Failure Mode	Sensor Type
Min/Max current	Sensor Element Type/Probe length
Process Temperature	SIL qualified sensor
Process fitting/material	HART Device Status
Message (Failure, "Maintenance requirement" or "Out of Specification")	Min/Max measured value
Unit of measurement	Begin/End of measurement range
Measured value presentation	Pressure Type
Medium	Device Address

Table 14 Summary of the available parameter over HART

7.1.1.3 Vegason 61



Vegason 61 is an ultrasonic sensor for continuous level measurement in liquids and for use in small bulk solids vessels, where short ultrasonic pulses are emitted by the transducer in the direction of the measured product.

These pulses are reflected by its surface and received back by the transducer in echoes. The elapsed time from emission to reception of the signals is proportional to the distance and hence the level. The determined level is converted into an appropriate signal and outputted as measured value. [34]

Figure 24 Vegason 61

Technical data

Voltage supply	Two-wire electronics 4-20 mA/HART for power supply and measured value transmission over the same cable.
Input variable	Measured value: distance between lower edge of the transducer and product surface
Ultrasonic frequency	70 kHz
Output variable	Output signal 4-20mA/HART, HART Output variables: -Primary: distance to the level -Secondary: temperature -Third: distance to the level - scaled Response time: > 3 seconds
Failure signal current output (adjustable)	mA-value unchanged 20.5 mA, >22 mA, < 3.6 mA
Max. output current	22 mA
Measuring range	In liquids: 0.25 - 5 meters In bulk solids: 0.25 - 2 meters
Measuring accuracy	± 10 mm

Table 15 Vegason 61 technical information

Sensor available parameters

Parameter (over 4-20 mA HART)	
Sensor Tag	Medium
Serial number	Linearisation curve
Version number	Scaling
Date of manufacture	Echo curve
Range	Damping
Calibration date	Sensor Status
Failure Mode	Sensor Element Type/Probe length
Min/Max current	SIL qualified sensor
Process Temperature	HART Device Status
Process fitting/material	Min/Max measured value
Message (Failure, “Maintenance requirement” or “Out of Specification”)	Begin/End of measurement range
Unit of measurement	Pressure Type
Measured value presentation	Device Address

Table 16 Summary of the available parameter over HART

7.1.2 Remote I/O modules

The remote I/O modules - I/O stations, Com Units, Power Supplies - act as an interface between signals from the hazardous area (Ex area) and the safe area (non-Ex area). The interface technology provides several device functions for evaluating and transferring sensor signals. Its primary task is to isolate, transform and amplify signals between the field circuit and the control circuit (DCS/Asset Management). [21, 23]

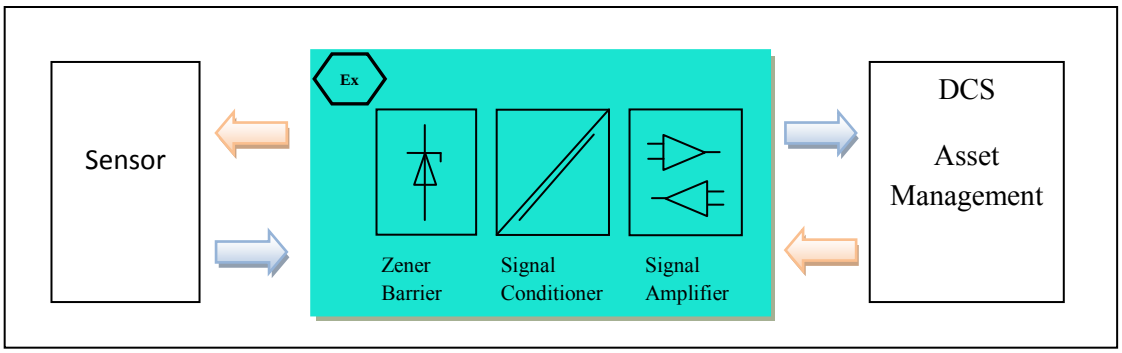


Figure 25 Interface technology - isolating, transforming and amplifying signals

Pepperl+Fuchs (P+F) GmbH is a leading developer and manufacturer of electronic components for the automation market. A variety of Remote I/O modules are available for applications associated with potentially explosive atmospheres (Ex areas). The succeeding section gives a description of the Remote I/O modules (and their features) that will be employed later in the test setup (see chapter 8).

7.1.2.1 LB 3102 HART analog input/transmitter power supply

Process instrumentation makes use of digital data transmission to interface the plant level of sensors with process control systems. The integration approach is to make use of built-in HART communication feature of Remote I/O stations, ensuring a high degree of measurement accuracy through the digital transmission and offering extensive possibilities for the remote control of connected field devices employing HART communications.

The LB 3102 HART analog input is the interface for the process signals from the pressure and differential pressure transmitters, level transmitters, externally supplied devices and flow/fluid level transducers to pass to the process control or asset management system. [22, 24]

Features

- 1 channel for the connection of 2-wire HART transmitters and 4-20 mA signals.
- LB Remote I/Os bring intrinsically safe inputs and outputs from sensors to all kinds of busses

- Power supply for 2-wire transmitters (HART) - Input isolator for separately powered devices, enabling also HART communication (via fieldbus) - The intrinsically safe input is galvanically isolated from the fieldbus and the power supply - Step response time: min 30 ms, max 140 ms	
Input	
Field device power supply	16.5 V (20mA) incl. 250 Ω
Input range	4 – 20 mA (0 – 26 mA) HART
Conversion time	≤50 ms
Internal bus	
Connection	backplane bus
Interface	manufacturer specific bus to standard ComUnit/Gateway

Table 17 LB3102 features

The analog input 3102 can be operated with max. 4 HART auxiliary variables, each occupies 4 bytes. I.e. the I/O module use up 18 bytes in the cyclical data traffic, including 2 "Standard" data bytes and all 4 HART variables, which are updated less frequently than the standard process data.

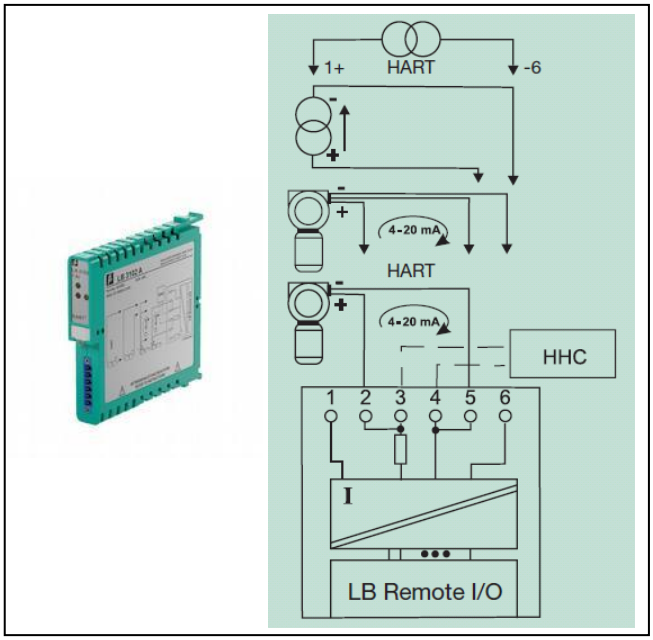


Figure 26 LB3102 connection diagram

Resolution

Input signals within a range of 4-20 mA are detected with a resolution of 12 bits (digital representation). The actual measurement range is calculated based on this resolution, thus a resolution of 2500 measurement points is obtained. [17]

LB Remote I/Os are generally mounted on a backplane that snaps onto a standard DIN[20] rail. The backplanes provide power to the modules and the internal wiring between the bus communication interface (Com Units or gateways) and the I/O devices.

[20] A DIN rail is a metal rail of a standard type widely used for mounting and industrial control equipment inside equipment racks.

7.1.2.2 Profibus Com Unit - Easycom LB 8106

The communication gateways or Com Units connect Remote I/O stations to process control systems (DCS, PLC), SCADA and Asset Management systems. Pepperl+Fuchs Com Units convert the protocol of the system bus integrated in the backplane into the protocol of the higher-level bus system. Thus, the scope of application of Remote I/O systems is mainly determined by the fieldbus system. [24]

Features

- Configuration via GSD parameters from the control system - Communication via Profibus DP - The Com Unit links intrinsically safe inputs and outputs from sensors to the Profibus - HART Communication via Profibus DPV1 - Supports 1-8 channel I/O modules	
Technical data	
Supply	Connection via backplane bus
Internal bus	Backplane bus
Supported I/O modules	All LB Remote I/O stations
Fieldbus interface	
Physical properties	As per RS485 standard Connection: 9-pin Sub-D socket via backplane
Transmitting medium	Twisted pair cable
Topology	Line structure
Interface profile	RS485
Bus access protocol	Profibus DP standard (cyclic), or DPV1 standard (acyclic) – read/write services
Baud rate	Up to 1.5 Mbit/s
Number of stations per bus line	≤125 (Profibus)
Number of channels per station	≤80 analog, ≤184 binary (digital)
Number of station per bus segment	≤31
Addressing	Via configuration software
Profibus address	0 – 126 (ex works standard: 126)
Node communication	according to the master/slave principle
HART Communication	Via Profibus

Table 18 Com Unit LB8106 features

The internal system bus integrated in the backplane is a proprietary bus, which scans cyclically all I/O stations (6.5 ms for a complete cycle), as depicted in figure 27.

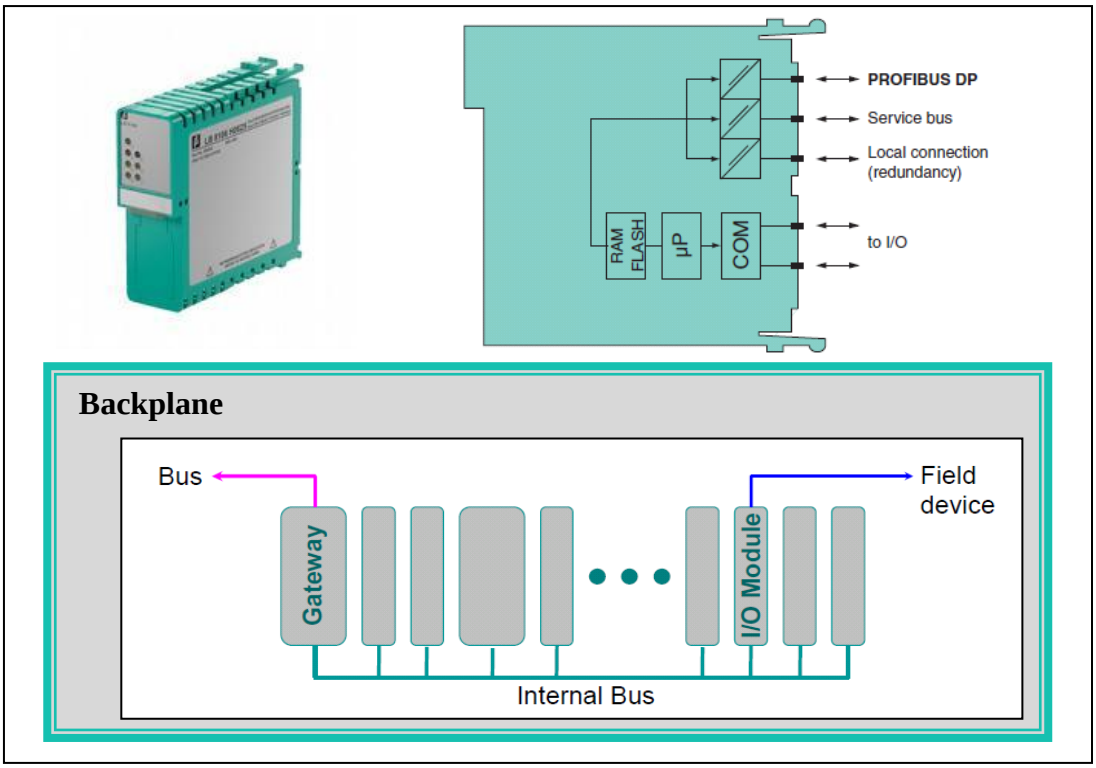


Figure 27 LB8106 connection and modular configuration diagram

Data transfer and general configuration

Data is transferred as specified in the Profibus protocol specification, thus it is important that both master and slave have the same configuration prior to data exchange. The process data for each channel is transferred in whole numbers without a prefix within a range from 0 – 65535. A 16-bit word is available for transferring data to each channel.

The Easycom LB8106 remote I/O is configured and parameterized exclusively via the Profibus. P+F provides a description file (GSD) with the definition of the configurable I/O components and their parameters, in order to integrate the LB remote I/O.

Once the Profibus master has imported the GSD file, the configuration and parameterization can then be executed using the parameterizing tool of the master (PLC). The configuration and all setting parameters are thus stored in a joint data base of the engineering system and can be stored in the gateway of the bus station by downloading them via the Profibus DP-V1.

The P+F Com Unit is also used to be able to connect HART communication systems. Using the Profibus DPV1 the HART properties and diagnostic functions can be accessed via the PLC, or a secondary master. However, DPV1 services may not be used to configure remote I/O stations so as to prevent conflicts between the master and slave components. [17]

Measuring and cycle time LB 3102 and LB8106 Remote I/O

The immediacy of the measured value depends on the cycle time of the data traffic in the Profibus, whereby the fastest Profibus (1.5 Mbit/s) normally addresses the connected slaves about once every 20ms.

Pepperl+Fuchs use a 20 ms conversion time, which takes place in each module independent form other scan cycles. The signals are then transmitted from the I/O station to the Com Unit every 6.5 ms, irrespective of the measuring time. [17]

7.1.3 Programmable Logic controllers (PLC)

The PLCs, also referred to as programmable controllers are an essential component of industrial applications due to they act as controllers for machines and processes through input-monitoring, decision-making and output-controlling. In that way, input modules, CPUs, and output modules are the elementary parts of a PLC.

The input modules accept a variety of analog and digital signals and convert them to logic signals that can be used by the CPU (PLCs use I/O modules to receive inputs from binary devices such as sensors). The CPU makes decisions and executes control instructions based on the program contained in its memory. To complete the cycle, the output modules convert the control instructions from the CPU into signals that can be used to control manifold field devices.

Siemens manufactures several PLC product lines in the SIMATIC S7 family. This includes the S7-300 modular controller series, which are used for complex computing and communication functions. A S7-300 PLC consists of a central unit (CU) and of one or multiple expansion modules. The rack containing the CPU is the central unit (CU). Racks equipped with modules and connected to the CU form the expansion modules (EMs) of the system. A concise description of the Siemens components used in the test setup system is given in the following sections. [36, 37]

Siemens S7, CPU 317-2 DP V2.6 (Component number 6S7317-2AJ10-0AB0)

The Siemens S7 CPU is a microprocessor system and it is the PLC decision-making unit. Based on the control instructions held in its program memory, it performs relay, counting, timing, data comparison, and sequential operations.

Features

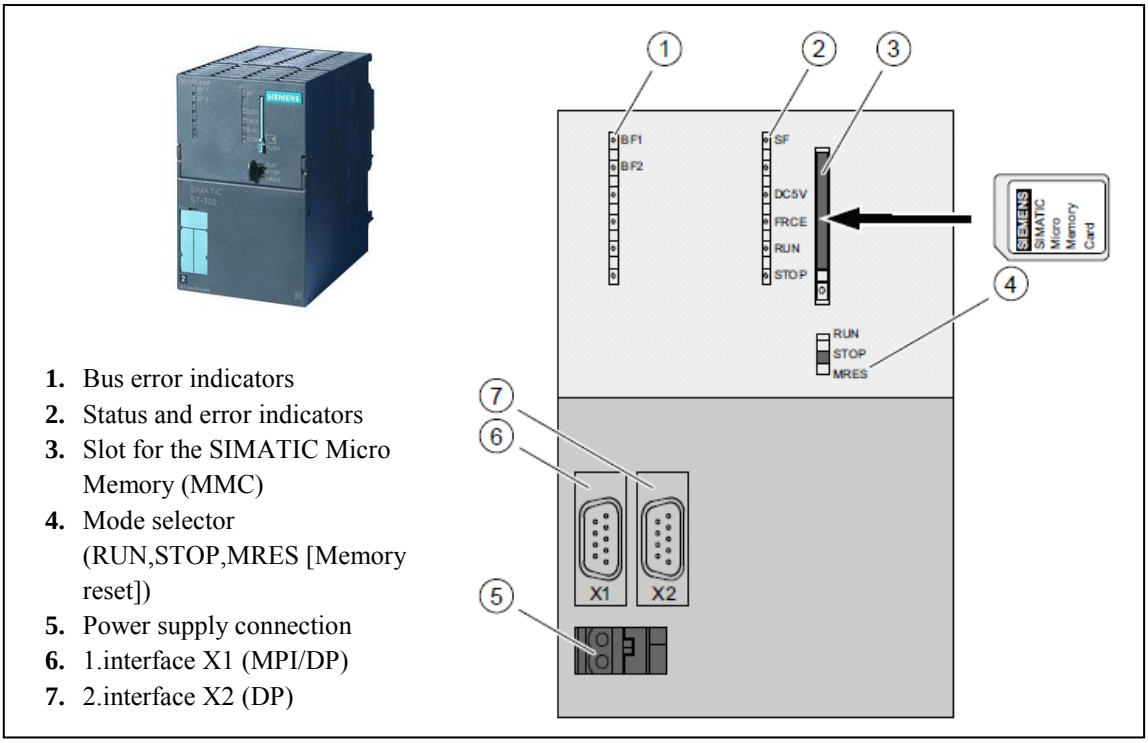
<table>
<tr><td colspan="4">- Compact design, mounting on DIN rail
- High processing performance in binary and floating-point arithmetic
- The 317-2 DP CPU features an MPI/DP interface plus an additional DP interface. Data exchange with other PLCs or PCs implemented via MPI[21], Profibus or Industrial Ethernet, allowing the features of the PLC to be used to their full extent within a distributed system
- Connection of field devices to the controllers is supported by Profibus DP
- PROFIBUS DP master/slave interface
- For extensive I/O configurations
- For setting up distributed I/O structures</td></tr>
<tr><td colspan="4">Technical Data</td></tr>
<tr><td>Power supply</td><td colspan="3">Integrated 2-pole power supply socket</td></tr>
<tr><td>Memory</td><td colspan="3">Integrated 512 KB
Load memory – pluggable (MMC), max 8MB</td></tr>
<tr><td colspan="4">Interfaces</td></tr>
<tr><td rowspan="4">1st interface
X1</td><td colspan="3">- Type: Integrated RS485 interface
- Hardware: RS485
- Isolated</td></tr>
<tr><td colspan="3">Functionality</td></tr>
<tr><td>MPI</td><td>- CPU interface for PG/OP connections
- Default baud rate: 187.5 Kbps
- The CPU automatically broadcasts its bus configuration via the MPI interface</td></tr>
<tr><td>Profibus DP</td><td>- Used to connect distributed I/O, allowing to create large subnets
- Can be configured for operation in master/slave mode
- Supports transmission rate up to 12 Mbps
- The CPU broadcasts its bus parameters via the Profibus DP interface when master mode is set</td></tr>
<tr><td colspan="3">DP Master – Services
- PG/OP[22] communication
- Routing/Data set routing
- Constant bus cycle time
- Activate/Deactivate DP slaves
- Data exchange between master and slave
- DPV1, User data per DP slave: Inputs/Outputs max. 244 bytes
- Clock synchronization - broadcast frames</td></tr>
</table>

[21] The MPI interface on each CPU allows simple cyclic data exchange and programmed exchange of larger data volumes, with and with/without acknowledgement respectively.

[22] Protocol for programming a SIMATIC S7 and for communication with HMI devices (PG=Programming device, OP=Operator panel)

2nd interface X2	- As for 1st interface X1	
	Functionality	
	Profibus DP	As for 1st interface X1
	DP Master – Services As for 1st interface X1	
GSD file	Required for using the DPV1 functionality with DP slaves. The current GSD file is available for download from the internet (www.siemens.com/profibus-gsd)	

Table 19 Siemens S7 CPU 317-2 DP features



1. Bus error indicators
2. Status and error indicators
3. Slot for the SIMATIC Micro Memory (MMC)
4. Mode selector (RUN,STOP,MRES [Memory reset])
5. Power supply connection
6. 1.interface X1 (MPI/DP)
7. 2.interface X2 (DP)

Figure 28 Siemens S7 CPU 317-2 DP Operator Controls and indicators

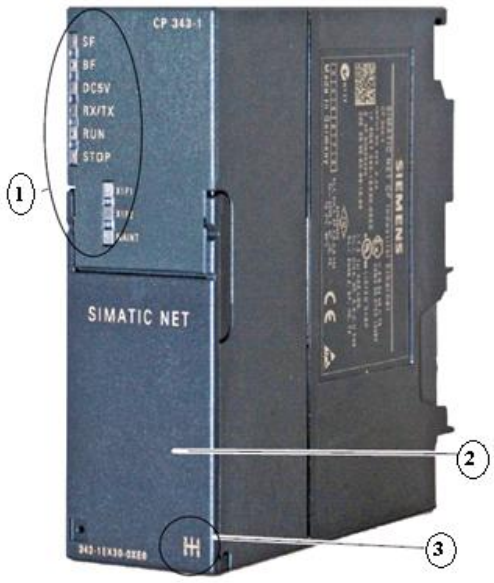
PG communication allows the exchange of data between engineering stations and SIMATIC S7 modules with communications capability, providing the functions needed to load programs and configuration data. This service is possible via MPI, Profibus, and Industrial Ethernet subnets.

The **routing** gateways (MPI, DP) between subnets are located in the SIMATIC station, allowing transfer of data beyond network boundaries. The **data set routing** is an extension of "normal routing" and is used in case the programming device is not connected directly to the Profibus DP subnet to which the target device is connected, but, e.g. to the Ethernet interface of the CPU. The data sent by means of data set routing include the parameter assignments for the participating field devices (slaves) and device-specific information. [37]

Communication Module for Ethernet CP 343-1 Lean (SIMATIC NET)

The Siemens CP 343-1 Lean communications processor is intended for operation in a SIMATIC S7-300, allowing attachment of the S7-300 to Industrial Ethernet. [40]

Features



1. LED displays (RUN, STOP, Link Status)
2. TP port: 2 x 8-pin RJ-45 jack (beneath the front panel)
3. Placeholder for hardware version

- It permits connection of the S7-300 to Industrial Ethernet
- PG/OP communication
- Transport protocol TCP/IP and UDP
- Open communication (SEND/RECEIVE)
- It is snap-mounted on the S7-300 DIN rail and connected to adjacent modules through the bus connectors

Technical Data	
Connection to Ethernet	2 x RJ-45 jack
Transmission rate	10 Mbit/s and 100 Mbit/s

Figure 29 CP 343-1 Lean

7.1.4 RS-485 IS transmission cable

The RS-485 transmission technology provides high transmission speed for systems in hazardous areas having a particularly high information flow. More details can be found in the following table [38]:

Technical data	
Data transmission	Digital; differential signals acc. to RS485, NRZ
Transmission rate	9.6 to 15000 Kbit/s
Data security	HD=4; parity bit; start/end delimiter
Cable	Twisted, shielded four-wire cable
Topology	Line topology with termination
Number of nodes	Up to 32 nodes per segment

Table 20 RS-485 IS technical data

7.1.5 SIMATIC NET CP5711

The CP5711 is a communication processor equipped with a Profibus interface. At the same time, it is a USB adapter for operation in programming devices (PGs) and PCs/laptops with USB interfaces.

The physical link between the MPI/DP interface and the MPI/DP network is via a RS-485 interface that is part of the module. Depending on the network configuration, data rates up to 12 Mbps are possible in the MPI/DP network.

Among the communication services available with the CP5711 are the Profibus DP master class 1 and 2, and PG/OP communication with STEP 7 software. [39]

Figure 30 CP5711

7.1.6 Ethernet – Profibus Interface (xEPI2)

The Trebing Himstedt (TH) xEPI2 module enables access to the communication system and connects the higher-level network structure with the field level, allowing integration into any type of automation structure. Its integration capability makes it e.g. possible to configure field devices via the communication levels Ethernet-Profibus and HART.

The xEPI2 can be used in the Profibus network acting as a master class 2 (acyclic DP master) and be operated in parallel to the master class 1 (of the control system). In this way, PC-based applications (such as FDT frame application, Asset Management systems and OPC servers) are able to exchange data with HART field devices, which are connected to the Profibus via a Remote I/O with HART functionality.

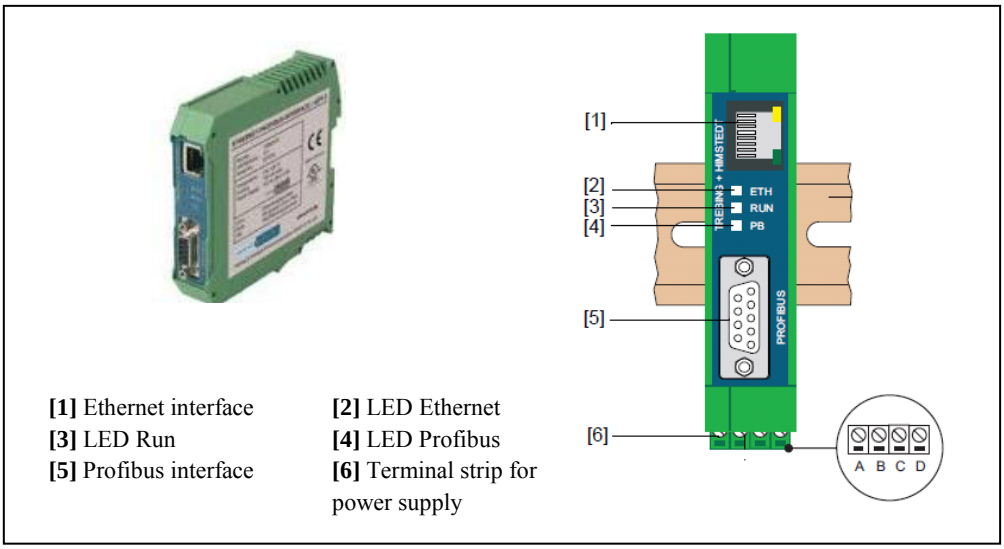
Hence, the T+H xEPI2 provides a flexible manner to; for instance, configure HART devices independently of the control system via Ethernet. See figure 31. [30, 42]

Features

- Access to the Profibus network
- Bus access for the FDT frame applications (such as PACTware), the HART over Profibus profile, and the TH OPC Server DP.
- Supports the EDD technology (Electronic Device Description)
- Supports Remote IO from Pepperl+Fuchs
- TH DTM Library (includes CommDTM Profibus DPV1) and TH

AMS Device Manager Communication Components (includes HART over Profibus) available for download on the internet (www.t-h.de)	
Technical data	
Transmission rate Profibus	Max. 12 Mbit/s
Profibus interface	RS-485
Ethernet interface	RJ-45
Mounting	35 mm DIN top hat rail

Table 21 TH xEPI2 features



[1] Ethernet interface
[2] LED Ethernet
[3] LED Run
[4] LED Profibus
[5] Profibus interface
[6] Terminal strip for power supply

Figure 31 xEPI2 module and connection diagram

7.2 Software

A short description of the software packages employed in this assignment is given in the current section.

7.2.1 SIMATIC S7

The Siemens SIMATIC S7 (STEP 7) software provides an integrated engineering environment to program and configure programmable logic controllers (PLCs), as well as PC-based automation systems. Furthermore, it has ability to communicate via MPI, PROFIBUS DP and TCP/IP ensuring a high degree of flexibility.

STEP 7 contains several tools and functions for the most varied tasks in an automation project, such as configuring and parameterizing the hardware, specifying the communication, programming, test, start-up and service, documentation and archiving, and operating/diagnostics functions.

The main components of STEP 7 are the **SIMATIC Manager** and the **Hardware Configuration Tool**. The SIMATIC Manager administers all data belonging to an automation project. In addition, it is used for creating, copying, downloading and archiving of projects. In this way, all project data can be stored in the memory card of the CPU. [43]

The Hardware Configuration tool is used for configuring and parameterizing the hardware used. This requires the corresponding GSD files to be installed. Consequently, the succeeding functions are enabled:

- ♣ configuration of the automation system racks which are selected from an electronic catalog, and then the selected modules are assigned to the require slots in the racks
- ♣ configuration of the distributed I/Os (e.g. DPV1 slaves)
- ♣ communications processor (CP) parameter adjustment
- ♣ configuration and display of communication links

7.2.2 DP Class 2 Master software

Profibus DP class 2 master devices make use of software packages, such as PACTware and Emerson AMS Suite, to offer HART communications**.** The succeeding sections give a brief description of these softwares.

7.2.2.1 FDT Frame Application - PACTware

PACTware is a fieldbus and manufacturer independent software for parameterization and configuration of HART field devices, remote I/O systems and communication components in field bus systems and networks.

The software interface between PACTware as a frame application and the devices corresponds to the open standard FDT, which provides complete functionality to manage data, to communicate with the device and to embed DTMs. [30, 44]

The individual software modules, so-called DTMs, are used by PACTware for operating the field devices and are therefore necessary to be installed. For the terms of this project, this applies for the DTMs corresponding to the VEGA sensors in use.

The main features of PACTware can be listed as follows:

- ♣ Displays measured values and diagnostic data from the HART field devices.
- ♣ Enables comprehensive adjustment of various field devices via any bus system able to, apart from actual values (process data), also parameter adjustment data.
- ♣ Communication DTMs are available for fieldbuses and communication types such as HART, Profibus and Ethernet.
- ♣ Manages access rights (roles) and access profiles for the field devices.
- ♣ Allows logical point-to-point communication between the DTMs and the field devices according to the network topology (routing), necessary for asynchronous communication.
- ♣ Supports the full range of functions of all field devices compliant with FDT/DTM technology.
- ♣ Offers diagnostics functionality and asset management functionality.

7.2.2.2 Emerson AMS Suite – Intelligent Device Manager

This diagnostic and predictive maintenance software provides advanced capabilities to handle HART communication. It enables access to the diagnostic data generated by HART field devices throughout a plant, increasing availability and giving the possibility to continuously monitor HART devices transmitting data via the HART Communication Protocol.

In addition, configuration and calibration of HART and Profibus DP devices can be performed by using the Emerson AMS Suite. Hence, identification, troubleshooting, and

resolving device issues can be managed remotely and efficiently within a single application. [45]

The core capabilities of the AMS Device Manager can be resumed as follows:

- **Diagnostics and monitoring** – to check the health of any connected wired device by viewing its status. Furthermore, an Alert Monitor feature is provided to have an overview of all device alerts. This includes alert latching and filtering.
- **Configuration** – making it simple, easy and user-friendly to configure the field devices. A number of functionalities such as change, store, compare and transfer of configuration information are available.
- **Documentation** – through the Audit trail, historical records of device configuration changes and performance are provided.
- **Host system interfaces** – enables full access to intelligent field devices. For instance, the HART over Profibus Interface allows the use of AMS Device Manager with HART devices in an online environment with a Profibus network via Ethernet. Thus, predictive diagnostics and real-time field device information such as process variables, device status, and alert events become available.

Based on EDD technology, the **HART over Profibus Interface** implementation on the AMS Suite enables diagnostics, configuration, and documentation activities on HART devices. The corresponding DDs are in that way required to be installed in order to take full advantage of the AMS software.

The interface allows central access via Ethernet to all HART devices connected to the Profibus using Remote I/O with HART functionality and an Ethernet-Profibus gateway such as the TH xEPI2 device (The AMS Device Manager System interface accesses the Profibus gateway by its IP address). Hence, the interface makes it possible to integrate different field devices of various manufacturers and manage them in one engineering system independently from the fieldbus.

7.2.2.2.1 TH AMS Device Manager Communication Components (TACC)

This software includes the HART over Profibus which works with AMS Device Manager to provide a communication solution for users with HART field devices connected to Profibus over Remote I/Os. It includes a “Set Bus Parameter Program” for configuring Profibus DP

master systems which are required for communication via Profibus (TH xEPI2 gateway is supported).

7.2.3 TH OPC Server DP

The Trebing Himstedt OPC Server DP is an OPC Server that functions via a class 2 DP master, enabling access to Profibus data. This comprises access to information provided by the class 1 DP master, DP slaves and their modules.

The OPC server makes diagnostics information on Profibus networks stations available, so that client applications can access information about failed devices, the status of the master system, and device diagnostics via OPC standards. Moreover, taking in consideration that the OPC Server DP supports OPC Data Access and Alarms & Events this information can be incorporated into any OPC-compatible software for the purpose of maintenance, alerting and production data acquisition. In that way, diagnostic messages, alerts and network status information can be saved and used as an alert history or for maintenance. [46]

The software includes a graphical configurator, which is used to set up the necessary configuration for the OPC Server operation. The configuration comprises entering device addresses and communication settings, creating the "namespace" which entails entering tag for each and every piece of information along with the memory register address for the parameter, its data type, and range where applicable. In this way, the parameters of DP master, DP slaves and modules which are to be monitored can be defined and assigned to OPC tags.

The functions of the TH OPC Server DP can be resumed as follows:

- ♣ Monitoring of all communicating slaves and detection of failed slaves (multi-master functionality)
- ♣ Diagnostics messages in plaintext as OPC alert
- ♣ Parallel reading of slave I/O data as OPC tags
- ♣ Flexible configuration of OPC tag names, e.g. by adopting the plant identification or structure
- ♣ OPC Data Access and Alarms & Events
- ♣ Reading and writing parameters
- ♣ Interpretation of manufacturer-specific diagnostics via GSD files

Nevertheless, the TH OPC Server DP has hardware and software requirements:

- **Communication processor:** TH xEPI2 device (max. transmission rate: 12Mbit/s, protocols: DP, DP-V1)
- **Software:** OPC Core Components 2.00 Redistributable 2.20

7.2.3.1 Matrikon OPC Explorer – OPC Client

The Matrikon OPC Explorer is a general-purpose OPC client, with functionality for identification of secure OPC servers, and for testing and troubleshooting OPC compliant servers and OPC connections.

The OPC Explorer provides the client interface for data acquisition via OPC Data Access, supported in the corresponding OPC server. [47]

8 GENERAL TEST SETUP SYSTEM

The following sections describe the configuration of the system employed to carry out the integration of HART sensors into components and software applications able to handle HART communication. In this sense, the system presents an integration approach to connect the devices to higher-level communication systems which provide the communication path for accessing the HART devices and data from remote locations.

8.1 Physical connection

The following figure gives a visual description of how the system is set up.

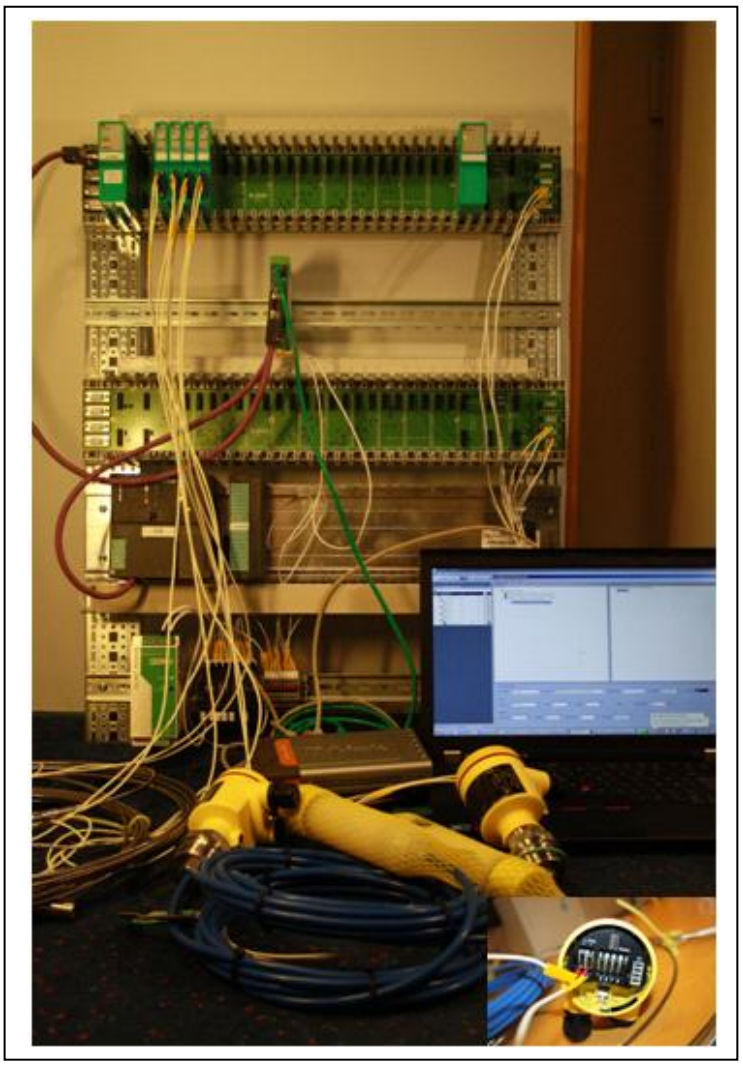


Figure 32 Physical connection

The HART field devices (Vega sensors) are connected to the Remote I/O stations (LB3102) which are mounted on a backplane that snaps onto a standard DIN rail. These I/O stations provide power supply to the 2-wire Vega sensors and feature HART communication through the single I/O channel available on each of the four LB modules used in the system.

The Profibus communication gateway LB8106 is connected at the left-end of the rail, providing further communication and enabling access to the HART devices via the Profibus network.

In this way, the HART sensors are connected to the HART-enabled signal conditioners (Remote I/O stations) which transfer the HART data to the Profibus DP gateway for incorporating the data into Profibus DP telegrams.

The Profibus network consists of the Siemens PLC (CPU 317-2 DP) which function as DP master class 1, the TH xEPI2 Ethernet-Profibus gateway that acts as the DP master class 2, and the Profibus gateway (LB8106) which serves as DP slave in the network. All of them are connected together using the RS-485 transmission cable, through the repective RS-485 interface port.

The HART data contained in the data field of the Profibus telegrams is transmitted from the DP slave to the DP master class 2. This is possible due to LB8106 module features DP-V1 asynchronous services.

The Ethernet-Profibus gateway embeds the Profibus DP telegrams in Ethernet (TCP/IP) messages, and connects the system to the Ethernet network via its RJ-45 port. An Ethenert cable is then connected from the xEPI2 device to a Ethernet router which provides several Ethernet ports for better accessibility. In this manner, the Siemens CP 343-1 Lean communications processor can also be linked, thus giving access to the Siemens PLC through the Ethernet network.

To complete the integration of the HART devices, an additional Ethernet cable is employed to attach the system (via the Ethernet router) to a PC/laptop where the HART data is finally recieved on the corresponding software.

8.2 System configuration

After performing the physical connection, it is necessary to configure the hardware components using the pertinent software, and establish the required communication paths. For this purpose, the Siemens SIMATIC S7 software is employed.

8.2.1 Configuring the hardware

The first step to take in order to configure the hardware components in the Profibus network is to open the SIMATIC Manager (built-in the SIMATIC software) and create a "New

Project"; assign it a name and insert a "New Object". In this case, the new object is the "Simatic 300 - station".

Once the Simatic 300 station is added, the specific CPU (317 - 2 DP) can be defined in the HW Config window which opens by selecting "Hardware" on the right side of the SIMATIC Manager window. See figure 33.

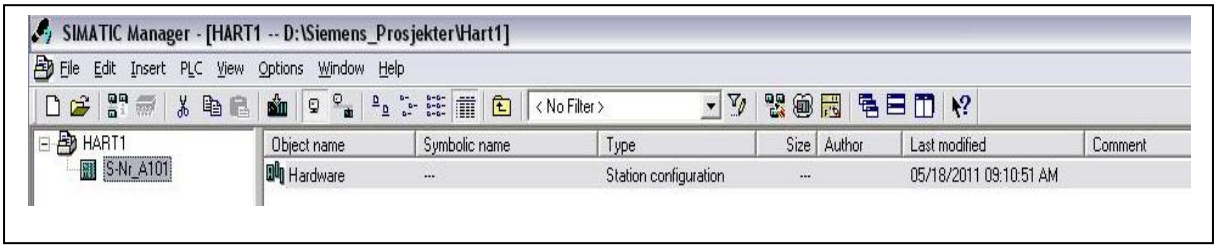


Figure 33 SIMATIC manager – New project

The assignment of the CPU is carried out easily but this requires that the "Rail" which will contain the CPU (and the communication module for Ethernet) had been created first, as shown in figure 34. The elements in the rail constitute the PLC and thus the Profibus DP master class 1.

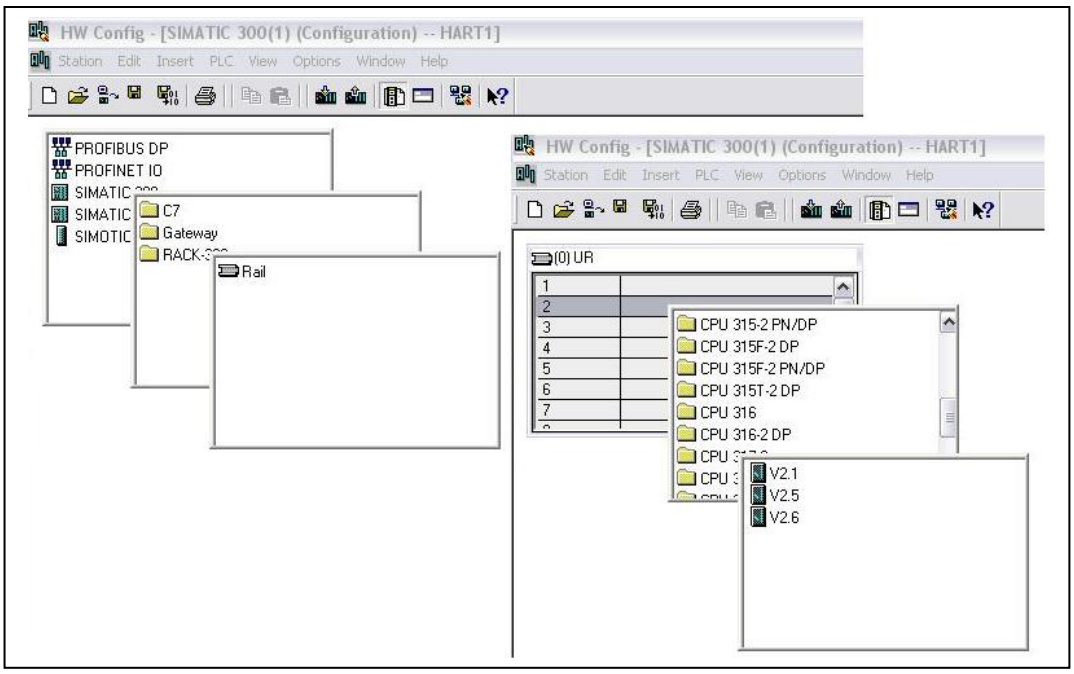


Figure 34 HW Config

After creating the rail, the CPU can be added and placed on track 2 in the rail (track 1 is reserved for power supply units). The “Properties – PROFIBUS Interface DP” dialog will pop up, where the parameters (address) and network setting can be set according to the project requirements. In this case, Address “2”, Transmission rate “1.5 Mbps” and Profile “DP”. See figure 35.

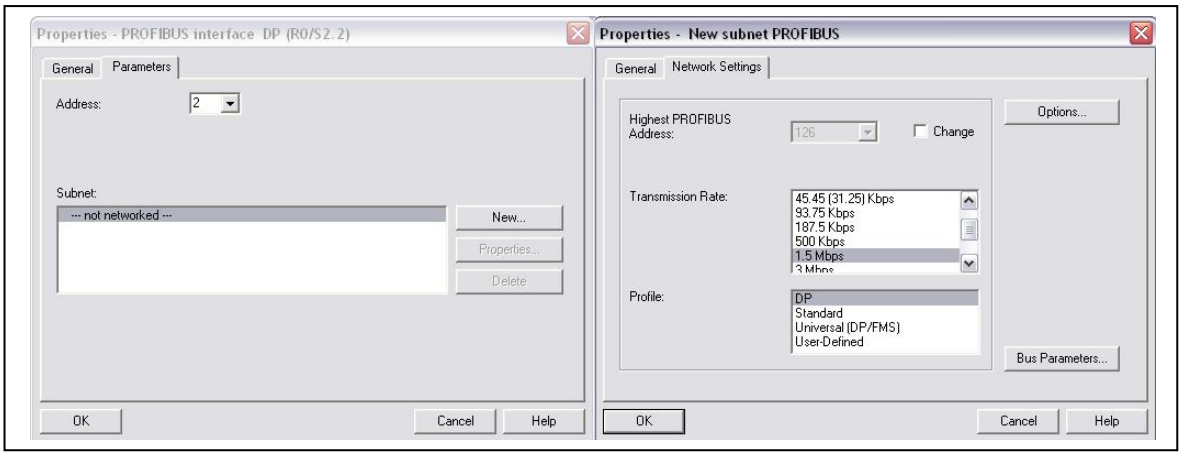


Figure 35 Profibus Interface DP and subnet

As seen on figure 35, the CPU is now assigned and the Profibus Interface DP created. Given that the required GSD files are already included and installed in the SIMATIC software, the next step is to assign the Profibus Com Unit LB8106 to the Profibus network by selecting the Profibus DP network line. The properties (parameters) of the module can then be set. For this project the Profibus address is set to 10.

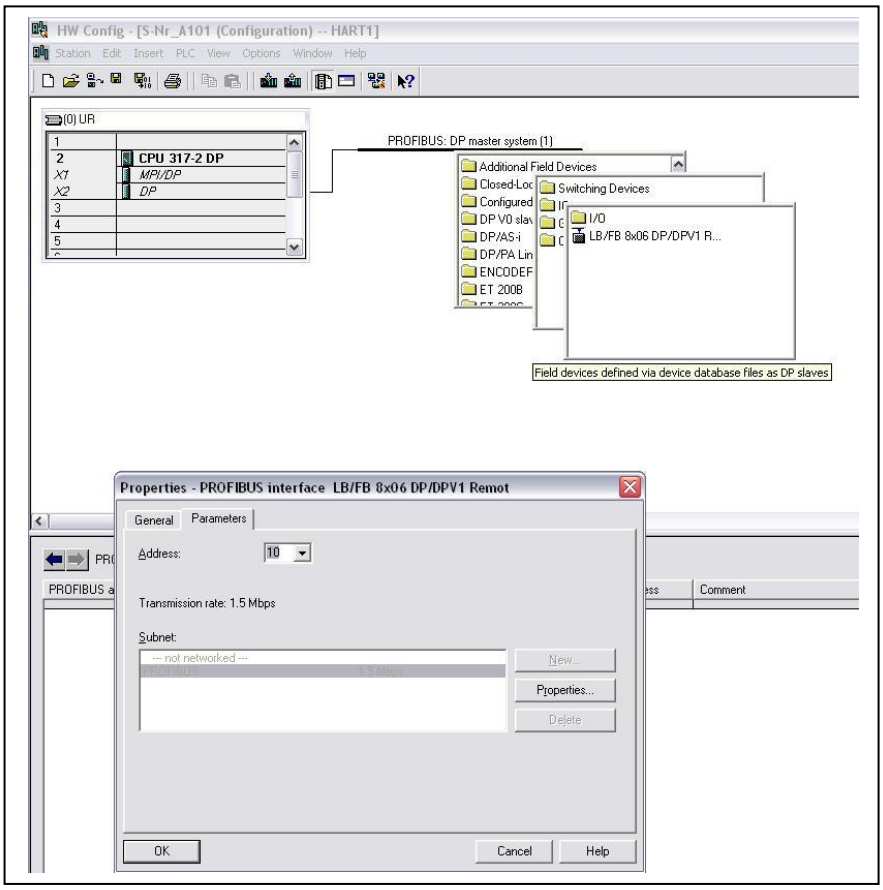


Figure 36 Profibus interface LB8106 DP/DPV1

The corresponding I/O stations (LB3102), which serve as HART interface to the connected sensors, can now be attached under the communication gateway (LB8106). For this purpose, the Com Unit must be selected to enable the slot list, and the module itself inserted into the first slot. This set up must coincide with the physical configuration on the backplane.

The LB8106 module occupies 2 slots on the backplane and there is one empty slot between the Com Unit and the I/O stations. For that reason there are two empty slots in the hardware configuration as seen on figure 37, and the four LB3102 modules are thus inserted on slot 4 to 7.

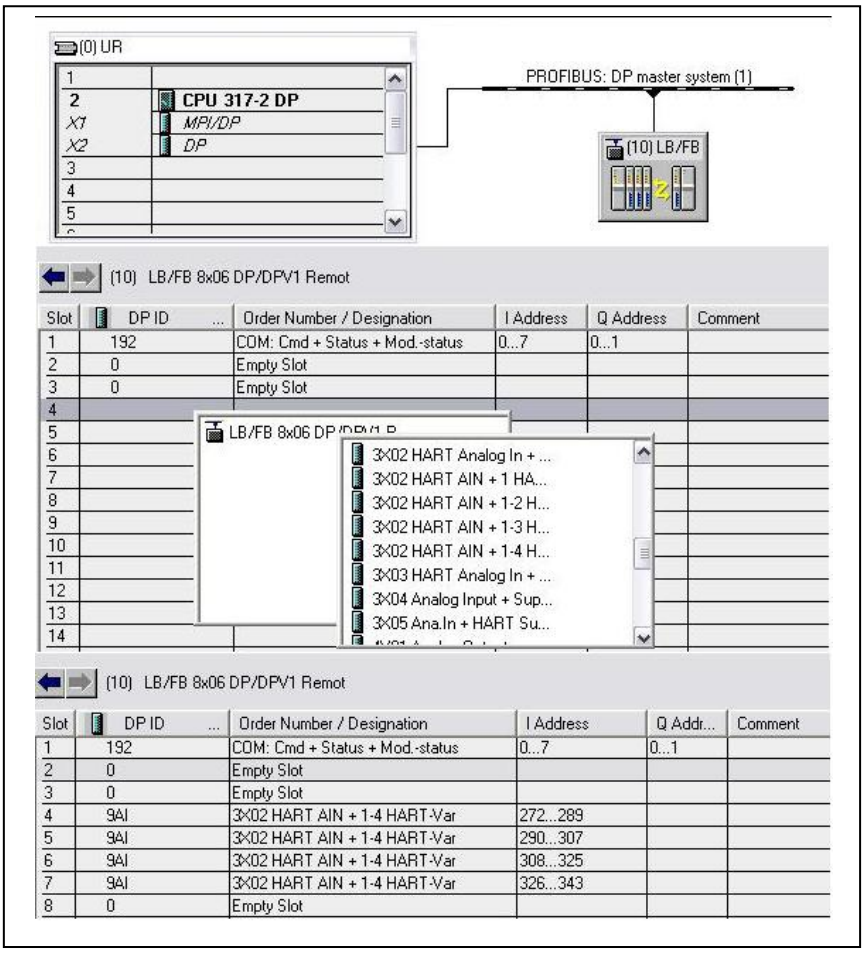


Figure 37 HART interface LB3102

Finally, the communication module for Ethernet (CP 343 – 1 Lean) can be added to the hardware configuration. The procedure is the same as it was for the CPU. This time the CP module occupies slot number four in the rail and it is assigned the IP address 192.168.10.101 (see figure 38). This enables the possibility to transfer data from and to the PLC utilising Ethernet communication in the event of later changes or updates in the hardware configuration.

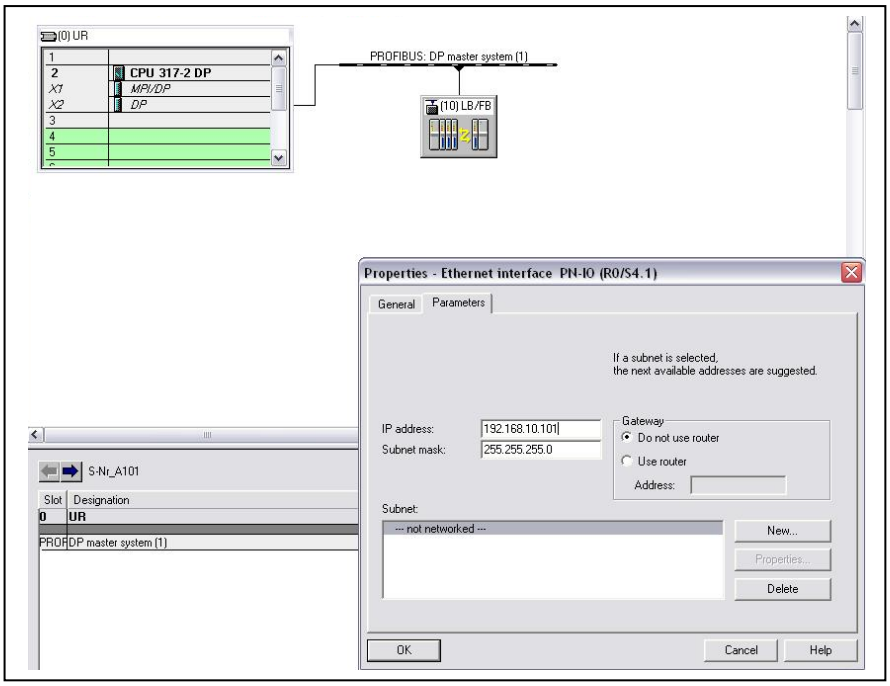
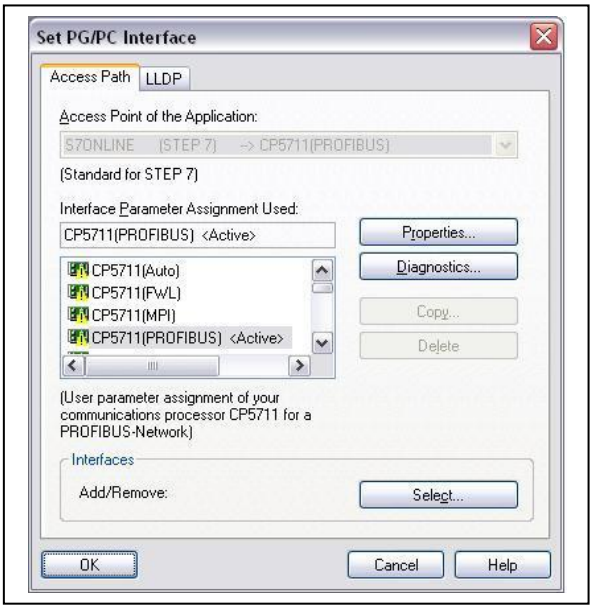


Figure 38 Ethernet interface

Before downloading the hardware configuration to the PLC which acts as DP class master 1 in the Profibus network, it is recommended to assign the Com Unit LB8106 a Profibus address. In this way, the communication between the PLC and the Com Unit can later take place swimmingly. This is done by using the RS-485 interface on the backplane of the connected gateway. The connection between the PC/laptop and the RS-485 interface is done using the CP5711 device.



Once the MPI/DP and USB cables from CP5711 device are connected to the backplane and PC/laptop respectively, the user must open the PG/PC interface dialog from the Options tab of the SIMATIC Manager window and select the Profibus interface, as shown on figure 39.

Back on the HW Config window, the Profibus address can be assigned from the PLC menu tab. The New Profibus Address is set to 10. See figure 40.

Figure 39 Set PG/PC Interface - Profibus

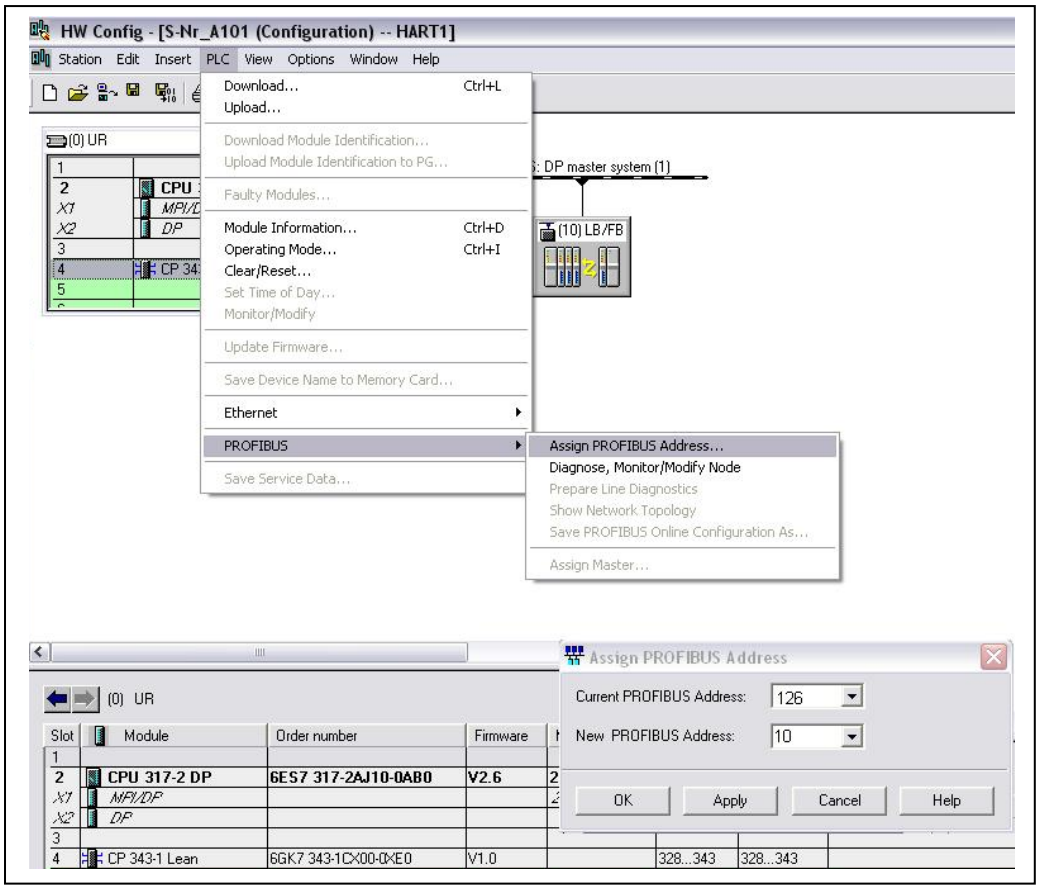
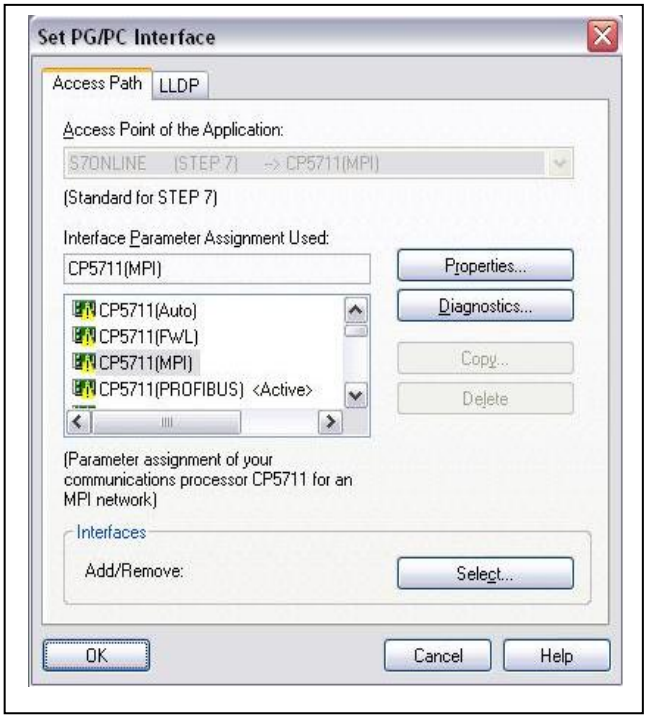


Figure 40 Assign Profibus Address



Prior to download the hardware configuration on the PLC, the MPI/DP cable must now be connected to the MPI/DP port on the PLC and the USB cable to the PC/laptop. Then a new PG/PC interface is set up. This time the MPI interface is used for this matter since the Profibus network had not been configured earlier on the PLC.

Finally, the configuration can be downloaded on the DP master (PLC) from the PLC menu tab from the HW Config window.

Figure 41 Set PG/PC Interface - MPI

8.2.2 Configuring the TH xEPI2 module

As already described, the xEPI2 module enables easy connection of Profibus networks to the Ethernet. In order to configure the device in the Ethernet network, a simple peer-to-peer method, where the IP addresses are manually assigned, can be employed.

Once the PC/laptop (with a web browser) is connected to the xEPI2 module using an Ethernet patch cable, the xEPI2 web site for configuration can be accessed entering the default (from fabric) IP address "http://169.254.0.1" in the web browser (e.g. Internet Explorer). For this purpose, the PC/laptop must be in the same subnet as the device, i.e. it has to be set to “255.255.255.0” which is the default address for the xEPI2 device.

After selecting the "Settings" tab, the information on the device is displayed. In order to be allowed to change the Ethernet network settings of the module, the user must be logged on as "Admin". The default password is the six-figure serial number of the device.

Thereafter, "Manual" configuration mode is selected as the configuration method, and the new IP address can be entered. For this project the new IP address is set to "192.168.10.10". See figure 41.

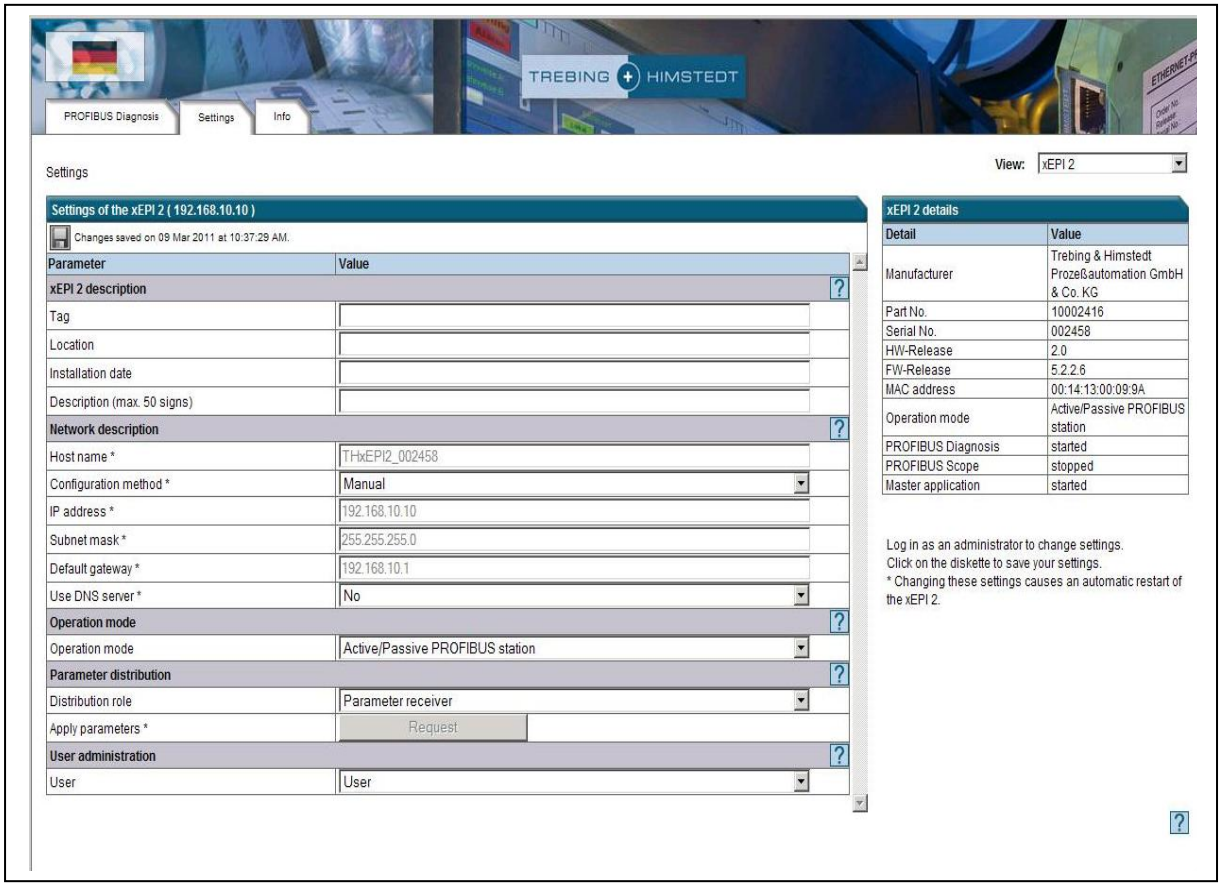


Figure 42 TH xEPI configuration web site

Furthermore, the xEPI2 has two operation modes "Passive PROFIBUS station" and "Active/Passive PROFIBUS station". The relevant mode for this project is the "Active/Passive" mode, since the xEPI2 module is going to act as a class 2 DP master in connection with an external master application (TH OPC Server DP, PACTware or Emerson's AMS Suite).

After selecting to save the entered settings and log out of the web site, the device will restart and the new settings become operative.

8.2.3 Configuration of the PC/laptop network card

To enable further data exchange between the TH xEPI2 module and the PC/laptop, it is necessary to change the network settings of the PC's network card.

The IP address on both units must be different. At the same time, they are required to have the same "Subnet mask" address. Microsoft Windows is the operating system running on the PC/laptop used for this study; hence the network card can be found listed under "Network Connections" in the "Control Panel".

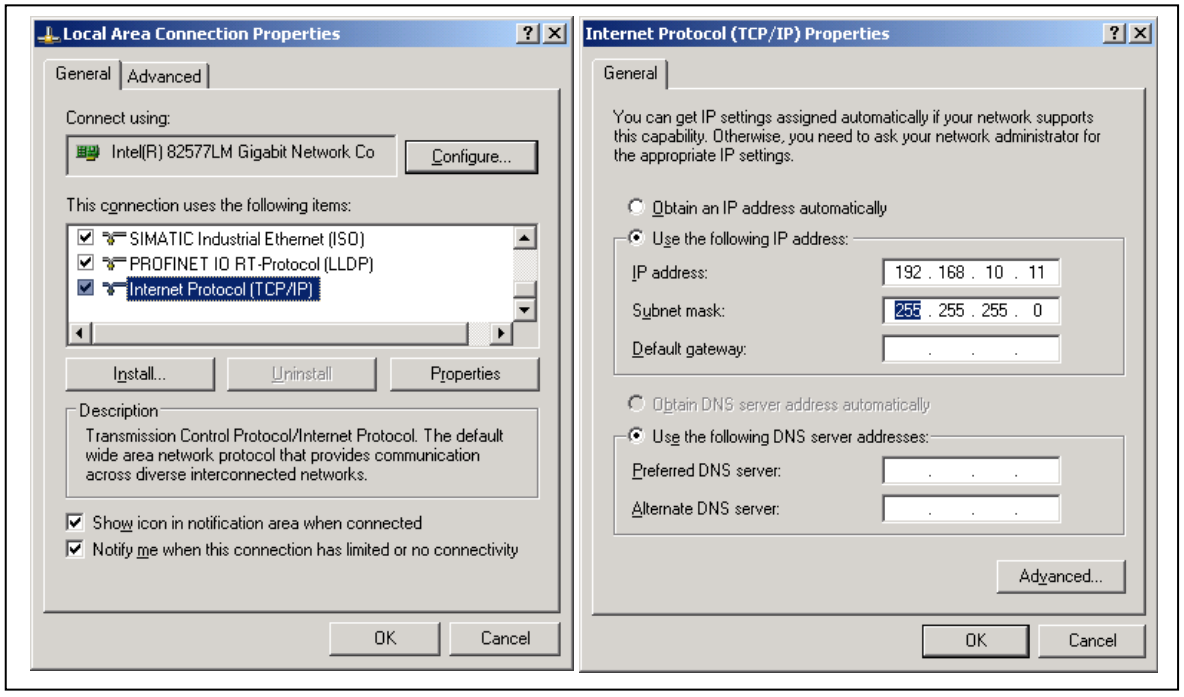


Figure 43 Network card settings

The IP address is then set to “192.168.10.11” under the Properties dialog for Internet Protocol (TCP/IP), as seen on figure 43.

9 TEST SETUPS

Under the previous chapter, the general configuration of the system has been carried out. The remaining and specific configuration which is required to perform the succeeding HART data integration tests will be taken along the test setups.

9.1 HART data integration, test setup 1 – PACTware (FDT/DTM)

The purpose of this test is to make use of FDT/DTM technology and the integration approach presented in chapter 8 to establish communication with the HART sensors and access to the HART data.

In order to integrate the HART data coming from the individual VEGA sensors attached to the test system (already described in chapter 8) into PACTware, certain software requirements must be fulfilled. All and each of the following DTMs have got to be installed before starting to use the PACTware FDT frame application:

1. The TH Communication DTM Profibus DP-V1 for the TH xEPI2 device
2. The Pepperl+Fuchs Communication DTM for the LB8106 gateway and the Device DTM for the LB3102 I/O modules.
3. The respective VEGA Device DTMs for Vegaflex 61, Vegawell 52, and Vegason 61 sensors.

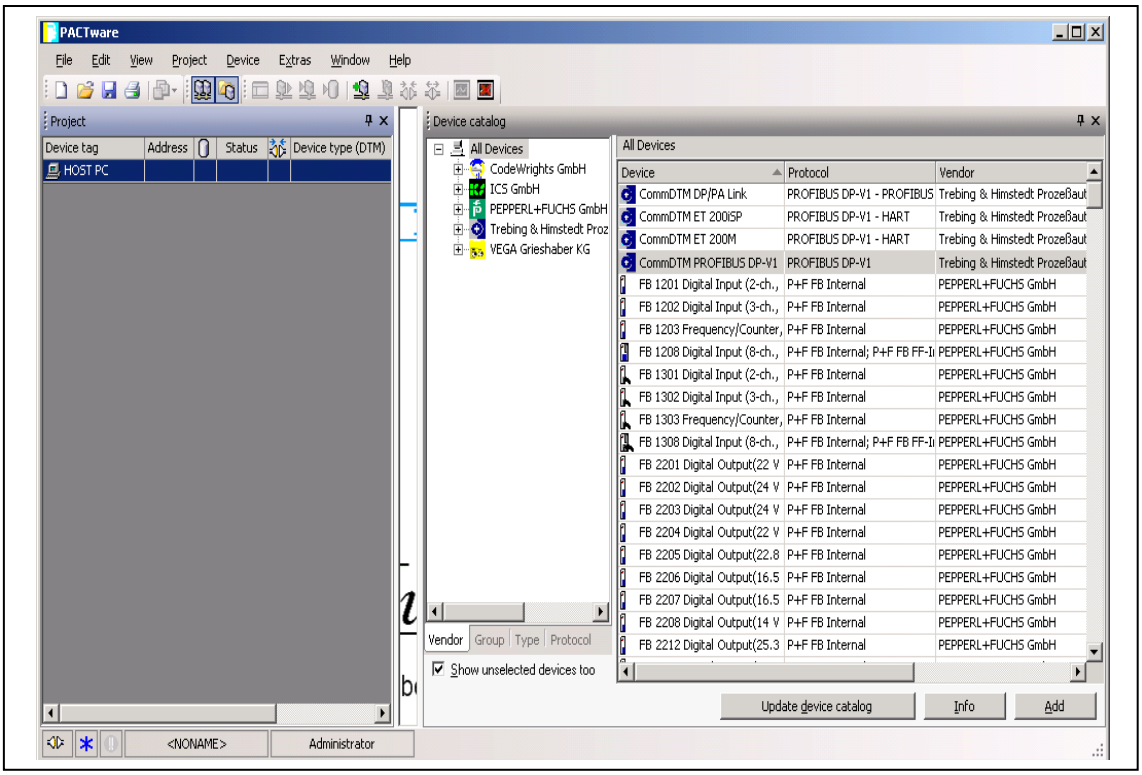


Figure 44 PACTware

Once the user starts PACTware, the devices can be added from the “Device catalog” into the “HOST PC” which in this case is the PC/laptop used for the integration test and that is connected to the xEPI2 device via Ethernet.

As seen on figure 45, the xEPI2 DTM incorporates a Parameter configuration dialog. After an automatic search, the device is found as well as its IP address (which has been set previously).

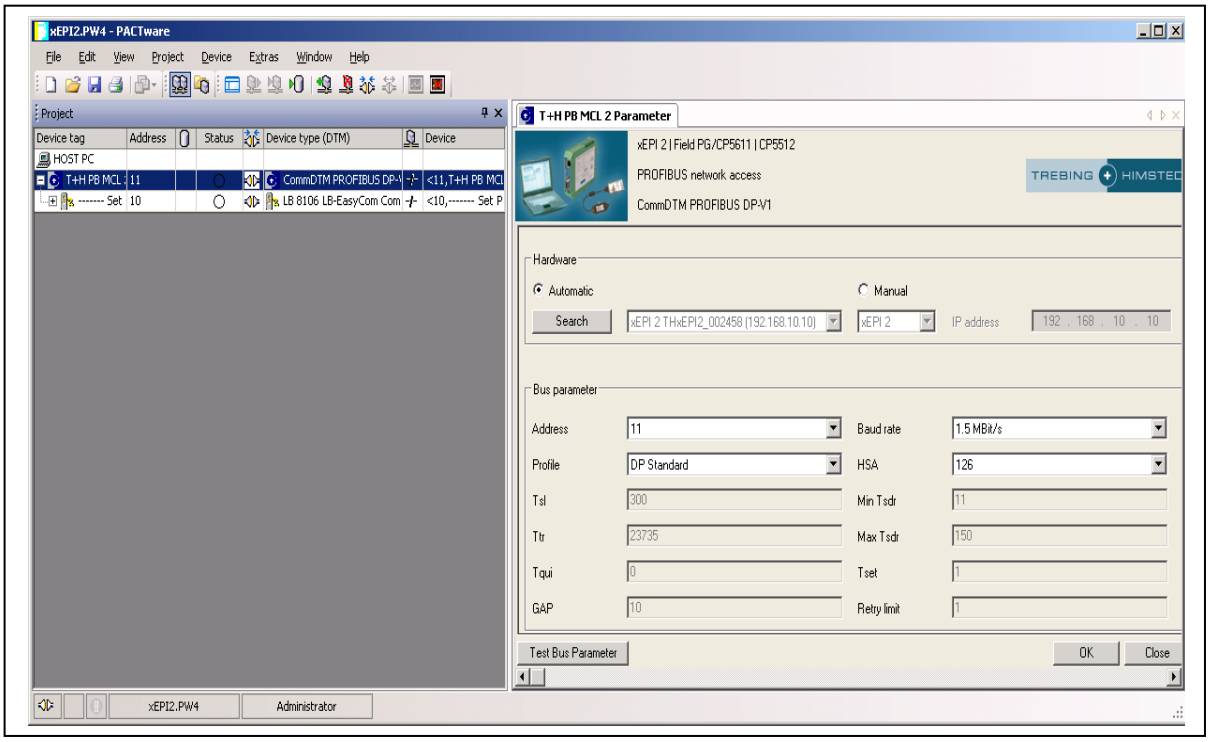


Figure 45 TH CommDTM

The “Bus parameter” address is set to 11 since the CommDTM is running on the PC/laptop with IP address 192.168.10.11.

On the same manner, when the LB8106 Com Unit is added the DTM address of the device must be set. In this case, it is set to 10 in order to match the already assigned Profibus address on the device.

The LB3102 I/O stations can then be added under the LB8106 Com Unit. The slot used for communication must correspond with the physical position on the rail where the stations are integrated. It can be noticed here that the slot number differs from the previously configured on the hardware configuration in SIMATIC. This due to the LB8106 Com Unit does not manage to see itself when adding the LB3102 I/O stations, while SIMATIC is able to see both the Com Unit and the I/O stations. Therefore the I/O stations are placed from slot 3 to slot 6 in PACTware and not from 4 to 7. See figure 47.

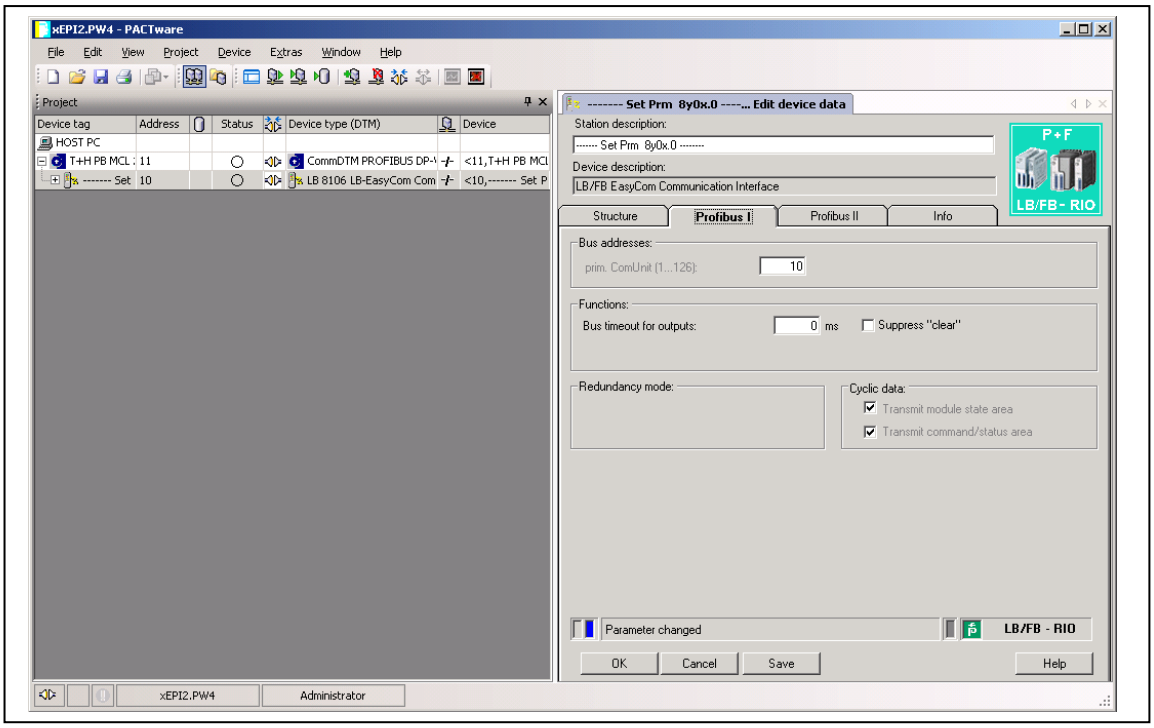


Figure 46 Pepperl+Fuchs LB8106 DTM

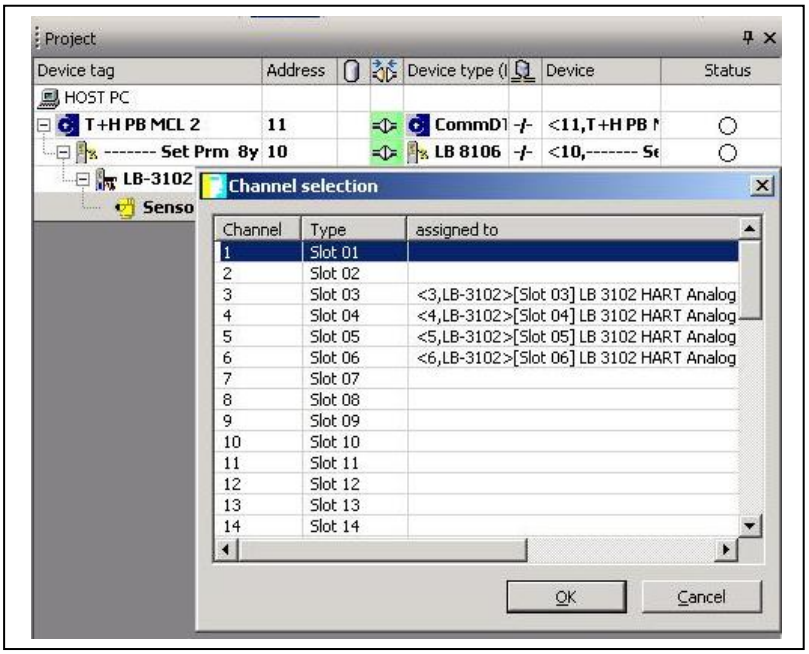


Figure 47 Slot selection

As last, the HART field devices (VEGA sensors DTMs) can be added as child objects of the I/O stations. The devices get automatically assigned address “0” due to the Pepperl+Fuchs LB3102 stations do not support any other HART communication mode than point-to-point[23].

[23] See chapter 3.2.1

To complete this test, the devices can now be connected in the same order as the topology (hierarchy structure) is built.

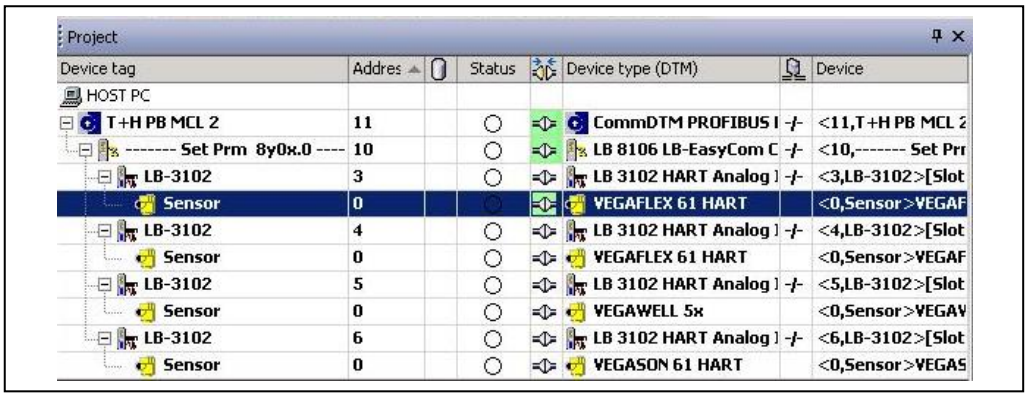


Figure 48 Communication path connection

9.2 HART data integration, test setup 2 – TH Profibus OPC server

As previously described the TH OPC Server DP enables access to Profibus data. The succeeding integration test setup aims to access the HART data incorporated in the Profibus telegrams and make use of the OPC functionalities to expose it.

For this purpose, the installed Configurator for OPC Server DP is started. Per default, the Configurator starts with a new project, where a Profibus network can be added.

Upon generation of the network, the DP master class 2 (TH xEPI2) which enables access of the OPC Server DP to the Profibus, is added automatically into the Profibus network in the event the device is already connected. However, it needs to be configured with the same bus parameters as the DP master class 1 in the network. Hence, the Profile must be set to DP, the Baud Rate to 1.5 Mbit/s and the HSA to 126. Besides, the device Address is set to 10 in order to match the previous configuration for the xEPI2 device (see ch. 8.2.2).

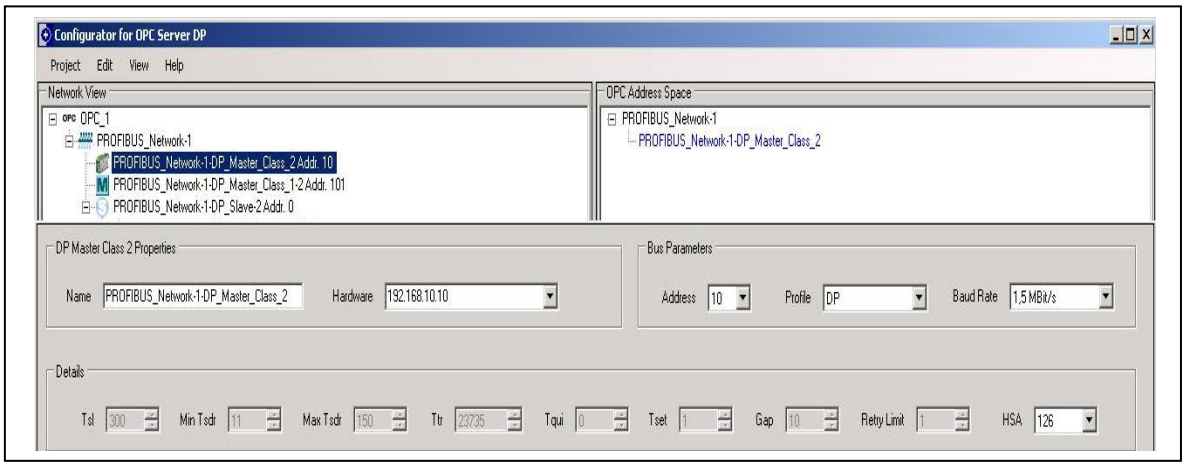


Figure 49 Profibus network view

The next step is to add the DP slave and modules under the selected the Profibus network. This will open a new dialog on the lowest part of the window where the name, address and GSD file path can be defined. In this project, the DP slave is the LB8106 Com Unit.

As last step, the OPC tags which will be used to read and write data can be created in the OPC Address Space and attached to the devices configured in the Network View. Once created, each tag must be configured according to their properties, service and mapping.

As an example, the "OPC_Tag-1" is created for the DP slave module and given Access to "Read". The Service is set to “DPV1 Read”, the Index is set to 211, the Length to 20 and the Polling Interval to 5000ms (minimum). See figure 50.

For the mapping configuration, the Byte Offset is set to 1, Bit Offset to 4, Data Length 16 and Data Type UnsignedInt16. This is set according to the data telegram information provided on the manual for the Pepperl+Fuchs Profibus ComUnit ([17]).

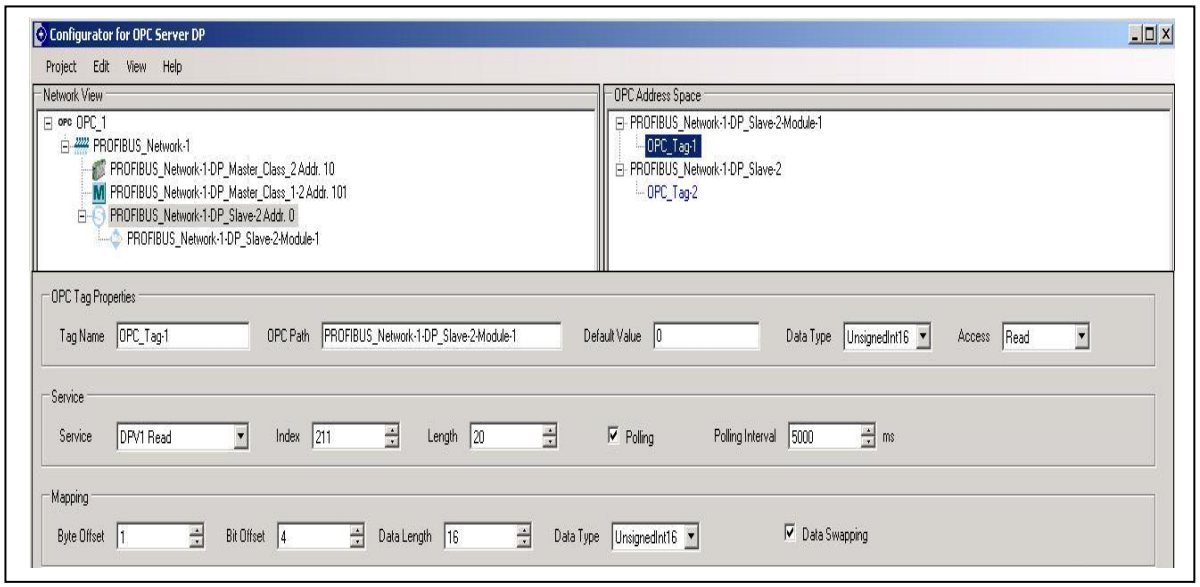


Figure 50 OPC Tag configuration

Once the creation and configuration of OPC tags is completed, the project must be saved and activated before running the OPC Server DP.

9.3 HART data integration, test setup 3 – Emerson AMS Suite

This test has the purpose to enable communication with the HART sensors and integrate the HART data into Emerson AMS Suite Device Manager employing the HART over Profibus integration profile. In this context, the TH xEPI2 device plays a significant role providing access to the Profibus network via Ethernet.

In addition to the AMS Device Manager software and the VEGA DDs, the T+H AMS Device Manager Communications Components (TACC) software is required to be installed.

Once the necessary software is installed, the first step to take is to configure the TH xEPI2 device using the "Set Bus Parameter Program". This is done by first adding the new hardware (Profibus DP master) and assigning it to a Profibus gateway (NOV).

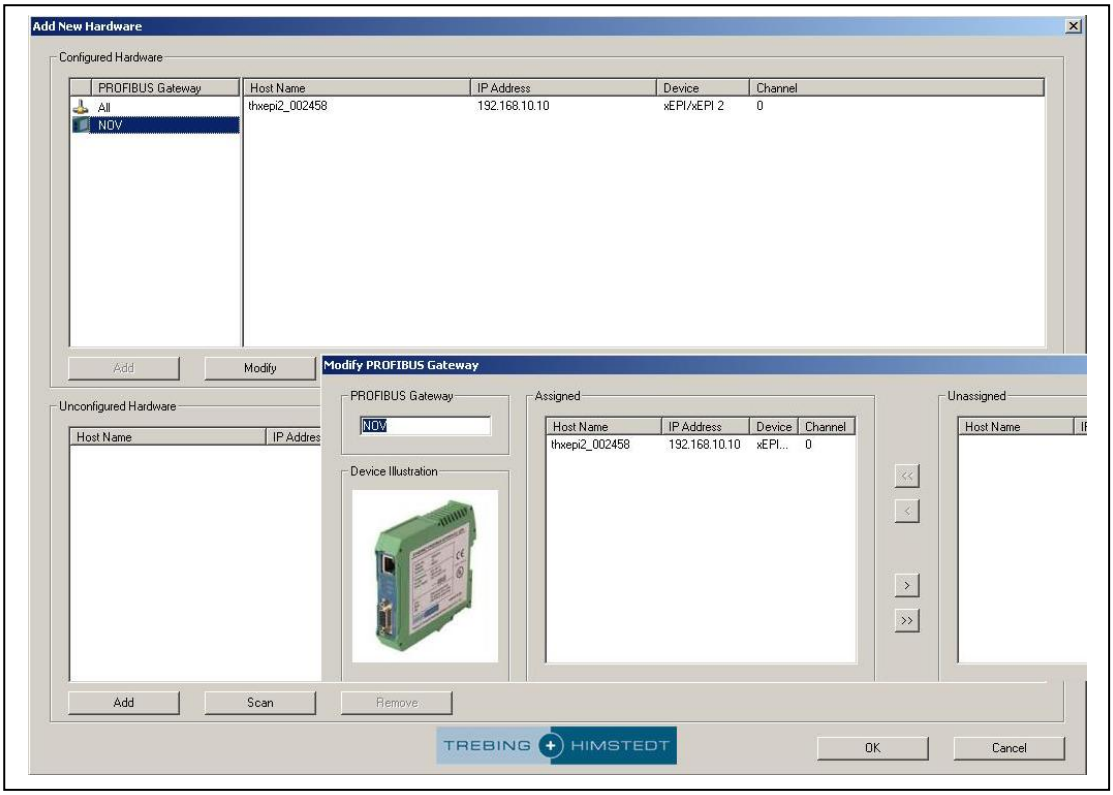


Figure 51 Set Bus Parameter – New hardware

Thereafter, the parameterization and activation of the Profibus DP master (class 2) can be performed in the Set Bus Parameter window. Selecting the required master, the current parameters are displayed. The device's Profibus address is set to any number different than 10, for instance 100, in order to avoid conflicts with the Profibus gateway LB8106. The Baud Rate must match the configured Profibus rate in the network, thus it is set to 1.5 Mbit/s. At last, the DP Bus parameter standard profile and HSA (126) is selected.

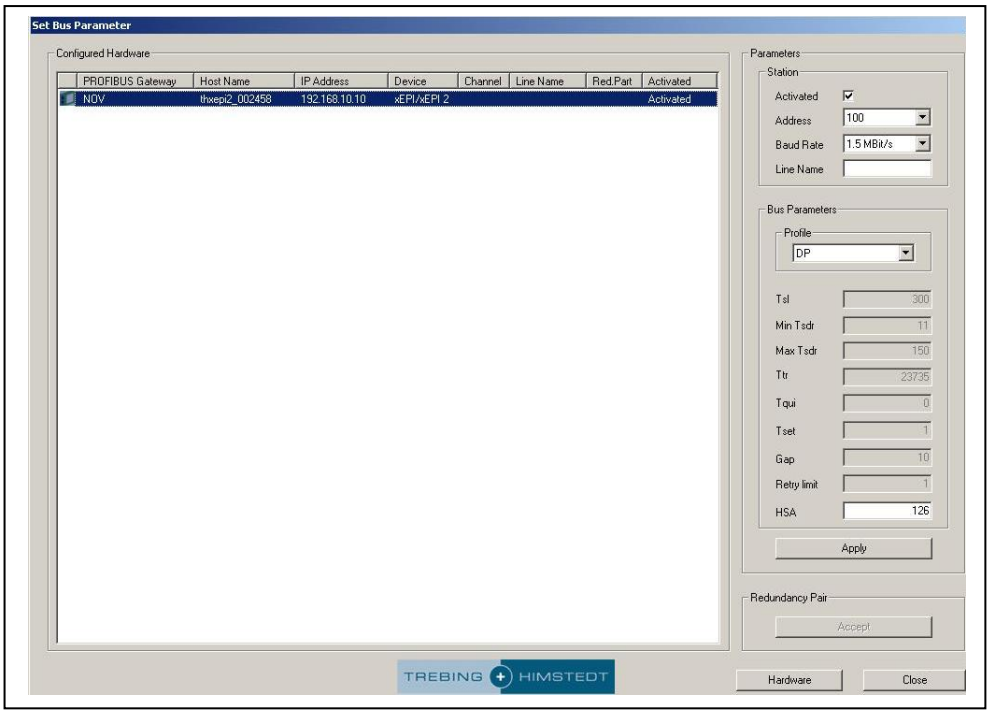


Figure 52 Set Bus parameters window

The network components, including the HART over Profibus, are installed utilising the Network Configuration utility which is part of the AMS software. This must be done before a network component can communicate with AMS Device Manager.

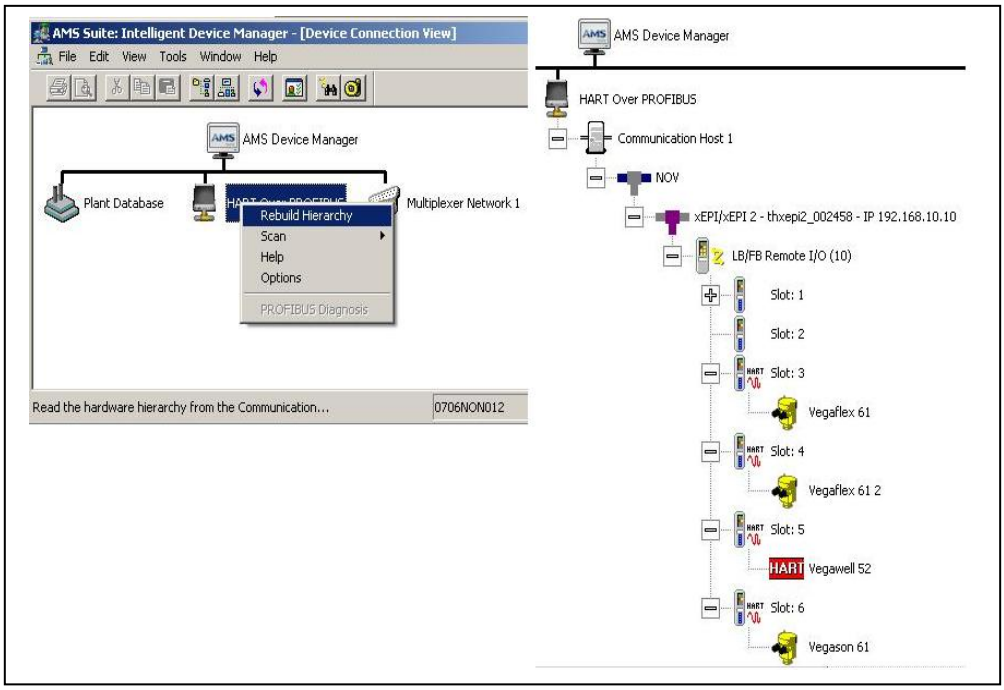


Figure 53 Network hierarchy

At the first AMS Device Manager startup the network hierarchy is displayed as shown in figure 53. It is then necessary to rebuild it in order to scan and find the devices on the network.

10 RESULTS AND DISCUSSIONS

10.1 Test results – HART data integration test setup 1

As seen on figure 48, the integration of the HART sensors is achieved using the Communication and Device DTMs specified for the components that feature HART, Profibus and Ethernet communication.

The FDT frame application, PACTware, provides access to the available sensor parameters making use of the corresponding device DTM. The information is presented in the "Measured values" and "Diagnosis" windows as depicted on figure 54. Additionally, "Online parameterization" options are enabled for device configuration prior to operation.

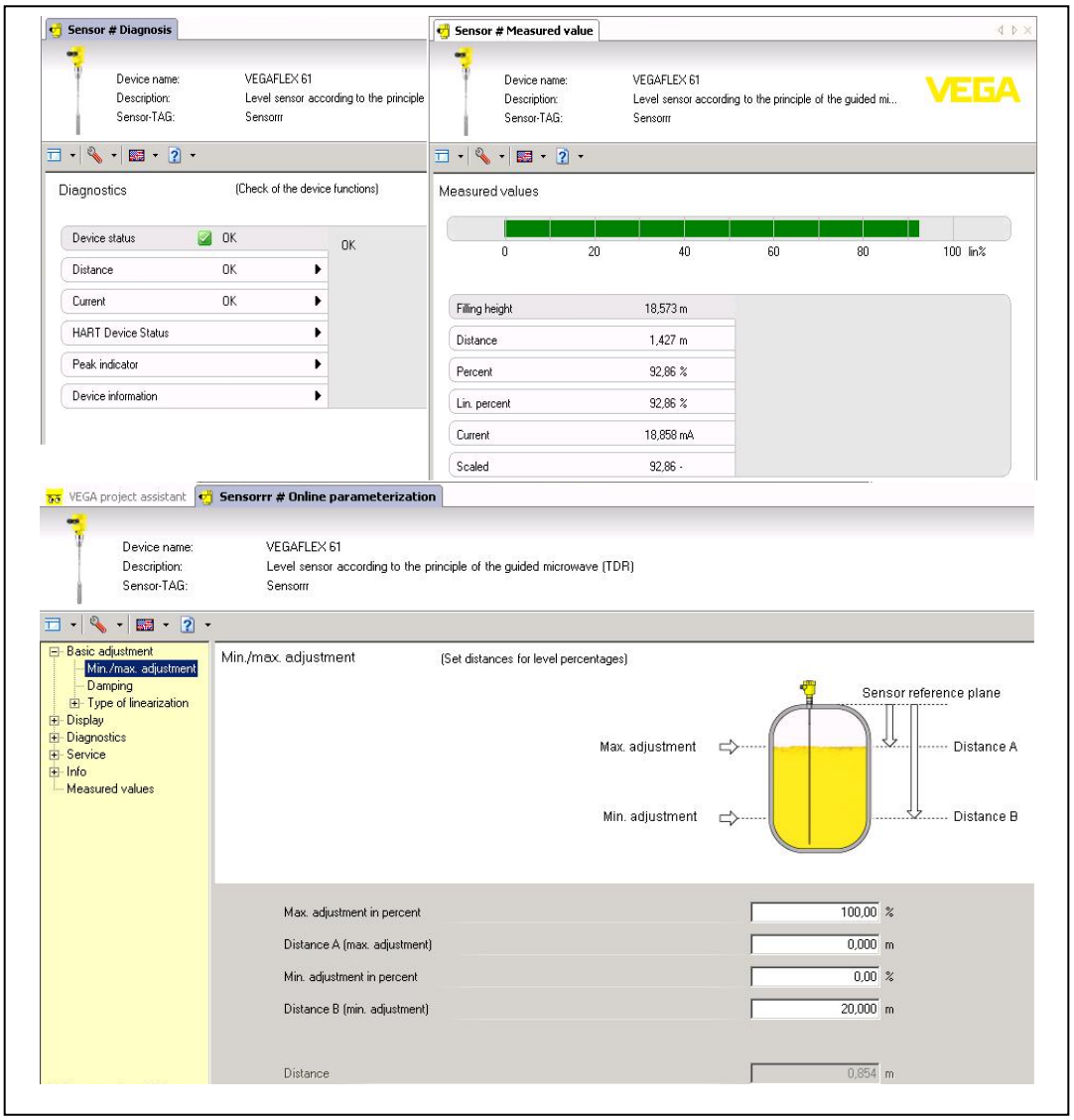


Figure 54 Sensor Diagnosis, Measured value and Online parameterization windows

The “Measured value” and “Diagnosis” windows provide direct information while the sensors are in operation. The first one shows the actual measured values including the PV, SV and TV

the sensors are capable to quantify via HART (Figure 54 shows only the measured values for the Vegaflex 61). The second one provides values such as peak measured values, device status and devices information which are useful for evaluating the overall condition of the sensors.

Among the configuration options, the “Basic adjustment” dialog allows to change the sensor-tag, unit of measurement, distances for level measurements, application medium and vessel type.

An important option under the Service tab is the “Current output” dialog which enables the user to set up the “Failure mode” current and “Min. and Max” current. The “Diagnostics” dialog provides comprehensive information of the device status and echo curves. Detailed information of the sensor such as version, material, process fitting, process temperature is displayed within the “Info” dialog.

When the user is logged on in "Service" modus, additional "Special parameters" becomes available under the Service tab, allowing the user to set different parameters such as start of operating range, threshold for first large echo detection, measurement value filter with hysteresis, fault signal on loss of echo, provided for the field device through its DTM.

Furthermore, PACTware includes an alert monitor window containing ongoing error messages so that any failure occurrence can be detected and may be resolved immediately.

All the available data on the sensor is uploaded to PACTware once it is selected to. However, due to the large amount of data contained in the device, the loading time is very high. The following table gives some approximate times when uploading data from the HART devices into PACTware.

Time Test	Sensor - Window	Time	
		Min	Max
1	**Vegaflex 61 – Measured value**	58 sec.	4 min. 20 sec.
2	**Vegaflex 61 – Diagnosis**	30 sec.	3 min. 10 sec.
3	**Vegaflex 61 – Online parameterization**	2 min.	2 min. 30 sec.
4	**Two Vegaflex 61 – Diagnosis**	57 sec.	4 min. 10 sec.
5	**Vegawell 52 – Diagnosis and** **Vegason 61 – Diagnosis**	55 sec. 3 min. 55 sec.	2 min. 12 sec. 4 min. 18 sec.
6	**Vegaflex 61 – Diagnosis** **Vegaflex 61 – Diagnosis** **Vegawell 52 – Diagnosis and** **Vegason 61 – Diagnosis**	1 min. 4 sec. 1 min. 11 sec. 1 min. 26 sec. 5 min. 3 sec.	5 min. 40 sec. 5 min. 23 sec. 3 min. 8 sec. 5 min. 17 sec.

Table 22 PACTware uploading times

The minimum times provided in table 22 corresponds to the amount of time it took PACTware to show the selected window after it had been shown more than one time before. I.e. the data had already been uploaded previously.

The maximum times represents the time it took to open the selected window for the first time. The times for the last two time tests are sequential. This means that the data for the first sensor was available after the first given time, for the second sensor after the second given time and so on.

It is to notice that PACTware tries to load up all available data from the sensors; no matter which window was selected to or how many times the same window had been opened before.

From table 22 it can be observed that the amount of time decreases when PACTware tries to load up data from several sensors at the same time; at least when the uploading had been executed more than one time. This can be connected to the fact that Profibus protocol is quite speed-effective when it comes to transfer large amounts of data.

In resume, the process time to activate HART DTM functions such as "Measured values", "Diagnosis" and "Online parameterization" on PACTware is in general quite high (see table 22). However, the general performance of this FDT frame application is good, and it offers a reliable and stable system which provides useful diagnostics and configuration functionalities for HART field devices.

10.2 Test results – HART data integration test setup 2

The network hierarchy view within the AMS Device Manager gives a comprehensive display of how the Vega sensors are integrated in the system. This enables the possibility for managing the field devices using the corresponding Device Description (DD). Figure 55 shows the list of available options, such as Configure/Setup, Device Diagnostics, Calibration, Process variables and Audit Trail.

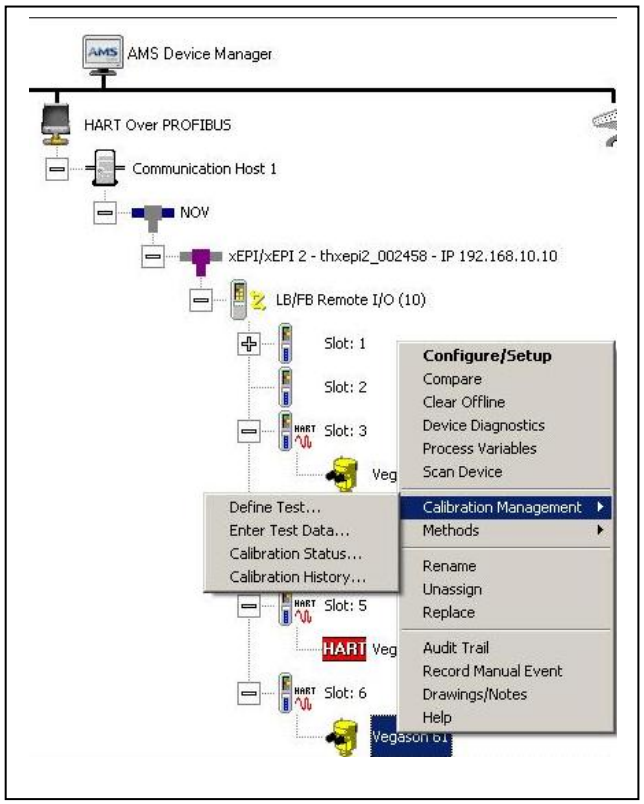


Figure 55 Managing options

When either the -Configure/Setup, Device Diagnostics or Process Variables- option is selected, a new window is displayed from which the information (transmitted via HART-Profibus-Ethernet) of the selected device is presented.

It is important to notice that the integration of the Vegaflex 61 devices is achieved although the Device Description (DD) files used by the AMS software do not correspond to the most updated one as seen on figure 56.

In the case of the Vegawell 52, the DD file was not at hand by the time of the test. However the device is integrated through a generic HART DD.

Among the configuration alternatives, the user is allowed to select the primary variable (PV) unit, upper and lower range values, tag ID, polling address and burst mode.

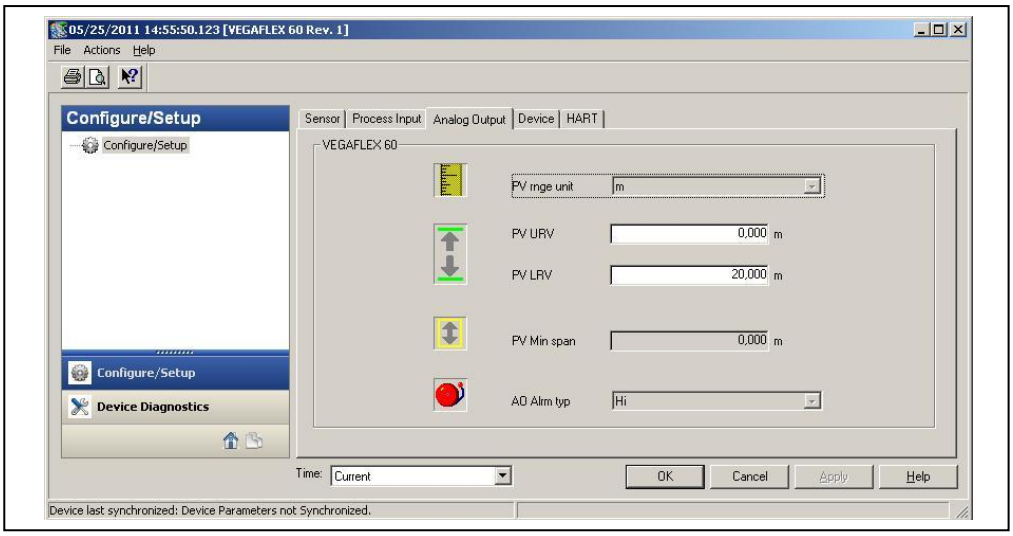
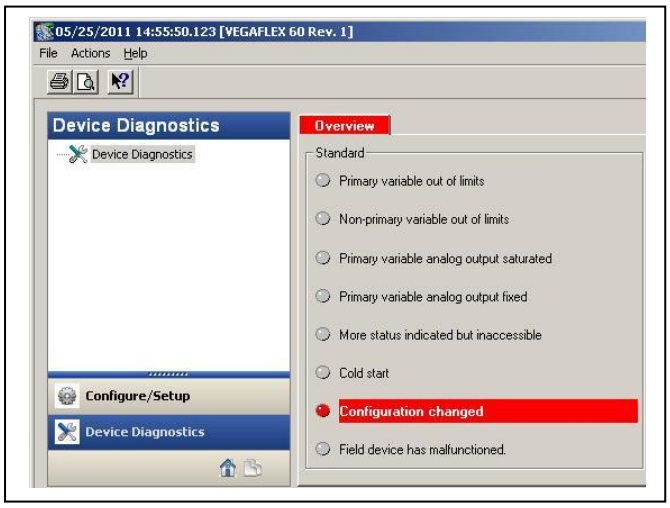


Figure 56 Configuration/setup



A brief overview of the accessible diagnostics is shown in the Device Diagnostics dialog. This includes: primary and non-primary variable out of limits, changed configuration, malfunctioning field device.

Figure 57 Device Diagnostics

The Process Variables dialog provides comprehensive sensor data such as the measurement of the primary variable (PV), PV% range, and current among others, as seen on figure 58.

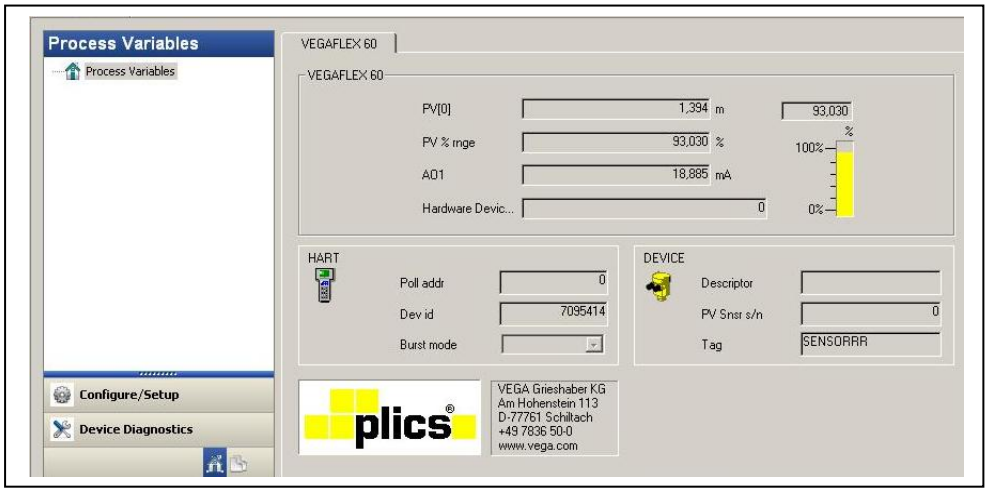


Figure 58 Process Variables

Another useful functionality of the AMS Device Manager is the Alert Monitor which is a diagnostic tool used to observe selected devices that may be malfunctioning or reporting false data. For this purpose, alerts must be configured in the Alert Monitor in advance. In this way device failures can then be corrected.

Once the device to monitor is selected, it is possible to configure the alert in the Alert Monitor configuration dialog. Different categories such as Abnormal, Maintenance Advisory and Failed alerts are to be found in the group of alerts. Furthermore, HART devices are polled at a configurable period in order to determine if an alert condition exists.

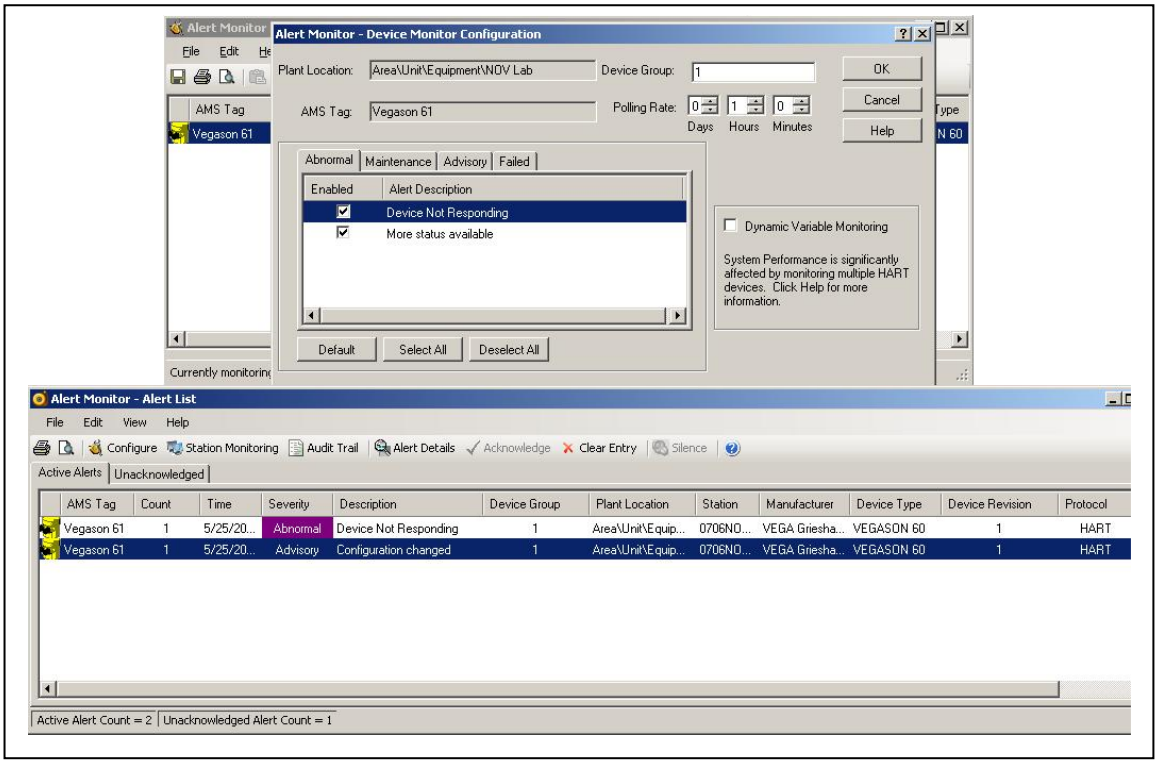


Figure 59 Alert Monitor Configuration

To measure the response time of the configured alarms, the Vegason 61 device was intentionally disconnected from the I/O station. In this way the alarm was triggered. This was executed several times, resulting in an average response time of 6 seconds.

The AMS Device Manager maintains an Audit Trail of historical records, also known as events. These include application, calibration, configuration and status alerts. In this way, the Audit Trail provides the ability to reconstruct every event that has occurred for the AMS Device Manager system and the field devices.

10.3 Test results – HART data integration test setup 3

The TH OPC Server DP acts as a transport mechanism for communicating with the HART field devices. The interface via the Server to the devices is achieved using the xEPI2 device supported by the Server software. The device receives and transmits the Profibus telegrams provided by the Profibus Communication gateway LB8106; which incorporates the HART data into the Profibus messages via DP-V1 acyclic services.

The OPC Server allows an OPC client to access the defined OPC items. For this purpose, the Matrikon OPC Explorer is employed. The built-in functionality of this OPC DA client provides easy connection to the TH OPC Server DP - ProfibusOPCSDA in figure 60.

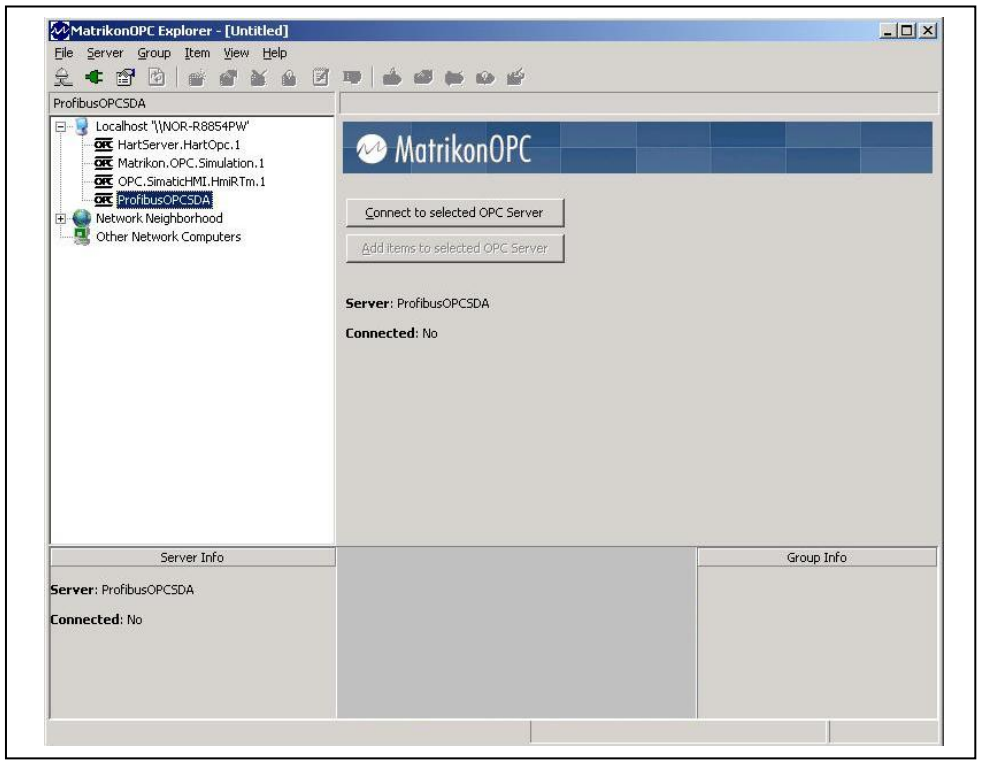


Figure 60 Matrikon OPC Explorer

As specified in the OPC specifications, an OPCGroup object called “TEST” is created in the server in order to access the data. This allows OPC items to be added to the group and provide read and write access to the OPC client. Basically the OPC client "Browses" the tag provided by the server and "Subscribes" to the data item of interest. In this way, the user can select the tag that the Server will “Publish” and present through the client, as depicted in figure 61.

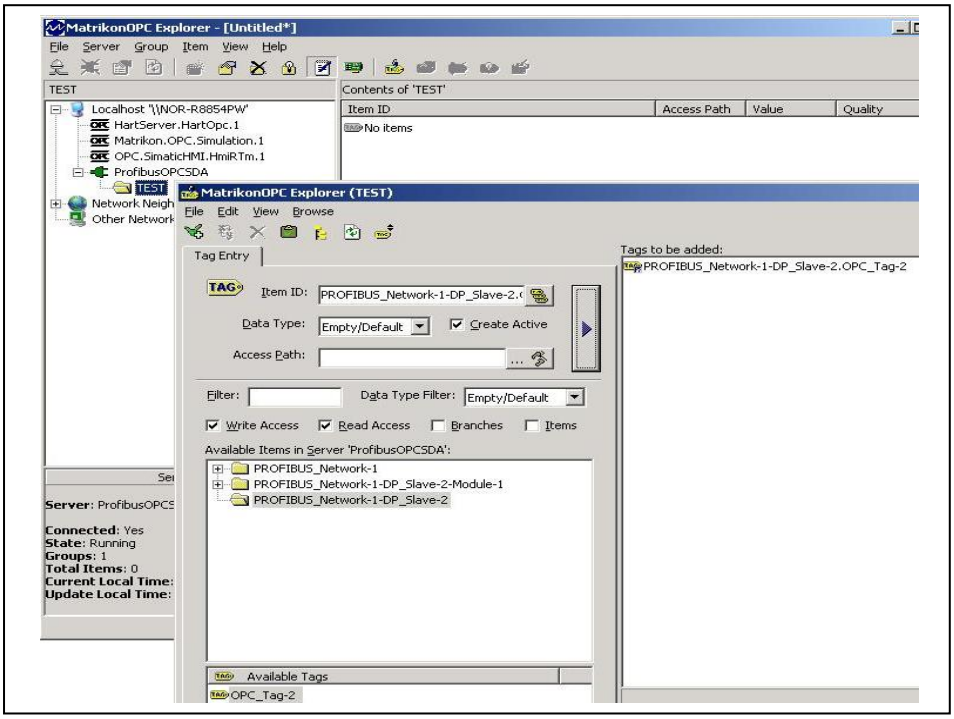


Figure 61 Adding OPC Tags

Once the items are added to the OPC Group, the Matrikon OPC client continuously updates the display for that group with real-time data, as depicted on figure 62. The items are listed with its item ID, value, active state, quality and timestamp.

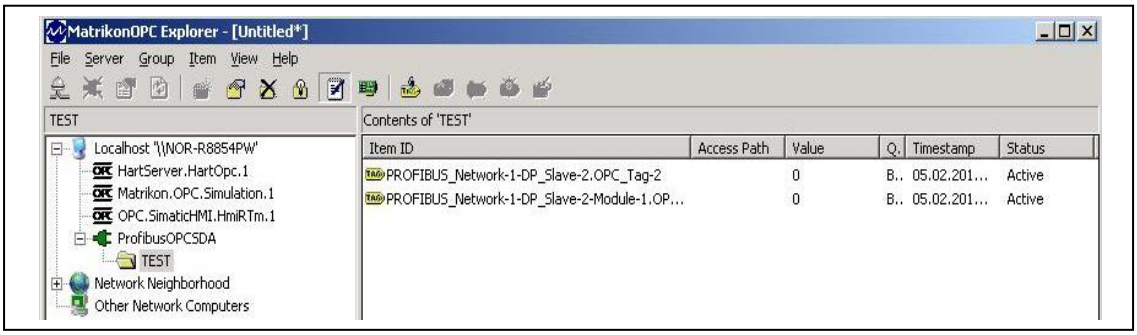


Figure 62 OPC Client

The value field shows the real-time value while the quality field indicates whether the value is valid or not. In this case, the configured tag in the OPC Server gives “0” as value. This suggests that the tag mapping settings are not completely in accordance to the Profibus telegram that contains the HART data provided by the HART sensors.

11 CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

- The test setup system presents an integration approach that enables access to HART field devices through communication components that interact with the devices and feature connection between the different integration levels described in chapter 6.3.

- The integration of HART field device's data to the bus level is achieved using the Remote I/O modules. The transmission of HART telegrams within the system bus is accomplished via the Profibus DP-V1 Read and Write services supported by the Profibus DP devices LB8106 Com Unit and xEPI2. The latter provides bus access for PACTware, TH OPC Server DP and Emerson AMS Suite via Ethernet TCP/IP.

- The PACTware software application handles HART communication utilising FDT/DTM technology. The corresponding Communication and Device DTMs are integrated in the FDT frame application following a hierarchical structure based on the integration approach (ch. 8). Consequently, the HART sensor's DTMs provide an efficient mechanism to transfer configuration, parameterization and diagnostic data enhanced on the field devices by the HART protocol.

- The AMS Intelligent Device Manager enables the integration of HART field devices and data through the standardized HART on Profibus profile, which due to its HART client system structure requires direct access to the Profibus network and Profibus devices (master class 2 and slave) that support DP-V1 Read and Write services. These requirements are met by the components employed in the integration approach presented in chapter 8. As a result, configuration, diagnostics and monitoring functionalities are available via the AMS Device Manager providing an effective asset management of HART field devices.

- The PACTware and AMS Device Manager software in conjunction with the introduced integration approach contributes to solve the challenges related to remote access to HART devices and integration of HART diagnostics data.

- The HART signal superimposed on the analog 4-20mA signal provides supplementary process measurements as well as device status, configuration options and diagnostics information of HART field devices. However, the response time of a system using HART protocol as first communication protocol is quite slow as seen on the results of the integration test with PACTware.

- The use of 1.5 Mbit/s as Profibus network transmission speed throughout the integration tests makes it possible to transmit the HART data and bring it to the examined softwares on a very high rate as indicated on chapter 6.4.2.4. In this way the allotted time to incorporate the HART data into the Profibus DP telegrams is very short. Consequently, the HART requests and responses are not fitted within a single Profibus telegram cycle which, in the worst case, may cause lost of data if the HART data link timers are exceeded; according to the HART communication modes and timing rules (chapter 3.9 and 3.18).

- The results of the integration test with the TH OPC Server DP demonstrate the software capability to integrate and interact with the Profibus devices. Furthermore, the OPC Server DP meets the expected functionalities regarding access to the OPC Server via OPC clients using standardized methods (described in chapter 6.5.4).

- Since the OPC specifications use the tag to connect to and access field device data, proper setup of OPC Tag properties in the OPC Server DP Configurator is required. The received values which are displayed on the OPC Client reveals that more specialized knowledge regarding Profibus telegrams is needed in order to accomplish access to the HART data contained in the Profibus data field. In this context, further research could be conducted in order to find better methods to expose HART data via OPC standards.

- The integration approach presented in this thesis leaves uncovered the integration of HART field devices using HART multiplexers. This and other software applications to integrate HART data should be taken in consideration further investigations.

12 BIBLIOGRAPHY

1. Bowden, Romilly, 2007. *"HART Field Communications Protocol - A technical Overview"*. HCF_LIT-20 Rev. 3.0, HART Communication Foundation.

2. HART Communication Foundation, 2010. *"HART Communication - Application Guide"*. HCF_LIT-039 Rev. 1.0

3. Zhang, Peng, 2008. "*Industrial Control Technology: A Handbook for Engineers and Researchers"*. William Andrew Inc., New York

4. SAMSON, 2005. "*SAMSON Technical Information - HART communications".* http://www.samson.de/pdf_en/l452en.pdf.

5. HCF (HART Communication Foundation), 2009. HCF-Main pages. http://www.hartcomm.org/

6. http://en.wikipedia.org/wiki/OSI_model

7. http://www.analogservices.com/

8. Kingestad Bjørn, 2005. "*HART - grensesnitt til feltutstyr"*. Universitetet i Stavanger.

9. Boyes, Walt, 2003. "*Instrumentation Reference Book*", 3rd Ed., Butterworth-Heinemmann Publications, Massachusetts.

10. Reynders Deon, Mackay Steve, Wright Edwin, 2004. *"Practical Industrial Data Networks - Design, Installation and Troubleshooting"*. Newnes Publications, Oxford.

11. Weigmann Josef, Kilian Gerhard, 2003. *"Decentralization with PROFIBUS DP/DPV1 - Structure, configuration and use of Profibus DP with SIMATIC S7".* Publicis Corporate Publishing, Erlangen

12. PROFIBUS Nutzerorganisation e.V. (PNO), 2010. *"PROFIBUS System Description - Technology and Application"*, http://www.profibus.com/nc/downloads/downloads/profibus-technology-and-application-system-description/display/

13. Diedrich Christian, Bangemann Thomas, 2007. *"Profibus PA - Instrumentation Technology for the Process Industry"*, Oldenborg Industriverlag GmbH, Munich

14. Reynders Deon, Mackay Steve, Wright Edwin, 2005. *"Practical Industrial Data Communications - Best Practice Techniques"*, Newnes Publications, Oxford,

15. SAMSON, 2005. *"SAMSON Technical Information - PROFIBUS-PA"*, http://www.samson.de/pdf_en/l453en.pdf.

16. Mathivanan N., 2007. *"PC-based instrumentation - Concepts and Practice"*, Prentice-Hall of India.

17. Pepperl+Fuchs, 2011. *"Manual PROFIBUS COM UNIT - Easycom LB8106/FB8106"*, http://files.pepperl-fuchs.com/selector_files/navi/productInfo/doct/tdoct1222b_eng.pdf

18. Nitaigour Premchand Mahalik, 2003. *"Fieldbus Technology - Industrial Network Standards for Real-Time Distributed Control"*, Springer-Verlag Berlin Heidelberg.

19. Miller Philip, 2009. *"TCP/IP - The Ultimate Protocol Guide: Volume 1 - Data Delivery and Routing"*, Brown Walked Press, Florida.

20. Pigan Raimond, Metter Mark, 2008. *"Automating with PROFINET - Industrial Communication Based on Industrial Ethernet"*, 2nd Edition, Siemens Aktiengesellschaft, Berlin and Munich

21. Pepperl+Fuchs, 2009. *"Interface Techonology Engineer's Guide - Intrinsic Safety, Surge Protection, HART Interface Solutions, Signal Conditioning"*, http://files.pepperl-fuchs.com/selector_files/navi/productInfo/doct/tdoct1551b_eng.pdf

22. Pepperl+Fuchs, 2009. *"Technical White Paper - HART via Remote I/O"*, http://pepperl-fuchs.com/global/en/index.htm

23. Pepperl+Fuchs, 2009. *"Manual LB REMOTE I/O SYSTEM HARDWARE"*, http://files.pepperl-fuchs.com/selector_files/navi/productInfo/doct/tdoct1130f_eng.pdf

24. Pepperl+Fuchs, 2007. *"Bus-Systems - LB Remote I/O, Div 2; FB Remote I/O, Zone 1; Profibus, Modbus, Fieldbus"*, http://pepperl-fuchs.com/global/en/index.htm

25. PROFIBUS Nutzerorganisation e.V. (PNO), 2006. *"Profibus Profile – HART"*, http://www.profibus.com/nc/downloads/downloads/profile-for-hart-on-profibus/display/

26. PROFIBUS Nutzerorganisation e.V. (PNO), 2006. *"Application Guideline - Profile for HART on Profibus"*, http://www.profibus.com/nc/downloads/downloads/profile-for-hart-on-profibus-application-guideline/display/

27. Simon Rene, 2007. *"Field Device Tool – FDT"*, Oldenborg Industriverlag GmbH, Munich (Germany)

28. http://www.fdtgroup.org/

29. Automatisierungs-technische Praxis, 2006, http://pepperl-fuchs.com/global/en/classid_259.htm?view=productgroupliterature

30. Hollender Martin, 2009. *"Collaborative Process Automation Systems"*, ISA - International Society of Automation, North Carolina.

31. Mahnke Wolfgang, Leitner Stefan-Helmut, Damm Matthias, 2009. *"OPC Unified Architecture"*, Springer-Verlag Berlin Heidelberg.

32. VEGA Grieshaber KG. Vegawell 52 Product Information, http://www.vega.com/en/For%20gauge%20measurements%20in%20water,%20waste%20water,%20deep%20wells%20and%20shipbuilding%20industry.htm

33. VEGA Grieshaber KG. Vegaflex 61 Product Information, http://www.vega.com/en/Level_measurement_TDR_GWR_VEGAFLEX61.htm

34. VEGA Grieshaber KG. Vegason 61 Product Information, http://www.vega.com/en/Level_measurement_ultrasonic_VEGASON61.htm

35. Pepperl+Fuchs, 2010. *"LB 8106 Data Sheet"*. http://files.pepperl-fuchs.com/selector_files/navi/productInfo/edb/t35026_eng.pdf

36. Siemens, *"Basics of PLCs"*, http://www3.sea.siemens.com/step/downloads.html

37. Siemens, *"SIMATIC S7-300 CPU31xC and CPU 31x: Technical Specifications"*, https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=12996906&caller=view

38. PROFIBUS Nutzerorganisation e.V. (PNO), 2006. *"PROFIBUS guideline - Profibus RS 485-IS User and Installation Guideline"*, http://www.profibus.com/nc/downloads/downloads/profibus-rs485-is-user-and-installation-guideline/display/

39. Siemens, *"CP 5711 Operating Instructions"*, https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objID=26586923&subtype=133300

40. Siemens, *"Manual Part B CP 343-1 Lean"*, https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK7343-1CX10-0XE0&objaction=csviewmlfbbeitraege&subtype=133300&caller=view

41. Trebing Himstedt, 2008. *"xEPI2 - Diagnostic Unit and Configuration Interface"*, http://www.t-h.de/fileadmin/inhalt/IC/produkte/xEPI_2/Data_sheet_xEPI_2_e.pdf

42. Siemens, 2010. *"Simatic controller Software, Tools for configuring and programming SIMATIC Controllers"*, https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-industrial-software_en.pdf?HTTPS=REDIR

43. http://www.pactware.com/

44. Emerson Process Management, *"AMS Suite: Intelligent Device Manager"*, http://www2.emersonprocess.com/en-US/brands/amssuite/amsdevicemanager/Pages/AMSDeviceManager.aspx

45. Trebing Himstedt, *"TH OPC Server DP"*, http://www.t-h.de/en/industrial-communication/products/th-opc-server-dp.html

46. Matrikon OPC Explorer, http://www.matrikonopc.com/products/opc-desktop-tools/opc-explorer.aspx

13 GLOSSARY

ASCII

ASCII (American Standard Code for Information Interchange) represents the alphabet (upper and lower case), numbers 0 to 9, and common punctuation characters, as 7-bit binary codes.

ASIC

The Profibus ASICs (ASIC - Application-Specific Integrated Circuit) facilitate the connection of third-party components and systems to the PROFIBUS fieldbus. Quick response times for the Profibus DP, which are required for transmission rates of up to 12 Mbit/s, can only be achieved by using these ASICs. Various ASICs are available for different functional needs and applications.

Asynchronous transmission

In “asynchronous” (without a clock) transmission, timing is defined by starting each character with a start bit (always 0) and following the character by a stop bit (always 1). Within a character, the bit timing is then defined by the baud rate.

The FSK HART protocol specifies a 1200-baud transmission rate with only two distinct values for each symbol (frequencies of 1200 or 2200 Hz); thus, each symbol represents only one data bit, and the data rate is 1200 bits per second (bps), the same as the baud rate.

Baud rate

The baud rate of a communication channel is the number of data symbols transmitted each second. Some systems code more than one data bit into each symbol (often by combining phase and amplitude modulation), so as to provide more possible values for each symbol and, therefore, a higher bit rate for the same baud rate.

Bell 202

Bell 202 is a U.S. standard, originated by AT&T (the Bell Telephone Company). It uses 1200 Hz and 2200 Hz as 1 and 0 respectively, at 1200 baud. Bell 202 is a full duplex communication standard, using a different pair of frequencies for its reserve channel.

HART uses Bell 202 signal frequencies, bus is a half-duplex system, so the reserve channel frequencies are not used. (The HART signal has other specification which are not derived from Bell 202)

Distributed Control System

Instrumentation (input/output devices, control devices, and operator interface devices) that permits transmission of control, measurement, and operating information to and from user-specified locations, connected by a communication link

Exclusive Or

"Exclusive Or" (XOR) is the logical combination function of two logical (0 or 1) values, such that the result is true (1) if one or other of the values is true, but not both.

Field device

The term is generally used to mean a measuring instrument ("transmitter") or a control device. This device is usually the slave in the HART master-slave relationship.

Filter

An electrical circuit (or software implementation) which removes signal components with frequencies above (low-pass filter) or below (high-pass filter) a specified cut-off frequency.

Floating point

Floating point represents a number in two parts: an exponent E and a mantissa M. The number represented is $MH2^E$ (M times 2 to the power of E). This allows a uniform proportional precision over a wide numerical range.

Frequency-shift keying

Frequency-shift keying (FSK) is a method of modulating digital information for transmission over paths with poor propagation characteristics. Two different frequencies are used to represent 0 and 1, usually in the audio frequency range (300 to 3000 Hz). Such a signal can be transmitted successfully over telephone systems. An FSK signal can also be modulated on to a radio carrier, or, as in HART, onto a DC current or voltage.

Galvanic isolation

Galvanic isolation implies a complete electrical insulation between two items. The term is commonly applied to electrical wiring free of any earth connection, such as the electronics of a transmitter, or to the two sides of an intrinsic safety isolator.

Hamming distance

Using geometric interpretation of error control coding, the number of bits that differ between two binary vectors *x* and *y* is called the Hamming distance.

Handheld communicator

A portable tool used to communicate with field devices, for commissioning or testing. For HART, a handheld communicator makes use of Device Descriptions to ensure full access to all the device's capabilities. Intrinsically safe models are available for use in hazardous areas.

Hexadecimal (hex)

Hexadecimal (base 16) representation of numbers is commonly used to describe the value of a data byte. One hex digit takes values 0 to 15, written as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (Upper- or lower-case letters may be used). Four bits are expressed in this way by a single hex digit. An 8-bit byte can be expressed as two hex digits, e.g. hex 43 represents binary 01000011 (decimal 67), and hex FF represents binary 11111111 (decimal 255).

Intrinsic Safety Barrier

A network or device designed to limit the amount of energy available to the protected circuit in a hazardous location.

IEC

The International Electrotechnical Commission is the leading global organization that publishes consensus-based International Standards and manages conformity assessment systems for electric and electronic products, systems and services, collectively known as electrotechnology.

I/O system

A physically discrete sub-unit of a process control system dedicated to the collection of a number of measurements, and/or to the distribution of a number of output signals. Depending on its construction, it may have several layers of structure and addressing, such as “rack”, “card” and “channel”. HART provides commands to identify such a structure and to communicate through it to each connected field device.

Modem

A modem (“modulator/demodulator”) is a device which converts binary digital signals to and from an FSK form. A modem does not provide a data coding mechanism, only a conversion of the physical form of signal used.

OSI model

The Open Systems Interconnection (OSI) reference model is a defined way of structuring the specification and implementation of a communication protocol into “layers”, each of which has a specific function. It originated from the International Standards Organization (ISO).

There is no implication that different “OSI model conformant” protocols will be able to inter-communicate directly. However, the implementation of gateways translating between different protocols should be easier than for non-OSI protocols.

Protocol

A communication protocol is a set of rules to be used in generating or receiving a message. It may include specifications for transaction rules (master-slave relationship, acknowledgement, timeouts, error-recovery), message structure (start character, addressing, data formats, error

checking), coding (text and numeric data formats), and physical signal characteristics (modulation techniques, signal type, signal level, transmission medium).

Re-ranging

The act of setting new lower and/or upper range values for a field device. Not to be confused with “calibration”.

Sensor

The measuring device that is connected to, or embedded in, a field measuring device (transmitter). Also referred to as a transducer.

Throughput and latency

Throughput indicates the maximum number of transactions per second that can be communicated by the system.

Latency measures the worst-case maximum time between the start of a transaction and the completion of that transaction.

UART

A UART (Universal Asynchronous Receiver Transmitter) provides the electronics needed to convert a byte of data (usually presented by the processor in parallel form) to and from serial form, and to add or remove the start, parity and stop bits. It may take the form of an integrated circuit chip, or may be embedded in a microprocessor chip. A typical UART can be configured to use 7- or 8-bit data, odd, even or no parity, and any standard baud rate. At the receiving end, the UART checks parity and the character frame format, and reports any errors to its controlling processor.

For FSK HART, the UARTs are set for 8-bit data, odd parity and 1200 baud.

Variable

In the mathematical sense (and in HART), a “variable” is any item of data which can take various values. This has nothing to do with data type: text strings are just as much variables as are numeric quantities. Nor does it relate to whether the value varies often, or only when “configured”.

14 APPENDIX

The information presented under this chapter has been taken from [10].

Appendix 1 – Major HART revisions

Revision	Date introduced	Features
2	1986	Fist public specification. Commands #0 to #6, #33 to #48
3	1987	New command #49
4	1988	Improved support for multiple variables. Write-protect status. Optional type-code expansion. New commands #50 to #56
5.0	1989	Long frame format, unique identifier. Burst mode. Block commands #4 and #5 replaced by new commands #12 to #18. New commands #11 to #19, #57 to #59, #108 to #112. Improved data link and physical layer specifications revisions 7.0.
5.1	1990	Support to multiple analog outputs and non-current analog outputs. New commands #60 to #70, #107.
5.2	1993	Physical layer specification revision 7.2.
5.6	xxx	Physical layer specification revision 8.0.
6.0	2001	Long tag (32 characters). Better support for multivariable devices and actuator. More device and variable status information. Device families. Block data transfer. New commands #7, #8, #20 to #22, #71 to #75, #79 to #83, #106, #111, #112, #113.

Appendix 2

Message type (from start character and master address bit)	Token (permission to transmit) goes to...	
	No bursting slave	Bursting slave present
Primary master to slave	Slave	Slave
Slave to primary master	Secondary master	Bursting slave
Secondary master to slave	Slave	Slave
Slave to secondary master	Primary master	Bursting slave
Bursting slave to primary master		Secondary master
Bursting slave to secondary master		Primary master

Appendix 3

Commands	Function
0, 11, 21	Read unique identifier (device manufacturer, device type, revision levels)
1, 2, 3	Read measured values
6	Set polling address
7	Read loop configuration (HART rev. 6)
8	Read dynamic variable families (HART rev. 6)
9	Read device variables with status
12, 13, 17, 18	Read and write user-entered text information (tag, descriptor, date, message)
14, 15	Read device information (transducer serial number, transducer limits, alarm operation, range values, transfer function, damping time constant)
16, 19	Read and write final assembly number
20, 22	Read and write long tag (HART rev. 6)
31	Indicates a 16-bit extended command in the data field

Appendix 4 - Example of data structure for HART Command 9:

HART Command 9: Read up to four device variables with status (6.0)

Master Device – Data in command (4 bytes):

Byte 0 dev. var. code for slot 0
Byte 1 dev. var. code for slot 1
Byte 2 dev. var. code for slot 2
Byte 3 dev. var. code for slot 3

(Truncated after last requested code)

Slave Device – Data in reply (33 bytes):

Byte 0		extended field device status	(B)
Byte 1	slot 0:	dev. var. code	
Byte 2	slot 0:	dev. var. classification code	
Byte 3	slot 0:	dev. var. units code	
Byte 4-7	slot 0:	dev. var. value	(F)
Byte 8	slot 0:	dev. var. status	
Byte 9-16	slot 1:	as bytes 1-8 above	
Byte 17-23	slot 2:	as bytes 1-8 above	
Byte 14-32	slot 3:	as bytes 1-8 above	

(Truncated after last requested code)

dev. Var. = device variable
B = bit-mapped
F = floating point

Appendix 5

Commands	Function
33, 61, 110	Read measured variables
34-37, 44, 47	Set operating parameters (range, damping time, PV units, transfer function)
38	Reset “configuration changed” flag
39	EEPROM control
40-42	Diagnostic functions (fixed current mode, self test, reset)
43, 45-46	Analog input/output trim
48	Read additional device status
49	Write transducer serial number
50-56	Use of device variables
57-58	Unit information (tag, descriptor, date)
59	Write number of preambles required
60, 62-70	Use of multiple analog outputs
71-75	Device, sub-device and I/O commands (HART rev. 6)
79	Write device variable (force fixed value) (HART rev. 6)
80-83	Device variable commands (HART rev. 6)
106	Flush delayed responses (HART rev. 6)
105, 107-109	Burst mode control
111, 112	Transfer service commands (HART rev. 6)
113	Catch device variable

*EEPROM = Electrically-erasable programmable read-only memory, used in early HART devices to store configuration information.

Appendix 6 - Example of data structure for HART Command 35 and 109

HART Command 35: Write range values

Master Device – Data in command (9 bytes):

Byte 0 range units code
Byte 1-4 upper range value (F)
Byte 1-4 lower range value (F)

Slave Device – Data in reply:

As in command

HART Command 109: Burst mode control

Master Device – Data in command (1 byte):

Byte 0 burst mode control code
(0 = exit, 1 = enter)

Slave Device – Data in reply:

As in command

F = floating point

Appendix 7 - Device Specific summary table

Command function
Write / Read low-flow cutoff valve
Start, stop or clear totalizer
Write / Read density calibration factor
Choose PV: mass, flow or density
Write / Read materials of construction
Write / Read transducer type
Valve characterization
Write / Read valve set point
Write / Read travel limits
Read local display information
Write PID set point
Trim transducer calibration

Appendix 8 - Device family commands

Command number (decimal)	Command number (hex)	Function
1024	0400	Read temperature status
1025	0401	Read temperature configuration
1026	0402	Read thermocouple configuration
1152	0480	Write temperature probe type
1153	0481	Write temperature standard
1154	0482	Write thermocouple probe connection

Appendix 9 – Device Information

Byte	Content
0	Hex FE (254), indicating expanded device type code (rev. 4 or later)
1	Manufacturer identification code
2	Device type code
3	Number of preambles required
4	Universal command revision (same has HART major rev. nr.)
5	Device specific command revision
6	Software revision
7	Hardware rev. (5 bits) and physical signaling code (3 bits)
8	Device function flags
9-11	Device ID number
12*	Minimum number of preambles in response from this device
13*	Maximum number of device variables
14-15*	Configuration change counter
16*	Extended field device status

* Introduced in HART 6

Appendix 10 - SAP's used by Profibus DP

SAP	SERVICE
Default SAP=0	Cyclical Data Exchange (Write_Read_Data)
SAP54	*Master-to-Master SAP (M-M Communication)*
SAP55	Change Station Address (Set_Slave_Add)
SAP56	Read Inputs (Rd_Inp)
SAP57	Read Outputs (Rd_Outp)
SAP58	Control Commands to a DP Slave (Global_Control)
SAP59	Read Configuration Data (Get_Cfg)
SAP60	Read Diagnostic Data (Slave_Diagnosis)
SAP61	Send Parameterization Data (Set_Prm)
SAP62	Check Configuration Data (Chk_Cfg)