

# **Data Streaming from a PLC to a PostgreSQL Database**

Nikolaj Backlund

Degree Thesis For Bachelor of Engineering

Degree Programme in Electrical Engineering and Automation

Vaasa, 2024

## **BACHELOR'S THESIS**

Author: Nikolaj Backlund

Degree Programme: Electrical Engineering and Automation, Vasa

Specialization: Automation

Supervisor(s): Jan Berglund, Kimi Roos

Title: Data Streaming from a PLC to a PostgreSQL Database

---

Date 18.4.2024    Number of pages: 56

---

### **Abstract**

This Bachelor's thesis was made for the automation department at VEO Oy. VEO Oy is an engineering company that focuses on the energy sector offering automation solutions to its customers with renewability in mind.

The purpose of the thesis is to look into whether data streaming from a PLC to a PostgreSQL database is possible and to make a base setup to use for new projects. The research explores if real-time data streaming to PostgreSQL is a viable option for historical archiving with a timestamp and to send alarms to a Scada system if data acquisition is stopped or interrupted.

The theoretical part includes how data acquiring is done and includes extensive information on data acquiring software, how a PLC works, virtual machines, SCADA systems, and Windows Server 2022.

The empirical part shows a test environment using the knowledge from the theoretical part. It examines how to set up and make a working solution for data streaming.

The results of the thesis show a working solution for data streaming from a PLC to a PostgreSQL database with a base setup.

---

Language: english

Keywords: PostgreSQL, Data streaming, Scada

# OPINNÄYTETYÖ

Tekijä: Nikolaj Backlund

Koulutus ja paikkakunta: Sähkö- ja automaatiotekniikka, Vaasa

Suuntautumisvaihtoehto: Automaatio

Ohjaajat: Jan Berglund, Kimi Roos

Nimike: Tiedonsiirto PLC:stä PostgreSQL-tietokantaan

---

Päivämäärä 18.4.2024 Sivumäärä: 56

---

## Tiivistelmä

Tämä kandidaatintyö tehtiin VEO Oy:n automaatio-osastolle. VEO Oy on energia-alaan keskittynyt insinööritoimisto, joka tarjoaa asiakkailleen automaatiotratkaisuja uusiutuvuuden huomioiden.

Opinnäytetyön tarkoituksena on tutkia, onko tiedon suoratoisto PLC:stä PostgreSQL-tietokantaan mahdollista ja voidaanko perusasetusta käyttää uusissa projekteissa. Tutkimuksessa selvitetään, onko reaaliaikainen tiedonsiirto PostgreSQL-tietokantaan käyttökelpoinen vaihtoehto historiallisten tietojen arkistoinniseksi aikaleimalla ja hälytysten lähettämiseksi Scada-järjestelmään, jos tiedonkeruu pysähtyy tai keskeytyy.

Teoriaosuus käsittää, miten tiedonkeruu tapahtuu, ja se sisältää runsaasti tietoa tiedonkeruuhjelmistoista, PLC:n toiminnasta, virtuaalikoneista, SCADA-järjestelmistä ja Windows Server 2022:sta.

Empiirisessä osassa esitellään testiympäristö, jossa käytetään teoriaosuuden tietoja. Siinä tarkastellaan, miten tiedonsiirtoa varten perustetaan ja tehdään toimiva ratkaisu.

Opinnäytetyön tulokset osoittavat toimivan ratkaisun tiedonsiirtoon PLC:stä PostgreSQL-tietokantaan perusasetuksilla.

---

Kieli: englanti

Avainsanat: PostgreSQL, Tietojen suoratoisto, Scada

## EXAMENSARBETE

Författare: Nikolaj Backlund

Utbildning och ort: El- och automationsteknik, Vasa

Inriktning; Automation

Handledare: Jan Berglund, Kimi Roos

Titel: Dataströmning Från en PLC till en PostgreSQL-databas

---

Datum: 18.4.2024 Sidantal: 56

---

### Abstrakt

Detta examensarbete gjordes för automationsavdelningen på VEO Oy. VEO Oy är ett ingenjörsköretag som fokuserar på energisektorn och erbjuder automationslösningar till sina kunder med förnybarhet i åtanke.

Syftet med examensarbetet var att undersöka om dataströmning från en PLC till en PostgreSQL-databas är möjlig och att skapa en basinstallation att använda för nya projekt. Uppgiften var att undersöka om dataströmning i realtid till PostgreSQL är ett genomförbart alternativ för historisk arkivering med en tidsstämpel och för att skicka larm till ett Scada-system om datainsamlingen stoppas eller avbryts.

Den teoretiska delen handlar om hur datainsamling går till och innehåller omfattande information om programvara för datainsamling, hur en PLC fungerar, virtuella maskiner, Scada-system och Windows server 2022.

Den empiriska delen visar en testmiljö med hjälp av kunskaperna från den teoretiska delen. Den undersöker hur man sätter upp och skapar en fungerande lösning för dataströmning.

Resultaten av examensarbetet visar en fungerande lösning för dataströmning från en PLC till en PostgreSQL-databas med en basinstallation.

---

Språk: engelska

Nyckelord: PostgreSQL, dataströmning, Scada

## Table of contents

1	Introduction .....	1
1.1	Background .....	1
1.2	About VEO Oy .....	1
1.3	Delimitations .....	1
1.4	Purpose .....	2
2	Theory.....	2
2.1	Overall picture of an automation system .....	2
2.2	Data acquisition.....	3
2.2.1	Purpose of Data Acquisition.....	4
2.2.2	Importance of Data Acquisition Systems .....	4
2.2.3	Basic types of data acquisition systems .....	4
2.3	To successfully acquire data from a plc .....	5
2.4	Basic PLC functions .....	5
2.4.1	Input scan .....	6
2.4.2	Program execution.....	6
2.4.3	Output update.....	6
2.4.4	Continuous scan cycle .....	7
2.4.5	Data storage.....	7
2.4.6	Communication capabilities.....	7
2.4.7	Ladder.....	7
2.5	ibaPDA.....	8
2.6	PostgreSQL .....	9
2.6.1	Why use PostgreSQL .....	10
2.6.2	Reliability and stability .....	10
2.6.3	Data types and data integrity .....	10
2.6.4	Performance.....	11
2.6.5	PGAdmin.....	11
2.7	VMware workstation.....	12
2.8	Windows Server 2022 .....	13
2.9	Siemens TIA portal.....	13
2.9.1	Integrated automation and support for multiple devices .....	14
2.9.2	Graphical programming.....	14
2.9.3	Simulation and Testing .....	14
2.9.4	Diagnostics and monitoring .....	14
2.9.5	Scalability .....	15
2.9.6	Overall .....	15

2.10	SNMP.....	15
2.11	OPC UA.....	16
2.12	WinCC OA.....	17
2.13	Task Scheduler.....	18
3	Methods and approaches .....	18
3.1	Test setup.....	19
3.2	Installing Windows Server 2022 .....	19
3.2.1	Installing ibaPDA.....	20
3.2.2	PostgreSQL installation.....	23
3.2.3	Installing Siemens TIA portal and WinCC OA.....	23
3.2.4	Setting up PLC and switch.....	24
3.3	PLC test program .....	24
3.4	Configuring IBA settings .....	27
3.4.1	Add PLC and import address book.....	27
3.4.2	Configuring in and Outputs .....	29
3.4.3	Connection from ibaPDA to database.....	30
3.4.4	Connection from ibaPDA to WinCC OA SNMP.....	31
3.4.5	ibaPDA connection to WinCC OA OPC UA.....	37
3.5	PostgreSQL configuration.....	44
3.6	Data deletion.....	46
3.7	Testing.....	47
4	Discussion and result.....	54
5	References.....	55
	Figure 1 Automation Overview .....	3
	Figure 2 DEWESoft .....	4
	Figure 3 Siemens output module.....	5
	Figure 4 Ploycase plc .....	6
	Figure 5 ibaPDA.....	9
	Figure 6 PostgreSQL logo.....	10
	Figure 7 VMware workstation pro logo.....	12
	Figure 8 Windows Server 2022.....	13
	Figure 9 Siemens TIA portal .....	15
	Figure 10- SNMP configuration.....	16
	Figure 11- OPC UA .....	17
	Figure 12 The role of WINCC OA in an automation system .....	18
	Figure 13 Connection diagram .....	19
	Figure 14 VMware install diagram.....	20
	Figure 15 Install progress .....	21
	Figure 16 IP configuration settings .....	22
	Figure 17 PostgreSQL wizard .....	23
	Figure 18 Communication settings for PLC.....	24

Figure 19 Tag table .....	25
Figure 20 PLC network layout.....	26
Figure 21 Main network.....	27
Figure 22 PLC connection.....	28
Figure 23 Address book .....	28
Figure 24 Adding digital connections .....	29
Figure 25- ibaPDA trend graph .....	29
Figure 26 SQL statement.....	30
Figure 27 ibaPDA PostgreSQL connection.....	31
Figure 28 SNMP enabled.....	32
Figure 29 SNMP IsMeasuring tag.....	33
Figure 30 WinCC OA drivers.....	34
Figure 31 WinCC OA to ibaPDA connection with SNMP .....	35
Figure 32 SNMP OID connection.....	36
Figure 33 SNMP rectangle wizard .....	36
Figure 34 SNMP rectangle wizard.....	37
Figure 35 OPC UA certificate .....	38
Figure 36 ibaPDA OPC UA settings .....	38
Figure 37 OPC UA Measuring tag.....	39
Figure 38 OPC UA client .....	39
Figure 39 OPC UA WinCC OA connection settings .....	40
Figure 40 PowerConfig for OPC UA .....	41
Figure 41 Periphery address.....	42
Figure 42 OPC UA rectangle wizard .....	43
Figure 43 WinCC OA screen.....	43
Figure 44 Database creation.....	44
Figure 45 Data insert.....	44
Figure 46 Data output in datastream1 .....	45
Figure 47 Data stream SQL statement.....	45
Figure 48 Data from datastream1 .....	46
Figure 49 Data deletion.....	46
Figure 50 Task scheduler setup .....	47
Figure 51 PLC program with online modification.....	48
Figure 52 ibaPDA trend graph.....	49
Figure 53 ibaPDA trend graph with updated PLC output.....	50
Figure 54 Query to show 100 last rows .....	50
Figure 55 datastream1 data.....	51
Figure 56 WinCC OA Quicktest module online test.....	51
Figure 57 Wincc OA with data acquisition stopped .....	52
Figure 58 datastream1 100 first rows.....	52
Figure 59 Modified bat file .....	53
Figure 60 Task scheduler.....	53
Figure 61 SQL statement.....	53

## **1 Introduction**

This thesis is made on behalf of VEO Oy. The task was to make a virtual server that could acquire signals from a PLC and store them in a PostgreSQL database using ibaPDA software. This is done so that in future projects the server can be easily deployed.

### **1.1 Background**

This thesis is done for the automation department of VEO to create a preconfigured virtual machine to easily deploy the machine into future projects.

It gets more and more important to save and monitor signals from PLCs and other controllers. This virtual machine can monitor signals from PLCs and save them to a PostgreSQL database with SNMP monitoring to ensure that all data is continuously monitored. This is important as Fingrid demands that power plants can save certain data for up to 14 days from when it was outputted from the PLC. To make a machine that can save data from PLCs you need to find an operating system, data-acquiring software, and database that can do this. In this case, Windows Server 2022, ibaPDA, and PostgreSQL were chosen.

### **1.2 About VEO Oy**

VEO Oy is a company in Vasa specializing in the energy sector and offers automation solutions. VEO has offices in Finland, Sweden, Norway, and Great Britain. More than 50% of its revenue comes from renewable energy. VEO has over 500 employees and had 121 million euros of turnover in 2021. (Veo, 2023)

### **1.3 Delimitations**

The thesis only tests the data streaming with simulated data from a PLC and not from an actual project. The test setup has virtual machines, PLC, and SCADA computers on the same network and it is not tested if signal acquisition is possible from an outside network. The network is behind a firewall but attacks from inside the network are not tested.

## **1.4 Purpose**

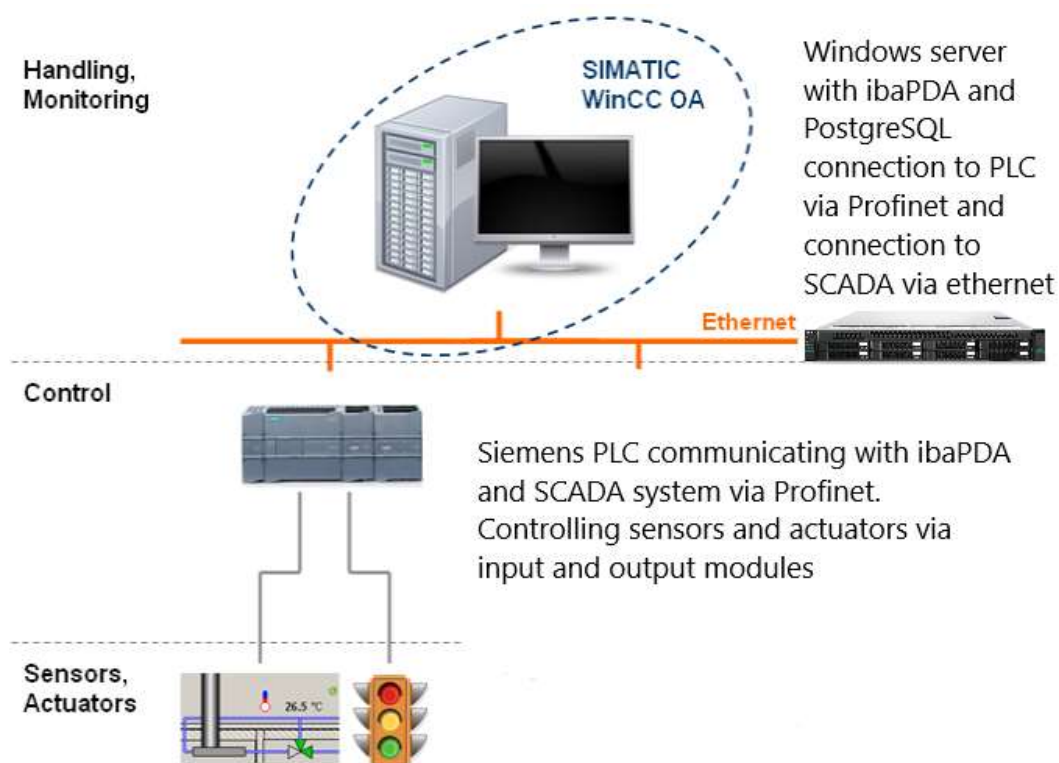
The purpose is to create a virtual machine that has all the software installed and preconfigured as a base. So when a new project comes the system engineer does not have to install everything and configure everything from scratch.

## **2 Theory**

This chapter has essential information and theory about data acquisition, basic PLC functions, PostgreSQL, VMware workstation, Windows Server 2022, WinCC OA, and Task scheduler. Having knowledge of these systems is important when making a base project containing all these components.

### **2.1 Overall picture of an automation system**

An automation system usually consists of a PLC controlling and monitoring sensors and actuators it does this via in and output modules sending and monitoring voltage and amperage. The PLC communicates with field devices such as other PLCs and frequency converters via profinet or other communication protocols. A Scada system which is the supervision and control system communicates via profinet or other communication protocols to PLCs and other field devices but Scada also communicates with servers such as data acquisition servers or historical archiving servers and manufacturing execution systems. In this case, a data acquisition server will be created, which will be communicated with by the SCADA system via SNMP or OPC UA, and the server will be communicated with by the PLC via Profinet.



**Figure 1 Automation Overview**

Here is an example of an automation system at the top is the SCADA system then there is the data acquisition server below is the PLC at the bottom is where the sensors and actuators are.

## 2.2 Data acquisition

Data acquisition theory involves the principles and techniques used to capture and store data from various sources in this case a Siemens plc. When acquiring data from a PLC there are some specific techniques used. A data acquisition system is a system containing measurement devices, sensors, computer, and data acquisition software. A data acquisition system is used for acquiring, visualizing, and storing data. To do this you need to understand the system you are working on. There are several types of acquisition systems from a basic single measurement device to big 1000 signal systems. A data acquisition system can monitor multiple different signals inputting and outputting voltage, current, frequency, angle, and digital in and outputs. (Smith, 2024)



**Figure 2 DEWEsoft**

### **2.2.1 Purpose of Data Acquisition**

The primary purpose of acquisition systems is to acquire and store data but also to provide real-time visualization. They also give historical data for analytics and reports. Customers usually have some requirements for the seller about data storing, real-time visualization, data analysis, and reporting. (Smith, 2024)

### **2.2.2 Importance of Data Acquisition Systems**

Data acquisition systems are important when you want to minimize downtime in a system and historical data is important. Before data acquisition was available it was harder to find faults which increased downtime in a system (Smith, 2024)

### **2.2.3 Basic types of data acquisition systems**

There are many different types of data acquisition systems handheld, portable, modular, and rack mountable. Here a modular system is used to connect to the PLC. (Smith, 2024)

### 2.3 To successfully acquire data from a plc

It is important to know how a PLC is set up. Input-output modules PLCs interface with the system through input and output modules these modules convert signals from sensors and actuators into digital data it can process. Having knowledge of these modules is important for effective data acquisition. Data addressing PLC programs access the data from the input and output modules via data addresses. Understanding data addresses and what data type they are is important to import the right data. Data processing PLC programs can perform various processing on the data such as scaling, arithmetic operations, and logical operations. Knowing how this works and when to acquire the signals is crucial to get the right data. Communication protocols PLCs use different protocols profinet, Modbus, etc knowing what protocol is used is important to get the right data.



Figure 3 Siemens output module

### 2.4 Basic PLC functions

Programmable logic controllers (PLCs) serve as the backbone of modern industrial automation providing a robust platform for controlling and monitoring automation processes. Understanding how a PLC works and its basic functions is necessary to make a good data acquisition system. (Polycase, 2021)

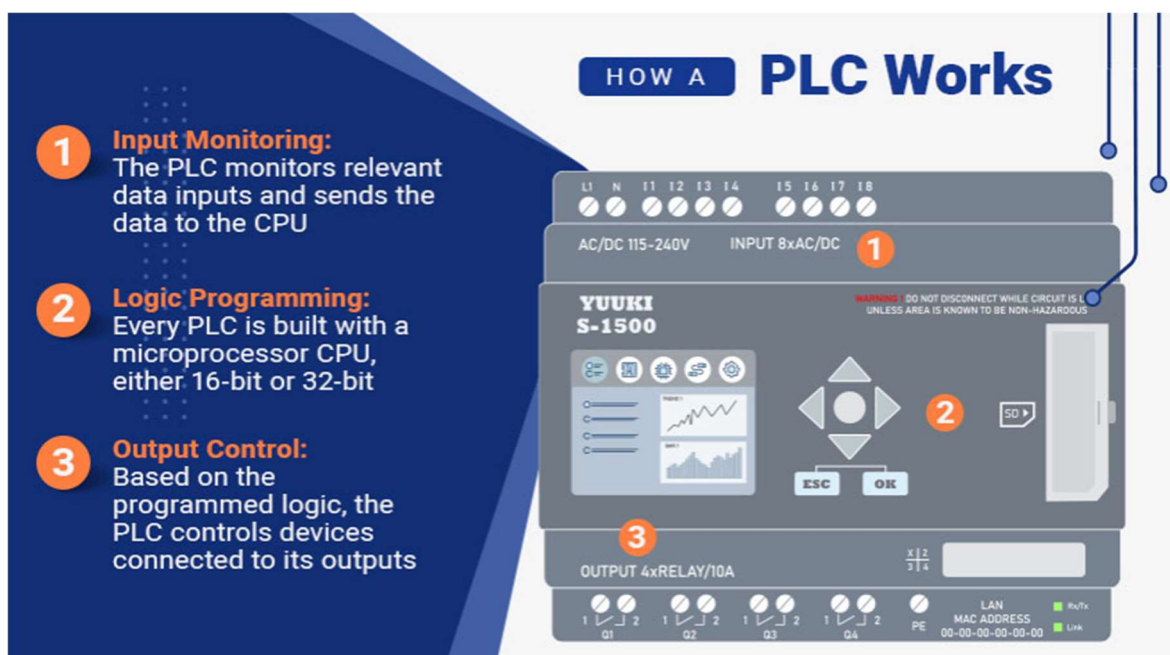


Figure 4 Polycase plc

### 2.4.1 Input scan

To get input data from sensors and switches the controller monitors and scans the input modules in real time to update the program accordingly. (Polycase, 2021)

### 2.4.2 Program execution

After the controller has read the input data it runs the user-made program made from different programming languages such as ladder, function blocks, and structured text. This dictates what the PLC will do with the information and determine the appropriate actions to be taken. (Polycase, 2021)

### 2.4.3 Output update

Following the program execution, the PLC updates the output modules which control relays, actuators, etc in accordance with the results from the program evaluation. Effecting changes that align with the program. (Polycase, 2021)

#### **2.4.4 Continuous scan cycle**

The repetitive nature of PLC operation is embodied in the scan cycle where the PLC repeats the cycle of input scanning program execution and output updating. This seamless cycle ensures real-time monitoring and control. Making the PLC safe and cost-efficient in industrial environments. (Polycase, 2021)

#### **2.4.5 Data storage**

A PLC usually has an internal memory to store the user-made program and input and output status and other critical information to maintain system operation. The data storage capability ensures the controller can start up quickly after a power outage. (Polycase, 2021)

#### **2.4.6 Communication capabilities**

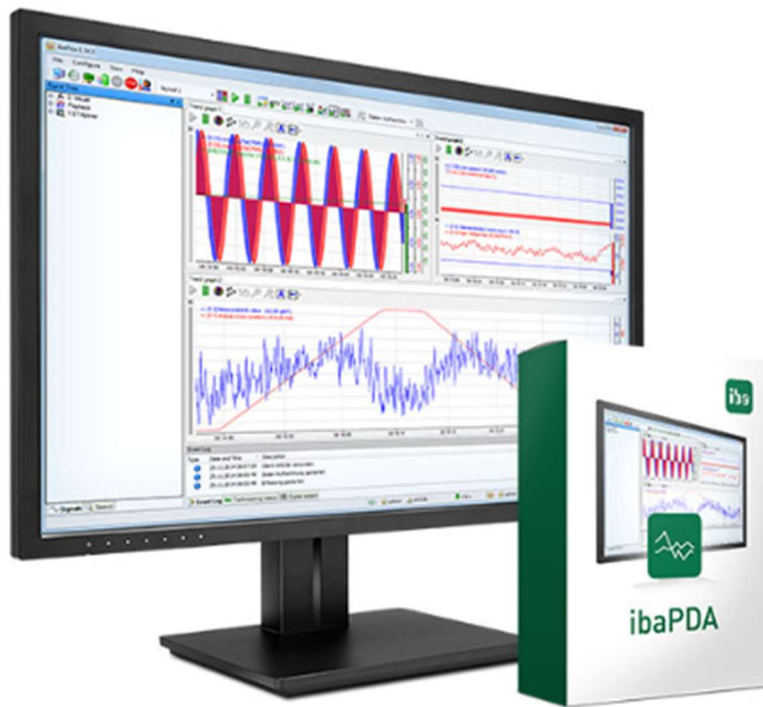
Most PLCs can communicate with other devices such as human-machine interface (HMI), supervisory control data acquisition (SCADA) systems, other PLCs, and other smart devices. Usually via a connection bus such as Profinet, Modbus, etc. (Polycase, 2021)

#### **2.4.7 Ladder**

Ladder is a programming language often used in PLCs the name ladder originates from the visual representation of the programming language which in the early days looked like a ladder. Ladder is made of different rungs which are horizontal lines connected to instructions. For inputs some contacts are connected to tags from sensors and actuators the contacts are usually normally open (NO) or normally closed (NC). For outputs, there are coils that are connected to tags from outputs such as relays. These coils are activated when all contacts and statements are true in a rung. There are several logical functions in ladder such as AND, OR, and NOT, as well as arithmetic functions and support for function blocks. Overall ladder is an easy-to-understand-programming-language. (Polycase, 2021)

## 2.5 ibaPDA

ibaPDA (intelligent basic data acquisition) is a comprehensive software designed for real-time data acquisition it is commonly used in process industries like energy, utilities and more. The program offers good characteristics such as real-time data acquisition down to 1ms sampling timebase from various PLCs and SCADA systems including Siemens. ibaPDA also has a customizable interface to easily select which signal to show and dashboards enabling users to visualize specific data. It has comprehensive data analysis tools to make analytics from real-time data such as charts, graphs trends, etc making it easy to understand the data. ibaPDA has good historical data storage options like select data signals to store or store the entire project to either a data file or database with many data streaming options such as Azure, Microsoft SQL, Postgresql, Saphana, and many more. ibaPDA has alarm and event management built-in with both internal and external alarms to minimize downtime. It has support for OPC UA and SNMP to make alarms to SCADA systems. Integration ibaPDA is designed to seamlessly integrate with existing or new systems with support to most of the industrial communication protocols such as Profinet, Modbus, ethercat, and many more it also has easy scalability for up to unlimited signal acquisition. Overall ibaPDA is a good and flexible data acquisition solution for small and big projects. (ibapda, u.d.)



**Figure 5** ibaPDA

## **2.6 PostgreSQL**

PostgreSQL is a powerful open-source object-relational database system that uses and extends the SQL language. Combined with many features to safely store data. PostgreSQL dates back to 1986 when it was a project made at the University of California at Berkeley and has over 35 years of development. PostgreSQL has a strong reputation for its proven architecture. PostgreSQL runs on all major operating systems and has been ACID-compliant since 2001 and has many popular add-ons. (About postgresQL, 2023)



**Figure 6 PostgreSQL logo**

### **2.6.1 Why use PostgreSQL**

PostgreSQL comes with many powerful features to help developers build stable databases and protect data. PostgreSQL is open source and free to use no matter how big or small of a project. PostgreSQL is highly customizable to build custom functions and custom data types. PostgreSQL tries to conform with the SQL standard and only differs where conformance could lead to poor architectural decisions as of September 2023 it conforms to 170 of the 179 mandatory features. (About postgresQL, 2023)

### **2.6.2 Reliability and stability**

PostgreSQL is known for its reliability and stability making it a suitable database where no data loss is critical. It has a robust reputation and has been proven in production environments across a wide range of industries. (About postgresQL, 2023)

### **2.6.3 Data types and data integrity**

PostgreSQL supports a wide range of data types and the ability to make user-defined data types the built-in data types.

- Primitives integer, numeric, string, Boolean
- Structured date/time, array, range/multirange, UUID
- Document JSON/JSONB, XML, key-value(Hstore)

- Geometry point, line, circle, polygon
- Customizations composite, custom types

#### Data integrity

- UNIQUE, NOT NULL
- PRIMARY keys
- Foreign keys
- Exclusion constraints
- Explicit locks, advisory locks (About postgresQL, 2023)

### **2.6.4 Performance**

While performance can vary depending on the configuration and hardware PostgreSQL is generally known for delivering good performance especially when optimized right. (About postgresQL, 2023)

### **2.6.5 PGAdmin**

PGAdmin is an open-source administration and development platform for PostgreSQL databases. It has a graphical interface for users to manage and develop their PostgreSQL database effectively. The graphical interface is easy to use and offers an easy view of the whole database. It can also manage multiple databases from one PGAdmin client. It also has a powerful SQL query tool to execute queries to the PostgreSQL database and provides syntax marking code completion. PGAdmin has a good data viewing and editing program built in to see tables and their contents and edit data. The program has a monitoring tool to see how many queries and data in incoming to the database. It is easy to backup and restore databases either full or partial or to backup them on a schedule. User management is easy to do with PGAdmin to create modify or delete users. PGAdmin is easy to customize with many add-ons available for free. (About postgresQL, 2023)

## 2.7 VMware workstation

VMware Workstation is a virtualization software developed by VMware Inc. It runs on Windows or Linux and is the industry standard for desktop hypervisors. It has the capability to run multiple VMs on the same host computer. With its virtualization technology it can create multiple virtual machines and each VM behaves like a separate computer with its own CPU, RAM, and network interface. VMware workstation takes advantage of hardware-assisted features in modern CPUs. It has built-in vSphere and ESXi support to easily connect to a remote vSphere environment to quickly manage and deploy virtual machines. VMware workstation also has good cloning capabilities essential for making a virtual base server to quickly configure and deploy in new projects. It is good for making test environments because you can make an isolated environment and put all test equipment in that environment to make the test like it is in a real end project. VMware Workstation is also heavily integrated into other products making it easy to work between many different VMware products for seamless integration. By creating the virtual server with VMware Workstation, the needed resources can be easily allocated to the server, and there will still be resources available to allocate to more resource-intensive drawing programs. (Wmware, 2024)



Figure 7 VMware workstation pro logo

## 2.8 Windows Server 2022

Windows Server 2022 is the latest version of Microsoft's server operating system it was released in August 2021. Windows Server 2022 is designed to provide a platform to manage and host server-based applications. Windows servers also have advanced security features essential for making a secure server for storing data in a PostgreSQL database. Windows server also has support for hybrid cloud into Microsoft Azure. Windows Server 2022 can be run as a virtual machine or a standalone computer. The server is being used as a virtual machine. The system has good container improvements from previous versions making it good for future improvements reducing disk size and better compatibility with Kubernetes orchestration. Overall Windows Server 2022 provides a robust platform for hosting PostgreSQL and ibaPDA with the security performance and management capabilities for industrial workloads in virtualized environments. (investing in your success with Windows Server, 2022)



Figure 8 Windows Server 2022

## 2.9 Siemens TIA portal

Siemens TIA Portal (Totally integrated automation) is a software used for engineering, programming, and commissioning for Siemens automation products. It is a central integrated engineering framework for various Siemens devices such as PLCs, HMIs, drives, and other industrial components. Here are some aspects of Siemens TIA. (Totally integrated automation portal, 2023)

### **2.9.1 Integrated automation and support for multiple devices**

Tia portal provides a large platform where users can configure and program the whole automation system if all products are from Siemens. It allows users to configure all automation components from a single program reducing complexity for engineers and taking away the need to have installed many different automation programs. The multiple-device support also supports older devices so you can easily integrate new and old products in a system making it easier to upgrade systems. (Totally integrated automation portal, 2023)

### **2.9.2 Graphical programming**

TIA has support for graphical programming for PLCs such as ladder, functions block diagrams, and structured text. This simplifies the programming coding process making it good to make a test environment. (Totally integrated automation portal, 2023)

### **2.9.3 Simulation and Testing**

TIA provides simulation tools to allow engineers to efficiently test their programs and compile programs. It allows you to virtually simulate the program so you will not need the input and output modules to test. This is needed as only a test CPU is available, rather than a whole PLC rack with input and output modules. (Totally integrated automation portal, 2023)

### **2.9.4 Diagnostics and monitoring**

In TIA built-in tools can be utilized to help monitor your program and have real-time monitoring of certain inputs or blocks. This helps to see if ibaPDA gets the right data and to see if the right data gets sent to the database. (Totally integrated automation portal, 2023)

### 2.9.5 Scalability

TIA Portal is very scalable allowing users to adapt to various automation projects of different sizes and complexity. (Totally integrated automation portal, 2023)

### 2.9.6 Overall

Siemens TIA Portal is perfect to use as a test CPU with its easy-to-make projects and monitoring tools. This is essential when testing data streaming with quick changes. (Totally integrated automation portal, 2023)



**Figure 9 Siemens TIA portal**

## 2.10 SNMP

SNMP stands for simple network management protocol and is an application layer defined by an internet architecture board made to exchange handling info between network devices. SNMP is one of the biggest protocols to manage and supervise network elements. An SNMP client-server architecture has 3 components: an SNMP manager, SNMP agent, and a MIB (management information base). The SNMP manager acts as the client, the agent acts as the server and the MIB acts as a database when the manager asks the agent a question the agent uses the MIB to answer. SNMP is so popular that most network devices come pre-bundled with agents to use the protocol. ibaPDA comes with an SNMP agent to use with different tags. The IsMeasuring tag is used to ensure that data is sent to the database. MIBs can be created for any network device for example video cameras, ibaPDA or industrial equipment.

SNMP is called simple because of the uncomplicated nature of the protocol. SNMP can issue read or write commands and even reset passwords. Here SNMP only reads commands overall SNMP is a good choice when you want to ensure a device works. (Scarpati, u.d.) (SNMP guide, u.d.)

## SNMP configuration

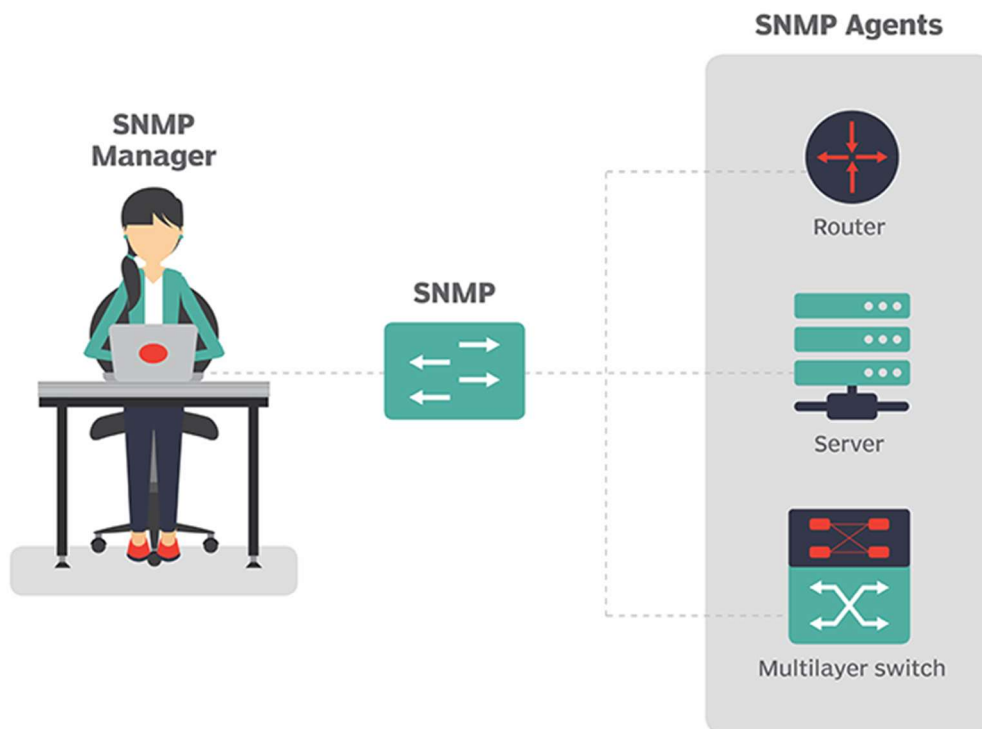


Figure 10- SNMP configuration

### 2.11 OPC UA

OPC UA stands for OPC unified architecture and is a machine-to-machine communication protocol. OPC UA is a secure and reliable data exchange between industrial devices. OPC UA is supported by a lot of suppliers and vendors which makes it easy to use between different devices. OPC UA has a lot of security implementation. It provides encryption and authentication to ensure confidentiality and integrity. It is highly scalable from a single signal from a device to many signals. OPC UA is utilized to send data to WINCC OA if ibaPDA is streaming data to the PostgreSQL database this is good because OPC UA has mechanisms to ensure the data is transmitted accurately and efficiently to WinCC OA. OPC UA is a good communication protocol to use between industrial devices. It is used to make

sure data is sent to the database and the operator can see it in the SCADA system. (Unified architecture, u.d.)

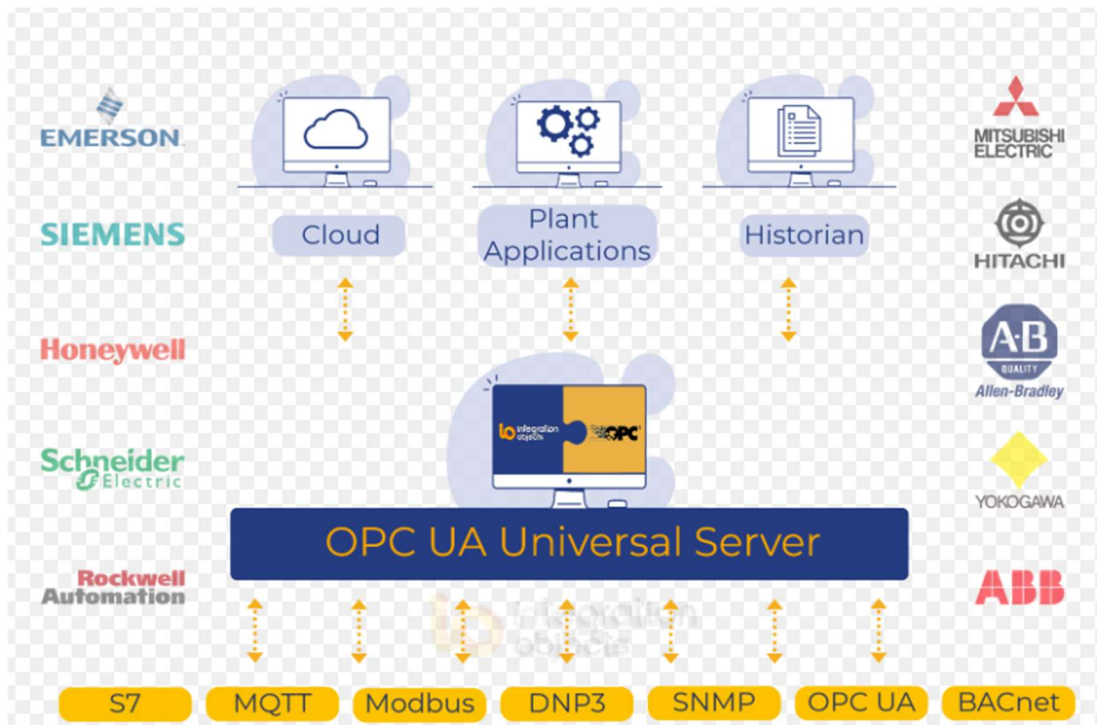


Figure 11- OPC UA

## 2.12 WinCC OA

WinCC OA is the abbreviation for SIMATIC WinCC Open Architecture. It's a SCADA software package designed for use in automation technology. The main application area is the operation and control of technical plants. This application in addition to presenting current process states can also send conditions and commands to the control equipment. WinCC OA also has historical archiving to store what the system has done. WinCC OA is modularly built so you can choose what functions you use. WinCC OA is used to see if ibaPDA is sending data to the PostgreSQL database and if there is no data being sent an alarm will come active for the operator to see. Because it is an open architecture different manufacturers can make their product work with it. Overall WinCC OA is a good SCADA system that many engineering firms use to make their SCADA systems. (Architecture, u.d.)

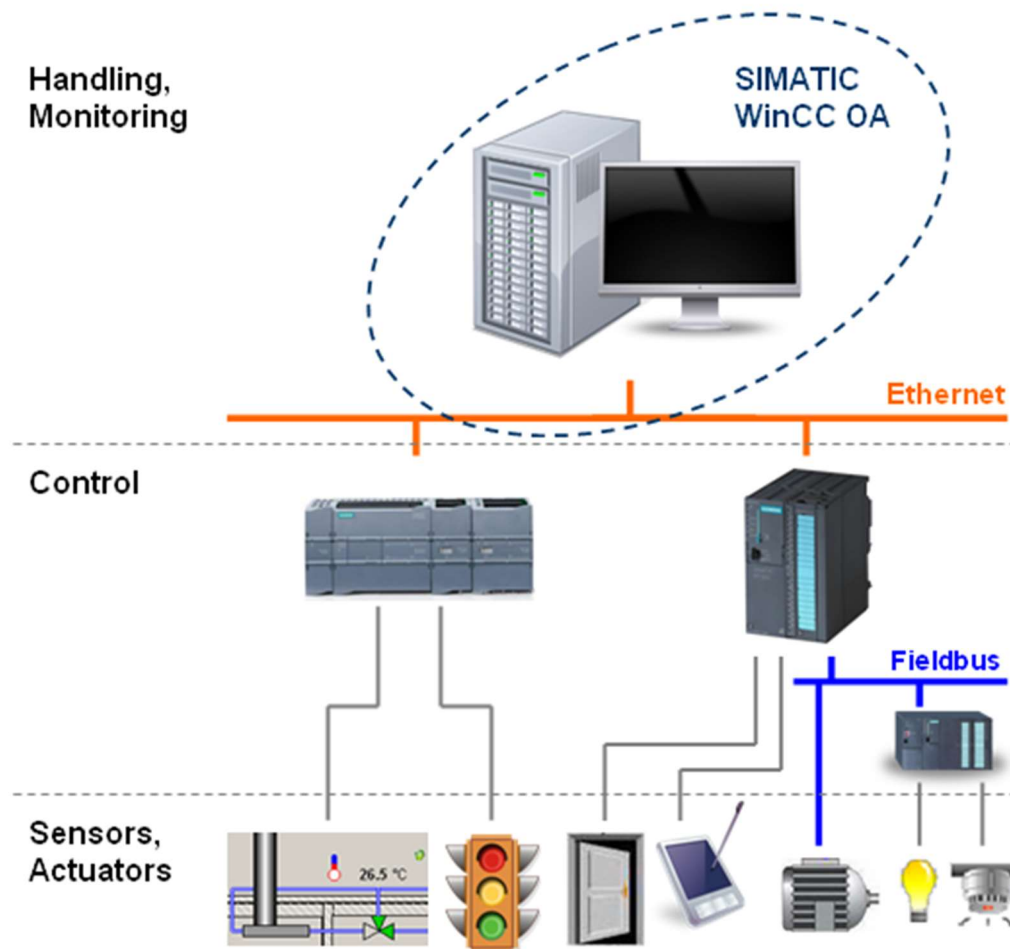


Figure 12 The role of WINCC OA in an automation system

### 2.13 Task Scheduler

Task Scheduler is a built-in program in Windows Server 2022 that enables you to automatically perform routine tasks such as a data deletion bat file run every day. It does this via triggers the developer chooses and performs the task when the trigger is met such as run a task once every day at 12.00. Task scheduler is used by system administrators to make a program run when certain conditions are met without user interaction. (Task scheduler for developer, 2023)

## 3 Methods and approaches

To test if data streaming is possible a test setup is made to test all the different things that are required to work. first Windows server was installed using VMware Workstation, and

ibaPDA and PostgreSQL were installed on it. Then a different virtual machine is installed with Siemens TIA installed. A Siemens 1517 PLC is used to make a test program. Last a virtual machine with WinCC OA will have a simple program to test if data is being sent to the database. After everything is installed ibaPDA is set up with a connection to the PLC to record signals from the PLC and send them to the PostgreSQL database. ibaPDA will also be connected to the WinCC OA virtual machine and send data to it via OPC UA and SNMP.

### 3.1 Test setup

Three virtual machines are being created using VMware Workstation one with Windows Server 2022 installed, ibaPDA, and PostgreSQL another with Siemens TIA and the last one with WinCC OA. A Siemens S7-1500 1517 PN DP PLC is being used for testing purposes. All virtual machines and the PLC will be connected to the same network via a switch.

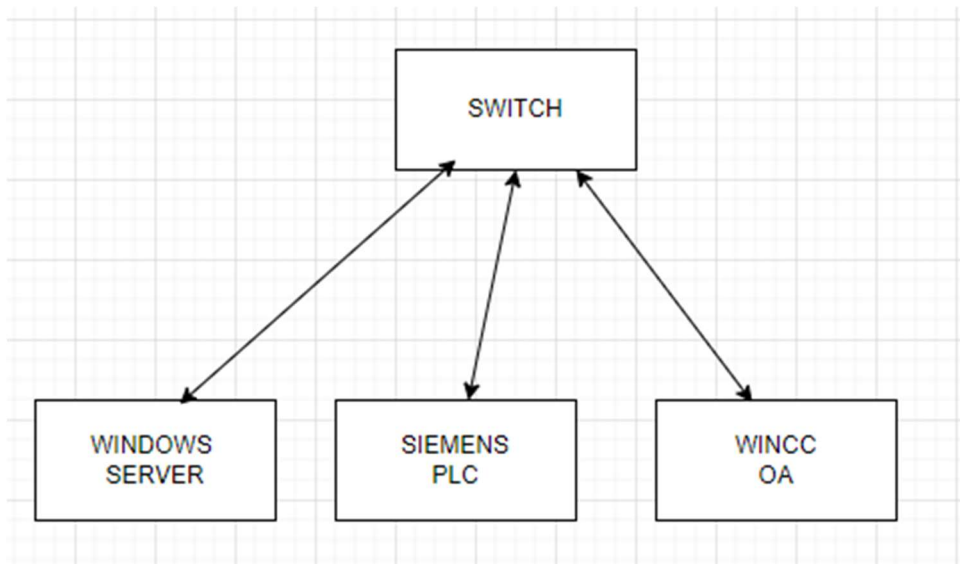


Figure 13 Connection diagram

### 3.2 Installing Windows Server 2022

Installing Windows Server 2022 with VMware Workstation is straightforward. The ISO file was used, prompting VMware to initiate the easy install process automatically. Windows Server 2022 Standard Edition was selected, and a username and password were provided. Then, the location for the virtual machine installation was chosen, and the process was

finalized by clicking "Finish". Windows installation was automatically carried out by VMware.

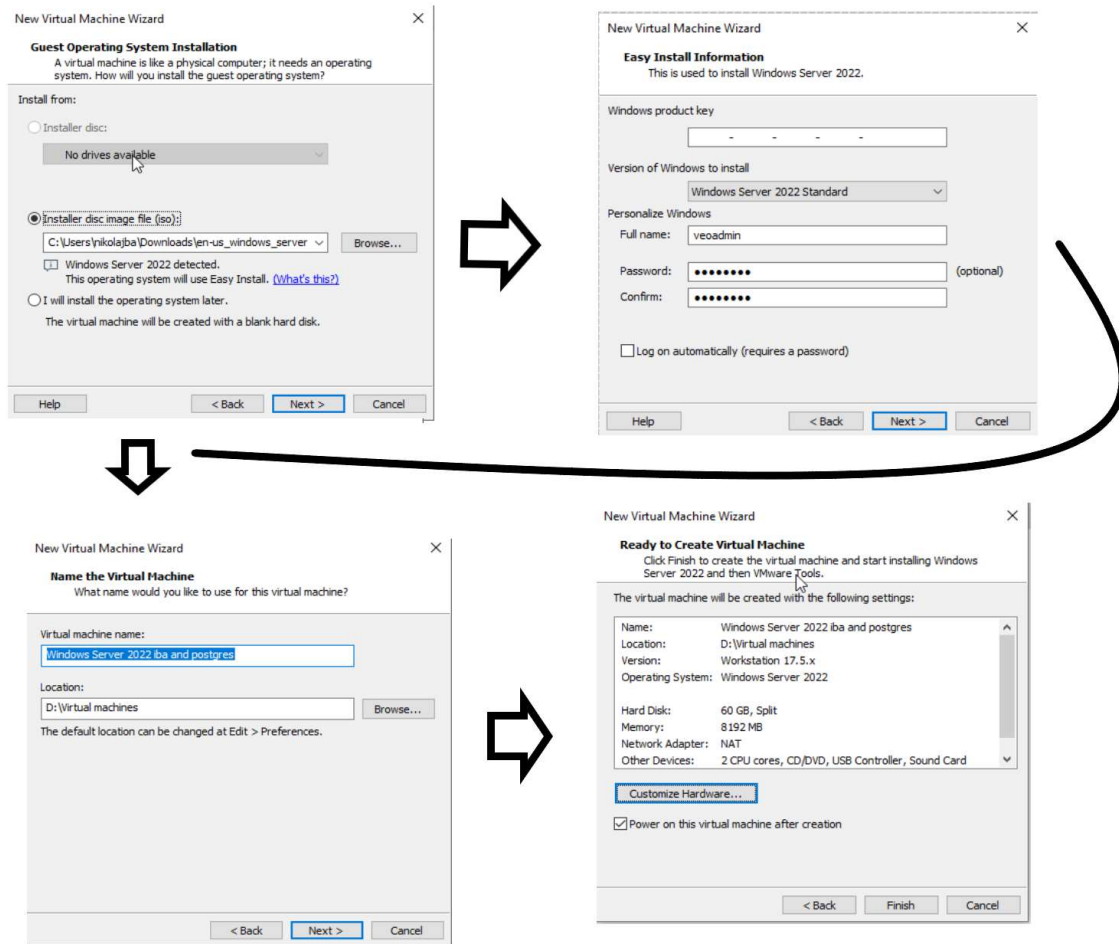
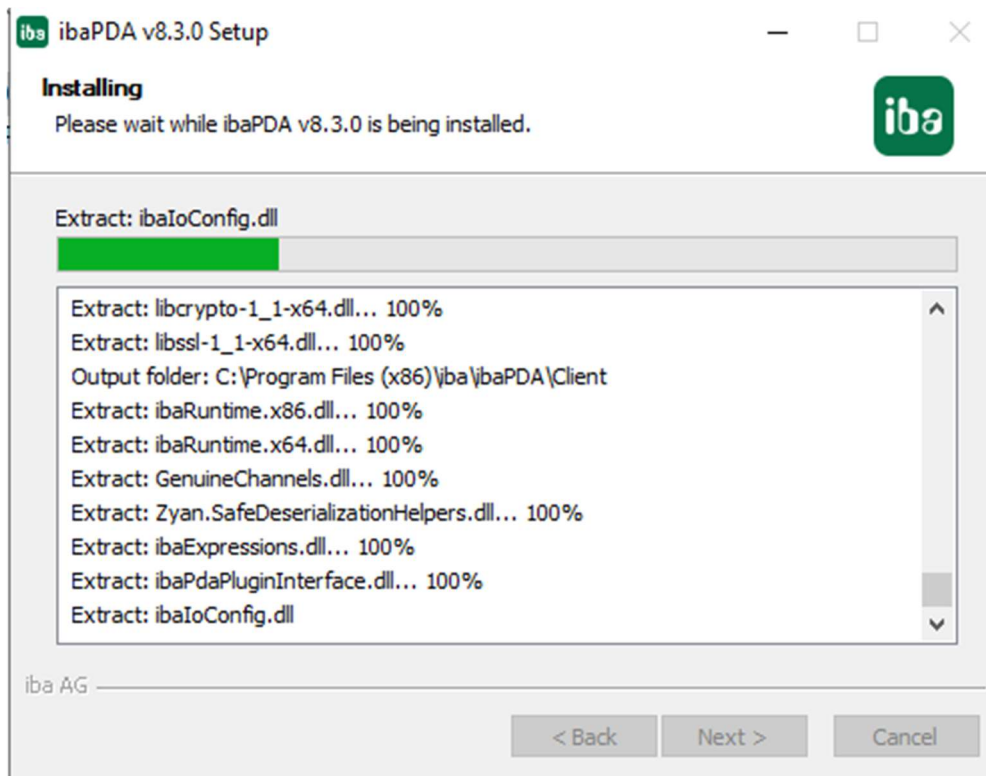


Figure 14 VMware install diagram

### 3.2.1 Installing ibaPDA

Then ibaPDA is installed with an install file I transferred over to the virtual machine.



**Figure 15 Install progress**

After ibaPDA was installed, the network settings were configured to ensure it is within the same network as the other devices. The PLC's IP address is 10.10.10.81. The server's IP address was set to 10.10.10.20 by accessing the network adapter settings, specifically by navigating to "Change adapter settings" and selecting "Internet Protocol Version 4 (TCP/IPv4)", then specifying to "Use the following IP address".

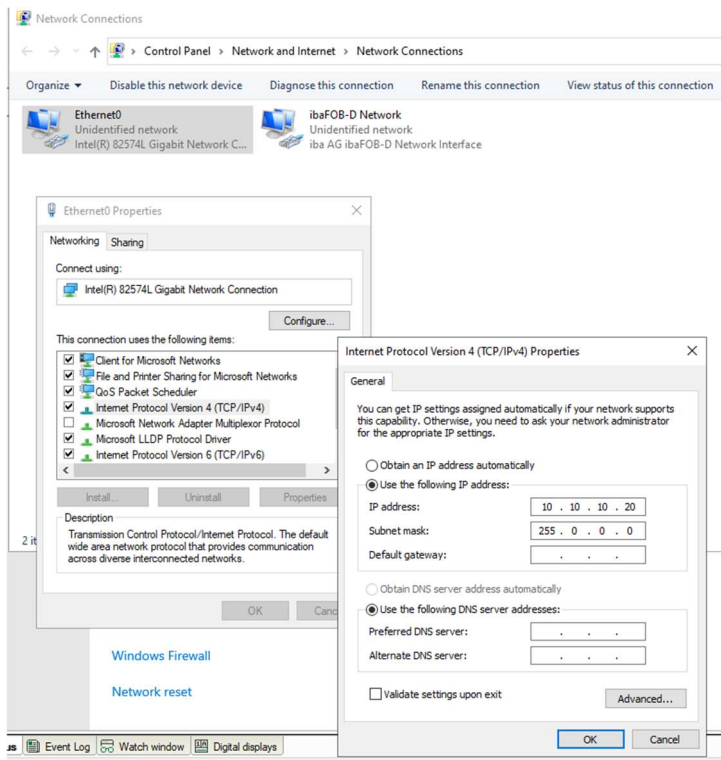


Figure 16 IP configuration settings

### 3.2.2 PostgreSQL installation

The PostgreSQL installation file was transferred to the virtual machine. Subsequently, the installation wizard was navigated through to set up PostgreSQL.

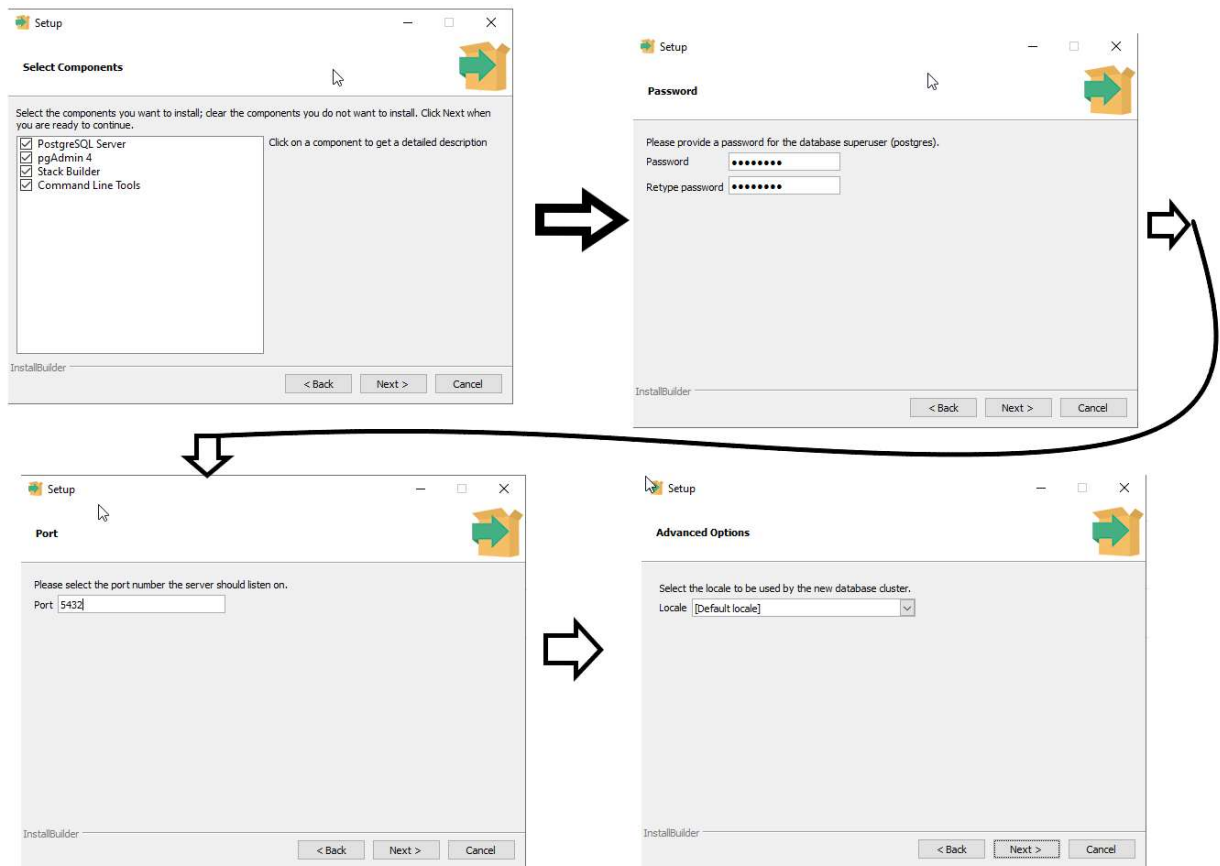


Figure 17 PostgreSQL wizard

PGAdmin, the PostgreSQL management software, was started, and verification was conducted to ensure that everything was functioning properly.

### 3.2.3 Installing Siemens TIA portal and WinCC OA

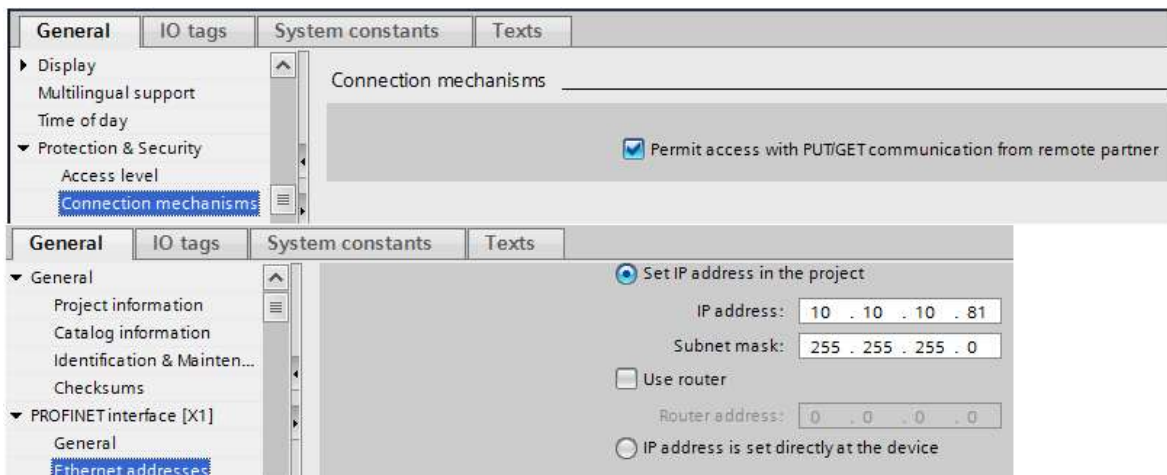
The Siemens TIA virtual machine was created by copying a pre-made virtual machine with TIA version 16 installed on VMware Workstation. Similarly, for the WinCC OA virtual machine, a pre-made virtual machine with WinCC OA installed was copied. Subsequently, the IP addresses for both computers were configured to be within the same network as all the other devices, as illustrated in Figure 15.

### 3.2.4 Setting up PLC and switch

The switch and PLC setup involved connecting the switch to a power outlet and the PLC to another power outlet. A memory card was inserted into the PLC. Subsequently, the PLC was connected to the switch, and all virtual machines were also connected to the switch, ensuring that they are interconnected with each other as depicted In Figure 12.









### 3.3 PLC test program

The test program was created using the machine equipped with TIA Portal. A new program was initiated, and an S7-1500 1517 PN DP CPU was added. In the CPU properties, the IP address was configured to 10.10.10.81. Additionally, PUT/GET communication was enabled under Connection Mechanisms to facilitate communication between ibaPDA and the PLC.



**Figure 18 Communication settings for PLC**

Tags were created to be utilized and recorded for ibaPDA. Within the PLC tags, 3 digital inputs and outputs, as well as 3 analog outputs, were generated. Boolean data types were employed for the digital tags, while real data types were used for the analog tags.

Default tag table							
	Name	Data type	Address	Retain	Acces...	Writa...	Visib
1	 input1	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	 input2	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	 input3	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	 output1	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	 output2	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	 output3	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	 analog1	Real	%MD0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	 analog2	Real	%MD4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Figure 19 Tag table**

A Function Block was incorporated using ladder as the programming language, consisting of 4 networks. In the first 3 networks, a normally open contact and output were included. The tags previously created were connected, such that input 1 and output 1 were linked in network 1, and the same configuration was applied to networks 2 and 3.

In network 4, 2 move instructions were added, with 50.1 and 49.9 designated to move into analog 1 and analog 2, respectively.

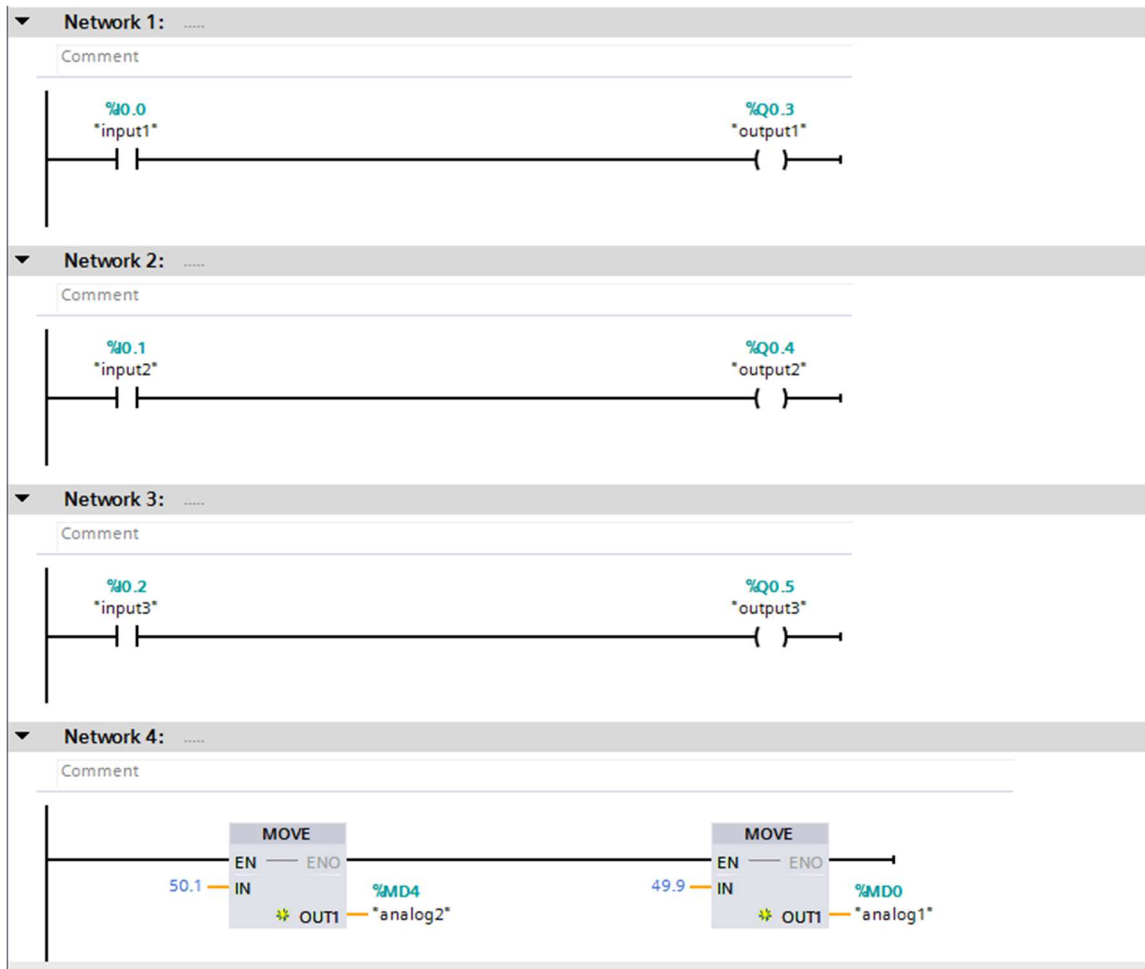


Figure 20 PLC network layout

The function block was subsequently integrated into the main block to activate its functionality.

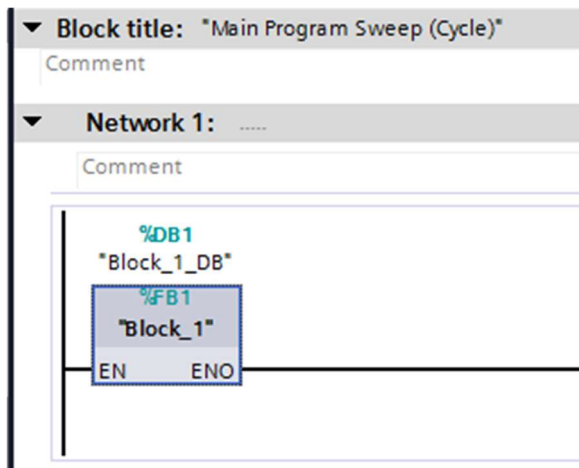


Figure 21 Main network

### 3.4 Configuring IBA settings

To configure ibaPDA, the program was initiated and the PLC intended for monitoring was incorporated. The address book was then imported, connections to the database and SCADA system were set up, and charts were incorporated to visualize the values retrieved from the PLCs.

#### 3.4.1 Add PLC and import address book

To establish a connection between ibaPDA and a Siemens PLC, the PLC program was exported to the virtual machine where ibaPDA is installed. This facilitated the creation of an address book within ibaPDA, enabling the linking of Siemens tags to its internal tags.

Subsequently, the I/O manager was accessed, and an S7-Xplorer was added to the connection tab, configured with the PLC's IP address.

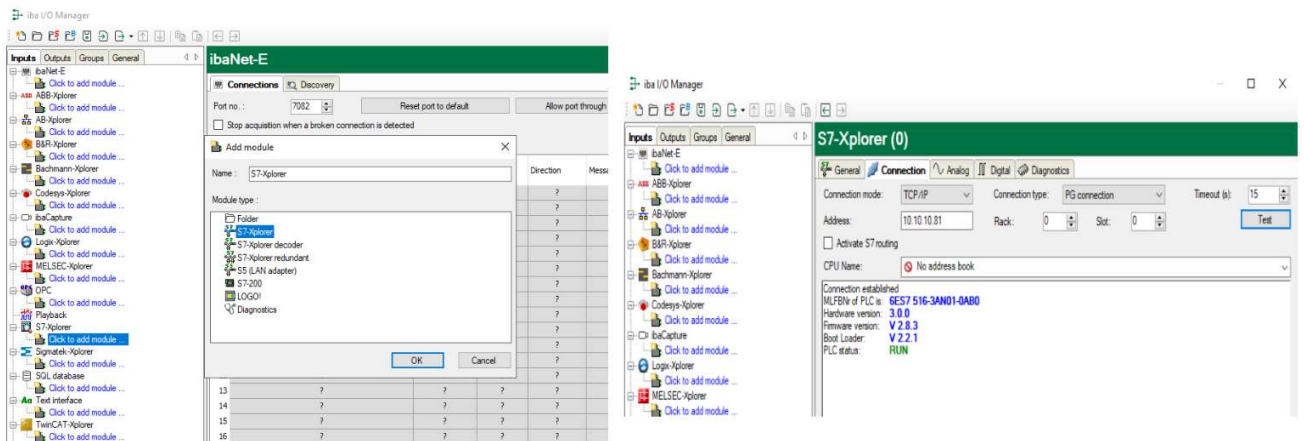


Figure 22 PLC connection

To create an address book the "Create Address Book" option is selected, and then browse to the location where the PLC program was imported. The .ap16 file is added, and ibaPDA automatically creates the address book.

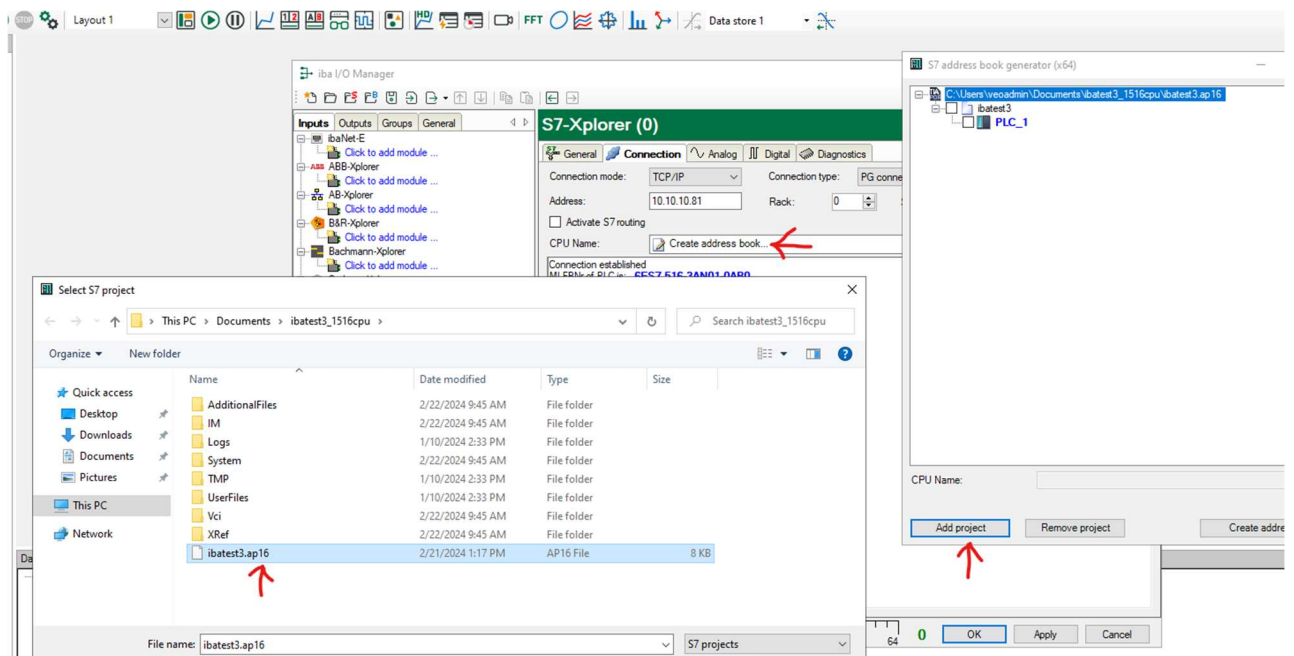


Figure 23 Address book

### 3.4.2 Configuring in and Outputs

Inputs and outputs are configured by navigating to the analog and digital tab, where the tags to be monitored are added. It is ensured that they are the correct operand and datatype, and then "Apply" is pressed.

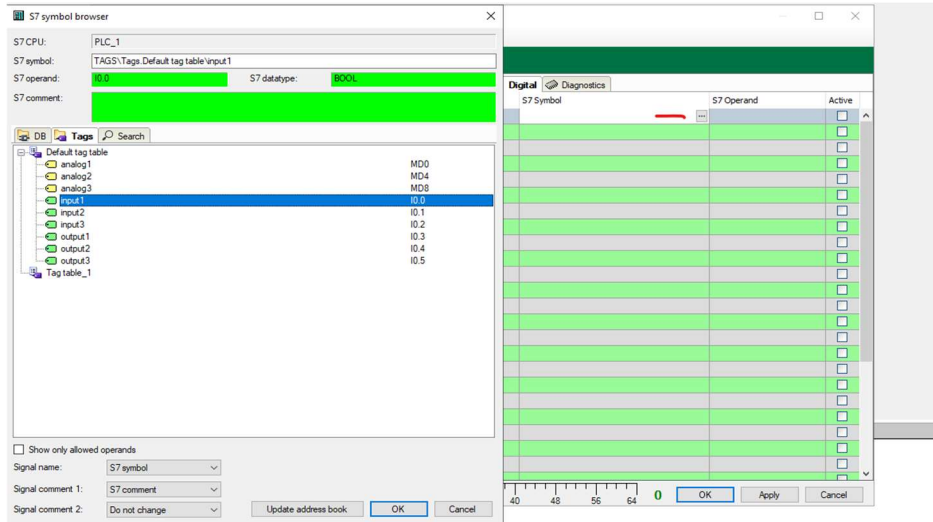


Figure 24 Adding digital connections

Now, the signals from the ibaPDA signal tree are added from the upper left corner to the trend graph to observe the output provided by the PLC. The trend graphs are customizable, set so that all analog signals are in one chart, with all inputs and outputs in separate charts.

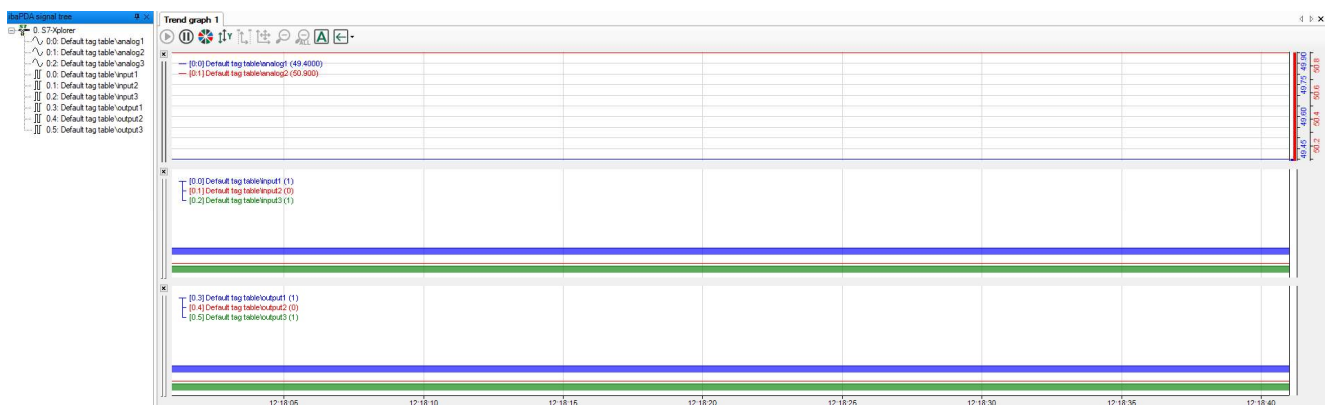
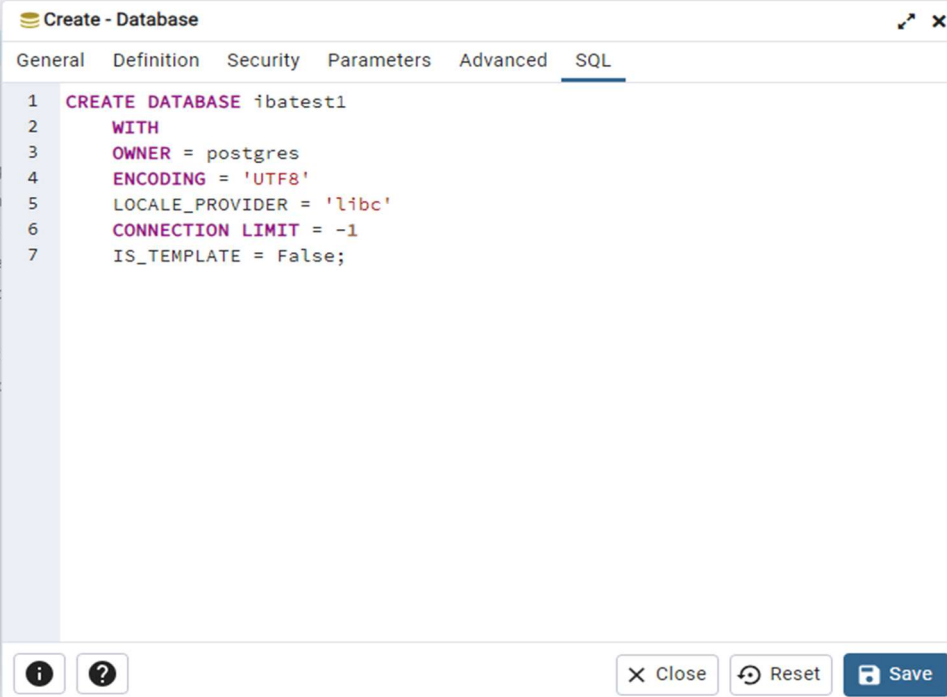


Figure 25- ibaPDA trend graph

Here, what the PLC is outputting in real time can be clearly observed, and specific data can be viewed by zooming in and out.

### 3.4.3 Connection from ibaPDA to database

To establish the connection for data streaming, the first step is to create a database in PGAdmin. PGAdmin is started, logged in, and a new database named `ibatest1` is created by right-clicking and selecting "Create Database." The database can be created either via the user interface or through an SQL statement.



```
1 CREATE DATABASE ibatest1
2 WITH
3 OWNER = postgres
4 ENCODING = 'UTF8'
5 LOCALE_PROVIDER = 'libc'
6 CONNECTION LIMIT = -1
7 IS_TEMPLATE = False;
```

**Figure 26 SQL statement**

After the database is created, I return to ibaPDA and navigate to the I/O manager. In the output tab, an SQL database module is added, and "Create Database Connection" is clicked. The database connection data, including the connection name (localhost as ibaPDA since PGAdmin is installed on the same computer), server address, username, password, and database name `ibatest1` is entered.

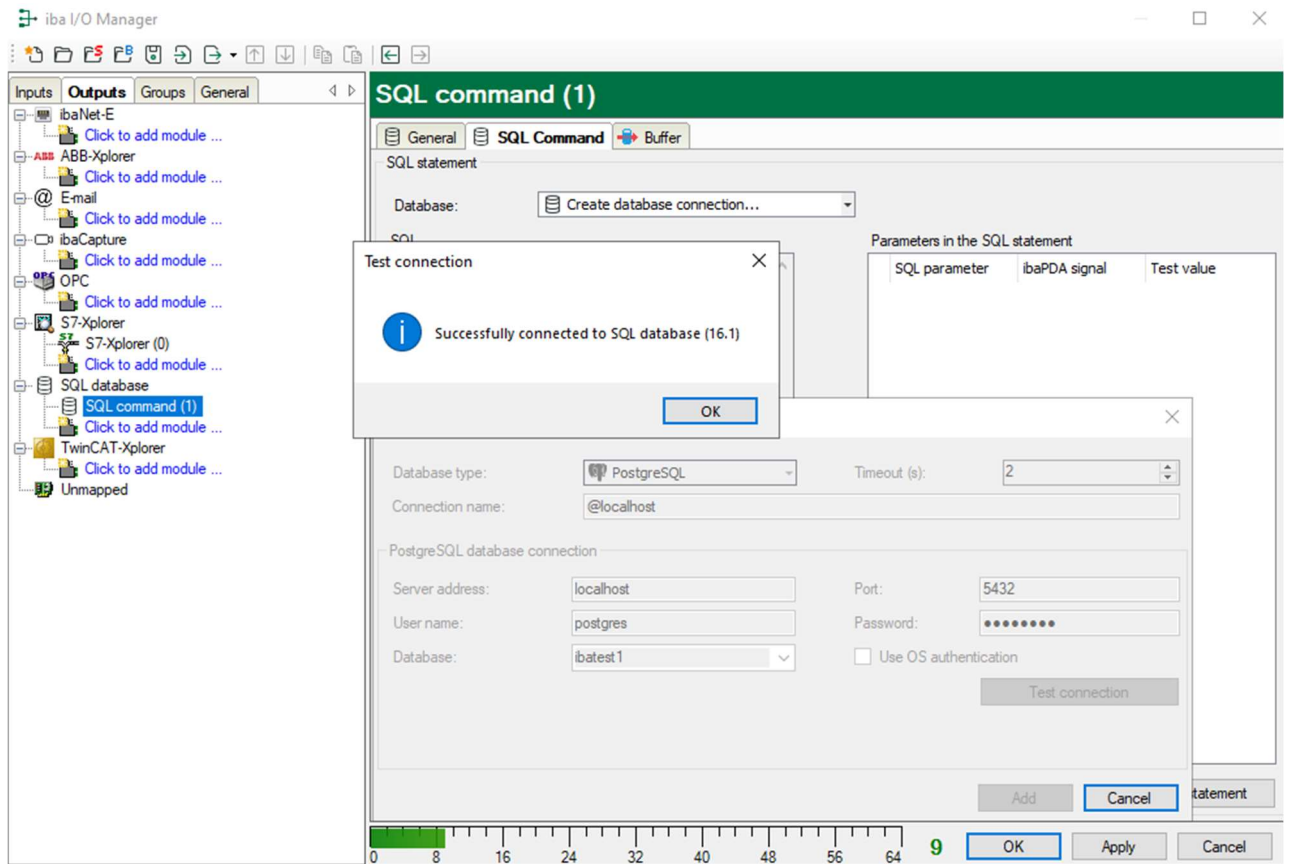
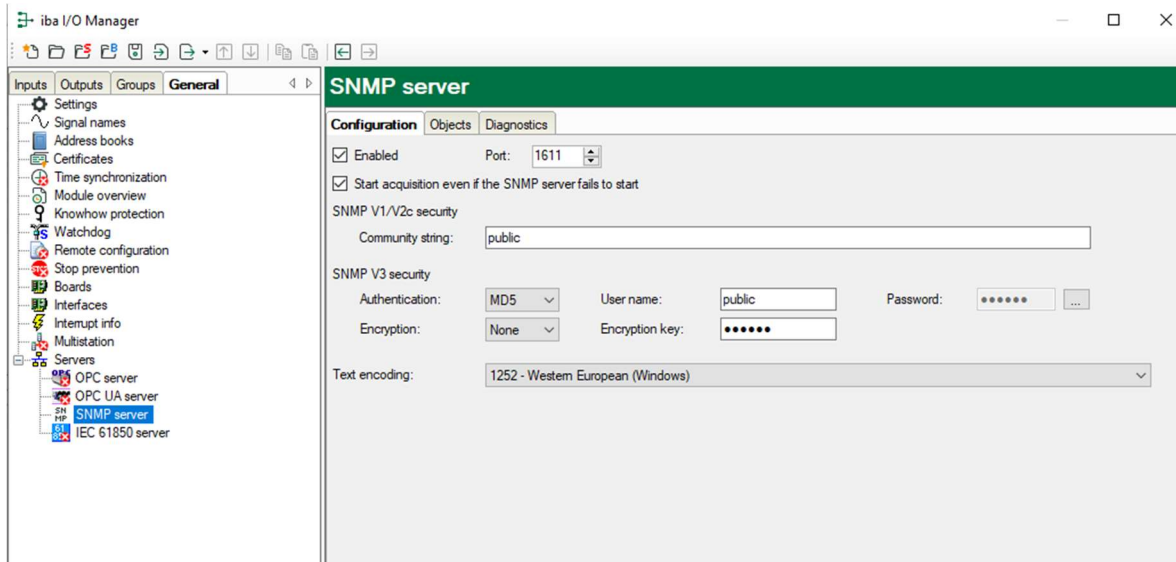


Figure 27 ibaPDA PostgreSQL connection

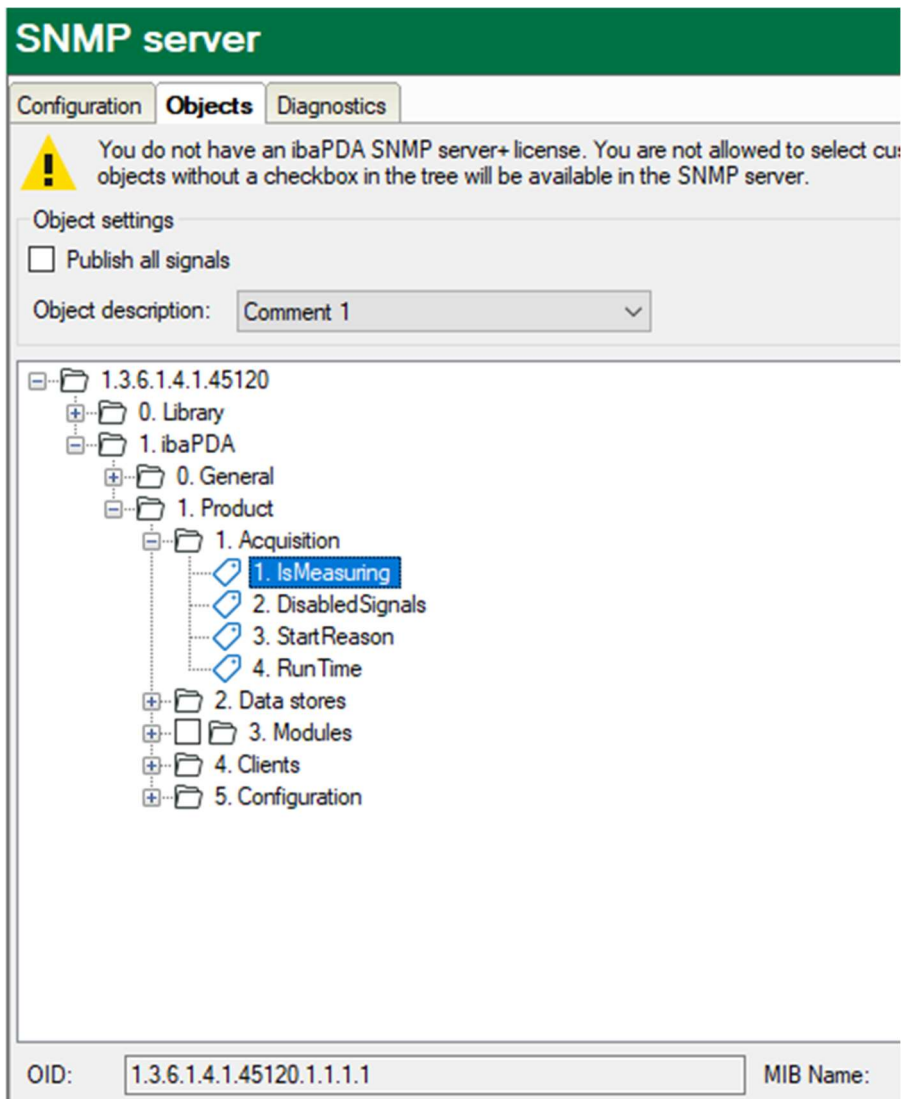
### 3.4.4 Connection from ibaPDA to WinCC OA SNMP

To connect ibaPDA to WinCC OA, the WinCC OA server is started, ensuring it is on the same network as all the other components. Then, in the ibaPDA I/O manager, the general tab is accessed, and the SNMP server is enabled.



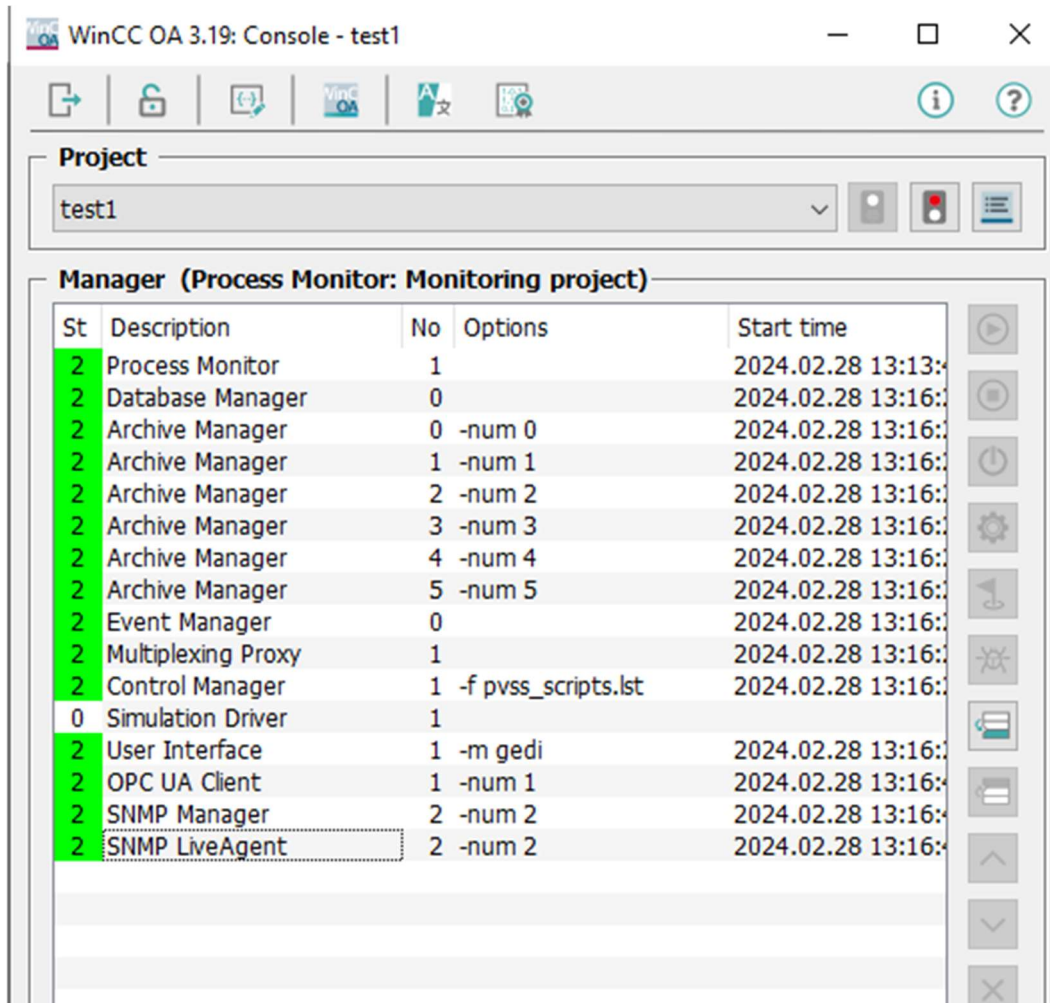
**Figure 28** SNMP enabled

The community string, username, and password are noted. Then, in the objects tab, the IsMeasuring tag is located.



**Figure 29** SNMP IsMeasuring tag

The OID string is noted. Then, the WinCC OA server is started, and a new project is created. Finally, the drivers are started.



**Figure 30 WinCC OA drivers**

The features can be utilized only if the OPC UA client, SNMP manager, and SNMP live agent are ensured to be running. Then, system management -> drivers -> SNMP agents is navigated to, and SNMP version V1, the ibaPDA server IP, and the password in the read community are entered.

SNMP-Agent configuration (System1 - test1; #1)

**Agent/Entity**

\_2\_SNMPSAgent\_1 Create

Description: \_2\_SNMPSAgent\_1 Remove

**Common settings**

SNMP version: V1 Driver number: 2 AgentId: 1

Configuration:  Redundant Agent/Entity Switch on: connTimeout, bad read/write req

Timeout [1/100s] 100 Retries 1

**1. Agent/Entity**

IP address: 10.10.10.20 Port: 1611

Read Community: ..... Write Community:

**Connection state**

State: First agent/entity connected 1. Agent: Connected 2. Agent:

**Figure 31 WinCC OA to ibaPDA connection with SNMP**

After all the data has been entered correctly, the connection state is checked, confirming it as "connected." Then, in the Para tab, a pre-made parameter for SNMP, such as ExampleDP\_INT, is selected. A PowerConfig is created using a fixed address. Next, in the "\_address" field, the OID noted from ibaPDA before is added.

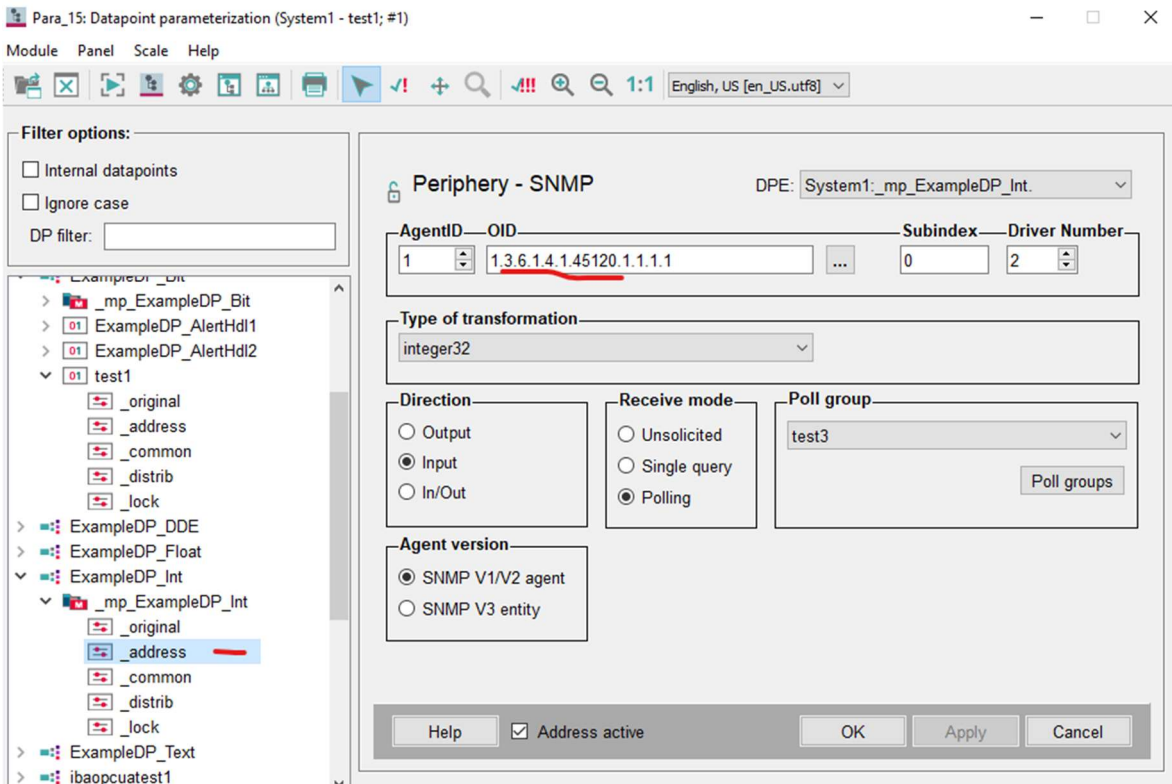


Figure 32 SNMP OID connection

Here, the OID string is set, along with the data direction as input and the SNMP version, which is V1, and "OK" is pressed. Then, in the Panels objects section, a new screen is created, and a rectangle is added. The initialize wizard for the rectangle is selected, and "Change Color" is chosen.

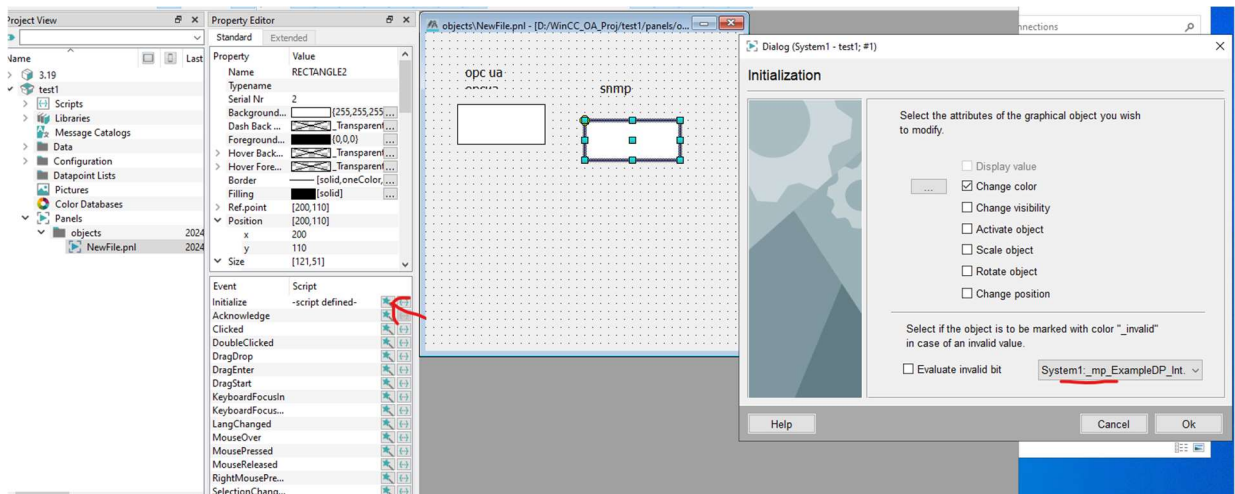
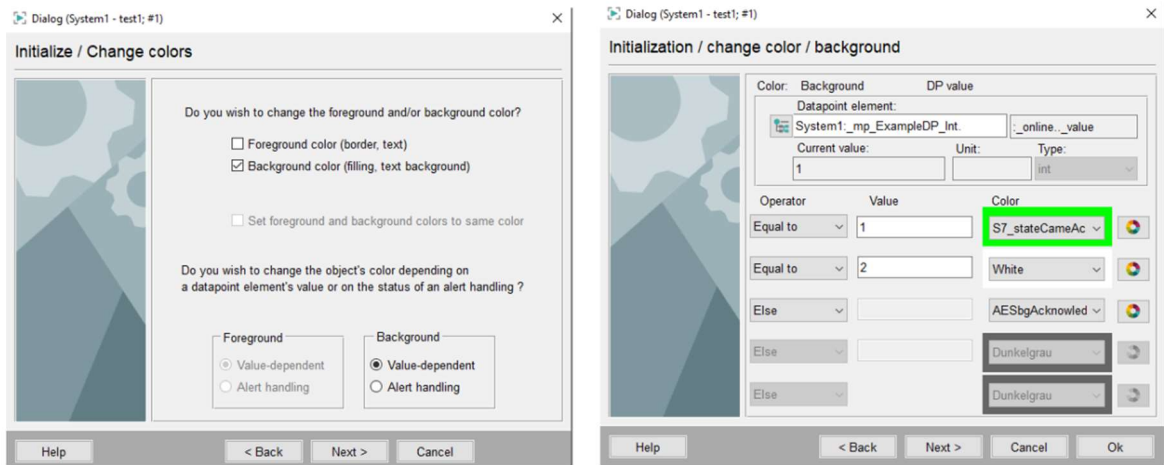


Figure 33 SNMP rectangle wizard

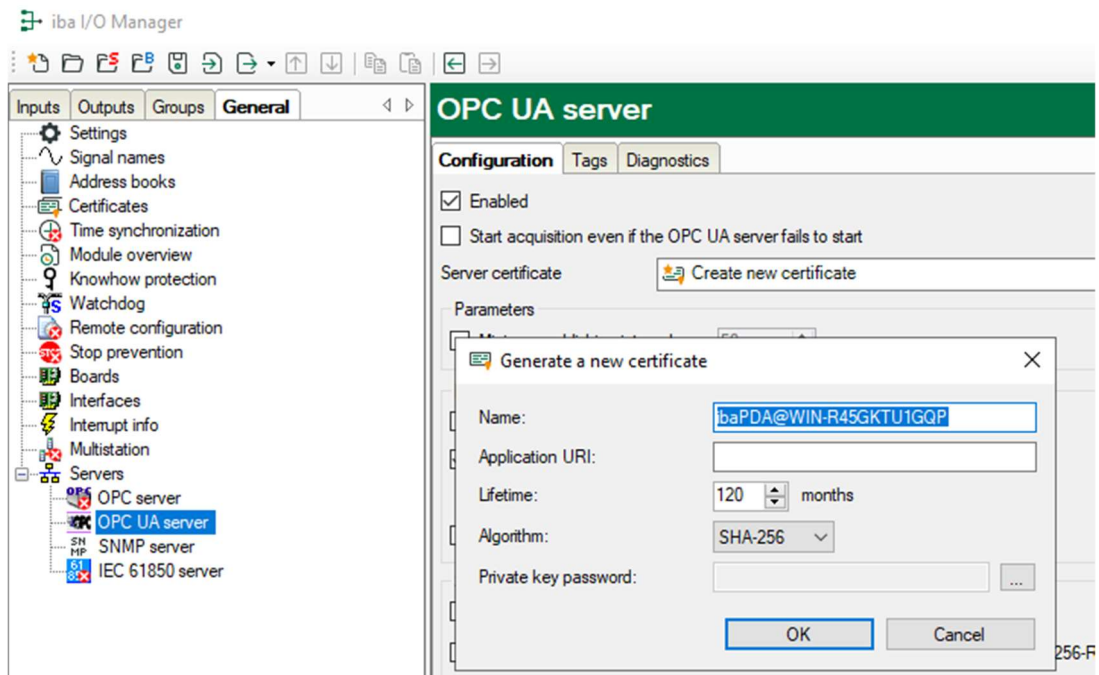


**Figure 34 SNMP rectangle wizard**

Background color and value dependence are selected, followed by choosing the datapoint element and selecting the SNMP tag created in the Para settings. The colors are adjusted such that when the value from the tag is 1, it appears green, and when it's 2, it appears white. This setup ensures that when ibaPDA is measuring data, it appears green, and when it's off, it appears white. With SNMP connected, the next step is to connect OPC UA.

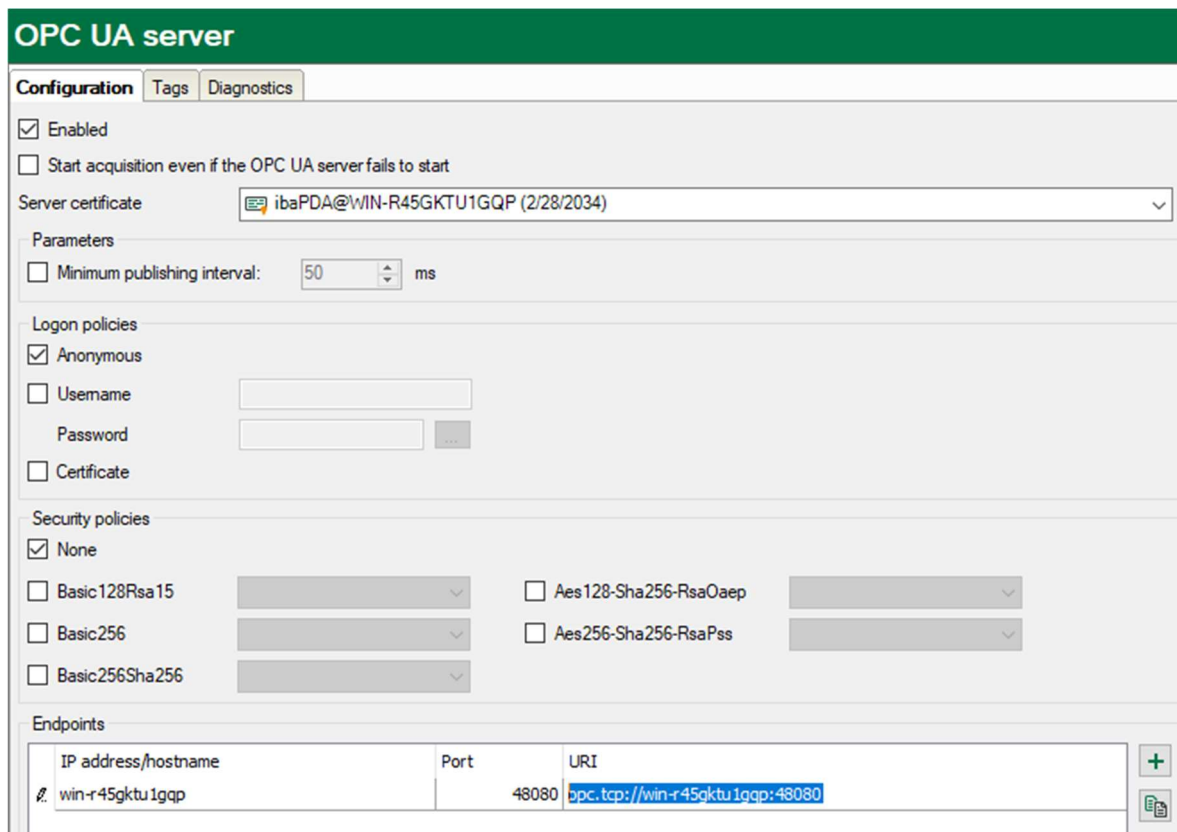
### 3.4.5 ibaPDA connection to WinCC OA OPC UA

In the ibaPDA I/O manager, the general tab is accessed, and under the OPC UA server section, it is enabled. A new certificate is created.



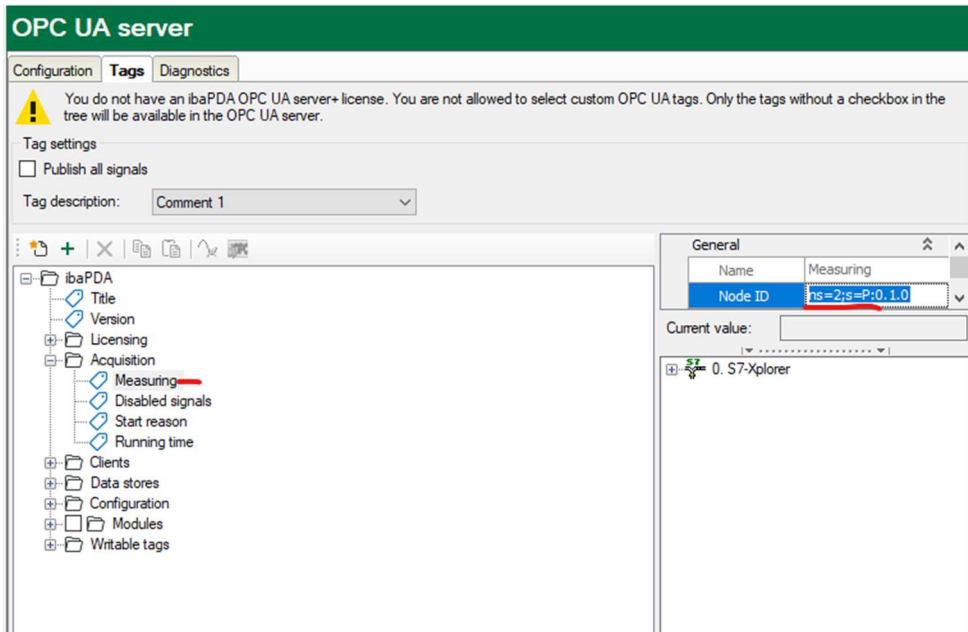
**Figure 35 OPC UA certificate**

A password is set for the certificate, and security policies are disabled while anonymous logon is enabled. The URI string is copied.



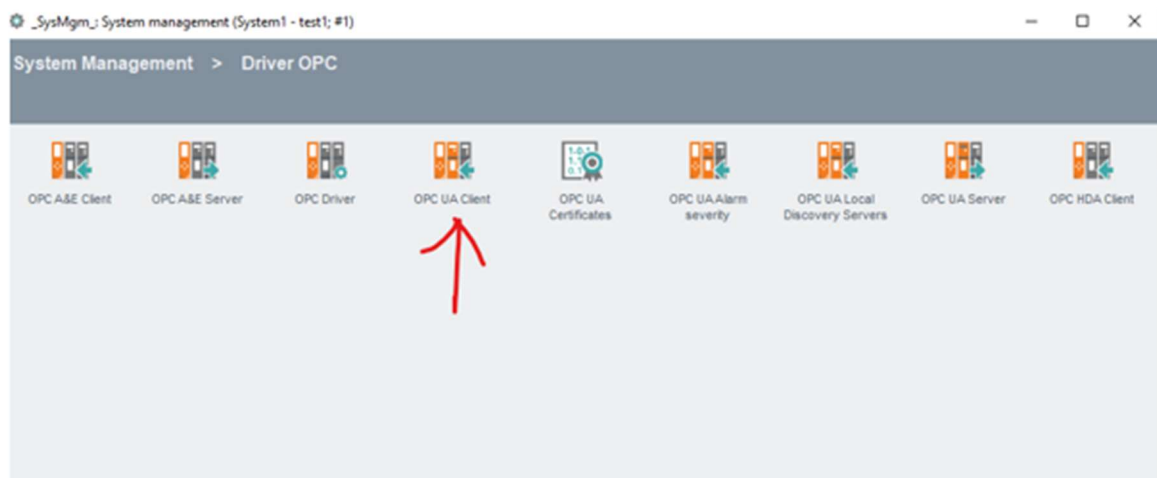
**Figure 36 ibaPDA OPC UA settings**

Next, in the tags section, the Measuring tag is found in the Acquisition table, and the node ID is noted.



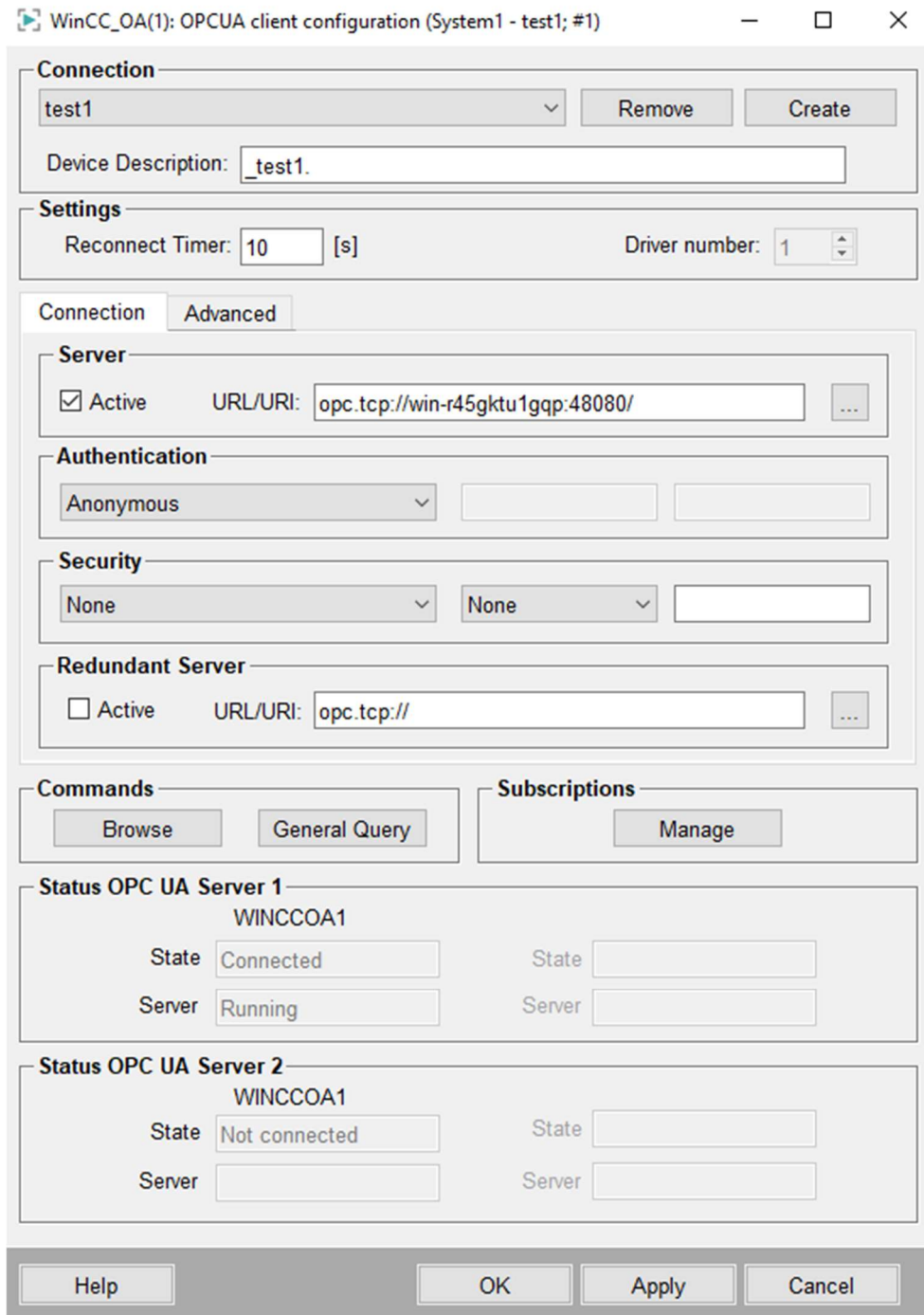
**Figure 37 OPC UA Measuring tag**

The WinCC OA server is started, and then in the system management, the driver OPC and OPC UA client are accessed.



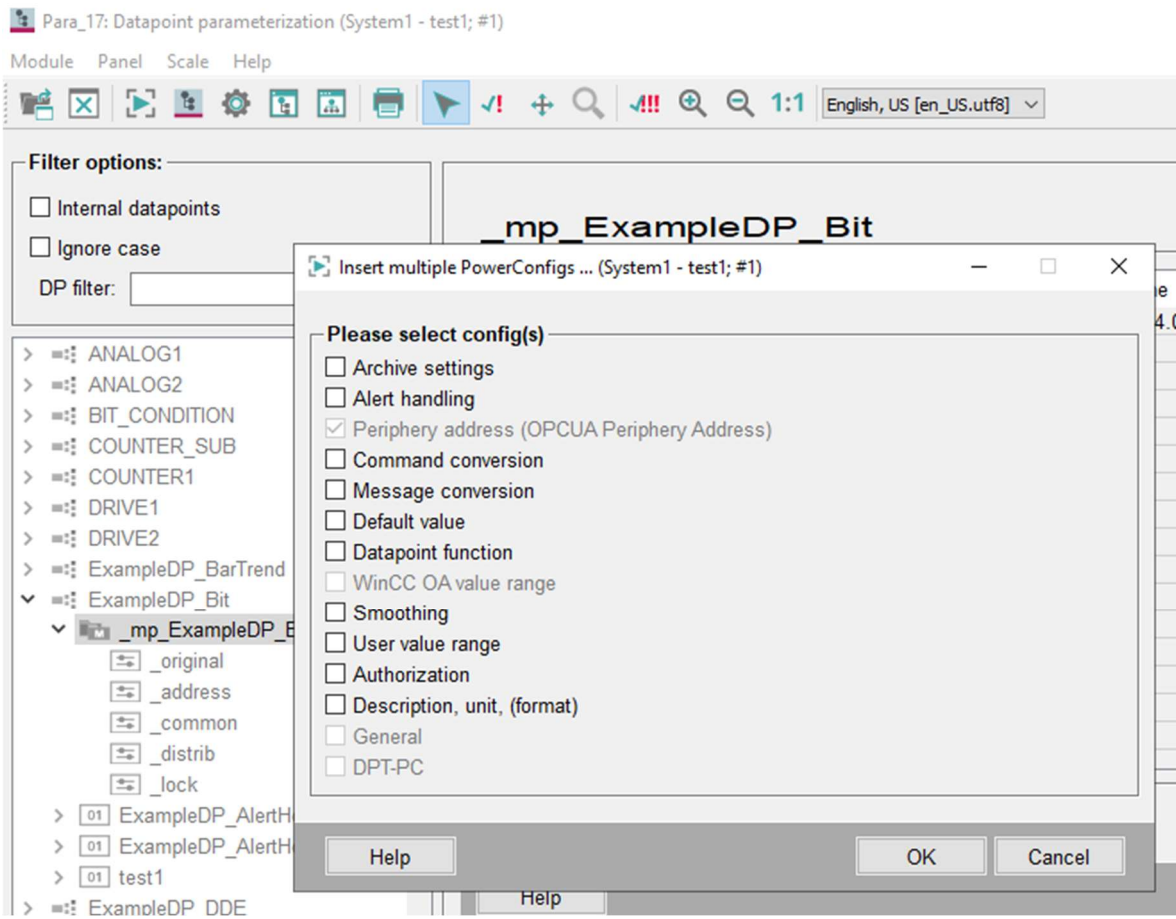
**Figure 38 OPC UA client**

A new client is added, and the correct information is entered.



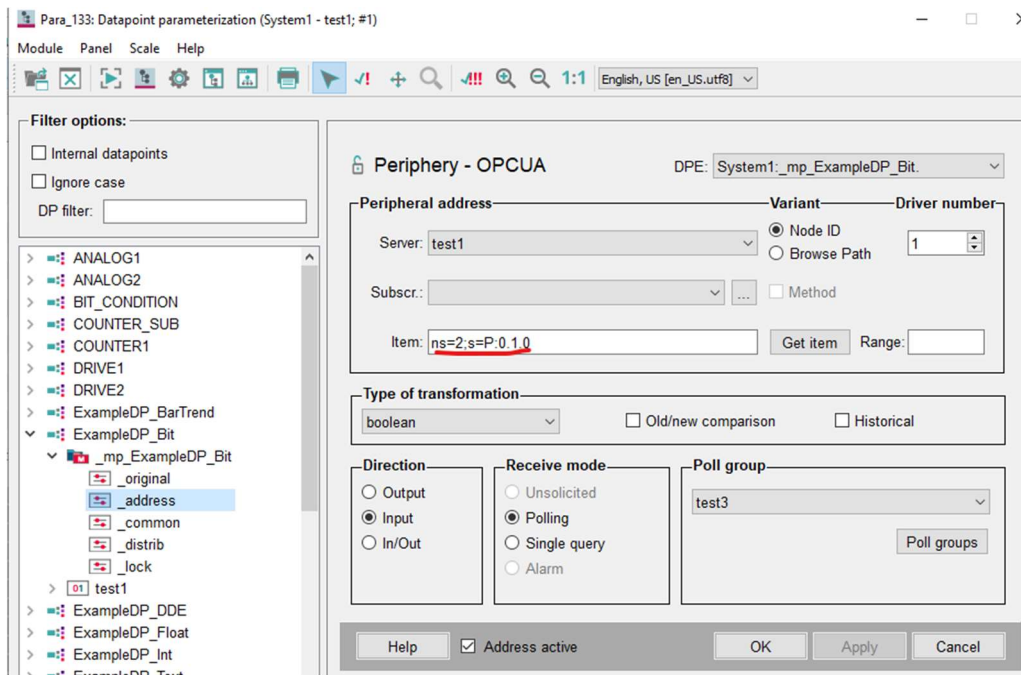
**Figure 39 OPC UA WinCC OA connection settings**

Here, the URI copied from the ibaPDA server is set, and authentication is configured as anonymous with security set to none. After pressing "apply," the connection is established, and the server is running. Next, in the Para section, a pre-made parameter is used, such as ExampleDP\_Bit. A new PowerConfig is added with a fixed periphery OPC UA periphery address, and the node ID from the Measuring tag from ibaPDA is copied.



**Figure 40 PowerConfig for OPC UA**

Then, in the `_address` section, the node ID from `ibaPDA` is set in the item section.



**Figure 41 Periphery address**

Then, a new datapoint is created in ExampleDP\_bit named test1, with the same address set as in the Figure 41 address. Next, in the screen created earlier, a new rectangle is made, and the initialize wizard is accessed with the same settings as in Figure 33 set to change the background color. When selecting the datapoint element, test1 is selected, and the colors are set so it appears white when it's 0 and green when the tag is green.

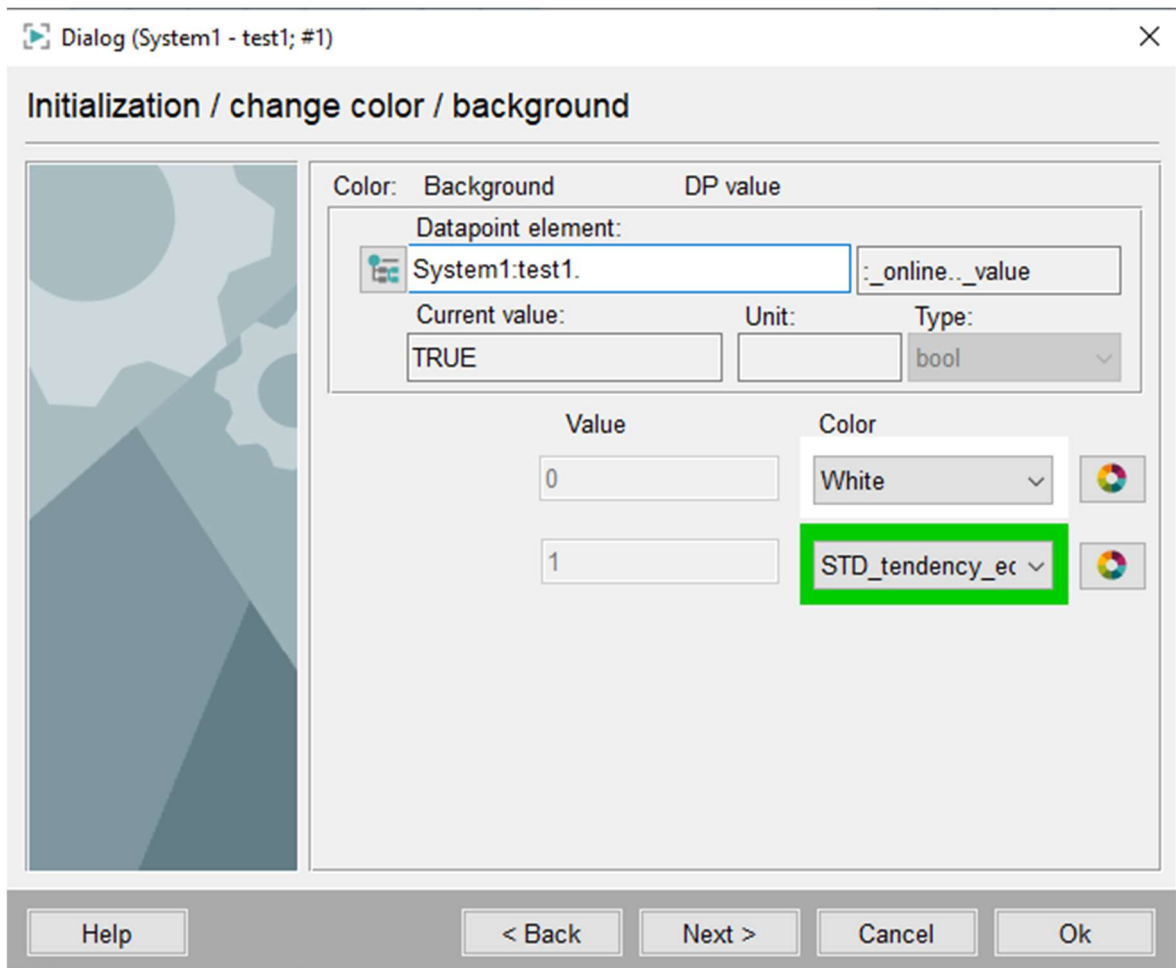


Figure 42 OPC UA rectangle wizard

Now, there is a screen with two rectangles, one with a connection to ibaPDA via OPC UA and the other with an SNMP connection. A text box is placed on top of each rectangle for identification, one indicating the connection to OPC UA and the other to SNMP.

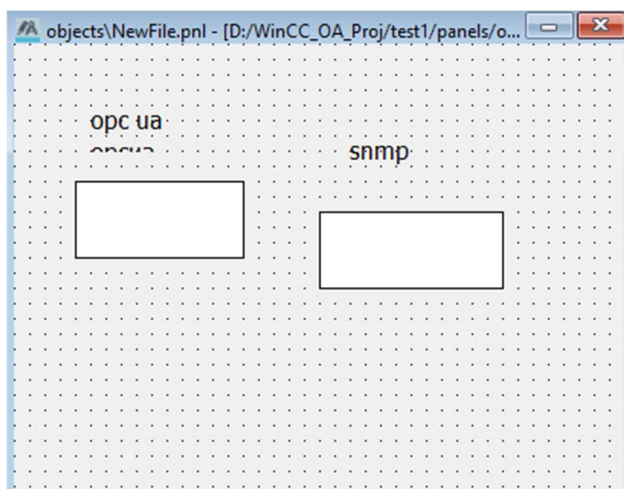
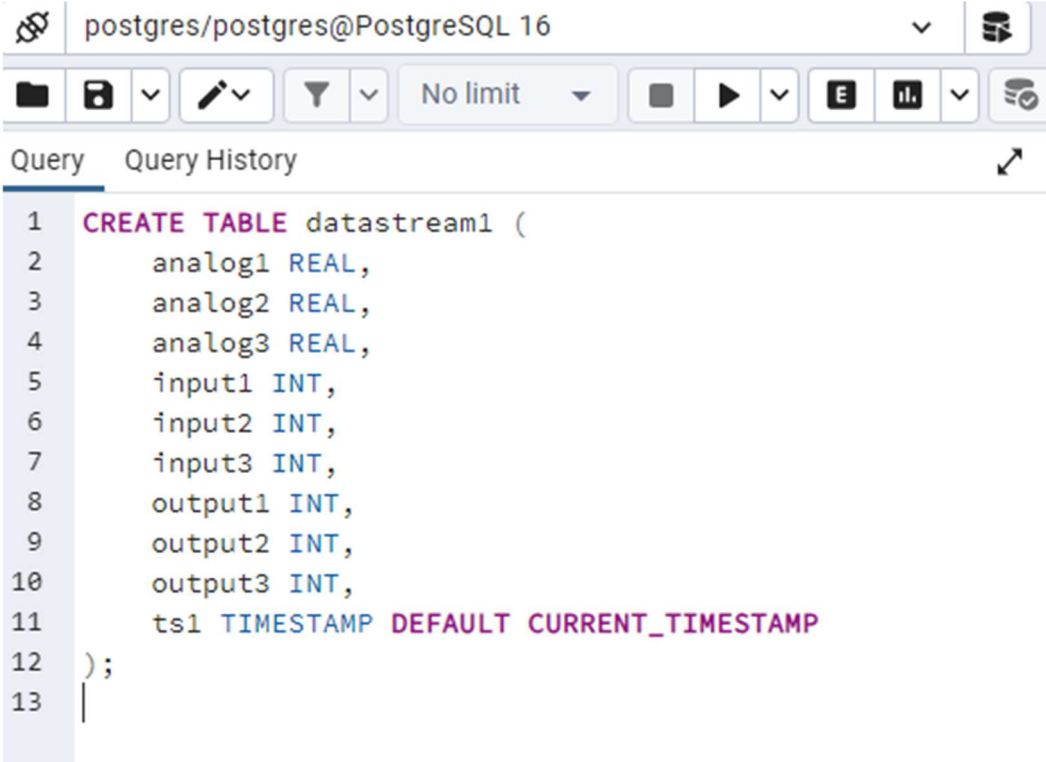


Figure 43 WinCC OA screen

### 3.5 PostgreSQL configuration

The database has already been created, and now a table needs to be created to stream data into it. For each signal to be stored, one column will be created, along with a timestamp column. With 9 signals in total, there will be 10 columns. The data types set will be "real" for the analog signals, "INT" for the digital signals, and "CURRENT\_TIMESTAMP" for the timestamp column. The query tool in PGAdmin will be utilized for this purpose.




```

1 CREATE TABLE datastream1 (
2     analog1 REAL,
3     analog2 REAL,
4     analog3 REAL,
5     input1 INT,
6     input2 INT,
7     input3 INT,
8     output1 INT,
9     output2 INT,
10    output3 INT,
11    ts1 TIMESTAMP DEFAULT CURRENT_TIMESTAMP
12 );
13 |

```

Figure 44 Database creation

The "CREATE TABLE" command was utilized to create a table named "datastream1" with 10 columns. Now, a value of 1 will be added in all columns except in the "ts1" column, where the current timestamp will be inserted.



```

1 INSERT INTO datastream1 (analog1, analog2, analog3, input1, input2, input3, output1, output2,
2 VALUES (1, 1, 1, 1, 1, 1, 1, 1, 1, CURRENT_TIMESTAMP);
3 |

```

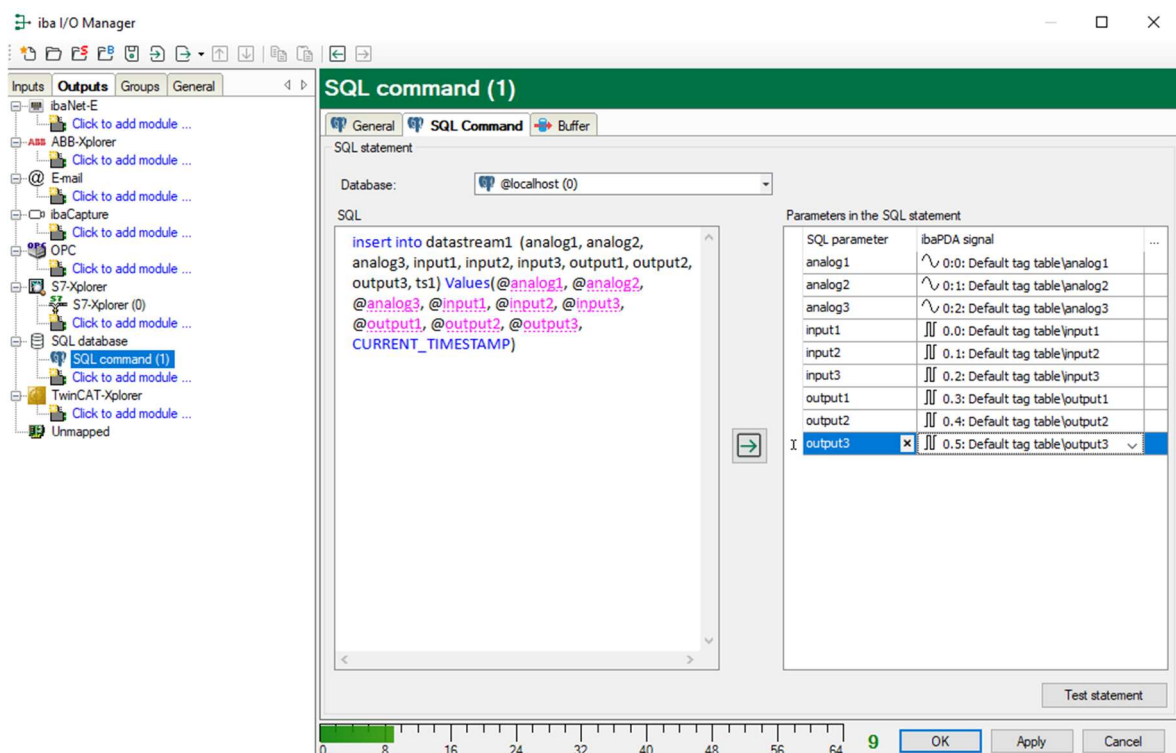
Figure 45 Data insert

The start data was set to indicate when the data was first inserted into the database.

Data Output											
	analog1 real	analog2 real	analog3 real	input1 integer	input2 integer	input3 integer	output1 integer	output2 integer	output3 integer	ts1 timestamp w	
1	1	1	1	1	1	1	1	1	1	2024-02-22 1	

**Figure 46 Data output in datastream1**

To stream data into the new table, ibaPDA is configured with an SQL statement. In the ibaPDA I/O manager, the SQL command is accessed, and the database to which a connection has been established is selected. The signals to be stored are then inserted using an SQL statement.



**Figure 47 Data stream SQL statement**

On the left tab, the SQL statement is located, while on the right tab, there is data linking between the PLC tags and the data being sent into the database. "Insert into" is used to send data to "datastream1" and link the columns with the PLC tags. The update time in the general

tab is changed to 10ms so that the SQL statement will run every 10ms, resulting in the current state of the signals being stored in the database every 10ms.

	analog1 real	analog2 real	analog3 real	input1 integer	input2 integer	input3 integer	output1 integer	output2 integer	output3 integer	ts1 timestamp
17	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
18	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
19	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
20	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
21	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
22	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
23	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
24	49.4	50.9	0	1	0	1	1	0	1	2024-02-2
25	49.4	50.9	0	1	0	1	1	0	1	2024-02-2

**Figure 48 Data from datastream1**

When the contents of "datastream1" are checked, it is observed that the same data as the PLC gives out is received, along with the correct timestamp.

### 3.6 Data deletion

A bat file was created with instructions to connect to "ibatest1" and delete data from "datastream1" where it is 2 weeks or older. Subsequently, a task was added to the task scheduler to ensure that the bat file runs once a day.

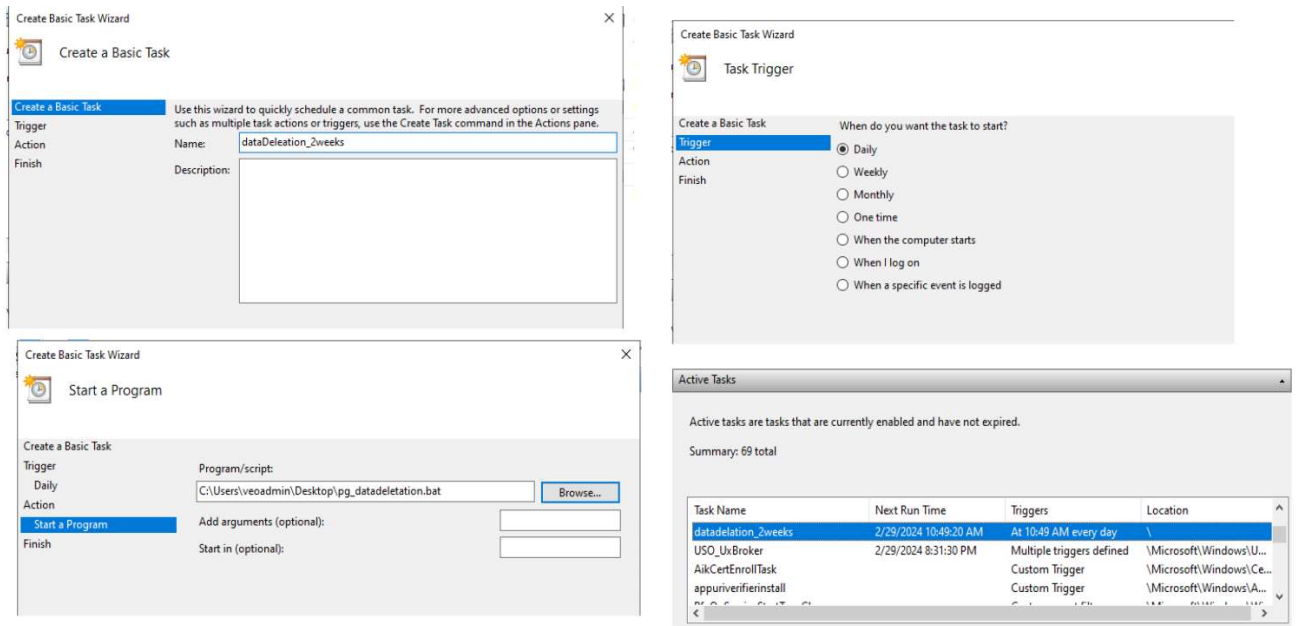
```

pg_data_delete - Notepad
File Edit Format View Help
@echo off
set PGPASSWORD=p
"C:\Program Files\PostgreSQL\16\pgAdmin 4\runtime\psql.exe" -U postgres -d ibatest1 -c "DELETE FROM datastream1 WHERE ts1 < NOW() - INTERVAL '2 weeks';"

```

**Figure 49 Data deletion**

The script connects to the database via psql, a program supplied with PGAdmin capable of executing SQL statements. Subsequently, "DELETE FROM" is utilized to remove all data older than 2 weeks.

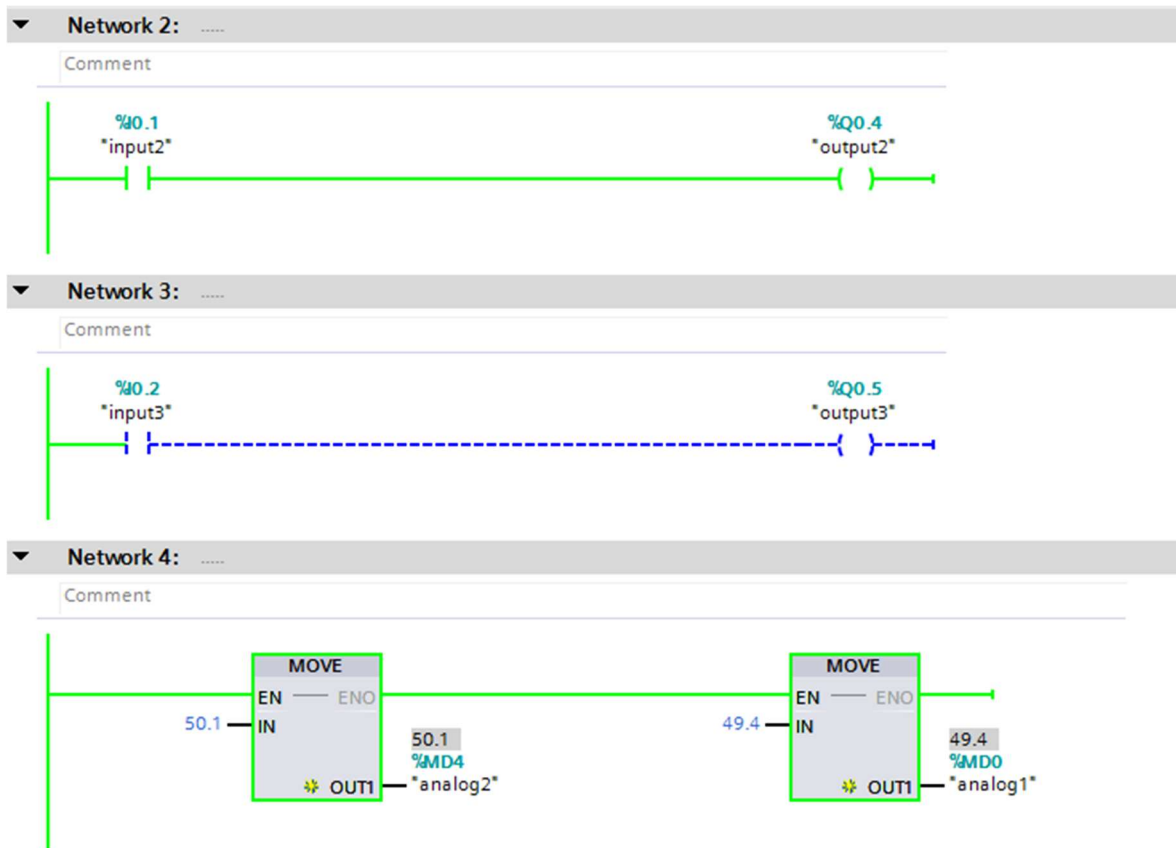


**Figure 50** Task scheduler setup

Here, the task scheduler is configured to run the bat file once every day. This ensures that all data older than 2 weeks is deleted.

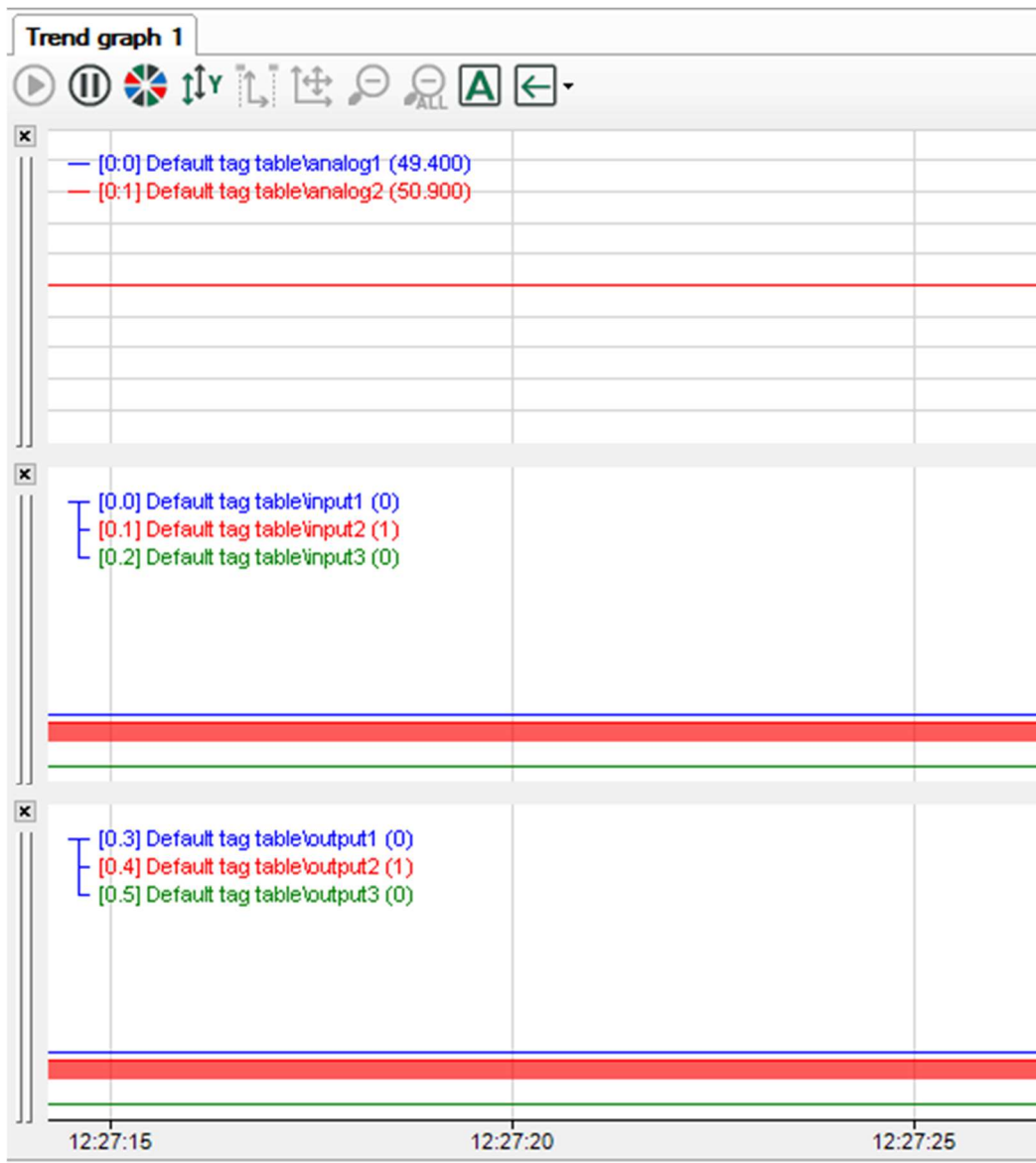
### 3.7 Testing

After everything was set up, the system was tested. The Siemens TIA virtual machine was accessed, and the PLC program previously created was downloaded to the PLC. Digital input 2 was modified to activate, resulting in digital input and output 1 and 3 being set to 0, while digital input and output 2 were set to 1. For the analog signals, 50.9 was input into analog2, and 49.4 was input into analog1.



**Figure 51** PLC program with online modification

Data acquisition in ibaPDA is now initiated. In the trend graphs previously created, the PLC values should be visible, with inputs and outputs 1 and 3 set to 0, input and output 2 set to 1, and the analog values set to 40.4 and 50.9.



**Figure 52 ibaPDA trend graph**

In the ibaPDA trend graph, it is observed that the same values are received as set in the PLC. When input 1 and output 1 are changed to 1, it is immediately noticed that ibaPDA updates the value for input and output 1.



**Figure 53 ibaPDA trend graph with updated PLC output**

It is now confirmed that data acquisition from the PLC to the ibaPDA server is functional. The next step involves checking if data streaming to the PostgreSQL database works. PGAdmin is accessed, and the ibatest1 database along with the datastream1 table is located. The query tool is utilized to display the 100 last rows of data in datastream1.

Query	Query History
1	<code>SELECT *</code>
2	<code>FROM datastream1</code>
3	<code>ORDER BY ts1 DESC</code>
4	<code>LIMIT 100;</code>
5	

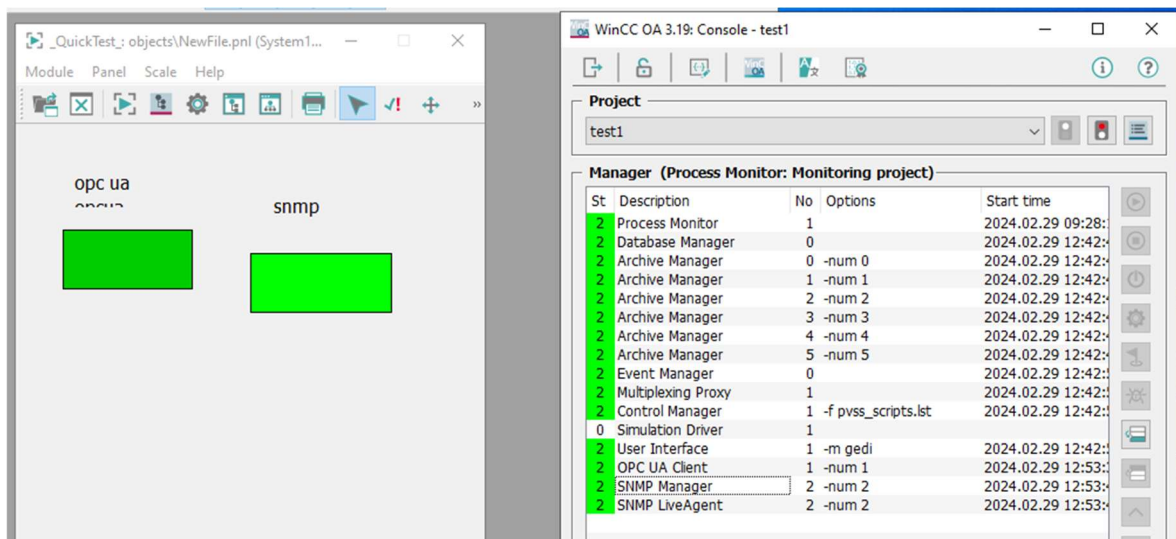
**Figure 54 Query to show 100 last rows**

	analog1 real	analog2 real	analog3 real	input1 integer	input2 integer	input3 integer	output1 integer	output2 integer	output3 integer	ts1 timestamp without time zone
1	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.216895
2	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.216579
3	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.215903
4	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.215548
5	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.214997
6	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.21441
7	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.165602
8	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.165276
9	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.16474
10	49.4	50.9	0	1	1	0	1	1	0	2024-02-29 12:39:08.1643

Total rows: 100 of 100    Query complete 00:00:00.844    Ln 5

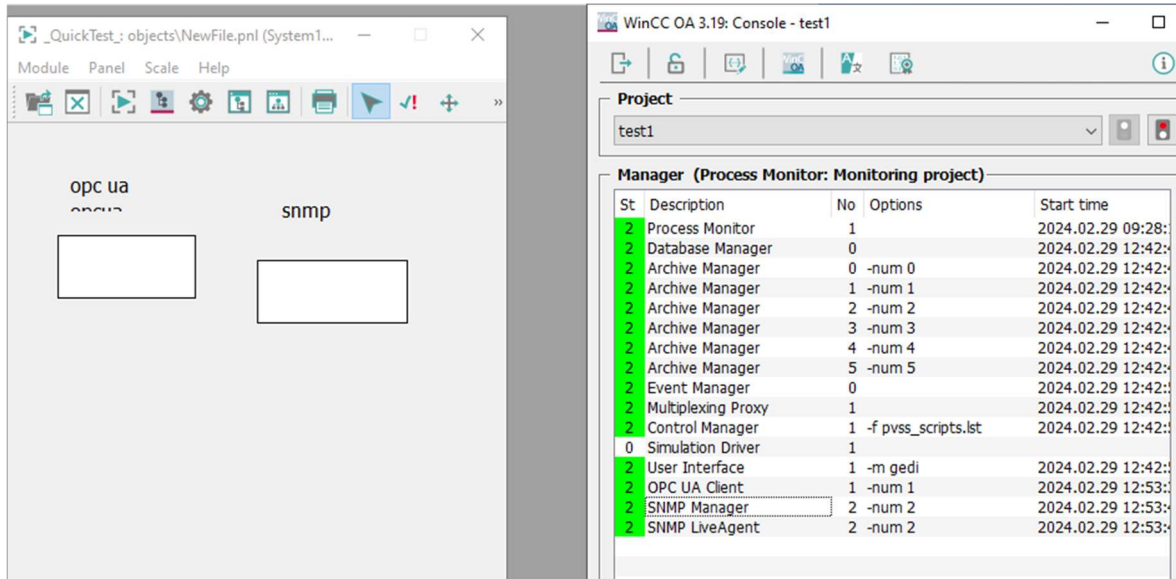
**Figure 55 datastream1 data**

Upon inspection, it is observed that the same values as those present in the PLC and ibaPDA, with correct timestamps, are displayed, affirming the functionality of data streaming. Subsequently, the functionality of WinCC OA in showing if data acquisition is active is checked. Accessing the WinCC OA virtual machine, the program and drivers are initiated. Within the panel created, the Quicktest module is utilized to simulate a SCADA program. If the rectangles function as expected, they should be green, indicating that data acquisition is active.



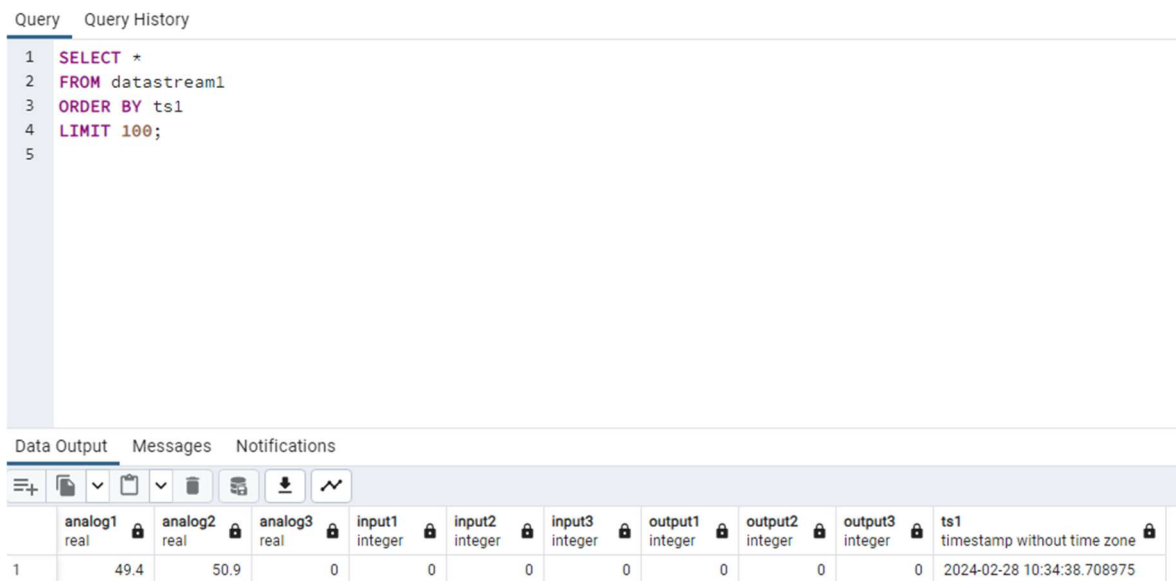
**Figure 56 WinCC OA Quicktest module online test**

It is observed that both the OPC UA and SNMP rectangles are green, indicating that both OPC UA and SNMP communication are operational, and data acquisition is active. Upon stopping the data acquisition, the rectangles turn white.



**Figure 57** Wincc OA with data acquisition stopped

All interconnected functions have been thoroughly checked—from PLC to ibaPDA, ibaPDA to PostgreSQL, and WinCC OA—and all are operational. Finally, the functionality of data deletion is verified. Initially, the first 100 rows of data in datastream1 are displayed using an SQL statement. It is noted that the first data was inserted 1 day ago. Therefore, the bat file is modified to delete all data older than 1 day.



**Figure 58** datastream1 100 first rows

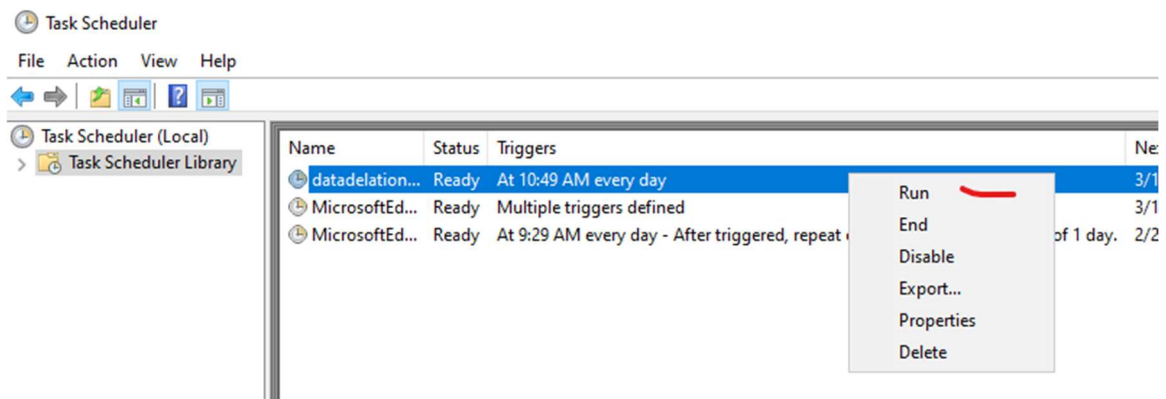
```

pg_data_delete - Notepad
File Edit Format View Help
@echo off
set PGPASSWORD=[redacted]
"C:\Program Files\PostgreSQL\16\pgAdmin 4\runtime\psql.exe" -U postgres -d ibatest1 -c "DELETE FROM datastream1 WHERE ts1 < NOW() - INTERVAL '1 day';"

```

**Figure 59 Modified bat file**

The task is run using task scheduler, and the SQL statement is executed again.



**Figure 60 Task scheduler**

```

1 SELECT *
2 FROM datastream1
3 ORDER BY ts1
4 LIMIT 100;
5

```

	analog1 real	analog2 real	analog3 real	input1 integer	input2 integer	input3 integer	output1 integer	output2 integer	output3 integer	ts1 timestamp without time zone
1	49.4	50.9	0	0	0	0	0	0	0	2024-02-28 13:21:43.671364

**Figure 61 SQL statement**

Upon running the statement again, it is observed that the data has been successfully deleted. All functions have now been tested and confirmed to be operational.

## 4 Discussion and result

The result of this thesis is a working solution for a data acquisition system to acquire and store signals from a PLC to a PostgreSQL database. The testing was made in a test environment with virtual machines and a physical switch and PLC. When looking at the efficiency and data integrity there was no data loss when testing and all signals were correctly stored in the database this shows that ibaPDA and PostgreSQL are a stable choice when selecting software to store and acquire data. Because ibaPDA can acquire signals from every 1ms it can accurately get high-frequency signals and store them. Windows Server 2022 run on VMware's virtualization platform is easy to set up and have a minimal downtime making it good when downtime is crucial. WinCC OA is a complex SCADA system that can handle and manage a whole system when making the connections from ibaPDA it was able to handle both connections from SNMP and OPC UA and worked as programmed this is crucial because the operator needs to know if data acquisition is lost it can start it again.

The challenges were understanding new programs and software and getting them to work together. The PLC program I made was quite simple but fulfilled its purpose to send signals to ibaPDA. ibaPDA is a program I had to learn to use and configure. It has good characteristics and is powerful when making a program it can send data to multiple systems while it is acquiring signals sending data to the database and WinCC OA with both SNMP and OPC UA. Getting SNMP and OPC UA to work in WinCC OA was a new experience as I have not used WinCC OA and getting it to work was a challenge as it is quite complex but this ensures that once the connections are made they are reliable. PostgreSQL was a new database program I had not used but it's working a lot like Microsoft SQL and the queries were the same making it easy to make and set up.

With my findings I think VEO can implement and use my solution when selling new projects.

Because buyers want to have data acquisition built in their projects to make historical data and easily find faults and patterns in their systems.

Further improvements would be to make the solution as a container in a docker environment to make it more scalable and for even faster deployment.

As it is now the system I made can handle large amounts of data as ibaPDA can acquire up to unlimited of signals only being hold back by the CPU in the server.

The server could also be pre-configured with routing for network connectivity. There could also be made a script to automatically create a SQL statement for all new signals to stream into the database.

Overall, I think the thesis was a success. I tested everything, and it worked and all requirements were met. The system can easily be used in a new project and minimal configuration is needed. The system is going to be used in upcoming projects which is a good sign that I made a good base system. I learnt a lot when making this thesis how to use multiple virtual machines and servers in a project and how a Scada system controls a system and mostly why data acquisition is important and how to acquire signals and smartly store them. I would recommend using ibaPDA and PostgreSQL when choosing to do a data acquisition system.

## 5 References

*About postgresQL*. (2023). Retrieved from postgresql:  
<https://www.postgresql.org/about/>

*Architecture*. (n.d.). Retrieved from SIMATIC WinCC Open Architecture Version 3.19 Documentation:  
[https://www.winccoa.com/documentation/WinCCOA/latest/en\\_US/GettingStarted/GettingStarted-06.html](https://www.winccoa.com/documentation/WinCCOA/latest/en_US/GettingStarted/GettingStarted-06.html)

*ibapda*. (n.d.). Retrieved from iba: <https://www.iba-ag.com/en/ibapda>

*investing in your success with Windows Server*. (2022). Retrieved from Microfost:  
<https://www.microsoft.com/en-us/windows-server#tabxd0e36b902361484d9513cd111ac2afcb>

*Polycase*. (2021, 9 10). Retrieved from What Is a Programmable Logic Controller (PLC)?:  
<https://www.polycase.com/techtalk/electronics-tips/what-is-a-programmable-logic-controller.html>

Scarpati, J. (n.d.). *Simple network management protocol*. Retrieved from techtarget:  
<https://www.techtarget.com/searchnetworking/definition/SNMP>

Smith, G. M. (2024, 6 5). *Data Acquisition (DAQ) - The Ultimate Guide [Updated 2024]*. Retrieved from DEWEsoft: <https://dewesoft.com/blog/what-is-data-acquisition>

*SNMP guide*. (n.d.). Retrieved from ManageEngine OpManager: <https://www.manageengine.com/se/network-monitoring/what-is-snmp.html>

*Task scheduler for developer*. (2023, 8 2). Retrieved from Microsoft learn: <https://learn.microsoft.com/en-us/windows/win32/taskschd/task-scheduler-start-page>

*Totally integrated automation portal*. (2023). Retrieved from Siemens: <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>

*Unified architecture*. (n.d.). Retrieved from OPC foundation: <https://opcfoundation.org/about/opc-technologies/opc-ua/>

*Veo*. (2023). Retrieved from Veo: <https://veo.fi/about-us/company/>

*Wmware*. (2024). Retrieved from WMware workstation pro: <https://www.vmware.com/nordics/products/workstation-pro.html>