RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electrotechnique

End-of-studies project dissertation

For the state engineer's diploma in "Electrotechnique"

Integration of a relay protection GemStart 4.3 into a PLC

HESSAINE Mohamed Akram & ROUABHIA Mohamed

Under the supervision of **Pr. BERKOUK El Madjid** (ENP)
Mr. HOUAR Adda (SCH)

Presented publicly on the day : (01/07/2024)

Composition of the jury:

President:	Pr.L.NEZLI	ENP
Advisor:	Pr. M.BERKOUK	ENP
Examiner:	Dr. R.BELKACEMI	ENP

ENP 2024

Département Electrotechnique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Algiers Algeria.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Electrotechnique

End-of-studies project dissertation

For the state engineer's diploma in "Electrotechnique"

Integration of a relay protection GemStart 4.3 into a PLC

HESSAINE Mohamed Akram & ROUABHIA Mohamed

Under the supervision of **Pr. BERKOUK El Madjid** (ENP)
Mr. HOUAR Adda (SCH)

Presented publicly on the day : (01/07/2024)

Composition of the jury:

President:	Pr.L.NEZLI	ENP
Advisor:	Pr. M.BERKOUK	ENP
Examiner:	Dr. R.BELKACEMI	ENP

ENP 2024

Département Electrotechnique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Algiers Algeria.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

سوناطراك
sonatrach

eni

Département d'Electrotechnique

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Électrotechnique

---

Integration du relais de protection GemStart 4.3 dans un API

HESSAINE Mohamed Akram & ROUABHIA Mohamed

Sous la direction de **Pr. BERKOUK El Madjid** (ENP)
Mr. HOUAR Adda (SCH)

Presenté et soutenu publiquement le : (01/07/2024)

Composition du jury:

Président:	Pr.L.NEZLI	ENP
Promoteur:	Pr. M.BERKOUK	ENP
Examinateur:	Dr. R.BELKACEMI	ENP

ENP 2024

Département Electrotechnique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Algiers Algeria.

ملخص

يمثل تقادم أجهزة التحكم جمستار 4 تحديات كبيرة لأنظمة الكهرباء بسبب توقف الإنتاج والدعم الفني المحدود. يعتبر استبدال هذه الأجهزة بنماذج أحدث مكلفًا. البديل هو دمج مرحلات الحماية الخاصة بها مباشرة في جهازوحدة التحكم المنطقية القابلة للبرمجة، مما يجعل النظام أبسط وأكثر مرونة. تُحسّن هذه الطريقة الإدارة، وتقلل من تكاليف الصيانة، وتطيل عمر النظام، مما يبرر الاستثمار الأولي.

الكلمات المفتاحية: حماية – محرك كهربائي – جمستار – وحدة التحكم المنطقية القابلة للبرمجة.

Résumé

L'obsolescence des contrôleurs Gemstart 4 pose des défis pour les systèmes électriques en raison de l'arrêt de leur production et du support technique limité. Remplacer ces contrôleurs par des modèles récents est coûteux. Une alternative est d'intégrer directement leurs relais de protection dans un PLC, simplifiant et rendant le système plus flexible. Cette approche améliore la gestion, réduit les coûts de maintenance, et prolonge la durée de vie du système, justifiant l'investissement initial.

Mots clés : Protection - Moteur électrique - Gemstart - API.

Abstract

The obsolescence of Gemstart 4 controllers poses challenges for electrical systems due to production stoppages and limited technical support. Replacing these controllers with newer models is costly. An alternative is to integrate their protection relays directly into a PLC, making the system simpler and more flexible. This approach improves management, reduces maintenance costs and extends the life of the system, justifying the initial investment.

Keywords : Protection - Electrical motor - Gemstart - PLC.

Thank you

Firstly, let us express our gratitude to Allah, the Almighty, for granting us the will, patience, and health throughout these long years of study.

We wish to express our gratitude towards our mentor, Prof. BERKOUK, for accepting to supervise us during the development of this work, as well as for his assistance and all his advice.

We also wish to thank Mr.HOUAR Adda, DCS Control engineer at JV Gas project In Salah, for entrusting us with this topic, for their patience, and for the invaluable assistance that he have always provided us until the completion of this work.

Our gratitude extends to all the electrical engineering professors at the National Polytechnic School, for guiding us throughout our university education to ensure quality learning and instilling the values of such a noble profession.

Finally, we would like to thank all those who have helped us, directly or indirectly, throughout our academic journey.

HESSAINE Mohamed Akram & ROUABHIA Mohamed.

Dedication

To our dearest parents,

No amount of words or expressions, however eloquent they may be, can adequately convey our gratitude, appreciation, and consideration for the sacrifices you have made for our education and well-being, especially our mothers.

To all the members of our families,for their encouragement and support.

To our close friends, for all the support that they have provided us throughout our lives.

To our dear classmates, for their encouragement, for their presence during difficult times.

To all those who, with a word, have given us the strength to continue.

HESSAINE Mohamed Akram & ROUABHIA Mohamed.

Contents

List of Figures

Acronym list:

- **CPF** : Central Processing Facility
- **PLC** : Programmable Ladder Controller
- **FLC** : Full Load Current
- **ALC** : Actuel Load Current
- **CPU** : Central Processing Unit
- **CT** : Current Transformer
- **LV** : Low Voltage
- **RTD** : Resistance Temperature Detector
- **I/O** : Input/Output
- **DI** :Digital input
- **AI** :Analogue Input
- **DQ** :Digital Output
- **DCS** : Distributed Control System
- **LED** : Light Emitting Diode
- **RTU** : Remote Terminal Unit
- **SCADA** : Supervisory Control And Data Acquisition
- **TIA** :Totally Integrated Automation
- **ISG** :In Salah Gas
- **HMI** :Human-Machine Interface
- **MICOM** :Multilin Integrated Control and Monitoring (a brand of protective relays)
- **S7-1200** :A series of PLCs from Siemens
- **GEM** :General Electric Multilin

General introduction

The issue of obsolescence concerning Gemstart 4 controllers in a substation poses a significant challenge for the management and maintenance of electrical systems. With the discontinuation of production and limited technical support for Gemstart 4, operators are faced with the difficulty of maintaining these aging devices in optimal working condition. While replacing Gemstart 4 entirely with newer models like Gemstart 5 is often considered a solution, the costs associated with such an upgrade can be prohibitive. This financial constraint presents a real dilemma as it hinders a swift transition to more advanced technologies, thereby jeopardizing the reliability and overall performance of the electrical system. Consequently, finding alternative strategies to extend the useful life of Gemstart 4 while planning a gradual transition to more modern solutions becomes imperative to ensure the sustainability and operational efficiency of the substation.

The elimination of the Gemstart interface and the direct integration of its protection relays into a PLC (Programmable Logic Controller) represents a promising solution to address the challenges related to equipment obsolescence and maintenance. By replacing the control and protection functions previously provided by the Gemstart with specific relay modules integrated into the PLC, the system is simplified and made more flexible and scalable. This approach would centralize the management of protection relays with the rest of the control system, providing better visibility and greater operational efficiency. Furthermore, by leveraging the advanced programming capabilities of the PLC, it becomes possible to implement customized protection and monitoring logic tailored to the specific needs of the substation. Although this process involves initial costs and development efforts, the long-term benefits in terms of system simplification, reduction of maintenance costs, and extension of useful life typically justify this transition to an integrated PLC solution.

In summary, the integration of relays and protection controllers into a PLC offers a comprehensive and intelligent solution for long term management of electrical equipments, providing significant economic operational and maintenance benefits. The solution is associated with a replacement of the existing RTU SCADA for operation and monitoring and provision of field electrical devices to PCS.

This dissertation is structured in 4 chapters as following:

Chapitre 1 : This chapter gives a general overview on the art of electrical motor's protections and how do they function plus the different existing protections in the sub-station.

Chapitre 2 : This chapter gives a general overview of the In Salah Gas project & the different types of relay protections of each electrical motor.

Chapitre 3 : This chapter introduces the Siemens software labeled "Tia Portal" and gives a rather general knowledge on how to use the software for the implementation of the solution.

Chapitre 4 : Application of the solution proposed & results backed with simulation .

Finally, we will conclude this dissertation with conclusions and a few perspectives.

Chapter 1

The art of electrical motor protection

1.1 Introduction

Chapter 1 introduces electrical motor protection, highlighting its importance for AC motors. It discusses key motor characteristics for protection and categorizes conditions requiring protection into external conditions and internal faults, setting the stage for detailed exploration of modern relay design and protection mechanisms.

1.2 Introduction to the relay protection

There are a wide range of AC motors and motor characteristics in existence, because of the numerous duties for which they are used.

All motors need protection, but fortunately, the more fundamental problems affecting the choice of protection are independent of the type of motor and the type of load to which it is connected.There are some important differences between the protection of induction motors and synchronous motors, as shown later in the paper.

Motor characteristics must be carefully considered when applying protection; while this may be regarded as stating the obvious, it is emphasised because it applies more to motors than to other items of power system plant. For example, the starting and stalling currents/times must be known when applying overload protection, and furthermore the thermal withstand of the machine under balanced and unbalanced loading must be clearly defined.

The conditions for which motor protection is required can be divided into two broad categories: imposed external conditions and internal faults.The following table provides details of most likely faults that require protection.

External Faults	Internal faults
Unbalanced supplies	Bearing failures
Undervoltages	Winding faults
Single phasing	Overloads
Reverse phase sequence	

1.3 Modern relay design

The design of a contemporary numerical motor protection relay must be sufficiently versatile to meet the protection requirements of the wide array of motor designs in operation, some of which may not tolerate any overloads.A relay that provides comprehensive protection will encompass the following features:

- Thermal protection
- Extended start protection
- Stalling protection
- Number of starts limitation
- Short circuit protection
- Earth fault protection
- Winding RTD measurment/trip
- Negative sequence current detection
- Undervoltage protection
- Loss of load protection
- Auxiliary supply supervision

In addition, relays may offer options such as circuit breaker condition monitoring as an aid to maintenance. Manufacturers may also offer relays that implement a reduced functionality to that given above where less comprehensive protection is warranted (e.g. induction motors of low rating).

The following sections examine each of the possible failure modes of a motor and discuss how protection may be applied to detect that mode.

1.4 Thermal overload protection

The majority of winding failures are either indirectly or directly caused by overloading (either prolonged or cyclic), operation on unbalanced supply voltage, or single phasing, which all lead through excessive heating to the deterioration of the winding insulation until an electrical fault occurs.

The generally accepted rule is that insulation life is halved for each 10ºC rise in temperature above the rated value, modified by the length of time spent at the higher temperature. As an electrical machine has a relatively large heat storage capacity, it follows that infrequent overloads of short duration may not adversely affect the machine. However, sustained overloads of only a few percent may result in premature ageing and insulation failure.

Furthermore, the thermal withstand capability of the motor is affected by heating in the winding prior to a fault. It is therefore important that the relay characteristic takes account of the extremes of zero and full-load pre-fault current known respectively as the 'Cold' and 'Hot' conditions.

The variety of motor designs, diverse applications, variety of possible abnormal operating conditions and resulting modes of failure result in a complex thermal relationship. A generic mathematical model that is accurate is therefore impossible to create. However, it is possible to develop an approximate model if it is assumed that the motor is a homogeneous body, creating and dissipating heat at a rate proportional to temperature rise. This is the principle behind the 'thermal replica' model of a motor used for overload protection.

The temperature T at any constant is given by:

$$T = T_{max}(1 - e^{-t/\tau}) \quad (1.1)$$

Where :

T_{max} = final steady state temperature
τ = Heating time constant
Temperature rise is proportional to the current squared:

$$T = KI_R^2(1 - e^{-t/\tau}) \quad (1.2)$$

where:

I_R current which, if flowing continuously, produces temperature max T_{max} in the motor Therefore, it can be shown that, for any overload current I , the permissible time t for this current to flow is:

$$t = \tau log_e[\frac{1}{1 - (\frac{I}{I_R^2})}] \quad (1.3)$$

In general, the supply to which a motor is connected may contain both positive and negative sequence components, and both components of current give rise to heating in the motor. Therefore, the thermal replica should take into account both of these components, a typical equation for the equivalent current being:

$$I_{eq} = \sqrt{(I_1^2 + kI_2^2)} \tag{1.4}$$

where:
I_1 = positive sequence current
I_2 = negative sequence current
K = negative sequence rotor resistance / positive sequence rotor resistance at rated speed.
A typical value of K is 3.
Finally, the thermal replica model needs to take into account the fact that the motor will tend to cool down during periods of light load, and the initial state of the motor. The motor will have a cooling time constant, τ_r , that defines the rate of cooling. Hence, the final thermal model can be expressed as:

$$t = \tau log_e \frac{(K^2 - A^2)}{(K^2 - 1)} \tag{1.5}$$

where:
τ = heating time constant
$K = \frac{I_q}{I_{th}}$
A^2 = Initial state of the motor(Cold or hot)
I_{th} = Thermal setting current
This equation takes into account the 'cold' and 'hot' characteristics defined in IEC 60255, part 8.

Some relays may use a dual curve characteristic for the heating time constant, and hence two values of the heating time constant are required. Switching between the two values takes place at a pre-defined motor current. This may be used to obtain better tripping performance during starting on motors that use a star-delta starter. During starting, the motor windings carry full line current, while in the 'run' condition, they carry only 57% of the current seen by the relay. Similarly, when the motor is disconnected from the supply,the heating time constant τ is set equal to the cooling time constant τ_r

Since the relay should ideally be matched to the protected motor and be capable of close sustained overload protection, a wide range of relay adjustment is desirable together with good accuracy and low thermal overshoot.
Typical relay setting curves are shown in the following figure:

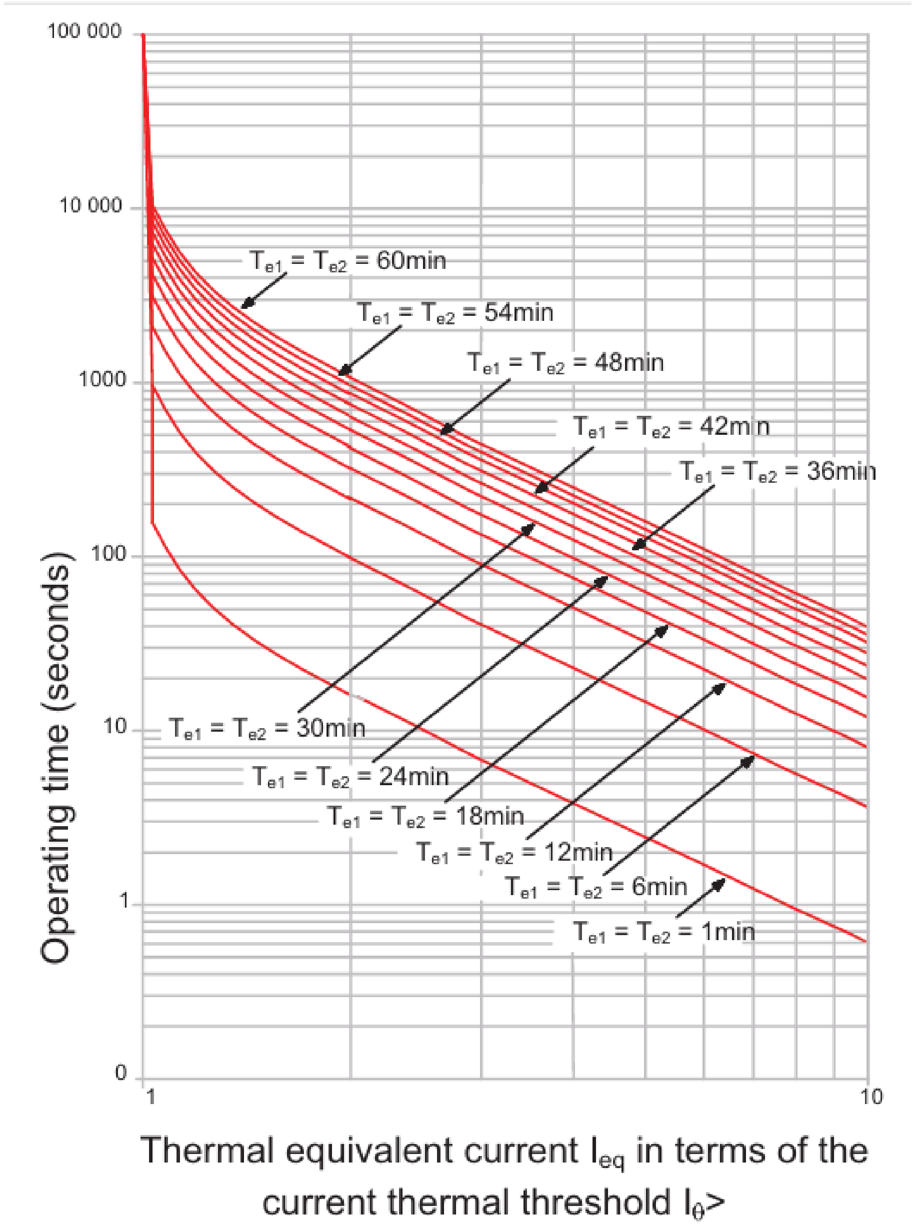


Figure 1.1: Thermal overload characteristic curves from cold – initial thermal state 0%

1.5 Start/Stall protection

When a motor is initiated, it initially pulls a current significantly higher than its rated load current as it ramps up to speed. Although the starting current diminishes as the motor gains speed, conventional protective measures usually treat the current as constant during this phase. The magnitude of the starting current varies based on motor design and starting method. For motors activated through direct-on-line (DOL) starting, the initial current may reach 4-8 times the full-load rating. However, with a star-delta starter, the line current will be equivalent to $\frac{1}{\sqrt{3}}$ of the DOL starting current.

In cases where a motor stalls during operation or fails to start due to excessive load, it will draw a current equivalent to its locked rotor current. Therefore, distinguishing between a stalled condition and a healthy start based solely on current draw is impractical. Differentiating between these states necessitates considering the duration of the current draw. Establishing protection for motors with starting times shorter than their safe stall times is straightforward.

However, when motors drive high inertia loads, the stall withstand time might be shorter than the starting time. In such scenarios, additional measures are required to discriminate between these conditions effectively.

1.5.1 Excessive Start Time/Locked Rotor Protection

A motor may fail to accelerate from rest for a number of reasons:

- Loss of supply phase
- Mechanical problems
- Low of supply voltage
- Excessive load torque

A large current will be drawn from the supply, and cause extremely high temperatures to be generated within the motor. This is made worse by the fact that the motor is not rotating, and hence no cooling due to rotation is available. Winding damage will occur very quickly – either to the stator or rotor windings depending on the thermal limitations of the particular design (motors are said to be stator or rotor limited in this respect). The method of protection varies depending on whether the starting time is less than or greater than the safe stall time. In both cases, initiation of the start may be sensed by detection of the closure of the switch in the motor feeder (contactor or CB) and optionally current rising above a starting current threshold value – typically 200% of motor rated current. For the case of both conditions being sensed, they may have to occur within a narrow aperture of time for a start to be recognised.

Special requirements may exist for certain types of motors installed in hazardous areas (e.g. motors with type of protection EEx 'e') and the setting of the relay must take these into account. Sometimes a permissive interlock for machine pressurisation (on EEx 'p' machines) may be required, and this can be conveniently achieved by use of a relay digital input and the in-built logic capabilities.

1.5.1.1 Start time < safe stall time

Protection is achieved by use of a definite time overcurrent characteristic, the current setting being greater than full load current but less than the starting current of the machine. The time setting should be a little longer than the start time, but less than the permitted safe starting time of the motor.The following figure illustrates the principle of operation for a successful start.

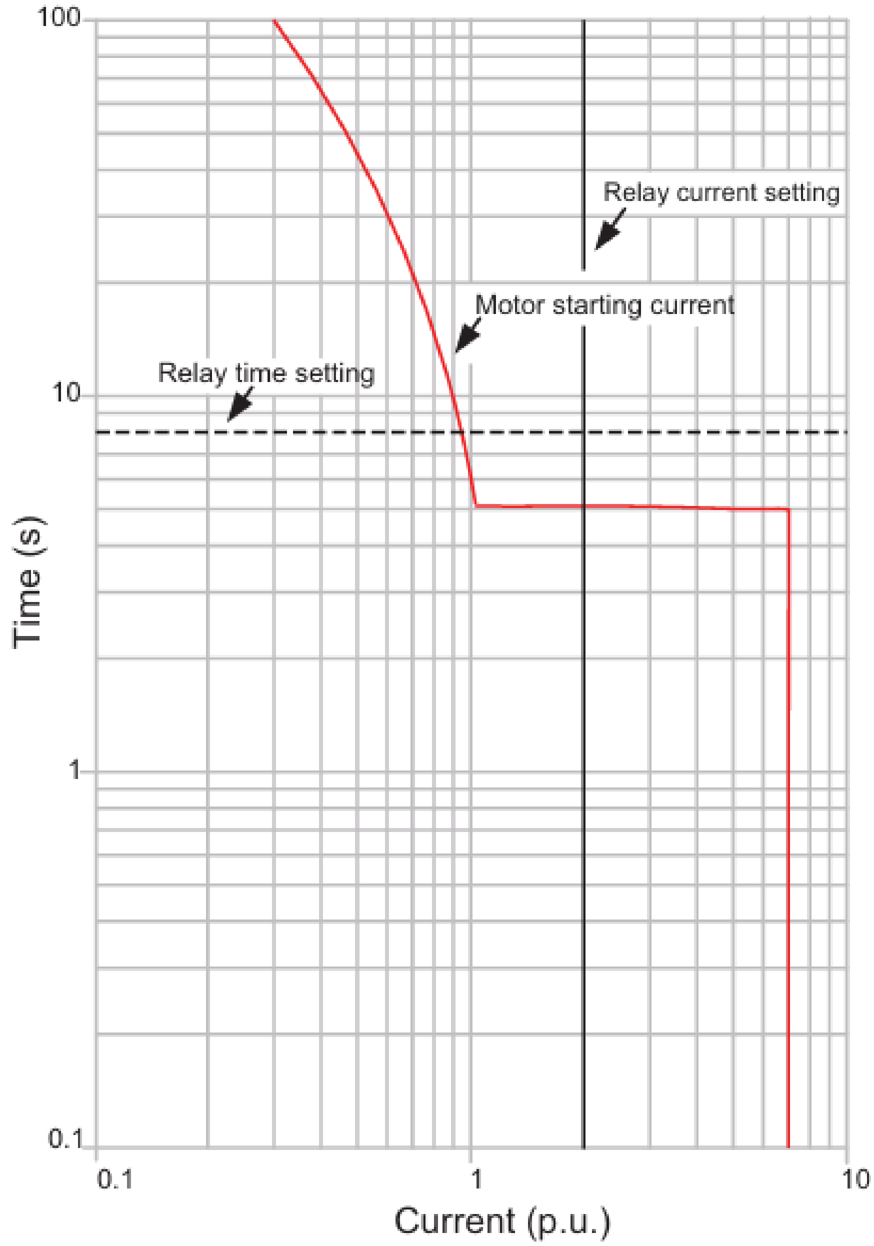


Figure 1.2: Relay setting for successful start: start time<stall time

1.5.1.2 Start time $\geq$ safe stall time

For this condition, a definite time overcurrent characteristic by itself is not sufficient, since the time delay required is longer than the maximum time that the motor can be allowed to carry starting current safely. An additional means of detection of rotor movement, indicating a safe start, is required. A speed-sensing switch usually provides this function. Detection of a successful start is used to select the relay timer used for the safe run-up time of the motor. This time can be longer than the safe stall time, as there is both a (small) decrease in current drawn by the motor during the start and the rotor fans begin to improve cooling of the machine as it accelerates. If a start is sensed by the relay through monitoring current and/or start device closure, but the speed switch does not operate, the relay element uses the safe stall time setting to trip the motor before damage can occur.The figure 1.3(a) illustrates the principle of operation for a successful start, and Figure 1.3(b) for an unsuccessful start.

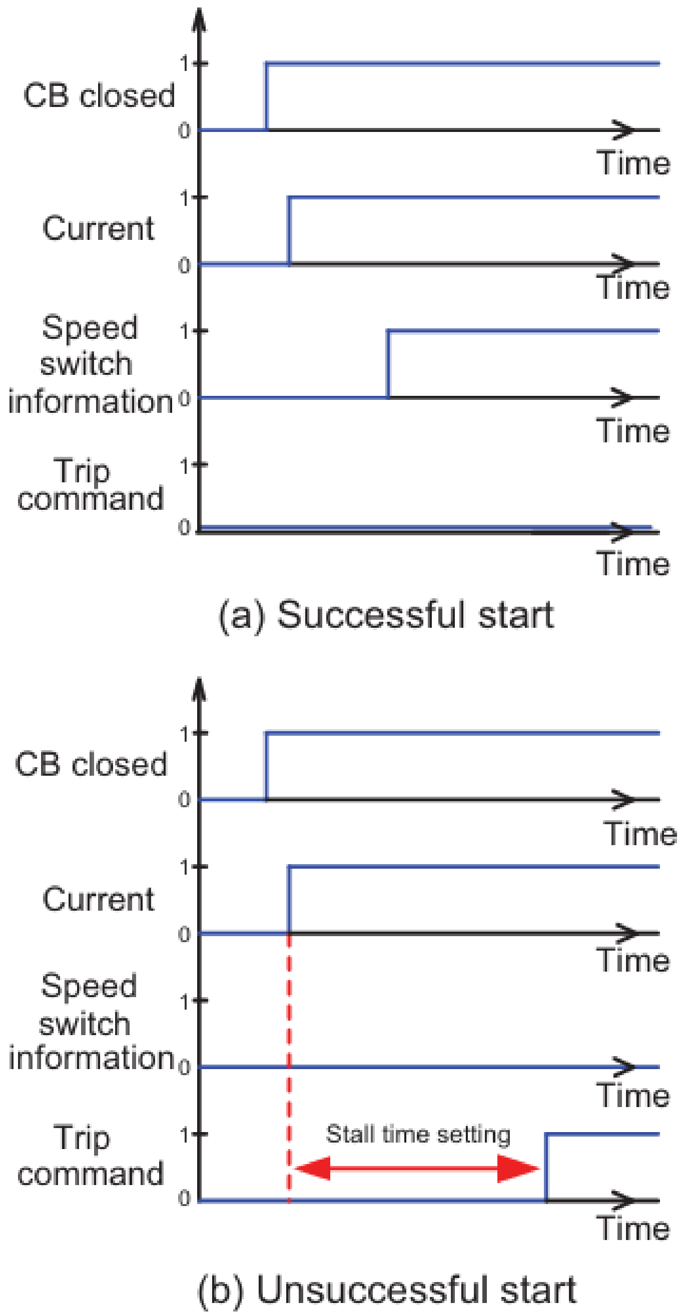


Figure 1.3: Relay settings for start time> stall time

1.6 Stall protection

If a motor stalls during operation or fails to start due to excessive load, it will draw a current from the power source equivalent to the locked rotor current. To prevent damage, it's crucial to disconnect the motor promptly when this situation occurs.

Detecting motor stalls involves monitoring the motor current. After a successful start, if the motor current exceeds the starting current threshold within the safe start time and then rises again above the starting current threshold, it indicates a stall condition. Tripping occurs if this condition persists beyond the stall timer setting, facilitated by an instantaneous overcurrent relay element.

In many systems, transient voltage loss (lasting typically up to 2 seconds) doesn't trigger tripping for designated motors. They're allowed to re-accelerate upon voltage restoration, drawing a current similar to the starting current for several seconds, surpassing the motor stall relay element threshold. To prevent unintended tripping during re-acceleration, a motor protection relay detects voltage dips and inhibits stall protection for a defined period after voltage recovery using the undervoltage protection element. This ensures protection against stalled motors even if re-acceleration fails. The time delay setting depends on the re-acceleration scheme and motor characteristics, determined through a transient stability study.

1.7 Number of starts limitation

Every motor has a limit on the number of starts it can undergo within a specific timeframe to prevent exceeding permissible winding temperatures. If the allowed number of starts is surpassed, starting should be prevented. This becomes intricate because the permitted count of "hot" starts within a given period is typically fewer than "cold" starts due to varying initial motor temperatures. Therefore, the relay must track separate counts for both "cold" and "hot" starts. Utilizing data from the motor's thermal replica enables distinguishing between hot and cold starts.

To allow the motor to cool down between starts, a time delay can be set between consecutive starts, with differentiation again between hot and cold starts. The start inhibition is lifted after a duration determined by the motor's specifications. The overall protection function is illustrated in Figure 1.4.

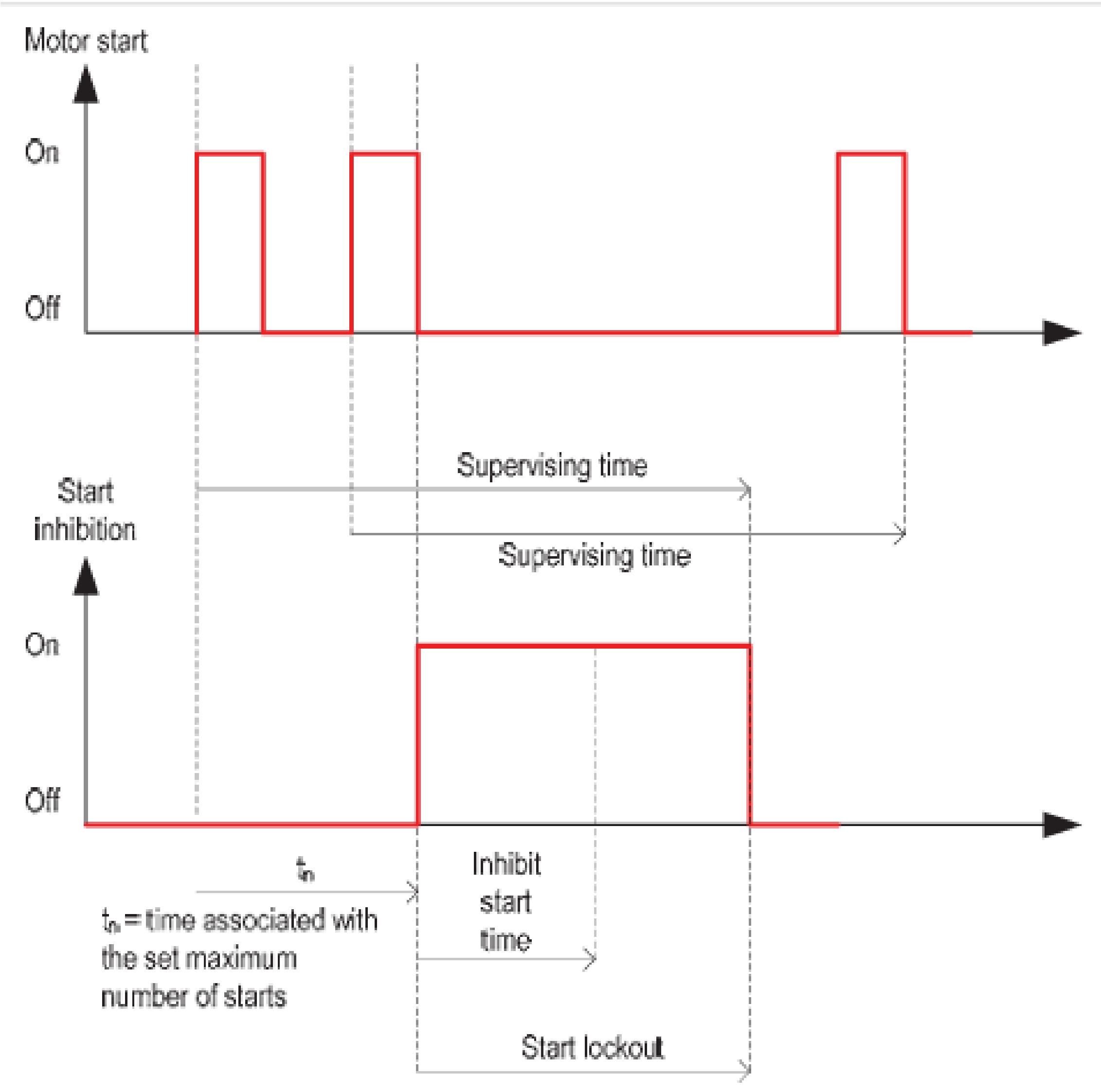


Figure 1.4: Number of starts limitation

In this example, the maximum number of starts within the Supervising Time has been reached, therefore the Inhibit Start Time is initiated. The remaining time is greater than the Inhibit Start Time, so the start inhibition remains for a duration equal to the supervising time minus the t_n.

1.8 Short circuit protection

Motor short-circuit protection is typically installed to address significant stator winding faults and terminal flashovers. Since faults between phases are rare due to the substantial insulation between phase windings, the fault usually involves earth quickly, triggering instantaneous earth fault protection as the stator windings are entirely enclosed in grounded metal.

For this purpose, a single definite time overcurrent relay element is sufficient, typically set at around 125% of the motor's starting current. A time delay is necessary to prevent false operation caused by current transformer spill currents, usually set at 100ms. However, if the motor is supplied through a fused contactor, coordination with the fuse is necessary, often requiring a longer time delay for the relay element.

As the goal of the protection is swift fault clearance to minimize damage, it's typically only installed on motors supplied via circuit breakers.

On larger high-voltage motors supplied via circuit breakers, differential (unit) protection might be employed to safeguard against phase-phase and phase-earth faults, especially in resistance-earthed power systems. This differential protection can be highly sensitive, enabling early fault detection to mitigate motor damage. In such cases, normal definite time overcurrent protection may not be sensitive enough, and sensitive earth fault protection might not be available.

However, the user might want to avoid the complex calculations necessary for setting sensitive non-directional earth fault overcurrent protection correctly on high-voltage systems, or there may be no provision for a voltage transformer (VT) to apply directional sensitive earth fault protection. Despite this, there's a minimum limit to the setting due to CT saturation spill currents during starting, and on some motors, neutral current flow during starting could trigger the differential protection.

In cases where sufficient sensitivity can be ensured, non-directional earth fault overcurrent protection is typically more cost-effective.

1.9 Earth fault protection

One of the most frequent issues encountered in motors is a fault in the stator winding. Regardless of the fault's initial form (such as phase-phase) or its cause (like cyclic overheating), the presence of the motor's metallic frame and casing ensures that it swiftly evolves into a fault involving earth. Therefore, implementing earth fault protection is of utmost importance.

The type and sensitivity of protection deployed largely hinge on the system's earthing configuration, with different types being addressed sequentially. Nevertheless, it's customary to incorporate both instantaneous and time-delayed relay elements to address both major faults and those that develop slowly over time.

1.9.1 Solidly-Earthed System

Most LV systems fall into this category, for reasons of personnel safety. Two types of earth fault protection are commonly found – depending on the sensitivity required.

For applications where a sensitivity of $> 20\%$ of motor continuous rated current is acceptable, conventional earth fault protection using the residual CT connection of Figure 1.5 can be used. A lower limit is imposed on the setting by possible load unbalance and/or (for HV systems) system capacitive currents.

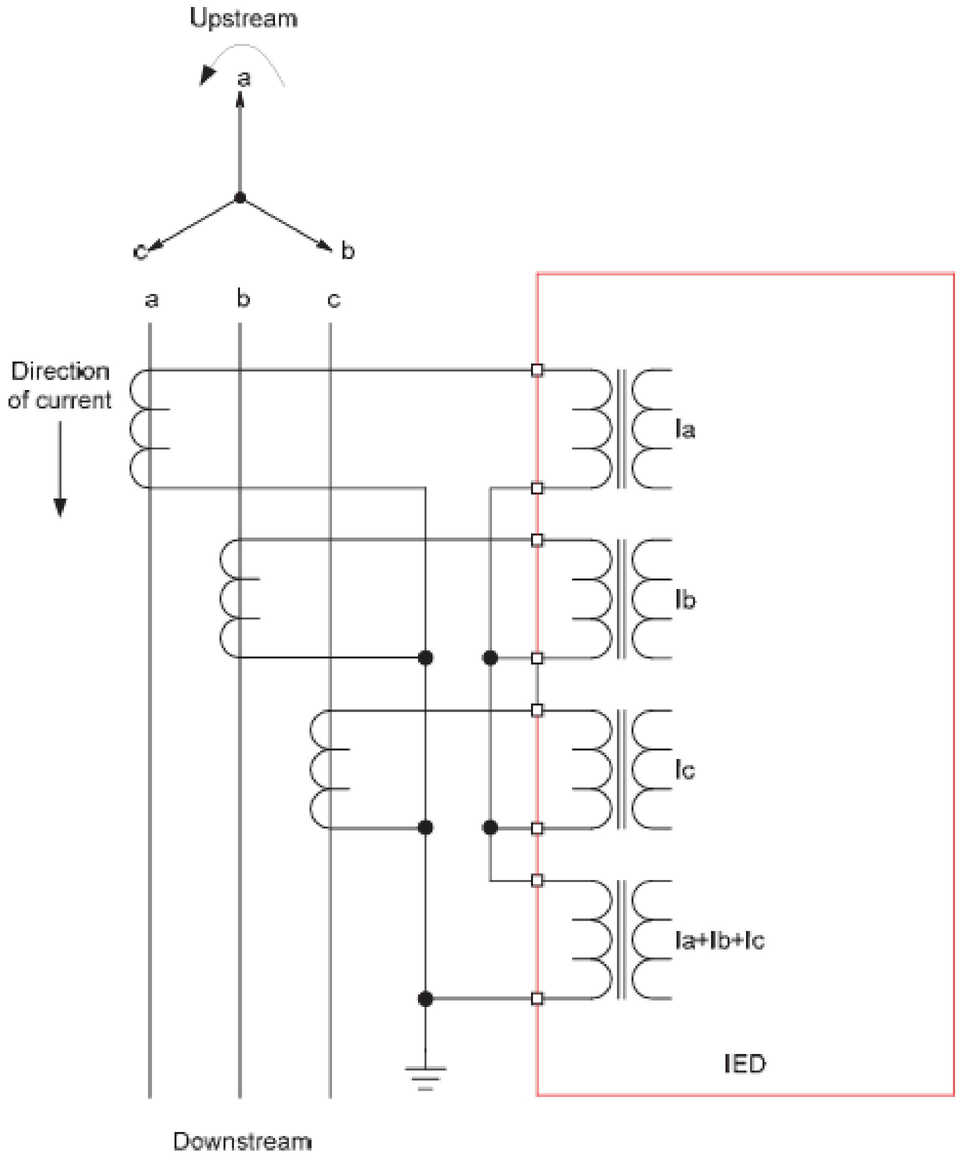


Figure 1.5: Residual CT connection for earth fault protection

Care must be taken to ensure that the relay does not operate from the spill current resulting from unequal CT saturation during motor starting, where the high currents involved will almost certainly saturate the motor CT's. It is common to use a stabilising resistor in series with the relay, with the value being calculated using the formula:

$$R_{stab} = \frac{I_{st}}{I_0}(R_{ct} + kR_l + R_r) \tag{1.6}$$

Where:
I_{st} = Starting current referred to CT secondary
I_0 = Relay earth fault setting (A)
R_{stab} = Stabilising resistor value (ohms)
R_{ct} = DC resistance of CT secondary (ohms)
R_l = CT single lead resistance (ohms)
R_r = Relay resistance (ohms)
k = CT connection factor (1 for star point at CT, 2 for star point at relay).

The stabilizing resistor serves to boost the relay's effective setting, thereby prolonging the tripping process. Normally, when employing a stabilizing resistor, the tripping response should be immediate.

An alternative approach, instead of utilizing a stabilizing resistor, involves employing a definite time delay characteristic. Determining the appropriate time delay typically involves trial and error to ensure it's long enough to prevent malfunctions during motor startups yet short enough to offer adequate protection in fault scenarios. Additionally, coordination with other devices, such as a fused contactor supplying power to a motor, is crucial. The contactor alone may not handle fault currents beyond a certain threshold, relying on the fuse for protection. Consequently, precautions must be taken to ensure the relay doesn't command the contactor to open before the fuse has time to act. Diagram 1.6(a) depicts improper grading, with the relay activating before the fuse for fault currents exceeding the contactor's capacity, whereas Diagram 1.6(b) demonstrates proper grading. Achieving this may necessitate intentionally introducing a definite time delay in the relay.

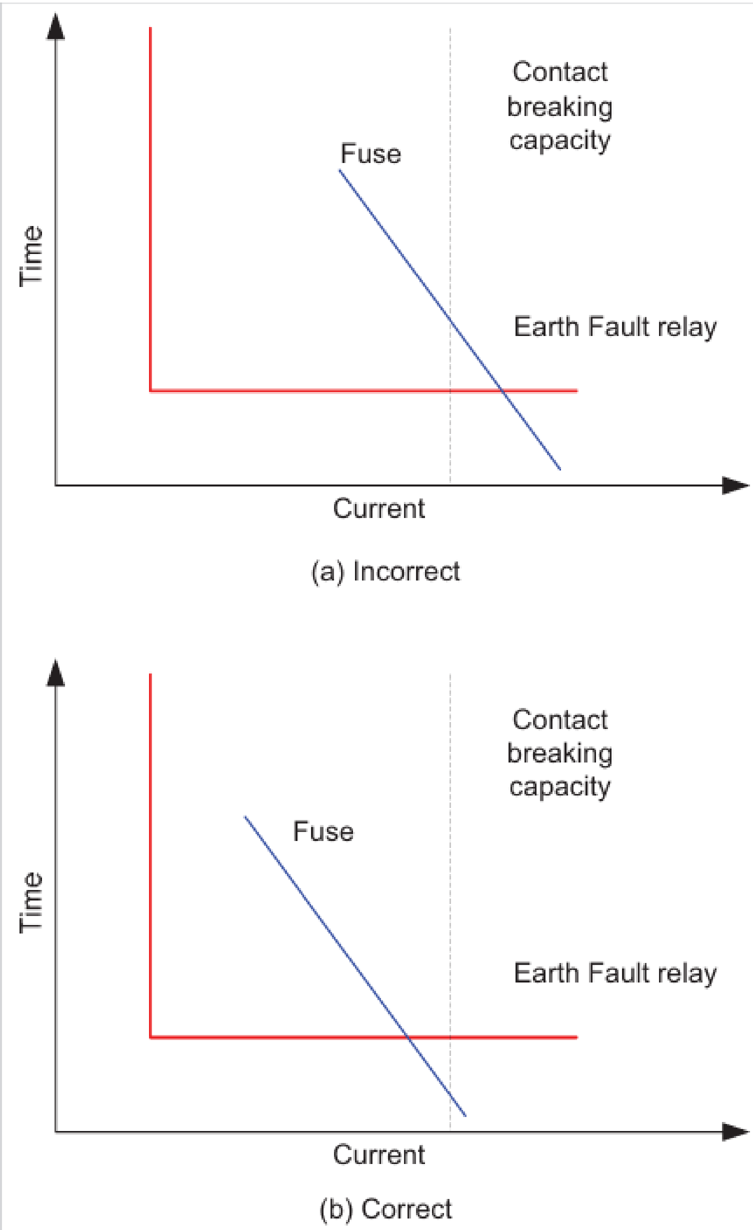


Figure 1.6: Grading of relay with fused contactor

If a more sensitive relay setting is required, it is necessary to use a core-balance CT (CBCT). This is a ring type CT, through which all phases of the supply to the motor are passed, plus the neutral on a four-wire system. The turns ratio of the CT is no longer related to the normal line current expected to flow, so can be chosen to optimise the pick-up current required. Magnetising current requirements are also reduced, with only a single CT core to be magnetised instead of three, thus enabling low settings to be used. Figure 19.7 illustrates the application of a core-balance CT, including the routing of the cable sheath to ensure correct operation in case of core-sheath cable faults.

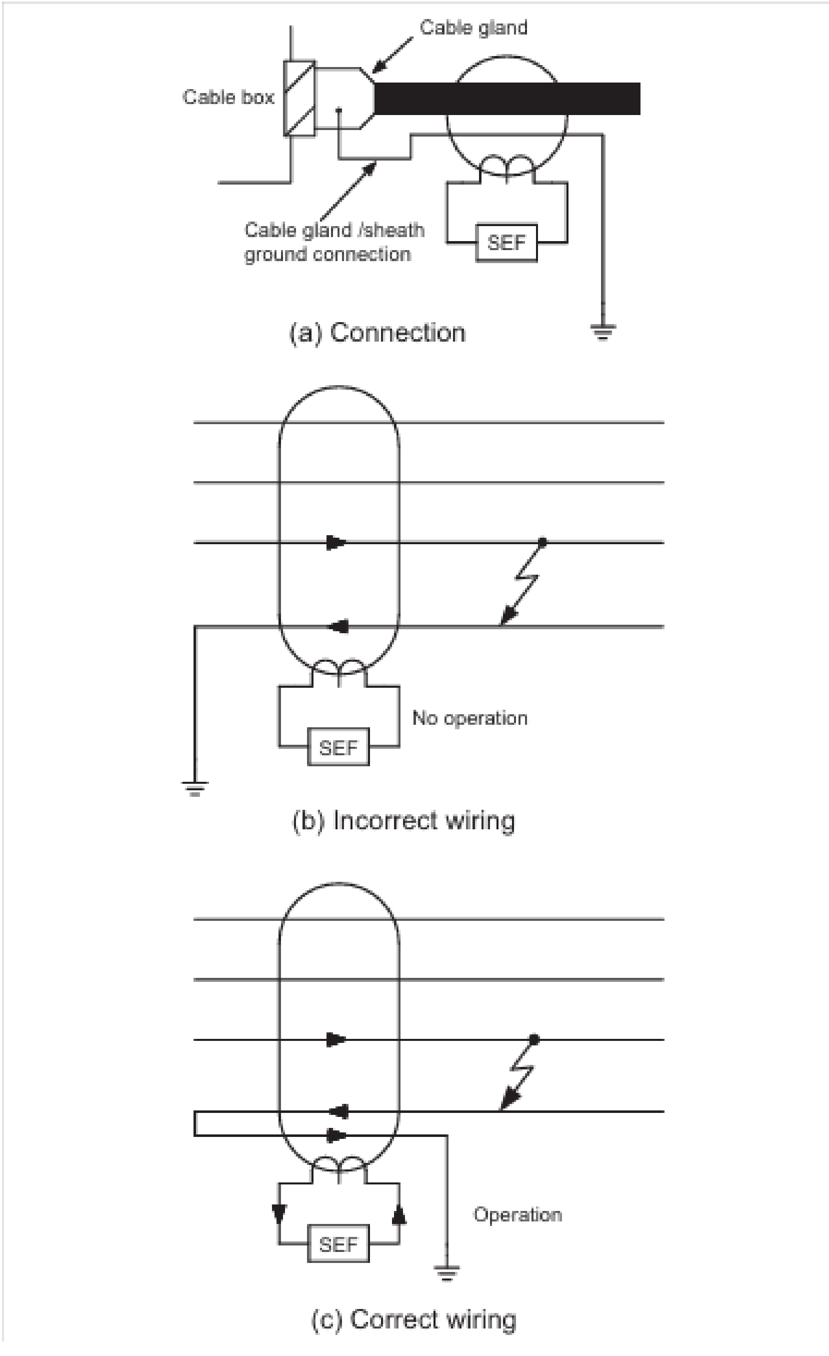


Figure 1.7: Application of core-balance CT

1.9.2 Resistance-Earthed Systems

These are commonly found on HV systems, where the intention is to limit damage caused by earth faults through limiting the earth-fault current that can flow. Two methods of resistance earthing are commonly used:

1.9.2.1 Low resistance earthing

In this method, the value of resistance is chosen to limit the fault current to a few hundred amps – values of 200A-400A being typical. With a residual connection of line CT's, the minimum sensitivity possible is about 10% of CT rated primary current, due to the possibility of CT saturation during starting. For a core-balance CT, the sensitivity that is possible using a simple non-directional earth fault relay element is limited to three times the steady-state charging current of the feeder. The setting should not be greater than about 30% of the minimum earth fault current expected. Other than this, the considerations in respect of settings and time delays are as for solidly earthed systems.

1.9.2.2 High resistance rating

In some HV systems, high resistance earthing is used to limit the earth fault current to a few amps. In this case, the system capacitive charging current will normally prevent conventional sensitive earth fault protection being applied, as the magnitude of the charging current will be comparable with the earth fault current in the event of a fault. The solution is to use a sensitive directional earth fault relay. A core balance CT is used in conjunction with a VT measuring the residual voltage of the system, with a relay characteristic angle setting of +45ºC. The VT must be suitable for the relay and therefore the relay manufacturer should be consulted over suitable types – some relays require that the VT must be able to carry residual flux and this rules out use of a 3-limb, 3 phase VT. A setting of 125% of the single phase capacitive charging current for the whole system is possible using this method. The time delay used is not critical but must be fast enough to disconnect equipment rapidly in the event of a second earth fault occurring immediately after the first. Minimal damage is caused by the first fault, but the second effectively removes the current limiting resistance from the fault path leading to very large fault currents.

An alternative technique using residual voltage detection is also possible, and is described in the next section.

1.9.3 Insulated earth systems

Earth fault detection presents problems on these systems since no earth fault current flows for a single earth fault. However, detection is still essential as overvoltages occur on sound phases and it is necessary to locate and clear the fault before a second occurs. Two methods are possible:

- detection of the resulting unbalance in system charging currents

- residual overvoltage.

1.9.3.1 System charging current unbalance

Sensitive earth fault protection using a core-balance CT is required for this scheme. The principle is the same as already detailed, except that the voltage is phase shifted by +90ºC instead of -90ºC. To illustrate this, Figure 1.8 shows the current distribution in an Insulated system subjected to a C phase to earth fault and Figure 1.9 the relay vector diagram for this condition. The residual current detected by the relay is the sum of the charging currents flowing in the healthy part of the system plus the healthy phase charging currents on the faulted feeder – i.e. three times the per phase charging current of the healthy part of the system. A relay setting of 30% of this value can be used to provide protection without the risk of a trip due to healthy system capacitive charging currents. As there is no earth fault current, it is also possible to set the relay at site after deliberately applying earth faults at various parts of the system and measuring the resulting residual currents.

If it is possible to set the relay to a value between the charging current on the feeder being protected and the charging current for the rest of the system, the directional facility is not required and the VT can be dispensed with.

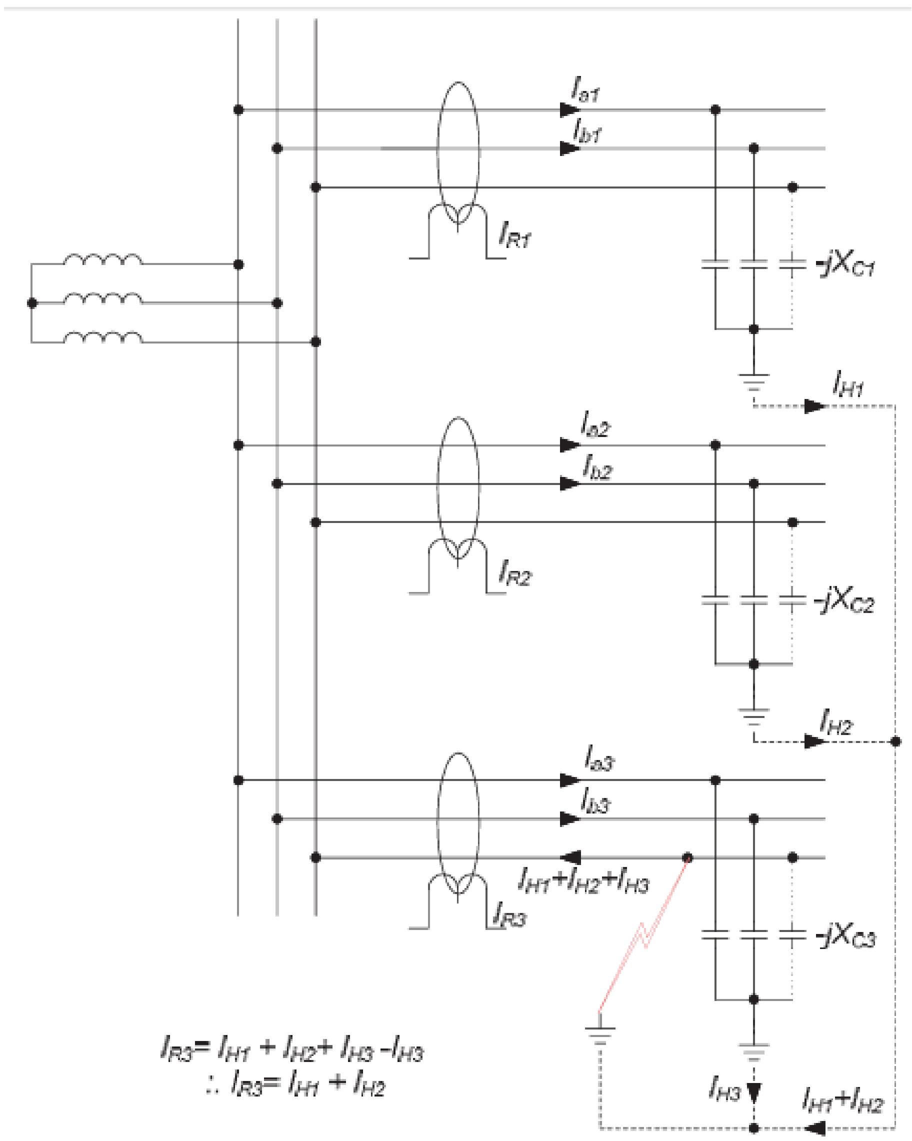


Figure 1.8: Current distribution in insulated-earth system for phase earth fault

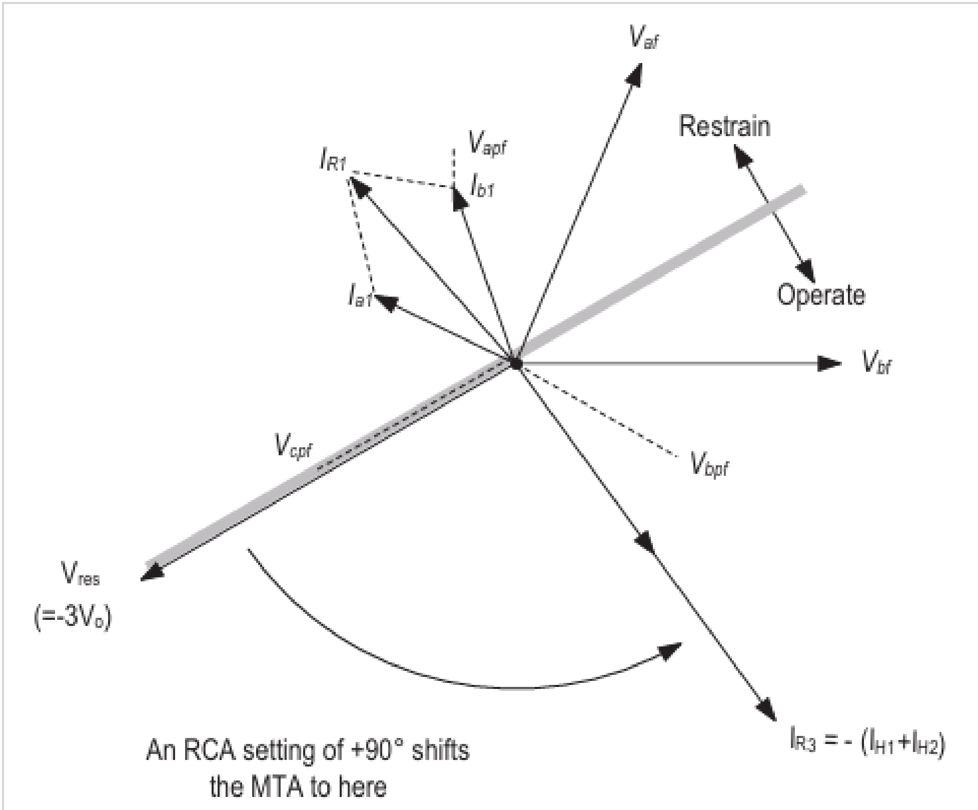


Figure 1.9: Relay vector diagram

1.9.3.2 Residual voltage method

A lone earth fault triggers an increase in voltage between the system's neutral and ground, a discrepancy that can be sensed by a relay monitoring the residual voltage of the system. Normally, this voltage remains at zero for a balanced and healthy system. This method eliminates the need for current transformers (CTs) and proves beneficial in situations where installing numerous core-balance CTs is impractical due to physical limitations or cost concerns. However, the voltage transformers (VTs) utilized must be appropriate for the task, meaning three-limb, three-phase VTs are unsuitable. Typically, the relay includes configurable alarm and trip settings, each featuring adjustable time delays. Determining the setting voltage requires understanding the system's earthing arrangement and impedances, exemplified in Figure 1.10 for a resistance-earthed system.

Careful grading of the relays is essential because the residual voltage will be detected by all relays within the affected section of the system. Grading is conducted with this in mind, often prioritizing time-based alarms for the initial stage, followed by a high-set definite time trip for backup protection.

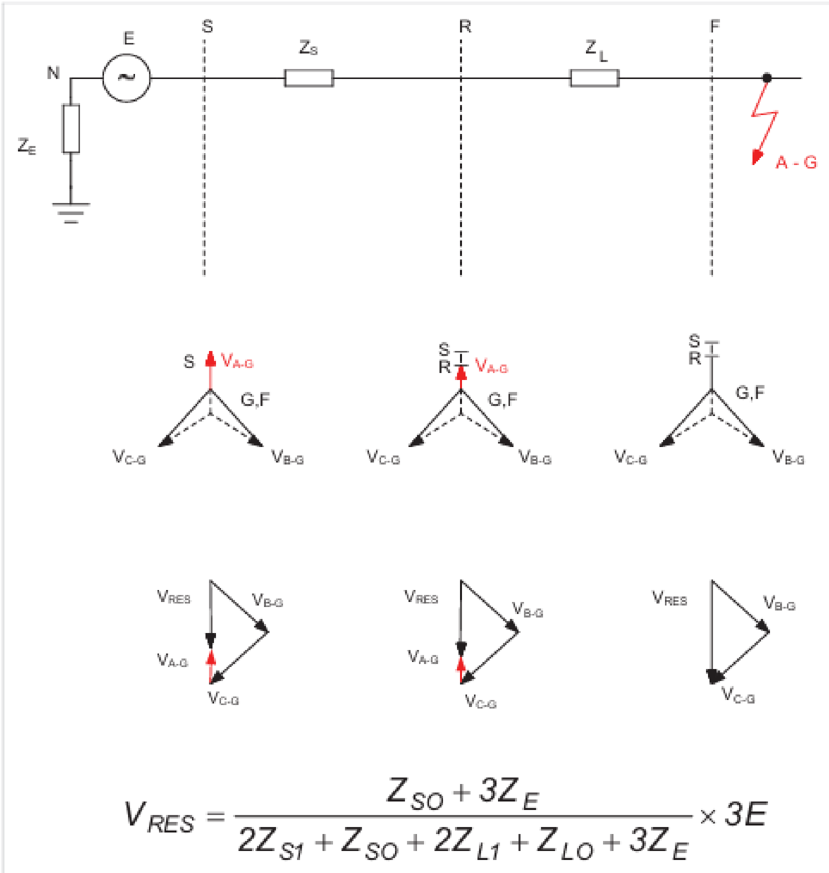


Figure 1.10: Residual voltage earth-fault protection for resistance earthed system

1.10 Negative phase sequence protection

Any imbalance in voltage conditions, like uneven loading, single-phase loss, or faults, leads to the generation of negative phase sequence current. Typically, single-phase faults trigger earth-fault protection, but unless it's sensitive, it might not detect motor winding faults.

The magnitude of negative sequence current varies based on voltage imbalance and the ratio of negative to positive sequence impedance. While various factors contribute to imbalance, determining negative sequence impedance is relatively straightforward, especially in the classical induction motor equivalent circuit, as shown in Figure 1.11, where magnetizing impedance is disregarded.
Motor positive sequence impedance at slip s:

$$= \sqrt{[(\frac{R_{1p} + R_{2p}}{2 - s})^2 + (X_{1p} + X_{2'_p})^2]} \tag{1.7}$$

The motor negative sequence impedance at slip s :

$$\sqrt{[(R_{1n} + \frac{R_{2'_n}}{S})^2 + (X_{1n} + X_{2'_n})^2]} \tag{1.8}$$

Where: suffix p indicates positive sequence quantities & suffix n indicates negative sequence quantities

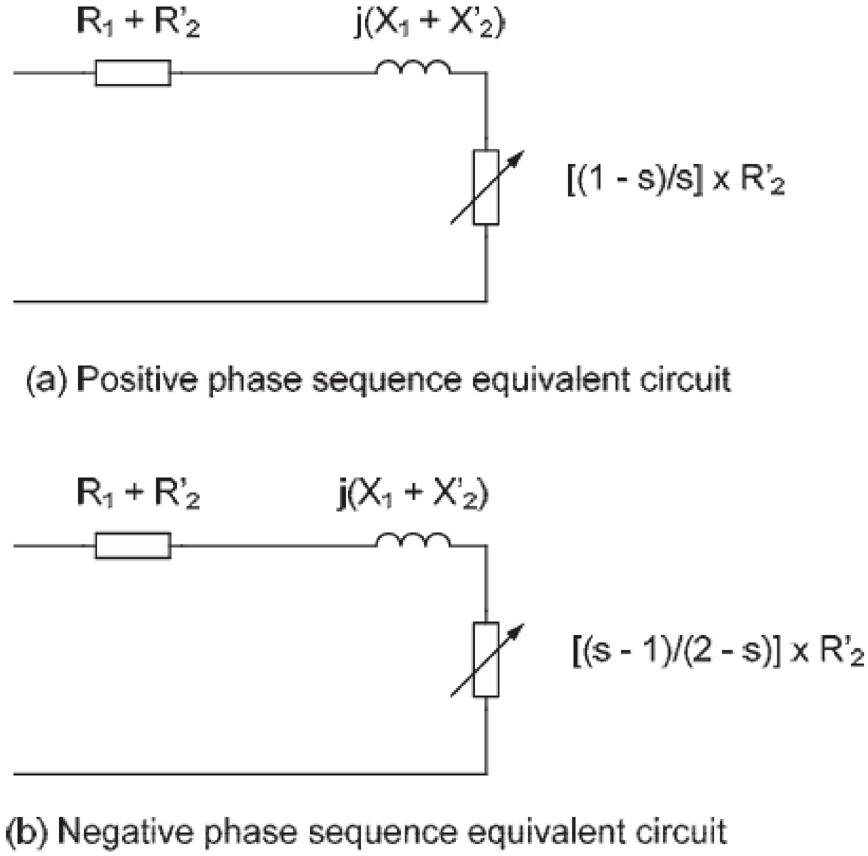


Figure 1.11: Induction motor equivalent circuit

The resistance can be neglected as it is small compared with the reactance. Thus the negative sequence reactance at running speed is approximately equal to the positive sequence reactance at standstill. An alternative more meaningful way of expressing this is:

$$\frac{Positiveseq.impedence}{Negativeseq.impedence} = \frac{Startingcurrent}{Ratedcurrent} \tag{1.9}$$

A typical LV motor starting current is 6 x full load current (FLC). Therefore, a 5% negative sequence voltage (due to, say, unbalanced loads on the system) would produce a 30negative sequence current in the machine, leading to excessive heating. For the same motor, negative sequence voltages in excess of 17% will result in a negative sequence current larger than rated full load current.

Negative sequence current is at twice supply frequency. Skin effect in the rotor means that the heating effect in the rotor of a given negative sequence current is larger than the same positive sequence current. Thus, negative sequence current may result in rapid heating of the motor. Larger motors are more susceptible in this respect, as the rotor resistance of such machines tends to be higher. Protection against negative sequence currents is therefore essential.

Modern motor protection relays have a negative sequence current measurement capability, in order to provide such protection. The level of negative sequence unbalance depends largely upon the type of fault. For loss of a single phase at start, the negative sequence current will be 50% of the normal starting current. It is more difficult to provide an estimate of the negative sequence current if loss of a phase occurs while running. This is because the impact on the motor may vary widely, from increased heating to stalling due to the reduced torque available.

When configuring negative sequence current protection, it's crucial to recognize that the motor circuit safeguarded by the relay may not always be the origin of the negative sequence current. Therefore, the protection settings should allow time for the appropriate measures to clear the actual source of the negative sequence current, ensuring the motor under consideration doesn't overheat. This necessitates a two-stage tripping characteristic akin to overcurrent protection.

An initial low-set definite time delay can serve as an alarm, while an inverse definite minimum time (IDMT) element is employed to trip the motor in cases of higher negative sequence current levels, like loss-of-phase conditions during startup. Usual settings might entail 20% of the current transformer (CT) rated primary current for the definite time element and 50% for the IDMT element. The IDMT time delay must be carefully selected to safeguard the motor and ideally coordinate with other negative sequence relays on the system. In instances where relays lack dual elements, the single element should prioritize motor protection, with coordination being a secondary consideration.

1.11 Faults in rotor winding protection

Wound rotor machines can be shielded to some extent from rotor winding faults through an instantaneous stator current overcurrent relay element. Given that starting current is typically restricted by resistance to a maximum of twice the full load, setting the instantaneous unit to around three times full load with a brief time delay of approximately 30 milliseconds can ensure safe operation. It's important to highlight that any faults emerging in the rotor winding wouldn't be identified by any differential protection mechanism applied solely to the stator.

1.12 Resistance temperature detection

RTDs serve the purpose of measuring temperatures within motor windings or shaft bearings. A temperature increase can indicate machine overload or the onset of a fault in the affected area. Consequently, a motor protection relay typically features the capability to accommodate multiple RTD inputs and internal logic to trigger an alarm and/or trip when temperatures surpass the appropriate setpoint(s). In certain scenarios, high voltage (HV) motors are supplied via a unit transformer. In such cases, some of the RTD inputs of the motor protection relay may be allocated to monitor the transformer winding temperatures, thus offering overtemperature protection for the transformer without necessitating a separate relay.

1.13 Under voltage protection

Motors can experience stalling under prolonged undervoltage situations. In cases of transient undervoltages, motors typically recover once voltage is restored, unless the power supply is insufficient. Motors powered by contactors typically come with inherent undervoltage protection, except when a latched contactor is in use. For instances where specific undervoltage tripping is necessary, a definite time undervoltage element is employed. If two elements are available, both alarm and trip settings can be utilized.

An interlock with the motor starter is essential to prevent relay operation when the starting device is open; otherwise, motor starting would be indefinitely inhibited. Voltage and time delay settings must be tailored to the system and motor characteristics, accommodating all probable voltage dips during transient faults, motor startups, etc., to prevent unwarranted trips. Given that motor starting can cause voltage dips down to 80% of nominal, the voltage setting will likely be below this threshold. Re-acceleration is typically feasible for voltage dips lasting between 0.5 to 2 seconds, contingent upon system, motor, and drive attributes, thus necessitating time delay settings that consider these factors.

1.14 Loss of load protection

Loss-of-load protection serves various purposes. It can safeguard a pump from losing its prime, halt a motor in the event of mechanical transmission failure (e.g., conveyor belt), or shield synchronous motors against supply loss. This function is typically executed through a low forward power relay element, which is interlocked with the motor starting mechanism to prevent operation when the motor is tripped, thereby averting unintentional motor starts.

During startup against minimal loads (e.g., compressor), this function may need to be temporarily disabled to prevent improper operation. The relay's setting is influenced by the specific protection required, and a time delay might be necessary after the element is activated to prevent operation during system transients, particularly crucial for synchronous motor loss-of-supply protection.

1.15 Motor protection examples

This section gives examples of the protection of LV induction motors.

1.15.1 Protection of LV motor

LV motors are commonly fed via fused contactors and therefore the tripping times of a protection relay for overcurrent must be carefully co-ordinated with the fuse to ensure that the contactor does not attempt to break a current in excess of its rating.The following table gives details of an LV motor and associated fused contactor.

Parameter	Symbol	Value	Unit
(a) LV motor example			
Standard		IEC 60034	
Motor Voltage		400	V
Motor kW		75	kW
Motor kVA		91.45	kVA
Motor FLC		132	A
Starting Current		670	%
Starting Time		4.5	sec
Contactor rating		300	A
Contactor breaking capacity		650	A
Fuse rating		250	A
(b) Relay settings			
Overcurrent		Disabled	-
Overload setting	Ib	4.4	A
Overload time delay	I>t	15	sec
Unbalance	I2	20	%
Unbalance time delay	I2>t	25	sec
Loss of phase time delay	<Ip	5	sec

1.15.1.1 CT ratio

The relay is set in secondary quantities, and therefore a suitable CT ratio has to be calculated. From the relay manual, a CT with 5A secondary rating and a motor rated current in the range of 4-6A when referred to the secondary of CT is required. Use of a 150/5A CT gives a motor rated current of 4.4A when referred to the CT secondary, so use this CT ratio.

1.15.1.2 Overcurrent (short-circuit) protection

The fuse provides the motor overcurrent protection, as the protection relay cannot be allowed to trip the contactor on overcurrent in case the current to be broken exceeds the contactor breaking capacity. The facility for overcurrent protection within the relay is therefore disabled.

1.15.1.3 Thermal overload protection

The motor is an existing one, and no data exists for it except the standard data provided in the manufacturer's catalogue. This data does not include the thermal (heating) time constant of the motor.
In these circumstances, it is usual to set the thermal protection so that it lies just above the motor starting current.
The current setting of the relay I_b, is found using the formula

$$I_b = 5\frac{I_n}{I_p} \tag{1.10}$$

Where :
I_n = motor rated primary current
I_p = CT primary current
Hence:

$$I_b = 5 * \frac{132}{150} = 4.4A \tag{1.11}$$

With a motor starting current of 670% of nominal, a setting of the relay thermal time constant with motor initial thermal state of 50% of 15s is found satisfactory, as shown in the previous figure.

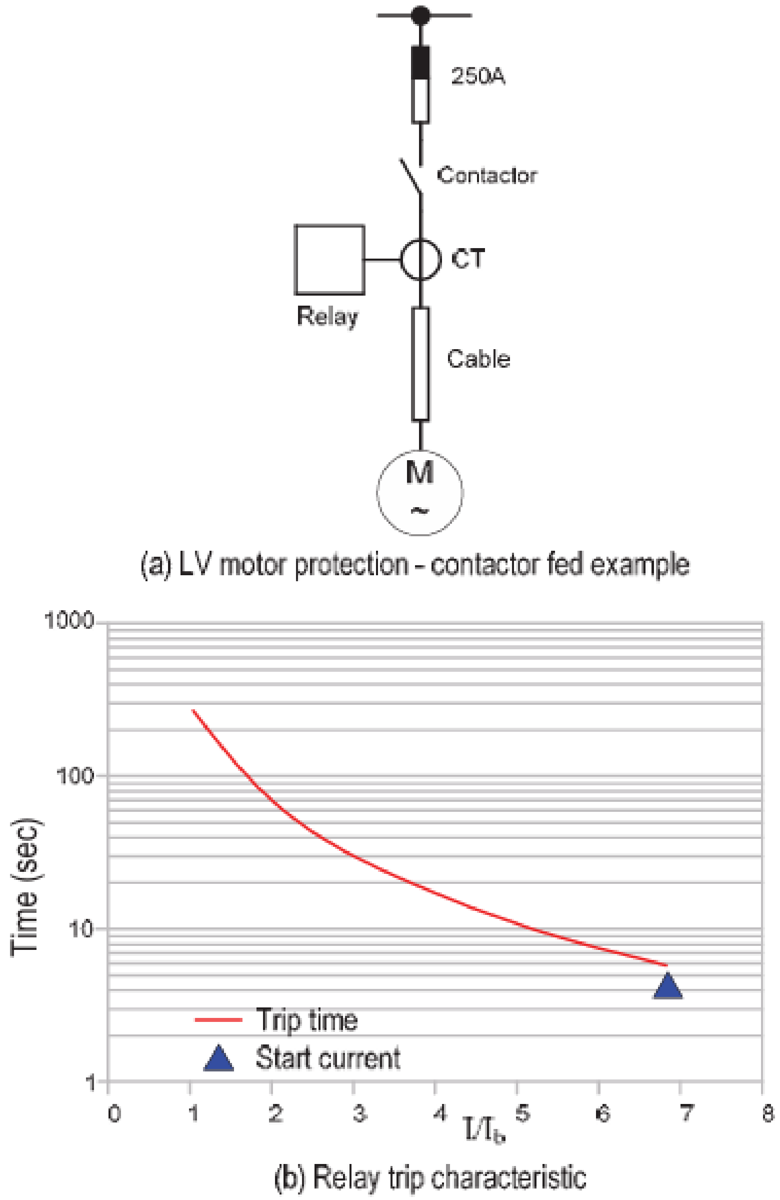


Figure 1.12: Motor protection example – contactor-fed motor

1.15.1.4 Negative sequence (phase unbalance) protection

The motor is built to IEC standards, which permit a negative sequence (unbalance) voltage of 1% on a continuous basis. This would lead to approximately 7% negative sequence current in the motor. As the relay is fitted only with a definite time relay element, a setting of 20% (from is appropriate, with a time delay of 25s to allow for short high-level negative sequence transients arising from other causes.

1.15.1.5 Loss of phase protection

The relay has a separate element for this protection. Loss of a phase gives rise to large negative sequence currents, and therefore a much shorter time delay is required. A definite time delay of 5s is considered appropriate.

1.16 Conclusion

Chapter 1 concludes by summarizing the essential aspects of electrical motor protection, highlighting the importance of safeguarding AC motors from various faults and conditions. It reiterates the need to consider motor characteristics and differentiate between external conditions and internal faults. This chapter sets the foundation for understanding modern relay designs and protection mechanisms that are critical for ensuring motor reliability and efficiency.

Chapter 2

A Comprehensive Overview of In Salah Gas Technology and Relay Protection Systems

2.1 Introduction

Chapter 2 introduces the In Salah Gas (ISG) project, detailing its organizational structure, site operations, and the integration of advanced relay protection systems like Gemstart 4 for effective motor control and monitoring.

2.2 In Salah Gas overview

2.2.1 Introduction

In Salah Gas is a subsidiary of Sonatrach that plays a significant role in natural gas production, accounting for almost 10% of Algerian gas.

It operates the gas from District 3 field, which is developed by Sonatrach, BP, and Statoil. Seven main fields have been identified: Krechba, Teg, Reg, Garet el Befinat, Hassi Moumene, In Salah, and Gour Mahmoud. Additionally, a new gas pipeline booster station and a fiscal metering facility are located in Hassi R'Mel, before connecting it to the national gas distribution center (CNDG). ISG is currently operated by joint-venture Sonatrach-ENI-Equinor.

2.2.2 In Salah Gas organisational philosophy

In Salah Gas is organized into different departments to cover functional and operational activities. Overall, ISG's organization is specific and differs greatly from traditional ones. The new aspect is that it's horizontal, allowing for greater flexibility and significantly facilitating communication between different hierarchical levels. In theory, the schematic aspect of this organization shows a certain complexity and significant scope, but in reality, traditional hierarchical boundaries are largely surpassed, thus allowing for the free flow of good ideas and personal innovations.

2.2.3 Sites overview

According to the development plan for the first phase, dehydrated gas from TEG and REG is sent to the central processing facilities (CPF) in Krechba through the 38-inch pipeline. After separation and cooling, dehydration, and conditioning, gas from the Krechba field is combined with gas from TEG and REG.

At the Krechba CPF, the gas mixture from the three sites is treated with an amine solution to remove CO2, and then the gas is dispatched to Hassi R'Mel. Hassi R'Mel is located 456.3 km north of Krechba. At Hassi R'Mel, the gas is recompressed and exported to the natural gas distribution center (CNDG) collector.

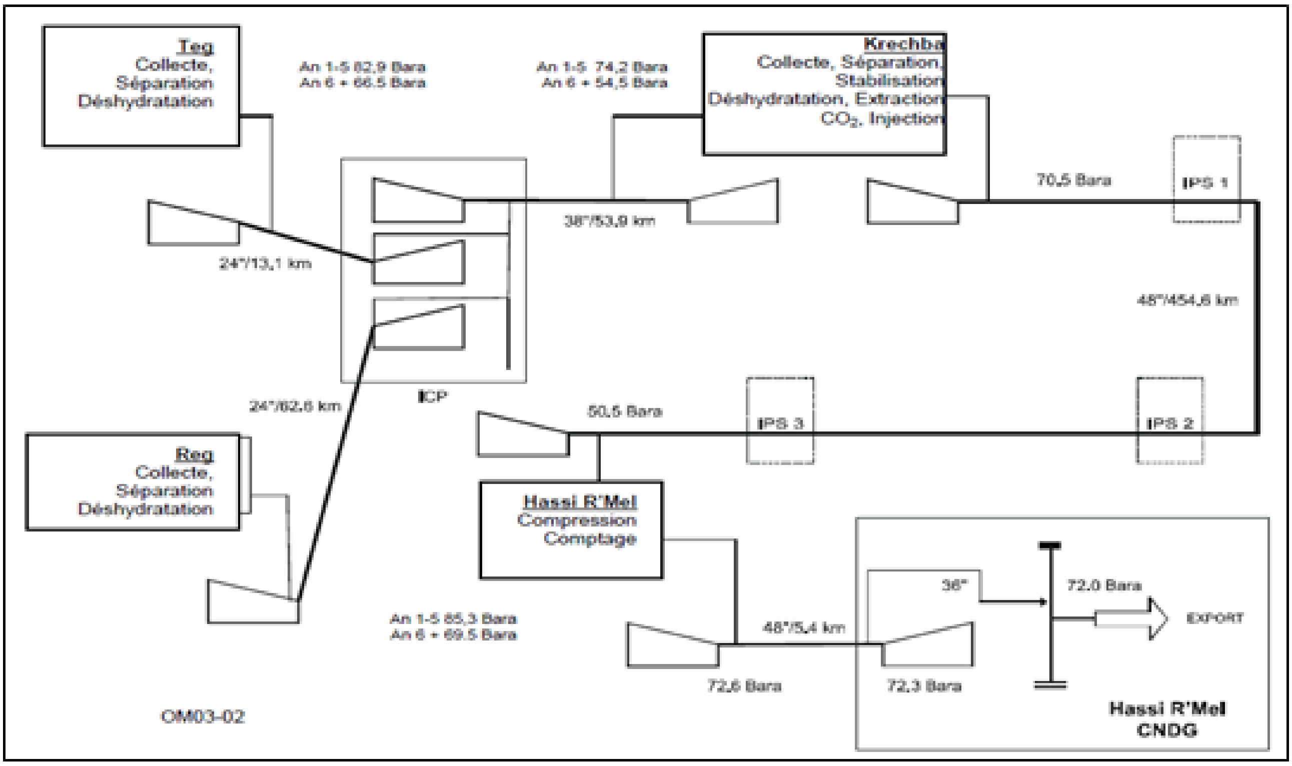


Figure 2.1: overview of installations

2.2.4 In Salah Central Processing Facility

The In Salah South Fields Project adds the four southern fields of Garet el Befinat (GBF), Hassi Moumene (HMN), In Salah (IS), and Gour Mahmoud (GMD) to support production when production decreases in the first three Northern Fields.

Gas from the IS, GMD, HMN, and GBF fields is dehydrated at a new CPF (IS CPF) located near the HMN field and then exported to REG CPF through a new carbon steel pipeline.

Gas from IS CPF will be routed to REG where it mixes with gas from the REG field, compressed, dehydrated, and exported in two parallel pipelines to ICP.

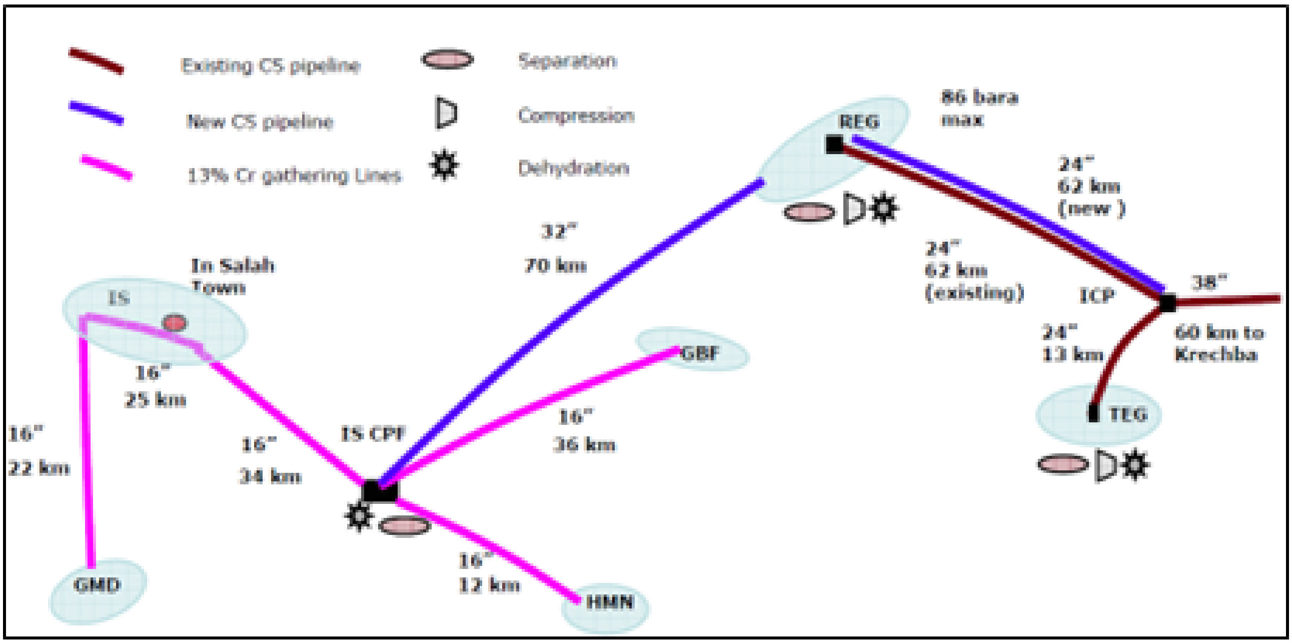


Figure 2.2: In Salah field (IS)

In Salah CPF includes:

- Three receiving pigs on the IS/GMD, GBF and HMN pipelines
- Finger-type condensate separator
- Input separator
- Two dehydration trains
- Produced water system including re-injection
- Launcher wiper
- Utilities

2.3 What is Gemstart 4

Gemstart 4 is at the heart of an intelligent motor control centre.It is used to control and monitor motors installed in a wide range of process applications. One Gemstart 4 unit is required per motor and can operate as part of a network or as a stand alone device.The next figure illustrates a typical Gemstart 4.3 unit.

It is used to provide any or all of the following facilities:

- Motor protection, for example protection against thermal overload, motor startup current and earthing faults
- Application protection; for example protection against under current and instantaneous overcurrent
- Process monitoring to check for phase current imbalance
- Maintenance prediction, for example the hours run and the number of start/stop operations
- Motor control, for example startup duration time
- User interface; which for example provides fault descriptions and warnings information

A Gemstart 4 unit consists of the following:

- Analogue inputs to monitor phase currents, phase voltage and earth faults
- Digital inputs to monitor plant conditions and push button control
- A serial link for remote control and condition monitoring (up to 100 units per link)
- Outputs to operate contactors or provide remote signalling
- A Liquid Crystal Display (LCD) to indicate status and mode
- Light Emitting Diodes (LED's) to highlight status
- Hand Held Programmer port

These facilities allow for control and monitoring of not just a single specific motor but, when used as part of an intelligent control system, the control and monitoring of a whole process.

The true power of Gemstart 4 is its ability to be integrated into such a scheme. You are provided with operator feedback, remote logging and full Distributed Control System (DCS) integration which enables you to introduce predictive maintenance schemes and fully integrated automatic plant. Further, because all communications are serial link based, your installation and commissioning costs are vastly reduced.

As well as remote monitoring Gemstart 4 also has a local information display.The LCD shows you information in your chosen language, for example motor current and fault condition. LED's highlight specific running conditions, for example running, stopped.

You are able to obtain more detailed information via the Hand Held Programmer(HHP) where the configuration of the unit can be viewed together with the trip history and a more detailed fault description. The HHP also allows you to control the motor locally. Security of the HHP is guaranteed by the inclusion of a key switch which avoids inadvertent or unauthorised operation of the control functions.

Figure 2.3: A typical Gemstart 4.3 unit

2.3.1 Gemstart protection functions

- Protection/FLC Setting
- Protection/Startup Current
- Protection/Startup Time
- Protection/Thermal Overload
- Protection/Restart Inhibit
- Protection/Single Phasing
- Protection/Imbalance
- Protection/Voltage protection
- Protection/Earth Fault Protection

2.3.2 Input functions

- **Interlock** An active signal from a field switch will cause an INTERLOCK trip.
  Usually a normally closed self clearing trip(Interlock). Used to indicate that an unsafe condition has occurred and the contactor requires opening. Usually applied to the Interlock input where it directly removes the feed to the contactor

- **Manual Start A / Manual Start B** A signal from Push-button requesting the motor to be started. An active signal will cause a start request for contactor A (or B). If the unit is in manual mode and available then the unit will close the relevant contactor. If a single contactor type (such as DOLE) is selected then Manual Input B acts as a second source of start for that contactor.

- **Process Stop A / Process Stop B** A signal from field push button. A microprocessor based STOP signal. This signal causes the contactor(s) to open only if Gemstart is in manual mode. The contactor will not open if manual mode has not been selected.

- **Hard Stop**Hard Stop is a signal from field push button. It is a microprocessor based STOP signal. This signal causes the contactor(s) to open in any mode. A trip will not be reported if the unit is in manual mode or manual mode can be forced.

2.3.3 Circuit diagram of Gemstart 4.3

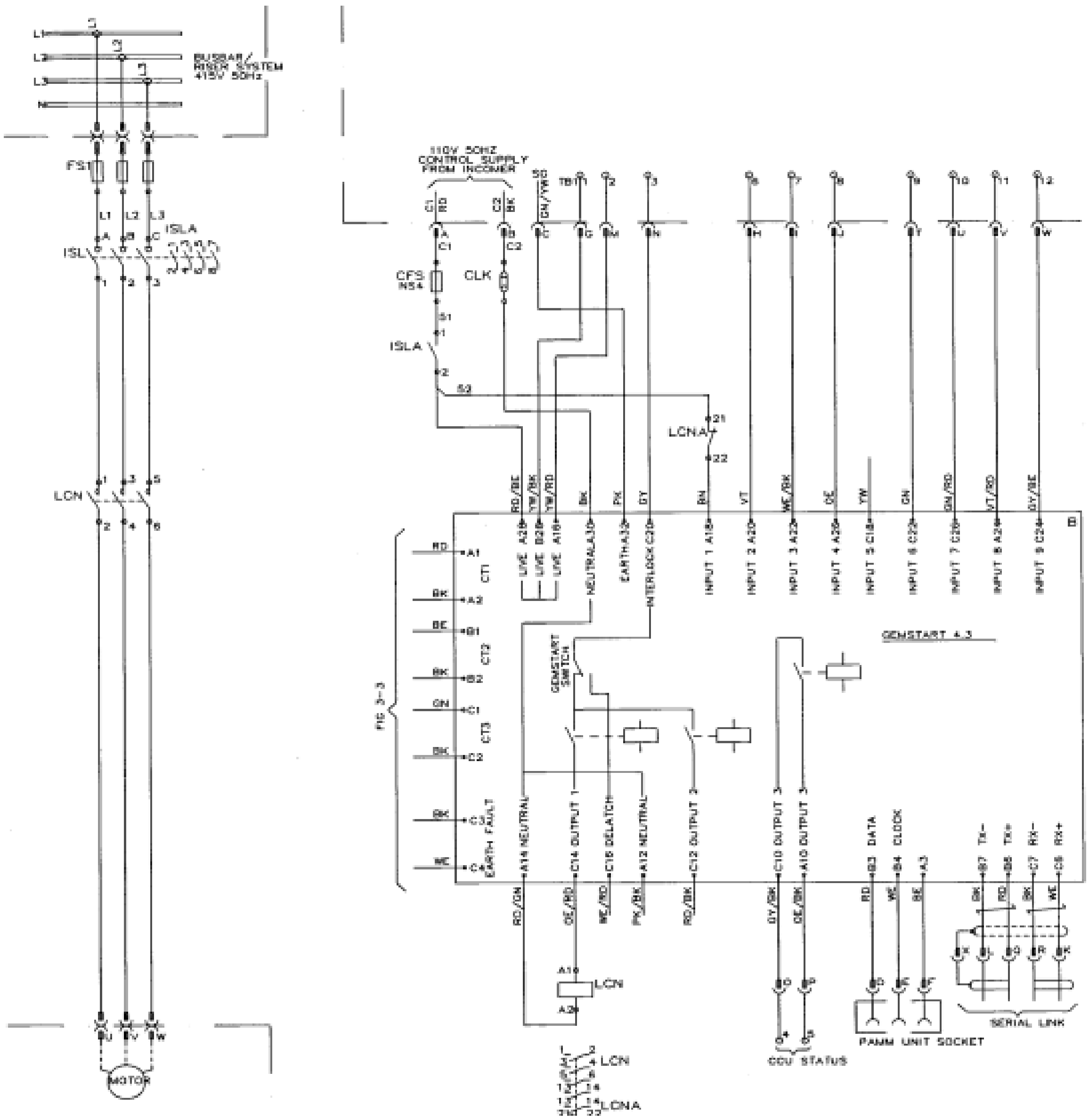


Figure 2.4: Gemstart Circuit

2.3.4 Current Transformer Selection

A current transformer (CT) is a type of instrument transformer used to measure alternating current (AC) by producing a reduced current proportional to the current flowing in a circuit.

It consists of a primary winding through which the current to be measured passes and a secondary winding connected to measuring instruments. CTs are crucial for accurate current measurement in power systems and electrical equipment.

To measure the current Gemstart 4 requires a Current Transformer (CT) to sense the current. Two or three CT's are required.
If you use two CT's then the third phase current is generated by residually summing the other two phases.

Note: If an earth fault occurs in the third phase a two CT Gemstart 4 will not sense it without a Core Balance Transformer

Motor Run Current (Amps)	CT Value	Phases measured
0.5 to 3 Amps 1.0 to 6 Amps 1.5 to 9 Amps 2.5 to 15 Amps 5.0 to 30 Amps 7.5 to 45 Amps	2Amps 4Amps 6Amps 10Amps 20Amps 30Amps	L1,L2,L3
10 to 60 Amps 15 to 90 Amps 25 to 150 Amps	40Amps 60Amps 100Amps	L1,L2,L3

The FLC rating is based on the nominal rating of the CT but you can adjust it down to 25% of the CT rating and up to 150% of the CT rating.

2.3.5 Earth fault measurement

It is also possible to use residually connected CT's to measure earth fault current. This is provided for by externally connecting the CT outputs through a load resistor.

Both methods of earth fault have their advantages and disadvantages as outlined in Table:

CBCT advantages	**CBCT disadvantages**	**Residual advantages**	**Residual disadvantages**
Sensitivity independent of phase current	Bulky ,Expensive, Inflexible	Cheap ,More compact	Accuracy relies on CT matching

2.4 Sub-Station PLC overview

2.4.1 PLC System

One Hot Standby RTU PLC system is installed at each package substation at the Krechba, Hassi R'Mel, Reg and Teg sites. Each PLC system is dedicated to provide the control and monitoring of the associated switchgear located within each substation.

Each PLC system is made up of two identical GE Fanuc PLC racks. At any one time, one PLC rack performs 'duty' mode operations (or online) providing communications functions to each part of the system, while the other PLC is in 'standby' mode (offline), ready to take the 'duty' status in the event of failure of the other PLC or other system component.

As far as is practicable each PLC rack utilises its own system components to provide the system with the greatest measure of redundancy possible. The PLC system communicates via serial links to Gemstart 4 units, Micom units (via KITZ) and the DCS Yokagawa system. It also provides communication to the Gemview SCADA system and between each pair of PLC racks via a TCP/IP Ethernet network.

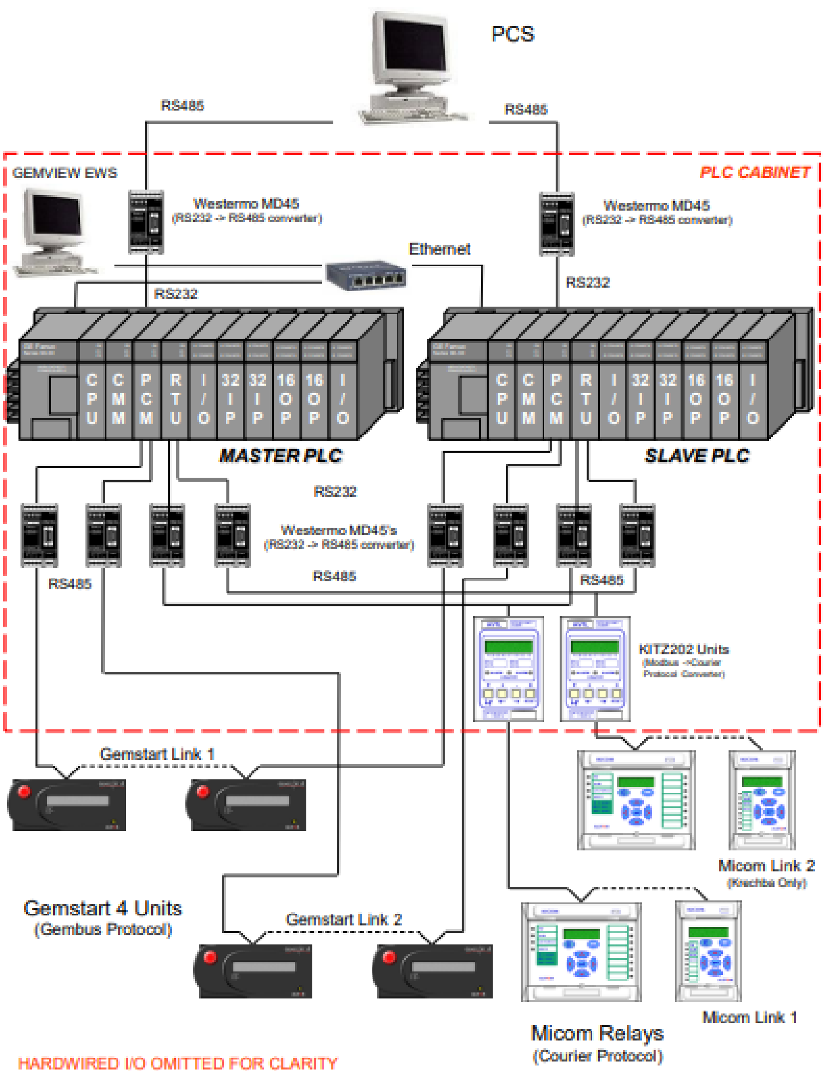


Figure 2.5: Generic overview

2.4.2 Gemview SCADA

The Gemview system is a bespoke SCADA system that communicates to the 'duty' (online) PLC via a TCP/IP Ethernet network. The Gemview SCADA uses the PLC system communications links to provide monitoring and configuration of the Gemstart4 units, and monitoring and control of the Micom units. Various data screens are available for the retrieval and monitoring of data. Available functions include, but are not limited to:

- **Gemstart and Micom status screens** Running / tripped / alarm status, communications status, fault codes, analogue measurement values, etc.

- **Gemstart Motor Trending** Real time trend graphs for all Gemstart motor currents, including historical trend current database.

- **Event Log** Database that logs all Gemview SCADA events and alarm, trip and communications events for Gemstart and Micom units

- **Single Line Diagram** Pictorial overview of the electrical distribution system for the substation, denoting circuit breaker states and conditions.

2.4.3 Gemstart

The PLC system references the Gemstarts by their serial link number and serial link address. This is clearly displayed on the left side of the LCD:

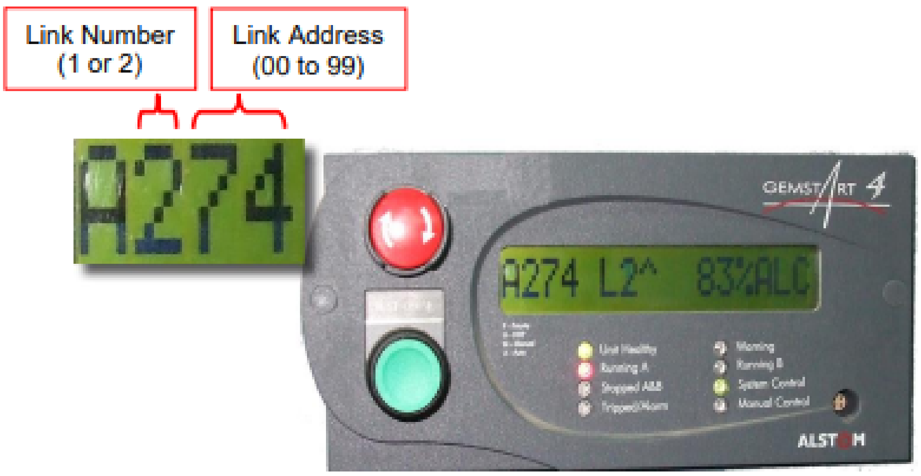


Figure 2.6: Link number and address

2.4.4 MICOM Relays

A range of Micom units are installed in the switchgear to provide motor and transformer protection. Micom units have the ability to communicate on a serial link to the PLC via a KITZ unit.
The PLC system references the Micoms by their serial link address.

Figure 2.7: MICOM

2.4.5 KITZ Units

The KITZ unit is an intermediary device that provides protocol conversion and data concentration between the Micom relays and PLC system. The PLC system is then able to receive data from, and send control commands to the Micom relays via the KITZ unit.
The KITZ unit requires data mappings to be programmed specific to the number and type of connected Micom relays. Each KITZ has the capability of communicating with up to a maximum of 20 Micom relays.
The KITZ unit is also programmed with its own serial link address of 21 (with addresses 1-20 being reserved for the use of any connected Micom relays).

Figure 2.8: KITZ

2.4.6 PLC Panel Layout

Following are photographs of the Krechba PLC panel

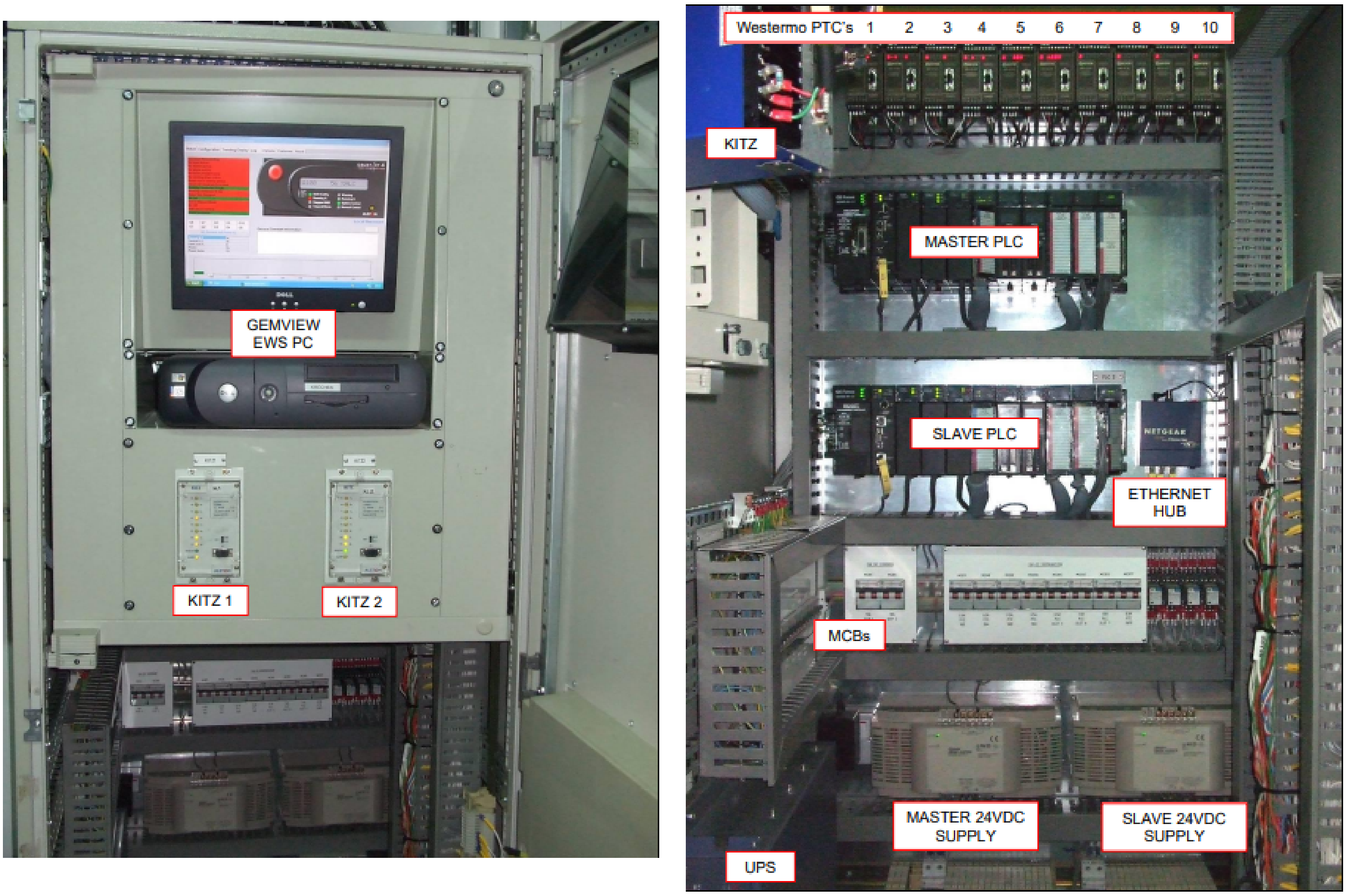


Figure 2.9: PLC Panel

Note: The system comprises two KITZ units, each configured to manage 20 motors rated at 5.5 kV. To facilitate seamless communication, the system employs a Modbus protocol, which interfaces with a Courier Protocol Converter. This arrangement ensures efficient and reliable data exchange for the control and monitoring of all 40 motors, allowing for effective management and operation within the network. The use of the Courier Protocol Converter optimizes the communication process, enhancing the overall performance and responsiveness of the system.

2.5 Conclusion

Chapter 2 concludes by emphasizing the importance of the In Salah Gas (ISG) project's organizational structure and its operational effectiveness in Algeria's natural gas production. The chapter highlights how ISG's integration of advanced relay protection systems, like Gemstart 4, ensures efficient motor control and monitoring. The detailed examination of ISG's processes and technologies underscores the project's critical role in maintaining reliable and safe gas production operations. This sets the stage for further discussions on technological advancements and their practical applications in industrial settings.

Chapter 3

General overview of a PLC & Tia Portal Software

3.1 Introduction

In this chapter, we will provide a general overview of the programming and supervision software (TIA Portal) and what is a programmable ladder controller. We will summarize all the possible tasks to be handled in our project and list the main parts that are found in the PLC,its programming language and the wich type we will be using. As a reader of this thesis (e.g., this chapter), you can learn about programming or creating automation projects using TIA Portal.

3.2 General information about Programmable Logic Controller (PLC)

3.2.1 Definition

A Programmable Logic Controller (PLC) is an industrial digital computer designed to perform control functions, typically for manufacturing processes or machinery. It is robust and adaptable, capable of withstanding harsh industrial environments. PLCs receive input signals from sensors and devices, process these signals based on a pre-programmed logic, and then send output signals to actuators and other equipment to control and automate processes. They are essential components in modern industrial automation due to their reliability, flexibility, and ability to integrate with various control systems.

3.2.2 PLC Architecture

- **Internal Architecture**

1. Power Module: It ensures the distribution of energy to the different modules.
2. Central Processing Unit (CPU): Based on a microprocessor, it performs all logical, arithmetic, and numerical processing functions (transfer, counting, timing, etc.).
3. Internal Bus: It allows communication between all blocks of the PLC and any extensions.
4. Memories: They store the operating system (ROM or PROM), the program (EEPROM), and system data during operation (RAM).
5. Input/Output Interfaces:
   Input Interface: It receives information from the Process Automation System (S.A.P.) or the control panel and formats (filtering, etc.) this signal while electrically isolating it (optocoupling).
   Output Interface: It controls various actuators and signaling elements of the S.A.P. while ensuring electrical isolation.

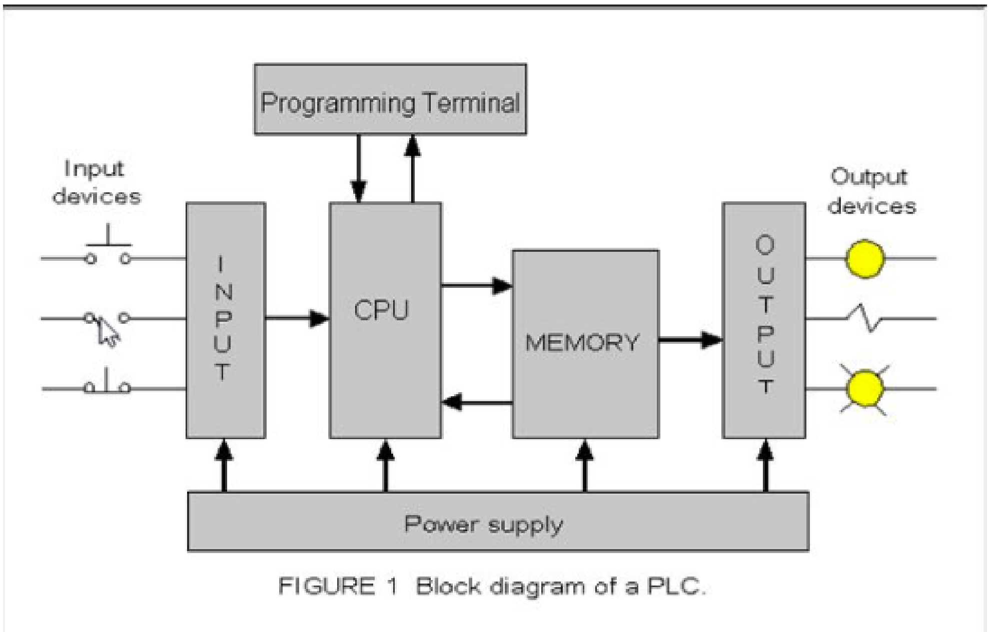


Figure 3.1: Internal Architecture

- **External Aspect**

PLCs can be compact or modular.

- Compact PLCs: These include programming modules (e.g., Siemens LOGO, Schneider ZELIO, Crouzet MILLENIUM) and micro PLCs. They integrate the processor, power supply, inputs, and outputs. Depending on the models and manufacturers, they may perform additional functions (fast counting, analog I/O, etc.) and receive a limited number of extensions.
- Modular PLCs: The processor, power supply, and input/output interfaces reside in separate units (modules) fixed on one or more racks. These PLCs are integrated into complex automation systems where power, processing capacity, and flexibility are required.

Figure 3.2: Compact PLC (Allen-Bradley)

Figure 3.3: Modular PLC (Siemens)

- **Elements of a PLC**

1. The Processor: The processor is the core of the PLC, responsible for processing the operation program instructions. In addition to this main task, it performs several other important functions (managing the PLC's inputs and outputs, conducting diagnostic tests, and monitoring the PLC's status).
   To accomplish these functions, the processor is equipped with registers, which are fast memories used to store and manipulate data. These registers can also be used to combine external data with that of the PLC to perform more complex operations.

2. Memories: The PLC stores information in its memory, which consists of several elements:
   • ROM (Read Only Memory): Used to store the operating system and data required by the processor.

   • RAM (Random Access Memory): Used to store the user program, temporary data, and information about the status of inputs and outputs.
   • EPROM (Erasable Programmable Read Only Memory): Sometimes used to permanently store programs.

3. Input/Output Modules: Input/output modules play a crucial role in industrial automation by connecting the CPU to the process. They retrieve information about the process state and coordinate actions. Several types of modules are available depending on the requirements:
   • Digital Input/Output (TOR) Modules: Process binary information (true/false, 0 or 1). They are used to process signals from sensors with electrical contacts such as push buttons or switching points.
   • Analog Input/Output Modules: Process continuous information within a given range (e.g., 4-20 mA). They are used to process signals from sensors with analog outputs such as flow meters, level sensors, thermometers, etc.
   • Specialized Modules: Process binary or hexadecimal code words. They are used to process signals from computers or intelligent modules.

4. Power Supply: The PLC's power supply transforms the 220V AC mains electricity or 24V DC from an external source into appropriate internal voltages for the PLC modules. It is equipped with monitoring devices that detect voltage drops or mains power failures and monitor internal voltages to ensure optimal safety. In case of failure, these devices can trigger a priority backup procedure.

5. Communication Links: Communication links in the PLC are essential for enabling communication between different blocks of the PLC and any extensions. They are made both externally through terminals for electrical signals transported by cables and internally through buses connecting different elements to exchange data, statuses, and addresses.

6. PLC Program Processing: The processing of the PLC program follows a standard procedure. Firstly, the PLC performs control operations and updates certain system parameters, such as detecting transitions in RUN/STOP mode and updating timestamp values. Then, the PLC reads inputs synchronously and stores them in the input image memory. Next, the PLC executes the program instruction by instruction and writes the corresponding outputs to the output image memory. Finally, the different outputs are switched to the defined positions synchronously.

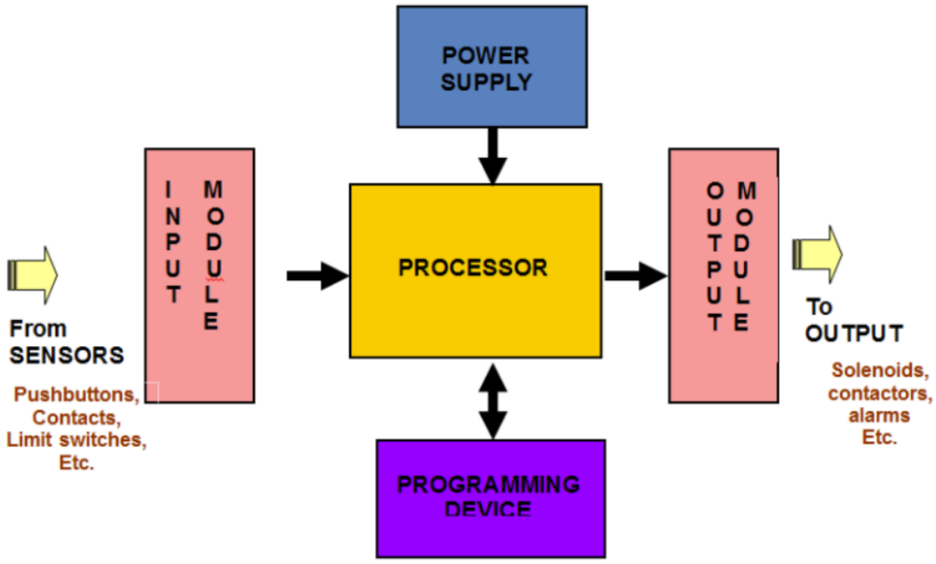


Figure 3.4: Elements of PLC

3.2.3 Programming Languages

Program descriptions can be created using the following graphical or textual languages:

- **Graphical Languages**

A/ Ladder Diagram (LD)

The Ladder Diagram language is based on a visual approach reminiscent of electrical schematics (with US symbols). In the case of Boolean processing, the fundamental elements are normally closed (NC) or normally open (NO) contacts, and coils. This language is very effective for combinational systems. However, Ladder is not suitable for complex programs involving a large number of sequences or for complex calculations.

B/ Function Block Diagram (FBD)

Function Block Diagrams are a graphical language that allows programming using interconnected functional blocks. Functional blocks can be provided by a library or customized by the user. One of the advantages of this language is the reduction of errors by using validated blocks. FBD is the most commonly used language.

C/ Sequential Flow Charts (SFC)

Sequential Flow Charts are derived from the GRAFCET language. This high-level language allows easy programming of all sequential processes and is similar to state-transition diagrams. They are particularly suited for controlling operating cycles. Graphically, there is an alternation of step(s) — transition-step(s), etc., with directed connections. Each transition is associated with a condition for crossing that transition; each step can define actions to be taken. Conditions for crossing and actions are expressed in one of the languages described above.

- **Textual Languages**

A/ Instruction List (IL)

Instruction Lists are very similar to assembly language. An instruction starts on a line, contains an operator, and one or more operands. Labels and comments can be introduced. This language is not considered the most practical by most users, as transitioning from often semi-graphical analysis to programming is not always easy, nor is debugging the program.

B/ Structured Text (ST)

Structured Text is a high-level textual language used to describe complex procedures. It is very similar to the Pascal language. It uses expressions, an ordered assembly of operators, with priorities. This language thus facilitates the implementation of complex algorithms involving a lot of numerical processing. On the other hand, it is less convenient for developing Boolean functions.

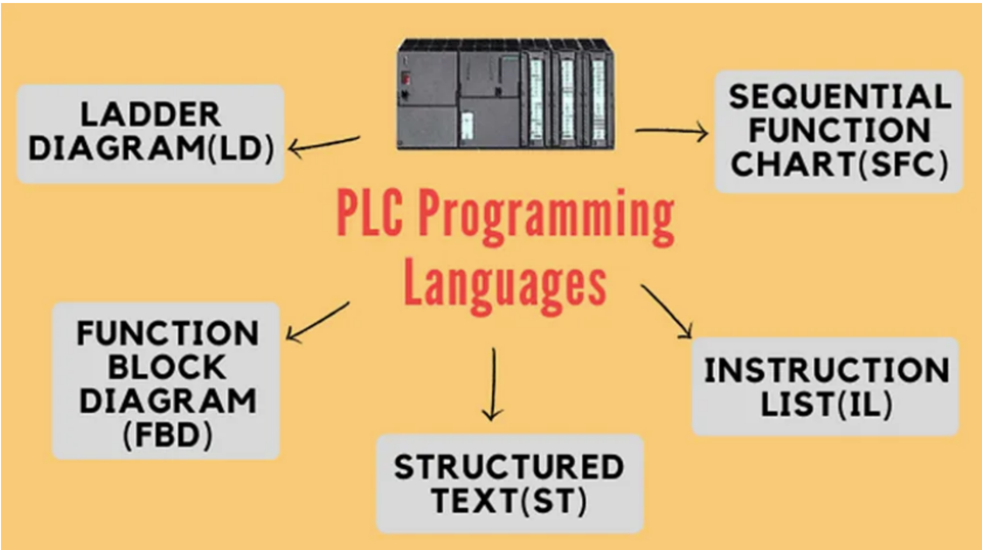


Figure 3.5: Different Types of PLC Programming Languages

3.2.4 PLC S7-1200

- **Definition of PLC S7-1200**

The Siemens S7-1200 is a programmable logic controller (PLC) designed for a wide range of automation applications. It is part of the SIMATIC family of controllers and is known for its compact design, flexibility, and ease of use. The S7-1200 is suitable for both simple and complex automation tasks, making it ideal for small to medium-sized control applications.

Figure 3.6: PLC S7-1200

- **Characteristics of PLC S7-1200**

1. Compact Design: The S7-1200 has a modular and compact design, which saves space and allows for easy installation in control cabinets.

2. Flexible Configuration: It supports a variety of expansion modules, including digital and analog input/output modules, communication modules, and special function modules.

3. Integrated PROFINET Interface: This allows for seamless communication and integration within industrial networks, providing connectivity to other devices and systems.

4. Scalability: The S7-1200 can be scaled to fit the requirements of different applications, from simple control tasks to more complex automation solutions.

5. High Performance: It features a powerful CPU that ensures fast processing speeds and high performance in executing control tasks.

6. User-Friendly Programming: The S7-1200 can be programmed using the TIA Portal (Totally Integrated Automation), which provides an intuitive and efficient programming environment.

7. Built-in Security Features: The PLC includes integrated security functions to protect against unauthorized access and data manipulation.

8. Versatile Communication Options: Besides PROFINET, the S7-1200 supports various communication protocols, such as Modbus TCP/IP, and can connect to a wide range of devices and systems.

9. Diagnostics and Maintenance: The PLC offers comprehensive diagnostic tools and maintenance features, which help in reducing downtime and improving system reliability.

10. Energy Management: It includes features for energy management and monitoring, which contribute to energy-efficient operations.

- **Conclusion:**

The Siemens S7-1200 PLC is a versatile and powerful controller suitable for a wide range of automation tasks. Its compact design, high performance, and flexibility make it an excellent choice for small to medium-sized applications. With its integrated PROFINET interface, user-friendly programming environment, and robust security features, the S7-1200 ensures efficient and reliable control of industrial processes. Whether used for simple control tasks or more complex automation solutions, the S7-1200 provides a scalable and efficient platform for modern industrial automation.

3.3 General Presentation of the Software

3.3.1 Definition

The Totally Integrated Automation Portal platform is the new Siemens working environment that enables the implementation of automation solutions with an integrated engineering system including SIMATIC STEP 7 and SIMATIC Win CC software.

3.3.2 Software Usage

Totally Integrated Automation optimally meets all requirements while being open to international standards and systems from other manufacturers. With its six system properties (engineering, communication, diagnostics, safety (Safety), data security (Security), and robustness), Totally Integrated Automation supports the entire lifecycle of a machine or installation.

3.3.3 TIA Portal User Interface Structure

There are three different views for each automation project, and you can switch between views using a link.

3.3.3.1 Portal View

The portal view (Figure 3.7) offers a task-oriented view of the tools. You can quickly decide what you want to do and call the required tool for the corresponding task.

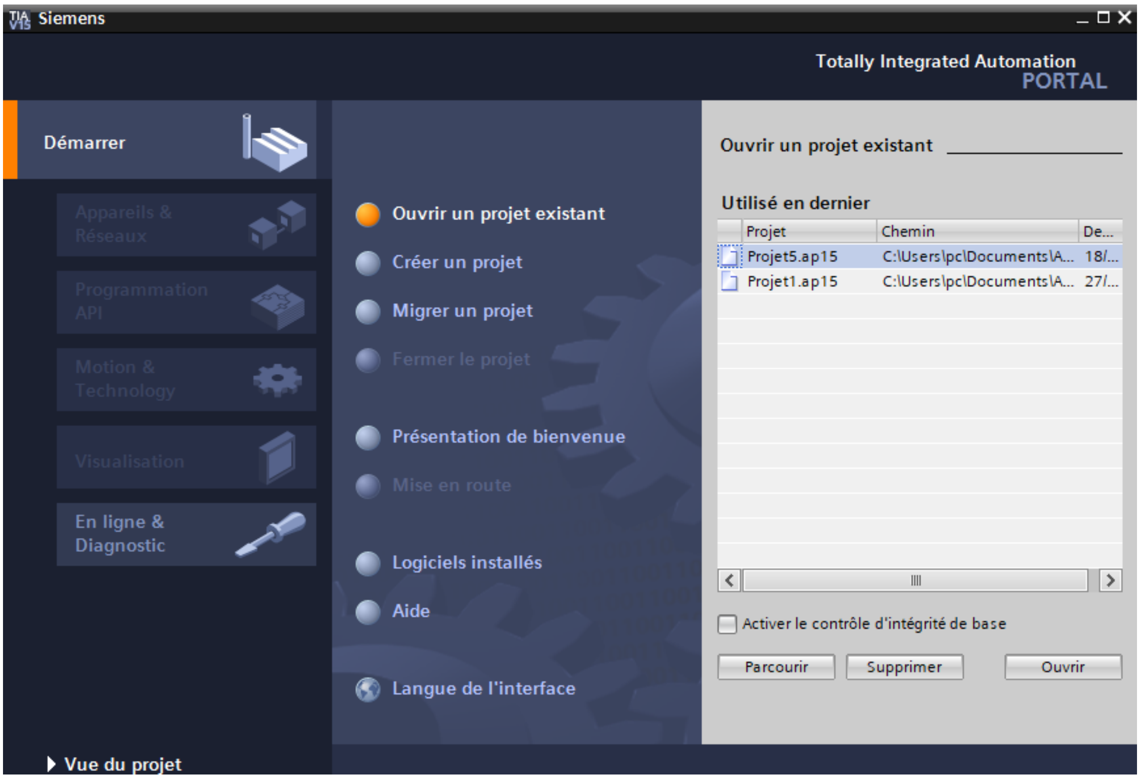


Figure 3.7: Portal view

- Portals for Different Tasks: Portals provide the basic functions required for each type of task. The portals available in the portal view depend on the installed products.

- Actions Corresponding to Selected Portal: Depending on the selected portal, the actions you can perform are offered.

- Selection Window Corresponding to Selected Action: The selection window available in each portal. Its content adapts to the current selection.

- The "Project View" link allows you to switch to the project view.

- Display of the Currently Open Project: Provides information about the currently open project.

3.3.3.2 Project View

The project view (Figure 3.8) corresponds to a structured view of all components of the project.

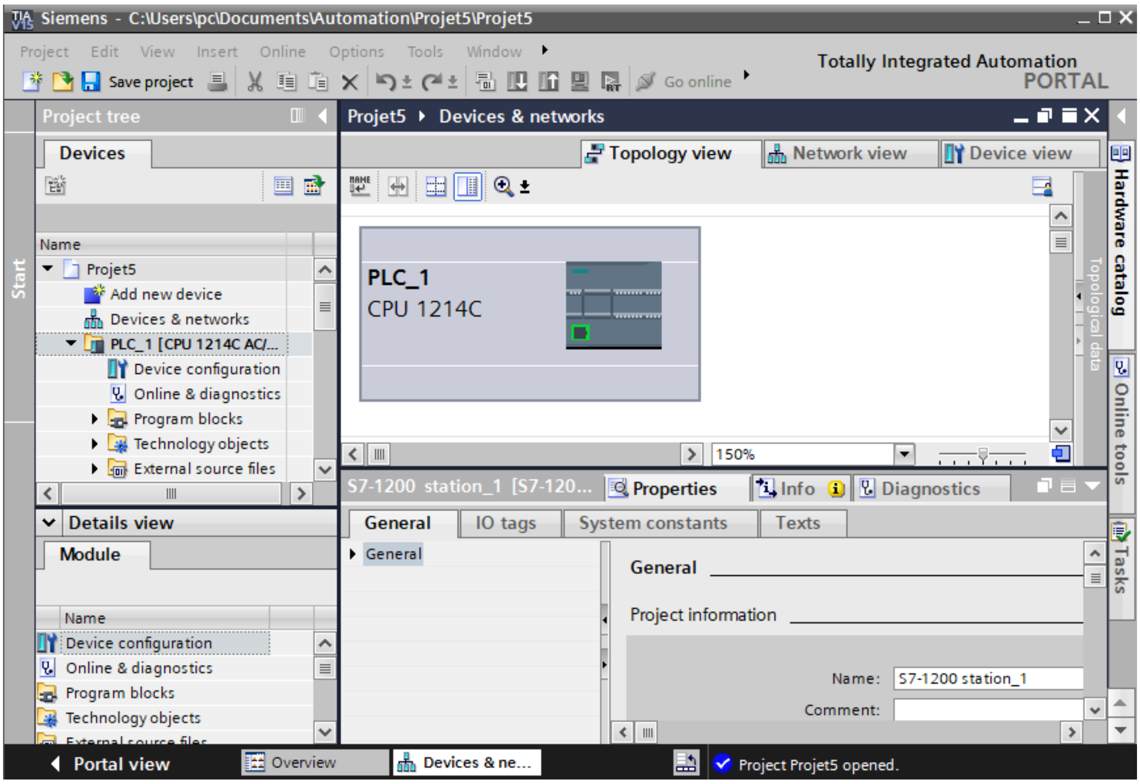


Figure 3.8: Project view

- The title bar displays the project name.
- The menu bar contains all the commands you need to perform your tasks.
- The toolbar provides buttons for executing the most commonly used commands.
- The project tree allows you to access all project components and data.
- The details view displays certain contents of a selected object in either the overview window or the project browser. This can include lists of texts or variables.
- The workspace displays the objects that are opened for editing. These can be editors and views or tables.
- Split bars separate different elements of the software interface, and the arrows on these split bars allow you to show or hide adjacent interface elements.
- The inspection window displays additional information about the selected object or the actions being performed.

- The "Portal View" link allows us to switch to the portal view.
- The editors bar displays the open editors.
- The status bar visualizes the progress of processes currently running in the background. You can interrupt background processes using the button next to the progress bar.
- Available Task Cards are displayed in a bar on the right edge of the screen. You can open or close this bar at any time, and the Task Cards offered depend on the installed products.

3.3.3.3 Library view

The library view displays an overview of the elements in the project library and any open global libraries. You can switch to the library view using the "Libraries" Task Card.

3.4 Create and manage projects

3.4.1 Creating a new project

To create a new project, follow these steps:

- Choose the "New" command in the "Project" menu (The "Create Project" dialog box opens).
- Enter the desired name and path for the project or use the suggested data.
- Click the "Create" button. The new project is created and displayed in the project browser, see (Figure 3.9).

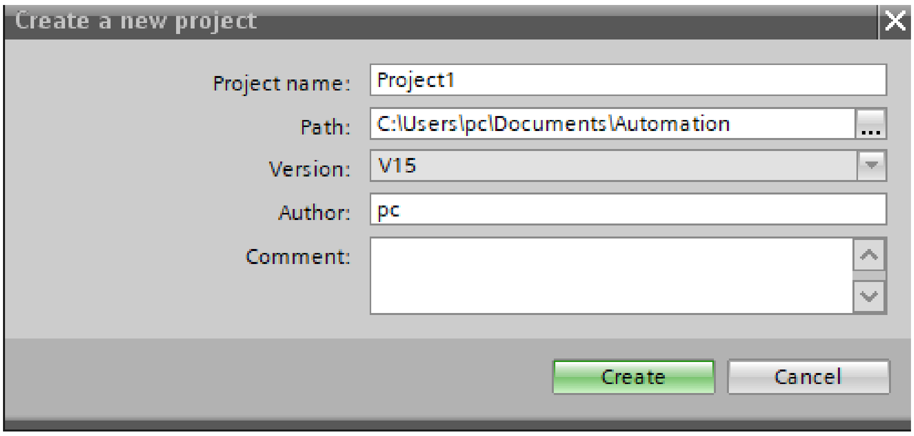


Figure 3.9: Creating a new project

3.4.2 Open projects

To open an existing project, follow these steps:

- Choose the "Open" command in the "Project" menu (The "Open Project" dialog box opens, displaying a list of recently used projects).
- Select a project from the list and click "Open."
- If the desired project is not in the list, click the "Browse" button, navigate to the desired project folder, and open the project file.

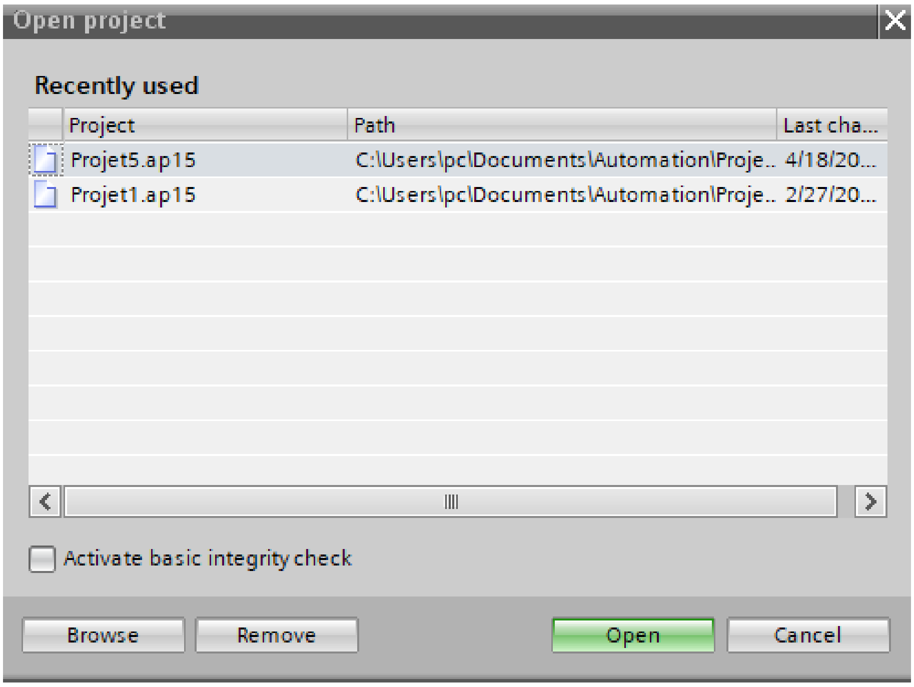


Figure 3.10: Open from an existing project

Projects in the current project format open in the project view. If you have chosen a project from an earlier version of TIA Portal, the "Upgrade Project" dialog box opens.

3.4.3 Viewing project properties

To display the project properties, follow these steps:

1. Select the open project in the project browser.
2. Choose the "Properties" command from the project's context menu (the dialog with the project properties opens).
3. Select the desired project properties from the screen navigation to display them.

Here are the points belonging to the properties:

- **Project Metadata:** Here, you can obtain information such as the creation date, author, backup directory, project size, copyright remarks, project languages, etc.

- **Project History:** The project history provides an overview of important elements in the lifecycle of a project. For example, you can find the version of TIA Portal with which a project was created and whether it has since been converted to another version, etc.

- **Support Packages in the Project:** An overview of additional software required for processing all devices in the project is displayed here. It also includes a list of installed GSD files (device description files for other devices in the hardware catalog).

- **Project Software Products:** You can display an overview of all installed software products required for the project.

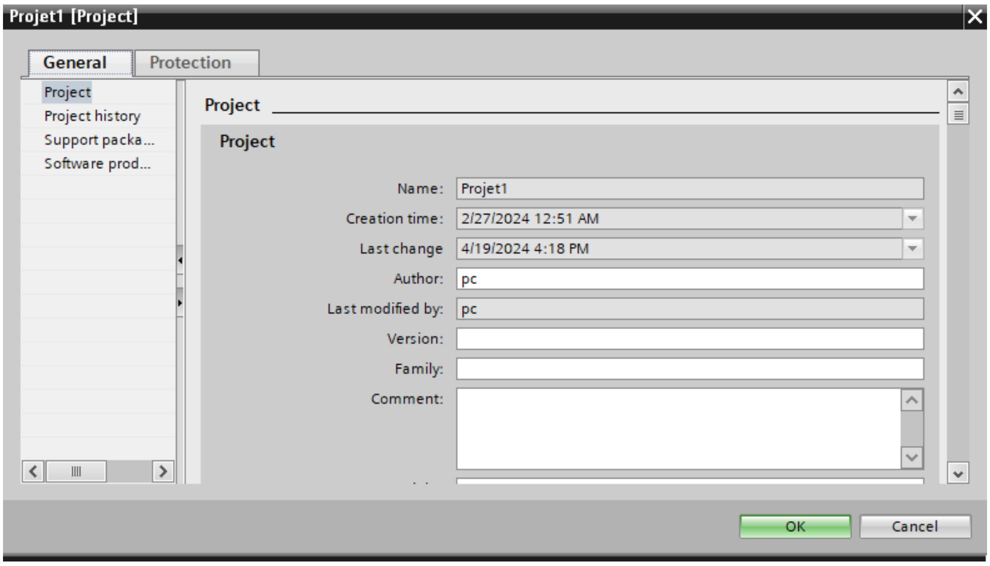


Figure 3.11: The window where we manage the properties of a project

3.5 Edit project data

3.5.1 Compilation of Project Data

During compilation, project data is converted so that it can be read by the device. Hardware configuration data and program data can be compiled together or separately.

The following project data must be compiled before loading:

- Hardware data of the project, e.g., device configuration data or networks and connections.
- Software data of the project, e.g., program blocks or process views.

To compile project data, follow these steps:

1. In the project browser, select the devices for which you want to compile the project data.
2. Choose the desired option from the "Compile" submenu of the context menu. When compiling project data, various options are available depending on the device:
   - Hardware and software (changes only)
   - Hardware (changes only)
   - Hardware (full compilation)
   - Software (changes only)
   - Software (full compilation of blocks)
   - Software (reset memory reserve)
3. The project data is compiled, so you can check the success of the compilation in the inspection window, under "Info == Compiler."

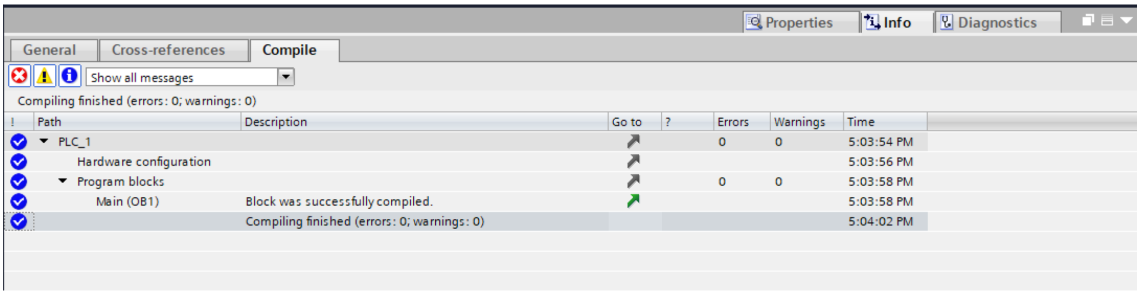


Figure 3.12: The window that displays the compilation status

3.5.2 Load project data

To configure your automation system, it's necessary to load the project data generated offline into the connected devices. This project data is generated, for example, during hardware configuration, network configuration, and programming of the user program.
During the initial loading, all project data is loaded. Subsequently, only modifications are loaded. You can load project data into devices and onto memory cards.

- **Load project data into a device:**

  If the project data is consistent and each device to be loaded is accessible via online access, you can load the project data into the selected devices. Here's how to proceed:

  1. Select one or more devices in the project browser.
  2. Right-click on a selected object. The context menu will open.
  3. Choose the desired option from the "Download into device" submenu of the context menu.

4. Check the messages in the "Loading Preview" dialog box.
5. Click the "Load" button. Once loading is possible, the "Load" button becomes active.
6. Click the "Finish" button. The selected project data has been loaded into the devices.

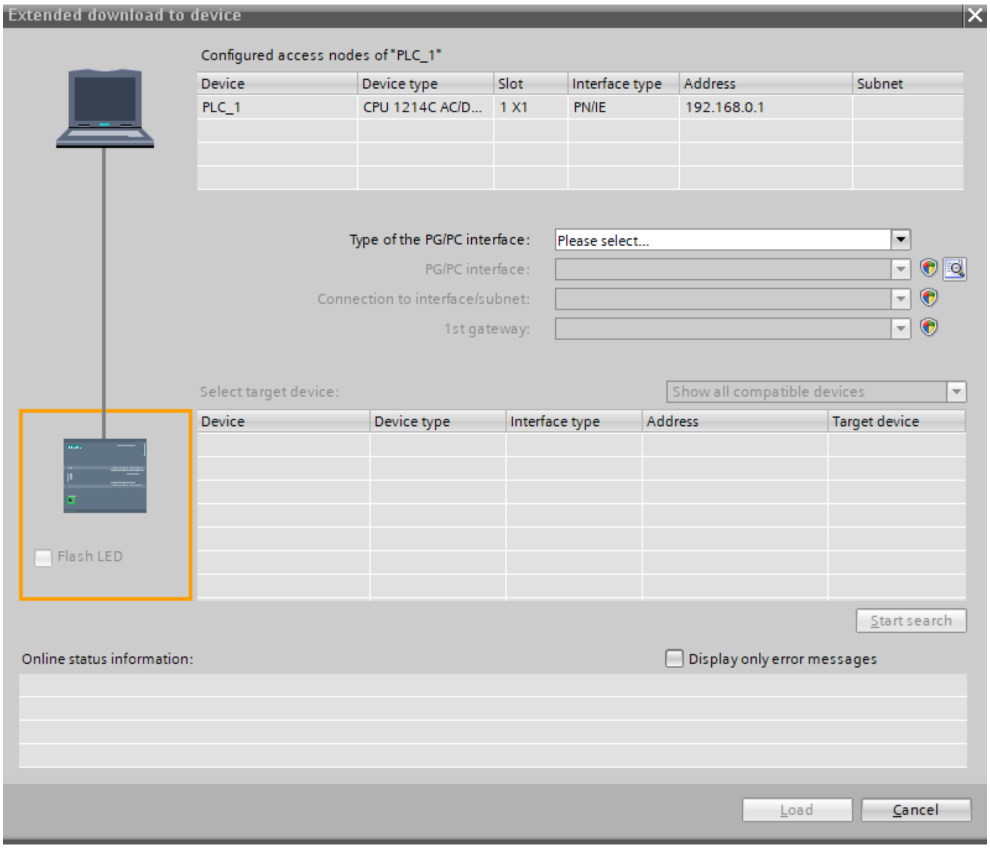


Figure 3.13: The loading window

Here are the options available for the object you want to load:

– Hardware and software (changes only): Both hardware configuration and software are loaded to the target in case of discrepancies between online and offline versions.
– Hardware configuration: Only the hardware configuration is loaded into the target.
– Software (changes only): Only elements that are different online and offline are loaded into the target.

- **Load project data from a device:**

If an object is open and if the hardware configuration and software to be loaded must be compatible with TIA Portal, you have two options for loading project data from a device to your project. To load the entire device into your project, follow these steps:

1. Select the project name in the project browser.
2. In the "Online" menu, choose the "Load device as new station (hardware and software)" command. The "Load device into PG/PC" dialog box opens.
3. Select the type of interface you want to use for the loading procedure and select the interface to be used.
4. In the table of accessible subscribers, select the subscriber from which you want to load the project data.
5. Click on "Load".

- **Load project data from a device:**

  To load only the project data from a device into your project, please follow these steps:

  1. Establish an online connection with the device from which you want to load the project data.
  2. Select the device in the project browser.
  3. Choose the "Load from device (software)" command in the "Online" menu. The "Preview for device loading" dialog box opens.
  4. Check the messages in the "Preview for device loading" dialog box.
  5. Click the "Load from device" button (Once loading is possible, the "Load from device" button becomes active). The loading process will then execute.

3.5.3 Project data protection concept

You have the option to protect your project data against unauthorized access. This includes, for example:

- Access protection for devices.
- Protection against copying and displaying objects. Limitations for the output of objects with Know-How protection.

3.6 Using reference projects

You have the option to open projects as references in addition to the current project. You can use these reference projects as follows:

- You can drag individual objects from a reference project into the current project and edit them there.
- You can open certain objects, such as code blocks, from a reference project in read-only mode. However, this is not possible for all elements.
- You can compare devices from the reference project with devices from the current project using an offline/offline comparison.

  To open a reference project, follow these steps:

  1. Click on "Open reference project" in the toolbar of the "Reference Projects" palette in the project browser. The "Open reference project" dialog box opens.
  2. Navigate to the desired project folder and open the project file.
  3. Click on "Open". The selected project opens in read-only mode as a reference project.

3.7 Simulate devices

TIA Portal allows you to simulate and test the hardware and software of the project in a simulated environment. The simulation runs directly on the PG/PC, so no additional hardware is required. Follow these steps to start the simulation function:

1. Select the device you want to simulate, for example, in the project browser.
2. Choose the "Simulation > Start" command from the "Online" menu.

3.8 Editing devices and networks

- The hardware and network editor is the integrated development environment for configuring, parameterizing, and networking devices and modules. In the project navigation, double-click on the "Devices and Networks" command to open the hardware and network editor.

- The hardware and network editor offers three different views of your project. You can switch between these three views at any time, depending on whether you want to create and edit individual devices and modules, complete device configurations and networks, or the topological structure of your project.

- You select the required devices and modules for your automation system from the hardware catalog and insert them into the device, network, or topological view.

- The inspection window contains information about the currently selected object. You can modify the settings of the selected object there.

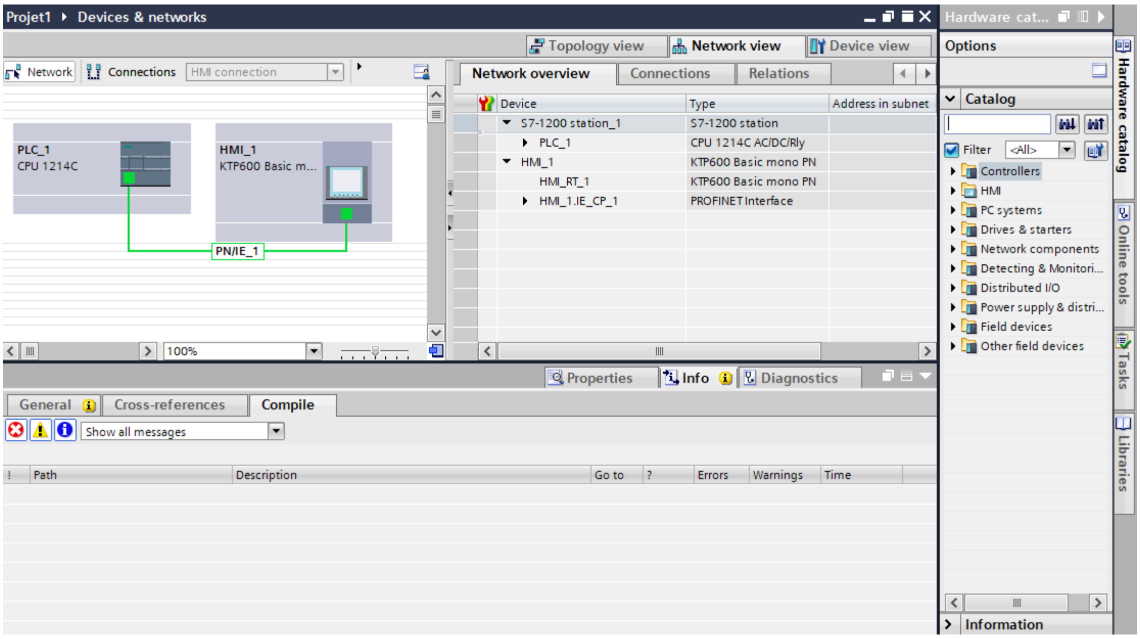


Figure 3.14: The devices and networks editor window

3.8.1 Device Configuration

To create an automation system, you need to configure, set up, and connect the various hardware components. When the automation system starts up, the CPU compares the intended software configuration with the actual configuration of the installation. This allows any potential errors to be detected and reported immediately.

- **Using Chassis:** To assign a module to a device, you need a chassis. You attach the modules to the chassis and connect them to the CPU. When you add a device in the network view, this automatically creates a station and an appropriate chassis for the selected device. In the device view, the chassis is displayed with available slots. The number of available slots depends on the type of the object.

- **Adding a Device to the Hardware Configuration:** To add a network-capable device to the hardware configuration, drag and drop from the hardware catalog into the network view or the topological view. There are other ways to add a device, such as:

  – Using the "Insertion == Device" command in the menu bar of the network view or the topological view.

  – Using the context menu of a device in the hardware catalog for "Copy" and "Paste".

  An appropriate chassis is created simultaneously with the new device. The selected device is plugged into the first available slot on the chassis.

  If you haven't selected a CPU yet but want to start programming or use an existing program, you can use an unspecified CPU. To create an unspecified CPU in the portal view, follow these steps: In the portal view, choose the "Devices and Networks" menu, then the action "Add Device." An unspecified CPU is created, and the device view of this CPU is displayed.

- **Module Arrangement on the Created Chassis:** Once you have added devices from the catalog into your configuration from the network view, you can proceed to insert modules into the devices. To plug a module into a chassis in the device view, you have the following options:

– For a valid empty slot, double-click on a module in the hardware catalog.

– Drag and drop from the hardware catalog onto a valid empty slot in the graphical or tabular area.

Once a module is plugged into a chassis with a CPU already plugged in, the address areas are automatically checked. You can copy (move, delete, and replace) hardware components in the device view.

- **Editing Properties and Parameters:** Once you have inserted the hardware components onto the chassis, you can edit the default properties in the hardware and network view, e.g., parameters and addresses.

  To modify the properties and parameters of the components, follow these steps:

  1. In the graphical representation, select the CPU, module, chassis, or interface to edit.

  2. Define the parameters of the selected object. In the inspection window, the various possible parameters are available under "Properties."

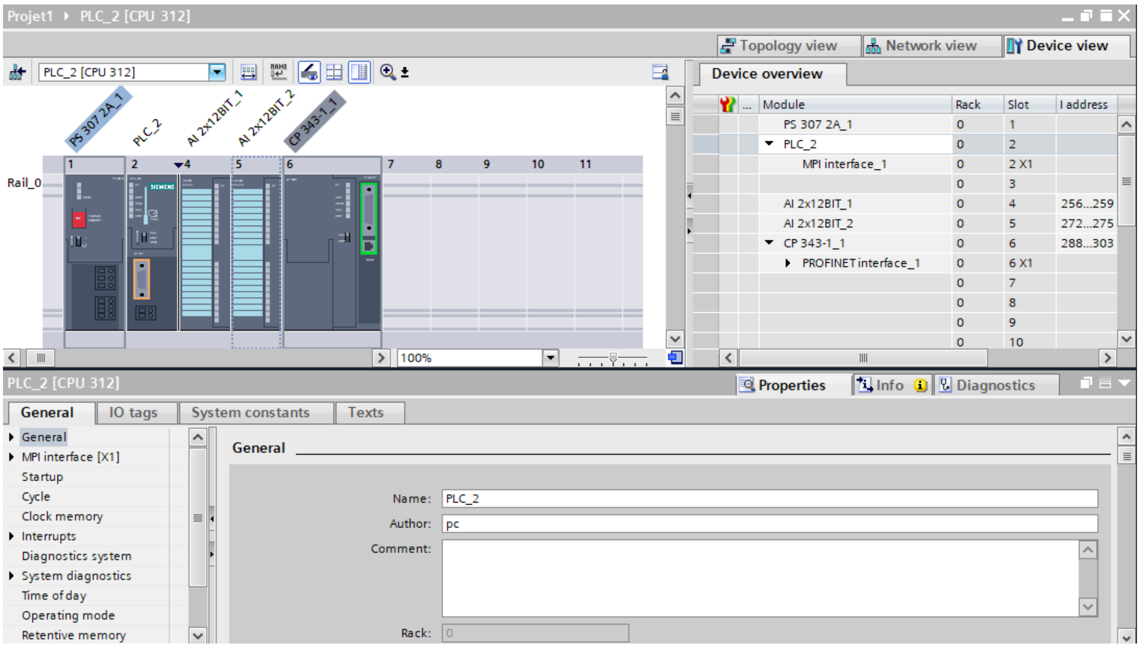


Figure 3.15: Editing properties in the device view

3.8.2 Network Configuration

In the project, you can create and network devices using communication-capable components. For networking devices, you have the following options:

- Connect the interfaces of communication-capable components together. This way, you create a new subnet suitable for the type of interface.
- Connect the interface of communication-capable devices to a new or existing subnet.
- Configure connections via graphical configuration of connections; missing connections are automatically detected and created, or through interactive dialogue.

Depending on the different tasks assigned to devices or due to the extension of the installation, it may be necessary to operate multiple subnets. These subnets are managed within a project.

3.8.3 Diagnose devices and networks

Diagnosing devices and networks is a very important part of editing projects in TIA Portal, as it provides an overview of the automation system's status. You can diagnose hardware in the "Online and Diagnostics" view, in the "Online Tools" Task Card, or in the "Diagnostic Device Information" area of the inspection window.

3.8.4 Structure of the Online and Diagnostic View

The Online and Diagnostic view consists of two windows:

- The left window (black frame) displays a tree structure with folders and groups (when folders are opened).
- The right window (red frame) contains the details of the selected folder or group.

It includes the "Online Access" group and the "Diagnostic" and "Functions" folders:

- "Online Access" Group: Indicates whether there is currently an online connection to the corresponding target. You can establish or suspend the online connection.
- "Diagnostic" Folder: Contains several diagnostic groups for the selected module.
- "Functions" Folder: Contains several groups where you perform settings or commands for the selected module.

3.8.4.1 Detecting faulty devices among those online

Detecting faulty devices among those online: In the "Diagnostic == Device Information" area of the inspection window, you have an overview of faulty devices for which there is or has been an online connection.

The "Diagnostic == Device Information" area of the inspection window consists of elements (Header containing the number of faulty devices and Table with detailed information on each faulty device).

The table includes the following columns:

- Online Status: Contains the online status as a diagnostic icon and in words.
- Operational Status: Contains the operational status as an icon and in words.
- Device/Module: Name of the relevant device or module.
- Message: Explains the entry in the previous column.
- Details: The link opens the Online and Diagnostic view of the device or brings it to the foreground. If there is no longer an online connection, the link opens the dialog box to establish the connection.
- Help: The link provides additional information about the fault.

3.9 PLC Programming

3.9.1 User Program Creation

The operating system is a component of the CPU and is already installed in the CPU upon delivery. Each CPU contains an operating system that organizes all functions and processes of the CPU that are not related to a specific automation task. The tasks of the operating system include:

- Boot process (hot start)
- Updating the input and output image memory
- Calling the user program
- Alarm acquisition and calling of alarm OBs
- Error detection and processing
- Memory area management

The user program contains all the functions required for processing specific automation tasks. You create the user program and load it into the CPU. Functions of the user program include:

- Checking prerequisites for startup (hot start) using start-up OBs, end position limit switches, or active safety devices, for example.
- Processing process data, such as binary signal combination, reading and evaluating analog values, determining binary output signals, and outputting analog values.
- Responding to alarms, such as diagnostic alarm in case of lower limit exceeding of an analog extension module.
- Handling disturbances in the normal program execution.

Linear and Structured Programming:

You can solve small automation tasks by writing the entire user program linearly in a cyclic OB. This approach is recommended only for simple programs. Realizing and maintaining complex automation tasks are simpler if these tasks are divided into several smaller partial tasks that correspond to the technological functions of the automation process or can be used multiple times. In the user program, these partial tasks are represented by blocks. Each block constitutes an independent section of the user program.

Structuring the program offers the following advantages:

- Programming large programs is clearer.
- Some parts of the program can be standardized and used multiple times with changing parameters.
- Program organization is simplified.
- It is easier to modify the program.
- Program testing is simplified because it can be performed section by section.
- Commissioning is simplified.

Overview of block types:

1. **Organizational Blocks (OB):** These blocks serve as the interface between the operating system and the user program. They are called by the operating system and control operations such as system startup behavior, cyclic program processing, alarm-triggered program processing, and error handling. OBs can be programmed to determine the CPU's behavior, and different types of OBs are available depending on the CPU used.

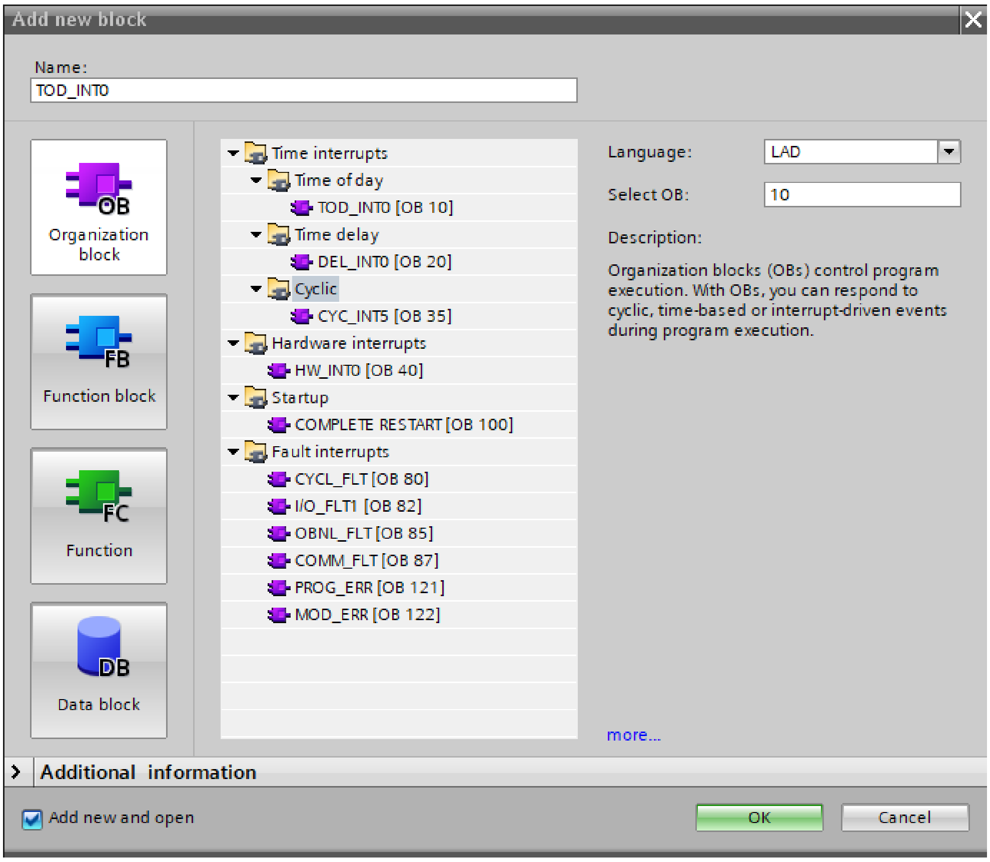


Figure 3.16: The different organizational blocks

2. **Functions (FC):** Functions are code blocks without memory. They do not have a data memory where parameter values can be stored. Therefore, effective parameters must be provided to all formal parameters when calling a function.

3. **Function Blocks (FB):** Function blocks are code blocks that persistently store their input, output, and input/output parameters in instance data blocks, allowing access to them even after block processing. That's why they are also called "Memory Blocks."

4. **Instance Data Blocks (DB) and Global Blocks:** Calling a function block is referred to as an "instance." For each instance of a function block, an instance data block is required, in which specific instance values for the formal parameters declared in the FB are stored. Global data blocks store data that can be used by all other blocks. The maximum size of global data blocks varies depending on the CPU, and you can define the structure of global data blocks as needed.

3.9.2 Creating and Managing PLC Variable (Tag) Tables

PLC variable (Tag) tables contain definitions of API variables and symbolic constants valid across the entire CPU. You can create multiple user-defined API variable tables in a CPU. Each variable table must have a unique name across the entire CPU. Follow these steps to create a new API variable table:

1. In the project navigation, open the "PLC Variables" folder located under the CPU.

2. Double-click on the "Insert New Variable Table" entry.

3. A new PLC variable table with the default name "Variable Table-X" is created.
4. In the project navigation, select the PLC variable table.
5. Choose the "Rename" command from the context menu.
6. Enter a unique name across the entire CPU.

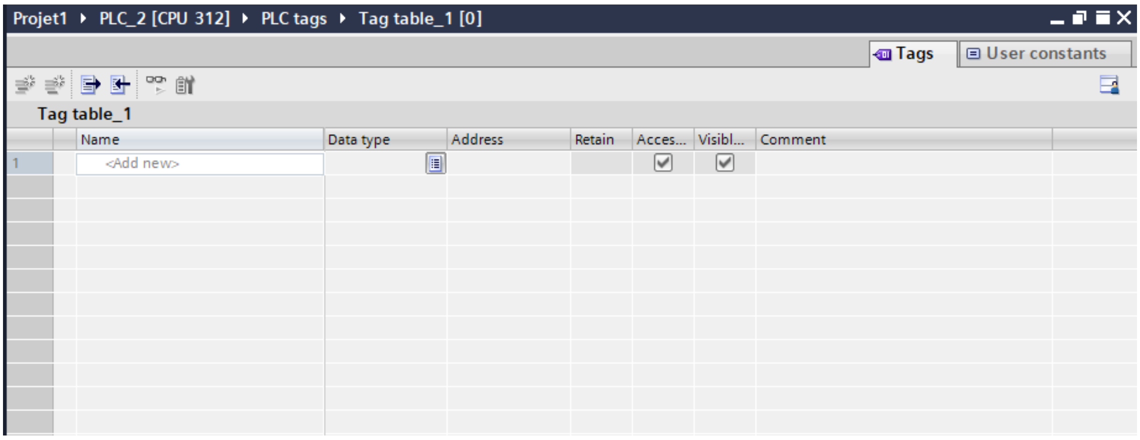


Figure 3.17: Creating PLC tags

3.9.3 Programming Blocks

The programming editor is the integrated development environment for programming functions, function blocks, and organization blocks. It provides optimal assistance when creating programs and searching for errors. The appearance and functionality of the programming editor may vary depending on the CPU used, the programming language, and the type of block being used.

Displaying Program Information:

The program information of a user program contains the following views:

1. **Assignment Table:** This table indicates which bits of the operands in the I, Q, and M memory areas are already assigned in the user program.

2. **Call Structure:** This view shows the call hierarchy of blocks within a user program and provides an overview of the blocks used and their dependencies.

3. **Dependency Structure:** This view lists the blocks used in the user program. The main block is listed at the top level, with calling or using blocks indented below it. Instance data blocks are listed separately from the call structure.

4. **Resources:** This view displays the utilization of hardware resources of the CPU for objects such as organization blocks (OB), function blocks (FC), function blocks (FB), data blocks (DB), user-defined data types, and API variables. It also includes information on the usage of CPU memory areas and present input/output modules.

   These program information views provide programmers with insights into the structure and resource utilization of their user program, facilitating debugging, optimization, and code maintenance.

3.10 Visualization of Processes

As the complexity of processes increases and machines and installations must meet ever stricter functionality specifications, operators need maximum transparency. This transparency is achieved through the human-machine interface (HMI).

3.10.1 Creating and Editing HMI Objects

First, select the supervisory device "HMI device." Follow these steps to add an HMI device:

1. In the project browser, double-click on "Add Device," and the "Add Device" window opens.
2. Select HMI or PC systems, then browse to the desired device, and select it.
3. Click "OK," and the desired device is added.

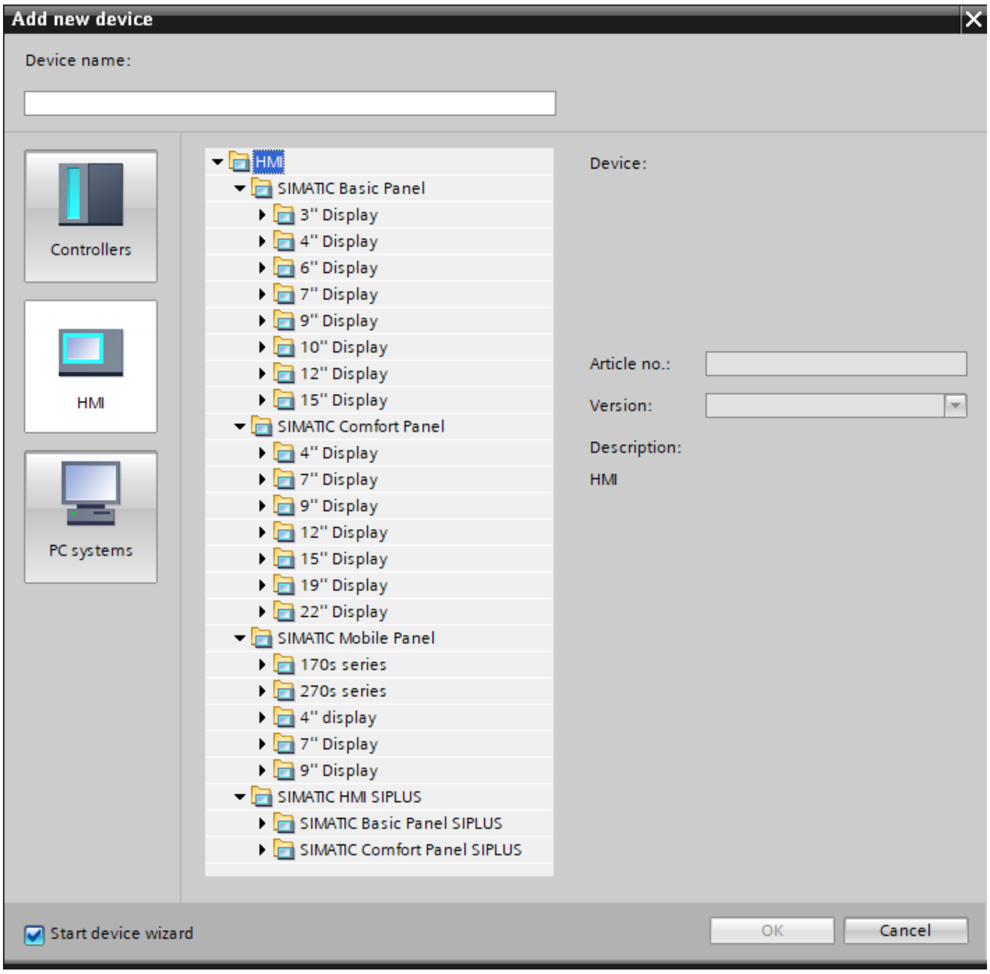


Figure 3.18: Adding a new device HMI

3.10.2 Creating Views

In WinCC, you create views for the control/command of machines and installations. To create views, you have predefined objects to represent your installation, display procedures, and define process values. To add a new view, follow these steps:

1. In the project browser, double-click on "Views > Add New View." The view is created in the project and displayed in the workspace. The view properties are displayed in the inspection window.

2. Enter a descriptive name for the view.

3. Configure the view properties in the inspection window. The view is created in your project. In the following steps, you can insert objects and control elements from the "Tools" task card and program function keys.

The initial view is the first view displayed at runtime when the project starts. You can define a specific initial view for each operator panel. Users call other views from the initial view. In the project browser, double-click on "Runtime Settings > General." Select the desired view as the "Initial View." Configure a model of objects that will be represented in all views based on this model.

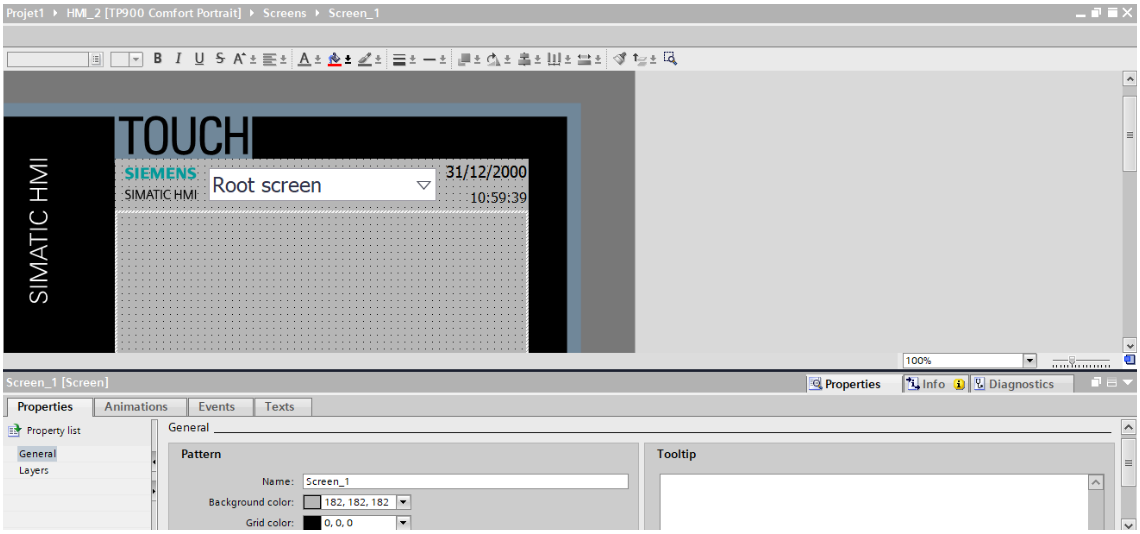


Figure 3.19: Adding a new View

3.10.3 Utilization of Objects

Objects are graphical elements used to create views in your project. The "Tools" task card contains all the available objects for the operator panel. Depending on the currently open editor, the tools window contains different palettes:

– **Simple Tools:** Simple objects are fundamental graphical objects, such as "Line," "Circle," "Text Field," or "Chart View," etc.

– **Elements:** Elements are fundamental control elements, such as "I/O Field," "Button," or "Needle Instrument."

– **Controls:** Controls have extended functions. They also represent process flows dynamically, such as the curve view and the recipe view.

– **Graphics:** Graphics are grouped by themes in a folder hierarchy. The different folders contain graphical representations such as machine and installation parts, measuring devices, control elements, etc.

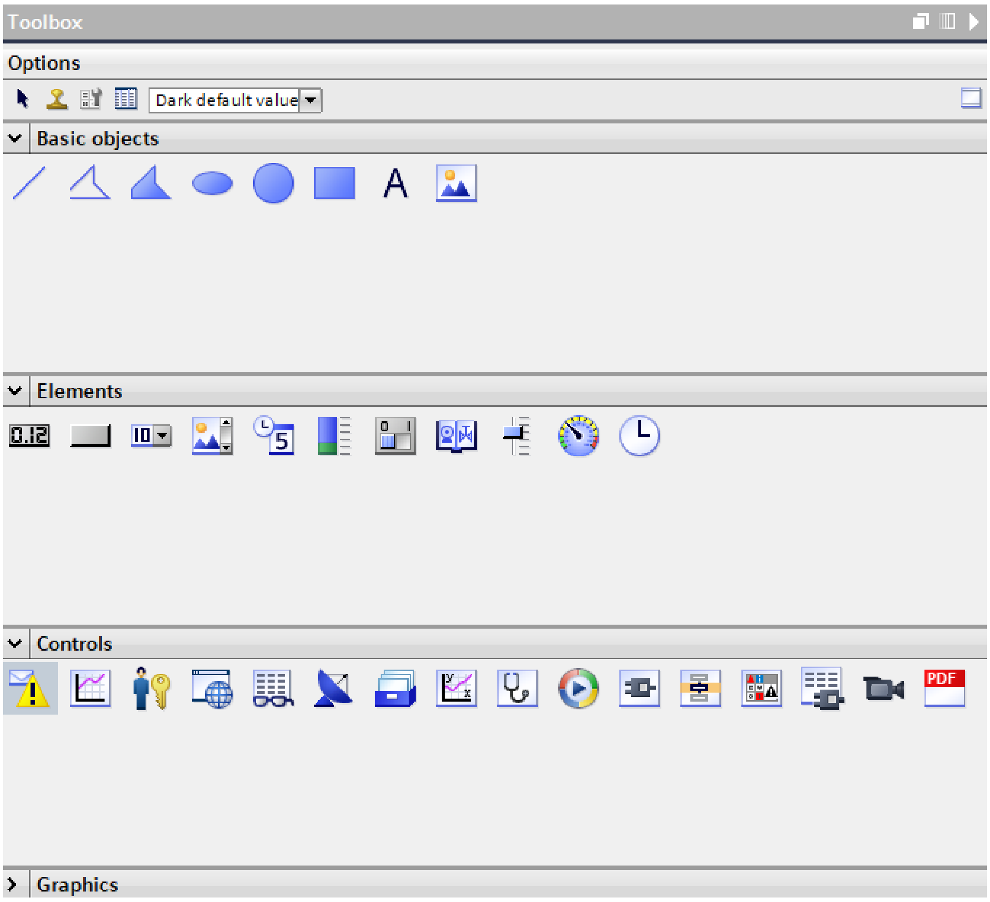


Figure 3.20: Toolbox

- **Libraries Task Card:** In addition to display and control objects, you also have library objects available. They are found in the palettes of the "Libraries" task card. A library contains pre-configured objects such as pipe graphics, pumps, pre-configured buttons.

To insert an object into a view, drag the desired object from the "Tools" task card to that view, and then configure it based on the object's properties in the inspection window.

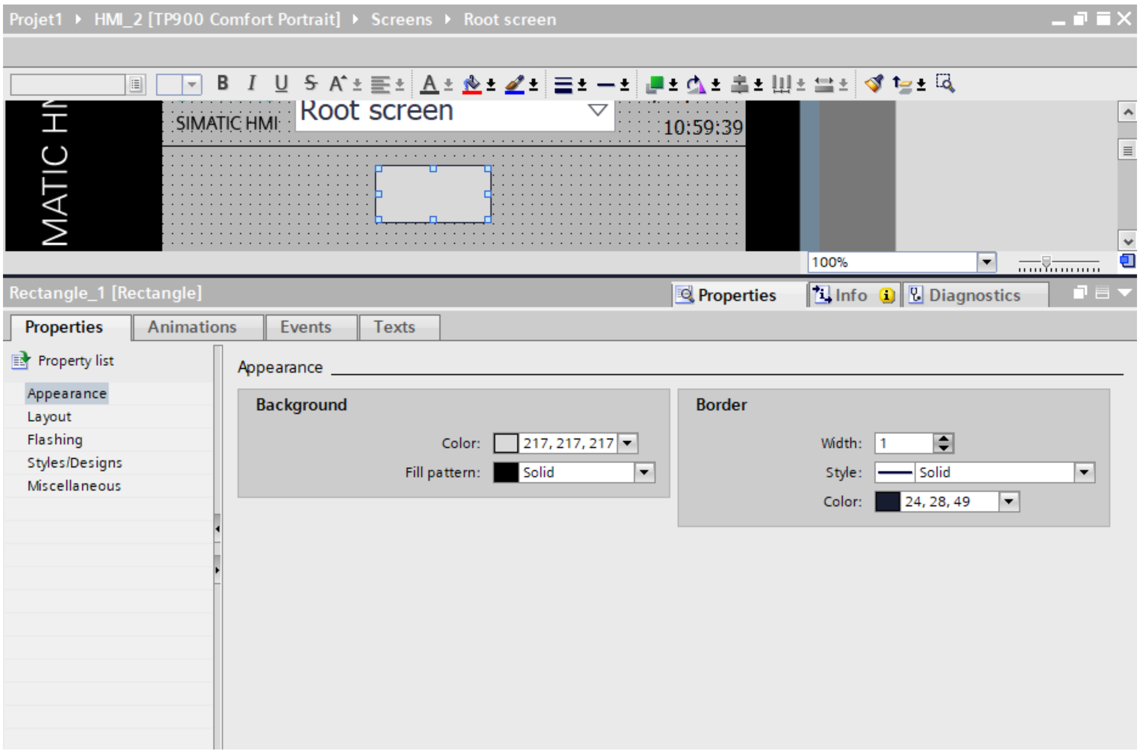


Figure 3.21: Insert an object into a view

3.10.4 Dynamizing Views

In Win CC, you animate objects to visualize your installation on operator panels and display the processing steps of a process. Perform dynamic animations using (Animations, Variables, System Functions). For example, consider an image of a tank where the liquid level increases or decreases according to a process value.

3.10.5 Use of Layers

1. Creating Layers:

   Open the graphical editor (e.g., HMI editor). Go to the layer management tool, usually found in the toolbar or the properties panel. Create new layers and give them meaningful names to represent their purpose.

2. Assigning Objects to Layers:

   Select the objects you want to assign to a layer. In the properties panel, assign the selected objects to the desired layer.

3. Managing Layers:

   Use the layer management tool to toggle visibility, lock/unlock layers, and change the order of layers. Layers can be reordered to bring certain elements to the foreground or send them to the background.

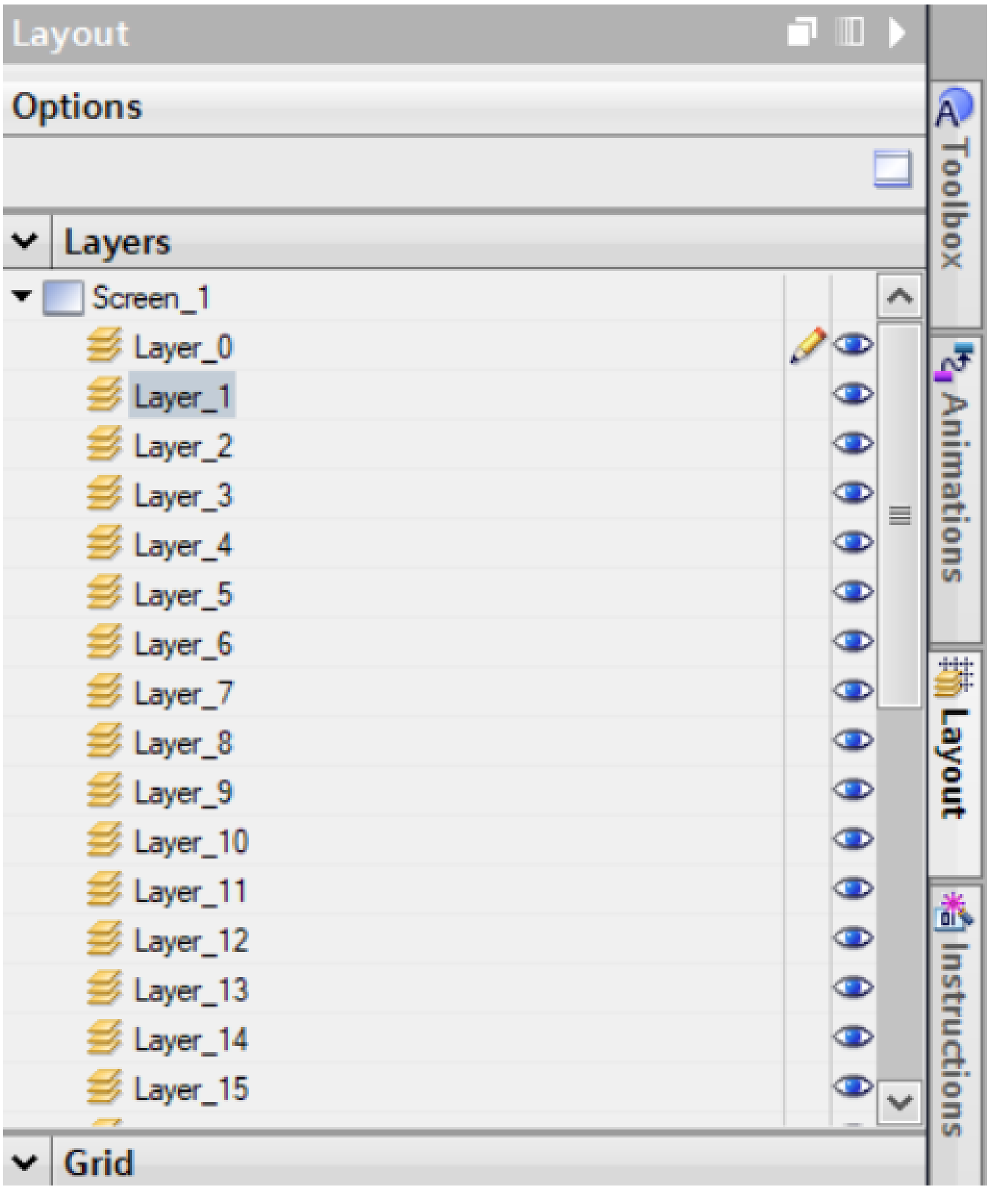


Figure 3.22: Layers List

3.10.6 Use of Variables

In Runtime, process values are transmitted through variables. Process values are data stored in the memory of one of the connected controllers. In the project browser, each HMI device has a "HMI Variables" folder. Win CC uses two types of variables:

– External variables link Win CC to automation systems. The values of external variables correspond to process values in the memory of an automation system.

– Internal variables are not linked to the process and only transmit values within Win CC. Variable values are only available while Runtime is running.

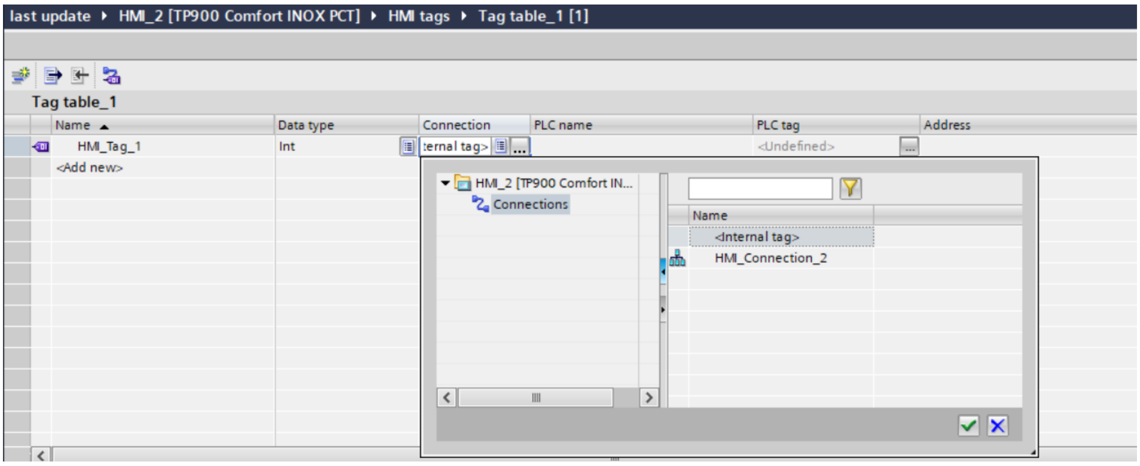


Figure 3.23: HMI Tags

3.10.7 Using data

Win CC provides the following archive types for archiving process data for HMI Runtime:

– A Data logs is used to archive process data from an industrial installation.

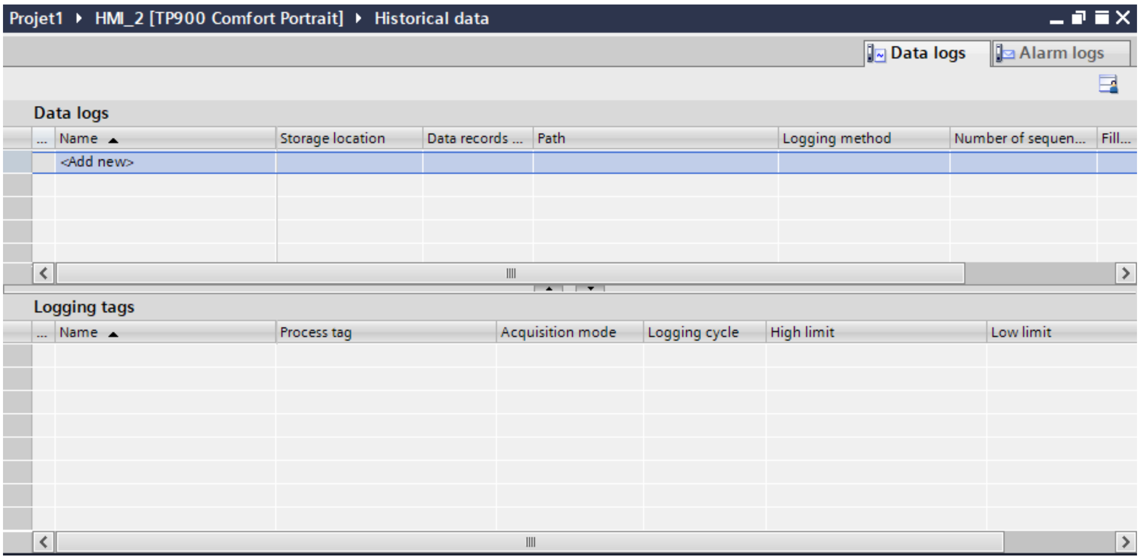


Figure 3.24: Data logs

– An alarm logs is used to archive alarms that occur in the visualized process.

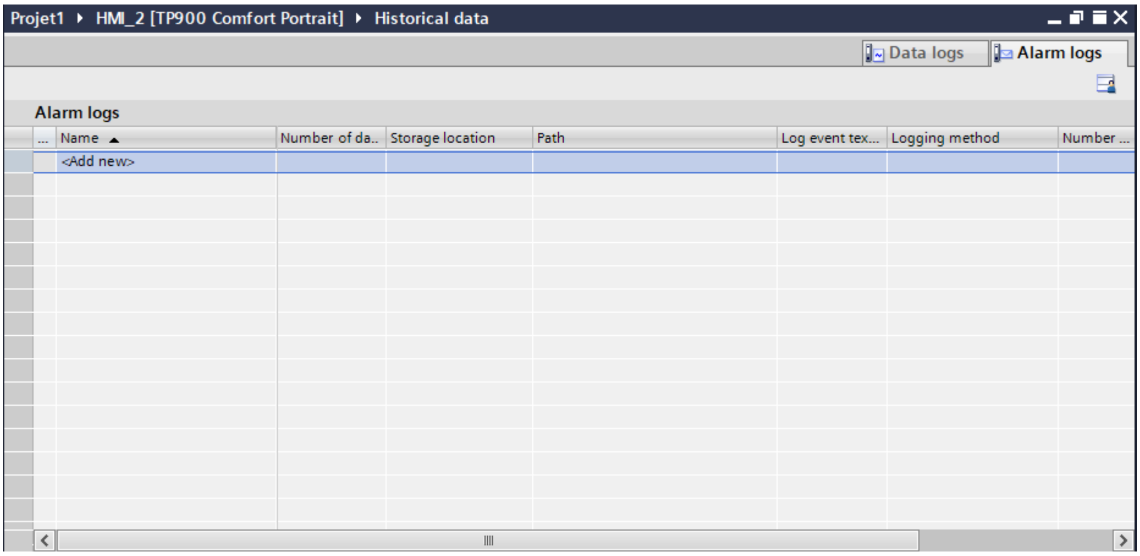


Figure 3.25: Alarm logs

3.10.8 Working with Alarms

1. Creating Alarms:
   - Access the Alarm Configuration: Go to the PLC program or HMI device where you want to configure alarms. For PLC alarms, navigate to the "PLC tags" or "Program blocks" section. For HMI alarms, go to the HMI configuration interface.

   - Define Alarm Tags: In the PLC, create tags that will trigger alarms. These tags can be linked to specific conditions or events in your PLC logic.

   - Configure Alarm Classes: Define different classes of alarms based on their severity (e.g., critical, warning, information). This helps in prioritizing and handling alarms appropriately.

2. Setting Up PLC Alarms:
   - Program Logic for Alarms: In your PLC program, add logic to monitor conditions that should trigger alarms. When a condition is met, the corresponding alarm tag should be set.

   - Alarm Acknowledgement: Configure how alarms should be acknowledged. This can include automatic acknowledgment or requiring operator action to acknowledge and reset alarms.

3. Setting Up HMI Alarms:
   - Alarm Control: In the HMI, add an alarm control object to your screens. This object will display alarms to the operator.

   - Alarm Display: Configure how alarms are displayed on the HMI. You can choose to show active alarms, historical alarms, or both.

   - Alarm Acknowledgement: Set up how alarms are acknowledged on the HMI. This can include buttons for acknowledging and resetting alarms.

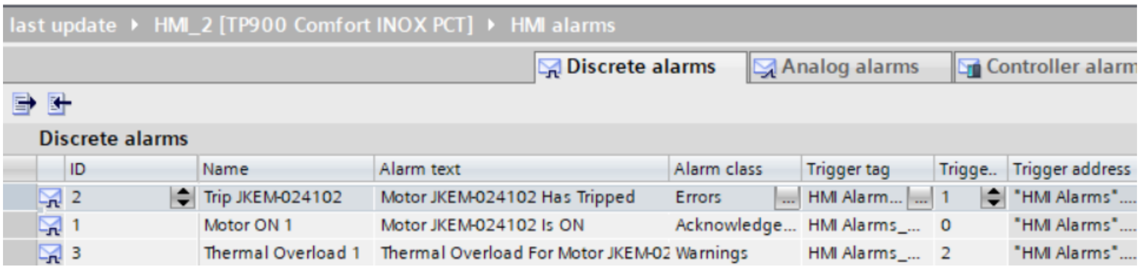


Figure 3.26: HMI Alarms Configuration

3.10.9 Using Recipes

Recipes group together data of the same type, such as machine settings or production data. Examples include:

- Machine settings needed to adapt production to another product variant.
- Components that yield different end products based on the proportions of their combination. Recipes are stored on the operator panel.

A recipe record is typically transferred completely in a single step between the operator panel and the controller.

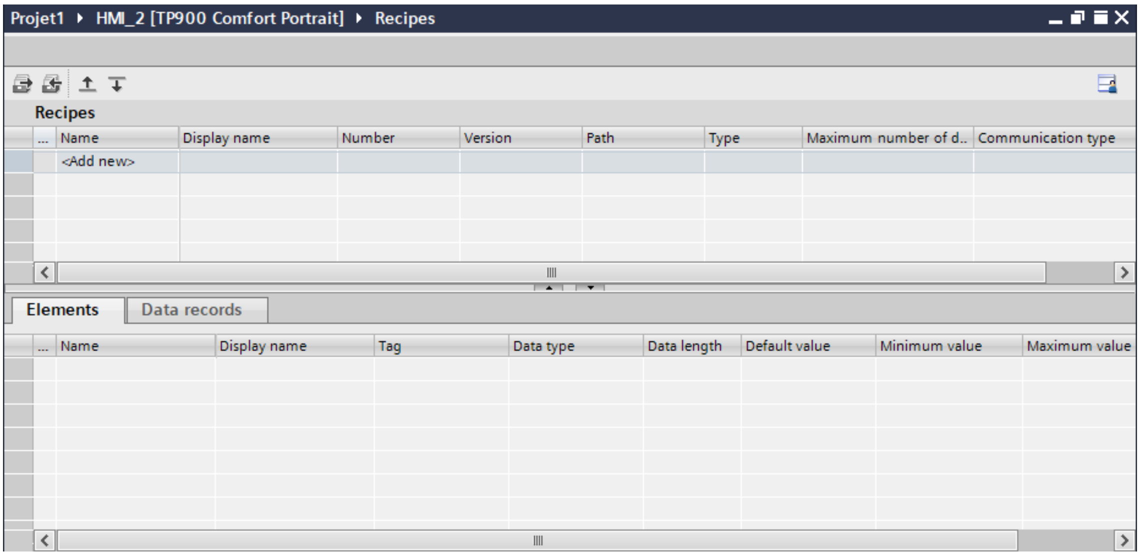


Figure 3.27: The recipe creation window

The simple recipe view is a predefined display and control object used to manage recipe records. It presents records in a tabular form.

3.10.9.1 Creating a Recipe

To create a recipe, follow these steps:

1. In the project browser under the HMI object, double-click on "Recipes." The "Recipes" editor opens.
2. In the "Recipes" editor, click "Add" in the first empty row of the table. The new recipe is created and displayed in a row.

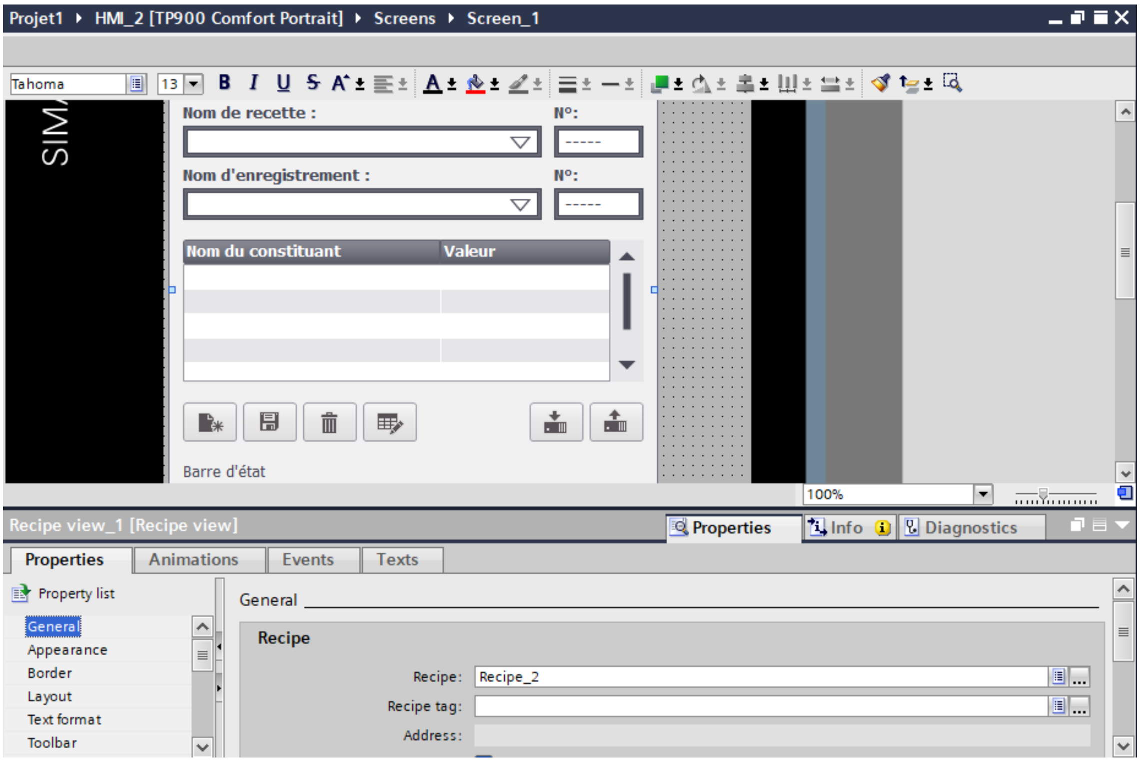


Figure 3.28: The recipe view object in the supervision view

3. In the "General" section, enter a descriptive name for the recipe under "Name." This name uniquely identifies the recipe on an operator panel.

4. Under "Display Name," enter the name that will be displayed in the recipe view according to the language.

5. Select a recipe number under "Number." This number uniquely identifies the recipe on the operator panel.The recipe is automatically provided with a version that indicates the date and time of the last modification. Another option is to enter information about the recipe as desired here.

6. Define the data storage location for recipe records under "Data Storage." The available options depend on the operator panel used.

7. Enter a tooltip that will be displayed to the operator in Runtime.

8. To grant access to recipe variables in Runtime that are configured in I/O fields, enable "Synchronize Recipe View and Recipe Variables" under "Attributes > Synchronization" in the inspection window.

9. To automatically transfer recipe variables to the controller when edited in I/O fields, disable "Manual Transfer of Modified Individual Values (Learning Mode)."

10. To monitor the transfer of recipe data using a zone pointer in Runtime, enable "Coordinated Transfer of Records."

11. Select the appropriate connection to the controller for this coordinated transfer under "Synchronize with."

3.10.10 Compiling and Loading Projects

Even as you configure a project in Win CC, the project is continuously compiled in the background. This helps reduce the duration of the final compilation. When you initiate compilation, an executable file for the relevant operator panel is created. Load the compiled project onto the operator panels where you want to use the project. If your project uses HMI variables associated with API variables, also compile all modified S7 blocks before loading them into the HMI operator panel with the "Compile == Software" command in the context menu.

3.11 Conclusion

In this chapter, we have provided a general description of the TIA Portal, where readers of our work can understand how TIA Portal is used. STEP7 TIA offers the possibility of programming and testing, such as visualizing the program to address any potential errors and making appropriate modifications before moving to implementation in the controller. WinCC TIA provides maximum transparency and is essential for operators working in an environment where processes are increasingly complex.

Chapter 4

Programming of the relay integration in the PLC

4.1 Introduction

This chapter focuses on the practical integration of relay control into a Programmable Logic Controller (PLC) using Siemens' TIA (Totally Integrated Automation) Portal software. It provides step-by-step instructions on configuring hardware, developing ladder logic diagrams, and deploying control logic to the PLC. Through practical examples and TIA Portal screenshots, readers will learn to program relay operations effectively, enhancing their ability to design robust and efficient automation systems.

4.2 Program elaboration

4.2.1 Connection between the CT & the plc input

To interface a CT output of 0-25 mA with a PLC input of 0-10 V, a 400 resistor can be used to convert the current to the desired voltage range, adhering to Ohm's Law (V = IR). This arrangement accurately represents the CT's current signal as a voltage signal suitable for the PLC input.

In the long term, even with numerous CTs, the resistor's energy dissipation remains approximately 5.475 kWh annually per motor. This energy loss is negligible even when considering a large number of motors and won't significantly affect the overall efficiency or budget of most projects.

4.2.2 PLC choice

In our case, we chose the SIEMENS S7-1200 1212 AC/DC/Rly CPU programmable logic controller, with reference of: 6ES7 212-1BE40-0XB0

4.2.2.1 Description of the PLC

Work memory 75 KB; 120/240VAC power supply with DI8 x 24VDC SINK/SOURCE, DQ6 x relay and AI2 on board; 4 high-speed counters (expandable with digital signal board) and 4 pulse outputs on board; signal board expands on-board I/O; up to 3 communication modules for serial communication; up to 2 signal modules for I/O expansion; 0.04 ms/1000 instructions; PROFINET interface for programming, HMI and PLC to PLC communication.

4.2.2.2 Reasons for choosing this PLC

– Economic reason (price)

– Work memory

– Number of DI/DQ

– CPU Scalability

Note: We chose to not use the DI/DQ of the CPU for the following reasons:

– DI cards can offer specialized features such as faster input response times, better isolation, and advanced filtering that might not be available with the built-in DI/DQ of the CPU.

– Adding a DI card increases the number of digital inputs available, allowing for more extensive and flexible connections to external devices and sensors.

– By offloading input processing to a dedicated DI card, the main CPU can focus on more critical tasks, improving overall system performance.

– DI cards often provide electrical isolation between the input signals and the CPU, protecting the main system from electrical noise, surges, and potential damage.

– Separate DI cards are designed specifically for handling digital inputs, making them more reliable and robust for specific tasks compared to the multifunction capabilities of built-in DI/DQ ports.

– Using separate DI cards allows for modular expansion of the system. Additional I/O can be added as needed without redesigning the entire CPU or system architecture.

4.2.3 Hardware configuration

Placement 2: DQ 16x24VDC_1
Placement 3: DI 16x24VDC_1
Placement 1: CPU 1212C AC/DC/Rly with a BB 1297 battery board referenced 6ES7 297-0AX30-0XA0

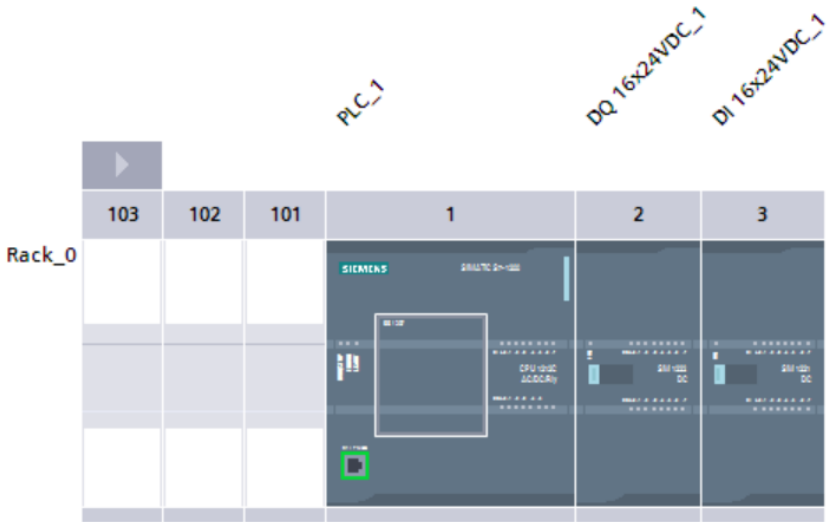
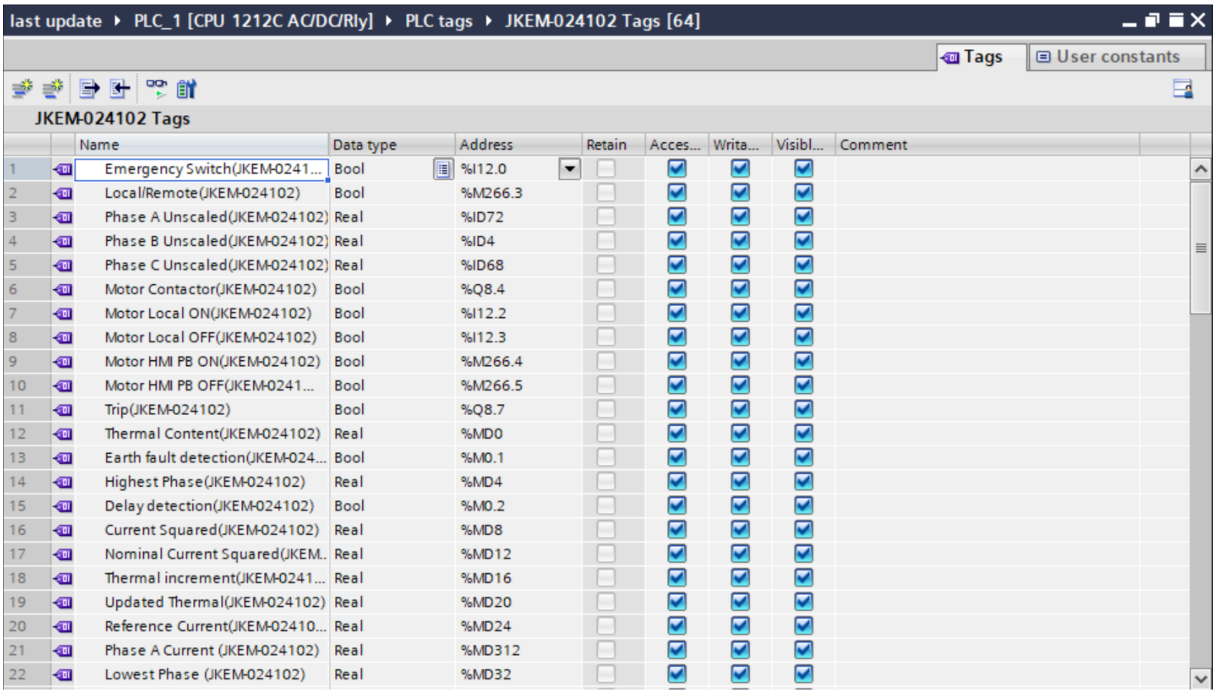


Figure 4.1: Hardware configuration

4.2.4 Creating PLC table tags

The PLC variable table allows for specifying the various variables, such as inputs, outputs, internal variables, timers, etc., that will be used in the program. It facilitates the organization and management of the variables by providing an overview of all the variables used in the project. One must define: the name of the variable, the data type whether it is BOOL, INT, REAL, etc., and its absolute address, for example %I12.0.

We created a table tag for each of our motor (In this case JKEM-024102)

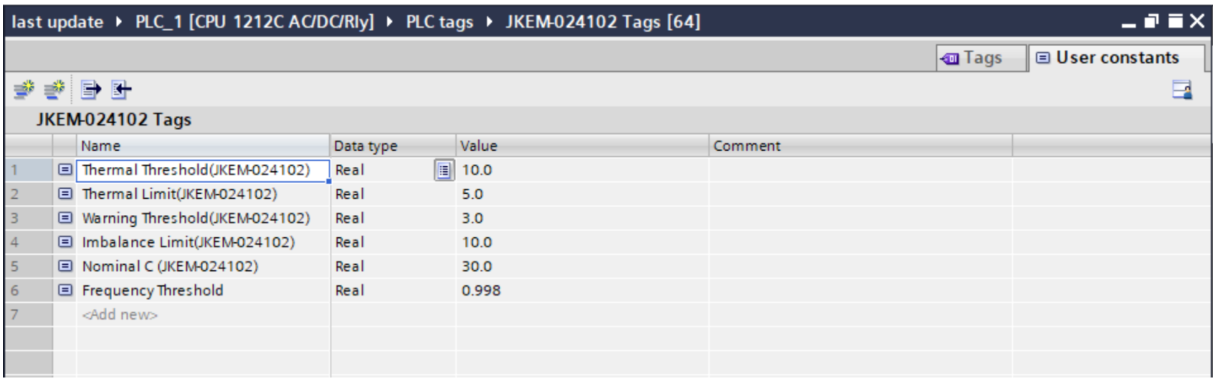


last update ▸ PLC_1 [CPU 1212C AC/DC/Rly] ▸ PLC tags ▸ JKEM-024102 Tags [64]

JKEM-024102 Tags

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Emergency Switch(JKEM-0241...	Bool	%I12.0	☐	☑	☑	☑	
2	Local/Remote(JKEM-024102)	Bool	%M266.3	☐	☑	☑	☑	
3	Phase A Unscaled(JKEM-024102)	Real	%ID72	☐	☑	☑	☑	
4	Phase B Unscaled(JKEM-024102)	Real	%ID4	☐	☑	☑	☑	
5	Phase C Unscaled(JKEM-024102)	Real	%ID68	☐	☑	☑	☑	
6	Motor Contactor(JKEM-024102)	Bool	%Q8.4	☐	☑	☑	☑	
7	Motor Local ON(JKEM-024102)	Bool	%I12.2	☐	☑	☑	☑	
8	Motor Local OFF(JKEM-024102)	Bool	%I12.3	☐	☑	☑	☑	
9	Motor HMI PB ON(JKEM-024102)	Bool	%M266.4	☐	☑	☑	☑	
10	Motor HMI PB OFF(JKEM-0241...	Bool	%M266.5	☐	☑	☑	☑	
11	Trip(JKEM-024102)	Bool	%Q8.7	☐	☑	☑	☑	
12	Thermal Content(JKEM-024102)	Real	%MD0	☐	☑	☑	☑	
13	Earth fault detection(JKEM-024...	Bool	%M0.1	☐	☑	☑	☑	
14	Highest Phase(JKEM-024102)	Real	%MD4	☐	☑	☑	☑	
15	Delay detection(JKEM-024102)	Bool	%M0.2	☐	☑	☑	☑	
16	Current Squared(JKEM-024102)	Real	%MD8	☐	☑	☑	☑	
17	Nominal Current Squared(JKEM..	Real	%MD12	☐	☑	☑	☑	
18	Thermal increment(JKEM-0241...	Real	%MD16	☐	☑	☑	☑	
19	Updated Thermal(JKEM-024102)	Real	%MD20	☐	☑	☑	☑	
20	Reference Current(JKEM-02410...	Real	%MD24	☐	☑	☑	☑	
21	Phase A Current (JKEM-024102)	Real	%MD312	☐	☑	☑	☑	
22	Lowest Phase (JKEM-024102)	Real	%MD32	☐	☑	☑	☑	

Figure 4.2: Table tags of JKEM-024102 motor



last update ▸ PLC_1 [CPU 1212C AC/DC/Rly] ▸ PLC tags ▸ JKEM-024102 Tags [64]

JKEM-024102 Tags

	Name	Data type	Value	Comment
1	Thermal Threshold(JKEM-024102)	Real	10.0	
2	Thermal Limit(JKEM-024102)	Real	5.0	
3	Warning Threshold(JKEM-024102)	Real	3.0	
4	Imbalance Limit(JKEM-024102)	Real	10.0	
5	Nominal C (JKEM-024102)	Real	30.0	
6	Frequency Threshold	Real	0.998	
7	<Add new>			

Figure 4.3: User constants of JKEM-024102

When using the SIMATIC S7-1200, an organization block called "OB1" is created by default when the CPU is added. This block serves as the interface for the CPU's operating system. The CPU automatically calls this block, which is processed cyclically.

the motors blocs are carried out by the bloc "OB1".

4.2.4.1 SCALE function

With the "Scaling" instruction, you convert the integer specified in the IN parameter into a floating-point number that is scaled to physical units between a lower limit value and an upper limit value. You define the lower and upper limit values of the range over which the input value is scaled using the LO_LIM and HI_LIM parameters. The result of the instruction is provided to the OUT parameter.

In our case we used 4 scale functions, three for the motor's 3 phased currents and one for the motor's speed

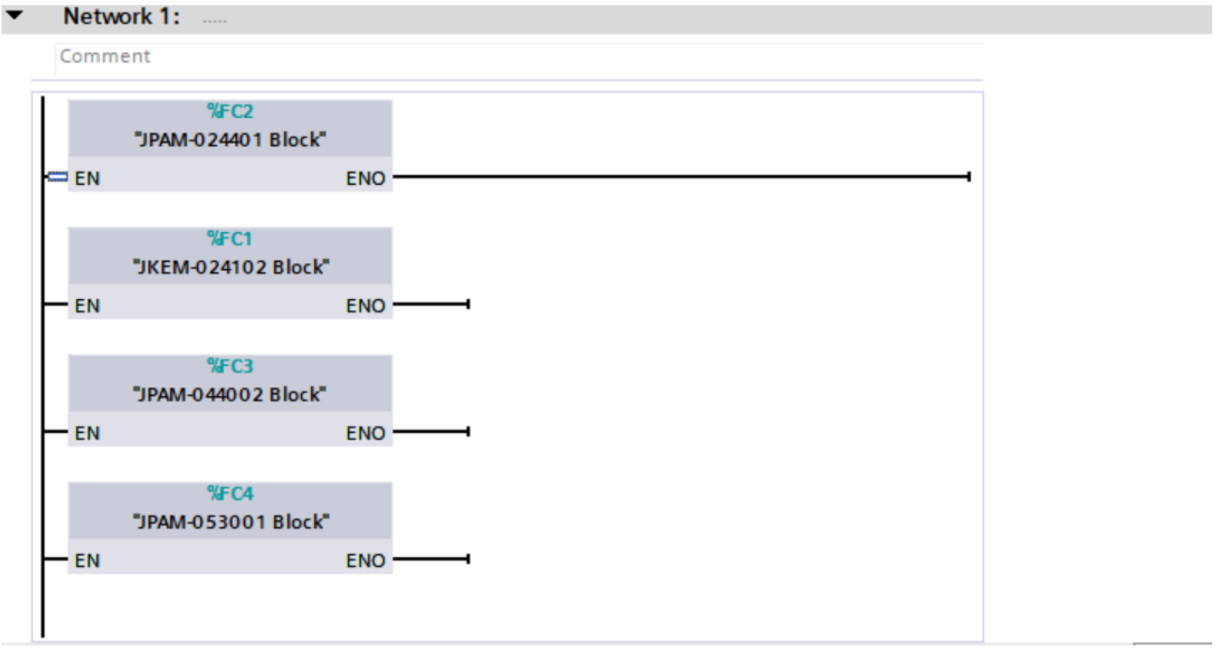


Figure 4.4: Main OB1

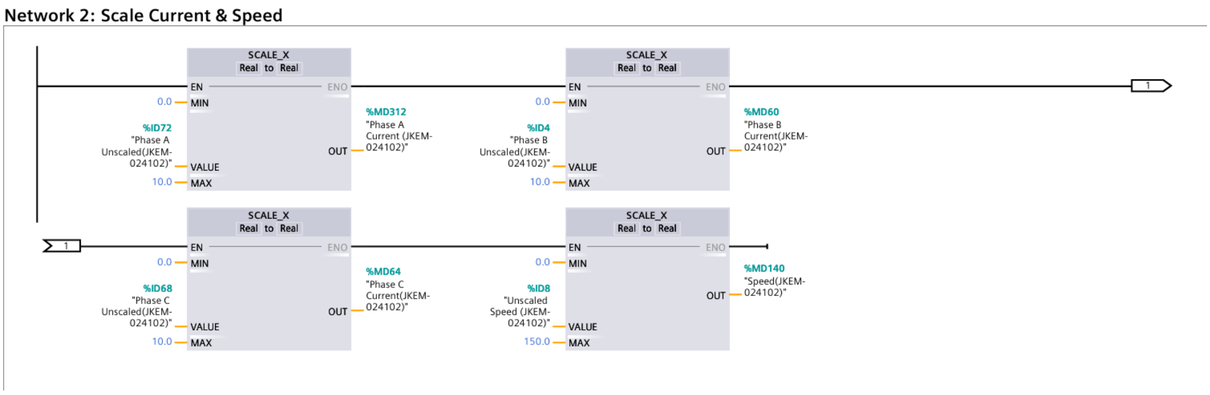


Figure 4.5: SCALE function

4.2.4.2 Local/remote select & start/stop

In TIA Portal, the remote/local select switch toggles between local and remote control modes:

- Local Mode: Manual control by operators at the device, used for maintenance and troubleshooting.

- Remote Mode: Automated control from a remote location, such as a control room or SCADA system.

The switch's state is read by the PLC program to determine the control source.This feature ensures flexibility and safety in operations.

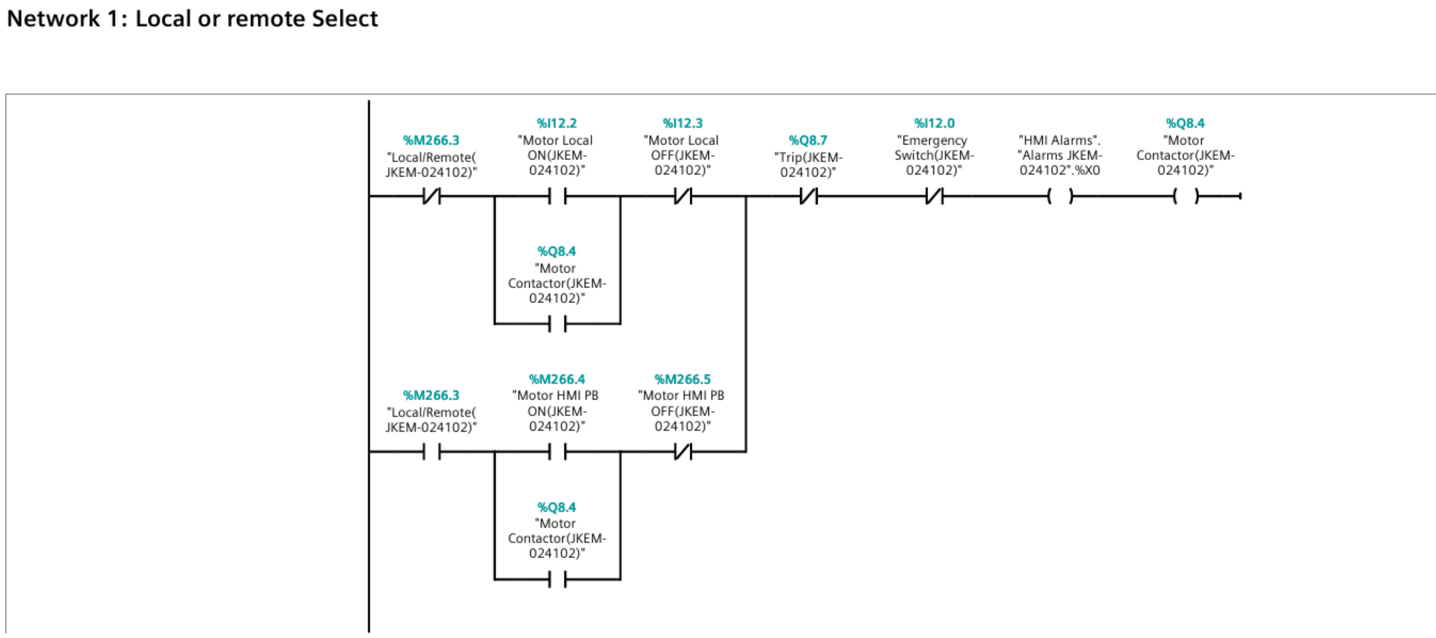


Figure 4.6: Remote/local select & Start/stop

4.2.4.3 Start Current Limit protection

This protection offers to trip the motor if the start-up current exceeds a preset value for a predefined time.
We use a function that compares the preset value (In the case of JKEM-024102 is 6FLC) with the motor's start-up current, if the latter exceeds that value for the timer's preset time value the plc will trip the motor.

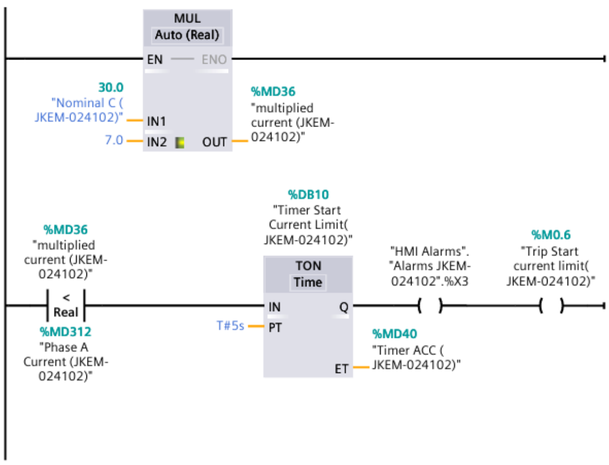


Figure 4.7: Start Current Limit protection

4.2.4.4 Overload protection

By the help of TIA Portal functions, thermal increment and thermal content are calculated to simulate the motor's thermal behavior using mathematical formulas.If the thermal content exceeds the thermal limit, an overload fault is set.

The thermal content is compared against a thermal limit to determine overheating.

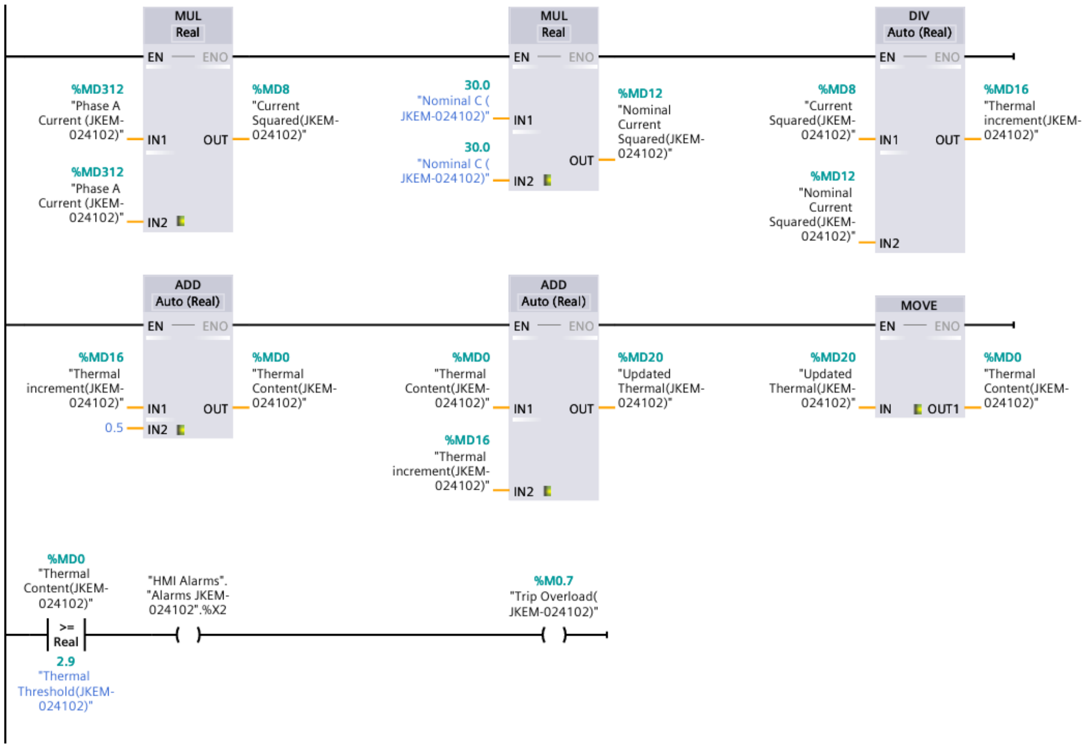


Figure 4.8: Overload protection

4.2.4.5 Imbalance protection

Network 5 is responsible for detecting and managing current imbalances between the phases of the motor. It measures the current in each phase (Phase A, Phase B, and Phase C) and determines the highest and lowest phase currents.

The network then calculates the imbalance by finding the difference between the highest and lowest phase currents and expresses this as a percentage of the highest phase current. If this imbalance exceeds a predefined threshold, it triggers a timer. If the imbalance persists beyond a specified duration, the network generates an imbalance trip signal.

This protective measure helps prevent damage to the motor caused by prolonged uneven distribution of current across its phases, ensuring the motor operates smoothly and reliably.

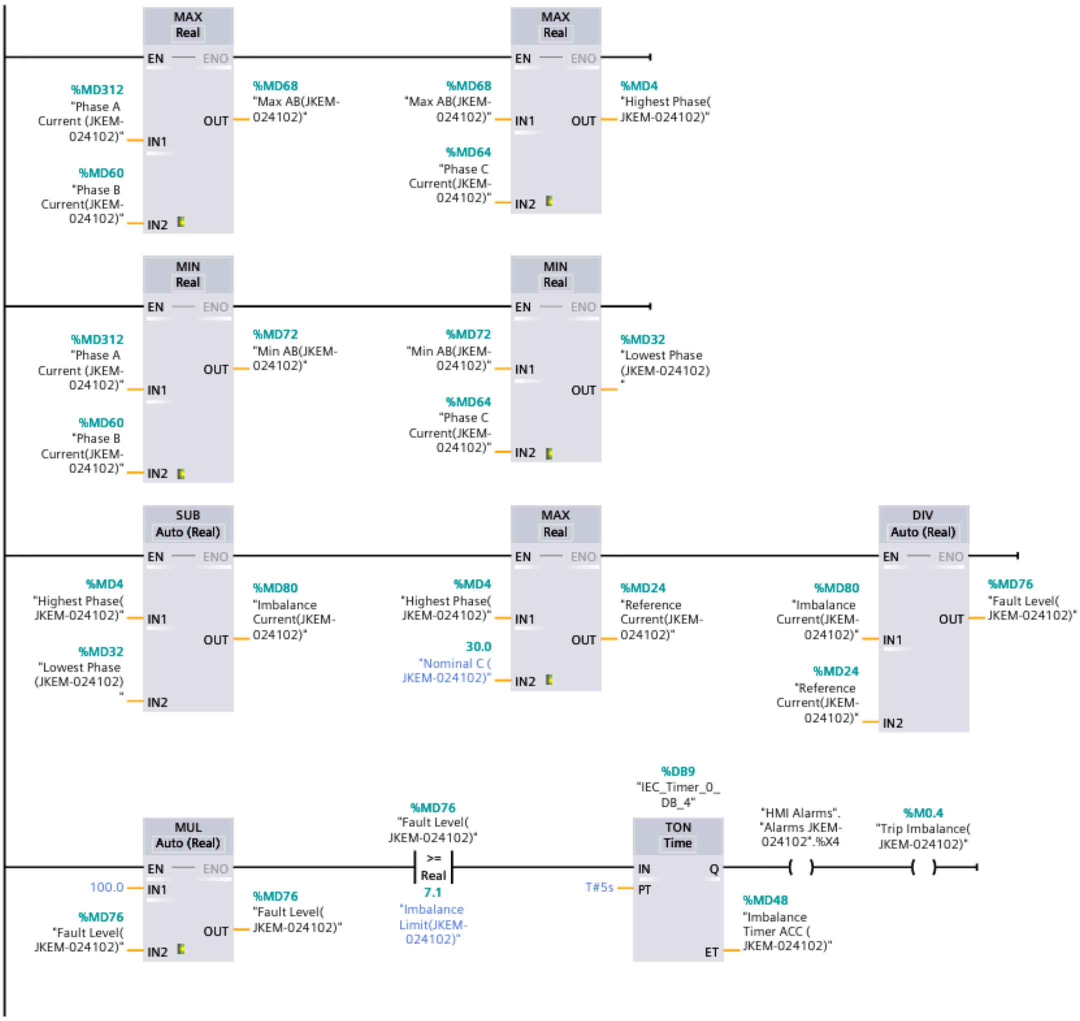


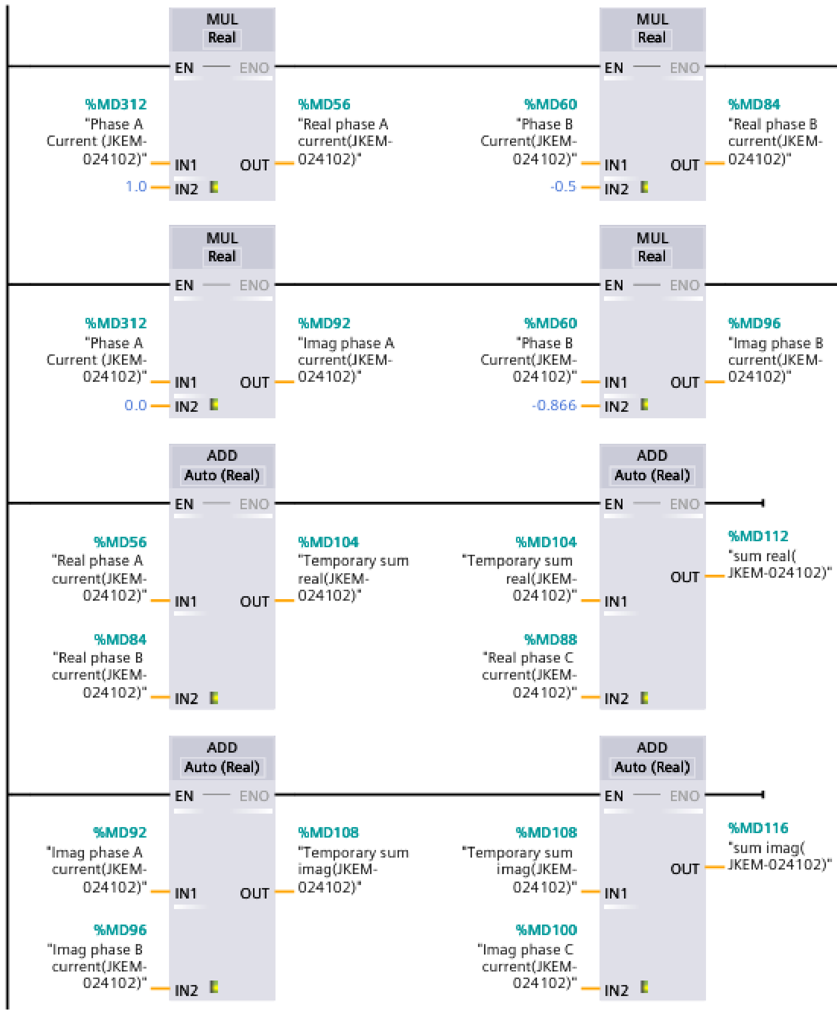
Figure 4.9: Imbalance protection

4.2.4.6 Earth fault protection

Network 6 in the ladder logic is responsible for Earth Fault Protection. This network monitors the current flowing through each phase of the motor (Phase A, Phase B, and Phase C) and calculates the resultant current vector.

It sums the real and imaginary components of the phase currents, squares these sums, and then computes the square root of the result to determine the magnitude of the fault current. If this calculated magnitude exceeds a predefined threshold, it signifies a potential earth fault. The network then initiates a timer, and if the fault condition persists for a specified duration, it triggers an earth fault trip signal.

This mechanism helps protect the motor from damage due to ground faults by promptly disconnecting the motor from the power source upon detection of such faults.



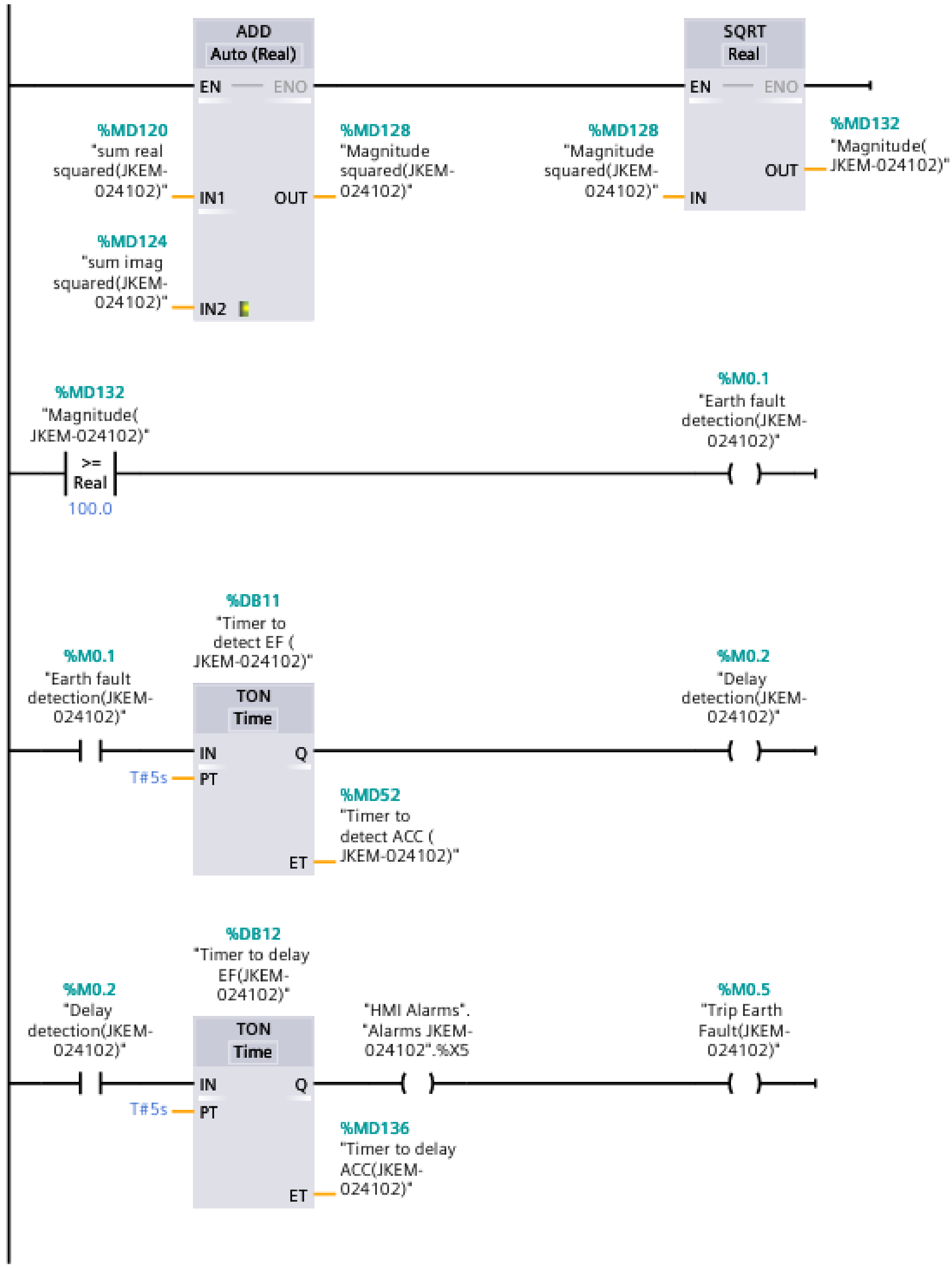


Figure 4.10: Earth fault protection

4.2.4.7 Under frequency protection

Network 6 is designed to monitor the motor speed and ensure it does not fall below a specified frequency threshold, which could indicate a malfunction or load issue.

The network reads the motor speed, calculates the corresponding frequency, and compares this frequency to a predefined threshold. If the motor speed drops such that the frequency falls below this threshold, the network generates an under-frequency trip signal.

This protective action helps prevent potential damage caused by operating the motor at speeds lower than its designed capability, thus maintaining the motor's efficiency and longevity.

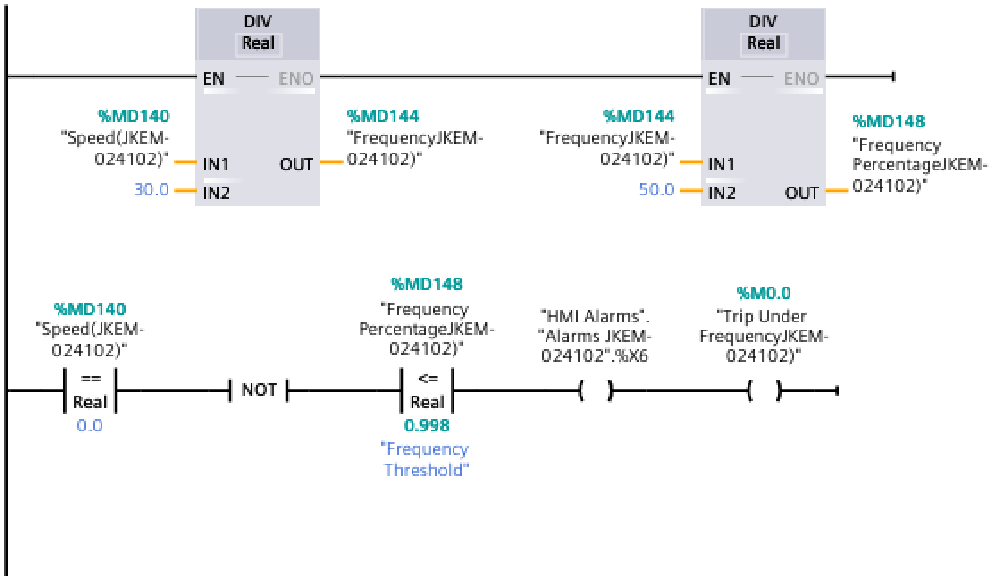


Figure 4.11: Under frequency protection

4.2.4.8 Trip & Reset

Networks 8 and 9 work together to manage the trip and reset functionalities of the motor protection system.

Network 8 consolidates various trip conditions, such as earth faults, overloads, imbalances, start current limits, and under-frequency conditions, into a single trip signal. When any of these conditions are detected, the main trip output (‘Trip JKEM-024102‘) is activated, disconnecting the motor from the power supply to prevent damage.

Network 9 handles the reset functionality, using a ‘MOVE‘ instruction to clear the trip signals and fault conditions. When the reset command (‘Reset JKEM-024102‘) is activated, it resets the trip signal and other fault-related parameters to zero, allowing the system to resume normal operation after resolving the fault conditions.

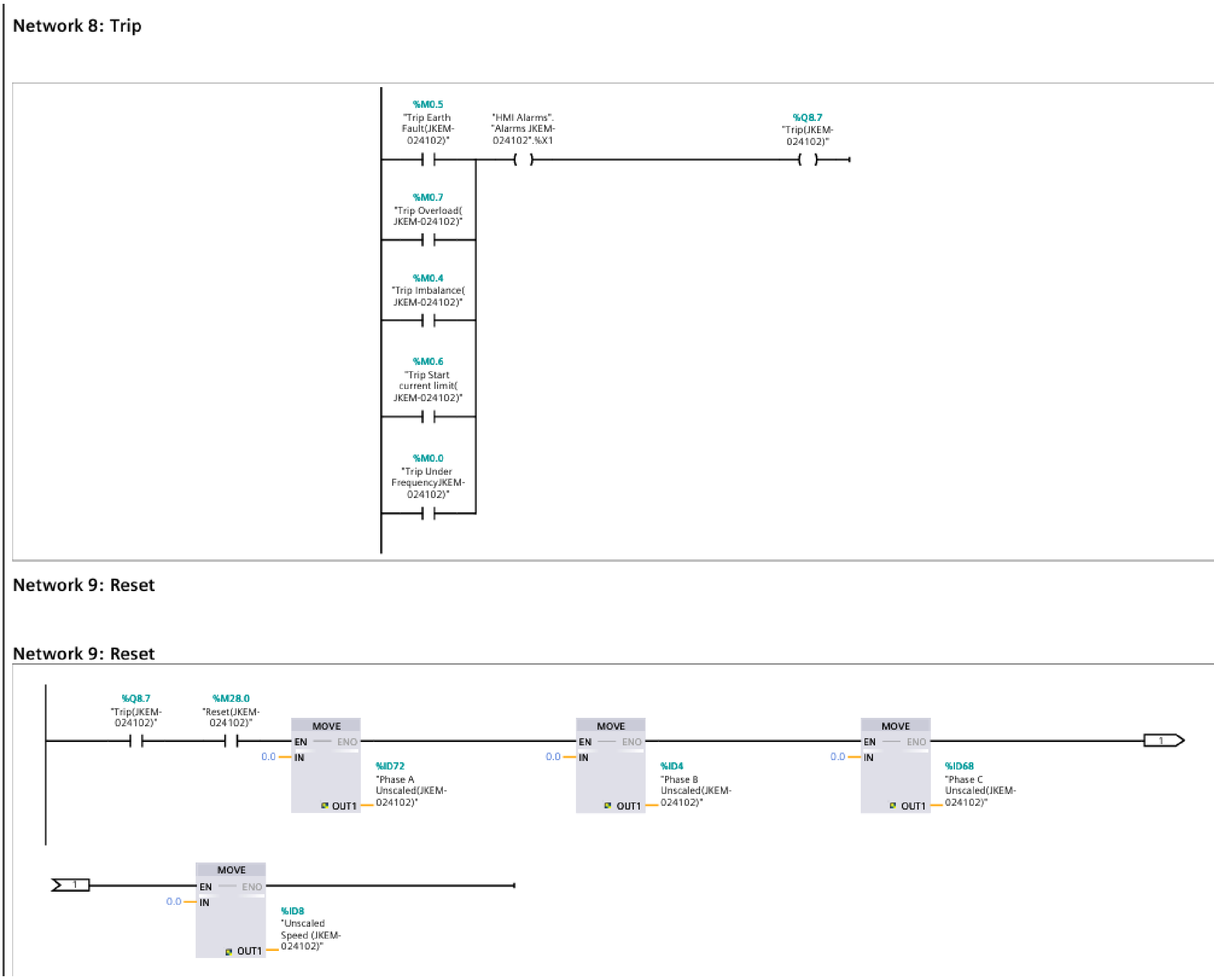


Figure 4.12: Trip & Reset

Note: We use the same ladder logic for all of our motors, the only difference is in the preset values of the current and the thermal variables of each motor.

4.2.5 Development of the station supervision

Thanks to the HMI, operators can monitor critical data and parameters in real-time, make quick decisions, and perform appropriate actions.
This optimizes the performance of the installation, improves operational efficiency, and reduces human errors. By providing a clear, user-friendly, and visually appealing interface, the HMI facilitates the understanding of complex processes and enables operators to quickly master the necessary tasks and controls.

4.2.6 Creating HMI tags

There are two types of variables commonly used in automated systems: external variables and internal variables.

- External variables: are used to facilitate communication and data exchange between different components of an automated system. They allow the transmission of information between an operator panel and a programmable logic controller (PLC).

- Internal variables, on the other hand, are used within the operator panel and are not linked to the PLC. They are used to store temporary information or internal states of the panel.

Default tag table

Name ▲	Data type	Connection	PLC name	PLC tag	Address
IEC_Timer_0_DB_1_ET	Time	HMI_Conne...	PLC_1	IEC_Timer_0_DB_1.ET	
Imbalance Timer ACC (JKEM-02...	Time	HMI_Connectio...	PLC_1	"Imbalance Timer ACC (J...	
Local/Remote(JKEM-024102)	Bool	HMI_Connectio...	PLC_1	"Local/Remote(JKEM-024...	
Local/Remote(JPAM-024401)	Bool	HMI_Connectio...	PLC_1	"Local/Remote(JPAM-024...	
Local/Remote(JPAM-044002)	Bool	HMI_Connectio...	PLC_1	"Local/Remote(JPAM-044...	
Local/Remote(JPAM-053001)	Bool	HMI_Connectio...	PLC_1	"Local/Remote(JPAM-053...	
Motor Contactor(JKEM-024102)	Bool	HMI_Connectio...	PLC_1	"Motor Contactor(JKEM-...	
Motor Contactor(JPAM-024401)	Bool	HMI_Connectio...	PLC_1	"Motor Contactor(JPAM-...	
Motor Contactor(JPAM-044002)	Bool	HMI_Connectio...	PLC_1	"Motor Contactor(JPAM-...	
Motor Contactor(JPAM-053001)	Bool	HMI_Connectio...	PLC_1	"Motor Contactor(JPAM-...	
Motor HMI PB OFF(JKEM-0241...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB OFF(JKEM...	
Motor HMI PB OFF(JPAM-0244...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB OFF(JPAM...	
Motor HMI PB OFF(JPAM-0440...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB OFF(JPAM...	
Motor HMI PB OFF(JPAM-0530...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB OFF(JPAM...	
Motor HMI PB ON(JKEM-024102)	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB ON(JKEM-...	
Motor HMI PB ON(JPAM-02440...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB ON(JPAM-...	
Motor HMI PB ON(JPAM-04400...	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB ON(JPAM-...	
Motor HMI PB ON(JPAM-053001)	Bool	HMI_Connectio...	PLC_1	"Motor HMI PB ON(JPAM-...	
Phase A Current (JKEM-024102)	Real	HMI_Connectio...	PLC_1	"Phase A Current (JKEM-0...	
Phase A Current (JPAM-024401)	Real	HMI_Connectio...	PLC_1	"Phase A Current (JPAM-0...	

Figure 4.13: HMI tags

4.2.7 Configuring the PC system-HMI interface

"In setting up the PC system WinCC-RT-Professional on the S7-1200 station, the folder 'PC-System [SIMATIC PC station]' is automatically generated in the project browser of TIA Portal. For configuring the HMI view, open the 'HMI_RT [WinCC RT Professional]' folder in the project browser, then navigate to the 'Views' folder and double-click 'Add View'. This will create an initial view (root view) that we rename as the home view. We plan the navigation within the main view and between different views (creating menus and toolbars to navigate between views).

We determine the structure representing the process by defining the necessary number of views in our case, 7 views then adjust the cycle time by accessing the inspection window and selecting 'Miscellaneous'. This allows us to set the appropriate cycle time; in our program, we set it to 250 ms (default is 2 s).

4.2.8 View description

4.2.8.1 Main screen view

This HMI main screen for a Siemens SIMATIC system, features a central navigation hub for selecting and monitoring motors in an industrial setting. It includes buttons for four motors (JKEM-024102, JPAM-024401, JPAM-044002, JPAM-053001), an "Alarms" button for system alerts, and a "Current visualization" button for viewing real-time data. The screen also displays logos for Sonatrach and École Nationale Polytechnique, indicating a collaborative effort. The footer contains navigation icons for home, information, and other functions, ensuring easy user interaction.

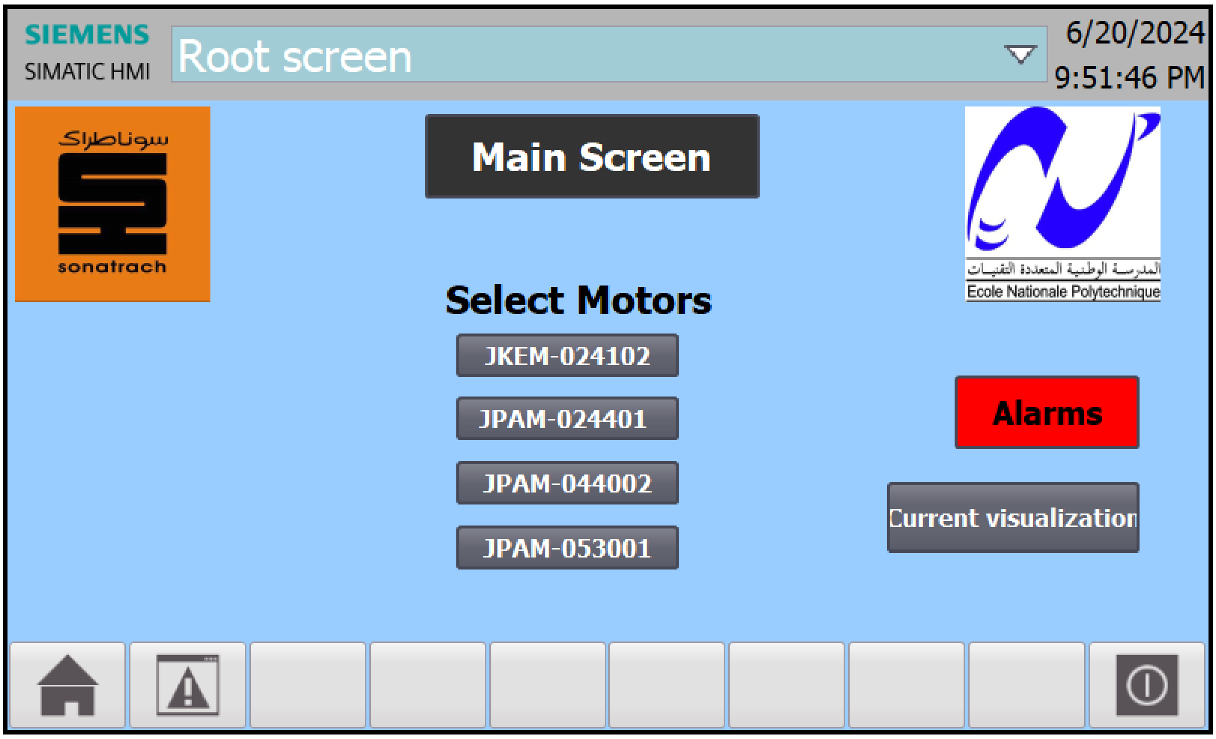


Figure 4.14: HMI main screen

4.2.8.2 Motor screen

The purpose of this HMI screen is to provide detailed control and monitoring for the specified motor (in this case JKEM-024102). It allows the user to switch between remote and local control, turn the motor on and off, reset the motor, and monitor real-time phase current values. Additionally, it displays various alarm conditions, such as trip motor, overload, start current limit, imbalance, earth fault, and under frequency warnings, helping operators quickly identify and respond to any issues. The screen ensures effective management and maintenance of the motor within the industrial system.

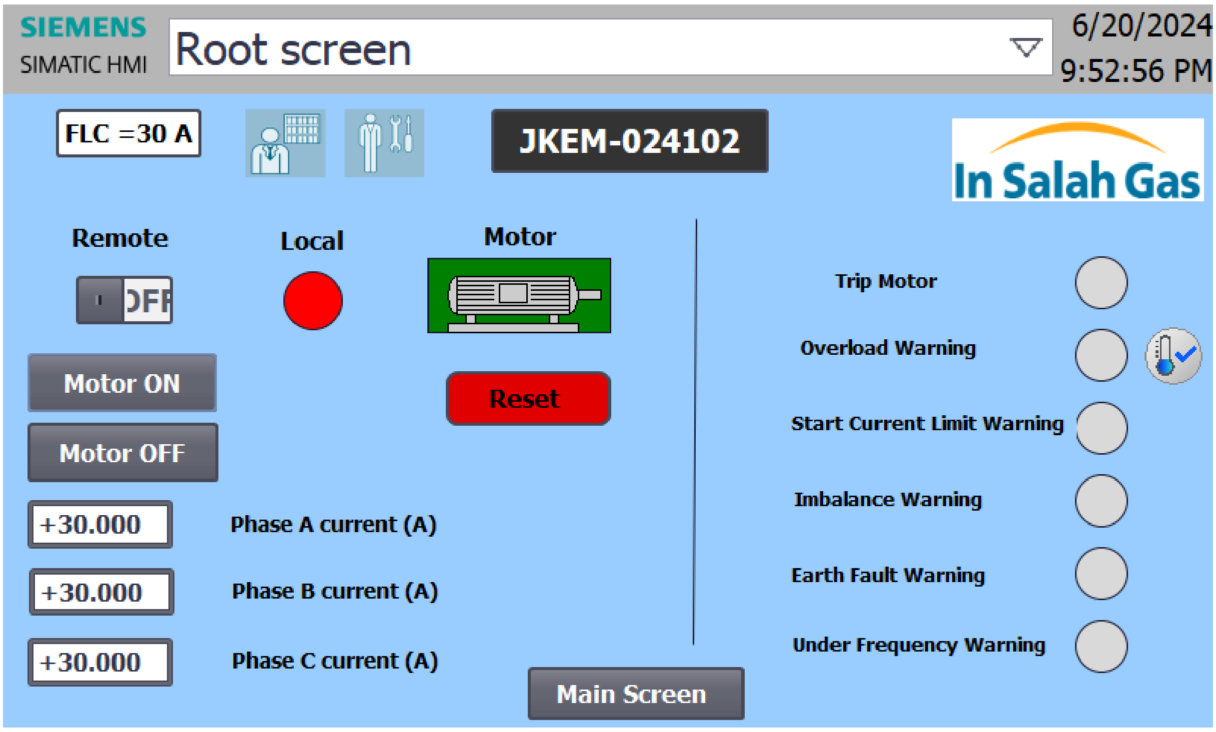
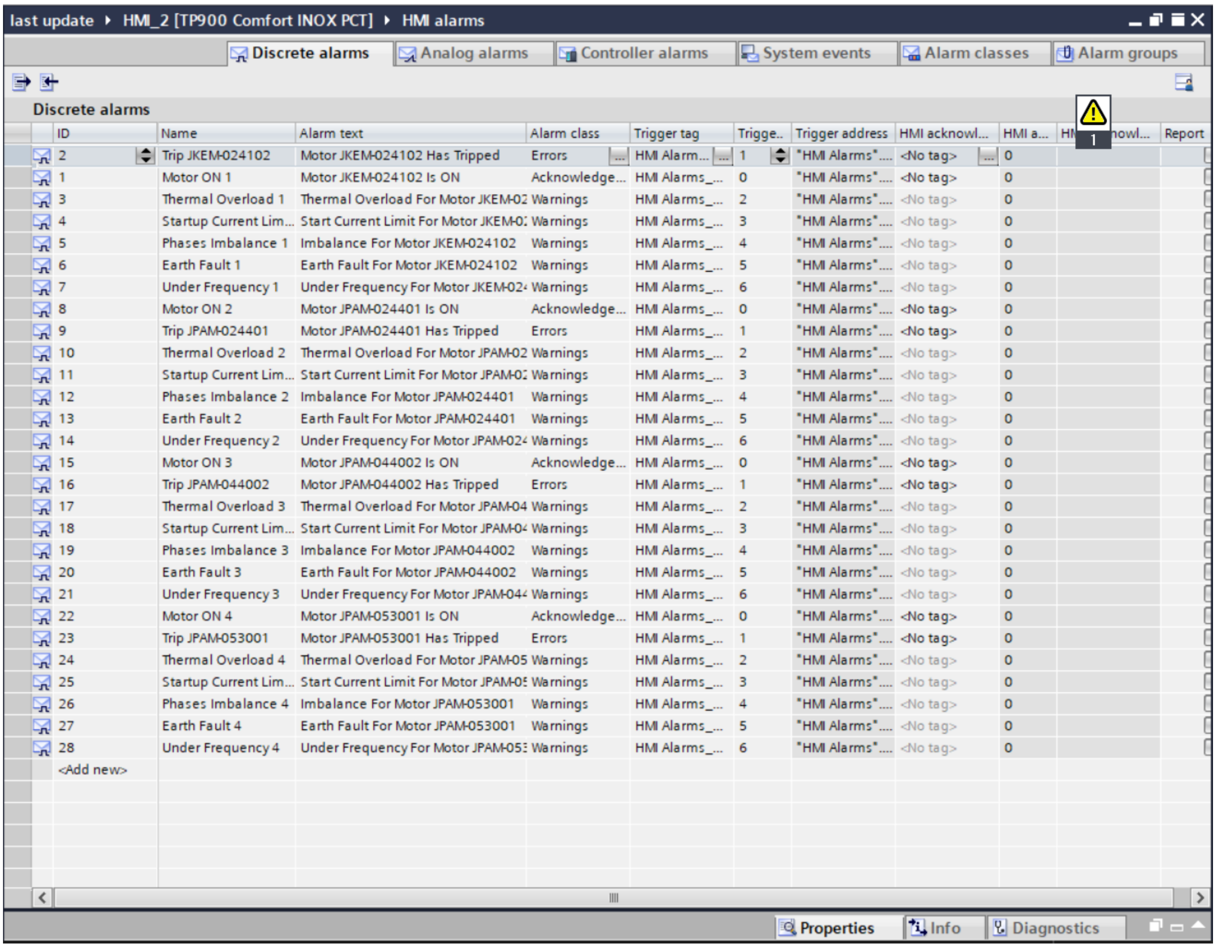


Figure 4.15: Motor screen

4.2.8.3 Bit alarms

This screen displays a comprehensive list of discrete alarms for various motors in an industrial control system. It includes details such as the ID, alarm type, alarm text, trigger tag, trigger address, and acknowledgment status for each alarm. The alarms cover a range of issues, including motor trips, thermal overloads, startup current limits, phase imbalances, earth faults, and under frequency warnings. This screen allows operators to monitor and manage alarms, ensuring quick identification and resolution of issues to maintain system efficiency and safety.



ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Trigger address	HMI acknowl...	HMI a...
2	Trip JKEM-024102	Motor JKEM-024102 Has Tripped	Errors	HMI Alarm...	1	"HMI Alarms"...	<No tag>	0
1	Motor ON 1	Motor JKEM-024102 Is ON	Acknowledge...	HMI Alarms_...	0	"HMI Alarms"...	<No tag>	0
3	Thermal Overload 1	Thermal Overload For Motor JKEM-02	Warnings	HMI Alarms_...	2	"HMI Alarms"...	<No tag>	0
4	Startup Current Lim...	Start Current Limit For Motor JKEM-0	Warnings	HMI Alarms_...	3	"HMI Alarms"...	<No tag>	0
5	Phases Imbalance 1	Imbalance For Motor JKEM-024102	Warnings	HMI Alarms_...	4	"HMI Alarms"...	<No tag>	0
6	Earth Fault 1	Earth Fault For Motor JKEM-024102	Warnings	HMI Alarms_...	5	"HMI Alarms"...	<No tag>	0
7	Under Frequency 1	Under Frequency For Motor JKEM-02	Warnings	HMI Alarms_...	6	"HMI Alarms"...	<No tag>	0
8	Motor ON 2	Motor JPAM-024401 Is ON	Acknowledge...	HMI Alarms_...	0	"HMI Alarms"...	<No tag>	0
9	Trip JPAM-024401	Motor JPAM-024401 Has Tripped	Errors	HMI Alarms_...	1	"HMI Alarms"...	<No tag>	0
10	Thermal Overload 2	Thermal Overload For Motor JPAM-02	Warnings	HMI Alarms_...	2	"HMI Alarms"...	<No tag>	0
11	Startup Current Lim...	Start Current Limit For Motor JPAM-0	Warnings	HMI Alarms_...	3	"HMI Alarms"...	<No tag>	0
12	Phases Imbalance 2	Imbalance For Motor JPAM-024401	Warnings	HMI Alarms_...	4	"HMI Alarms"...	<No tag>	0
13	Earth Fault 2	Earth Fault For Motor JPAM-024401	Warnings	HMI Alarms_...	5	"HMI Alarms"...	<No tag>	0
14	Under Frequency 2	Under Frequency For Motor JPAM-02	Warnings	HMI Alarms_...	6	"HMI Alarms"...	<No tag>	0
15	Motor ON 3	Motor JPAM-044002 Is ON	Acknowledge...	HMI Alarms_...	0	"HMI Alarms"...	<No tag>	0
16	Trip JPAM-044002	Motor JPAM-044002 Has Tripped	Errors	HMI Alarms_...	1	"HMI Alarms"...	<No tag>	0
17	Thermal Overload 3	Thermal Overload For Motor JPAM-04	Warnings	HMI Alarms_...	2	"HMI Alarms"...	<No tag>	0
18	Startup Current Lim...	Start Current Limit For Motor JPAM-0	Warnings	HMI Alarms_...	3	"HMI Alarms"...	<No tag>	0
19	Phases Imbalance 3	Imbalance For Motor JPAM-044002	Warnings	HMI Alarms_...	4	"HMI Alarms"...	<No tag>	0
20	Earth Fault 3	Earth Fault For Motor JPAM-044002	Warnings	HMI Alarms_...	5	"HMI Alarms"...	<No tag>	0
21	Under Frequency 3	Under Frequency For Motor JPAM-04	Warnings	HMI Alarms_...	6	"HMI Alarms"...	<No tag>	0
22	Motor ON 4	Motor JPAM-053001 Is ON	Acknowledge...	HMI Alarms_...	0	"HMI Alarms"...	<No tag>	0
23	Trip JPAM-053001	Motor JPAM-053001 Has Tripped	Errors	HMI Alarms_...	1	"HMI Alarms"...	<No tag>	0
24	Thermal Overload 4	Thermal Overload For Motor JPAM-05	Warnings	HMI Alarms_...	2	"HMI Alarms"...	<No tag>	0
25	Startup Current Lim...	Start Current Limit For Motor JPAM-0	Warnings	HMI Alarms_...	3	"HMI Alarms"...	<No tag>	0
26	Phases Imbalance 4	Imbalance For Motor JPAM-053001	Warnings	HMI Alarms_...	4	"HMI Alarms"...	<No tag>	0
27	Earth Fault 4	Earth Fault For Motor JPAM-053001	Warnings	HMI Alarms_...	5	"HMI Alarms"...	<No tag>	0
28	Under Frequency 4	Under Frequency For Motor JPAM-05	Warnings	HMI Alarms_...	6	"HMI Alarms"...	<No tag>	0
<Add new>								

Figure 4.16: bit alarms

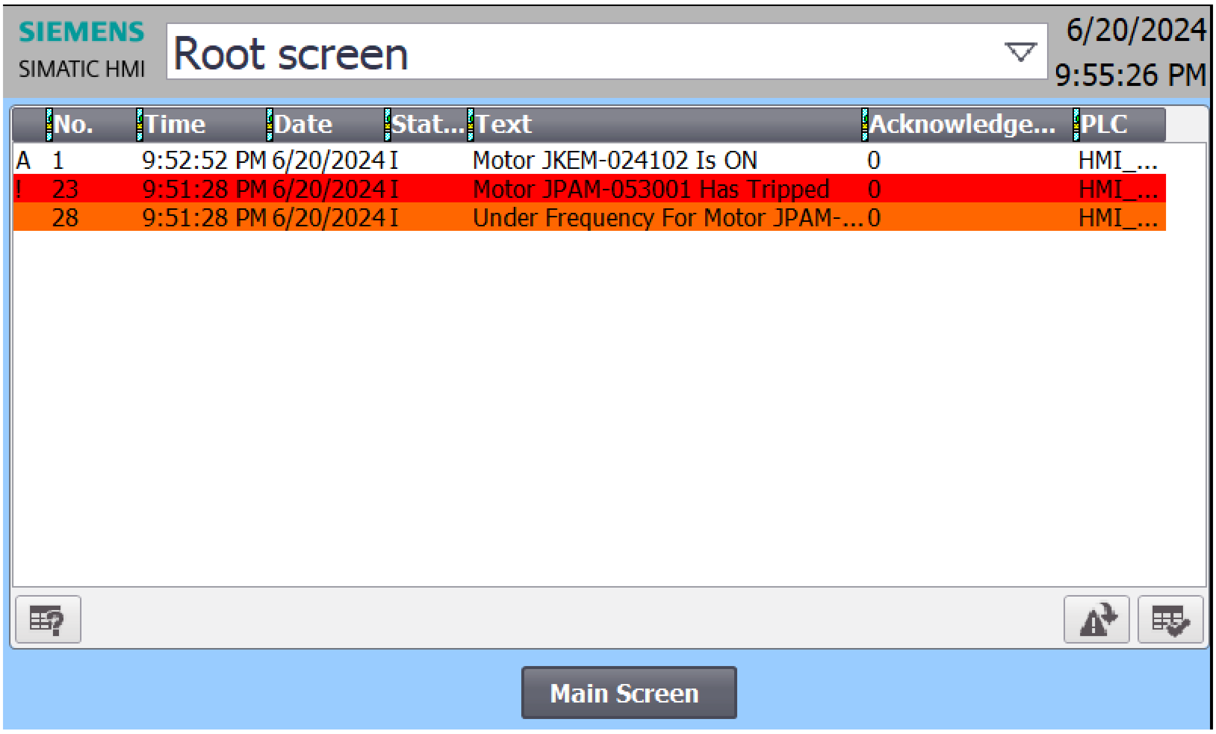


Figure 4.17: Alarme message

4.2.8.4 Current visualization

The screen displays a trend graph plotting values over time to visualize the motor's phase currents as well as the rating current.

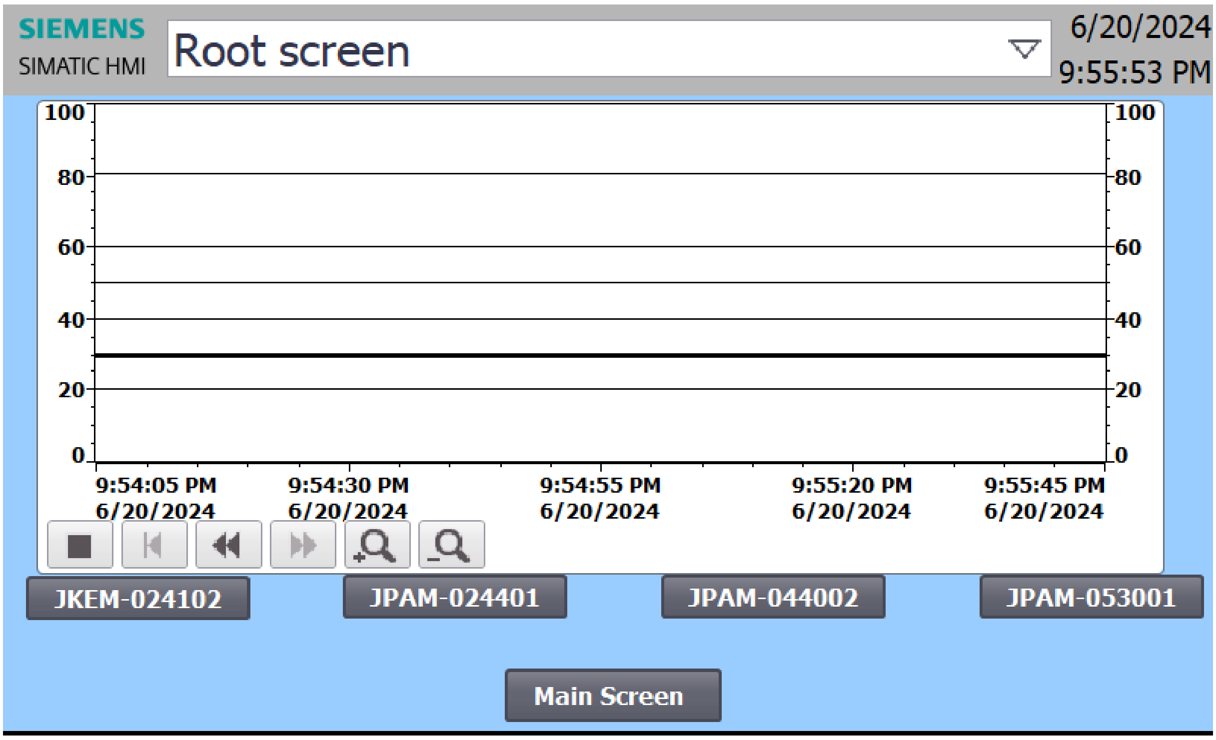


Figure 4.18: Current visualization

4.2.9 Conclusion

Chapter 4 details the integration of relay control into a Siemens S7-1200 PLC using TIA Portal software. Key points include the selection of the S7-1200 for its economic benefits, memory capacity, and scalability, hardware configuration, and creation of PLC table tags. The chapter also covers implementing ladder logic for functions such as scaling, local/remote control, start/stop, and various protections (start current limit, overload, imbalance, earth fault, and under frequency). Additionally, it emphasizes the importance of HMI for real-time monitoring and operational efficiency, providing a thorough understanding of relay integration in PLCs.

4.3 Advantages & desadvantages of this method

4.3.1 Advantages

– Simplification of System Architecture:
  - **Centralized Control**: Integrating all relay protections into a single PLC centralizes the control functions, simplifying the overall system architecture. This centralization enhances visibility and management of the system, allowing for easier monitoring and troubleshooting.

  - **Reduced Complexity**: By eliminating multiple discrete protection relays and interfaces, the system becomes less complex. This reduction in complexity can lead to fewer points of failure and easier maintenance.

– Cost Efficiency
  - **Lower Maintenance Costs**: With a single PLC managing all protections, maintenance efforts and costs are reduced. There's no need to service multiple different relay devices, leading to savings in both time and resources.

  - **Initial Investment**: Although the initial investment in a PLC may be significant, the long-term savings due to reduced maintenance and increased efficiency can justify this cost.

– Flexibility and Scalability
  - **Advanced Programming Capabilities**: PLCs offer advanced programming capabilities, allowing for customized protection and monitoring logic tailored to specific needs. This flexibility ensures that the system can adapt to future changes and requirements.

  - **Scalability**: PLC systems can be easily scaled to accommodate additional functions or expanded operations, making them suitable for growing systems or changing operational demands.

– Enhanced Reliability
  - **Consistent Performance**: A PLC system provides consistent performance and reliability, reducing the risk of protection failures. This reliability is crucial for maintaining the safety and operational integrity of electrical systems.

– Improved Data Management and Analysis
  - **Data Collection and Analysis**: PLCs can collect and analyze operational data, providing valuable insights into system performance and potential issues. This data can be used to optimize operations and improve preventative maintenance strategies.

4.3.2 Disadvantages

– Initial Costs and Implementation Efforts
 - **High Initial Investment**: The cost of purchasing and implementing a PLC system can be high. This includes the cost of the PLC hardware, software licenses, and the labor required for installation and programming.

 - **Implementation Time**: Transitioning to a PLC-based system can be time-consuming, requiring significant planning, programming, and testing before it can be fully operational.

– Dependency on a Single System
 - **Single Point of Failure**: With all relay protections integrated into a single PLC, the system becomes a single point of failure. If the PLC fails, it can lead to a complete loss of protection functions, which can be catastrophic.

 - **Reliability Concerns**: Ensuring the PLC is highly reliable and has redundancy measures (such as backup systems) is essential to mitigate this risk.

– Technical Complexity
 - **Advanced Skills Required**: Managing and programming a PLC requires specialized knowledge and skills. This technical complexity can be a barrier for some organizations, necessitating training and potentially hiring skilled personnel.

 - **Complex Troubleshooting**: While the system is simplified overall, troubleshooting PLC issues can be complex and may require expert intervention.

– Integration Challenges
 - **Compatibility Issues**: Integrating existing relay protections into a PLC system can present compatibility challenges. Ensuring that all components work seamlessly together requires careful planning and execution.

 - **Customization Needs**: Customizing the PLC to meet specific protection requirements can be complex and time-consuming, requiring detailed knowledge of both the PLC and the protection needs.

General conclusion

In this dissertation, we have explored the critical aspects of electrical motor protection and the transition from traditional protection systems to more advanced, integrated solutions using Programmable Logic Controllers (PLCs). The study has been driven by the challenges associated with the obsolescence of Gemstart 4 controllers, which pose significant maintenance and operational difficulties.

We began by detailing the fundamentals of electrical motor protection, outlining the various protection mechanisms essential for maintaining system reliability and safety. The study then provided an overview of the In Salah Gas project, highlighting the specific relay protections employed for different electrical motors within the substation.

The core of the dissertation focused on the use of Siemens' TIA Portal software, a comprehensive tool for the programming and integration of PLCs. This section provided detailed instructions on utilizing the software to implement the proposed solutions, ensuring a seamless transition from the outdated Gemstart 4 controllers to a more robust and flexible PLC-based system.

The practical application of these solutions was demonstrated through simulations, validating the effectiveness of the integrated PLC approach in enhancing the reliability and efficiency of the substation's operations. This transition not only simplifies the control and protection functions but also offers significant long-term benefits in terms of reduced maintenance costs and extended equipment lifespan.

In conclusion, the integration of relay protections into a PLC system represents a forward-thinking approach to managing electrical motor protections amidst technological obsolescence. This solution provides a scalable, efficient, and economically viable pathway for maintaining the operational integrity of substations. Future work may explore further advancements in PLC technology and their applications in broader contexts within the electrical engineering field.

Bibliography

[1] ALSTOM TD Ltd. Gemstart 4 Volume 1 - Users Guide PEL004-1071-002. Salford, England: ALSTOM TD Ltd., April 2003.

[2] ALSTOM TD Ltd. Gemstart 4 Volume 2 - Design Guide PEL004-1071-003. Salford, England: ALSTOM TD Ltd., April 2003.

[3] ALSTOM TD Ltd. Gemstart 4 Volume 3 - Specification Reference PEL004-1071-004. Salford, England: ALSTOM TD Ltd., April 2003.

[4] ALSTOM TD Ltd. Substation PLC and SCADA Operating Maintenance Manual. Salford, England: ALSTOM TD Ltd., April 2003.

[5] ALSTOM TD Ltd. FUNCTIONAL DESIGN SPECIFICATION. In Salah Project PEL004-0029. Salford, England: ALSTOM TD Ltd., April 2003.

[6] ALSTHOM. Network protection automation guide.: ALSTOM TD Ltd., 2002.

[7] Jargot, P. Langages de programmation pour API, Norme IEC 1131-3. Technique de l'ingénieur, vol. S 8 030.

[8] Bolton, W. Les automates programmables industriels. 5ème édition, 2009.

[9] Siemens. Available online: www.siemens.com. [Consulted on: 26/06/2024]

[10] Alexander Bürkle. Available online: https://alexander-buerkle.com/de-de/magazin-news/2018/simatic-s7-1200. [Consulted on: 26/06/2024]

[11] Instrumentation Blog. Available online: https://instrumentationblog.com/types-of-plc-programming-languages/. [Consulted on: 26/06/2024]

[12] Gonzaga, A. Les automates programmables industriels. Paris, France: Dunod, 1979.

[13] Petruzella, F. D. Programmable logic controllers. 4ème édition.2011.