Expertise
and insight
for the future

Ville Paasovaara

Applications of HART-Protocol for Diagnosing Smart Instruments

Metropolia University of Applied Sciences

Bachelor of Engineering

Electrical and Automation Engineering

Bachelor's Thesis

9 April 2020

Author Title Number of Pages Date	Ville Paasovaara Applications of HART-Protocol for Diagnosing Smart Instruments 35 pages + 2 appendices 9 April 2020
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Technology
Instructors	Juha Karttunen, Senior Automation Specialist Reijo Leinonen, Senior Lecturer

This study explores how to enhance process plants' predictive maintenance and response time to device failures by utilizing the HART-protocol to design a method for displaying four condensed status flags recommended by NAMUR NE107. The aim is to display these status flags in real time for the plant operator and maintenance worker, instead of having to plug in a handheld HART communicator device and request the information when the device has already malfunctioned.

In the beginning of the work the objective was to clarify the structure and function of HART-protocol, how HART commands work and the structural and operational layer of smart instruments. Different industrial field device communication protocols are also explained, and their use case and functions are compared to HART protocol. HART communication was tested in Siemens PCS7 environment and results were employed to create a method for displaying the device status.

The result of the project is an established method of communication with HART devices using HART commands and a communication block that was coded in structured command language to fit everything in one block for convenience. The results are overall as expected before the study, and the resulting method can be used to read device status and the practice of using HART commands can be implemented to be used for different use cases.

Keywords	HART-protocol, smart instruments, device diagnostics

Tekijä Otsikko Sivumäärä Aika	Ville Paasovaara HART-väylän hyödyntäminen instrumentaation diagnostiikassa 35 sivua + 2 liitettä 9.4.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	Senior Automation Specialist Juha Karttunen lehtori Reijo Leinonen

Tämän työn tavoitteena oli tutkia, miten prosessiteollisuuden ennakoivaa huoltoa voidaan tehostaa käyttämällä HART-protokollaa NAMUR NE107-suosituksen määrittelemällä tavalla. Tarkoitus oli näyttää neljä määriteltyä diagnostiikkatilaa reaaliajassa tehtaan operaattoreille ja huoltohenkilöille sen sijaan, että huoltohenkilön pitäisi kiinnittää käsin käytettävä HART-kommunikaattori laitteeseen sen diagnosoimiseksi. Diagnostiikkatilan reaaliaikainen esittäminen antaa huoltohenkilökunnalle mahdollisuuden korjata laite häiriötilanteessa, ennen kuin se menee kokonaan rikki ja pahimmassa tapauksessa pysäyttää prosessin.

Työn alussa tarkasteltiin HART-protokollan rakenne ja toiminta, toisin sanoen miten HART-komennot toimivat ja millaisia ovat älykkäiden instrumenttien käyttötoiminnot. Myös erilaisten teollisuuden kenttälaiteprotokollien toiminta käytiin läpi ja niiden toimintaa verrattiin HART-protokollaan. HART-kommunikointi testattiin Siemens-PCS7 ympäristössä ja tuloksia hyödynnettiin rakentamalla niiden pohjalta tapa esittää laitteen tila.

Työn tuloksena luotiin menetelmä HART-yhteensopivien laitteiden kanssa kommunikointiin HART-komentoja käyttäen. Menetelmän käyttämisen helpottamiseksi kaikki toiminnot ohjelmoitiin SCL-kielellä yhden lohkon sisälle. Työn lopputulos on pitkälti sellainen, mihin pyrittiin ennen työn aloitusta, ja luotua menetelmää on haluttaessa mahdollista muokata eri komentoja ja käyttötarkoituksia varten.

Avainsanat	HART-protokolla, älykkäät kenttälaitteet, laitediagnostiikka

Contents

List of Abbreviations

CFC	Continuous function chart
DCS	Distributed control system
DTM	Device type manager
FSK	Frequency shift keying
HART	Highway addressable remote transducer
OSI	Open systems interconnection
PDM	Process device manager
PLC	Programmable logic controller
SCL	Structured control language

1 Introduction

The goal of this work was to find out how HART-protocol can be utilized to diagnose smart field devices. The purpose was to develop a practice to show digital information transmitted through HART in real time instead of asking the information when the device has already malfunctioned. This real time digital information is categorized in four different alarms based on NAMUR NE107 standard. These four alarms are *Failed*, *Out of specification*, *Maintenance required* and *Function check*. Categorizing the alarms makes it easier for plant operators and maintenance workers to diagnose devices and plan device maintenances before they break.

First part of the work includes finding out the physical- and operational structure of HART-protocol, introducing the universal and common-practice HART commands and learning their functions. Second part is the testing and implementation phase, where HART commands are tested on HART compatible transmitter. Based on the results, a program is made in structured command language where HART communication functions, like write and read, can be configured in single function block.

In the experiment a pressure transmitter is configured in Siemens PCS7 software. Communications with the transmitters are established in Simatic PDM. HART modem is used to simulate status warnings to the device. HART command #48 is used to get additional device status from the device and information to be displayed is selected according to the NAMUR NE107 recommendation. A program is made to write and read HART commands and to display status flags. Instrument used in the test and documentation is Endress+Hauser Cerabar M PMP51 pressure transmitter.

2 Outotec Oyj

Outotec was founded in Northern Karelia in 1908, where first signs of Outokumpu mine were discovered by dredge operator who found an unusual meteorite-like stone. Supervisor of the site later sent sample of the stone to Geological Commission in Helsinki. Later, after further analysis of the site, Outokumpu mine was founded in 1910. [1, p. 51.]

In early 1930s Outokumpu had become Europe’s fifths largest copper producer thanks to big deposits found in Pechenga [1, p. 71].

The metals industry had expanded rapidly in late 1800's all the way to beginning of 1900's thanks to new innovations like electrochemical smelting and growing production rate of electricity. In the beginning of 20th century rapidly growing need for electricity was slowing down development of metal processing. Even though the rate of electricity production expanded steadily it was not enough to meet the ever-growing demands of metal processing industry. Canada and the Soviet Union tried to solve this problem of high electricity demand by using the combustion heat of the iron and sulfide contained in concentrate to produce heat for the smelting process, also known as autogenous smelting, but their results proved in vain. It was not until 1949 that Outokumpu succeeded in developing a method for autogenous smelting that is known today as flash smelting. Developing and scaling flash smelting for business was now one of the most important priorities for Outokumpu, but the laboratory at Outokumpu did not meet these demands so a new research center was founded in Pori that later became an integral part of Outotec Oyj as did the licensing sales for the technology. [1, p. 44.]

The beginning of 21th century saw some big acquisitions by Outokumpu.
Most notably its long rival Lurgi Metallurgie and its Frankfurt headquarters, which later became part of Outotec Oyj. Other Lurgi group companies kept rights to all general patents of the company and Outokumpu Technology got rights to metallurgical patents. [1, p. 219.]

In the year 2006, Outokumpu technology separated from its former parent company Outokumpu and listed on the Helsinki stock exchange. Later in the year 2007, it was named Outotec Oyj to further separate it from its old parent company [1, p. 255]. Today there are over 4000 employees working for Outotec Oyj in 36 different countries. Sales for Outotec Oyj were 1,210 million euros in 2019. Operations are separated in three different categories: mineral processing, Metals, Energy & Water and services. [2.]

3 Smart Field Devices and Device Diagnostics

The first three industrial revolutions are regarded for harnessing steam and water power for production, mass production using electricity and rise of personal computers. Development of smart field devices that started in the 1980's, when integrating microprocessors in field devices became the standard industry practice is often thought of as the fourth industrial revolution or Industry 4.0. Industry 4.0 can offer many benefits to processing plants and it is often categorized into six different trends as seen in figure 1. [3.]

Figure 1. Trends for Smart Transmitters for Industry 4.0 Reprinted from [3]

As seen in figure 1, most of the industry 4.0 trends have something to do with increasing efficiency or lowering the cost of production. Increasing the safety usually increases efficiency too. However, it has more important benefits as well, such as improving the working conditions of plants and preventing work time injuries. Safety instruments in processing plant make up a safety instrument system or SIS. Having smart instruments as SIS instruments offers many benefits over generic instruments as seen in table 1. Diagnostics can indicate if safety devices are not operating inside their intended operating range. This allows operators some room to compensate instead of waiting for the instrument to fail and having to shut the operation down for maintenance. Smart instruments have a lot of data in them, such as sensor temperature and condition of processing circuits, which allows for faster troubleshooting of the device. [4.]

Table 1. Benefits of using smart field devices in safety systems Reprinted from [4]

1. Diagnostics can indicate out-of-spec instrument operation.
2. Diagnostics can indicate failure of communication links.
3. Diagnostics can predict incipient failure.
4. Smart instruments can ease the task of redundant system design.
5. Process data, in addition to the process variable, can improve safety system performance.
6. They have the ability to automate some testing protocols, such as PST for ESD valves.
7. Safety instruments can feed information to the BPCS in appropriate situations.

As seen in table 1 reason number six, another benefit of having smart instruments in safety system is the ability to perform reliability function tests for emergency shut down valves using automated partial stroke testing. This can greatly reduce the potential of fail on demand. [5.]

3.1 Electromagnetic Flowmeter

Electromagnetic flow measurement makes use of Faraday's law of induction that was discovered in 1831 by Michael Faraday. Outside of the metering tube, two coils are placed on the opposite sides. They produce a magnetic field inside the tube, which results in a potential difference proportional to the flow velocity perpendicular to the flux lines. Electrodes are placed on opposite sides of the tube in 90° angle to the coils and when conductive liquid flows in the tube it creates a voltage that is detected by the electrodes, as seen in figure 2. [6.]

Formula for calculating the flow [7]: $E = \mathrm{k} * \mathrm{B} * \mathrm{D} * \mathrm{V}$

- E = Induced Voltage
- k = proportionality Constant
- B = Magnetic Field Strength
- D = Distance Between Electrodes
- V = Velocity of Process Fluid

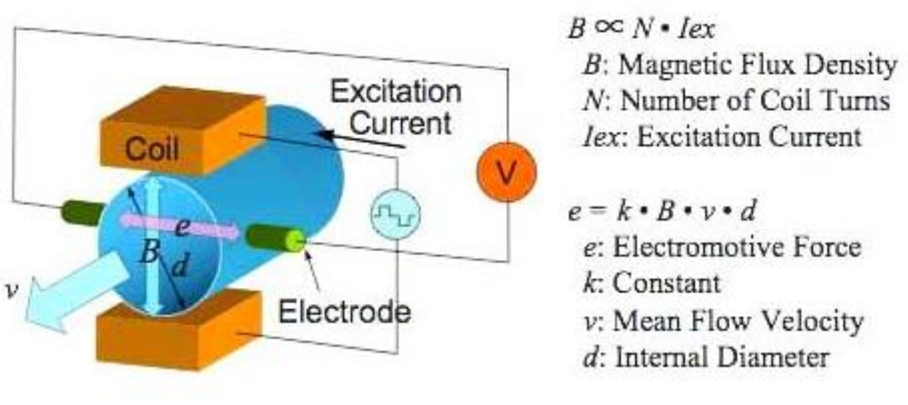


Figure 2. Working Principle of Electromagnetic Flow Meter Reprinted from [6]

Measuring flow is a very important practice in mineral processing. Because there is such wide variety in different measured substances, it is paramount to find reliable and effective way to measure those different materials. Different materials may include effluent, slurry, reagents and instrument air. Electromagnetic flow measurement is a good solution for materials like mineral slurry, because it has little obstruction of flow and abrasive properties of the slurry has little effect to the electrodes that are exposed inside the tube, sometimes liners made of polyurethane or neoprene are added for additional resistance. Also, effluent flow is commonly measured using magnetic flowmeter, because it is reliable and is not affected by the viscosity or pressure variations. [8.] However, since electromagnetic flowmeter is a closed-pipe system it is possible for debris to accumulate in the measuring tube, that is why regular cleaning routine is important [9].

3.1.1 Diagnosis of Electromagnetic Flowmeter

Inappropriate grounding is one of the main cause of issues with electromagnetic flow meters, even more so if pipeline is made from nonconductive material, like plastic. This can cause the device to experience significant measuring errors. Five to six mV of potential near the measuring device is enough to render the signal practically useless. Pipelines made of conductive material, like stainless steel, don't typically experience any grounding issues. However, pipe flanges can be wired to general ground to make sure the grounding is enough. Some of the electromagnetic flow meters have spectral analyzer built-in the device to detect too high voltage readings, indicating of faulty grounding.

High process noise is caused by eroding of the electrodes inside the pipe and/or process material sticking to the surface of electrodes, causing the exchange of ions between process fluid and electrodes to be obstructed. High process noise problems are especially problematic in mineral processing industry where slurries of different types might create noisy signal, which is usually mitigated by having dead time in control loop. Dead time in control loop prevents unwanted actuation of flow control valves and so expanding the lifetime of the valve. Typical way of combatting signal noise is to record noise levels with new electrodes and monitor the coating of electrodes and service them at a set point. Having diagnostic capabilities to monitor the coating is a big benefit for the plant.

Most of the modern electromagnetic flow meters can detect when the pipe is empty of any conductive material. This is a very important feature especially in operations that handle different type of slurries, that have low viscosity since these materials need internal friction to prevent solidification. When operations are stopped, the pipes must be flushed with water, so the slurry does not turn into solid material and block the pipe. Empty pipe detection can detect that the pipe is properly emptied after the flushing. Another reason to have empty pipe detection in any process operation is to prevent the pumps from operating when they are empty. An empty pipe can be detected by measuring resistance of the electrodes. When the pipe is filled electrodes are connected via the process liquid. Resistance between electrodes is a factor of conductivity and when the level of process liquid change in the pipe so does the resistance. This method is cheap, because it does not require any additional hardware, but it does not work very well in vertical pipelines where coating is a problem. Resistance measuring can be supplemented with additional electrode to measure partial filling of the pipe, this method is called full pipe electrode. [10.]

3.2 Resistance Temperature Detector

Resistance Temperature Detector or RTD is a temperature-measuring device that uses resistance of an electrical conductor like, Platinum or Copper, for temperature measurement as seen in figure 3. Metals have positive temperature coefficient, which means that their resistance increases as the temperature increases. Materials with negative temperature coefficient like carbon, silicon and germanium, however, experience inverse resistance gain with temperature. [11.]

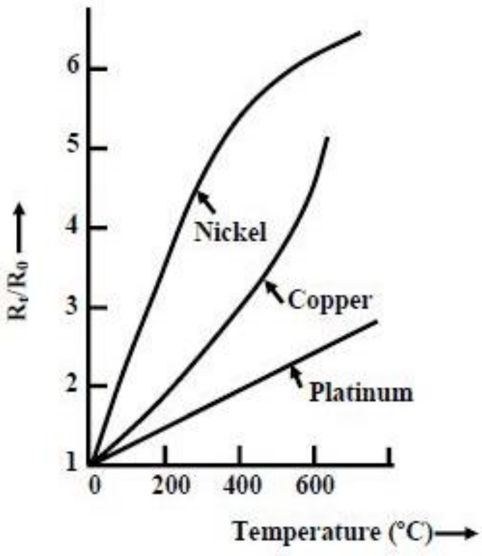


Figure 3. Temperature Coefficient of Resistance for Different Metals Reprinted from [11]

The simplest configuration of RTD is called two wire circuit where the RTD is connected to a bridge circuit and constant electrical current is applied to the circuit. As the temperature changes the resulting voltage change across the bridge is observed to determine the temperature. In two wire circuit the resistance of extension wire affects the temperature reading. This can be negated by three or four wire circuit where additional wire is used to calculate and negate the effect of the extension wire. [11.]

RTD is used when high accuracy is needed. However, RTD has some drawbacks compared to sensors that use a different method like thermocouple. RTD has narrower measuring range than thermocouple, usually ranging from 0° to 600° degrees Celsius and it is also more expensive. RTD is however still commonly used in mineral processing because high temperature operations are usually not required, and accurate temperature readings are usually high priority. [12.]

3.2.1 Diagnosis of Resistance Temperature Detector

Common issue with resistance temperature detectors is thermal lag, which is caused by delayed response to temperature change. It can take minutes for the temperature sensor to reach the correct value. This can be compensated with mathematical model in the processor of the device. [13.]

In order to measure temperature, current must be applied to the sensing resistor of the device. This current can cause the sensor to self-heat. In modern sensors this is likely not causing any major issues. [13.]

Most common issue, however, is caused by immersion error, where the RTD is placed in the process to measure temperature. Some of that temperature will be conducted away from the actual measuring point by the body or the protective casing of the sensor. Mathematical formulas have been designed to calculate this immersion error, however the most reliable method to negate this error is to place the measuring part deep enough in the process to maximize thermal contact area . The depth depends on the temperature of the process and the outside atmosphere. [13.]

3.3 Structure of Smart Field Devices

Smart measurement devices have an analog sensor as their primary method of measuring the process variable. This sensor sends 4–20mA current loop to signal processing unit that converts the analog signal to digital signal trough analog-to-digital converter (ADC). AD7124, as seen in appendix 1, is an ADC with 4 to 8 different inputs and programmable power source to be able to accommodate different type of sensors including passive type. AD7124 has diagnostic capabilities including:

- Read/Write with no exceptions in valid registers
- Only valid data are read to the register
- Validation of clean decoupling of voltage regulator (LDO)
- Validation of the performance of ADC modulators and filter specifications
- Validation of overvoltage or undervoltage

After the analog signal is processed by ADC, it passes an isolator. The purpose of this is to prevent ground loop formation and protect the processing unit and programmable logic controller from overvoltage. [3.]

Microcontrollers that are used in field devices are often ARM based controllers, because of their small size and energy consumption. ARM is acronym for Advanced RISC Machine, which refers to the architecture of the device's processor chip. ARM architecture is also very commonly used in consumer smart devices such as personal smart phones. These microcontrollers allow the device hardware to be diagnosed and calibrated to make sure the device measurement is accurate. [3.]

Microcontroller passes the digital signal to digital-to-analog converter to convert the digital signal back to analog form before sending it to controller. One of the reasons why this must be done is to allow the microcontroller to perform its functions to the signal and remove possible errors from the signal. Microcontroller also passes the signal to HART modem that modifies the signal to HART format.

Once the diagnostic capabilities of the smart instrument are in use, the flood of diagnostic data can cause the plant operator to miss critical system alarms and messages, which is why there are software tools in the market to categorize the device information for the operators and only show them what they need to see in order to do their job. Some of the reasons for using such software are [3]:

- Providing visualization for different process conditions
- Monitoring historical trends and change over time
- Providing contingency plans to deal with safety incidents

3.4 NAMUR NE107 Recommendation

The digital diagnostics data of the device is usually difficult to read and requires a maintenance technician to troubleshoot the device. It does not benefit plant operators to see all the messages that the devices transmit, instead signals are categorized to make plant operators work easier and more efficient. Different devices have different diagnostic signals, for HART devices these diagnostic messages can be mapped into different NAMUR NE107 category using HART commands 523, 524 and 525. [14.]

NAMUR stands for User Association of Automation Technology in Process Industries and NE107 is their recommendation of what status information plant operators should see. NE107 categorizes all diagnostics information that smart devices transmit into four categories that represent the state of the device to make plant operators task of diagnosing the device easier. NE107 specified statuses are Failed, Out of Specification, Maintenance Required and Check Function as seen in figure 4. These statuses help the operator to know the severity of the issue and what should be done about the issue, for example if maintenance request is on the operator knows that the issue might be related to the life cycle of the instrument. Check function status indicates that the signal is temporarily frozen possibly due to ongoing work on the device. Status also let’s operators know how severe the issue is. [14.]

Failed	Out of Specification	Maintenance Required	Check Function
			
High severity: signal invalid due to malfunction in the device, sensor, or actuator	Medium severity: permissible ambient or process conditions exceeded or the measuring uncertainty of sensors or deviations from the set value in actuators is probably greater than expected	Low severity (advisory): although the signal is valid, the remaining life is nearly exhausted or a function will soon be restricted due to operational conditions e.g. aging of a pH-electrode.	Signal temporarily invalid (e.g. frozen) due to on-going work on the device.

Figure 4. NAMUR NE107 Specified Device Status Flags Reprinted from [14]

Devices can detect when the process value is not tracking the process accurately and send status message to the operator who will see that the device needs to be calibrated. Operators are also notified if the device is in manual mode or work is in progress with the device. After the operator has acknowledged the issue maintenance technicians will do the final check if that is required, to see that the device is functioning properly. [14.]

As seen in figure 5 operators and device specialists have different responsibilities regarding the process. Operators typically use supervisory and data acquisition system to see the main process variables and device status. Operators can also perform basic process related operations in the system. Device Specialists have access to the Asset Management systems. Asset Management Systems have complete register of device diagnostic data and error codes of the device. [14.]

Operator	User	Device Specialist
Control system operator workstation	System	Intelligent Device Management (IDM) software part of Asset Management System (AMS)
Safely operate a plant	Responsibility	Troubleshoot devices
No details: only essential summary NE107 status signals	Detail	As much detailed information as possible: Diagnostics with probable cause and recommended action
Communicated as soon as possible, real-time alarm	Reporting	Accessed at daily maintenance and turnaround planning

Figure 5. Responsibilities of plant operator and device specialist Reprinted from [14]

Responsibilities of operators and device specialists are also different. Operators need to make sure that the plant is functioning safely and efficiently, when devices malfunction operators can sometimes preform basic tasks on them that may require some basic knowledge of the device. Device specialists have access to all diagnostic data of the device and their task is to troubleshoot malfunctioned devices. [14.]

4 Process Automation Communication Protocols

Protocol means a method of representing, encoding and transmitting data. In the early days of process automation, process control was achieved mechanically. Pneumatic control was used to power the controllers. Air was compressed to 3-15 psi to determine the process variable. This system was replaced later in the 1950s by a more convenient current loop system that is still used today. Current loops of 4–20mA became the industrial standard for many decades, because they were easy to install and operate. They also offered a reliable way to transmit the process signal to controller device. Later, however the need to transmit more data through single cable and wireless communication gave rise to modern digital communication protocols. [15.]

4.1 HART-protocol

HART-protocol was born out of economical need to combine digital properties and 4–20mA current loop in one cable. It was originally developed by Emerson Electric Co. in the 1980s but was later made completely open and the rights were transferred to FieldCOMM Group in 2015. FieldCOMM Group is the result of joined assets of Fieldbus Foundation and HART Communication Foundation, its mission is to promote global standards for integrating digital devices to on-site systems and lead the development of a unified information model of process automation field devices. [16, p. 8-9.]

Using digital communication these smart instruments can provide more than one measurement, their health can be checked remotely through digital transmission and information about the device can be stored inside the device, such information may include: [16, p. 2.]

- Information about the device or “device label”
- Process related information
- Maintenance and calibration records

Over the years, the HART protocol has undergone several revisions to improve its capabilities and add new features without compromising its compatibility with older devices. Revision 4 introduced optional manufacturer ID, which became mandatory later in revision 5. With revision 6 actuators and other output devices’ electrical characteristics were defined, more status information was added to the devices and new physical layers were introduced for faster communication between the device and host(s). Latest revision is revision 7, which Introduced WirelessHART 2.4GHz radio physical layer and transmission of HART messages over internet protocol-based networks (HART-IP). [16, p. 25-27]

Most DCS systems today use the 4–20mA analog signal to read the process value and for real-time closed loop control, because of its faster response time. HART communication is used to access secondary variables, diagnostics and device status. Sometimes primary value is read with both digital and analog means to cross check the value to see the health of analog signal and that digital-to-analog conversion is working properly. [16, p. 15.]

HART uses a Frequency-Shift Keying (FSK) to superimpose digital communication on to the 4–20mA current loop, thus connecting the PLC to the field device. Two different frequencies of 1200 and 2200 are used to represent binary values, where the controller reads 1200hz wave as 1 and 2200hz as 0 as described in figure 6. These sine wave frequencies have an average value of zero, so the digital signal does not interfere with analog signal. Binary digits are transferred at a rate of 1200 bits per second both ways, which means that the slave device can send information to the master and vice versa. This frequency and transmission rate are based on Bell 202 telephone network standard; however, the amplitude of the HART signal is different than in Bell 202 standard. [16. Page 15.] In addition to FSK, HART can be used with other physical layer configurations. Most popular of these alternatives is RS-485 format, which abandons analog capabilities for faster digital communications reaching maximum speeds of 19.2 - 38.4 Kbits/s making it comparable to other all-digital protocols such as Foundation Fieldbus and Profibus PA. C8PSK (coherent 8-way phase shift keying) is another alternative to FSK offering faster communication with 9.6 Kbits/s and supporting around 10 transactions per second. HART devices using C8PSK will still use FSK in the event of communication problems. [16, p. 24.]

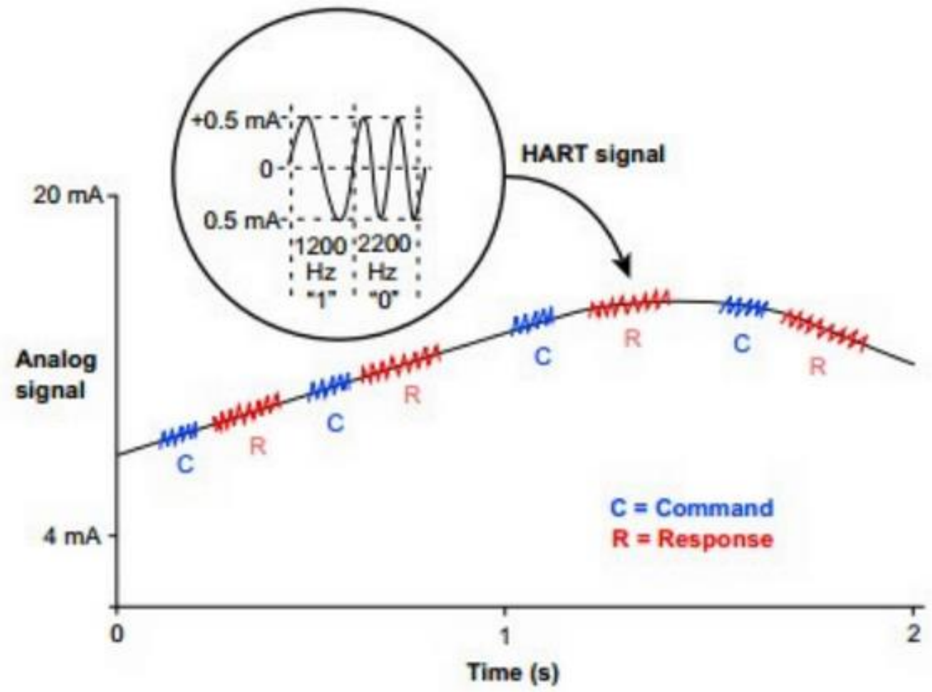


Figure 6. Frequency Shift Keying Reprinted from [17]

FSK HART implements layers 1, 2 and 7 of the Open Systems Interconnection (OSI) 7-layer protocol model as seen in table 2. In an environment where automatic retries for corrupted and lost messages are not used, layers 3 to 6 are not necessary. Excluding these layers, results in far more simplicity of HART compared to other protocols. WirelessHART, being a multi-device network, requires functionality from layers 3 to 6. [16, p. 9.]

Table 2. HART OSI layers Reprinted from [17]

OSI Layers	HART Layers
Application	HART commands
Presentation	
Session	
Transport	
Network	
Data link	HART protocol rules
Physical	Bell 202

Second layer is the data link layer, sometimes called HART telegram as seen in figure 7. Preamble is the first field in the message: its job is to initialize and synchronize receivers of connected devices. Delimiter is a single-byte field that marks the end of the preamble. Delimiter contains metadata of the message, such as the type of the message frame and type as seen in table 3:

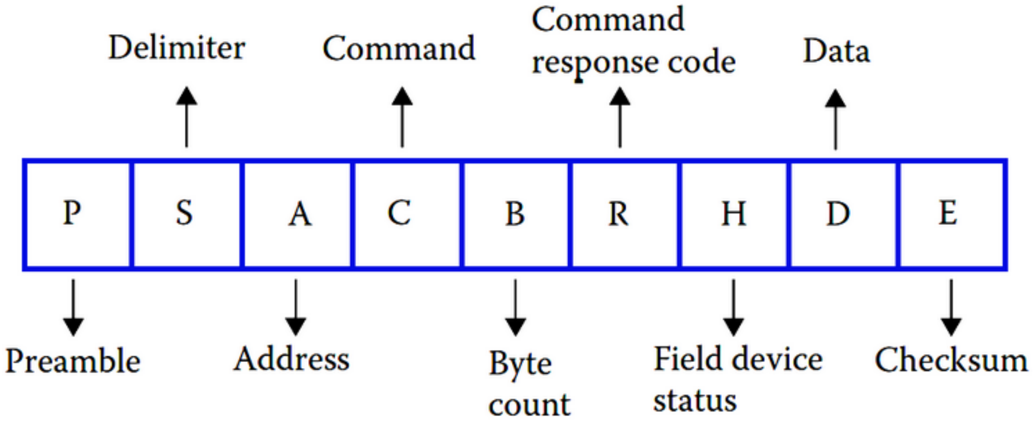


Figure 7. HART message structure Reprinted from [17]

Address field can be in short single byte format (short-frame format) or long 5-byte format (long-frame format) for unique ID address. Command field is the fourth field in the data link layer, that contains single byte integer in the range 0 to 253. Data link layer does not interpret HART commands, but displays the command set in the application layer. [17.] Byte count is single byte of length and it displays how many bytes are still left in the message, not including checksum. Command response and field device status are both single-byte fields. Command response counts any communication errors and field device status has information about the device. [16, p. 46.]

Table 3. Start Delimiters Reprinted from [16]

Message Type	Short Frame (Hex)	Long Frame (Hex)
Master to Slave	02	82
Slave to Master	06	86
Burst Message from Slave	01	81

HART 6 specifies commands with up to 33 bytes of data. However, some devices use more bytes in device specific command messages. If the device is not able to use the data issued in the command data field, it will issue a warning in the communication status field of the response. The number of data bytes is different for every command. Final byte is the checksum containing all the bytes that precede it in the message, starting with start character. This provides a further check on transmission integrity. [16, p. 50.]

Application layer of the OSI-model is where HART commands are defined. Communication routines are based on these command routines. Each field device is considered a slave device and can have two masters at a time. [17.] One master is considered a primary master and the other secondary. Usually the primary master is the DCS and the secondary master can be a hand-held communicator or HART modem. [16, p. 55.]

4.1.1 HART Commands

HART command byte contains an integer in the range 0 to 253 in hex format. Commands number 256 or above utilize an extended command where command 31 (hex 1F) is used in the command field to indicate that extended command is used. In this case first two data bytes are used to write the actual command in. HART commands are categorized in five different groups:

- Universal commands
- Common-practice commands
- Non-public commands
- Device Specific commands
- WirelessHART commands

Universal commands are implemented in all HART-conformant devices. Universal commands include general device functions that are meant to work for every HART device as seen in table 4. Since HART revision 7, commands #38 and #48 have been transferred from common-practice commands to universal commands. Most of the universal commands can be used without device description. Its good practice to start communications with command #0 to see how many preamble bytes the slave device requires. Universal command #48 is used in this work read additional device status of the device. [16, p. 67-69.]

Table 4. HART universal commands Modified from [16, p. 69]

commands	function
0, 11, 21	Read unique identifier
1, 2, 3	Read measured variables
6	set polling address
7	Read loop configuration
8	Read dynamic variable families
9	read device variables with status
12, 13, 17, 18	read and write user-entered text information
14, 15	read device information
16, 19	read and write final assembly number
20, 22	read and write long tag
31	Indicates a 16-bit extended command in the data field
38	reset configuration changed flag
48	read additional device status

Common-practice commands are in the range 32 to 121 and 512 to 767. Common-practice commands are commonly implemented in HART devices. However not all HART-conformant devices use common-practice commands. Common-practice commands like universal commands work for any type of device, however they usually have more specific purpose as seen in table 5. [16, p. 69.]

Command #523 as seen in table 5 can be used to read command #48 status bit mapping. Command #523 requires following data bytes in the command, byte0: index of starting status map, byte1: number of entries read. Status mapping codes are as following: 0=no effect, 1=maintenance required, 3=failure, 4=out of specification, 5=function check, 6=not defined. Command #534 is used to write new values to status bit mapping and command #525 to reset status bit mapping to its original state. [16, pp. 80, 173, 187.]

Table 5. Some of the common-practice HART commands Modified from [16, p. 71]

commands	function
33, 61, 110	Read measured variables
34-37, 44, 47	Set operating parameters
40-42	Diagnostic functions
43, 45-46	Analog input/output trim
49	Write transducer serial number
59	Write number of preambles
74-77, 84, 87	Device, sub-device and I/O commands
523-525	Condensed status commands
526, 527	Status simulation commands

Every HART command receives its response in standard response format. Standard response contains information on the device and two bytes of status known as “response code” as seen in table 6 and “field device status” as seen in table 7. Communication errors are shown in the response code, when the most significant bit of the response code is 1 there is a communication error present. Communication errors can be caused by framing errors and discrepancy between the message content and the received checksum. [16, p. 84-86.]

Table 6. HART response code Modified from [16, p. 85]

Bit mask	Definition
0x80	Bit is set to indicate communication error
0x40	Vertical parity error
0x20	Overrun error
0x10	Framing error
0x08	Longitudal parity error
0x04	Reserved
0x02	Buffer overflow
0x01	Reserved

Field device status as seen in table 7 describes basic operating condition of the device and informs the user when there is additional status information available. When the 0x10 bit is set there is additional status available and command #48 should be sent to get additional device status data as seen in table 8. [16, p. 86.] Since HART revision 7, multiple field device status bits are set. For example, when there is a loss of echo in ultrasonic level measurement device, bits 0x80 and 0x10 are set. Prior to revision 7 only bit 0x80 would be set. [18.]

Table 7. HART field device status Modified from [16, p. 86]

Bit mask	Definition
0x80	Field device malfunction
0x40	Configuration changed
0x20	Cold start
0x10	More status available
0x08	Analog channel fixed
0x04	Analog channel saturated
0x02	Non-primary variable out of limits
0x01	Primary variable out of limits

Universal command #38 can be used to reset status bit 0x40 of HART field device status byte “device configuration changed”. Since revision 5, it does not require any additional data. Many devices offer more status than is offered in one status byte, in that case command #48 should be used to read the additional device status as seen in table 8. Since HART revision 5, command #48 does not require any additional data in the command. [16, p. 87.]

Table 8. Indicing of HART command #48 additional device status Copied from [16, p. 80]

Response byte	Description	Index of LS bit	Index of MS bit
n/a	Field device status	0	7
0-5	Device specific status	of byte 0:8	of byte 5:55
6	Extended field device status	56	63
7	Device operating mode	64	71
8	Standardized status	72	79
9	Standardized status 1	80	87
10	Analog channel saturated	88	95
11	Standardized status 2	96	103
12	Standardized status 3	104	111
13	Analog channel fixed	112	119
14-24	Device specific status	of byte 14:120	of byte 24:207

In HART revision 7.5, NAMUR NE107 condensed status flags were added in extended field device status byte of command #48 response, where all status bits in additional device status are mapped into these four status categories as seen in table 9. [16, p. 214.] However, systems that use older revisions of HART may not support all status bits. Mapping can be changed with command #524 with data bytes of:

- Byte0: Index of LS bit to be changed
- Byte1: Number of entries to change
- Byte2: Byte2 onwards contains the status map code changes

Table 9. Extended device status bit map Modified from [16, p. 214]

Bit	Status	Meaning
0	Maintenance Required	NE107 status M - device is working, but needs maintenance
1	Device variable alert	Device variable alarm or warning state
2	Critical power failure	Device power critically low
3	Failure	NE107 status F - device has malfunctioned and cannot be trusted
4	Out of specification	NE107 status S - Deviations from permissible process conditions
5	Function check	NE107 status C - device variables temporarily invalid
6	Reserved	-
7	Reserved	-

4.2 Alternative Protocols to HART

HART-protocol does not have any direct competitors since there is no other protocol that retains analog signal compatibility. However, there are purely digital communication protocols that are used with industrial instruments. [16, p. 5.]

Perhaps most notable of these all-digital protocols are the fieldbus family of protocols standardized in IEC 61158 including FOUNDATION Fieldbus H1 and Profibus PA. Fieldbus family of networks share the same physical layer. The standard defines the cable type, supply voltage and data transmission rate of 31.25 Kbit/second. Fieldbus systems are designed to replace legacy systems like 4–20mA and HART in process automation, the main benefit over HART protocol is that their digital data transfer is a lot faster which enables closed loop control, more flexible networking topologies and their all-digital communication is less susceptible to noise. [19.]

4.2.1 Profibus PA

PROFIBUS PA protocol is one of the many Profibus protocols that are built under the parent organization of Profibus International. What separates Profibus PA from the rest is that it is made specifically for process automation. Profibus PA was designed to operate in hazardous environments and to be used with Profibus DP through DP/PA coupler. [20.]

The data link layer of Profibus PA is completed through a fieldbus data link. Profibus PA combines master-slave methodology and token passing in a master-slave network, where masters send commands to slaves and slaves respond accordingly. Each Network segment of Profibus PA contains up to 31 slave devices. [20.]

Common topology for Profibus is the bus topology, where all devices are connected by a node to single cable. The fallback of this is that if the cable is broken the entire bus segment fails. Another common topology is the star topology where each device is connected independently to a central controller. Star topology requires more cable but is more resistant to failure. [20.]

4.2.2 Foundation Fieldbus

Foundation fieldbus enables two-way communication between field devices and the DCS. It also enables access to diagnostic data. Foundation fieldbus uses peer-to-peer communication, unlike HART and Profibus that use master-slave method of communication. Peer-to-peer method means that devices can talk to each other without waiting for a command from master, fieldbus devices can communicate with between 2 and 32 devices. Foundation fieldbus also supports NAMUR NE107 recommendations for data management and condensed status. Foundation fieldbus uses function blocks to organize devices and set program parameters. [21.]

4.2.3 Industrial Ethernet Protocols

Like Fieldbus protocols, industrial ethernet too is an all-digital two-way communication protocol. Currently fieldbus protocols are more popular than ethernet in the all-digital industrial protocol section, but the popularity of ethernet is growing every year. Industrial ethernet have been widely used for years already in the routing level of process systems. Industrial ethernet utilizes a handshaking method to confirm that commands are received by device. Controller keeps sending the command until receiving device verifies the command with a response. [22.]

Complications that have prevented wider use of industrial ethernet are often harsh environmental elements of process plants. Ethernet was first adapted in an office environment where dirt, vibration or temperature are not a concern. When ethernet is used in industrial setting tougher components like connectors with IP67 protection may need to be used. [23.]

One of the benefits of industrial ethernet over fieldbus protocols is that higher data transfer speed of 100 Mbps allows more precise control of applications and reduces the risk of data collision, this is especially important in processes where speed and response times are critical. Furthermore, industrial ethernet makes large network expansions possible with cascading of switches. [23.]

5 Testing Communication and HART Command 48

The work started by installing test instruments to the Siemens ET200M distributed I/O-device. Siemens PDM was then used to configure instruments into the PCS7 system and assign the right device type manager (DTM) to devices. DTM can be downloaded directly from the manufacturers' own website. Communication with HART command 48 is tested to see how the device responds to commands and to get the additional device status information.

The goal was to build a method in continuous function chart (CFC) to write commands to the devices and read results cyclically, function blocks SFC 58 "WR_REC" (Write Record) as seen in figure 8 and SFC 59 "RD_REC" (Read Record) are used. To send commands and read results successfully following parameters must be defined:

- ID of the address area in hexadecimal number. B#16#54 for peripheral input modules and B#16#55 for output modules
- Logical address of the module. Logical address can be obtained from hardware configuration window, right clicking the I/O-module and opening symbol table where address of each device can be seen.
- Data record number in hexadecimal number. Diagnostic data of the module is contained in records 0 and 1. Each I/O-module has data record numbers for channels 0-7. Each device of the module can use its own data record as seen in table 10.

Figure 8. Writing HART command to the device with SFC 58 "WR_REC"

Table 10. Data record numbers for write/read commands Reprinted from [24]

channel	client	data record
0	command	80
0	reply	81
1	command	82
1	reply	83
2	command	84
2	reply	85
3	command	86
3	reply	87
4	command	88
4	reply	89
5	command	90
5	reply	91
6	command	92
6	reply	93
7	command	94
7	reply	95

In addition to defining the addresses and data records, two data blocks are made in Simatic Manager S7 Program. There are no naming and numbering rules for the blocks and since the blocks are only used for the testing, arbitrary name and number are given for the blocks. First block is DB11 for writing command to the HART device and it is connected to block SFC 58 in CFC. Data block 11 is made in compact format so it only contains 4 bytes of data as seen in table 11.

Table 11. HART compact command structure

Byte	Value	Comment
Byte0	20	20 = Compact message
Byte1	06	Number of preamble bytes
Byte2	30	Command 48 in hex
Byte3	00	Length in bytes

First byte tells the device that the command is sent in compact message format without SHC sequence. When message is sent in compact format with SHC sequence 20 is replaced with 60. SHC sequence is faster than normal messaging, but messages cannot be sent simultaneously, making it impractical if there are multiple devices communicating simultaneously in one module.

Second byte is the number of preamble bytes used. The number of preamble bytes are between 5 – 20, the exact number can be confirmed with command #0. First command should always be sent with 20 preamble bytes. Third byte is the command #48 in hex. Finally, the final byte is the length of the message to be written, but since it is unnecessary to write anything to the device this field is left to 0. Once the command is sent to the device, we can read the response into data block 51 using SFC 59 “RD_REC” as seen in figure 9. The program reads the device response cyclically and when there is an available response, writes it to data block 51. Response contains 28 bytes of data. First 10 bytes of the response are always in transparent message format, and the rest hold device status for command #48.

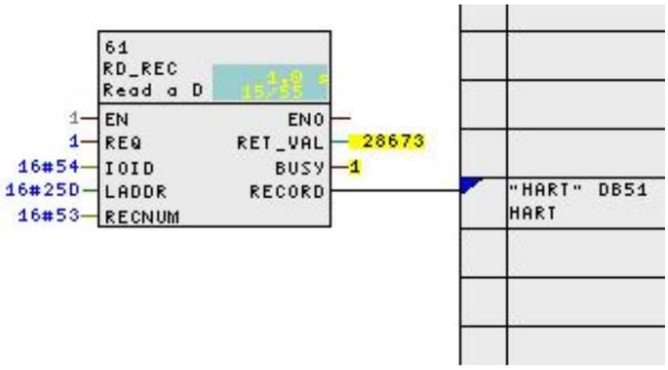


Figure 9. Reading HART response to DB51 with SFC 59 "RD_REC"

Next step was to simulate NAMUR NE107 device statuses to the device. This can be done with following HART commands:

- #526: Write status simulation mode
- #527: Simulate status bit

Another way to simulate status for the device is to use HART-modem that acts as a secondary master to the device. Using a HART-modem typically requires an external software. In this test, Endress+Hauser FXA195 modem was used with their DeviceCare SFE100 software, which is free to use for their own devices. Device manufacturers usually offer error codes for the devices in their user manuals. Device was simulated for all NAMUR NE107 statuses and results are listed in table 12. See appendix 2 for simulated statuses.

Most of the bytes are not important for finding out the device status. Initially the idea was to get all NAMUR NE107 status flags from Extended device status byte as seen in table 9, but it was later discovered that extended device status only contains first two bits in older HART revisions. First two bits being Maintenance required and Device Variable alert. Next step was to see if any important status information can be obtained from device status byte as seen in table 7. Device failure is indicated in most significant bit of device status byte, so now status C and S were still needed from the device. One possible option was to use device specific status bytes for it, but it was not a very practical solution since the idea was to develop a universal method that works for all devices.

Table 12. Response of HART command #48, simulated for each NAMUR NE107 status. M = Maintenance Required, S = Out of Specification, F = Failure, C = Function Check. Device Cerabar 5 M.

BYTE	OK	M	S	F	C	COMMENT
byte0	4	4	4	4	4	4 = Successful, 3 = Waiting, running
byte1	0	1	1	1	1	Extended response control (Appendix 1)
byte2	86	86	86	86	86	Start Delimiter. See (Table 3)
byte3	91	91	91	91	91	Device identification number 24 bits
byte4	19	19	19	19	19	-
byte5	9A	9A	9A	9A	9A	-
byte6	0E	0E	0E	0E	0E	Expanded device type code 14 bits
byte7	6A	6A	6A	6A	6A	-
byte8	30	30	30	30	30	Command #48 in hexadecimal
byte9	11	11	11	11	11	Byte count in hexadecimal. (Decimal 17)
byte10	0	0	0	0	0	Response code. See (Table 55)
byte11	0	10	10	90	10	Device Status. See (Table 56)
byte12	0	0	0	2	0	Byte12-17,26 are for device specific status
byte13	0	0	0	0	0	-
byte14	0	0	0	0	0	-
byte15	0	0	40	0	0	-
byte16	0	40	0	0	0	-
byte17	0	80	80	80	80	-
byte18	0	1	2	2	2	HART Extended Device Status Byte
byte19	0	0	0	0	0	Operating Modes (reserved)
byte20	0	0	0	0	0	Standardized status 0
byte21	0	0	0	0	0	Standardized status 1
byte22	0	0	0	0	0	Analog channel saturated
byte23	0	0	0	0	0	Standardized status 2
byte24	0	0	0	0	0	Standardized status 3
byte25	0	0	0	0	0	Analog channel fixed
byte26	0	0	0	0	0	Truncated after last used byte
byte27	D1	0	3	C1	43	Checksum

Since there was not a practical way of getting C and S status from the device at this moment, they were combined into one status flag U for Unknown. Next step was to add some extra functionality to the CFC program to first isolate the status bits and to combine them again into one single status byte, that can be used in status monitoring. This was done with two BY_BO converters to isolate the single bits from status bytes and one BO_BY converter to combine those bits again. Final byte values indicating the following status flags:

- 16#1: Failure
- 16#2: Maintenance Request
- 16#4: Unknown (Out of Service/Function Check)
- 16#8: Device status OK

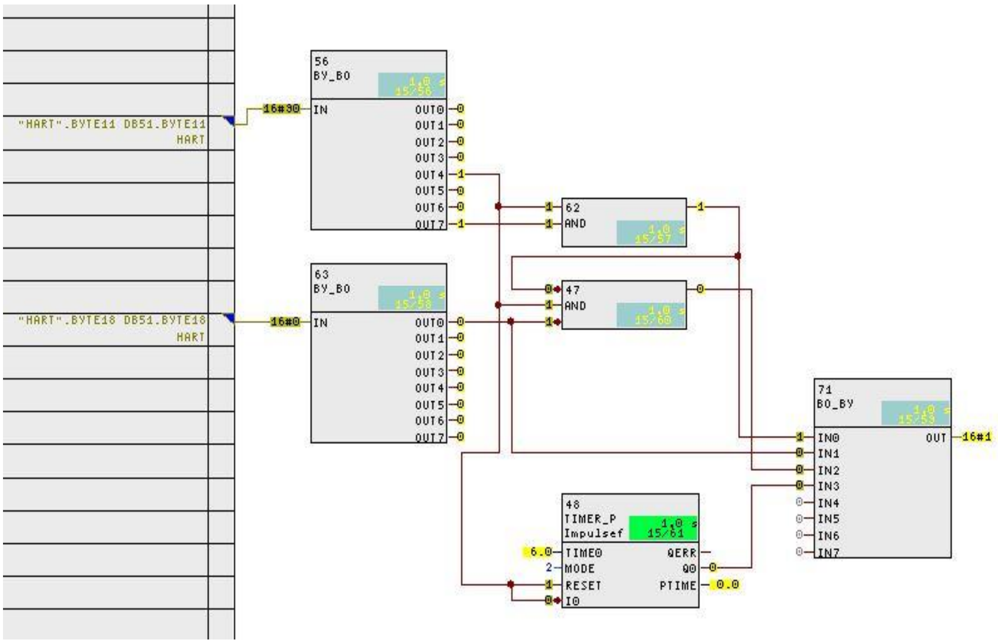


Figure 10. Combining all status bits into one byte in CFC using function blocks

Combining all status information to one byte makes it more practical for monitoring the status of the device, since all those statuses can be obtained from single byte. Final status byte is attached to analog monitoring block “UserStat” field as seen in figure 11. Three binary status messages:

- Failure bit
- Maintenance Required
- Unknown

are attached to “ExtMsg” fields to start events in winCC window. Such events may include alarms, warnings and historical data.

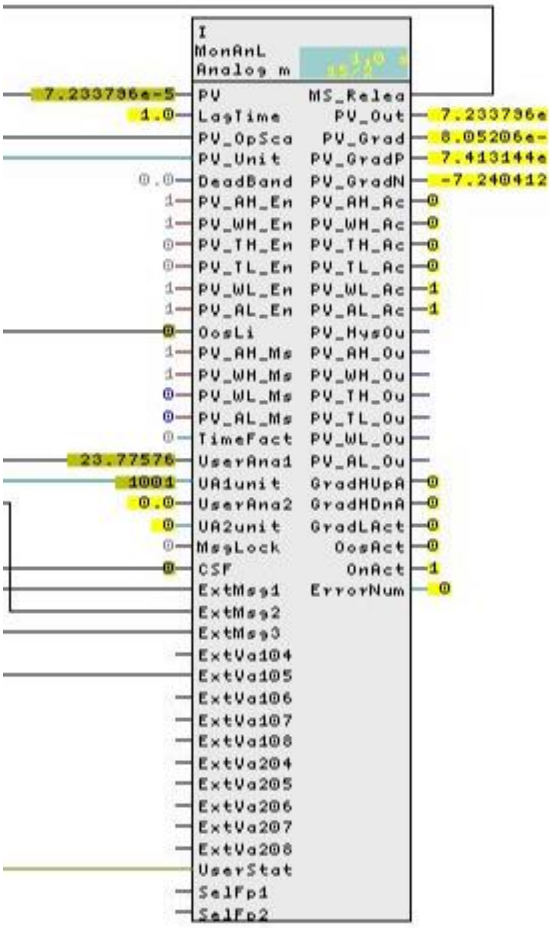


Figure 11. MonAnL monitoring block

Final status byte can now be used in winCC device faceplate. In winCC object can be assigned with bit values of the final status byte.

Graphical box is made for every faceplate with letters that have colors to represent different statuses as seen in figure 12. User inputs like disabling status monitoring can be added later in development process.

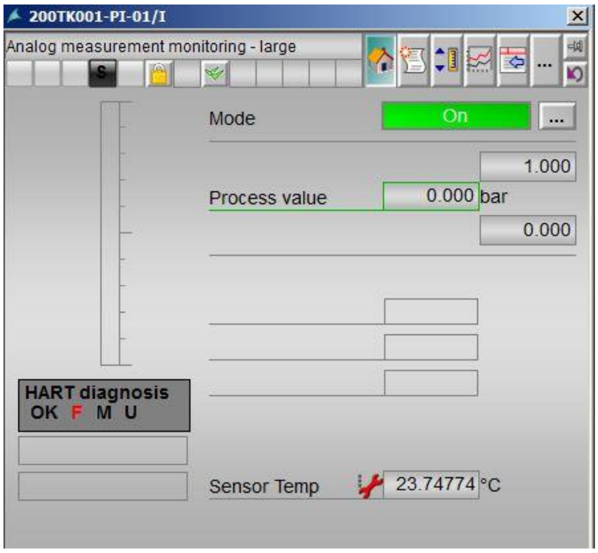


Figure 12. Device faceplate with HART diagnosis statuses

6 Implementation and Results

Using many function blocks for all HART devices is not very practical, because it would take too much time and adding functionality to the method would become increasingly difficult. Making a program in structured control language or SCL to include all functionality in single block would offer many benefits in scalability and flexibility. Furthermore, using structure data types in SCL means that HART command responses would no longer be required to be written in data blocks. Program inputs, outputs and structures can be defined as variables in SCL using the VAR function as seen in listing 1.

```
VAR_INPUT
    Writestart:     BOOL;
    Readstart:      BOOL;
    Addr:           BOOL;
    HART_COMMAND:   STRUCT
        X1:         BYTE:=20;
        X2:         BYTE:=6;
        X3:         BYTE:=30
    END_STRUCT;
END_VAR
```

Listing 1. Example of defining inputs in SCL

SFC blocks like “RD_REC” and “WR_REC” used earlier in testing phase of the study can also be called into the SCL code as seen in listing 2. It is good practice to store block inputs and outputs as defined variables in code to be later modified or used.

```
WriteRecord := “WR_REC” (REQ := writestart
    ,IOID:=iomod
    ,LADDR := addr
    ,RECNUM := rcnm_wr
    ,BUSY := busy_wr
    ,RECORD := HART_COMMAND
    ,RET_VAL := err_wr
    );
```

Listing 2. Defining SFC 58 “WR_REC” in SCL

Having potentially many devices with different addresses, it may be advantageous to add system clock into the code as seen in listing 3. System clock can be configured to reach certain time before it resets.

```
PostTime := TIME_TCK();
IF PostTime < PreTime THEN
    ElapsedTime := TIME_TO_DINT((T#24D20H31M23S647MS -
    PreTime) + PostTime);
ELSE
    ElapsedTime := TIME_TO_DINT((PostTime)/5000);
END_IF;
PreTime := PostTime;
```

Listing 3. Adding TIME_TCK to SCL and getting elapsed time since last run. Modified from [25]

In this program system clock was used to read devices when their address matches the time on the system clock as seen in listing 4.

```
IF ElapsedTime = WORD_TO_DINT(addr) OR user THEN
    OperatingMode := TRUE;
ELSE
    OperatingMode := FALSE;
END_IF;
```

Listing 4. Sending command when address matches elapsed time

Since the device status must be obtained from more than one status byte of the additional status response, the response bytes must be mapped into bit maps and made into a single byte of status in the code. Overall the code contains over 150 lines. However, it is not necessary to present the entire code in this study. The completed block has the following inputs as seen in figure 13:

- addr: Logical address of the device
- init (optional: for user to disable/enable HART status)
- user (optional: for user to send HART command)
- iomod: 54 for input modules
- rcnm_wr: record number for “WR_REC” block

- rcnm_rd: record number for “RD_REC” block

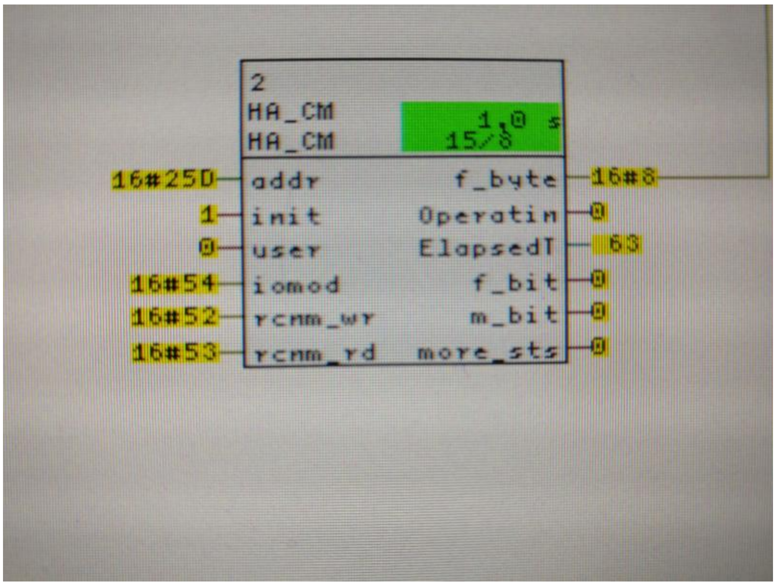


Figure 13. HART communication block made in SCL

Outputs of the block are as follows:

- f_byte: final status byte (see page 27 for contained bits)
- OperatingMode: Sending HART command active
- ElapsedTime: Time since last run
- f_bit: Failure bit
- m_bit: Maintenance required bit
- more_sts: More status available bit

7 Conclusion

This section concludes the study. Results and outcome of the project are explained in this section and possible future improvements to the design are discussed. At the beginning of the study, the proposal was to obtain four NAMUR NE107 device status flags from HART compatible instruments. This was mostly a success, since two out of the four statuses can currently be obtained. Failure status can be obtained from the most significant bit of the device status byte and maintenance required can be obtained from extended device status byte. Two of the missing statuses are out of specification and function check, which can be added later in the source code when HART revision 7.5 becomes more commonly used. As per HART application guide the extended field device status byte has two used bit fields, that are the bit 1 and bit 2, in revision 7.5 four more bits were added including all the NAMUR NE107 statuses.

In addition to creating a method for obtaining device status, this study establishes a way to communicate with field devices using HART commands. HART commands allow, with a little time invested in programming, user a way to access all functions and data that the device has. Furthermore, this method of obtaining data is not system dependent and can be easily modified in the future, to include all status flags or to work with other HART commands. In the future this method of obtaining the device status could be added in to the operating environment as an additional feature or standardized to be included in every plant operating view. Having an easy access to the device status means that operators and plant maintenance workers have more time to react to device malfunctions and optimization of the plant would be easier.

This study focuses on researching and implementing diagnostic capabilities of smart instruments using HART-protocol. Even though there are other completely digital protocols in the market that have similar capabilities with HART-protocol, it continues to be widely used perhaps partly due to being compatible with analog signal and having a huge selection of devices that support it. As industry 4.0 becomes more widely adapted in the field of process industry, standards change and new methods of using digital systems are developed. However, the demand for HART seems to be still relevant and new features to the protocol keep it relevant with changing process solutions.

8 References

[1] P. Nykänen, "Outotec historiikki," 2016. [Online]. Available: http://www.outotec.com/globalassets/company/about-outotec/outotec_historiikki_sivut.pdf. [Accessed January 2020].

[2] Outotec -Oyj, "Our business," [Online]. Available: https://www.outotec.com/company/about-outotec/our-business/. [Accessed January 2020].

[3] Christoph Käemmerer, Analog Devices, Inc, "Intelligent field instrument the smart way to indusrty 4.0," [Online]. Available: https://www.analog.com/media/en/technical-documentation/tech-articles/Intelligent-Field-Instruments-The-Smart-Way-to-Industry-4-0.pdf. [Accessed January 2020].

[4] Automation.com, "Using smart field devices to improve safety system performance," [Online]. Available: https://www.automation.com/pdf_articles/Using_smart_field_devices_to_improve_safety_system_performance.pdf. [Accessed January 2020].

[5] A. Geipel-Kern, "Process Worldwide," 4 11 2009. [Online]. Available: https://www.process-worldwide.com/automated-partial-stroke-testing-enhances-process-reliability-and-system-availability-a-302178/. [Accessed February 2020].

[6] Instrumentationtools, "Electromagnetic flow meters working principle," [Online]. Available: https://instrumentationtools.com/electromagnetic-flow-meters-working-principle/. [Accessed March 2020].

[7] ELECTRONET, ”Electromagnetic flow meters working principle,” [Online]. Available: https://www.eeplindia.com/Electromagnetic-Flow-Meters-Working-Principle.html. [Haettu March 2020].

[8] Cross Company, "Magnetic flowmeter function application," [Online]. Available: https://www.crossco.com/resources/articles/magnetic-flow-meter-function-application/. [Accessed March 2020].

[9] O. Grievson, ”the rights and wrongs of measuring flow in wastewater,” 10 February 2015. [Online]. Available: https://www.wateronline.com/doc/the-rights-and-wrongs-of-measuring-flow-in-wastewater-0001. [Haettu March 2020].

[10] Emerson, "Magnetic flow meters diagnostics and smart meter verification," 2018.

[11] Electrical 4 U, "RTD construction and working pronciple," 15 May 2018. [Online]. Available: https://www.electrical4u.com/resistance-temperature-detector-or-rtd-construction-and-working-principle/.

[12] Surecontrols, "RTD vs thermocouple," [Online]. Available: https://www.surecontrols.com/rtd-vs-thermocouple/. [Accessed March 2020].

[13] Instrumentationtools, "resistance temperature detectors sources of error," [Online]. Available: https://instrumentationtools.com/resistance-temperature-detectors-sources-of-error/. [Accessed March 2020].

[14] instrumentation tools, "NAMUR NE107 standard," [Online]. Available: https://instrumentationtools.com/namur-ne107-standard/. [Accessed January 2020].

[15] B. M. Simon Paonessa, Precision Digital, [Online]. Available: https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops. [Accessed January 2020].

[16] R. Bowden, "HART Technology A Technical Overview," Austin, TX, FieldComm Group, 2017, p. 266.

[17] Instrumentationtools, "HART communication tutorial part 3," [Online]. Available: https://instrumentationtools.com/hart-communication-tutorial-part-3/. [Accessed January 2020].

[18] Siemens Canada Limited, "Setting up and working with HART networks," Peterborough, Ontario.

[19] B. Schuessler, 29 January 2013. [Online]. Available: https://blog.pepperl-fuchs.us/blog/bid/261988/Are-FOUNDATION-fieldbus-and-PROFIBUS-PA-the-same-thing. [Accessed February 2020].

[20] Real Time Automation, "Profibus," [Online]. Available: https://www.rtautomation.com/technologies/profibus/. [Accessed February 2020].

[21] Visaya Solutions, "Everything you need to know about foundation fieldbus," 18 September 2018. [Online]. Available: https://visaya.solutions/en/article/everything-you-need-to-know-about-foundation-fieldbus/. [Accessed February 2020].

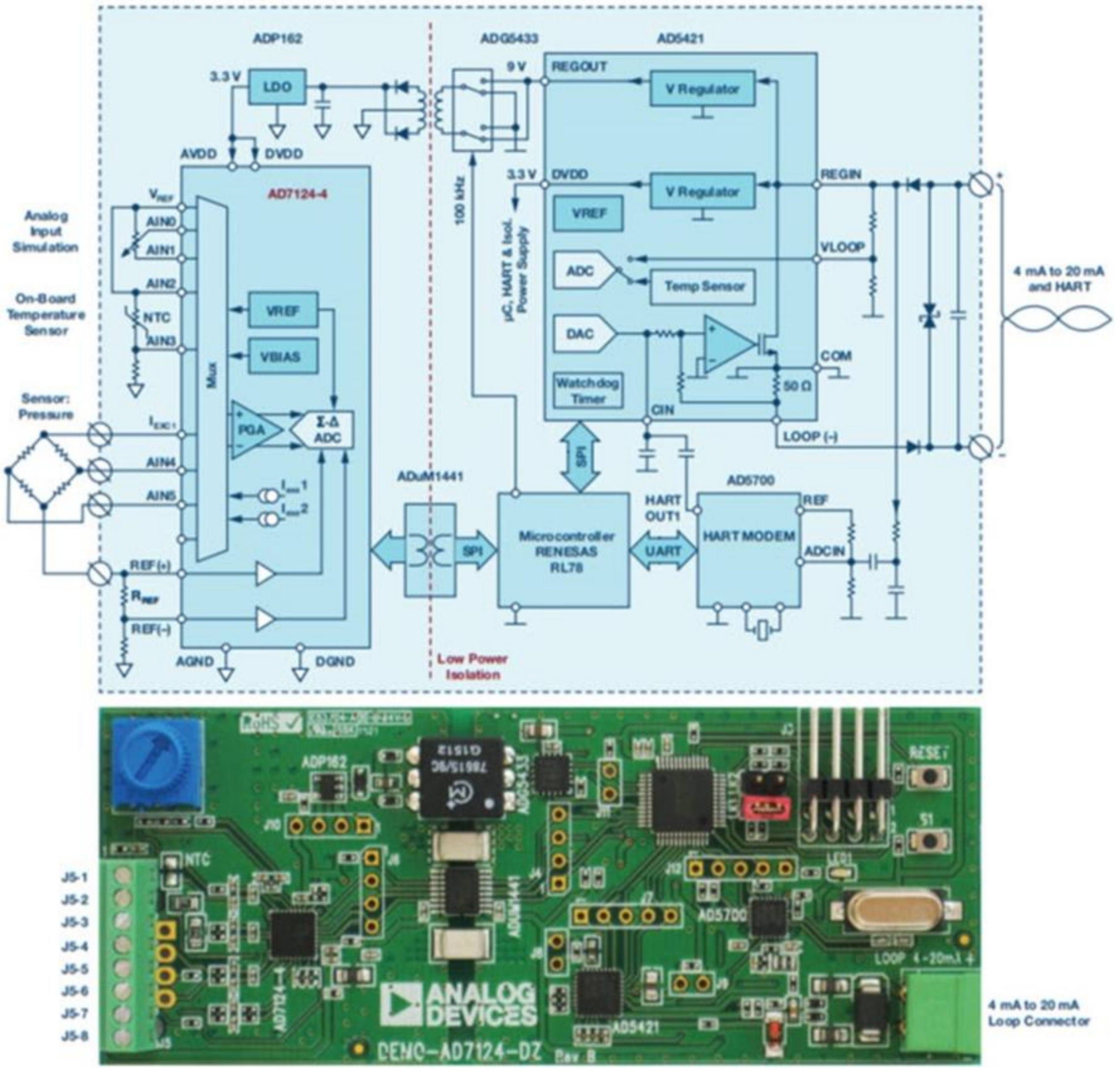
[22] Real Time Automation, "What is industrial ethernet," [Online]. Available: https://www.rtautomation.com/industrial-library/what-is-industrial-ethernet/. [Accessed February 2020].

[23] Kunbus, "Industrial ethernet basics," [Online]. Available: https://www.kunbus.com/industrial-ethernet-basics.html. [Accessed February 2020].

[24] Siemens, "ET 200M distributed I/O device HART analog modules," 2011.

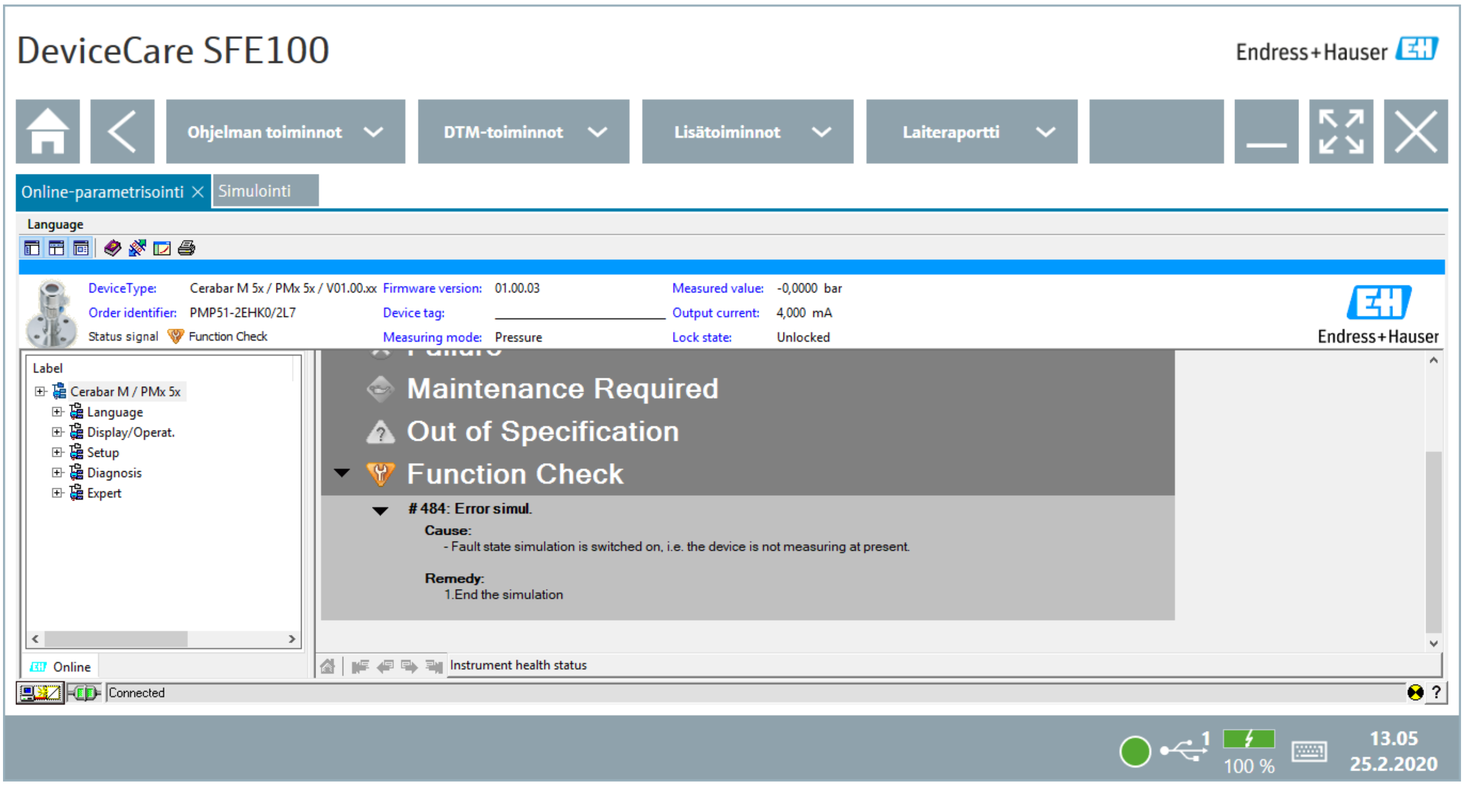
[25] Smiffy, "Siemens support forum," 2 January 2017. [Online]. Available: https://support.industry.siemens.com/tf//WW/en/posts/how-to-read-time-period-in-scl/160992?page=0&pageSize=10. [Accessed February 2020].

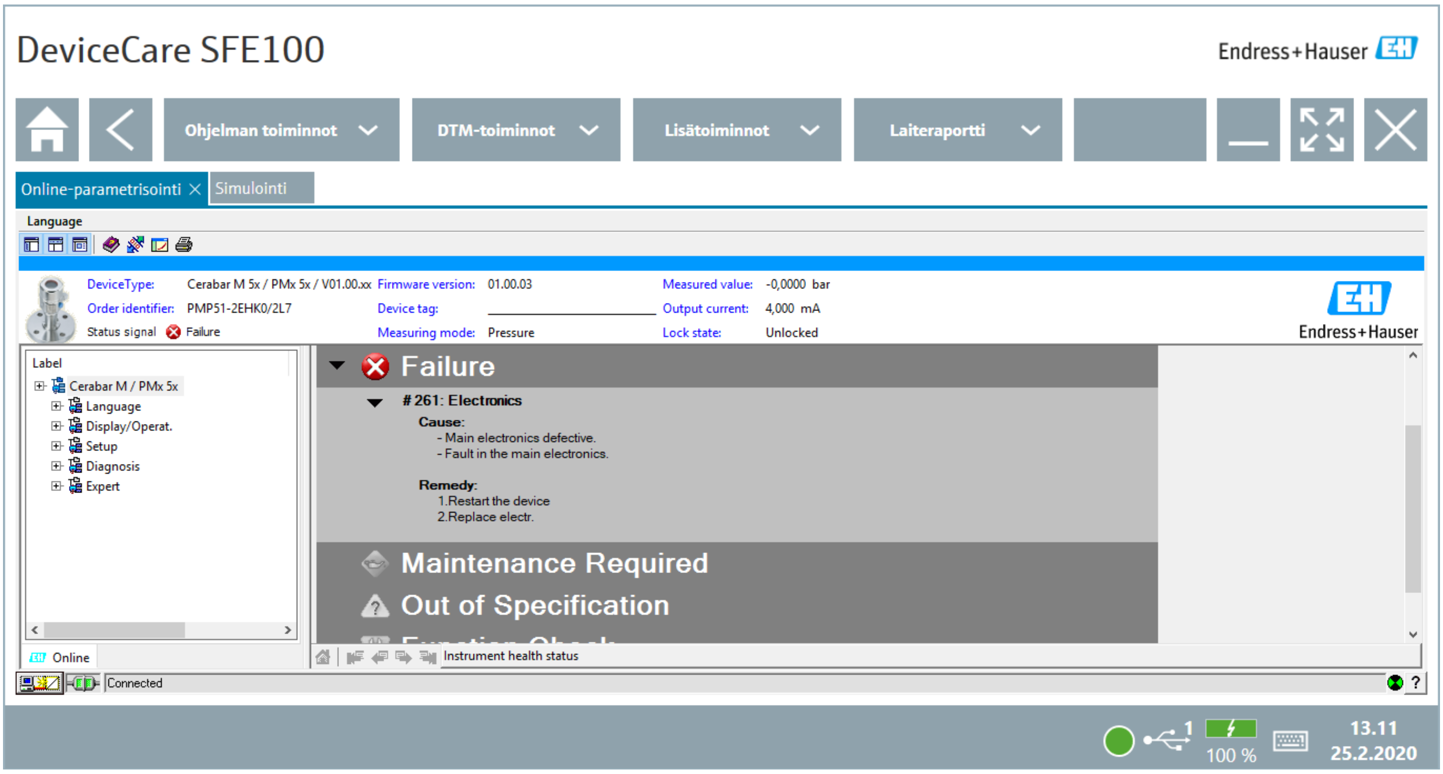
Appendix 1: AD7124

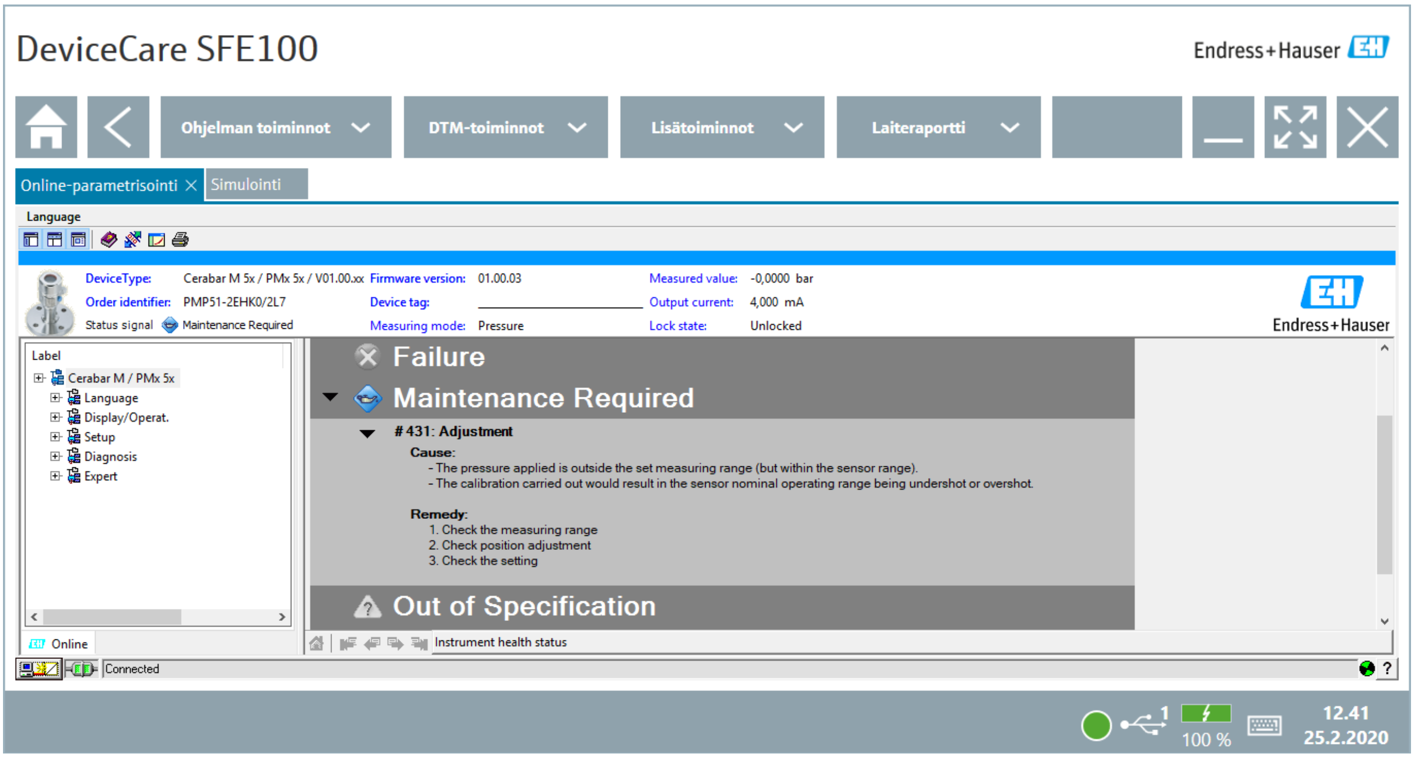


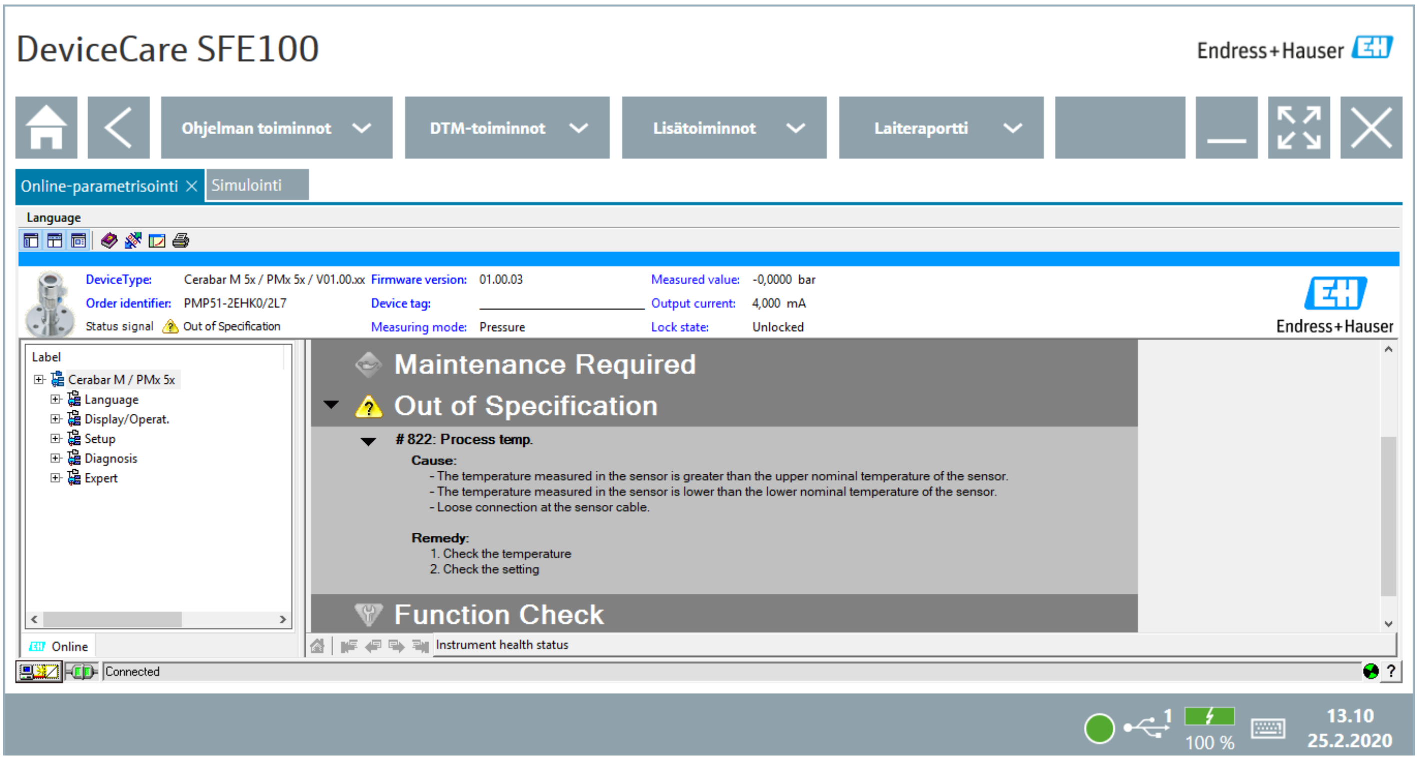
Metropolia University of Applied Sciences

Appendix 2: Status simulation values for Cerabar 5 M




DeviceCare SFE100
Endress+Hauser
Ohjelman toiminnot
DTM-toiminnot
Lisätoiminnot
Laiteraportti
Online-parametrisointi
Simulointi
Language
DeviceType: Cerabar M 5x / PMx 5x / V01.00.xx
Firmware version: 01.00.03
Measured value: -0,0000 bar
Order identifier: PMP51-2EHK0/2L7
Device tag:
Output current: 4,000 mA
Status signal Failure
Measuring mode: Pressure
Lock state: Unlocked
Label
Cerabar M / PMx 5x
Language
Display/Operat.
Setup
Diagnosis
Expert
Failure
261: Electronics
Cause:
- Main electronics defective.
- Fault in the main electronics.
Remedy:
1.Restart the device
2.Replace electr.
Maintenance Required
Out of Specification
Online
Instrument health status
Connected
100 %
13.11
25.2.2020


DeviceCare SFE100
Endress+Hauser
Ohjelman toiminnot
DTM-toiminnot
Lisätoiminnot
Laiteraportti
Online-parametrisointi
Simulointi
Language
DeviceType: Cerabar M 5x / PMx 5x / V01.00.xx
Order identifier: PMP51-2EHK0/2L7
Status signal Maintenance Required
Firmware version: 01.00.03
Device tag:
Measuring mode: Pressure
Measured value: -0,0000 bar
Output current: 4,000 mA
Lock state: Unlocked
Label
Cerabar M / PMx 5x
Language
Display/Operat.
Setup
Diagnosis
Expert
Failure
Maintenance Required
431: Adjustment
Cause:
- The pressure applied is outside the set measuring range (but within the sensor range).
- The calibration carried out would result in the sensor nominal operating range being undershot or overshot.
Remedy:
1. Check the measuring range
2. Check position adjustment
3. Check the setting
Out of Specification
Online
Instrument health status
Connected
100 %
12.41
25.2.2020


DeviceCare SFE100
Endress+Hauser
Ohjelman toiminnot
DTM-toiminnot
Lisätoiminnot
Laiteraportti
Online-parametrisointi
Simulointi
Language
DeviceType: Cerabar M 5x / PMx 5x / V01.00.xx
Firmware version: 01.00.03
Measured value: -0,0000 bar
Order identifier: PMP51-2EHK0/2L7
Device tag:
Output current: 4,000 mA
Status signal Out of Specification
Measuring mode: Pressure
Lock state: Unlocked
Label
Cerabar M / PMx 5x
Language
Display/Operat.
Setup
Diagnosis
Expert
Maintenance Required
Out of Specification
822: Process temp.
Cause:
- The temperature measured in the sensor is greater than the upper nominal temperature of the sensor.
- The temperature measured in the sensor is lower than the lower nominal temperature of the sensor.
- Loose connection at the sensor cable.
Remedy:
1. Check the temperature
2. Check the setting
Function Check
Online
Instrument health status
Connected
100 %
13.10
25.2.2020