ALMA MATER STUDIORUM - UNIVERSITÁ DI BOLOGNA

SCHOOL OF ENGINEERING

DEPARTMENT OF
ELECTRICAL,ELECTRONIC AND INFORMATION ENGINEERING
“Guglielmo Marconi”
DEI

MASTER DEGREE IN AUTOMATION ENGINEERING

MASTER THESIS

in

- Automation software and design patterns -

MODBUS TCP-IP INTERFACE OF PARTICLE AND MICROBIOLOGICAL COUNTS FOR THE STERILE MANIPULATION OF HIGHLY POTENT ACTIVE INGREDIENTS.

CANDIDATE:
Richard Duval Foasse Nyie

SUPERVISORS:
Prof. GIANLUCA PALLI
Ing. MARCO GUARDIGLI

Academic Year 2023/2024
Graduation Session VI

Contents

Abstract

Highly active ingredients require special rules for their production, because accidental contamination with other materials could have serious consequences, for the health of patients and because they, could also represent an occupational risk for personnel, who come into direct contact with these substances, during all production phases [1-3]. As with regulations regarding the production and handling of some food products, even more attention must be paid to medicines. Among the greatest dangers associated with the production of any drug, and particularly during the production of those classified as highly active or sensitizing, we find cross-contamination (a term used to describe the process by which microorganisms or unwanted substances are transferred from a surface , food or object to another) characterized by the spread of germs. Above all, GMPs underline the problem of contamination by substances that could be harmful to health, therefore the production of some products (in particular, sensitizing and/or highly active) must take place in dedicated areas in order to guarantee a clear separation of these substances from other materials [8]. The environmental and production process monitoring program is part of the overall CCS (Contamination Control Strategy) and is used to monitor controls aimed at minimizing the risk of microbial and particulate contamination. In this thesis work we will focus on Modbus interfacing with particle and microbiological counts as a control strategy to detect the presence of contaminants during the production phase of ingredients considered highly active.

To my family and to the people close to me who have supported and encouraged along the way

List of Figures

List of Tables

Introduction

Tema Sinergie is a company based in Faenza, Italy. Founded in 1985, it represents one of the most qualified companies in the world in the production of manipulation and dosing systems, for radiopharmaceuticals used in Nuclear Medicine. The company boasts over 30 years of expertise, which has enabled it to pursue continuous research and enhance its expertise. Its extensive range of technology and unparalleled support extends from synthesis and dispensing cells to radiopharmaceutical fractionators and injectors, monitoring, and beyond.
The present pharmaceutical manufacturing procedures of the company are subject to regulatory guidelines, established by the Food and Drug Administration (FDA) or Europe, Middle East and Africa (EMEA), which emphasize ensuring product sterility as well as comprehending the operator safety and health and environmental protection regulations, outlined in the Control Of Substances Hazardous to Health (COSHH) and the Occupational Safety and Health Act (OSHA)
This thesis, aims to perform a Modbus TCP-IP interfacing with particle and microbiological counts on a modular isolation system for aseptic containment designed specifically for the sterile manipulation of highly potent active ingredients to verify the presence of bacteriological fungi.

Chapter 1

System representation

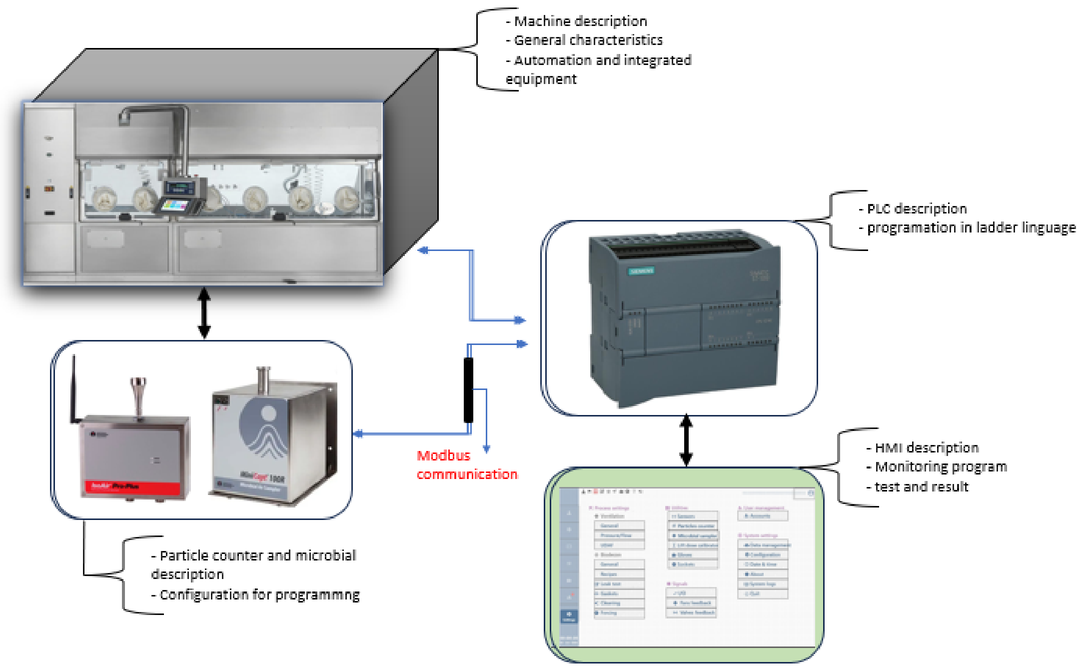


Figure 1.1: Overview system operation.

1.1 Machine description

The Aseptic Containment Isolator System is a modular manufacturers' compliance with, its Current Good Manufacturing Practice (CGMP) of Class A/ISO 5, specifically designed for the Sterile processing of High Potent Active Pharmaceutical Ingredients (HPAP Is).

The machine is based on a versatile modular concept, composed of a single 4-gloves working chamber which can be easily expanded with additional chambers/ airlocks. Integrated manual and automatic Wash-In-Place (WIP) for cleaning processes, thus improving safety for the operators involved in the production of potentially hazardous compounds and reducing the risk of possible cross contamination. Integrated sliding control module, containing electrical and pneumatic components, allows direct access for all maintenance operations.
A Virtual Private Network (VPN) connection allows a secure and private communication via Internet network and provides remote access and assistance, updates and maintenance any time, everywhere. Integrated Automatic Glove Leak Testing System (AGLTS) which performs an independent leak test cycle for each glove mounted on the isolator system according to the Positive Pressure Decay Method described in the international standard ISO 14644-7; Wireless Wi-Fi data transmission based on the TC P/IP protocol is used to recognize the glove to be tested. Integrated Supervisory Control And Data Acquisition (SCADA) with Historian Data Management Software (DMS),DataWall, a software application developed by Tema Sinergie for data management. Integrated an energy saving Thanks to the new sustainable low voltage ventilation fans consumption throughout its life.

1.2 General characteristics of machine

In this section some important machine characteristics is enumerated:

- Fully PLC controlled isolator system (PLC Siemens S7-1200 and SCADA integration on SIMATIC IPC277E (Nanopanel PC), 7” Touch Panel).
- Modular configuration concept of the system.
- Integrated manual and automatic Wash-In-Place (WIP).
- Integrated Vapor-Phase Hydrogen Peroxide (VPHP) Decontamination Systems (HYPER) and distribution circuit.
- VPHP resistant relative humidity and temperature sensor.
- Main shell structure made of high-grade stainless steel
- Integrated Viable and Non-viable (Particle) monitoring system
- UPS for PLC reliability
- Complete documentation package according to Tema Sinergie standards.

- VPHP sensor for environment (safety sensor)
- VPN connection for a secure and private communication via Internet network – remote access,updates and maintenance.

Chapter 2

Environmental Monitoring And Aseptic Processing - Technology And Microbiology

Environmental monitoring is crucial in ensuring the quality and safety of pharmaceutical products, especially in aseptic processing. Let's delve into some key aspects:

2.1 Why We Monitor?

- Contamination can adversely affect product quality and effectiveness.
- Manufacturing failures, human errors, and undetected contaminants can compromise the final product.
- Particulate matter (both inert/nonviable and viable) can vary in size, shape, and density. Sources include personnel, equipment, air filtration systems, and more.

2.2 Typical Monitoring Systems:

- Particle Monitors: Monitor airborne particulates (both inert and viable).
- Microbial Samplers: Collect microbial samples from fixed or portable locations.
- Parameters such as temperature, humidity, and pressure also impact clean-room operations.

2.3 Automation and Data Integrity:

- Automation ensures consistency and eliminates routine tasks influenced by humans.
- Data integrity, compliance with regulations (e.g.Good Manufacturing Practice – GMP), and system integration are essential.

2.4 Monitoring Air:

- Total Particulates: Includes both inert/nonviable and viable particles.
- Particle Counting: Measures particles 0.5 μ m in diameter and larger.
- Viables: Typically bacterial and fungal spores.
- Settle Plates (Passive Air Sampling): Commonly used for viable sampling.
- Air Samplers (Active Air Sampling): Collects airborne microorganisms.

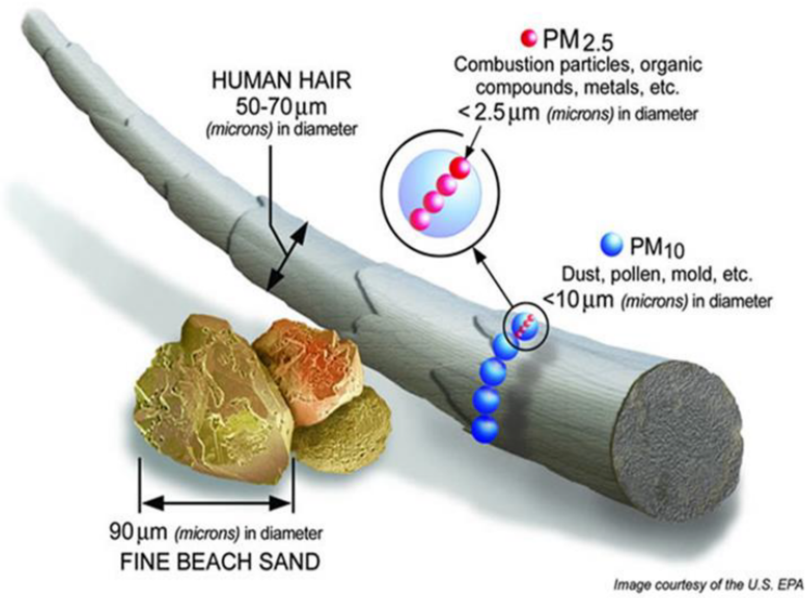


Figure 2.1: Total Particulates.

2.5 Sample point placement:

- Position the sample inlet close to critical locations.
- Avoid placing it directly over critical points to prevent turbulence and air starvation.
- Locations must be risk-based and strategically chosen.

(a) (b)

Figure 2.2: Examples of Positioning the Sample Inlet.

2.6 Physical installations:

- Straight connections between sample inlets and particle counters are ideal.
- Consider non-viable sampling cones, machine baseplates, and filling lines.
- Monitor critical areas continuously and the zone surrounding the product during exposure.

Environmental monitoring in pharmaceutical manufacturing faces several challenges. Let's explore some of them:

1. Contamination Control:

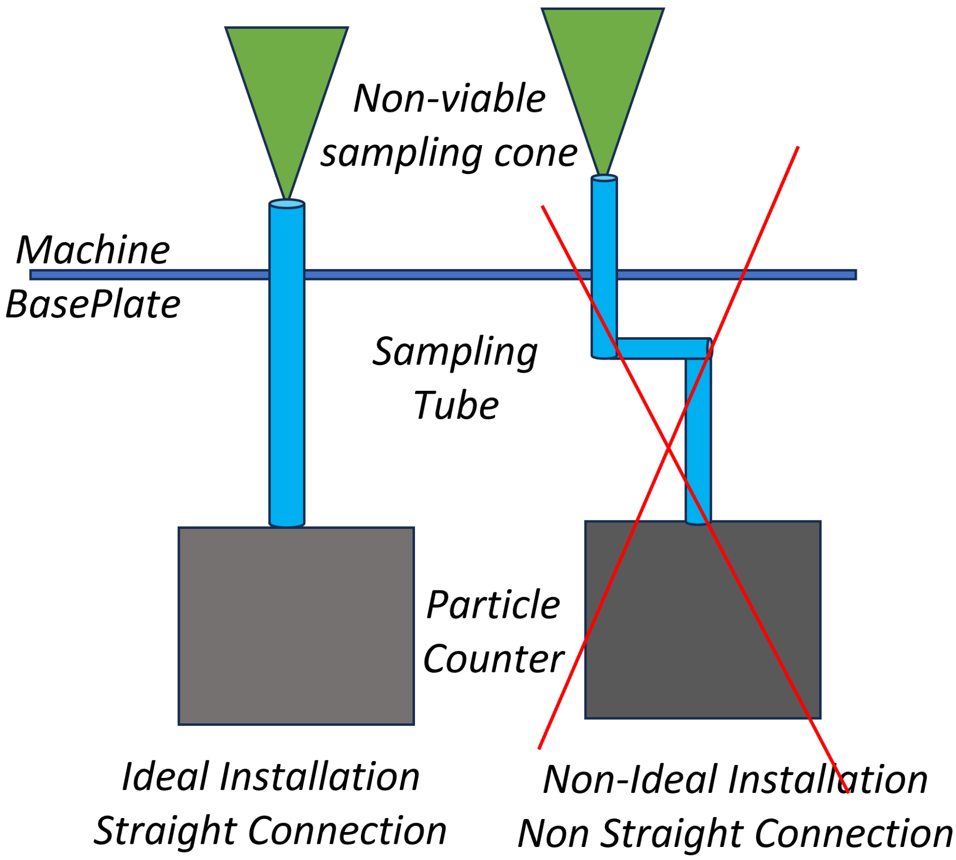


Figure 2.3: Ideal Vs Non straight connection.

- Airborne Contaminants: Maintaining clean air in aseptic areas is critical. Contaminants can enter through personnel, equipment, or ventilation systems.
- Surface Contamination: Surfaces (walls, floors, equipment) can harbor microbes. Regular cleaning and disinfection are essential.

2. Sampling Variability:

- Sampling Frequency: Balancing the need for frequent sampling with practicality.
- representativeness: Ensuring samples accurately reflect the overall environment.

3. Data Interpretation:

   - Thresholds and Limits: Determining acceptable levels of contamination.
   - False Positives/Negatives: Interpreting results correctly to avoid unnecessary actions or overlooking issues.

4. Personnel Training and Behavior:
   - Gowning and Aseptic Techniques: Proper training is crucial for minimizing contamination risk.
   - Behavioral Compliance: Ensuring consistent adherence to protocols.

5. Equipment and Calibration:
   - Calibration: Regular calibration of monitoring equipment.
   - Maintenance: Ensuring equipment functions optimally.

6. Validation and Qualification:
   - Validation of Monitoring Systems: Ensuring accuracy and reliability.
   - Qualification of Cleanrooms: Demonstrating compliance with standards.

7. Data Integrity and Documentation:
   - Recording Data: Properly documenting monitoring results.
   - Audit Trails: Ensuring data integrity and traceability.

Clean-rooms play a crucial role in ensuring aseptic conditions for various industries, including pharmaceuticals, medical devices, and research. Here's how they achieve and maintain aseptic conditions:

1. Design Features:
   - Materials Selection: Use materials that do not contribute to contamination (e.g., shedding particles or outgassing).
   - Smooth Surfaces: Floors, walls, ceilings, and counters should be easily cleanable and free from crevices where microbes can accumulate.
   - Temperature and Humidity Control: Maintain optimal conditions to prevent microbial growth.
   - Air Filtration: Install HEPA filters to filter the air supply into the cleanroom.

2. Personnel Training:
    - Gowning Procedures: Proper gowning techniques are essential to prevent contamination from personnel.
    - Cleaning Protocols: Train employees on effective cleaning procedures to maintain sterility.
    - Waste Removal: Proper disposal of waste to prevent cross-contamination.
3. Sterile Practices:
    - Aseptic Gowning: Employees wear sterile clothing, including gloves, masks, and coveralls.
    - Hand Hygiene: Rigorous handwashing and sanitization.
    - Strict Procedures: Follow established protocols for movement, handling, and interactions within the cleanroom.
4. Regular Monitoring and Validation:
    - Environmental Monitoring: Regularly sample air, surfaces, and personnel to detect any deviations.
    - Validation Studies: Periodic validation ensures that the cleanroom meets aseptic standards.
5. Preventing Contaminants:
    - Airflow Control: Unidirectional airflow minimizes particle movement.
    - Isolators and Laminar Flow Hoods: Used for critical processes.
    - Sterilization Techniques: Autoclaving, gamma irradiation, or chemical sterilization.
6. Risk-Based Approach:
    - Critical Areas: Focus efforts on critical zones (e.g., filling lines, aseptic processing areas).
    - Risk Assessment: Identify potential sources of contamination and prioritize control measures.

We note that, achieving aseptic conditions is a collaborative effort involving design, training, and strict adherence to protocols.

2.7 Sterilization Process with Vaporized Hydrogen Peroxide - VHP

Hydrogen Peroxide Sterilant is a specially formulated, high purity, stabilized, 35% aqueous hydrogen peroxide solution.
The VHP process consists of three stages: Conditioning, sterilant exposure and post-conditioning, all stages performed in a single chamber. The process is performed under deep vacuum conditions, typically 1-10 millibars, at a temperature range typically of 28-40°C. The total cycle time (door closed to door open) is typically less than or equal to eight hours. However, cycle time may vary depending on product composition, packaging materials, temperature, load size and configuration. It is compatible with a wide range of polymeric materials, therefore it represents an effective sterilization method for disposable healthcare products such as:

- Systems and devices with electronic components
- Pharmaceutical containers
- Parenteral drug delivery systems such as pre-filled syringes
- Combination of devices and drug delivery
- Complete assemblies or devices with loose components (e.g. needles)
- Temperature sensitive devices

2.8 Environmental monitoring - total particle

A total particle monitoring program, should be established to obtain data, for assessing potential contamination risks, and to ensure the maintenance of the environment for sterile operations in a qualified state.The limits for environmental monitoring of airborne particle concentration for each graded area are given in fig 2.5

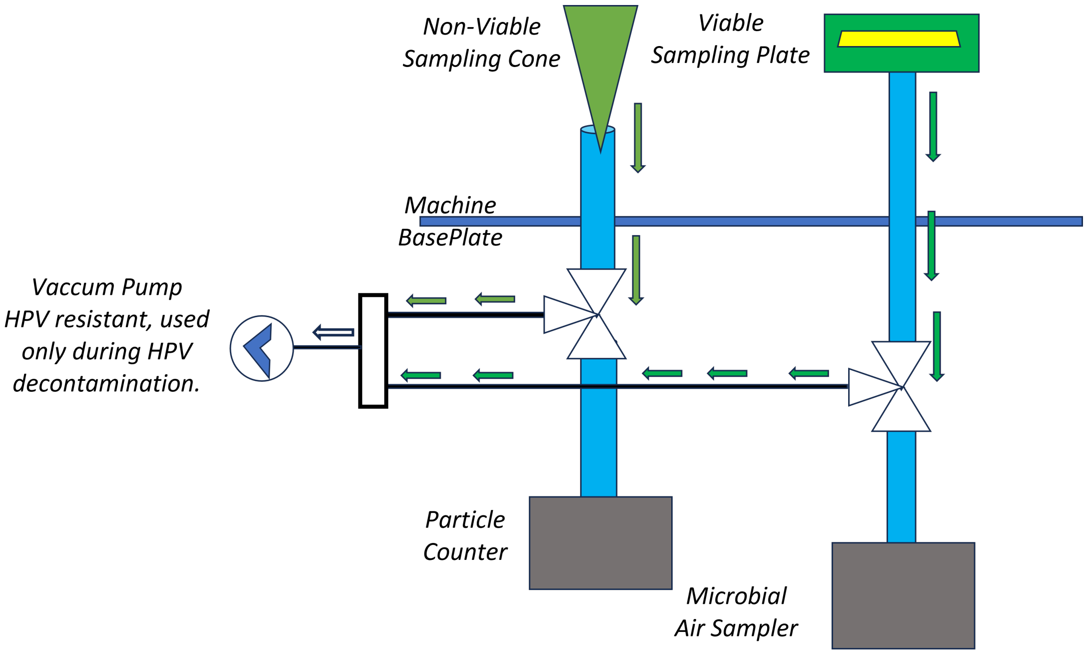


Figure 2.4: Air flow during decontamination mode.

	Maximum permitted number of particles per m^3 equal to or greater than the tabulated size			
	At rest		In operation	
Grade	0.5 µm	5.0µm	0.5 µm	5.0µm
A	3 520	20	3 520	20
B	3 520	29	352 000	2 900
C	352 000	2 900	3 520 000	29 000
D	3 520 000	29 000	Not defined	Not defined

Figure 2.5: Maximum action limits for viable particle contamination second GMP annex1 vol4 -Manufacture of Sterile Medicinal Products 25-11-2008.

Chapter 3

Hardware and Software Presentation

3.1 Viable and Non-viable (particle) monitoring system

Thanks to the modularity of the machine, it is possible to interface with external and optional devices, to expand the functionality of the machine itself. In this part we will introduce the particle counting and microbial sampling concerts as an optional module to the machine for monitoring the particles in the box after a decontamination, cleaning, process phase...

3.1.1 Viable particle monitoring system description

Figure 3.1: MiniCapt 100R Microbial Air Sampler and mounting bracket.

The MiniCapt Remote Microbial Air Sampler is a device used for microbial air sampling in cleanrooms and aseptic areas. It is designed to monitor the microbiological cleanliness of cleanrooms and aseptic areas in compliance with ISO Class 5/7, GMP Grade A/B standards . The sampler is equipped with the latest cleanroom monitoring technologies and can be customized to meet specific monitoring requirements . It is also designed to be mounted in tight spaces and has a small footprint, making it easy to install . The MiniCapt Pro model has a stainless steel exterior cabinet that protects the unit during cleaning and from harsh environments . The device can be fully integrated into the Particle Measuring System or other FMS via MODBUS communications.

Operation principle

Figure 3.2: MiniCapt 100R Microbial Air Sampler with BioCapt Impactor sampling.

For an efficient biological sample collection environment, a biocapt is installed on the inflet of the Minicapt,whose output is passed through the mass flow measurement sensor before exiting through the exhaust HEPA filter. The airflow into the biocapt impactor sampling is controlled by a mass flow controller , thus ensuring an operating flow rate of 25 L/min, 50 L/min or 100 L/min. This controller ensures all the fluid dynamic conditions are maintained throughout the sample, and allows for tubing and altitude corrections. The Minicapt communicate by means of Modbus TCP with facility pro, a software package that stores and displays sampled data, accessible by the user for a eventual control.

Basics configuration for Ethernet setting

The sampler should have the appropriate Ethernet settings, configured before it is wired to a network.This can be performed either before fluidic installation, if access to the back of the sensor is problematic, or after, if back panel access is not an issue Configuration is the process of entering Ethernet settings that will allow the MiniCapt Remote Microbial Air Sampler to communicate over a specific network.This configuration require the knowledge of terminal emulation software as Hyper-Terminal and the RS-232 cable or USB to RS-232 adaptor (if the PC does not have a serial port). The communication parameters are shown in the table 3.1

Baud Rate	9600
Data Bits	8
Parity	N
Stop Bits	1
Flow Control	off
Carrier Detect	off
Connector	Appropriate to hardware (COM1, for example)

Table 3.1: Terminal Emulator Communications Parameters

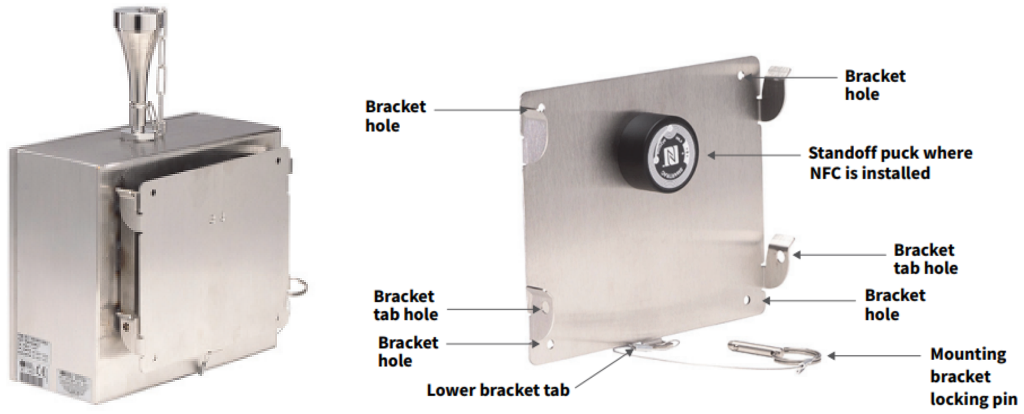
The important set of parameters to communicate with the devices on the network are :

* IP address: The IP address is used when communicating across networks.sets in the form of aaa.bbb.ccc.ddd.
* Mask : The mask separates the network address from the host address and is common to all devices on the logical network.
* Gateway address: used when communicating across different networks.
* Enter a data queue size in order to buffer data. Queue size (applicable for PMS Ethernet and Modbus TCP connections). Note: The RS-485 has its own queue fixed to 10.
* Flow rate
* Connection status (if used with Modbus)
* Operational mode (Sampling/Idle)

Figure 3.3: IsoAir Pro-Plus Particle Sensor.

3.1.2 Non Viable particle monitoring system description

A particle counter is a device used to monitor and diagnose particle contamination within specific clean media, including air, water, and chemicals. It is used in a variety of applications in support of clean manufacturing practices, such as electronic components and assemblies, pharmaceutical drug products, and medical devices. There are different types of particle counters available in the market, including aerosol particle counters, liquid particle counters, and solid particle counters. The type of particle counter we choose depends on the application and the medium you want to monitor. Aerosol particle counters are used to count and size particles in the air, while liquid particle counters are used to count and size particles in liquids. Aerosol particle counters use light scattering to detect particles, while liquid particle counters use light obscuration. The accuracy of particle counters is dependent on various factors, including the calibration of the instrument, the flow rate of the sample, and the size of the particles being measured. The IsoAir Pro-Plus remote particle counter manufactured by Particle Measuring Systems is a multi-channel remote particle counter, that is compact and easy to use. It is designed for minimal downtime and flexibility of multiple options. It can measure particulate contamination in up to 8 channels from 0.3 -25 µm and is resistant to Vaporized Hydrogen Peroxide (VHP), has an integrated HEPA filtered blower that reduces the need for extraneous external connections and supporting equipment. It can be connected to monitoring systems and software for data management of Particle Measuring Systems (PMS). Is powered by both 100 – 240 VAC and 24 VDC and compliant with global regulations and ISO standards. The device comes with a quick-release mounting bracket that stores essential sensor data such as the IP address at the sampling point, reducing time and facilitating unit installation after calibration or maintenance.



(a) mounting plate installed. (b) Mounting bracket

Figure 3.4: Particle Sensor and NFC Tag.

Basics configuration for Ethernet setting

The basic communication parameters for terminal emulation are the same as given in the table 3.1. As the Microbial air sampler, the TCP/IP addresses also need to be set before it can interface with other devices on the net. At the terminal emulator session's command prompt we configure the IP address, mask and gateway with additional setting for The NFC(Near Field Communication) programming. Note that all settings on the particle counter sensor are connected to the NFC tag. We set the number of samples to transmit in the queue in the [0,3000] interval. If 1 is set, the feature will be disabled.The samples is automatically save in RAM (up to 3000) and can be sent via TCP/IP. The IsoAir Pro-Plus particle sensor is set up for either a 2-channel pharmaceutical mode, that only outputs 0.5 and 5.0 µm channels on a configuration between 2 and 8 channels of various particle sizes.

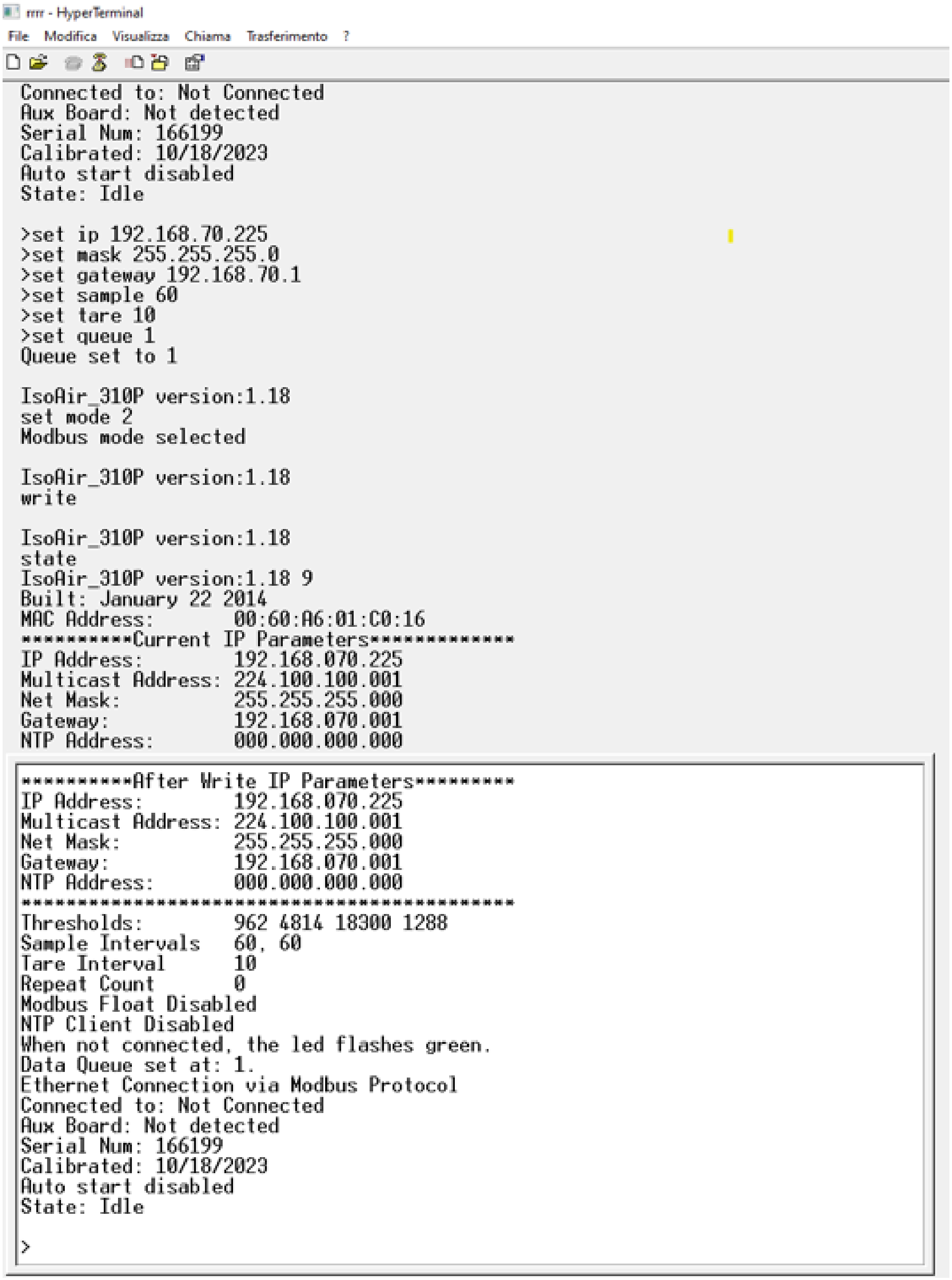


Figure 3.5: IsoAir Particle sample configuration.

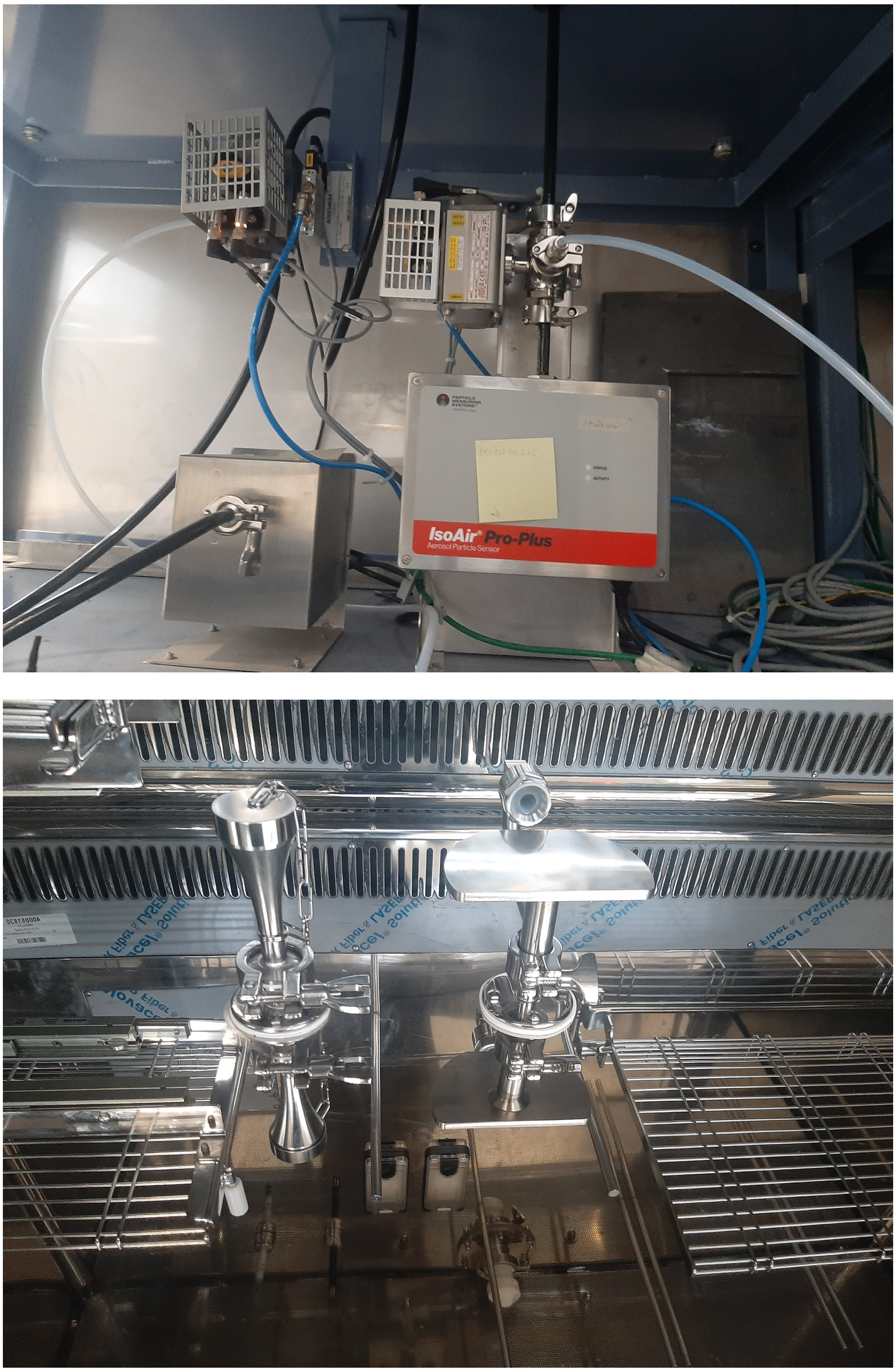


Figure 3.6: IsoAir Particle sample/Microbial installation.

3.2 Totally Integrated Automation(TIA) with PLC Simatic Step7-1200 and graphical interface presentation

Figure 3.7: Simatic 1200 and expansion modules (SIEMENS).

A Simatic S7-1200 controller is a type of programmable logic controller (PLC) manufactured by Siemens. It is designed for small to medium-sized automation solutions, with integrated technology and communication functions. The controller is available in both standard and failsafe versions. The basic controller is ideal for executing automation tasks in a flexible and efficient manner in the medium-low performance range. The scalable SIMATIC S7-1200 controllers offer integrated inputs and outputs, as well as communication options, and can be expanded in a modular fashion.

3.2.1 Simatic S7-1200 controller analysis

The S7-1200 PLC can be powered by direct or alternating current. As regards the direct current power supply, this is 24 VDC; if we want to power it directly with the 220V mains voltage we can choose the AC model which accepts voltages from 85 to 264V. Its outputs can be static or relay. The static outputs work only in direct current, while the relay ones are used to drive circuits that can have different voltages, both direct and alternating current. The power supply, the type of digital inputs and the type of outputs can be identified in the model codes as follows:

* AC/DC/Relay: alternating current power supply, direct current inputs and relay outputs.

* DC/DC/Relay: DC power supply, DC inputs and relay outputs.
* DC/DC/DC: DC power supply, DC inputs and solid state DC outputs.

From a hardware point of view, this PLC offers very different expansion possibilities depending on the CPU model we choose. At the low end of the family of these controllers we find the 1211C and 1212C CPUs, the latter only accepts two additional I/O modules, a signal module that can be installed on board, and three communication modules. Moving up the hierarchy of these controllers, we find at the top the 1217C central unit (the most performing) which accepts up to 8 additional input and output modules.

3.2.2 Programming the S7-1200 series controller

The S7-1200 is programmed within the Siemens TIA Portal. The installation of this programming environment requires a fair amount of resources on the computer, and the licenses for the software products we want to use must also be purchased. As regards programming languages, the Simatic S7-1200 can be programmed in ladder language, in function block language, and with structured text language (LAD; FBD and SCL). A peculiarity of this PLC is that it also accepts a signal card that can be installed directly on board, on the front of the unit. To put the PLC 1200 in communication with the outside world we can use the Ethernet network (via the on-board port), or install communication modules of different types, including those for RS232, RS422/485 serial communications, Profibus ones, the interfaces AS, and the Masters for the IO-Link system. We can also find GPRS and LTE communication modules for wireless communications.

3.2.3 Graphical interfaces for the S7-1200 plc

This plc is programmable within the TIA Portal environment, with which monitoring systems of varying complexity can also be designed using the WinCC software. Among the different options for graphic interfaces we find panels from the Basic and Advanced series, which can be easily integrated into the automation system based on this PLC. Programming can also be supported by the simulators that the TIA Portal makes available.

The web server integrated into the 1200 system.

Through the web server integrated into the units we can access the device with the web browser and carry out some functions, including viewing and controlling variables in the PLC, consulting the diagnostic buffer, or even accessing customized pages; all obviously in a local network or via the Internet, appropriately configuring the hardware of the systems.

Communication with the 1200 system via the open OPC UA standard.

With the advent of industry 4.0, the integration between factory areas and between local and remote areas requires a standard to refer to. Through the implementation of the OPC UA protocol, the Simatic S7-1200 allows secure communication, also including digital signatures and encryption using special certificates.

3.3 Modbus TCP protocol overview

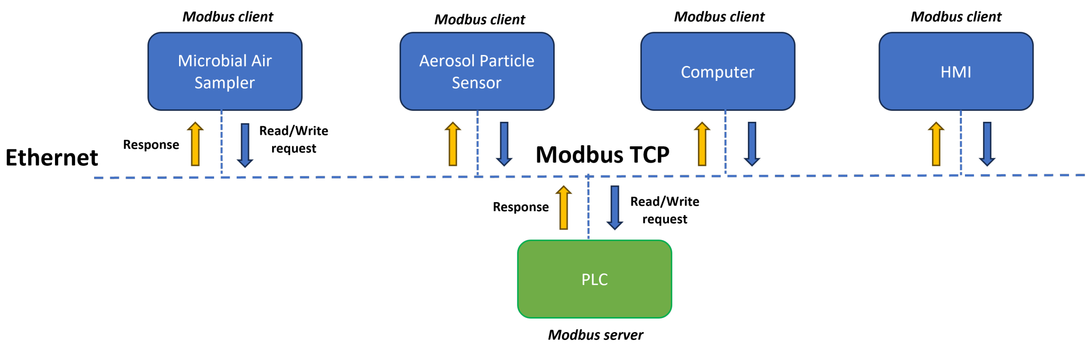


Figure 3.8: Modbus TCP Protocol.

Modbus TCP protocol is an industrial communication protocol that allows two or more devices to communicate with each other via an Ethernet network. ModbusTCP is actually the Modbus protocol (Or Modbus RTU) running over an Ethernet TCP/IP network.
Every control engineer today has to understand the Modbus TCP protocol. However, it doesn't mean the engineer needs to know every bit of the protocol. Instead, it's better to understand Modbus TCP uses and setup.
In this part, we will touch on the fundamentals of the Modbus TCP protocol, and in the next, we will dive into the real uses of this industrial protocol.
To explain Modbus TCP, we have first to discuss his ancestor, the serial Modbus.

Modbus presentation

The Modbus protocol was created by a company called Modicon in 1979. It is initially developed for Modicon PLCs and industrial automation systems. It has since become an industry-standard method for transferring data between industrial control devices.

Over the years, Modbus RTU has become very popular, and even though today it is considered an outdated protocol, we can still find it in many applications.

On a Modbus network, devices communicate using a master-slave (client-server) approach constraint for only one device, the master (client), to initiate transactions.The rest of the Modbus network devices act as slaves (servers) and respond with the desired action requested in the transaction, which can be sending the requested data back to the master.

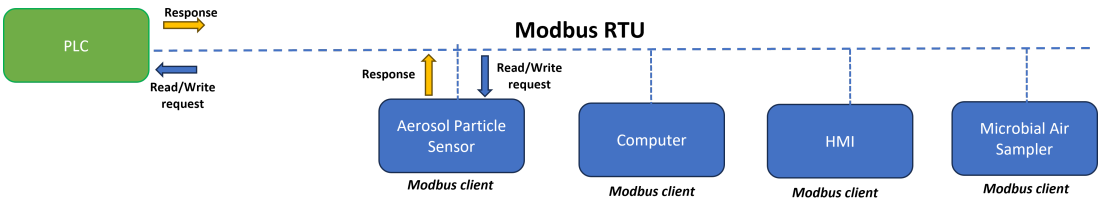


Figure 3.9: Modbus RTU Protocol.

Modbus TCP is basically the Modbus RTU protocol with a TCP interface running on Ethernet. Like Modbus RTU, Modbus TCP operates on a client/server principle, but in this case, the client (master) initiates both requests and responses from a server (slave). Any device can be a client or a server. Therefore, many devices are the same time clients and servers. A network can bring together many clients. At the same time, several clients can send requests, many servers can respond, and a client can talk to various servers. A server can simultaneously respond to several clients. Furthermore, Ethernet directs and ensures data delivery to all the devices at the same time. ModbusTCP protocol uses a 10 Mbps Ethernet standard to convey the entire structure of Modbus messages and offers fast communication to many devices in a single network.

To establish connections and data exchange between devices, the Modbus TCP messaging service must provide a listening socket on port 502. It is important to note that the listening port 502 TCP is reserved for Modbus communications. If the number of client and server connections exceeds the number of allowed connections (from 1 to 16), the oldest unused connection is shut. Access control mechanisms can also be activated to verify that the IP addresses of remote clients are authorized. By default, IP addresses that the user does not configure are barred from access when the security mode is enabled. A Modbus server module is responsible for receiving requests and implementing actions (reading and writing in particular) to respond to them. The execution of these actions is carried out in a completely transparent way for the application programmer.

Difference between Modbus TCP and Modbus RTU

The table below specifies the most impotent differences between Modbus RTU and Modbus TCP:

Modbus Protocol		
*	RTU	TCP
Physical Layer	Serial RS485 - RS422 - 2/4 wire	Ethernet, Cat5,6,7
Multi clients (Master)	Only one	Multi-Client Support
Multi Servers (Slave)	Supported	Supported
Speed	9600 - 19200 bps	10 Mbps
Distance	Limited, usually not more than 1000m	Unlimited with fiber and satellite

Table 3.2: RTU Vs TCP Modbus

3.3.1 Open Systems Interconnect (OSI) model for Modbus TCP

To better understand how Modbus TCP/IP is structured, we need to review (OSI) Model illustred with the table 3.3
The top layers, 5-7, are usually united to form the fifth layer, the Application layer. When Modbus is stated as the application layer, we get the Modbus TCP. The first four layers are the same for every Ethernet-based protocol.

3.3.2 Application layer

There are a few protocols in the application layer, For example, FTP, DNS, SMPT, HTTP, and more. Each one of these protocols has its specific purpose. For Modbus TCP, the primary application layer protocol of interest is Modbus.
Modbus functions operate on memory registers to monitor and control devices on the network. Manufacturers of Modbus devices usually publish a Register Map as the case of Spectris company of Particle Measuring Systems. Before trying to communicate with a device, we should refer to the register map for the specific device to understand its operation.

Layer	Layer Name	Description	Protocols
7	Application	The application layer is exposed to the end-user like web browsers or applications.	**Modbus**
6	Application	The Presentation layer prepares the translation of the application data and format into network format (encryption/description of data for secure transmission).	
5	Session	A session occurs between two devices.This layer connects the application to the network.	
4	Transport	layer is also called the Transmission Control Protocol or TCP. The TCP is built on top of the Network layer (IP)	TCP
3	Network	the network layer (IP) is where packet forwarding (routing), setup and creation of connections occur	IP, ARP, RARP
2	Data Link	The data link layer is responsible for data transfer.	Ethernet, CSMA/CD, MAC
1	Physical	The Physical layer is where physical components connect. This includes the cables, sufficient voltages and frequencies.	Ethernet Physical layer - Cat5,6,7 for example

Table 3.3: Modbus Layer Description

Modbus data model structure consists of four basic data types:

* Discrete Inputs – 0xxxx.
* Coils (Outputs) – 1xxxx.
* Input Registers (Input Data) – 3xxxxs.
* Holding Registers (Output Data) – 4xxxx.

The service request (Modbus Protocol Data Unit) comprises a function code and additional data bytes, depending on the function. In most cases, the additional data is usually a variable reference, such as a register address, as most Modbus functions operate on registers.

Modbus functions and registers

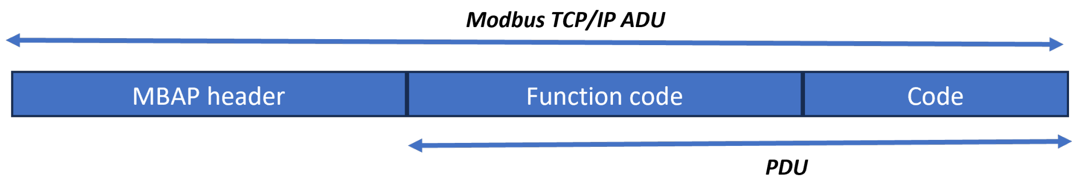
The client (master) provides the server (slave) with additional information required by the slave to complete the operation specified by the function code. The request data typically includes register addresses, offset values, and data to be written.

* Read Coil Status (01) – This command will read the ON/OFF status of discrete outputs or coils (0xxxx reference addresses) in the slave/server.
* Read Holding Registers (03) – Reads the binary contents of holding registers (4x reference addresses) in the slave device.
* Read Input Registers (04) – This command will read the binary contents of Input registers (3x reference addresses) in the slave device.
* Force Single Coil (05) – Forces a single coil/output (0x reference address) ON or OFF.
* Preset Single Register (06) – This command will preset a single holding register (4x reference address) to a specific value.
* Force Multiple Coils (15) – Simultaneously forces a series of coils (0x reference address) either ON or OFF.
* Preset Multiple Registers (16) – Presets a block of holding registers (4x reference addresses) to specific values.
* Report Slave ID (17) – This command returns the model, serial, and firmware.

For a function to work, one must specify the starting register and quantity of registers to be read/write. When the server device responds to the client, it uses the function code to indicate either a normal response or an exception response if an error has occurred. A normal response simply echoes the original function code of the query, while an exception response returns the original function code with its most significant bit set to '1'.

Modbus Application Data Unit (ADU) format

The Modbus protocol defines the PDU (Protocol Data Unit), independent of other communication layers. The Encapsulation of the protocol Modbus over TCP / IP introduces an additional field (MBAP Header) at the unit level of application data or ADU (Application Data Unit).
The client (who initiates the Modbus transaction) builds the ADU. In the same way, the Function field PDU code indicates to the server the action to



take. Upon receiving a client request, the server analyzes the MBAP header of the Modbus ADU. If this corresponds to a Modbus header, a Modbus transaction is instantiated.When the maximum number of simultaneous transactions allowed exceeds, the server constructs an exception response (Exception Code 6: Server Busy).Otherwise, a transaction is instantiated and initialized with the following information:

The identifier of the TCP connection used (provided by the TCP module Management). The Modbus transaction ID is contained in the MBAP header. The Unit Identifier is also contained in the MBAP header. Then, it is the turn of the PDU field to be analyzed. The server is ready to perform the requested service if this is recognized as a valid field.

Once the request has been processed, the Modbus server builds a response that must be sent to the TCP Management module. Depending on the treatment outcome, two types of responses are possible: a positive response or an exceptional response, whose objective is to provide the customer with information concerning the detected errors while processing the request.

The PDU of the Modbus response must be prefixed with an MBAP header built from the information stored when receiving the request: Unit Identifier, Protocol Identifier, and Transaction Identifier. The server also provides information, the "Length" field, indicating the length of the PDU + Unit Identifier field. Finally, the Modbus response is sent to the client. Upon receiving the response, the Modbus client uses the Transaction Identifier to find the corresponding request sent over the TCP connection. The response is ignored and discarded if this does not match any query known to the client. Otherwise, the Modbus client analyzes and uses the response to build the confirmation, which will then be sent to the application.

The response analysis consists of checking the MBAP header and the PDU of the response Modbus. Suppose this comes from a Modbus server directly connected to the TCP / IP network.

In that case, identifying the TCP connection is sufficient to identify the remote server without ambiguity. Thus, in this case, the Unit Identifier is not meaningful and should be ignored.

On the other hand, if the server is connected to a network, and the response comes from a router, the Unit Identifier (which in this case will not be equal to 0xFF) must be taken into account for the identification of the server by the client.

Regarding the PDU analysis of the response, the client checks the compliance of the Function Code as well as the format of the reply. // Even though the Modbus protocol itself is standard, the contents of the registers are application-specific. A description of the register map is the definitive specification for that application interface. The following Modbus register map definition has comments and notes to help with its intended us in code development.

Input Registers		Description (Configuration)	Comment	Notes
30001	unary	Modbus Map Version	Version 1.00 (Static 100)	
30002	unary	Sensor Firmware Version	Encoded Firmware version	
30003	unary	Product Name: Char 00, 01	Product Name	
30004	string	Product Name: Char 02, 03	Product Name	
30005	string	Product Name: Char 04, 05	Product Name	
30006	string	Product Name: Char 06, 07	Product Name	
30007	string	Product Name: Char 08, 09	Product Name	
30008	string	Product Name: Char 10, 11	Product Name	
30009	string	Product Name: Char 12, 13	Product Name	
30010	string	Product Name: Char 14, 15	Product Name	
30011	unary	Scale: Flow Rate	Multiplier in fixed: Flow Rate	2 (100)
30012	unary	Scale: Volume	Multiplier in fixed: Volume	5 (100000)
30013	dual	Flow Rate (high)	Flow * 10 Scale	Float Mode
30014	dual	Flow Rate (low)	Flow * 10 Scale	Float Mode
30015	unary	Number of Particle Channels	Fixed number of sizes	0 fixed
30016	unary	Number of Analog Channels	Fixed number of analogs	3 fixed
30151	unary	Sensor Firmware Version	Revision/Build	
30152	unary	Sesnor Firmware Version	Minor/Major	

Figure 3.10: Input register-Data Packet 1.

Input Registers		Description (Data Packet)	Comment	Notes
30201	unary	Calibration Date (Month)	Calibration Date	
30202	unary	Calibration Date (Day)	Calibration Date	
30203	unary	Calibration Date (Year)	Calibration Date	
30204	string	Serial Number: Char 00, 01	Serial Number	
30205	string	Serial Number: Char 02, 03	Serial Number	
30206	string	Serial Number: Char 04, 05	Serial Number	
30207	string	Serial Number: Char 06, 07	Serial Number	
30208	string	Serial Number: Char 08, 09	Serial Number	
30209	string	Serial Number: Char 10, 11	Serial Number	
30210	string	Serial Number: Char 12, 13	Serial Number	
30211	string	Serial Number: Char 14, 15	Serial Number	
30212	unary	Device Status (high)	Device Status Mask	
30213	unary	Device Status (Low)	Device Status Mask	
30214	unary	Device State	Device State	
30215	unary	Number of data samples in queue	Number of samples in queue	
30216	dual	Time Stamp (high)	time_t	
30217	dual	Time Stamp (low)	time_t	
30218	dual	Sample Time (high)	Seconds * 100	Float Mode
30219	dual	Sample Time (low)	Seconds * 100	Float Mode
30220	dual	Data Packet Status (high)	Bit Mask	
30221	dual	Data Packet Status (low)	Bit Mask	
30222	dual	Flow Rate (high)	Flow * 10 Scale	Float Mode
30223	dual	Flow Rate (low)	Flow * 10 Scale	Float Mode
30224	dual	Volume (high)	Volume * 10 Scale	Float Mode
30225	dual	Volume (low)	Volume * 10 Scale	Float Mode

Figure 3.11: Input register-Data Packet 2.

Coils	Description (Coils)		Comment	
00/01	Data Collection	On/Off	Sampling Control	
00/02	Data Available	Yes/No	Data Available/Queue Control	
00/03	Data Clear	Toggle	Data Queue Delete	
00/04	Reset Sensor	Toggle	Reset control	
00/05	Status LED: Green	On/Off	Status LED: Green Control	n/a
00/06	Status LED: Red	On/Off	Status LED: Red Control	n/a
00/07	External Alarm	On/Off	External alarm	n/a
00/08	IEEE-754 Float	On/Off	Modbus Float representation	
00/09	Volume Mode	On/Off	Sample Volume or Interval	
Spare	23 spare coils		Reserve “common” coils	

Figure 3.12: Coils.

Holding Registers		Description (Setup)	Comment
40001	dual	Real Time Clock (high)	time_t
40002	dual	Real Time Clock (low)	time_t
40003	unary	Sample Interval/Volume	Seconds
40004	unary	Hold/Tare Time	Seconds
40005	unary	Repeat Count	Repeat Count
40006	unary	Delay Time	Seconds
Spare	unary	26 spare registers	Reserve future “common” registers

Figure 3.13: Holding registers.

Device Status Entries	0x0001	Data Collection	Matches the common coil settings
	0x0002	Data Available	
	0x0004	Data Clear	
	0x0008	Reset	
	0x0010	Green Status	
	0x0020	Red Status	
	0x0040	External Alarm	
	0x0080	IEEE Float mode	
	0x0100	Volume Mode	
Device State Entries (LSB)	0	Idle	
	1	Sampling	
	2	Maintenance	n/a
	3	Error	
Device SubState Entries (MSB)	0	Idle	Normal
	0	Sampling	Normal
	1	Sampling	Hold/Tare
	0	Error	Flow
Data Packet Status Entries Low register	0x0001	Laser Good	
	0x0002	Flow Good	
	0x0004	Hardware Good	n/a
Data Packet Status Entries High register	0x0001	Battery	Specific to MiniCapt
	0x0002	Pressure	Specific to MiniCapt
	0x0004	Hi/Lo	Specific to MiniCapt

Figure 3.14: Associated values for specific registry entries.

Chapter 4

Software Realisation and Parameters Monitoring

In this chapter, we will discuss in detail the logic behind each device to create our software, and directly introduce the names of the variables used to create the graphical interface. This work will be able to interest us in some functions of the machine itself, given that the interfacing and correct functioning of the particle counter and microbial requires, that the machine is in a precise state with the validation of interlocks.

4.1 NON-VIABLE PARTICLE COUNT

Premise:

The particle count sensor (PMS) IsoAir Pro-Plus, provides an analogue signal which is equivalent to the number of particles sampled in the last minute. This data is updated every second. By recording the data every 60 seconds we automatically have information on the number of particles acquired from minute to minute.
Once the isolator enters the Production state, the particle count is enabled for sampling. It is possible to define two operating modes: automatic and manual, selectable via a specific parameter on the HMI.

4.1.1 Automatic mode

Following activation of the Production state, the valve that connects the particle count and Process Chamber, opens and the particle count is activated after 2 seconds. After a time equal to START DELAY the actual sampling starts with acquisition of values, and then stops on its own when the target volume is reached. The delay is designed for those who want to let the count and duct clean before actual sampling (similar behavior to validation counts). At the end of the sampling the counting is stopped and after 2 seconds the valve closes.

Number of particles in the sample

Knowing that the particle counting pump guarantees a fixed air flow F, it is possible to determine the number of samplings necessary to cover a target volume V. This volume is typically equal to $1m^3$. By expressing V in cubic meters (m^3) and F in litres/minute (L/min), we find out for how many minutes (or how many samplings) T the pump must acquire to reach the target volume:

$$T = \frac{V * 1000}{F} \tag{4.1}$$

(we always round T up). Every minute the PLC acquires the P_i value of particles and performs the following calculation:

$$C_i = T\frac{\sum_i P_i}{i} \tag{4.2}$$

where C_i is the particle concentration calculated after the first samples and is displayed on the NON VIABLE PARTICLE COUNTER STATUS page in the PARTICLES IN SAMPLE 0.5/5um (particles) fields. This value cannot be considered definitive: it represents the number of particles that the target volume would have, on the assumption that the samplings performed are on average representative of the entire volume, consequently in the event that the Production state is exited before T minutes the unreliability of sampling is signaled with the NV PART alarm. COUNTER – SAMPLE NOT COMPLETED and the report result will be NULL.
The sampled volume is displayed on the monitoring page, updated every minute and is simply calculated with the formula:

$$V_i = i * F \tag{4.3}$$

At the end of T minutes the counting stops automatically. When the target volume has actually been sampled is the final C_i value, i.e. C_T, compared with the THRESHOLD 0.5/5um threshold and this comparison determines the test result. The result, which can be viewed on the report, will be one of the following three:

* NULL, if the sampling takes less time than T minutes (it is possible by exiting the production state) or one of the critical alarms for air monitoring is activated: valve, device itself, inflate of the process chamber seals.

* PASSED, if at the end of T minutes C_T is $<$ THRESHOLD 0.5/5um.
* FAILED, if at the end of T minutes C_T is $>$ THRESHOLD 0.5/5um; in this case the NON VIABLE PARTICLES IN SAMPLE 0.5/5um OVER LIMIT alarm is triggered.

Concentration per cubic foot

The particle count flow is fixed at 1 $ft3/min$. Consequently, by drawing the data every minute we directly have the concentration per cubic foot which we display in the PARTICLE COUNT 0.5/5um field.
The concentration per cubic foot does not determine the result of the air monitoring, but must only give an indication to the operator of what is happening during Production.
The comparison between the value of PARTICLE COUNT 0.5/5um, updated every minute, and the ALARM THRESHOLD threshold for 0.5/5um particles , gives rise to the NON VIABLE PARTICLES 0.5/5um ALARM alarm, while the comparison with the PREALARM THRESHOLD threshold gives rise to the NON VIABLE PARTICLES 0.5/5um PREALARM alarm.
In conjunction with the generation of the alarm, on the count monitoring page, the PARTICLE COUNT 0.5/5um field will be colored red or yellow depending on the threshold exceeded. The alarm will recognize itself if in the following minute the value falls below the threshold.
All cubic feet sampled during the count acquisition are averaged and the result is included in a report. In this way you are informed of how many particles were generated on average in one cubic foot: this data must represent the ”average” cleanliness state of the air inside the isolator during the process. Furthermore, the most recent minute in which the highest peak of particles occurred and the value is recorded, so that the operator can trace any erroneous operations committed: in the case of several minutes with the same ”maximum” value of particles we prefer the most recent one since it is believed that when sampling starts there is a greater probability of reading higher concentrations due to any residues.

4.1.2 Manual mode

Following activation of the Production state, after a time equal to START DELAY in seconds, the particle count is available through the appearance of a manual start button on the Production page. As soon as the button is pressed, the valve that connects the counter and isolator opens and after 2 seconds the counter starts sampling, only to stop when the operator presses the stop button. This key becomes available only after the count has sampled a volume equal to the target volume. At the end of the sampling the counting is stopped and after 2 seconds the valve closes.

Number of particles in the sample

The algorithm is the same as in automatic mode. The substantial difference is that once the target volume has been reached the count can continue sampling, consequently to always have the number of particles present in the target volume as a result the value of C_i must be calculated exactly on a number of samples equal to T, no more: if the count has reached sampling for $T + 1$ minutes, the calculation of C_i must be performed for $i = 2, ..., T + 1$, and so on. In this way C_i represents the number of particles present in the last sampled target volume and the particles seen in the first minutes do not influence the last measurements. The test result (Passed or Failed) is available only after sampling a volume equal to the target volume, as is the NON VIABLE PARTICLES IN SAMPLE 0.5/5um OVER LIMIT alarm. The rest of the operation remains identical.

Concentration per cubic foot

Operation is identical to that described in session 3.1.1. In this case the average of particles per cubic foot is performed on all individual samples, without worrying about the target volume.

4.1.3 Parameters and Alarms

Parameters on HMI – NON VIABLE PARTICLE COUNTER - SETUP page

HMI Parameters				
Name	Description	Default value	Unit of measure	Range
Enable Devive	Enables the use of counts for air monitoring during the Production phase	0	N/A	
Device Flow	Particle count sampling flow (F)	30	l/min	
Target Volume	Volume (V)	1	m^3	0.1 - 5
Start Delay	Delay in starting sampling after entering production (only for automatic mode)	25	s	0 - 600
Enable Manual Mode	Automatic (0) or manual (1) mode	0	N/A	
Threshold 0.5um	Threshold that determines success/failure of air monitoring tests for 0.5um particles	3520	P	
Threshold 5um	Threshold that determines success/failure of air monitoring tests for 0.5um particles	20	P	
Alarm Threshold $>$0.5um	Threshold that determines alarm and related warning of the presence of 0.5um particle concentration in the sampled cubic foot	100	p/ft^3	
PreAlarm Threshold $>$0.5um	Threshold that determines alarm and related warning of the presence of 0.5um particle concentration in the sampled cubic foot	50	p/ft^3	
Alarm Threshold $>$5um	Threshold that determines alarm and related warning of the presence of 5um particle concentration in the sampled cubic foot	3	p/ft^3	
PreAlarm Threshold $>$5um	Threshold that determines alarm and related warning of the presence of 5um particle concentration in the sampled cubic foot	2	p/ft^3	

Table 4.1: Non Viable Particle Counter - Setup page

HMI Alarms			
Name	Reference variable	Threshold parameter	Alarm activation condition
NV PART.COUNTER-SAMPLE NOT COMPLETED	Sampled volume V_i	Target Volume V	At the end of sample $V_i < V$
NONVIABLE PARTICLES IN SAMPLE 0.5um OVER LIMIT	Concentration of 0.5um particles per cubic meter C_T	THRESHOLD 0.5um	At the end of sampling $C_T >$ THRESHOLD 0.5um
NONVIABLE PARTICLES IN SAMPLE 5um OVER LIMIT	Concentration of 5um particles per cubic meter C_T	THRESHOLD 5um	At the end of sampling $C_T >$ THRESHOLD 5um
NONVIABLE PARTICLES 0.5um ALARM	Concentration of 5um particles per cubic foot	ALARM THRESHOLD $>$ 0.5um	Every minute results $P_i >$ ALARM THRESHOLD $>$ 0.5um
NONVIABLE PARTICLES 0.5um PREALARM	Concentration of 0.5um particles per cubic foot	PREALARM THRESHOLD $>$ 0.5um	Every minute results $P_i >$ PREALARM THRESHOLD $>$ 0.5um
NONVIABLE PARTICLES 5um ALARM	Concentration of 5um particles per cubic foot	ALARM THRESHOLD $>$ 5um	Every minute results $P_i >$ ALARM THRESHOLD $>$ 5um
NONVIABLE PARTICLES 5um PREALARM	Concentration of 5um particles per cubic foot	PREALARM THRESHOLD $>$ 5um	Every minute results $P_i >$ PREALARM THRESHOLD $>$ 5um
NV PART.COUNTER-COMMUNICATION ERROR	N/A	N/A	Enable device = 1, the pump is interrogated by the PLC but there is no communication

Table 4.2: Non Viable Particle Counter - Alarm Setup

4.1.4 Report generation and importance for pharmaceutical companies

Pharmaceutical companies are among the most demanding when it comes to Quality Control. This applies both to drugs and chemical substances, following national and international safety regulations, but also to packaging, which is equally of fundamental importance. Certifying and keeping track of the quality of the services provided by a company is a very important step in establishing itself in its reference sector. This concept takes on a particularly high value for acquiring the knowledge and skills necessary to carry out quality control on the product. In short, in order for the production processes to respect the quality criterion, as well as the current regulations (GMP: Good Manufacturing Practices) and align with the ISO directives, a very important document to certify the company's compliance with the guidelines is reproduced, through operational techniques and activities implemented to satisfy these requirements.
For this particular reason, at the end of our test a CSV file that constitutes the report is populated by the following fields:

* Production batch.
* User logged into the HMI who starts the Session (in automatic mode) or the test (in manual mode).
* Date and time of start and end of test.
* Sampled volume (V parameter)
* Number of particles seen in the "dirtiest" sample ("particles" unit of measurement) and relative minute: this data makes sense only if the sampled volume is at least equal to V, therefore we insert it in the report only if the number of samplings performed is greater than or equal to T; otherwise we enter " – ", both for the maximum value and for the minute. From this it follows that the possible results for the minute are values greater than or equal to T.
* Average particle concentration per cubic foot (unit of measurement "particles/ft^3").
* Most recent minute in which the maximum concentration of particles per cubic foot was recorded and the corresponding value.
* Test result.
* Generic notification of alarms appearing during sampling.

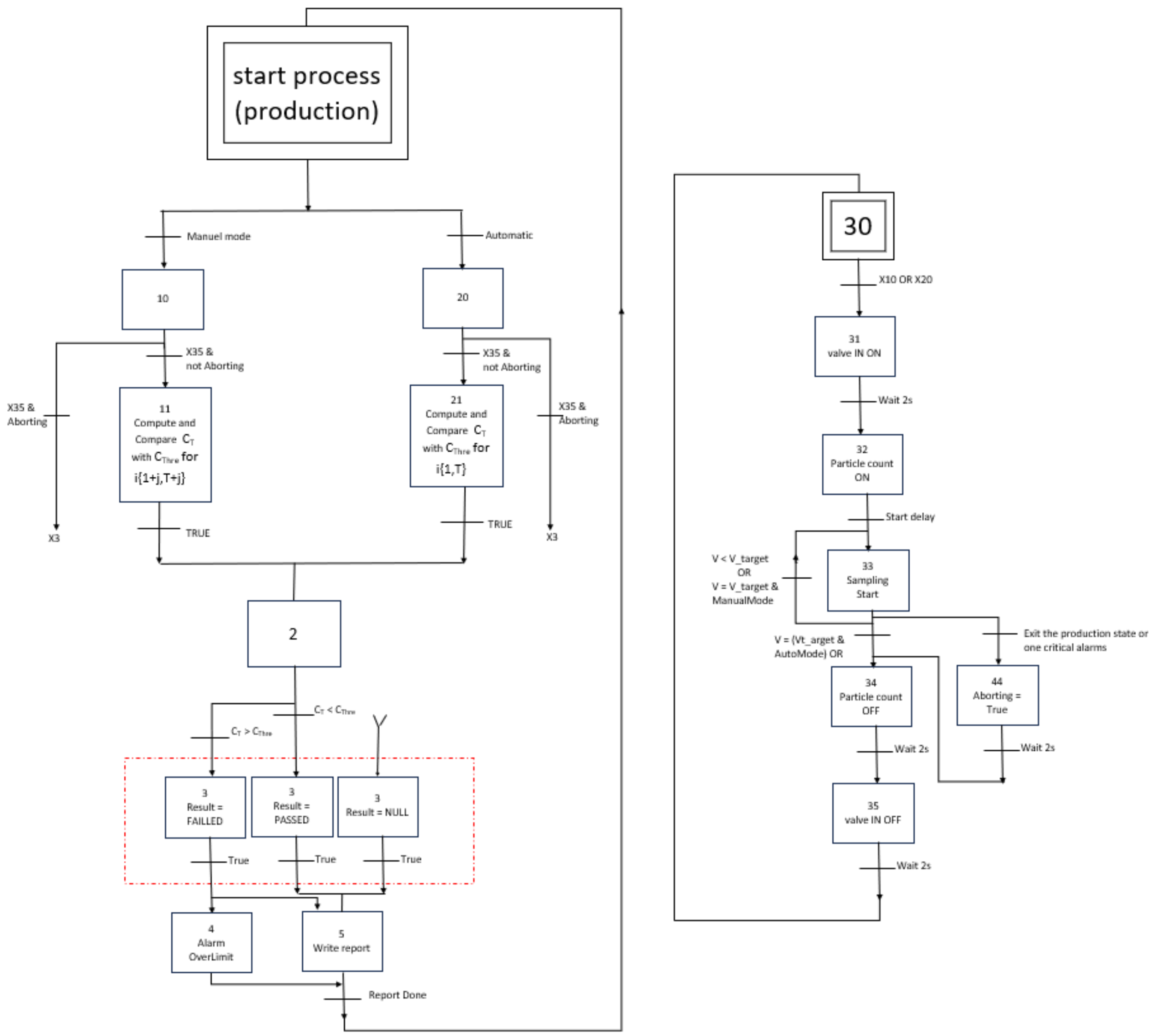


Figure 4.1: Non Viable: State evolution of control program.

4.2 VIABLE PARTICLE COUNT

Premise:

The particle count MiniCapt 100R Microbial Air Sampler (PMS) makes all the necessary data available to the PLC: real-time feedback on the flow with which it is sucking in the air and the volume sampled up to that moment. Once the isolator enters the Production state, the particle count is enabled for sampling. It is possible to define two operating modes: automatic and manual, selectable via the specific Automatic start in Production parameter.

4.2.1 Automatic mode

Following activation of the Production state, the valve that connects the particle count and the Process Chamber opens, and the particle count, after a delay necessary for communication, starts sampling, and then stops on its own when reach the target volume TARGET VOLUME (m^3). At the end of the sampling the counting is stopped and after 2 seconds the valve closes. If we want to repeat the sampling within the same Production batch, we can set the REPEAT parameter variable to be different from 0: in this way, a sampling identical to the previous one, will start after a time equal to the SAMPLE DELAY parameter.

4.2.2 Manual Mode

Operation is the same as in automatic mode, the only difference is that the initial action by the operator becomes necessary. Upon entering Session the main screen will show, after a delay equal to START DELAY, the Start V-Sampling button, which when pressed by the operator opens the valve and after a delay due to communication starts sampling. As soon as it reaches the target volume, the counting stops automatically resulting in the valve closing. In the case of REPEAT other than 1, after a time equal to SAMPLE DELAY, a sampling identical to the previous one automatically starts.

4.2.3 Parameters and Alarms

Parameters on HMI – VIABLE PARTICLE COUNTER - SETUP page

HMI Parameters				
Name	Description	Default value	Unit of measure	Range
Enable Devive	Enables the use of counts for air monitoring during the Production phase	0	N/A	
Device Flow	Particle count sampling flow (F)	25	l/min	
Target Volume	Volume (V)	1	m^3	0.1 - 5
Sample Delay	Delay Between Two Successive Samples	30	s	0 - 600
Enable Manual Mode	Automatic (0) or manual (1) mode	0	N/A	
Repeat	Number of target volume sampling to be carried out in the same Production cycle	1		1 - 5
IP1,2,3,4	Communication parameters: IP address	192.168.70.2		
Port	Communication parameter: port	502		

Table 4.3: VIABLE PARTICLE COUNTER - SETUP page

HMI Alarms			
Name	Reference variable	Threshold parameter	Alarm activation condition
V.PART. COUNTER-SAMPLE NOT COMPLETED	Sampled volume V_i	Target Volume V	At the end of sample $V_i < V$
V.PART. COUNTER – COMMUNICATION ERROR			Enable device = 1, the pump is interrogated by the PLC but there is no communication.

Table 4.4: VIABLE PARTICLE COUNTER - ALARM SETUP

4.2.4 Report Generation

At the end of the test a CSV file that constitutes the report is populated by the following fields for the same reason explained in 4.1.4:

* Production batch.
* User logged into the HMI who starts the Session (in automatic mode) or the test (in manual mode).
* Date and time of start and end of test.
* Volume sampled in each step (replicated for the number of samples)
* Generic notification of alarms appearing during sampling.

The implementation of the program also requires knowledge of the production principle of the machine, in order to create interlocks for the transition from one work phase to another, without creating conflicts, thus avoiding the generation of alarms and the interruption of the entire production process.

4.3 modbus interfacing

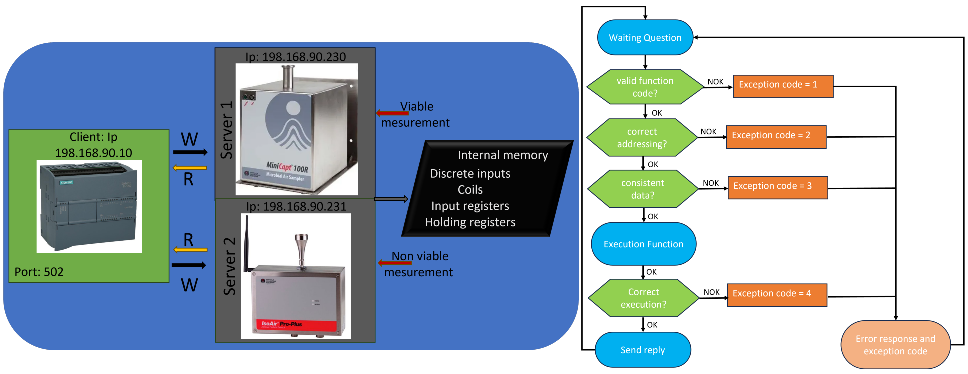


Figure 4.2: Communication server-client via Modbus overview.

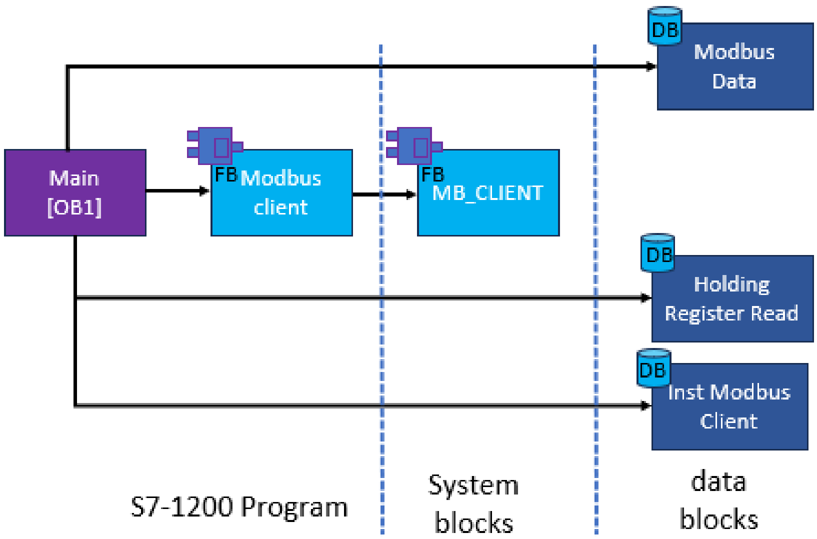


Figure 4.3: An overview of the Modbus functions implemented during elaboration.

The Modbus TCP communication between S7-1200 CPUs and Particle Measuring Systems (PMS) is presented in fig 4.2. The actual communication occurs via the MB CLIENT instruction that are called and parameterized in the PMS program of the S7-1200 CPU. The MB CLIENT instruction communicates as a Modbus TCP client via the PROFINET connection of the S7-1200. A connection can be established between the client and the server, requests can be sent and responses received, and the connection or disconnection of the Modbus TCP server can be controlled.

4.3.1 Modbus TCP-client

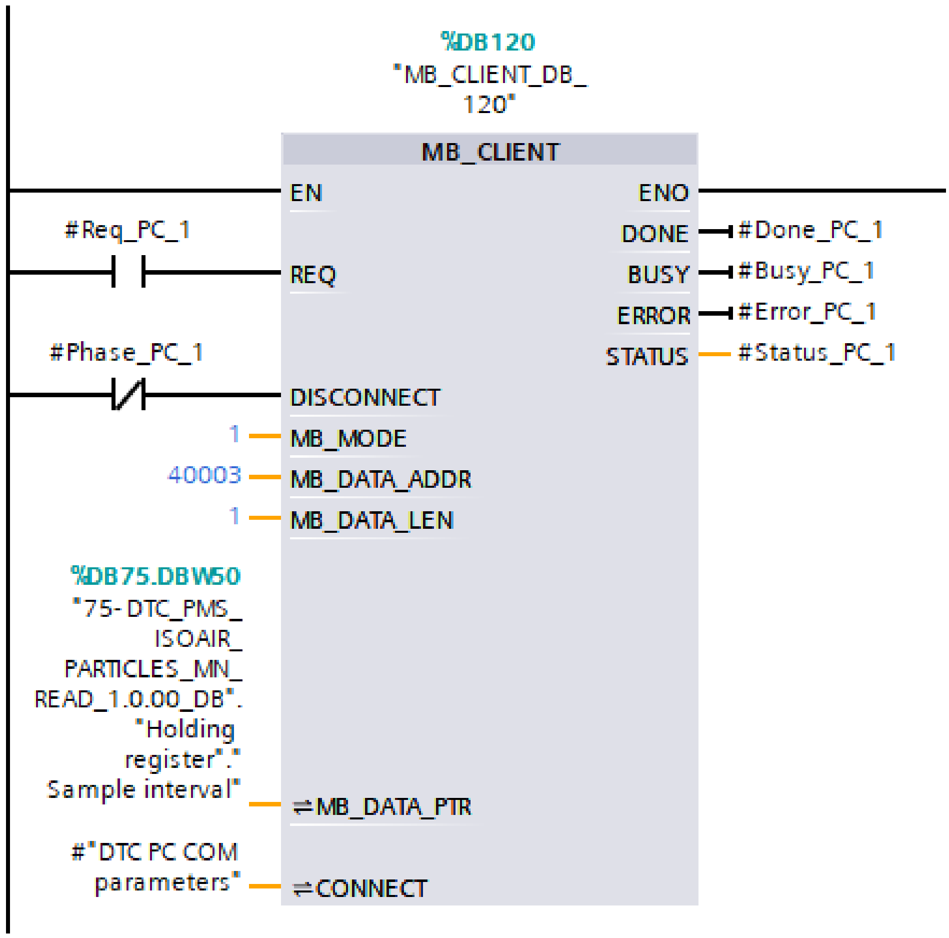


Figure 4.4: Modbus client.

In the Modbus TCP program, the MB CLIENT instruction is called for each Modbus TCP connection with a unique ID and separate instance data block. The call of the MB CLIENT instruction is done each time in a separate function. In this elaboration, the Modbus TCP connection with connection number = 1 is established to port 502 of the Modbus TCP server.the TCP server has the IP address 192.168.90.10. One holding registers are written on the remote address 40003 that correspond to an Unary sample interval/volume internal the holding registers of the particle.For this we set the input parameters MB MODE, MB DATA ADDR and MB DATA LEN respectively 1, 40003, 1.

4.3.2 Input and Output Parameters of the ”MB CLIENT” Instruction

Input parameters

The ”MB CLIENT” has the following input parameters. 4.5, 4.6

Parameter	Data type	Description
REQ	BOOL	Modbus request to Modbus TCP server As long as REQ = true, the instruction sends communication requests. • After starting the Modbus request, the instance DB is blocked for other clients. • Changes to input parameters take effect only after a server response is received or an error message is issued. • If the REQ parameter is set again while a Modbus request is in progress, no further transmissions will be performed.
DISCONNECT	BOOL	The parameter allows you to control the activation/deactivation of the connection to the Modbus server: • 0: Establish communication connection to the specified IP address and port number configured in the CONNECT parameter. • 1: Disconnection communication connection. No other functions are performed while the connection is deactivated. After a successful logout, the value 0003 is output in the STATUS parameter. If the REQ parameter is set during connection establishment, the Modbus request is sent immediately.

Table 4.5: MB CLIENT Input parameters 1

Parameter	Data type	Description
MB MODE	USINT	Selection of the Modbus request mode (read, write or diagnostics) or direct selection of a Modbus function.
MB DATA ADDR	UDINT	Initial address of the data which the ”MB CLIENT” instruction accesses. It depends on MB MODE.
MB DATA LEN	UINT	Data length: number of bits or words for the data access.
MB DATA PTR	VARIANT	Pointer to the Modbus data register. The register is a buffer for the data received from or to be sent to the Modbus TCP server. The pointer must refer to a global data block with standard access. In this example the pointer refers to DB75.

Table 4.6: MB CLIENT Input parameters 2

Output parameters

The ”MB CLIENT” instruction has the following output parameters 4.7.

Parameter	Data type	Description
DONE	BOOL	The bit at the DONE output parameter is set to ”1” as soon as the last job has been executed without error.
BUSY	BOOL	• 0: No ”MB CLIENT” job being processed. • 1: ”MB CLIENT” job being processed. The BUSY output parameter is not set upon connection establishment and disconnection.
ERROR	BOOL	• 0: No error. • 1: Error occurred. The cause of the error is displayed by the STATUS parameter.
STATUS	WORD	Error code of the instruction.

Table 4.7: MB CLIENT Output parameters

4.3.3 Parameters MB MODE and MB DATA ADDR Vs Function Code

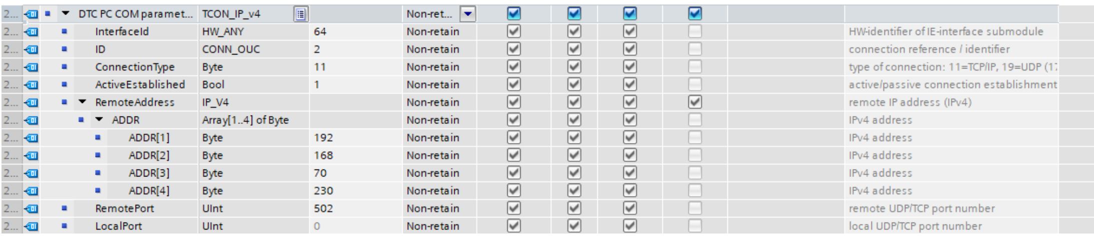
The ”MB CLIENT” instruction uses the ”MB MODE” instead of a function code. We use the ”MB DATA ADDR” parameter to define the Modbus start address which we wish to access. The combination of the ”MB MODE” and ”MB DATA ADDR” parameters defines the function code that is used in the current Modbus message. The table 4.8 shows the relationship between the ”MB MODE” parameter of the Modbus function used in this example and the address area.

MB MODE	Modbus Function	Data length	Function and Data Type	MB DATA ADDR
0	03	1 to 125	Read holding register 1 to 125 WORDs per call	40001 to 40006
1	06	2 to 123	Write Single Holding Register 2 to 123 WORDs per call	40001 to 40006

Table 4.8: Relationship between the MB MODE parameter of the Modbus function

4.3.4 Data Structure at the Parameter ”PC COM parameters”

We use the following structure for connection description according to TCON IP v4 for programmed connections at the CONNECT input.



DTC PC COM paramet...	TCON_IP_v4		Non-ret...	
InterfaceId	HW_ANY	64	Non-retain	HW-identifier of IE-interface submodule
ID	CONN_OUC	2	Non-retain	connection reference / identifier
ConnectionType	Byte	11	Non-retain	type of connection: 11=TCP/IP, 19=UDP (1
ActiveEstablished	Bool	1	Non-retain	active/passive connection establishment
RemoteAddress	IP_V4		Non-retain	remote IP address (IPv4)
ADDR	Array[1..4] of Byte		Non-retain	IPv4 address
ADDR[1]	Byte	192	Non-retain	IPv4 address
ADDR[2]	Byte	168	Non-retain	IPv4 address
ADDR[3]	Byte	70	Non-retain	IPv4 address
ADDR[4]	Byte	230	Non-retain	IPv4 address
RemotePort	UInt	502	Non-retain	remote UDP/TCP port number
LocalPort	UInt	0	Non-retain	local UDP/TCP port number

Figure 4.5: TCON IP v4 data structure parameter.

4.3.5 Parameter ”Data Buffer”

In this part we specify the data area for storing the data that is sent to the Modbus TCP server.The data that is written to the holding register of the Modbus TCP server is stored in the Block DB75 ”PMS ISOAIR PARTICLES MN READ 1.0.00”.

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static			☐	☐	☐	☐	☐		
2	Particles/m^3 (0,5um)	Array[0..106] of DInt		☐	☑	☑	☑	☐		
3	Particles/m^3 (5um)	Array[0..106] of DInt		☐	☑	☑	☑	☐		

Figure 4.6: TCON IP v4 data buffer parameter.

4.4 Modbus Processing Example

The following flowchart provides a basic example of controlling the IsoAir Pro-Plus using Modbus protocol. The flowchart is meant to be a starting point and does not detail error handling and not provide other functionalities which may be desired.

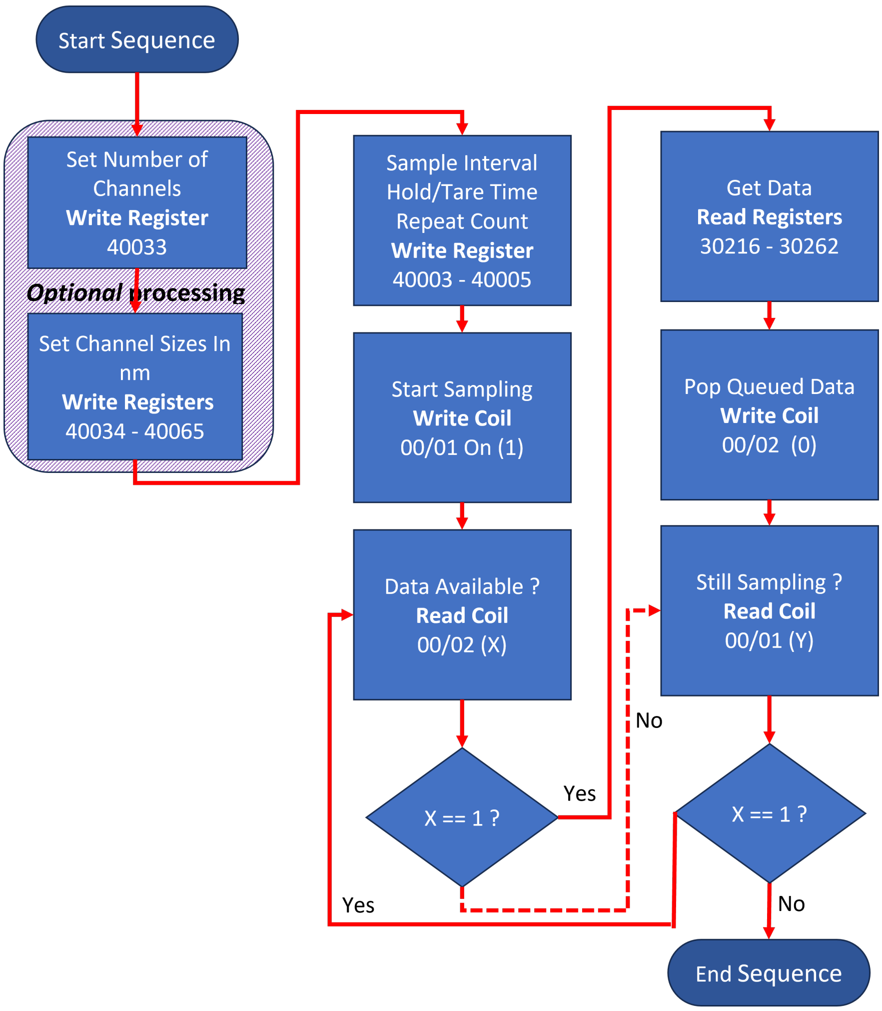


Figure 4.7: Modbus setup and running example.

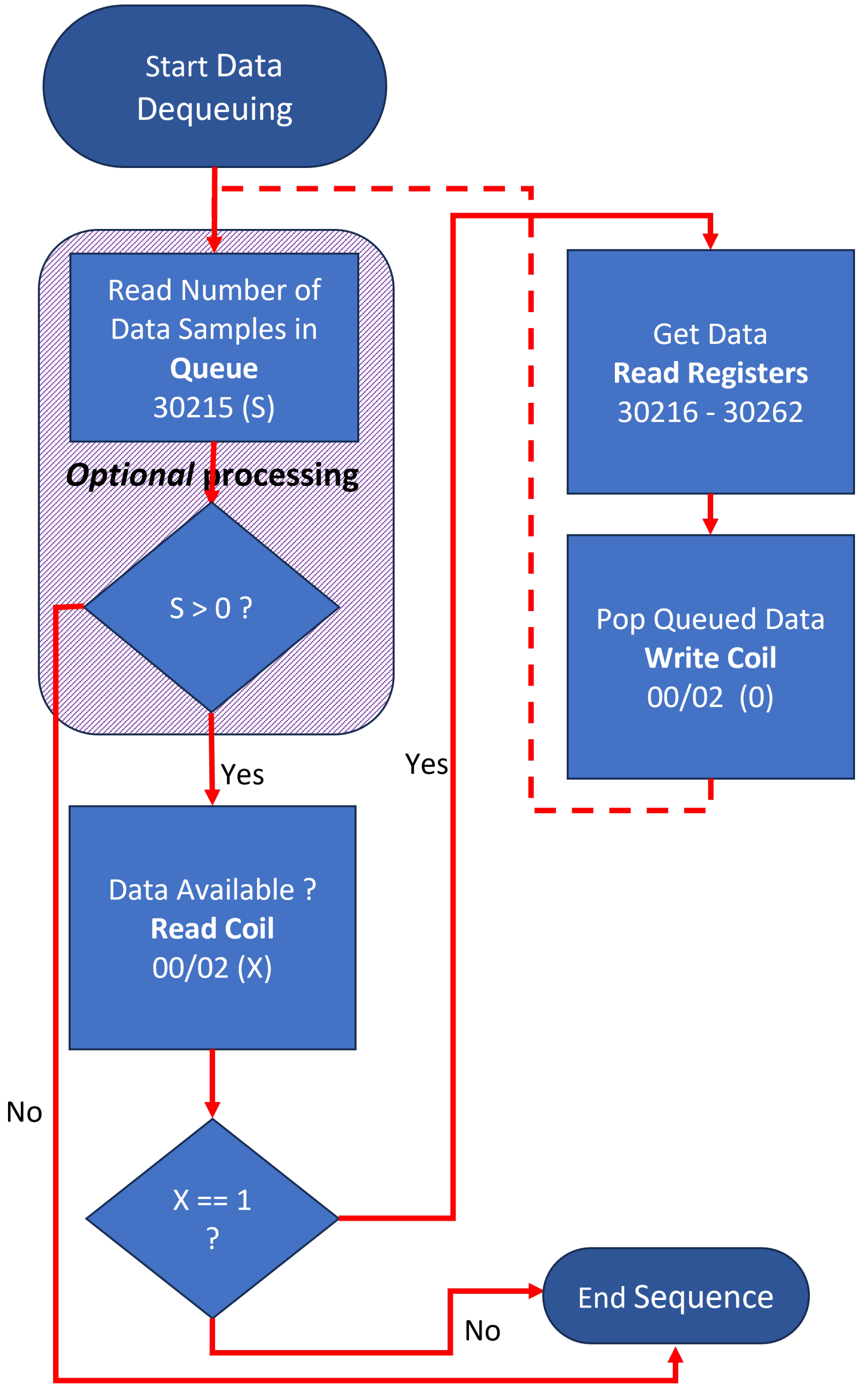


Figure 4.8: Modbus dequeuing example.

4.5 Finite State Machine Implementation for Modbus request

Finite State Machine, It is an abstract machine that can be in exactly one of a finite number of states at any given time.

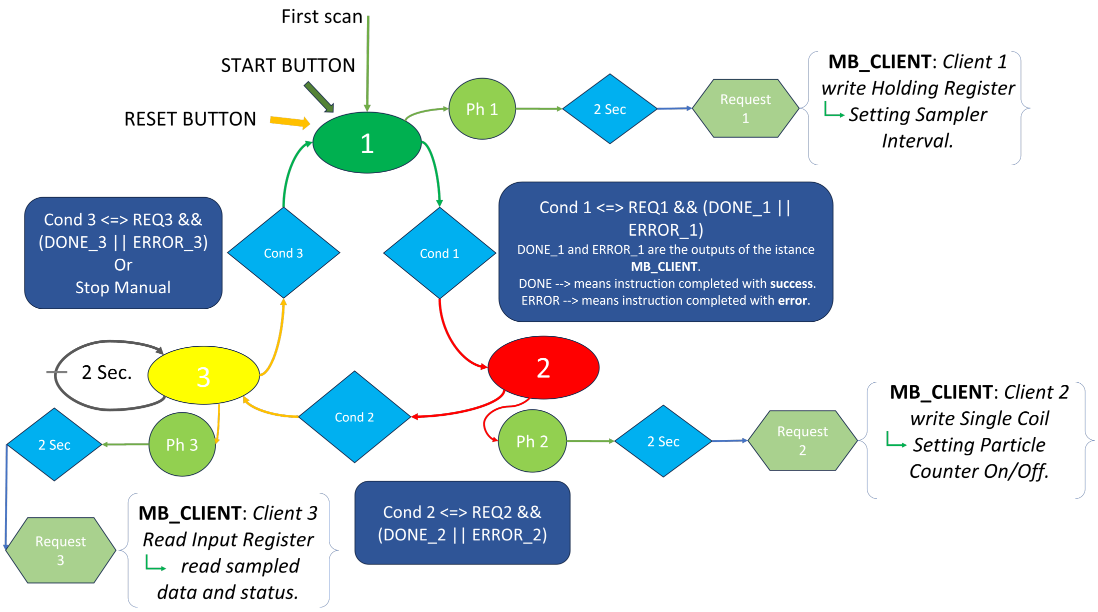
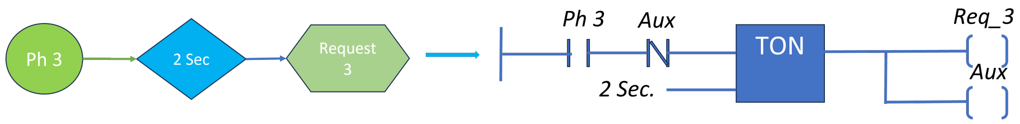


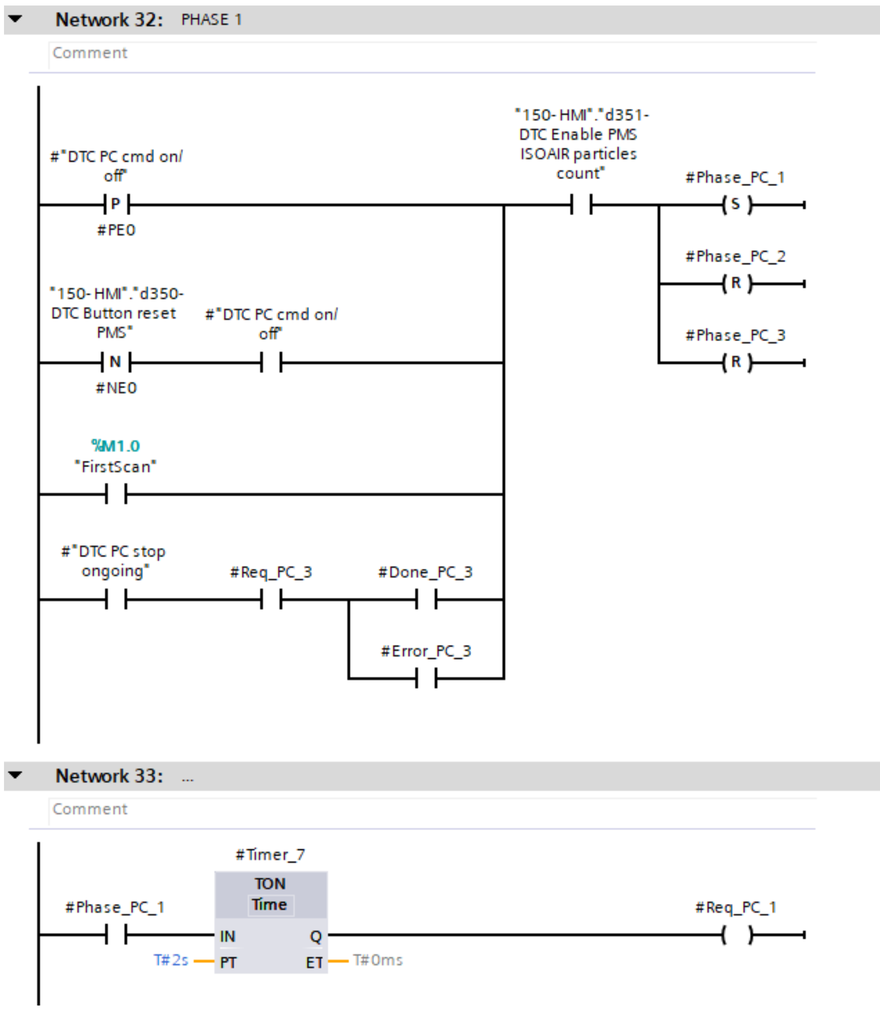
Figure 4.9: Finite State machine for software implementation.

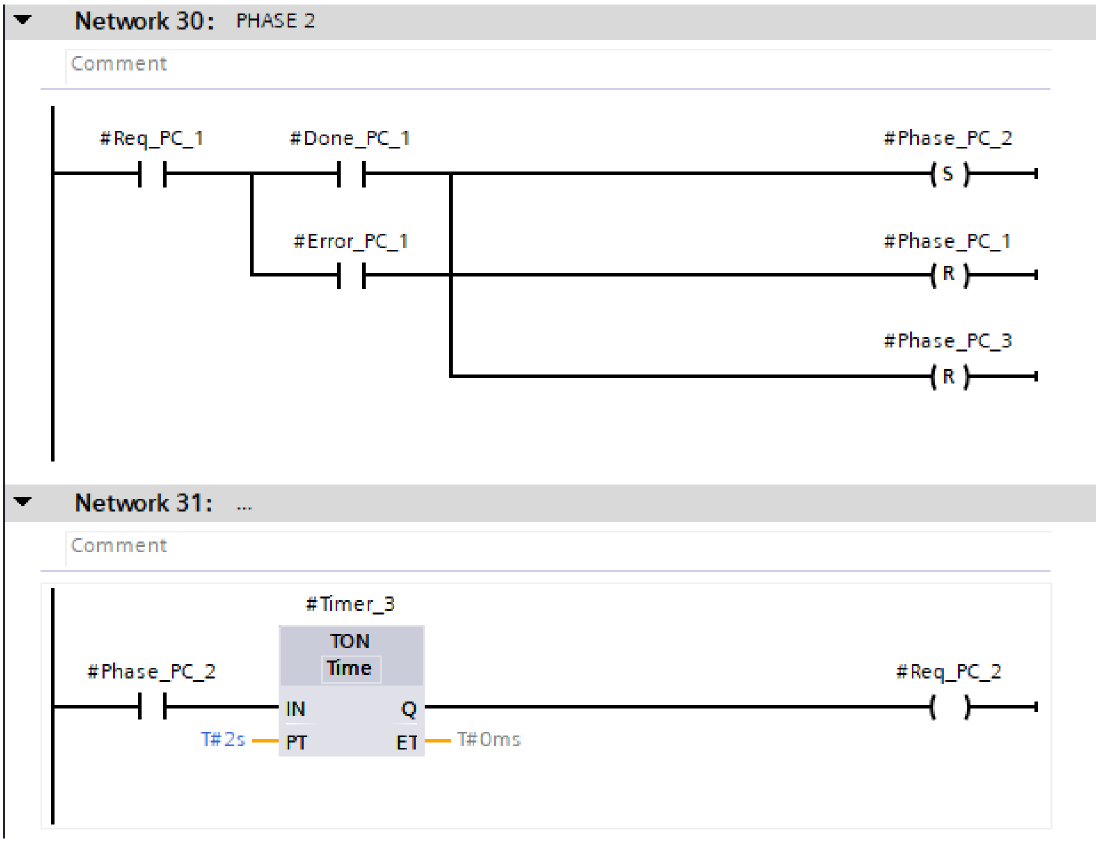
The FSM can change from one state to another in response to some inputs; the change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the inputs that trigger each transition. The FSM's memory is limited by the number of states it has. An example of finite state machine for our case study is illustrated in figure 4.9. As we can see from the FSM, the first operation we are going to carry out with the particle count via Modbus (MB CLIENT), is to set the number of seconds to run a sample.
The Sample interval is a variable accessible by the operator via HMI. After deciding the sampling duration or the target volume to reach, we enable sampling by writing the 1 on coil 00/01. with sampling enable, we begin to interrogate the particle count periodically by reading some of its internal registers and device status.To ensure that the waiting time (2 seconds) is effective even at the first access, the timing could be implemented as indicated in figure 4.5; combining the TON with an auxiliary coil.

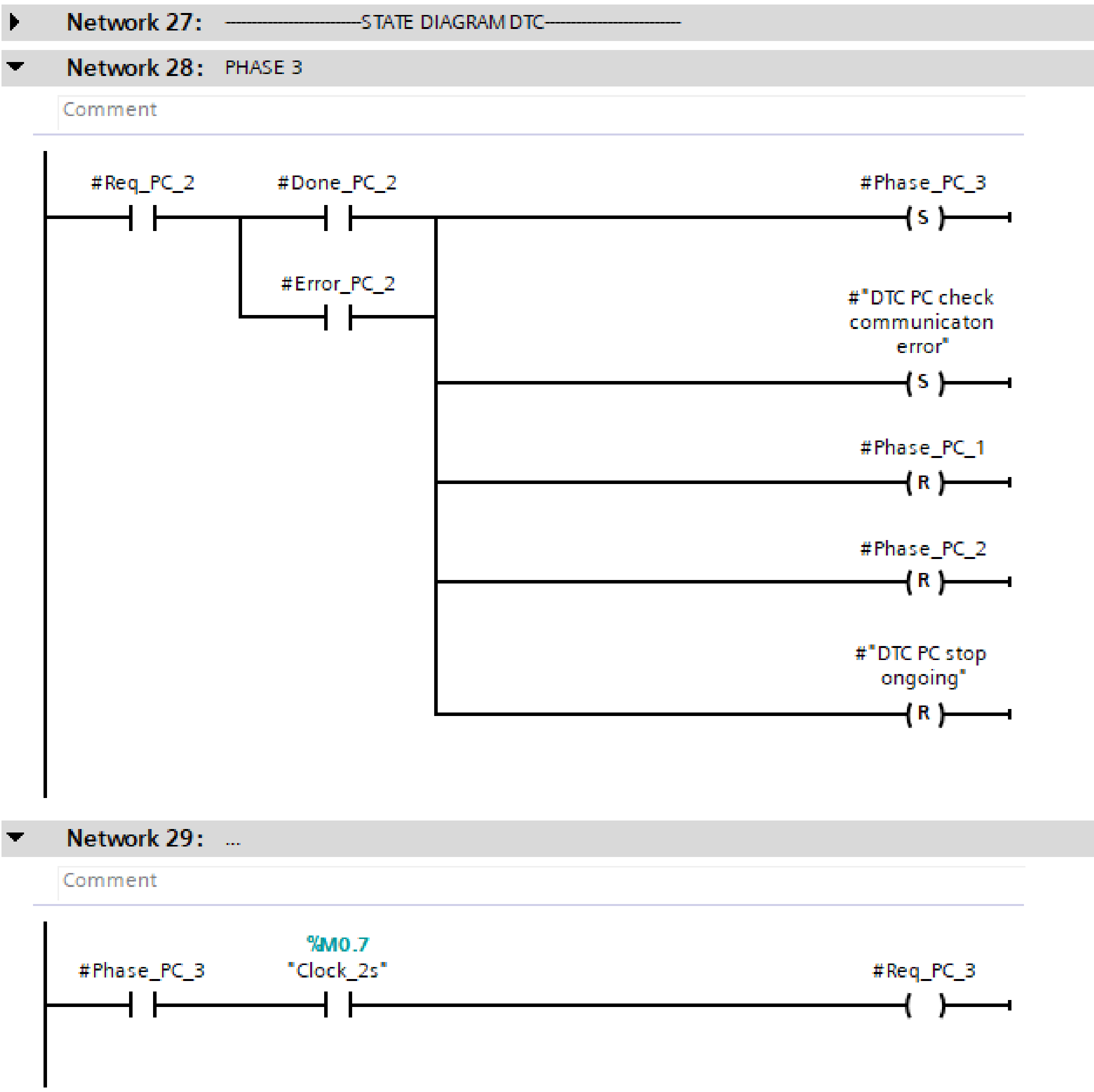


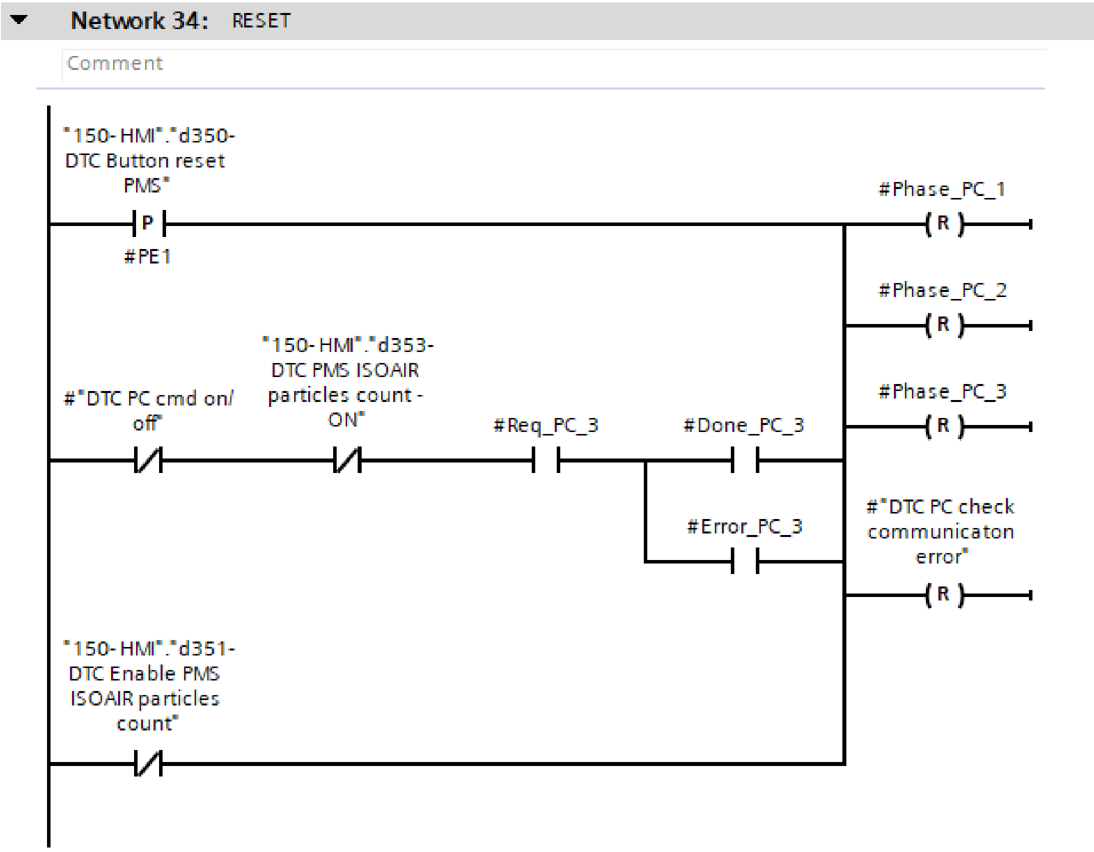
4.6 FSM Implementation

This program written in ladder language in Siemens' Tia Portal implements the illustrated logic of the state machine in the figure 4.9.
The first three figures show how the activation request of the individual phases takes place, and how the write requests are sent to the server. we can see a sort of mutual exclusion between states, which symbolizes the concept of a state machine. The activation of a state requires the deactivation of the previous state; one state is processed at a time.



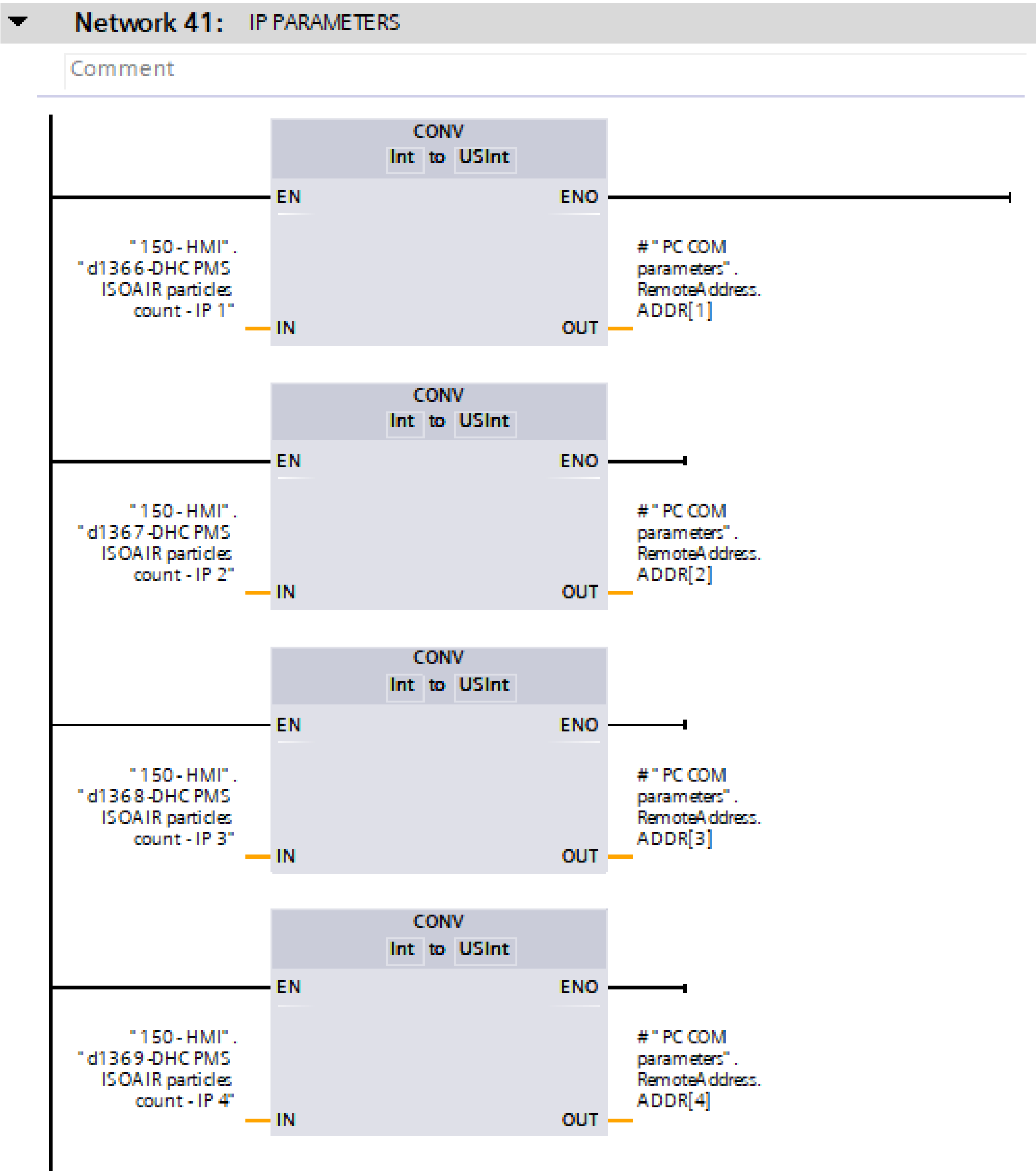

Network 30: PHASE 2
Comment
#Req_PC_1
#Done_PC_1
#Phase_PC_2
S
#Error_PC_1
#Phase_PC_1
R
#Phase_PC_3
R
Network 31: ...
Comment
#Timer_3
TON
Time
#Phase_PC_2
#Req_PC_2
IN
Q
T#2s
PT
ET
T#0ms


Network 27: ——————STATE DIAGRAM DTC——————
Network 28: PHASE 3
Comment
#Req_PC_2
#Done_PC_2
#Error_PC_2
#Phase_PC_3
S
#"DTC PC check communicaton error"
S
#Phase_PC_1
R
#Phase_PC_2
R
#"DTC PC stop ongoing"
R
Network 29: ...
Comment
#Phase_PC_3
%M0.7
"Clock_2s"
#Req_PC_3

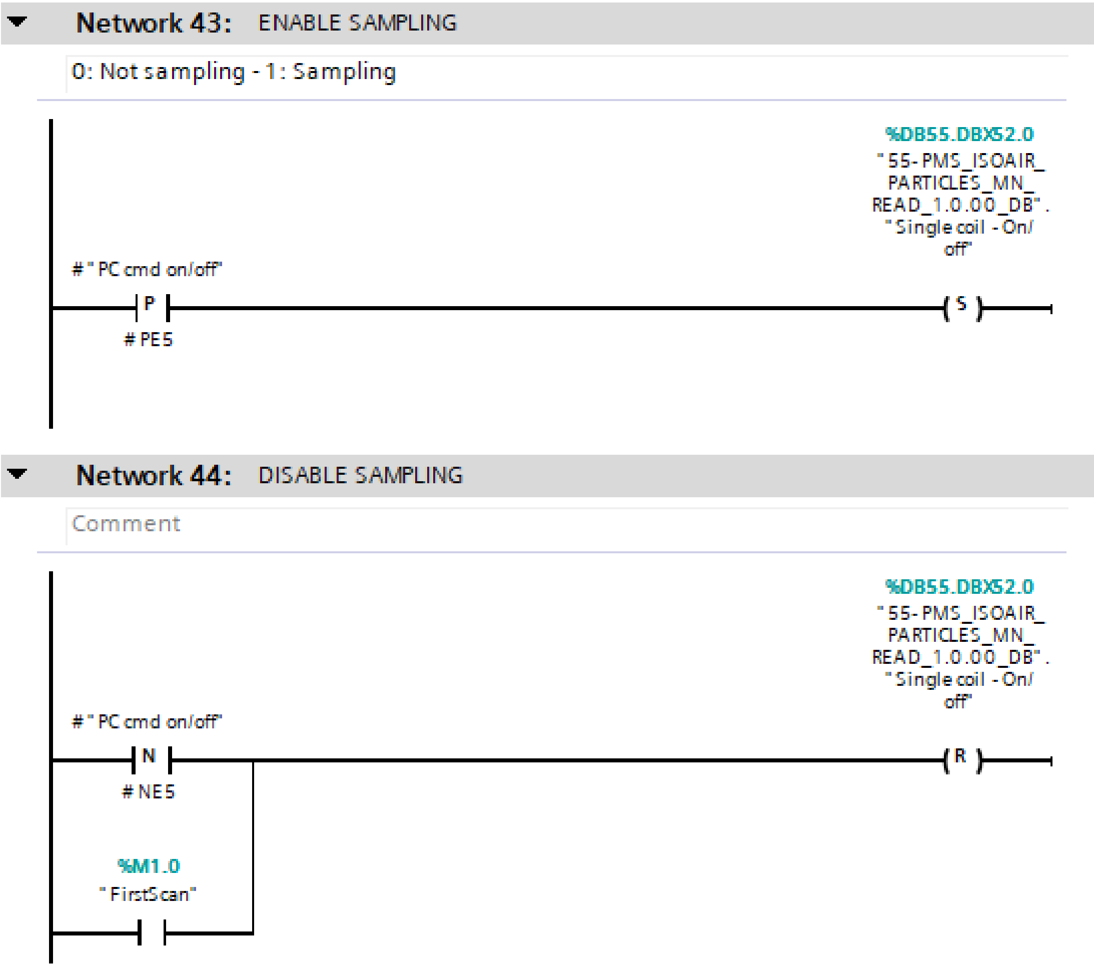


During the monitoring process, the operator can decide to interrupt the sampling, using a specific button made available via HMI. In this case, at the next processing cycle, it will be necessary to reset the state machine, to bring it back to its initial state.

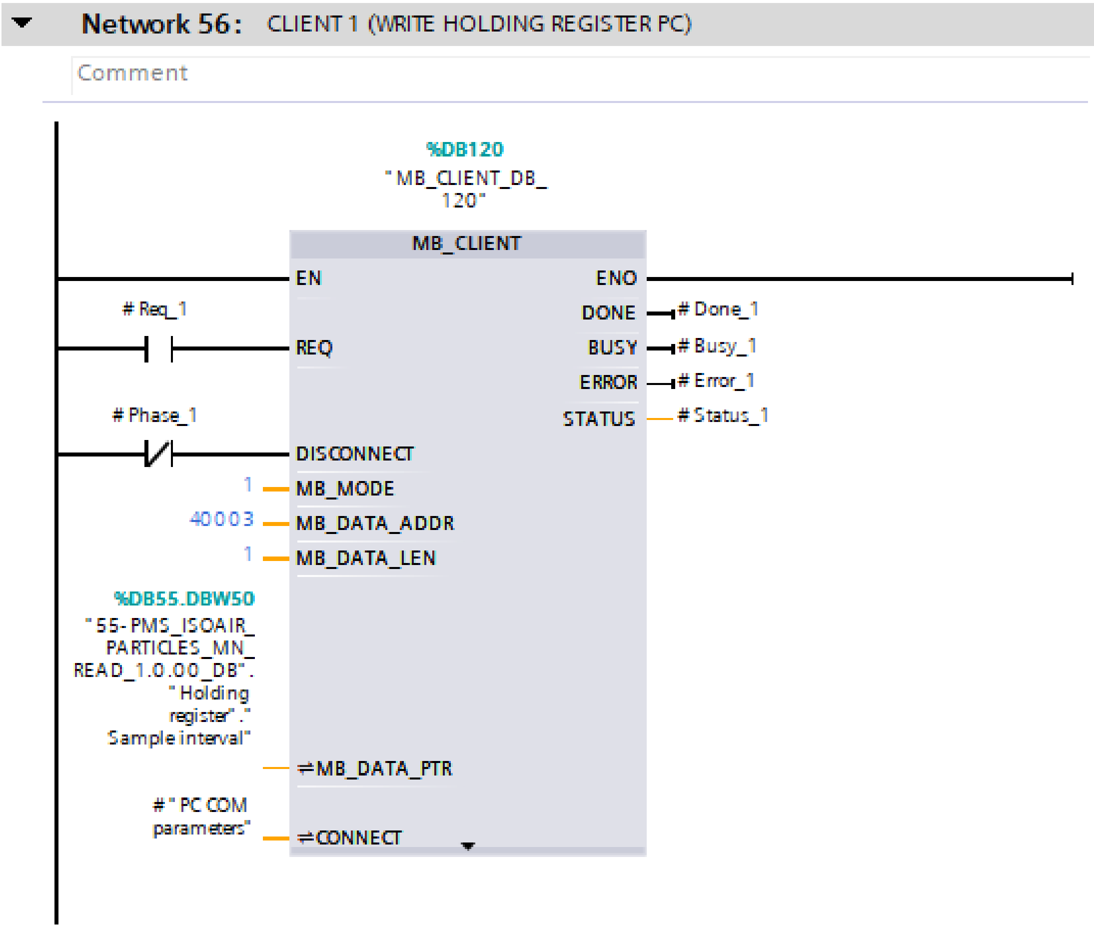
The communication parameters such as IP addresses and Ethernet ports have been divided into four data fields on the HMI to facilitate entry by the operator. This parameter is attached to the Connect input of the FB Client function block.

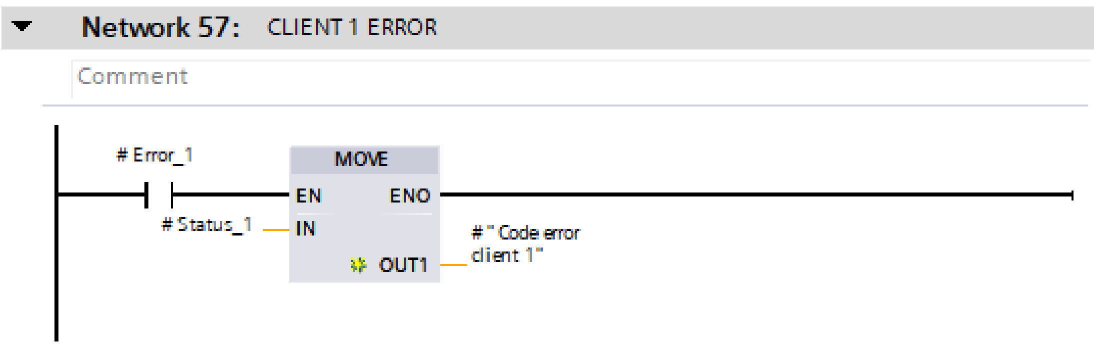


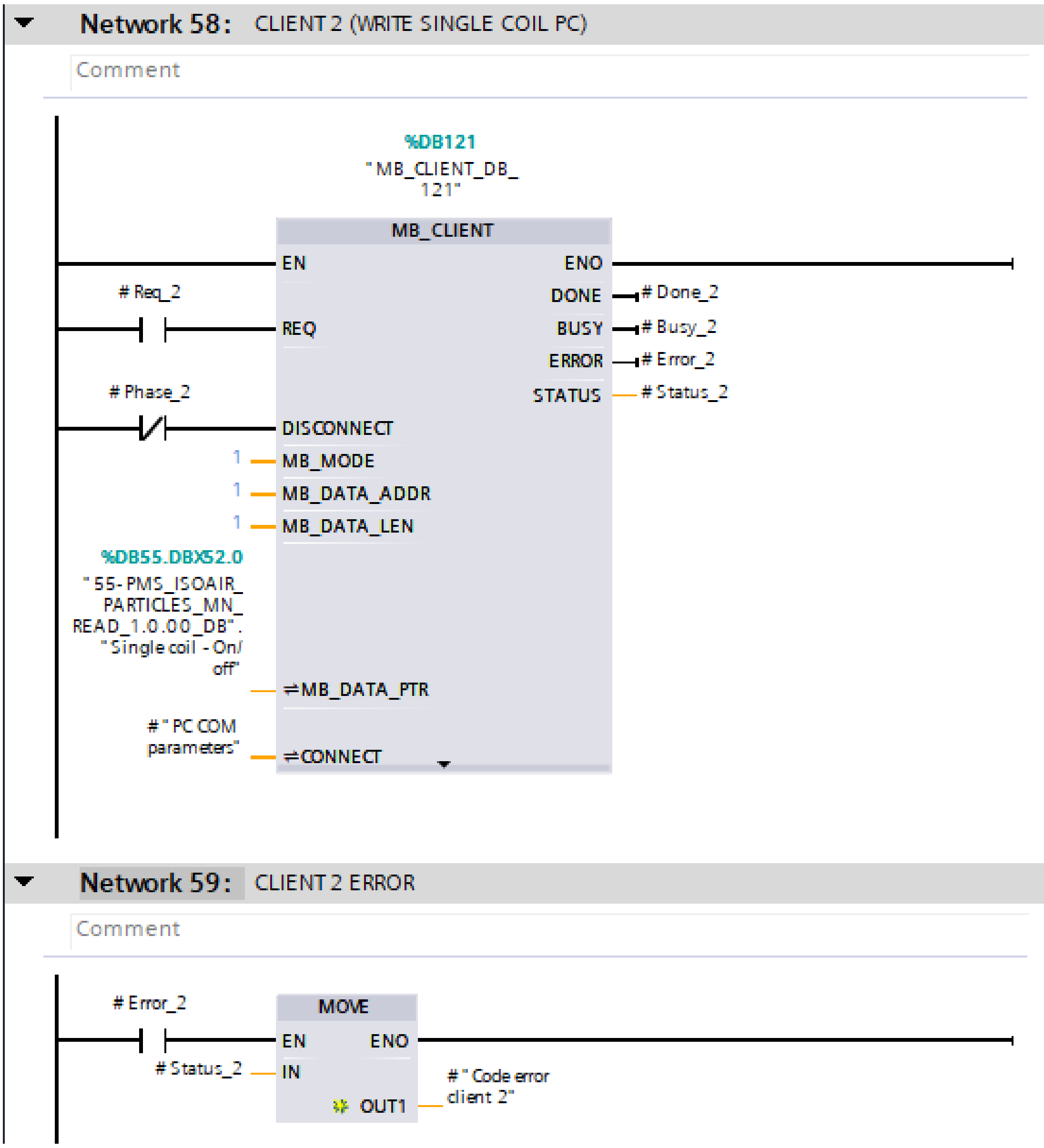
Sampling is enabled by writing a bit (00/01) inside the device register. The operator presses the Enable On/Off button on the HMI, and at the rising edge of the command, the single coil is set to 1 (TRUE); thereby sending the sample enable request to the server.

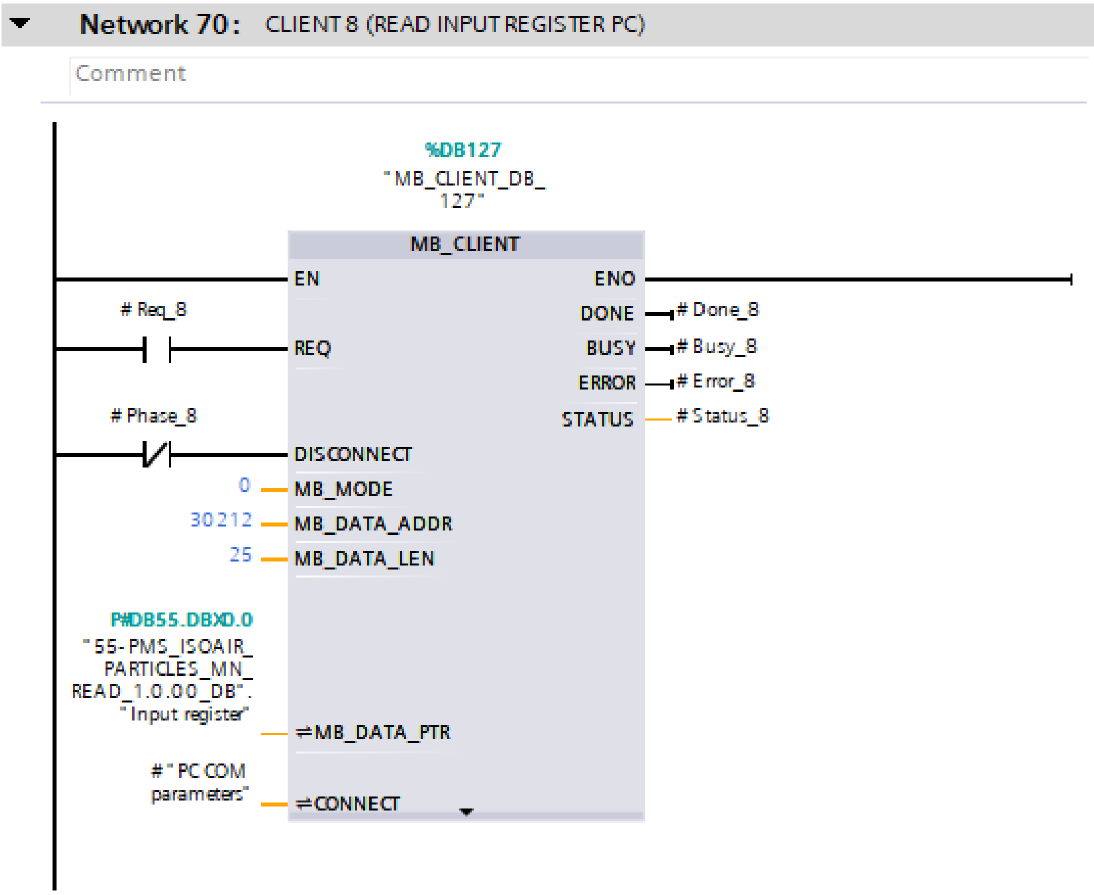


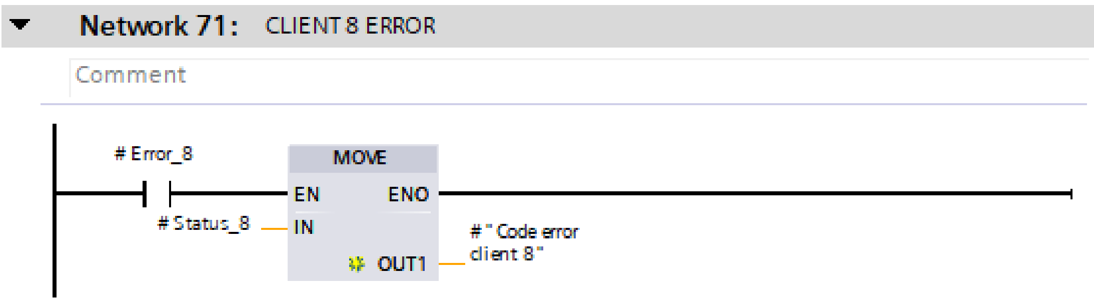
In this part of the program, information is exchanged between clients and servers. As highlighted by the state machine, we can see how, after the client's requests to write or read process data to the server, the state of the request is evaluated; checking if it was successful without any error code. see tables 4.5, 4.6 e 4.7, to understand the use of each parameter of the MB CLIENT.

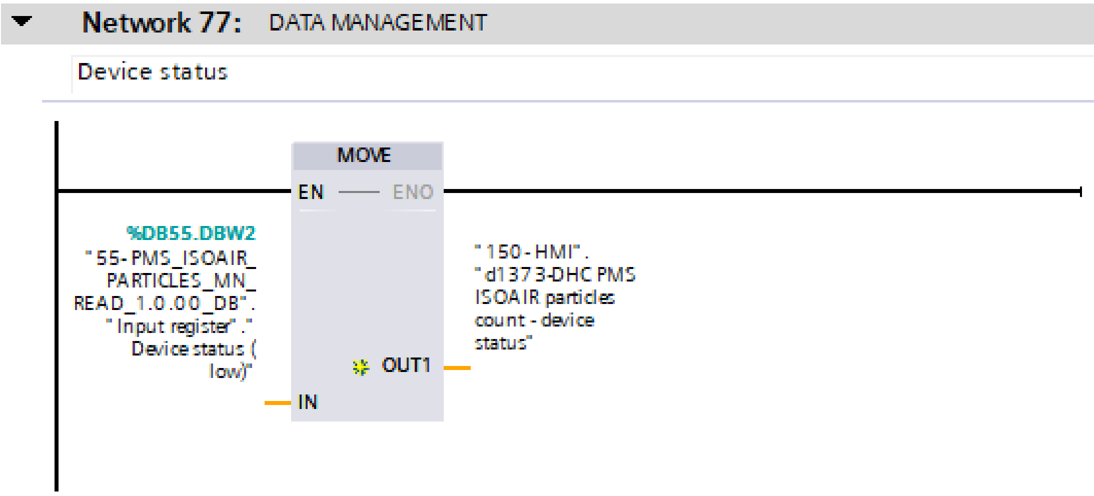


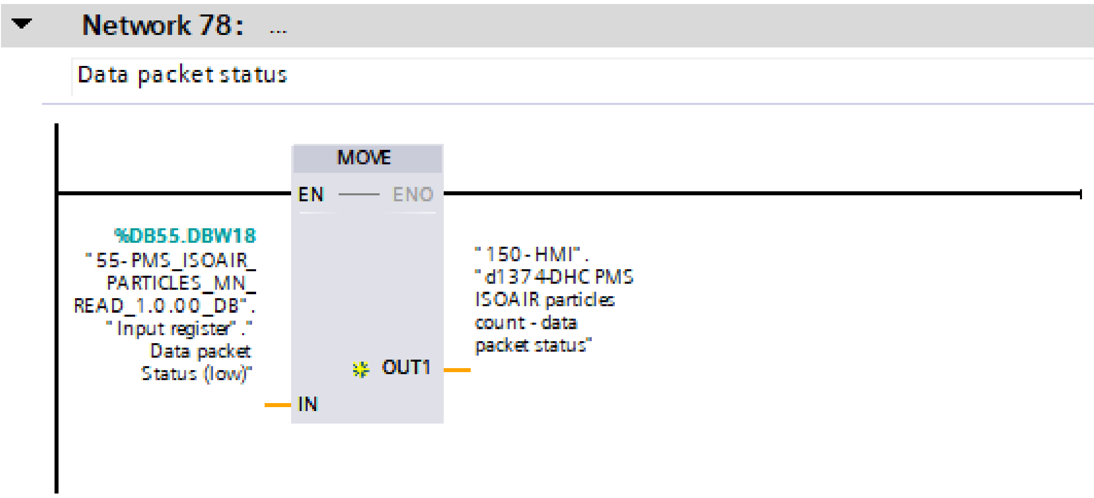


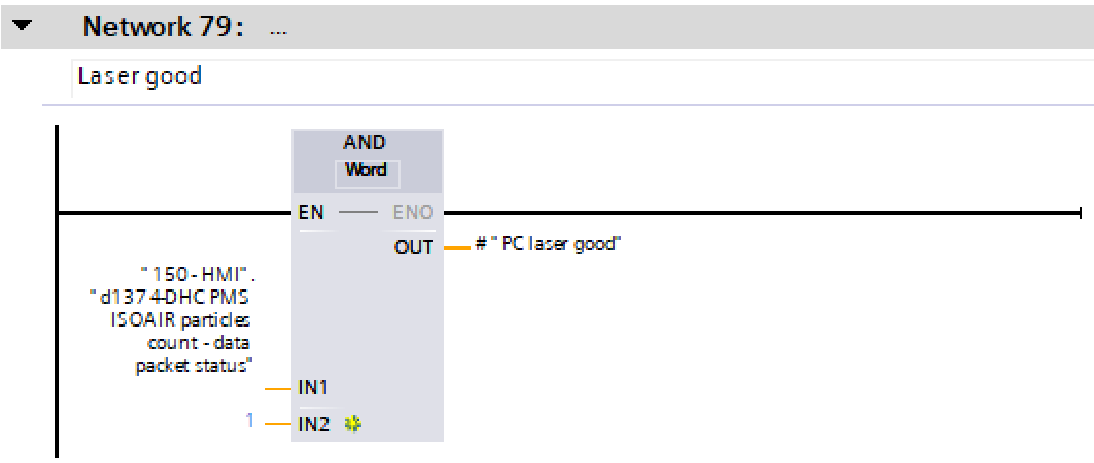

Network 58: CLIENT 2 (WRITE SINGLE COIL PC)
Comment
%DB121
"MB_CLIENT_DB_121"
MB_CLIENT
EN
ENO
Req_2
REQ
DONE
Done_2
BUSY
Busy_2
ERROR
Error_2
STATUS
Status_2
Phase_2
DISCONNECT
1
MB_MODE
1
MB_DATA_ADDR
1
MB_DATA_LEN
%DB55.DBX52.0
"55-PMS_ISOAIR_PARTICLES_MN_READ_1.0.00_DB".
"Single coil - On/off"
MB_DATA_PTR
"PC COM parameters"
CONNECT
Network 59: CLIENT 2 ERROR
Comment
Error_2
MOVE
EN
ENO
Status_2
IN
OUT1
"Code error client 2"


Network 70: CLIENT 8 (READ INPUT REGISTER PC)
Comment
%DB127
"MB_CLIENT_DB_127"
MB_CLIENT
EN
ENO
#Req_8
REQ
DONE
#Done_8
BUSY
#Busy_8
ERROR
#Error_8
STATUS
#Status_8
#Phase_8
DISCONNECT
0
MB_MODE
30212
MB_DATA_ADDR
25
MB_DATA_LEN
P#DB55.DBX0.0
"55-PMS_ISOAIR_PARTICLES_MN_READ_1.0.00_DB".
"Input register"
MB_DATA_PTR
#"PC COM parameters"
CONNECT


Network 71: CLIENT 8 ERROR
Comment
#Error_8
MOVE
EN
ENO
#Status_8
IN
OUT1
#"Code error client 8"


Network 77: DATA MANAGEMENT
Device status
MOVE
EN
ENO
%DB55.DBW2
"55-PMS_ISOAIR_PARTICLES_MN_READ_1.0.00_DB"."Input register"."Device status (low)"
IN
OUT1
"150-HMI"."d1373-DHC PMS ISOAIR particles count - device status"


Network 78: ...
Data packet status
MOVE
EN
ENO
%DB55.DBW18
"55-PMS_ISOAIR_PARTICLES_MN_READ_1.0.00_DB"."Input register"."Data packet Status (low)"
IN
OUT1
"150-HMI"."d1374-DHC PMS ISOAIR particles count - data packet status"


Network 79: ...
Laser good
AND
Word
EN
ENO
OUT
#"PC laser good"
"150-HMI"."d1374-DHC PMS ISOAIR particles count - data packet status"
IN1
1
IN2

4.7 Parameters Monitoring of particle counter

The HMI has been designed to facilitate direct use of the device by the operator. It is composed of a configuration page in figure 4.10, containing the communication parameters (IP address and network port), the sampling mode (manual or automatic), the target volume to be reached and the threshold values for the pre-alarm and alarm.
The status of the device, the reading of the samples for each type of particle (0.5 and 5 μ m) is represented in the figure

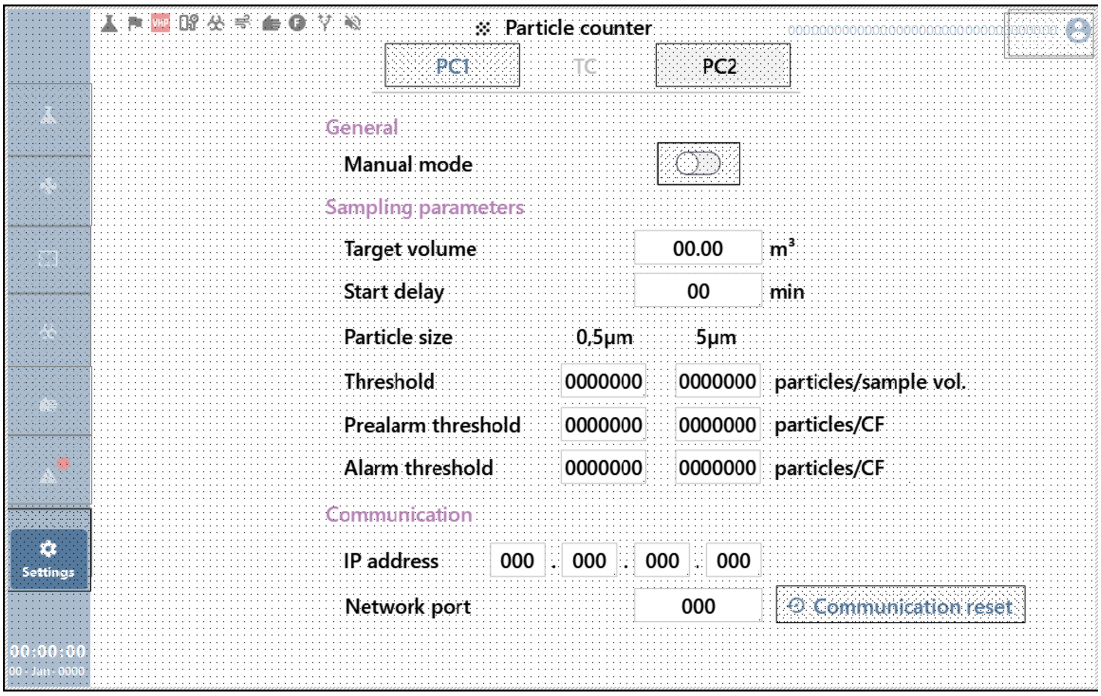


Figure 4.10: Particle counter setting page

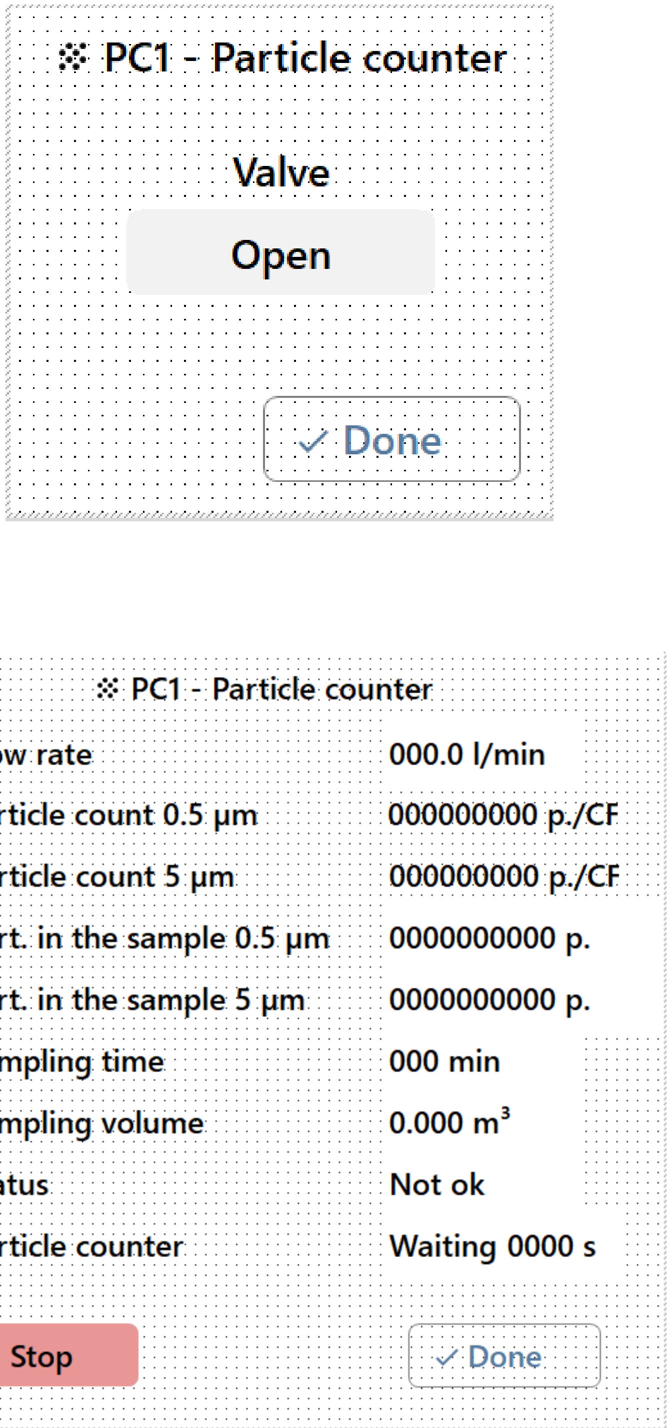


Figure 4.11: Particle counter feedback information during processing.

4.8 Finite State Machine for Modbus request versus Minicapt Remote Microbial Sampler

The microbiological sampling process begins with a configuration of the process parameters. Thanks to the human-machine interface, the operator is able to choose the sampling mode (manual or automatic), to set the target sample volume to be reached, and to decide how often to repeat the operation; See figure 4.8.
The first phase of the state machine consists of writing some specific internal registers of the device, with the process parameters configured by the operator via HMI. After which we proceed with enabling sampling and reading the samples. These sample readings are made available to the operator's view via HMI and reports; see figure 4.14.

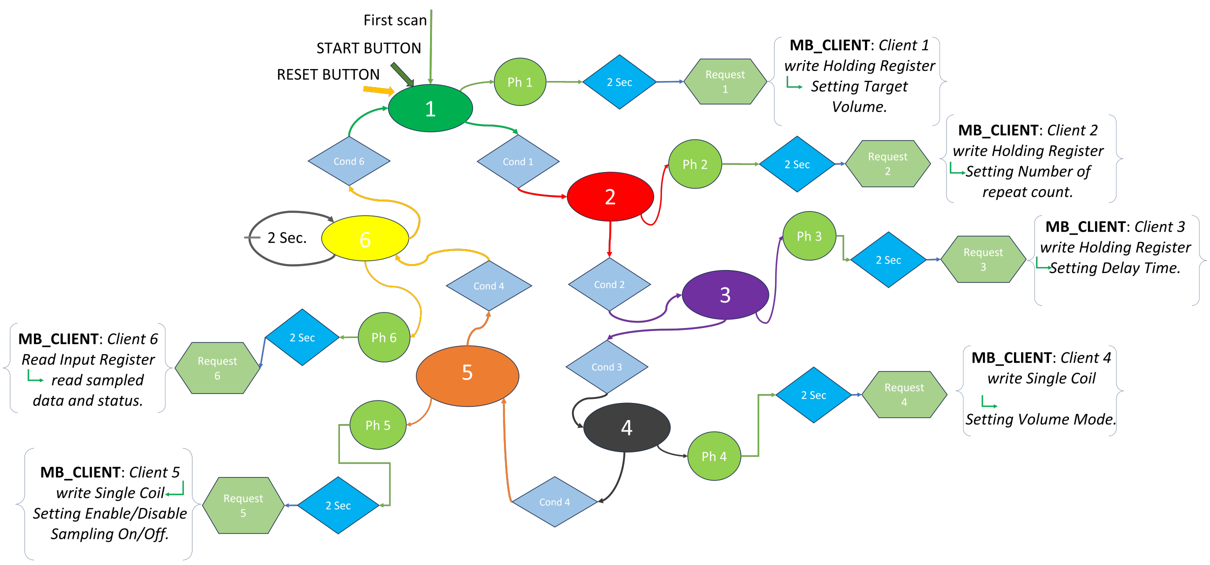


Figure 4.12: Example of finite state machine for Microbial sampler process.

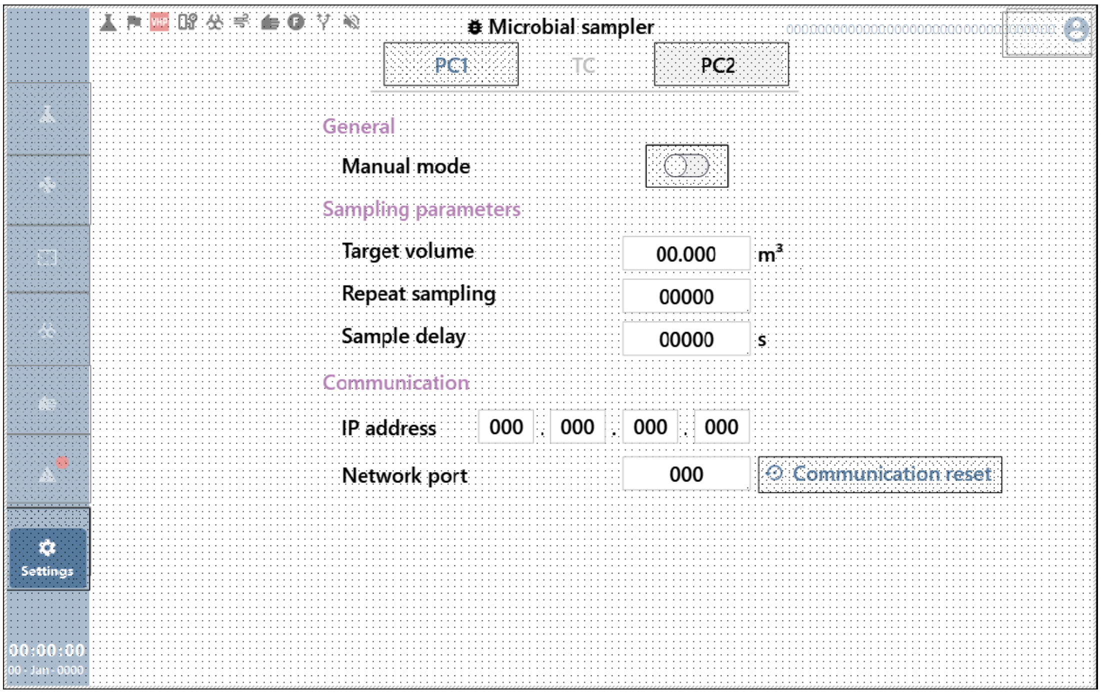
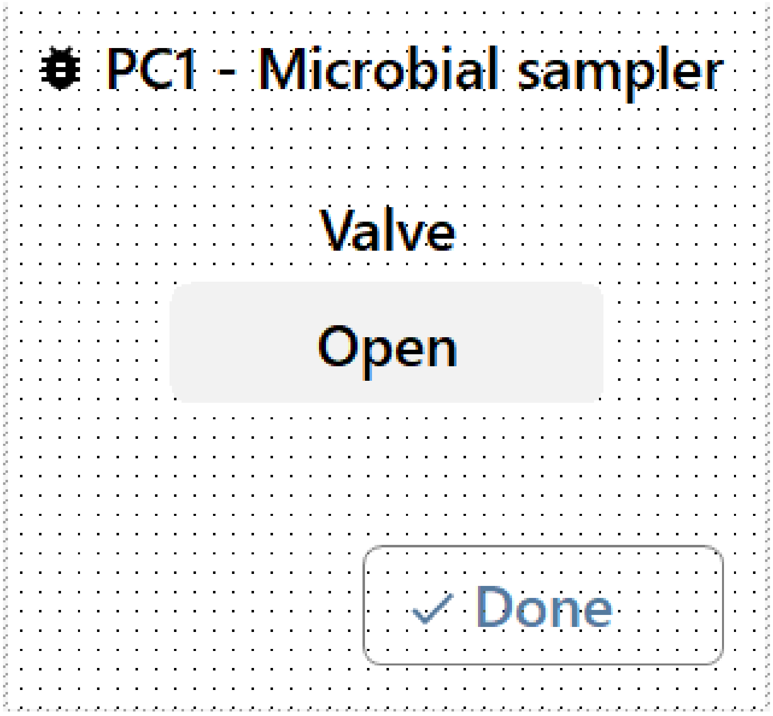


Figure 4.13: Microbial Sampler setting page



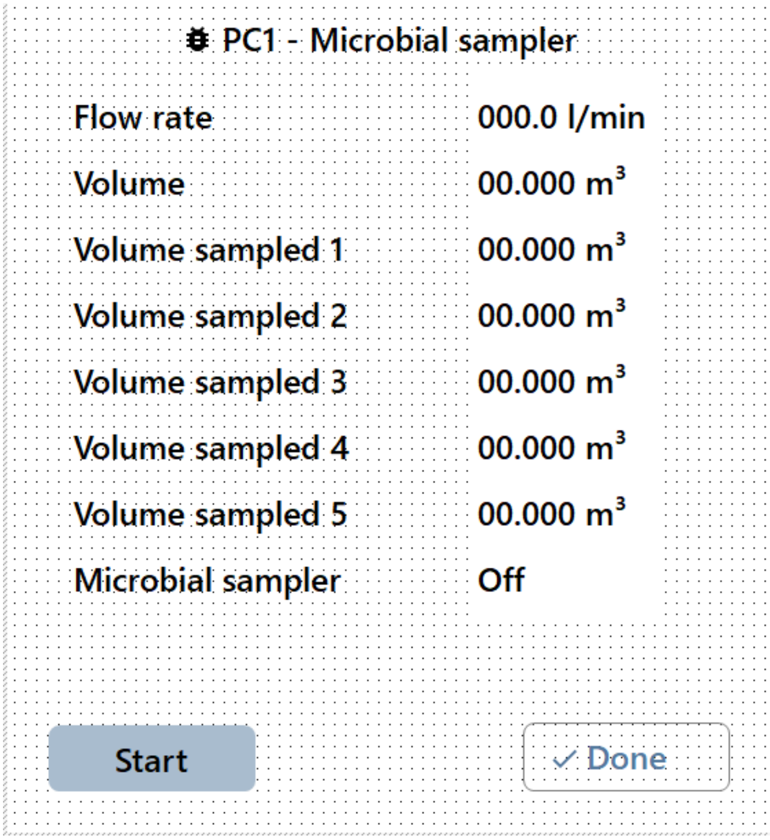


Figure 4.14: Microbial sampler feedback information during processing.

4.9 Particle counter/Microbial Enable conditions

The enable page shown in figure 4.15 illustrates a set of conditions, translated into machine states that must be respected; so that the operator can enable particle and microbiological counting. However, if one of the conditions were not met, the monitoring process would not take place.

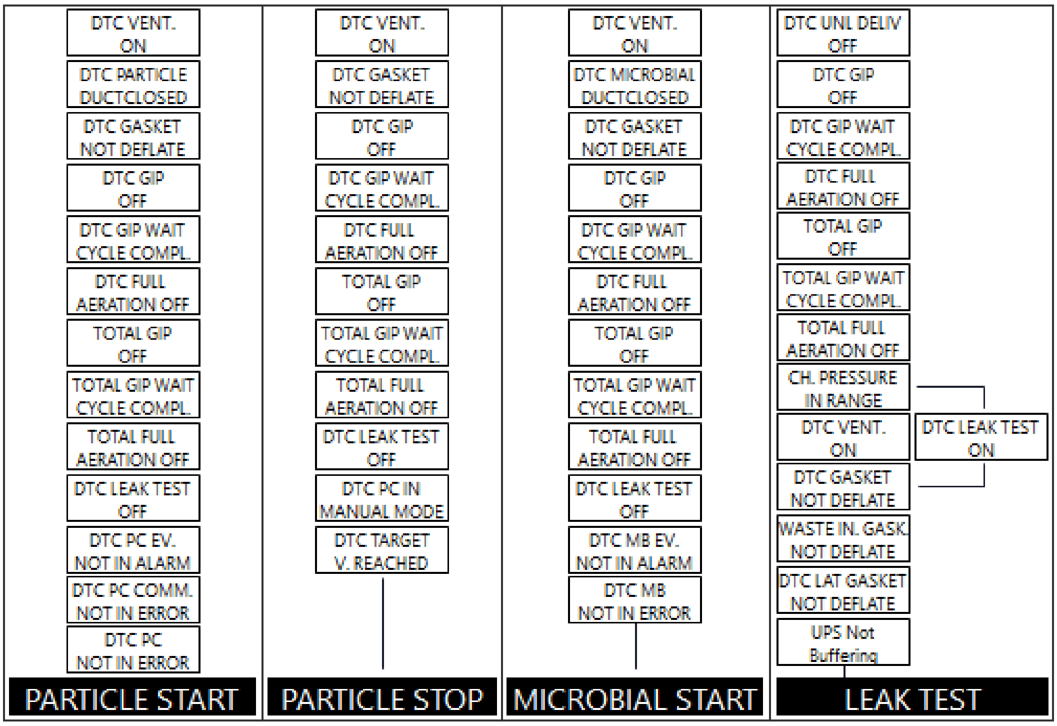


Figure 4.15: Interlocks machine for particle measuring systems enabling.

Conclusion

The environmental and process monitoring program is part of the overall CCS (Contamination Control Strategy) and is used to monitor controls aimed at minimizing the risk of microbial and particulate contamination. In this paper, the activity carried out during the internship period spent at the Tema Sinergie company has been illustrated in a brief and general context. The central idea of the activity aimed at interfacing via Modbus to a particle measurement system consisting of an IsoAirPro-plus model particle counter and a Mincat 100R microbiological one, to monitor the production cell and create a sterilized environment for the handling of ingredients considered highly active. To obtain satisfactory results in the time available, some essential points have been identified. Points like; the study of the company's production process, to get an idea of the reason for the PMS system inside the machine and understand under what conditions to activate it. The analysis of the available instruments and the method of interfacing with them. The application of knowledge of automation software and design patterns within the Siemens Tia Portal environment for the creation of the software. It should be noted that the reliability of each of the elements of the monitoring system (viable, non-viable and aseptically manufactured product (APS) only) when taken in isolation is limited and should not be considered individually as an indicator of asepsis. When considered together, the results help confirm the reliability of the design, validation and operation of the system you are monitoring.

Bibliography

IsoAir® Pro-Plus Operations Manual
[1]. Cioffi P, Marotta V, Fanizza C, Giglioni A, Natoli C, Petrelli F, Grappasonni I. Effectiveness and response predictive factors of erlotinib in a non-small cell lung cancer unselected European population previously treated: a retrospective, observational, multicentric study. J Oncol Pharm Pract. 2013 Sep;19(3):246–53. doi: 10.1177/1078155212465994. Epub 2012 Dec 7. PubMed PMID: 23223401. [PubMed] [Google Scholar]
[2]. Nguyen CTT, Petrelli F, Scuri S, Nguyen BT, Grappasonni I. A systematic review of pharmacoeconomic evaluations of erlotinib in the first-line treatment of advanced non-small cell lung cancer. Eur J Health Econ. 2019 Mar 6 doi: 10.1007/s10198-019-01040-7. [PubMed] [Google Scholar]
[3]. Cioffi P, Laudadio L, Nuzzo A, Belfiglio M, Petrelli F, Grappasonni I. Gemcitabine-induced posterior reversible encephalopathy syndrome: a case report. J Oncol Pharm Pract. 2012 Jun;18(2):299–302. doi: 10.1177/1078155211424628. [PubMed] [Google Scholar]
[4]. PIC/S Guide to Good Manufacturing Practice for Medicinal Products Part I, Chapter 3 - Premises and Equipment
[5]. Jimenez, Luis, ed. "Microbial Contamination Control in the Pharmaceutical Industry" 2004 Marcel Dekker, Inc.
[6]. Kushwaha, Poonam, "Microbial Contamination: A Regulatory Perspective" Journal of Pharmacy Research Vol.3. Issue 1. January 2010
[7]. Docherty, Scott E. "Establishing Microbial Cleaning Limits for Non-Sterile Manufacturing Equipment" Pharmaceutical Engineering 1999. Vol.19, Iss. 3
[8]. Sargent EV, Flueckiger A, Barle EL, Luo W, Molnar LR, Sandhu R, Weideman PA. The regulatory framework for preventing cross-contamination of pharmaceutical products: History and considerations for the future. Regul Toxicol Pharmacol. 2016 Aug;79(Suppl 1):S3–S10. doi: 10.1016/j.yrtph.2016.05.029. [PubMed] [Google Scholar]

Sitography

https://www.emsparticlesolutions.co.uk/wp-content/uploads/2021\/05/MiniCapt-Remote-Microbial-Air-Sampler-Operations-Manual-Rev-E.pdf
https://pmeasuring.it/manuals/isoair-pro-plus-aerosol-particle-sensor/
https://www.manualslib.com/manual/1453610/Siemens-S7-1200.html
https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf
https://plcynergy.com/modbus-tcp-protocol/
https://www.rotork.com/uploads/documents-versions/20213/1/pub059-003-00_0115.pdf
https://wingpath.co.uk/docs/modbus_tcp_specification.pdf
https://www.temasinergie.it/product/isolatore-per-colture-cellulari/
https://support.industry.siemens.com/cs/ww/de/view/102420337
https://support.industry.siemens.com/cs/document/65601780/tia-portal-an-overview-of-the-most-important-documents-and-links-controller?dti=0&dl=en&lc=it-IT
https://www.solisplc.com/tutorials/modbus-tcp-communications-in-siemens-tia-portal
https://www.youtube.com/watch?v=KgODo1syduU
https://en.wikipedia.org/wiki/Finite-statemachine
https://tsi.com/products/cleanroom-particle-counters/
https://www.cleanroomtechnology.com/five-fundamentals-of-particle-counters-179565
https://angstromtechnology.com/how-to-achieve-aseptic-conditions-in-medical-cleanrooms/
https://picscheme.org/docview/6608