ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

REDUNDANT AND SAFE WORK IMPLEMENTATION FOR S7-1200 PLC FAMILY

M.Sc. THESIS

Okan KONUK

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JUNE 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

REDUNDANT AND SAFE WORK IMPLEMENTATION FOR S7-1200 PLC FAMILY

M.Sc. THESIS

Okan KONUK
(504171118)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Salman KURTULAN

JUNE 2020

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

S7-1200 PLC AİLESİ İÇİN YEDEKLİ VE EMNİYETLİ ÇALIŞMA UYGULAMASI

YÜKSEK LİSANS TEZİ

Okan KONUK
(504171118)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Salman KURTULAN

HAZİRAN 2020

Okan KONUK, a M.Sc. student of İTU Graduate School of Science Engineering and Technology student ID 504171118, successfully defended the thesis entitled "REDUNDANT AND SAFE WORK IMPLEMENTATION FOR S7-1200 PLC FAMILY", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Salman KURTULAN** İstanbul Technical University	.....
Jury Members :	**Assis. Prof. Dr. İlker ÜSTOĞLU** Istanbul Technical University	.....
	Assis. Prof. Dr. A.Evren GÖKSUNGUR Nişantaşı University	.....

Date of Submission : 04 May 2020
Date of Defense : 18 June 2020

To my family and friends,

FOREWORD

Firstly, I would like to express my most sincere appreciation to Prof. Dr. Salman KURTULAN for guiding and supporting me in every aspect.

I would like to thank to my parents and dear childhood friends for being helpful towards me and not withholding their support whenever I needed.

Lastly, a very hearty thank you goes to Tuğçe ESENDURAN for creating a breaking point in my life and making it more meaningful. I will be grateful until my last breath for your love and affection.

June 2020

Okan KONUK
(Control and Automation Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ATM : Automatic Teller Machine
CFC : Continuous Function Chart
CPU : Central Processor Unit
DC : Direct Current
FBD : Function Block Diagram
FS : Fail-safe
HFT : Hardware Fault Tolerance
HMI : Human Machine Interface
IE : Industrial Ethernet
IEC : International Electrotechnical Comission
IL : Instruction List
IM : Interface Module
ISO : International Organization of Standardization
I/O : Input/Output
LD : Ladder Diagram
MooN : M-out-of-N
MTBF : Mean Time Between Failure
PFD : Probability of Failure on Demand
PL : Performance Level
PLC : Programmable Logic Controller
PROFINET : Process Field Net
PN : Profinet
SFC : Sequential Function Chart
SFF : Safe Failure Fraction
SIL : Safety Integrity Level
SIS : Safety Instrumented Systems
ST : Structured Text
TA : Transfer Area
TMR : Triple Modular Redundancy
TSE : Turkish Standards Institution

LIST OF TABLES

Page

LIST OF FIGURES

Page

REDUNDANT AND SAFE WORK IMPLEMENTATION FOR S7-1200 PLC FAMILY

SUMMARY

One of the most important components of the automation field, which has an undeniable importance in today's technology, is undoubtedly seen as Programmable Logical Controllers (PLC). These devices are found in almost every area of industrial applications and are usually at the center of control and automation systems. Before these devices were discovered, industrial control systems needed relay and contactor based technology. However, the fact that the relays and contactors are not long-lived, are very easily affected by the industrial environmental conditions, and most importantly, the lack of a modular structure led to the debut of PLCs with the developing industry. This process, which started in the late 1960s, has improved day by day and has taken its current form.

As the operating principle, a PLC is a microprocessor based device that processes the data it receives from the input elements to the output elements according to the program written in it. Input elements are generally composed of units such as sensors, switches and buttons, while output elements are usually units such as leds, drivers and motors. A PLC can also be defined as an electronic system with sections such as input/output units, memory and central operating unit (CPU). Many basic logical operations such as timing, counting and sequencing are performed by these devices. On the other hand, these devices can be used in many functions such as motion control, process control and data control. With the developing technology and industry, new requirements have forced these devices to develop continuously. The advancement of communication, processor power and memory management each period has enabled the development of new PLCs to be more compact, faster, smaller and also have more memory.

The process that started from the debut of PLCs brought important changes for machine manufacturers. The way the machines are connected to each other has been changed and as a result, PLCs have replaced the classical relays. However, these regulations caused PLCs to stay away from machine safety. For this reason, certain standards have been introduced for systems where safety is critical and potential consequences can harm both the environment and human life. The main purpose of these standards is to avoid the potentially dangerous consequences of such systems or to reduce the effects on the environment and people as much as possible. Many commissions, such as the International Electrotechnical Commission (IEC), have been involved in the formation of these standards. IEC 61508 and IEC 61511 are the most used standards for ensuring safety in industrial automation. With the development of standard PLCs, safety requirements in the industry have led to the design of safety PLCs. Safety PLCs meet the safety need in industrial automation at an important level with its ease of maintenance, advanced diagnostics and high reliability. Thanks to the

modular feature of PLCs, the use of safety PLCs is highly preferred in applications where the machine structure or control rules to be applied are complex.

System reliability is one of the important concepts emerging with these developments. The reliability of a system can be explained in direct proportion to its ability to eliminate faults that may occur without affecting the operation of the system. Redundancy is a significant way to ensure fault tolerance. Redundancy is, in principle, having more than one unit that can perform the same task in a system. The important point here is that while a single unit is sufficient for the system to operate, at least one more unit is kept as a backup. Redundancy technology is often used where there is continuous production. The basic philosophy of redundancy in industrial automation systems can be said that in the event of a malfunction in a PLC which actively controls the system, it is the activation of the PLC, which is kept as the backup, without interrupting the operation. The Redundancy concept is divided into many different categories according to the way it is provided and executed.

Recently, there are special PLC systems that many PLC manufacturers have produced for applications that require redundancy as well as safety and used in large scale projects throughout the world. The fact that these systems are high-end, having more input/output units compared to a standard PLC, and they cost highly at installation and commissioning is very expensive solution for small businesses.

The design and implementation of control circuits, which are common in the field of industrial automation, and their implementation with PLC is another important issue. In order for them to be realized properly and optimally, they must be designed well. There are many known methods for modeling systems. Among them, intuitive methods that provide temporary and practical solutions are easy to implement. Asynchronous and synchronous sequential circuits are the methods that have an algorithm of certain rules, but are limited in use due to some issues. Petri Nets method, which is easier to adapt to PLCs based on its mathematical infrastructure, is a method that simplifies the design of complex industrial systems. Thanks to its graphical representation, it is very simple to understand and monitor them.

In this thesis, it was aimed to design the conveyor belt system with Petri Nets and to operate it in accordance with safety or redundancy principle when demanded, using standard PLCs. It was intended to develop a cheaper solution that can perform the same basic functions instead of the costly solutions offered in the market.

In the introduction part, firstly, the principles of safety and redundancy were introduced and their basic functions were explained. Along with these, safety instrumented systems and system reliability were noted. Then, the known and applied methods in the design of industrial automation systems were simply explained and compared to each other.

In the second part of the thesis, Petri Nets method chosen for the design of the conveyor belt system was explained. The basic components and features of Petri Nets, state transition mechanism and graphical interface of them were examined in details. In the following subtitles, state equations were given by using the mathematical infrastructure of this method and the whole section was supported with examples for better understanding. Finally, the classification of Petri Nets according to the main features was explained with examples.

In the third part, in order for the concepts of safety and redundancy to be better understood by the reader, firstly, some definitions and terms used throughout the

section were given. Afterwards, safety instrumented systems mentioned in the introduction were explained in detail and the basic safety standards such as IEC 61508, IEC 62061 and ISO 13849 were summarized. Then, *1oo1*, *1oo2*, *2oo2* and *2oo3* structures, which are mentioned in the IEC 61508 standard, were explained. The redundancy principle was clarified in detail in the last part of the third chapter.

In the next part, an application was made in line with the purpose of the thesis. First of all, the conveyor belt system was modeled with Petri Nets. In order to convert the Petri Net design into the PLC code, a previously proposed method was used and this method was explained in details. Afterwards, it was tried to establish *1oo2* and *2oo2* structures described in the previous section with standard PLCs so that the entire system can be operated in conformity with safety or redundancy principle. For this purpose, how PLCs in the system are communicating with each other was explained in detail. Finally, a special touch screen designed for monitoring and control of the entire system was mentioned.

In the last part of the thesis, the results of the application were shared. It was concluded that the targeted safety and redundancy principles could be mainly established with the standard PLCs used in the project.

S7-1200 PLC AİLESİ İÇİN YEDEKLİ VE EMNİYETLİ ÇALIŞMA UYGULAMASI

ÖZET

Günümüz teknolojisinde yadsınamaz bir öneme sahip olan otomasyon alanının sahip olduğu en önemli bileşenlerinden biri hiç kuşkusuz ki Programlanabilir Mantıksal Kontrolörler (PLC) olarak görülmektedir. Bu cihazlar endüstriyel uygulamaların neredeyse her alanında bulunmaktadırlar ve genellikle kumanda ve otomasyon sistemlerinin merkezinde bulunurlar. Bu cihazlar keşfedilmeden önce endüstriyel kumanda sistemleri röle ve kontaktör tabanlı teknolojiye muhtaçlardı. Ancak röle ve kontaktörlerin uzun ömürlü olmamaları, endüstriyel çevre şartlarından çok kolay etkilenmeleri ve en önemlisi modüler bir yapıya sahip olmamaları gelişen sanayi ile birlikte PLC'lerin ortaya çıkmasına vesile olmuştur. 1960'lı yılların sonlarında başlayan bu süreç, her geçen gün daha da gelişerek günümüzdeki halini almıştır.

Bir PLC çalışma prensibi olarak giriş elemanlarından aldığı verileri, bünyesinde yazılmış olan programa göre işleyerek çıkış elemanlarına aktaran mikroişlemci tabanlı bir cihazdır. Giriş elemanları genellikle sensörler, anahtarlar ve tuşlar gibi birimlerden oluşurken, çıkış elemanları genellikle led lambalar, sürücüler ve motorlar gibi birimlerdir. Bir PLC, giriş/çıkış birimleri, bellek ve merkezi işletim birimi (CPU) gibi bölümleri olan bir elektronik sistem olarak da tanımlanabilir. Zamanlama, sayma ve sıralama gibi bir çok temel mantıksal işlemler bu cihazlar tarafından gerçekleştirilir. Öte yandan, hareket kontrolü, süreç kontrolü ve veri denetimi gibi bir çok işlevde bu cihazlar kullanılabilir. Gelişen teknoloji ve sanayi ile birlikte yeni gereksinimler bu cihazları sürekli gelişmeye zorlamıştır. Haberleşme, işlemci gücü ve bellek yönetiminin her dönem ilerlemesi, yeni dönem PLC'lerin daha kompakt, daha hızlı, daha küçük ve aynı zamanda daha çok belleğe sahip olacak şekilde geliştirilmesini sağlamıştır.

PLC'lerin ortaya çıkmasından itibaren başlayan süreç makine üreticileri için önemli değişiklikleri beraberinde getirdi. Makinelerin birbirlerine bağlanma şeklinde değişikliğe gidildi ve bunun sonucunda klasik rölelerin yerini PLCler aldı. Ancak bu düzenlemeler PLC'lerin makine emniyetinden uzak kalmasına sebep oldu. Bu sebeple, emniyetin kritik olduğu ve olası sonuçların hem çevreye hem de insan hayatına ciddi derecede zarar verebilecek potansiyelde olduğu sistemler için belli başlı standartlar getirilmeye başlandı. Bu standartların temel amacı bu tarz sistemlerin olası tehlikeli sonuçlarından kaçınmak ve ya bu sonuçların çevreye ve insana olan etkilerini mümkün olduğunca azaltmaktır. Uluslararası Elektroteknik Komisyonu (IEC) gibi bir çok komisyon bu standartların oluşumunda rol almışlardır. IEC 61508 ve IEC 61511 endüstriyel otomasyonda emniyetin sağlanması amacıyla en çok kullanılan standartlar arasında gösterilebilir. Standart PLC'lerin gelişimiyle beraber endüstrideki emniyet gereksinimleri, emniyet PLC'lerinin tasarlanmasına vesile olmuştur. Emniyet PLC'leri bakım kolaylığı, ileri seviye diagnostik ve yüksek güvenilirlik sağlaması ile endüstiryel otomasyonda emniyet ihtiyacını önemli seviyede karşılamaktadır.

PLC'lerin doğasında bulunan modüler olma özelliği sayesinde de makine yapısının ya da uygulanacak kontrol kurallarının karmaşık olduğu uygulamalarda emniyet PLC'lerinin kullanımı çokça tercih edilmektedir.

Sistem güvenilirliği bu gelişmelerle birlikte ortaya çıkan önemli kavramlardan biridir. Bir sistemin güvenilir olması, oluşabilecek arızaları sistemin işleyişini etkilemeden giderebilmesi yeteneği ile doğru orantılı olarak açıklanabilir. Artıklık, fazlalık gibi anlamlara gelen *redundancy* ise sisteme hata toleransı kazandırmanın en önemli yoludur, ancak 100% olarak hata toleransını garanti etmez. *Redundancy* prensip olarak bir sistemde belirli bir işi yapabilecek birden fazla birimin bulundurulmasıdır. Burada önemli husus, sistemin çalışabilmesi için tek bir birim yeterli olacak iken yedekte en az bir birimin daha hazır bulundurulmasıdır. *Redundancy* teknolojisi genellikle sürekli üretimin olduğu yerlerde kullanılır. *Redundancy* kavramının endüstriyel otomasyon sistemlerinde uygulanmasının temel felsefesi; bir PLC aktif bir şekilde sistemi kontrol ederken bir arıza yaşanması durumunda yedekte bekletilen PLC'nin çalışmayı aksatmadan devreye girmesidir. *Redundancy* kavramı sağlanış, işleniş ve yürütülüş biçimlerine göre bir çok farklı kategoriye ayrılmaktadır.

Günümüzde endüstriyel otomasyonda söz sahibi olan bir çok PLC üreticisinin emniyet gerektiren uygulamalar için olduğu gibi *redundancy* gerektiren uygulamalar için de üretmiş olduğu ve dünyanın bir çok bölgesinde büyük ölçekli projelerde kullanılan özel PLC sistemleri mevcuttur. Bu sistemlerin üst düzey olmaları, standart bir PLC'ye göre daha fazla giriş/çıkış modülleri barındırması, kurulumunun ve yürütülmesinin maliyetli olması küçük işletmeler için fazlaca maddi yük anlamına gelebilir.

Endüstriyel otomasyon alanında çokça karşılaşılan kumanda devrelerinin tasarımı ve PLC ile gerçeklenmesi önemli konulardan bir diğeridir. Mevzu bahis kumanda devrelerinin ya da otomasyon sistemlerin amaca uygun ve optimum bir şekilde çalışabilmesi için iyi bir şekilde tasarlanıp PLC diline öyle aktarılmaları gerekmektedir. Sistemlerin modellenmesi için bilinen birçok modelleme yöntemi bulunmaktadır. Bunların başında geçici ve pratik çözümler sağlayan ve uygulaması kolay olan sezgisel yöntemler gelmektedir. Asenkron ve senkron sıralı devreler yöntemleri ise belli kurallar çerçevesinde oluşturulan ancak bazı hususlar neticesinde kullanımı sınırlı olan yöntemlerdir. PLC'lere uyarlaması daha kolay olan ve bir matematiksel alt yapıya dayanan Petri Ağları yöntemi karmaşık endüstriyel sistemleri basit hale indirgeyerek tasarımı oldukça kolaylaştıran bir yöntemdir. Grafiksel olarak ifade edilebilmesi sayesinde anlaşılırlığı ve takibi basittir.

Bu tez çalışması dahilinde, konveyör bant gibi çokça karşılaşılan bir endüstriyel sistemin Petri Ağları yöntemi ile tasarlanması ve standart PLC'ler kullanılarak mevzubahis sistemin gerektiğinde emniyetli gerektiğinde *redundancy* prensibine uygun bir şekilde çalıştırılabilmesi amaçlanmıştır. Nispeten ucuz ve kompakt PLC'ler kullanılarak piyasada sunulan maliyetli çözümlerin yerine temelde aynı işlevi yerine getirebilen, küçük işletmeler tarafından kullanılabilecek daha ucuz bir çözüm geliştirilmesi hedeflenmiştir.

Bu çalışmanın giriş bölümünde, öncelikle emniyetli çalışma ve *redundancy* prensibinden bahsedilmiş ve temel işlevleri açıklanmıştır. Bu kavramlarla birlikte çokça işitilen emniyet enstrümanlı sistemler ve sistem güvenilirliği kavramlarına değinilmiştir. Sonrasında, endüstriyel otomasyon sistemlerinin tasarımında bilinen ve uygulanan yöntemler basitçe açıklanıp birbirileri ile kıyaslanmıştır.

Çalışmanın ikinci bölümünde bu tez çalışması için ele alınan konveyör bant sisteminin tasarımı için seçilen Petri Ağları yöntemi anlatılmıştır. Petri Ağları'nın temel

elemanları ve özellikleri, durum geçiş mekanizması ve grafiksel ara yüzü detaylı bir şekilde incelenmiştir. İlerleyen alt başlıklarda, bu yöntemin matematiksel alt yapısından faydalanarak durum denklemleri verilmiştir ve daha iyi anlaşılabilmesi için tüm bölüm örneklerle desteklenmiştir. Son olarak belli başlı özelliklere göre Petri Ağları'nın sınıflandırılması örneklerle anlatılmıştır.

Çalışmanın üçüncü bölümünde emniyet ve *redundancy* kavramlarının okuyucu tarafından daha iyi anlaşılabilmesi için öncelikle bölüm boyunca kullanılan bazı terimlerin açıklamaları verilmiştir. Sonrasında giriş bölümünde de bahsedilen emniyet enstrümanlı sistemler detaylıca anlatılmıştır ve temel emniyet standartlarından olan IEC 61508, IEC 62061 ile ISO 13849 standartları özetlenmiştir. Sonrasında IEC 61508 standardında bahsedilen sistem mimari yapılarından *1oo1*, *1oo2*, *2oo2* ve *2oo3* yapıları anlatılmıştır. Üçüncü bölümün son kısmında *redundancy* prensibi detaylı bir şekilde açıklanmıştır.

Çalışmanın dördüncü bölümünde tezin amacı doğrultusunda bir uygulama yapılmıştır. Öncelikle konveyör bant sistemi belirli senaryolara göre hareket edecek şekilde Petri Ağları yöntemi ile modellenmiştir. Elde edilen modeli PLC koduna dönüştürmek amacıyla mantıksal ifadelere çevirebilmek için daha önce önerilmiş olan bir yöntemden faydalanılmıştır ve bu yöntem detaylıca açıklanmıştır. Sonrasında tüm sistemin istenildiğinde emniyetli istenildiğinde *redundant* çalıştırılabilmesi için standart PLC'ler ile bir önceki bölümde anlatılan *1oo2* ve *2oo2* yapıları oluşturulmaya çalışılmıştır. Bu amaç doğrultusunda sistemde var olan PLC'lerin birbirileri ile nasıl haberleştirildikleri detaylıca anlatılmıştır. Son olarak, tüm sistemin takibi ve kontrolü için tasarlanmış olan özel dokunmatik ekrandan bahsedilmiştir.

Çalışmanın son bölümünde ise gerçekleştirilen uygulamanın sonuçları paylaşılmıştır. Çalışmada kullanılan standart PLCler kurulan yapıda, hedeflenen emniyetli ve *redundant* çalışma prensiplerinin temel özellikleri ile sağlandığına kanaat getirilmiştir.

1. INTRODUCTION

A Programmable Logic Controller (PLC) is a specialized computer responsible for the control of automation systems. PLCs are basically in contact with sensors and actuators, so they are capable of controlling and monitoring automation systems or processes. Thanks to the ability of PLCs to be configured with many elements in the industry, they are utilized in many industrial areas such as automotive, chemical, aircraft, railway and so on [1]. By virtue of the technology developing day by day, the more complex systems make it necessary to make some changes in the way PLCs work. Modern electronic systems are one of the most suitable examples of the complex systems mentioned. While controlling some critical processes where safety is important, any malfunctions that may arise during the operation phase may have irreversible consequences for the environment or people. Thereupon, the concept of safety instrumented systems (SIS) became involved in the industry. The primary purpose of such a system is to carry out one or more safety functions in case of any failure that leads to the safety risk to be enhanced. As a result of the said functions, the relevant system can be moved to a safe state or its operation can be stopped completely. For example, closing the main gas valve as a result of the gas pressure being too high than necessary is a common safety function in the chemical industry.

Safety instrumented systems, in summary, are designed to prevent events that may lead to dangerous consequences or to mitigate the impacts of these events. Therefore, these systems have to meet certain standards that express some requirements. Some standards prioritize safe software development, while others emphasize safety requirements. In chapter 3, detailed information about the standards will be given.

Along with these developments, system reliability has become an important issue in industry. System reliability refers to the capability of technical equipment to fulfill its function during its operation period and is directly proportional to the system's ability to tolerate failures. A system capable of fault tolerance means that it can continue its desired operation even though there is a malfunction in any part of the system. When designing such systems, in the event of a malfunction, it is preferred that the intended

operation continues at low levels rather than it is completely stopped. Based on this information, it can be clearly said that fault tolerant systems are in accordance with the redundancy principle. Redundancy, as will be mentioned in chapter 3, is the presence of more than one unit to perform the required functions of a system. Redundant systems are mostly preferred in manufacturing factories, hospitals, aviation, oil and gas, wastewater, electricity distribution lines and many other areas. A factory producing plastics is a good example for the use of these systems. In places where such production is made, injection blow molding machines are used to finalize the related objects. These machines use air compressors. As will be expected, a malfunction in the air compressor will negatively affect production. For this reason, a second air compressor is usually kept ready to operate whenever it is needed.

In order to meet the safety or redundancy requirements of industrial systems that are getting more and more complex day by day, many companies, especially Siemens, are producing special series PLCs. Such PLCs are usually high-end and contain a large number of inputs and outputs. For this reason, the installation and commissioning of these PLCs are more costly than expected. Generally, such PLCs are suitable for control of large and complex systems or processes. On the contrary, such a solution is too expensive for simpler, smaller or not too safety-critical systems or processes [2]. Within the scope of this thesis work, using relatively cheaper and compact PLCs, a structure that can compily with redundancy or safety principles for a sample conveyor belt system will be tried to be established.

While the control circuits, which have an important place in industrial automation systems, are realized, methods based on intuitive, logical or discrete event system approach can be preferred [3]. Heuristic methods are often preferred for this purpose, although they have not been systematized. The success of a system realized with such a method is completely related to the experience and intuition power of the designer. Likewise, systems designed by heuristic method differ from person to person when programming with PLC. In this case, it will be very difficult to make changes to the system later, when necessary.

Asynchronous sequential and synchronous sequential circuit models are the logical circuit design methods used in the implementation of control circuits. These methods consist of creating a flow table, making the necessary reductions and then obtaining functions related to the system, on paper. In PLC side, programming these functions

created by following a certain algorithm does not vary. Having a systematic makes it easy to monitor and make changes for control circuits. However, according to the complexity of the relevant circuit, high number of inputs and outputs increases the probability of the designer to make mistakes in direct proportion. On the other hand, the design process that takes place on paper takes a lot of time [3].

The discrete event approach is another method used when implementing control circuits. Petri Nets are one of the methods used in the design of such systems. Petri Nets are based on a strong mathematical infrastructure. There are many different classes of Petri Nets that meet the needs of the system to be designed. Petri Nets' ability to be expressed graphically is a significant advantage which is provided to designers. In this way, it is very easy to follow the designed system and make changes on it later. In order to implement a system with PLC designed with Petri Nets, logical functions must be produced. In some studies related to design and implementation with Petri Nets, the state machine class of Petri Nets is used [4,5]. LAD language is preferred when implementing such Petri Nets with PLC. While the logical functions produced over the designed Petri Nets are realized with PLC, set & reset commands are often used.

In this thesis work, Petri Nets method was used while designing the sample system, which is a conveyor belt system. By making use of the features of the main Petri Net classes, a complete Petri Net model has been created. While implementing the designed Petri Net with PLC, a proposed logical expression method was used instead of the usual set & reset method. After obtaining the necessary logical functions with this method, LAD language is used in the implementation of these with PLC.

2. PETRI NETS

Petri Nets are a graphical tool used for modeling and analysis of simultaneous processes occuring in distributed systems. It is foreseen that this method was found by Carl Petri in 1939 in order to model chemical processes [6].

As with many engineering fields, the design of manufacturing systems can be realized benefiting from models. Petri nets ensure that these established models are suitable for accuracy and effective analysis. Moreover, these models can be implemented using many different programs or software languages. Petri nets facilitate the relationship between the designer and the user due to its easily understandable graphical nature [7]. More importantly, since a model built with petri nets can be expressed mathematically, testing for faulty or unwanted situations that may occur in the design can be easily performed.

2.1 Components of Petri Nets

Petri nets are defined as a bidirectional graph consisting of places and transitions. Places and transitions are connected to each other by arcs, and system dynamics are represented by tokens [8]. In the Figure 2.1, these concepts are shown on an exemplary Petri Net.

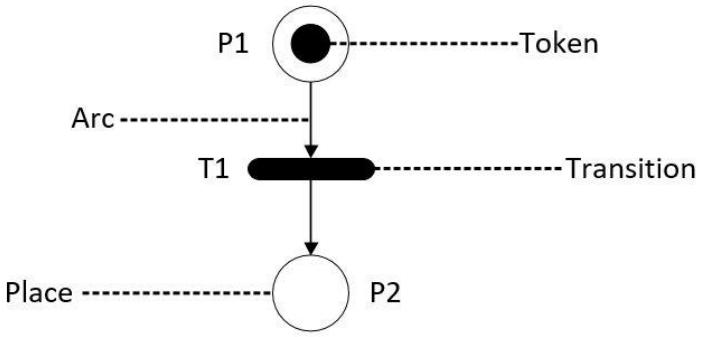


Figure 2.1 : Components of Petri Nets.

Places refer to the conditions associated with the modeled system, while transitions refer to events that occur within the modeled system. The task of the arcs is to connect

the places and transitions in both directions, if desired. It is not possible to connect neither two places nor two transitions to each other using arcs [9,10].

Tokens are located inside places of Petri Nets. If a place contains a token, it means that the condition set for that place is met. Otherwise, the condition assigned for that place is not satisfied yet. Status information of the whole system can be expressed by distribution of tokens in Petri Nets [9]. In any given time, a place belonging to a Petri Net may contain more than one token or no token at all. This feature can be determined by the designer and, if desired, a Petri Net can also be designed so that places in a Petri Net can include at most one token at any time. In this regard, how the system to be modeled should behave plays an important role.

2.2 Notations and Definitions

In this section, some necessary notations and definitions related to Petri Net is given.

A Petri Net structure is a weighted bipartite graph

$$PN = (P, T, A, w) \tag{2.1}$$

where

- $P = \{p_1, p_2, \dots, p_n\}$ is defined as the set of places
- $T = \{t_1, t_2, \dots, t_m\}$ is the transition set
- A is the set of arcs both transitions to places and places to transitions. An arc connecting the i^{th} place to j^{th} transition is shown as (p_i, t_j). Similarly, (t_m, p_n) is the expression for the arc connecting m^{th} transition to n^{th} place.
- w represents the weight function for the arcs. In Petri nets, there may be more than one arc connecting a place to a transition. The number of these arcs is indicated by the weight function, and shown as $w : A \rightarrow \{1,2, \dots\}$. In other words, weight is the number of token to be accumulated at a transition or to be transferred from a transition to the relevant place.
- When defining a Petri Net, the set of input places belonging to t_m transition is expressed by $I(t_m)$. Similarly, $O(t_m)$ is the set of output places belonging to

t_m transition. These terms will be taken into consideration in the process of building the mathematical model. These notations are defined as follows:

$$I(t_m) = \{p_n \in P : (p_n, t_m) \in A\} \tag{2.2}$$

$$O(t_m) = \{p_n \in P : (t_m, p_n) \in A\} \tag{2.3}$$

Similarly, the sets $I(p_n)$ and $O(p_n)$ are defined as the set of input transitions and output transitions belonging to p_n place, respectively.

- An arc drawn from p_n place to t_m transition shows that $p_n \in I(t_m)$. A function is defined like $w(p_n, t_m) = k$ means there are k arcs connecting p_n place to t_m transition. Likewise, it can be said that there is one arc with the weight k.
- If a weight value is specified at the top of an arc in Petri Nets, this means that there is that amount of arcs between the related place and transition. Alternatively, if such a weight value is not specified, the considered number of arcs between the related place and transition is 1.The following expressions can be written for weight functions:

$$p_n \notin I(t_m) \Rightarrow w(p_n, t_m) = 0 \tag{2.4}$$

$$p_n \notin O(t_m) \Rightarrow w(t_m, p_n) = 0 \tag{2.5}$$

For better understanding, the explained terms above are detailed on a basic Petri Net in the Figure 2.2.

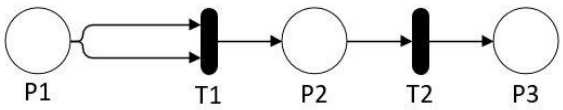


Figure 2.2 : Basic Petri Net example.

According to the Petri Net above:

- $P = \{p_1, p_2, p_3\}$
- $T = \{t_1, t_2\}$
- $A = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3)\}$
- $w(p_1, t_1) = 2,\ w(t_1, p_2) = w(p_2, t_2) = w(t_2, p_3) = 1$ or $w: A \rightarrow \{2,1,1,1\}$.

2.3 State Change in Petri Nets

The state change event in Petri Nets is provided by transferring tokens from one place to another. In this way, the Petri Net changes from its current state to a new state. Activated transitions must be triggered in order for this mentioned state change to take place. For a transition $t_m \in T$ to be activated, the following condition must hold:

$$x(p_n) \geq w(p_n, t_m) \quad , \quad \forall p_n \in I(t_m) \tag{2.6}$$

where $x(p_n)$ is the expression for the number of tokens that the place p_n includes. Otherwise speaking, for the transition t_m to be activated, the number of tokens included by the place p_n must be greater than or equal to the weight of the arc which connects the place p_n to the transition t_m.

As it is said before, Petri Nets change their state from current to the next one after an activated transition is triggered. Whenever an activated transition in Petri Net is triggered, the token distribution at the next state for all places in Petri Net can be formulized as following:

$$x'(p_n) = x(p_n) - w(p_n, t_m) + w(t_m, p_n) \tag{2.7}$$

According to (2.7), after triggering the activated transition t_m, the number of tokens at the place p_n is decreased by the weight of the arc which connets the place p_n to the transition t_m. This statement is valid only when the place p_n is the input place belonging to the transition t_m or -to state differently- $p_n \in I(t_m)$. If the place p_n is said to be the output place belonging to the transition t_m, then the number of tokens is increased by the weight of the related arc. In conclusion, taking into consideration the possibility that the place p_n can be both input and output place of the transition t_m, the number of tokens at p_n is declined by amount of $w(p_n, t_m)$ from it and then added by amount of $w(t_m, p_n)$ to it. For a clearer picture, the example at Figure 2.3 is given.

For the purpose of demonstrating the process of triggering transitions and changing state of a Petri Net, an illustration of a Petri Net whose inital state vector which gives information about token distrubition is $x_0 = [4 \quad 0 \quad 0 \quad 0]$, given at Figure 2.3. Since the transition t_1 demands at least 3 tokens to be activated and the place p_1 contains 4 tokens, (2.6) is satisfied. Therefore, it can be said that the only activated transition in

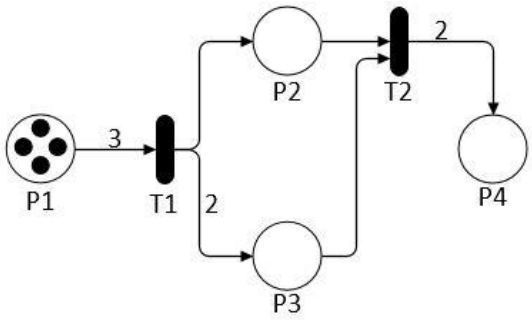
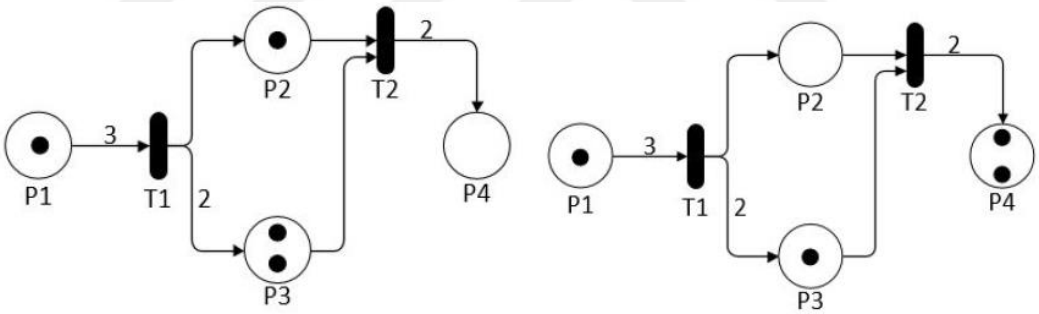


Figure 2.3 : State transition example for a Petri Net.

this Petri net is t_1. As soon as t_1 triggers, considering the weight of arcs, 3 tokens are reduced from p_1 then 1 token is placed to p_2 whereas p_3 takes 2 tokens. In another way, the next state can be calculated as $x_1 = [1 \quad 1 \quad 2 \quad 0]$ by just applying (2.7). After the first state changing process, the token distribution of the Petri Net is shown at Figure 2.4a. Subsequently, it can be seen from Figure 2.4a, the transition t_2 is activated. Triggering t_2 results in loss of 1 token for both p_2 and p_3 considering the weights of each arc connecting them to t_2. As it is shown in 2.4b, the next state is obtained as $x_2 = [1 \quad 0 \quad 1 \quad 2]$.



(a) Petri Net after t_1 triggers. (b) Petri Net after t_2 triggers.

Figure 2.4 : State changing stages of the Petri Net.

In conclusion, for a state change to occur in Petri Nets, an activated transition must be triggered. After triggering, tokens are taken from each input places belonging to the triggered transition by the amount of the weights of arcs which connects input places to it. Lastly, each output places belonging to the triggered transition gain tokens by the amount of the weights of arcs connecting the mentioned transition to output places.

2.4 State Equations

In the previous section, after triggering an activated transition in Petri Nets, token transfers were calculated and the new states was shown in Figures 2.4a and 2.4b.

Calculating these transitions step-by-step in more complex Petri Nets is highly probable to be very troublesome. For this reason, it is easier to use mathematical methods for investigating the state change dynamics of Petri Nets. For a single place, the next state is calculated with the help of (2.7). In this section, a mathematical model covering all places and transitions in a Petri Net will be given.

Benefiting from (2.7) the vector representations of the current and next state of Petri Nets are developed as follows:

$$x = [x(p_1), x(p_2), \dots, x(p_i)] \quad , \quad i = 1,2, \dots, n \tag{2.8}$$

$$x' = [x'(p_1), x'(p_2), \dots, x'(p_i)] \quad , \quad i = 1,2, \dots, n \tag{2.9}$$

where n is the number of places existing in Petri Net.

Considering that there are m transitions in Petri Net, an m-dimensional triggering vector u is identified as follows:

$$u = [0, \dots, 1, \dots, 0] \tag{2.10}$$

Here, for the transition t_j to be triggered, the j^{th} element in the triggering vector takes value 1 whereas the rest takes 0. In case of triggering more than one transitions, the elements at related locations have to take value 1. Regarding the numbers of places and transitions, an incidence matrix D with $m\ x\ n$ dimension is defined as follows.

$$d_{ji} = w(t_j, p_i) - w(p_i, t_j) \quad , \quad i = 1,2, \dots, n \quad , \quad j = 1,2, \dots, m \tag{2.11}$$

As it can be clearly understood from (2.11) the incidence matrix is based on weight differences of arcs between an input place and an output place in aspect of a transition. Knowing that x is the current state vector and x' is the next state vector and benefiting from (2.10) and (2.11), the next state vector regarding a Petri Net is obtained as follows:

$$x' = x + uD \tag{2.12}$$

To explain (2.12) in details, it is better to investigate the example provided in Figure 2.3. The first state was found as $x_0 = [4 \quad 0 \quad 0 \quad 0]$. Before appyling (2.12), the

produced incidence matrix is given below:

$$D = \begin{bmatrix} -3 & 1 & 2 & 0 \\ 0 & -1 & -1 & 2 \end{bmatrix} \tag{2.13}$$

To comment about the incidence matrix found at (2.13), one may assume that the rows are related to the transitions whereas the columns are associated with the places. For the purpose of deciding the signs of the elements in the incidence matrix, the Petri Net given in Figure 2.3 should be examined. Basically, if a transition is an output transition of a place, then the regarding element in the incidence matrix takes value -1. If one investigate D_{11}, which gives information about the relationship between t_1 and p_1, it can be concluded that t_1 is the output transition of p_1. Since the weight of the related arc is 3, it is multiplied with -1 and D_{11} is found as -3. Otherwise speaking, if a place transfer coins through a transition, the related element in the incidence matrix is specified with minus sign. In case of having no direct connection between a transition and a place, the related element takes value 0.

At a time, more than one transition can be triggered if desired. However, the only condition for triggering a transition is that they must be activated. At the example in Figure 2.3, the only activated transition is t_1. As a result, the triggering vector u has only one option to be selected as, which is $u = [1 \quad 0]$. Putting everything into (2.12) gives the results below:

$$x' = [4 \quad 0 \quad 0 \quad 0] + [1 \quad 0] \begin{bmatrix} -3 & 1 & 2 & 0 \\ 0 & -1 & -1 & 2 \end{bmatrix} \tag{2.14}$$

$$x' = [4 \quad 0 \quad 0 \quad 0] + [-3 \quad 1 \quad 2 \quad 0] = [1 \quad 1 \quad 2 \quad 0] \tag{2.15}$$

As it is seen from (2.15), the same result as in the previous section has been achieved. To conclude, it can be said that the use of mathematical model derived in (2.12) alleviates difficulties in analyzing more complex Petri Nets.

2.5 Petri Net Classes

In the literature, Petri Nets usually are found in three primary classes which are ordinary Petri Nets, abbreviations and extensions [11].

The information given in previous sections forms the base of ordinary Petri Nets to a

large extent. In addition, it is assumed that a place can contain infinite numbers of tokens in ordinary Petri Nets. In otherwords, there is no a capacity limit on a place. Moreover, a time condition is not involved in ordinary Petri Nets [11].

The abbreviations can be accepted as the simplified images of ordinary Petri Nets. They are generally more beneficial so as to relieve graphical representation. Generalized Petri Nets, finite capacity Petri Nets and colored Petri Nets are the examples of abbreviations.

The extensions are associated with Petri Nets to which some operational rules are added. This is done for improving the original Petri Net, which enables numerous applications to be handled. Inhibitor arc Petri Nets, hybrid Petri Nets and stochastic Petri Nets can be given as a few examples of extensions [11]. Considering (2.1), the following classes can be identified.

2.5.1 Binary petri nets

A Petri Net is named binary if all weights are 1. Formally,

$$w: A \rightarrow \{1\} \tag{2.16}$$

An illustration of binary Petri Net is given in Figure 2.5.

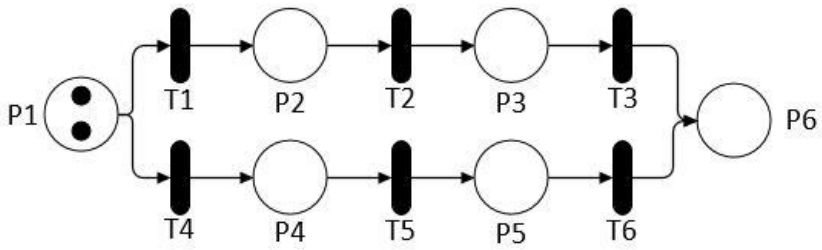


Figure 2.5 : Binary Petri Net example.

2.5.2 Petri net state machines

A Petri Net state machine belongs to binary Petri Nets. The only distinctive feature of this class is that each transition has exactly one input and one output place. Explicitly,

$$\#(p_i) = \#(p_j) = 1 \quad , \quad p_i \in I(t_m) \quad , \quad p_j \in O(t_m) \quad , \quad t_m \in T \tag{2.17}$$

In Figure 2.6, an example of Petri Net state machine is given.

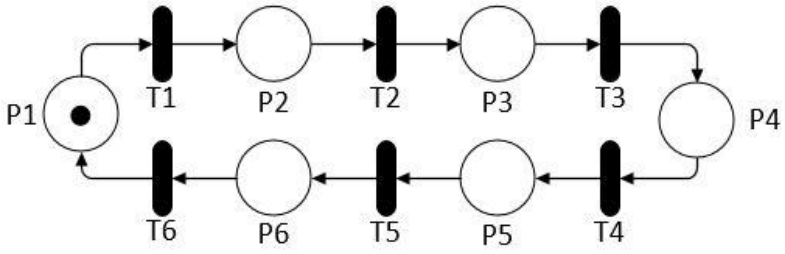


Figure 2.6 : Petri Net state machine example.

2.5.3 Free choice nets

A binary Petri Net is called a free choice net if one of the following conditions occur:

- A transition has more than one input place and each arc going out of the input places is unique for them.
- A place has more than one output transition and each arc going out of the place is unique for each output transition.

To clarify, the example in Figure 2.7 is shown.

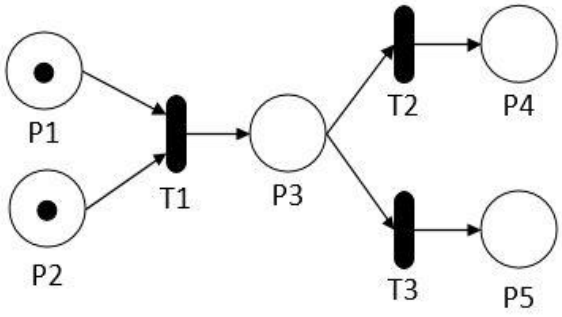


Figure 2.7 : Free choice net example.

As it can be obtained from Figure 2.7, the transition t_1 satisfies the first condition whilst the second condition is fulfilled by the place p_3. Here, for t_m to be enabled, both p_1 and p_2 have to contain a token. This is an example for synchronization activity. Moreover, as it can be seen that t_2 and t_3 share the same place. To prevent a conflict, both transitions should not be triggered at the same time.

2.5.4 Safe petri nets

A Petri Net is said to be safe if all markings are reachable M_0. Formally,

$$M(p_i) \leq 1 \quad , \quad \forall p \in P \tag{2.18}$$

For a clearer picture, one may define Petri Nets with an addition to (2.1), as follows:

$$PN = (P, T, A, w, M_0) \tag{2.19}$$

where M_0 is the representation of distribution of tokens at initial state. To put it in a formal way, $M_0: P \rightarrow \{0,1,2, \dots\}$ can be written.

A marking is reachable from another marking if there exists a sequence of transition triggerings starting from the original marking that result in the new marking. The reachability set of a Petri Net is the set of markings that are reachable from its initial marking M_0 through any possible triggering sequences of transitions. For a Petri Net to be a safe, its reachability set has to include all markings.

Tokens located in Petri nets generally represent resources in systems. Since having unlimited resources lead the system to instability, it is not a desired situation. Using the information given above, (2.18) can be interpreted as that each place in a Petri Net should not contain more than one token in order to satisfy the safety property. In Figure 2.8, an example about Safe Petri Nets is provided.

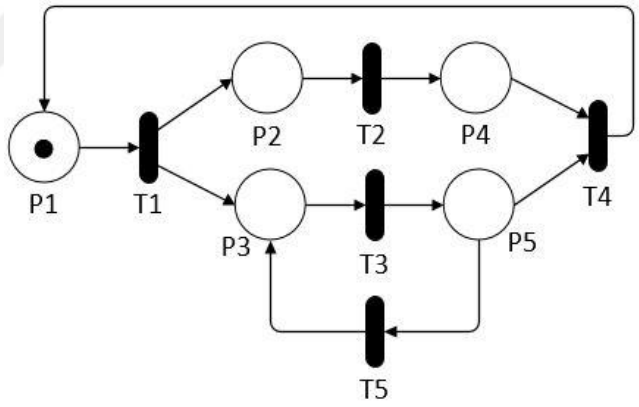


Figure 2.8 : Safe Petri Net example.

According to Figure 2.8, taking the initial marking M_0 as $M_0 = (1,0,0,0,0)$, any transition triggering sequence does not result in any place having more than one token. For example, if the triggering sequence occurs as $t_1 \rightarrow t_2 \rightarrow t_3$, the corresponding markings are $M_1 = (0,1,1,0,0)$, $M_2 = (0,0,1,1,0)$ and $M_3 = (0,0,0,1,1)$.

3. SAFETY AND REDUNDANCY

3.1 Definitions and Terms

Before going into details about redundancy and safety, it is beneficial to give the following definitions:

- Reliability: It expresses the ability of a technical equipment to perform its duty throughout the working cycle.
- Availability: It refers to the probability that a system is able to operate at a certain time.
- Fault Tolerance: It is the ability to function even if a system has a limited number of faulty components.
- Hardware Fault Tolerance (HFT): The hardware fault tolerance "N", $N \rightarrow (0,1,2,\dots)$, is the limit that $N+1$ fault may result failure in safety function [12].
- Safe Failure: It is a kind of failure that has no effect on a safety function to switch a system into a safe state or maintain the safe state when demanded. A safe failure can give rise to loss of production at most. Safety is preserved.
- Dangerous Failure: It is the failure that a safety function is affected which may result in reaching a safe state when requested is probably prevented. As a consequence, a dangerous failure may lead to loss of safety.
- Probability of Failure on Demand (PFD_{avg}): It means the chance of an error at the request of a safety function. In other words, it is the probability of a failure of a system when needed. Smaller value of PFD is a signature of better system.
- Mean Time Between Failure (MTBF): It is the foreseen time between errors. It is a statistical value and its unit is hours. It is investigated for an extended period and with many units [13].

- Safety Integrity Level (SIL): It defines both hardware integrity and standardized safety integrity in safety associated systems. Its level is measured from 1 to 4 [12].
- Performance Level (PL): It is the representation of the assessment of a safety quality for the execution of safety function. Its level is apportioned into steps from a to e [13].
- Safe Failure Fraction (SFF): It is the indication of non-dangerous failures or dangerous failures detected by some tests, in percentage.
- Redundancy: It means the presence of more operating resources than necessary for the function to be performed in a system.
- Safety: It refers the reliability in terms of critical failures. A critical failure is harmful, unlike a non-critical failure which can be classified as benign. A critical failure can pose a hazard to the material, environment or people, which may cost several times higher than the normal operating costs of a system.
- Fail-safe: It is the ability of a system to keep a safe state or instantly shift into different safe state right after some failures take place.
- Functional Safety: It means detecting a possible hazardous situation which gives rise to an action in order to avoid or reduce the outcomes of that dangerous event. As an example, the detection of smoke with the help of sensors and consequently activation of fire suppression system can be given.

3.2 Safety Instrumented Systems

A set of elements that interact with each other according to a structural design creates a system. Here, an element can be another system called a sub-system. It is possible that this subsystem is a controlling or controlled system.

A SIS includes of at least three subsystems consisting of an input element designed for certain reasons, an output element designed to perform certain tasks, and a logic system used to establish the necessary relationships and calculations between the input element and the output element. Conditions related to the security of the whole system should be considered separately for both the system and its subsystems.

The input element usually consists of sensors and transducers. These elements are responsible for detecting potential hazards and sending the relevant electrical signal to the logic element. Pressure sensors, temperature sensors, optical sensors can be given as examples. As the next link of the chain, the logic element evaluates the signals from the input elements and sends commands to the output elements via electrical signals in case of undesirable situations. A logic element can be a computer, a relay circuit or a programmable logic controller (PLC). Finally, the task of the output elements is to execute commands from the logic unit. Circuit breakers, shut-off valves, alarm alerts are some examples [14]. In case of a failure in the system, the ability of these subsystems to perform their duties in full and in order is associated with the related system being fail-safe.

For a system being fail-safe is a design feature and means that it immediately switches to a predefined safe state in the event of a particular failure. fail-safe systems allow the operation of an industrial system within predetermined safe limits, and when it goes beyond these limits, it takes the relevant system to a safe state. In other words, these basic tasks undertaken by fail-safe systems can be considered as a risk reduction function. Reducing the risk sometimes takes place by lowering the possibility of a hazard, and sometimes by changing the magnitude of the outcome [15].

3.3 Functional Safety Standards

A SIS is designed to reduce the likelihood of events that may have dangerous consequences or to reduce the risk of the results of these events. To do so, this kind of systems have to meet certain standards that express some necessary requirements. Certification process is directly proportional to maintaining the relevant standards, and it is done by responsible certification authorities. Many different commissions work on the development and implementation of standards. Organizations such as International Electrotechnical Commission (IEC) and International Organization of Standardization (ISO) have prepared internationally valid standards, which have been recognized by Turkish Standards Institution (TSE) in Turkey.

3.3.1 IEC 61508

IEC 61508 is a set of international standards set for electircal, electronic and programmable electronic safety associated systems. This standard collects systems

under 4 different risk groups according to their probable hazard potential. This concept, called SIL, is composed of 4 different levels that indicate the reliability levels of electrical or electronic devices [13]. The increasing level from SIL 1 to SIL 4 is the indication of having more reliable systems. Each SIL level corresponds to PFD_{avg}, which indicates the probability that a system fails. The SIL level can be specified for the whole system or separately for all components of the system. The relationship between SIL and PFD_{avg} is given at Table 3.1.

Table 3.1 : SIL and PFD_{avg} relation.

SIL	PFD_{avg}
1	$\geq 10^{-2}$ to $< 10^{-1}$
2	$\geq 10^{-3}$ to $< 10^{-2}$
3	$\geq 10^{-4}$ to $< 10^{-3}$
4	$\geq 10^{-5}$ to $< 10^{-4}$

In IEC 61508, many methods are mentioned for the quantitative detection of risk reduction and the calculation of the required SIL value. In order for determining the SIL value of a system, it is not sufficient to calculate PFD_{avg} of the system alone. In addition to the mentioned methods, there are 3 other parameters that must be examined to determine the SIL value of a system. These parameters are SFF, HFT and whether the system is comprised of type A or type B components. SIL value can be obtained with the help of Table 3.2.

Table 3.2 : Determining SIL against SFF,HFT and type of components.

SFF	Type A Components			Type B Components		
	HFT 0	HFT 1	HFT 2	HFT 0	HFT 1	HFT 2
$< 60\%$	SIL 1	SIL 2	SIL 3	N.A	SIL 1	SIL 2
$> 60\%$ to $< 90\%$	SIL 2	SIL 3	SIL 4	SIL 1	SIL 2	SIL 3
$> 90\%$ to $< 99\%$	SIL 3	SIL 4	SIL 4	SIL 2	SIL 3	SIL 4
$> 99\%$	SIL 3	SIL 4	SIL 4	SIL 3	SIL 4	SIL 4

According to Table 3.2, said type A components are the ones such as transistors or resistors whose failure modes are known under certain conditions. Type B components, on the other hand, are more complex and unpredictable components, such as microprocessors. In order to evaluate a system's SIL value according to B type, it is sufficient that even one of the components in the system is B type [13].

Secondly, HFT refers to the maximum number of dangerous failures that can occur without disrupting the system's functionality. It means also a degree of redundancy of a system. While the HFT value is 0 for systems without redundancy, it can be equal to 1 and 2 according to the structure of the system. In the next section, this subject is explained in more detail. Finally, the SFF variable is the rate at which the system goes into safe failure. For example, for a system with an SFF of 90%, only 10% of possible errors can make the system dangerous [13].

3.3.2 IEC 62061 and ISO 13849

Both standards set requirements for the design and implementation of safety-related machinery systems. The methods developed in these standards differ from each other. These standards categorize systems that perform safety-related functions according to PFD_{avg} [16]. Whereas ISO 13849 has 5 PLs, from a to e, IEC 62601 has 3 SILs, from 1 to 3. When evaluating the PL of a system, the severity of the hazards that may occur, the length of the exposure time and how much it is possible to avoid hazards are taken into account. In terms of degree of risk for a system, it increases from PL_a to PL_e.

A designer may choose between these standarts according to certain characteristics of application when realizing a machinery system. Without regard to used methods, it is significant to guarantee that SIL and/or PL is established by making suitable decisions on risk parameters [16]. In the Table 3.3, the relationship PL, SIL and PFD_{avg} is shown.

Table 3.3 : Correlation of PL, SIL and PFD_{avg}.

SIL	PL	PFD_{avg}
N.A	a	$\geq 10^{-5}$ to $< 10^{-4}$
1	b	$\geq 3.10^{-6}$ to $< 10^{-5}$
1	c	$\geq 10^{-6}$ to $< 3.10^{-6}$
2	d	$\geq 10^{-7}$ to $< 10^{-6}$
3	e	$\geq 10^{-8}$ to $< 10^{-7}$

3.4 Architectural Structures of Systems

IEC 61508 standard defines the architecture of a system as placing the hardware and software elements in that system according to a certain configuration. PFD_{avg} value is calculated taking into account the hardware architecture for the entire system and each subsystem connected to it. This value, as known before, gives an idea about the performance of the system in connection with the system being free from failures.

The architectural structure of a system is shown as *MooN*. This display means that the safety function of at least *M* of *N* channels in the system is sufficient for the operation of the system. For example, *1oo2* structure indicates that the system can work properly if at least one of the two channels in the system performs its function. Generally, all channels in such structures are independent channels that perform the same function. The elements in this channel can be a logic system or input/output elements.

3.4.1 *1oo1* Structure

1oo1 is a structure that contains only one channel. Compared to others, it is the cheapest and simplest to preserve among them. If safety objectives set for a system can only be accomplished with a single element, this structure is suitable for use. In Figure 3.1, the representation of the mentioned structure is given.

Figure 3.1 : 1oo1 structure.

In a SIS, recently many system elements are designed according to the "de-energize-to-trip" principle. This principle is to trigger the relevant element when the existing energy in the system is cut under normal conditions and to switch the system into a safe state [14] For this reason, this principle is the most important basis for fail-safe concept. The most significant disadvantage for this structure is having no tolerance in case of any failure taking place.

3.4.2 *1oo2* structure

In *1oo2* structure, unlike the previous one, there are 2 independent channels which are connected in a parallel manner. In a system set up in this way, it is sufficient for a channel to perform its function in order for the system to maintain its operation. In this structure, both channels must be defective at the same time for the system to be in dangerous state. Thanks to this configuration, system reliability is enhanced due to lowering of dangerous states. In Figure 3.2, the demonstration of this structure is given.

In case of detectable or undetectable dangerous failures happening in both channels at the same time, the system fails dangerously. The main disadvantage of this structure

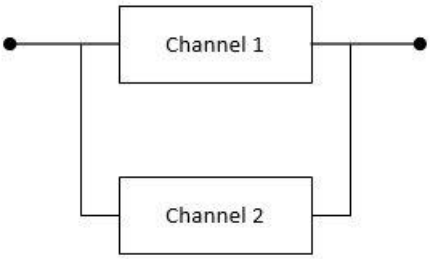


Figure 3.2 : 1oo2 structure.

is having decreased system availability because of the increase in general failure rate.

3.4.3 *2oo2* structure

2oo2 is the simplest structure to ensure high availability in dual controller systems. In this structure, the channels are connected in series to each other. In order to maintain the operability of the system, both channels must fulfill their duties without any failures. In Figure 3.3, a graphical illustration of the mentioned structure is given.

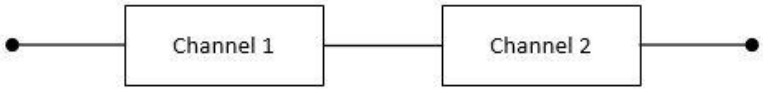


Figure 3.3 : 2oo2 structure.

Having a failure in any channel or both channels in this structure leads to system failure. Consequently, it can be said that such systems are not tolerant any failure which may occur in any channel. Although this is the similar feature as in *1oo1* structure, a higher availability is achieved thanks to this structure. However, as a downside, system reliability is decreased.

3.4.4 *2oo3* structure

2oo3 is a safe structure consisting of 3 independent channels. These channels are connected in such a way that the functioning of any two channels is sufficient for the continuity of the system. In case of a failure in one of the channels, the system goes into safe state and continues to operate. In Figure 3.4, this structure is shown. Possible dangerous states in *1oo2* structure are reduced, whereas in *2oo2* structure, the probability of occurance of dangerous states is reduced. In other words, *1oo2* structure provides higher reliability, while *2oo2* structure provides higher availability. Taking

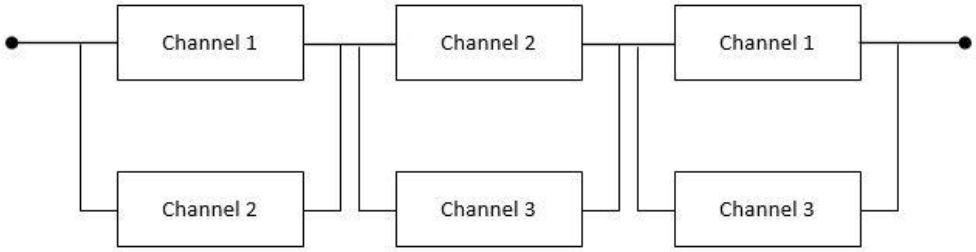


Figure 3.4 : 2oo3 structure.

these into consideration, *2oo3* structure can be thought of as a combination of the two previous structures. So, thanks to this structure, both higher availability and higher reliability can be provided.

In theory, assuming that every independent channel includes identical units, these structures can be compared to each other in terms of HFT and SIL as given in Table 3.4.

Table 3.4 : Relation between structure, HFT and SIL.

Structure	HFT	Maximum SIL
1oo1	0	SIL 2
1oo2	1	SIL 3
2oo2	0	SIL 2
2oo3	1	SIL 3

As a result to be drawn here, the structure of a system can be adjusted to have a better SIL value according to the application to be made.

3.5 Redundancy Principle

In engineering, the term redundancy is essentially means pairing of systems, components or functions that are critical. Generally, while a system performs a task, one-to-one copy with the same features waits as a backup at the back. The main purpose here is to increase the performance of the system as well as to increase the reliability of the system. Some vital systems, such as hydraulic systems in airplanes, are copied in triplicate, called triple modular redundancy (TMR) in the literature [17]. In case of more than three components involved in a redundant structure, it is called N-modular redundancy. In spite of increasing the reliability of a system, proportionally increase in the cost and complexity of the system can be said as the disadvantages of redundancy. Usually, a redundant structure is not required for the successful execution

of many applications, but the high cost of a possible error can be seen as a sufficient reason for using such a structure.

In aspect of safety related systems, a redundant structure is crucial in order to maintain the functionality of a system in case of a failure. Therefore, it can be said that for a system failure to take place, all components included by the redundant structure must fail [13]. Redundancy can be categorized into two groups according to the components utilized. If all the components are identical in a redundant strucutre, it is called as "Homogeneous Redundancy". Otherwise, it gets the name "Heterogeneous Redundancy". In Industrial Automation, with the assumption of identical components, mainly there are three kinds of redundancies:

- Active Redundancy: It consists of the presence of two components that perform exactly the same tasks. However, in the implementation of the system, decisions made by only one component are taken into consideration. In such an arrangement, a designer can assign priority to one of these two components with the same feature. Since both components are in operation, if an error occurs in the component that is already running the system, the backup component immediately takes over the system and starts sending commands. In this way, the operability of the system is not subjected to any interruption. Active redundancy is used in many safety related systems exclusively.
- Passive Redundancy: It is the structure consisting of two idetical components, but one of them waits without doing anything while the other performs its duty. In case of an error in the component running the system in this structure, switching between components does not occur automatically. In other words, an external system is needed that activates the backup component and deactivates the main component. It is not suitable for use in safety critical systems. As an example, using a spare tire on a vehicle in the event of a flat tire can be given.
- Majority Voting Redundancy: It is a special case of M-out-N redundancy. In this case, the majority of components must function so as to secure the functionality of a system. This structure is often used when it is difficult or impossible to detect the failure of a component from the outputs produced. To solve this problem, the output produced by the defective component is

compared with the outputs produced by other components. It can be classified into two categories:

- Non-Adaptive Majority Voting: In this case, all the components are involved in voting action without regarding that they are defective or not.
- Adaptive Majority Voting: Here, the faulty components are excluded from voting action. Such systems are generally used in need of proper outputs.

The concept of redundancy is handled differently in various articles. For example, if an external system is needed to switch between components, this is called as "Standby Redundancy" [18]. This type of redundancy consists of 3 main categories: hot standby, warm standby and cold standby.

Hot standby is the case that switching between the components takes place in milliseconds. Otherwise speaking, it is the case that the switching system is excellent. Broadly, hot standby is associated with active redundancy. In addition, slightly different than hot standby, the switching occurs in seconds in warm standby. It can be related to active redundancy; however, it is less effective when compared to hot standby. Lastly, cold standby can be said as the exact equivalent of passive redundancy since it requires manual switch-over.

4. APPLICATION ON A CONVEYOR BELT SYSTEM

In this part of the thesis work, the sample application on a real conveyor system will be explained by blending the information given in the previous sections. The mentioned conveyor belt can be seen in Figure 4.1.

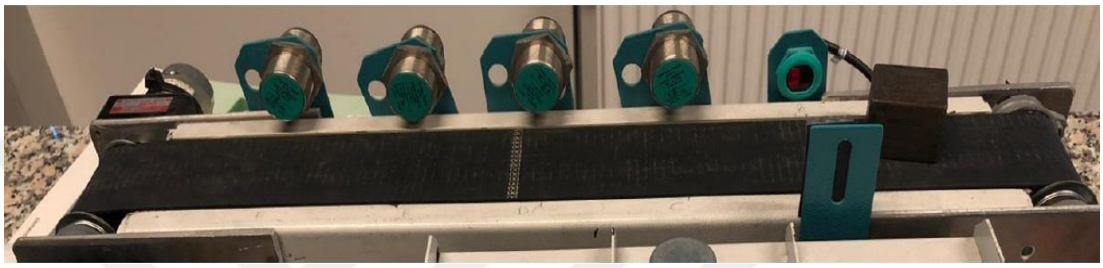


Figure 4.1 : Conveyor Belt.

In the conveyor belt system given above, there is a DC motor, 1 optical sensor and 4 inductive sensors. The presence of inductive sensors obliges the objects on the band to be metal, because these sensors can only detect metal objects. In contrast, there is no such restriction for capacitive sensors. The optical sensor, on the other hand, works on the principle of being able to receive the beam that it emits thanks to the reflector placed opposite of it. In this project, this sensor is used to measure the object length. The DC motor on the system can be driven to rotate both clockwise and counter-clockwise direction when necessary.

Excluding the external buttons or switches, it can be said that the system consists of 5 inputs and 2 outputs. Optical sensor and 4 inductive sensors are connected to the input module of PLC, whereas the outputs from DC motor are connected to the output module. When a metal object is in range of an inductive sensor, logic 1 is read from the related input channel. Without an object interruption, the optical sensor gives logic 1 signal since it is able to receive the emitted beam. When interrupted, logic 0 is immediately read from the related input channel. Lastly, when demanded, the DC motor can be rotated both clockwise or counter-clockwise direction by just making the relevant output channels to be logic 1. For a clearer picture, details about PLCs will be given in upcoming sections.

4.1 Definition of PLC

A PLC is basically a specialized computer that can be programmed for automation or electromechanical processes such as railway signaling, control of machinery of factory assembly lines, oil refineries, water & wastewater treatment plants, chemical plants and so on. But mostly, PLC-based controllers are encountered in many areas of daily life such as traffic lights, elevators and even home security systems. Generally, a PLC consists of three parts: inputs, outputs and central processor unit (CPU). In Figure 4.2, a general representation of the working principle of a PLC is given.

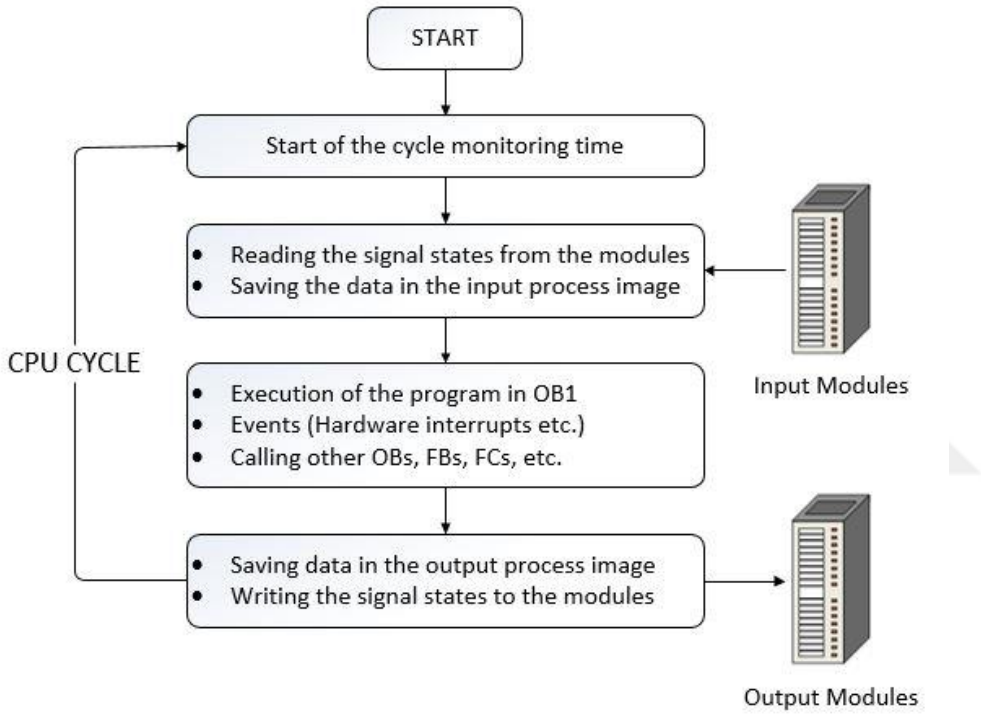


Figure 4.2 : Working Principle of a PLC.

The CPU is a very important element here; since, it receives commands from those inputs, processes these signals according to the specific program designed on a computer, and transmits the necessary commands to the outputs. The CPU acts as a hub for all activities that take place in a PLC. The inputs translate signals into a language the CPU can understand so that the CPU can process them. As an example, a push-button can be an input of a process. When it is pressed, it generates an electrical signal (generally 24 Volts) to the PLC inputs. According to the application, this can be a current or voltage signal. This electrical signal pass through the PLC inputs and is converted into a logical data since the CPU is not able to process raw electrical signals.

After executing, CPU sends required data to PLC outputs. As a contrary to the initial procedure, the data is converted into an electrical signal by PLC outputs for use in industrial processes. To illustrate, in an industrial process, an electric motor can be a PLC output via a contractor. In order to turn the motor on or off, the contractor must be triggered electrically so that it is activated or deactivated. As a result, CPU's logical data is sent to the contractor in form of an electrical signal. As similar with inputs, output signals can be either current or voltage signal.

4.1.1 PLC models used in the project

In this application, 3 PLCs belonging to Siemens were used. Two of these PLCs are the identical S7-1214C DC/DC/DC models. The other PLC is S7-1214FC DC/DC/DC which is a fail-safe PLC. It is important to note that the fail-safe PLC was used as a standard PLC in the project. In other words, any safety feature of this PLC has not been utilized. This PLC acts as a voter in the project and communicates directly with the conveyor belt. The other two standard PLCs are responsible for executing the written programs of the system. As a result, it can be accepted that this project is realized with 3 standard PLCs. In Table 4.1 , some technical information about this standard PLC can be found [19].

Table 4.1 : Some information about the PLC used in the application.

General Information	**S7-1214C DC/DC/DC**
Engineering with	STEP 7 V11 SP2 or higher
Supply Voltage	24 VDC (20.4V-28.8V permissible range)
Work Memory	75 KB
Load Memory	4 MB
Interface Type	PROFINET (1x RJ45)
Digital Inputs	14 channels
Digital Outputs	10 channels
Analog Inputs	2 channels
Analog Outputs	0
Approximate Weight	415 g
Degree of Protection	IP20

In the project of the thesis work, the inputs and outputs of the conveyor belt system are not directly connected to the controller PLC. There is an interface module (IM 155-6 PN/2HF) which connects distributed input/output system with Process Field Net (PROFINET).

4.1.2 PROFINET communication protocol

PROFINET is an industrial communication protocol used extensively by Siemens control systems. It is specified by PROFIBUS and PROFINET International (PI), and is member of IEC 61158 and IEC 61784 standards. The physical interface used for PROFINET is a standart RJ45 ethernet jack. PROFINET cables are easily distinguishable by their green color. However, in some cases, a standart ethernet cable can be used to communicate PROFINET devices. Though, as an advantage of official PROFINET cables over standard ethernet cables, they have greater shields and isolation, which provides more durability under industrial conditions and more precise and noise-free communication. PROFINET operates at 100 megabits per second, and cables may be up to 100 meters length according to applications. Due to its response time which is less than 1 millisecond, PROFINET is ideal for high-speed applications. Another great advantage of PROFINET is that it uses the same physical connection standards as Ethernet. By this way, standard ethernet switches can be used to expand the network. As shown at Figure 4.3, PROFINET may be used to connect a variety of devices including PLCs, Human Machine Interfaces (HMIs), servo controllers, distributed I/O racks, temperature or pressure sensors and more all on one network.

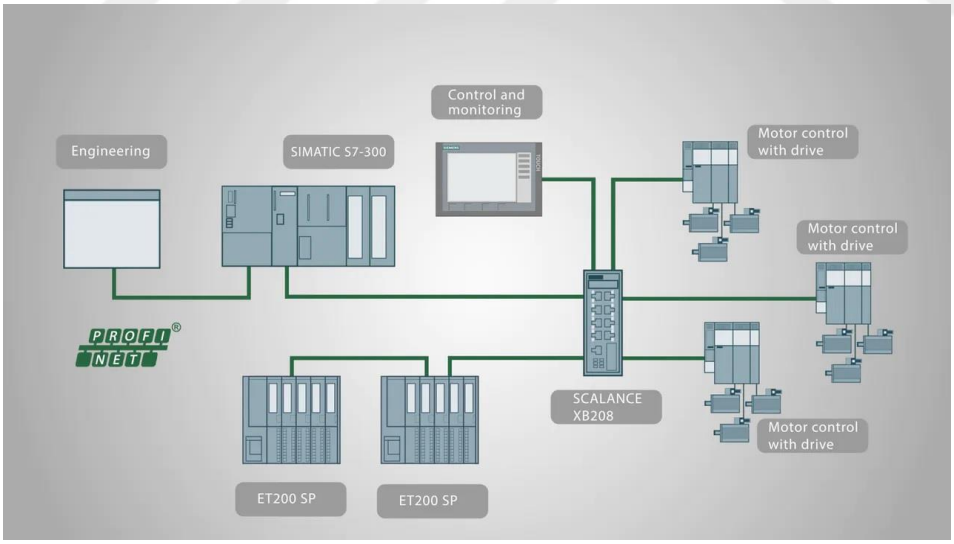


Figure 4.3 : A PROFINET newtork example.

Furthermore, there are some specified PROFINET profiles for particular areas. For instance, PROFIdrive is for characterizing essential properties for drives and encoders. Additionally, PROFIsafe is the profile for safety communication and enhancing performance in Functional Safety applications. Lastly, PROFIenergy is intended for energy management applications in production systems.

4.1.3 Programming languages and methods

PLC programming is under an international standard which is determined by IEC. IEC 61131 is the standard for PLCs, and its third part, also known as IEC 61131-3, is related to PLC programming languages. According to the IEC 61131-3, programming languages for PLCs are categorized into 3 groups: textual, graphical and others. In Table 4.2, the mentioned programming languages are illustrated.

Table 4.2 : PLC Programming Languages.

Textual	Graphical	Others
Structured Text (ST)	Ladder Diagram (LD)	Sequential Function Chart (SFC)
Instruction List (IL)	Function Block Diagram (FBD)	Continuous Function Chart (CFC)

In this thesis work, LD is chosen in order to program the PLC. Also, there are various programming methods such as asynchronous sequential circuits, synchronous sequential circuits, discrete event system approach, and so on. Petri Net is a well-known design method of discrete event system approach. In this project, Petri Net is benefited so as to design the conveyor belt to program in PLC.

4.2 Conversion from Petri Net Design to PLC Code

PLCs play a big role in the implementation of industrial automation systems. It is very important that coding is done in a reasonable way in order for a system to be performed correctly. While coding process, heuristic or formal methods are preferred. However, formal methods are used more because heuristic methods are not based on any systematic basis. It is the 3 steps of the formal method to create a flow diagram regarding the operation of the system, to design a suitable Petri Net model and to convert it into PLC code within the framework of certain rules. Set & Reset method is one of the most used formal methods for conversion from Petri network to PLC code [20,21].

4.2.1 Set & reset method

The most important reason for using the set & reset method is its practicality. According to this method, "set" command is used for putting one token to a place which does not contain any token whereas "reset" command is for taking a token away

from a place. As it can be understood from here, this method is valid for safe petri nets, which is explained in section 2.5.4, since the transactions are always made with one token. For a better understanding, considering the safe petri net example given before in Figure 2.8, a part of relevant PLC code is given in Figure 4.4.

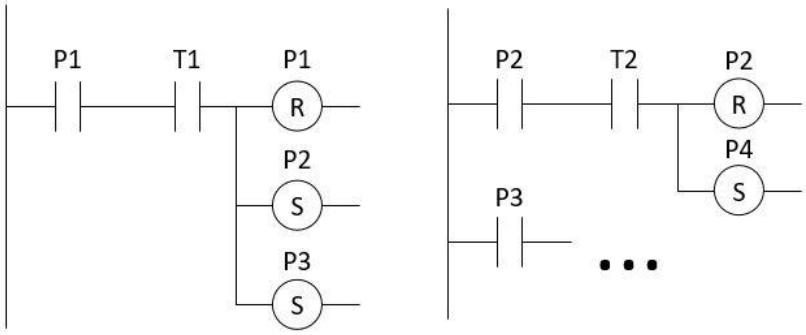


Figure 4.4 : A PROFINET newtork example.

When the given PLC code is examined, in the case that there is a token in p_1 and the transition t_1 is triggered, the token is taken with the "reset" command and transferred to p_2 and p_3 places with the "set" command. After that, in the event of triggering t_2, the token is taken away from p_2 and put into p_4 using the same commands. To summarize, the transfer of coins from one place to another is done in this method using "set" and "reset" commands.

4.2.2 Logical expressions method

Although the Set & Reset method described in the previous section is practical, it has certain disadvantages. Firstly, the "set" and "reset" commands occupy a significant space in PLC memory compared to normal bit operations. For this reason, realizing large-scale Petri Nets with this method can put an extra load on the memory and processor of PLCs. In addition, a code created by this method is difficult to understand and follow. Therefore, it will be equally difficult to detect errors that may occur in the code. Finally, considering a large-scale Petri Net, a large number of "set" and "reset" operations are required. Therefore, some of them may easily be forgotten. As a consequence, it may cause serious errors in the operation of the system. Logical expressions method is a proposed method to eliminate all these disadvantages [22].

In order to apply this method, logical expressions related to places and transitions, which are the two main components of a Petri Net, should be obtained separately. For

a transition t_j to be triggered, all the input places of the transition must contain a token. In addition, a logical condition S_j can be bounded to the transition. Consequently, for a triggering action, this logical condition must hold too. Considering that the transition t_j has n input places, the logical expression can be given as follows [22].

$$t_j = \prod_{i=1}^{n} p_i S_j \tag{4.2}$$

In (4.1), p_i is the indication of whether a token is included or not. p_i is logic "1" if it includes a token, otherwise it is logic "0". Simirlarly, being logical conditions on t_j satisfied makes S_j logic "1". When extracting logical expressions about places, both input and output transitions have to be taken into account. As remembered, a triggered input transition results in a token transferred into the relevant place. Similarly, a triggered output transition gives rise to a token to be taken away from that place. Considering this information, the logical expression for a place p_i which has n input and $m - n$ output transitions can be given as follows $(m > n)$ [22].

$$p_i{}^{+} = \sum_{i=1}^{n} t_i + p_i \prod_{j=n+1}^{m} \overline{t_j} \tag{4.2}$$

As it can be understood from (4.2), in order to transfer a token into p_i it is sufficient for any transition from t_1 to t_n to be triggered. Likewise, triggering any transition from t_{n+1} to t_m is enough to take a token from p_i. As a result, the disadvantages caused by the previous method can be overcome with logical expressions method. It should also be noted that this method is valid under the assumption that each place can hold a maximum of 1 token. For clarity, a Petri Net example in Figure 4.5 is given.

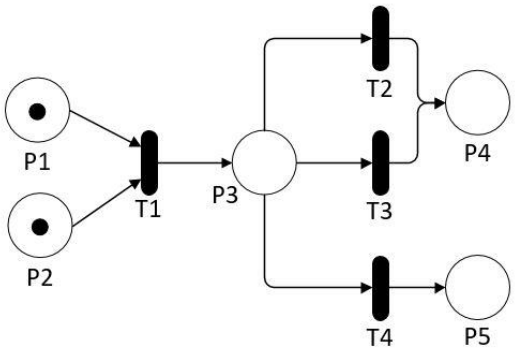


Figure 4.5 : A Petri Net example for logical expressions method.

As it can be seen from Figure 4.5, the Petri Net consists of 5 places and 4 transitions. Following the procedure, logical expressions related to transitions should be obtained first. According to (4.1), they can be obtained as follows.

$$t_1 = p_1 p_2 S_1 \tag{4.3}$$

$$t_2 = p_3 S_2 \tag{4.4}$$

$$t_3 = p_3 S_3 \tag{4.5}$$

$$t_4 = p_3 S_4 \tag{4.6}$$

As the next step, logical expressions about places should be obtained. According to (4.2), they can be constructed as below.

$$p_1{}^+ = p_1 \overline{t_1} \tag{4.7}$$

$$p_2{}^+ = p_2 \overline{t_1} \tag{4.8}$$

$$p_3{}^+ = t_1 + p_3(\overline{t_2 t_3 t_4}) \tag{4.9}$$

$$p_4{}^+ = t_2 + t_3 \tag{4.10}$$

$$p_5{}^+ = t_4 \tag{4.11}$$

As it can be seen from the example, using this method, the conversion from Petri Net to PLC code can be realized quickly and effectively. It should be noted here that the same memory space can be used in PLC for the next state of a place, denoted by $p_i{}^+$, and the current state, denoted by p_i. In the light of the given formulations, it is impossible for these two expressions to be different from each other at the same time; hence, using the same memory area will not cause any problems.

4.3 Petri Net Design of the Conveyor Belt

Before designing a system with a Petri Net, the elements that the system contains and how the system should work have to be determined. Drawing a suitable flow diagram after these steps definitely facilitates Petri Net design. Due to the elements of the

conveyor belt system, the operations on the PLC side is bitwise. In Figure 4.6, the flow diagram of the conveyor belt system is given.

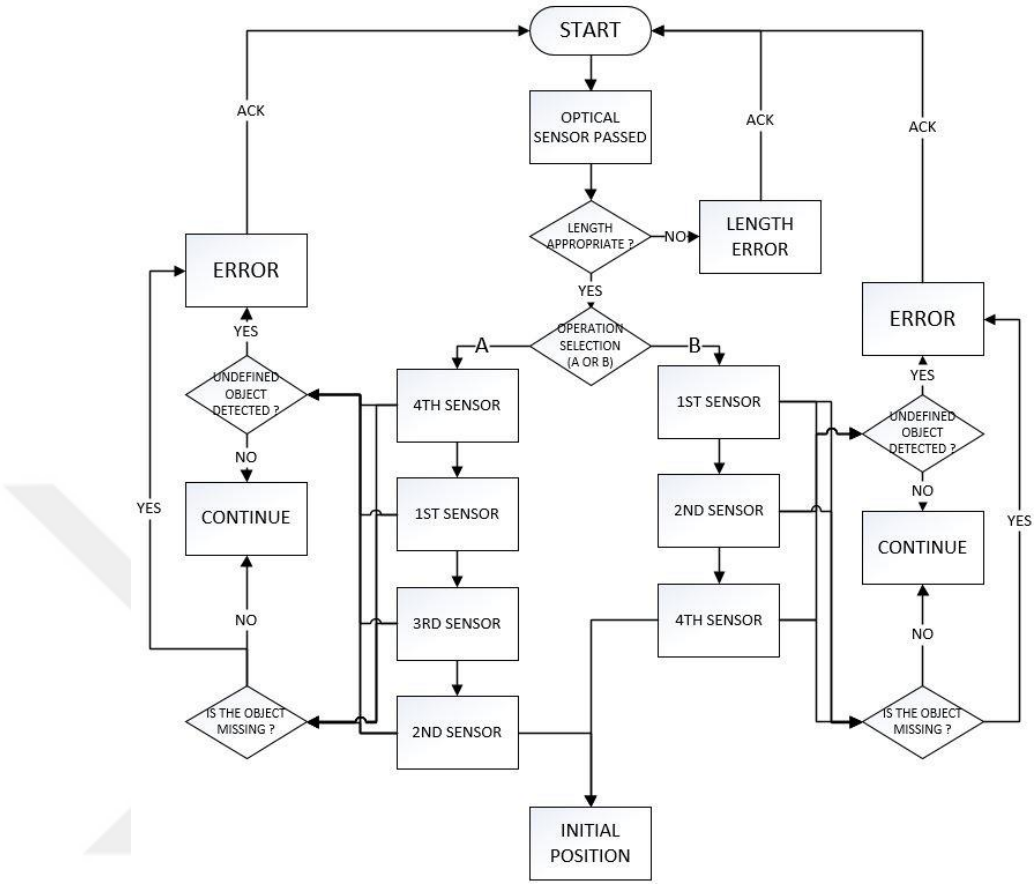


Figure 4.6 : Flow diagram of the conveyor belt system.

According to the flow diagram of the conveyor belt system, the system is composed of 4 processes: measuring the length of the object, selecting operation for the object, detecting for undefined object and detecting whether the object is missing or not. Firstly, according to the design, the minimum and maximum length limits for the object to be processed in the conveyor and according to which operation the object will move in the conveyor are determined by the operator before starting the system. After the system has started, its length is measured as the object passes in front of the optical sensor. As the object passes in front of the optical sensor, the length measurement process starts with the first falling edge signal of the sensor and ends with the rising edge signal. The time taken between these two signals is measured and multiplied with the constant speed value of the DC motor. The result is the length of the object. If the measured length is within the range it should be, the object continues on the conveyor. Otherwise, the object is thrown back from the conveyor. According

to the selected operation, the object goes to the sensors in a certain order and waits for the predetermined time there. After the actions required by the selected operation are completed, the object goes to its starting position, that is, behind the optical sensor, and the operation is finished.

When the conveyor belt system is operating according to a specified operation, the condition of having only the relevant object on the conveyor applies. If a second object is detected by any sensor in the system during the process, the system falls into an error state, the operation of the conveyor switches from automatic mode to manual mode, and the operator responsible for the process thus throws the unwanted object out of the conveyor. Finally, after the related error is removed, the acknowledge signal is sent to the PLC and the conveyor system is restarted. In addition, if any object is detected by the sensors on the conveyor before the system starts, the system will not operate unless this object is removed from the conveyor. Thus, the harmful consequences that may arise from any unidentified object before the system is operating or while the system is running is prevented.

Another important point is that the object can complete the process on the conveyor without any problem according to the determined operation. For this reason, a decision have to be made about whether the object is missing or not, based on some physical features of the system. These features are the distances between the sensors on the conveyor and the constant speed of the DC motor. Accordingly, a reasonable time limit is determined and if the object cannot be detected by any sensor for the specified period while the system is running, it is decided that the object is missing. This can happen either because the object is really missing, or because of a malfunction in the sensors. For this reason, in order not to ignore the sensor failure case, the system switches from automatic mode to manual mode in case of missing object error, the necessary actions are taken by the operator and the error is eliminated. Finally, the acknowledge signal is sent to the PLC and the system is restarted from the beginning.

After drawing the flow diagram on how the system should operate and determining the conditions for the system to run smoothly and continuously, the system is ready to be converted into a Petri Net design. In Figure 4.7, the complete Petri Net design of the conveyor belt system is given. In the given Petri Net design, the parts related to the 4 basic processes carried out in the system are shown in color. In addition, while the

object is moving on the conveyor, the relevant transitions and places have been added so that the system can be paused and resumed by the operator.

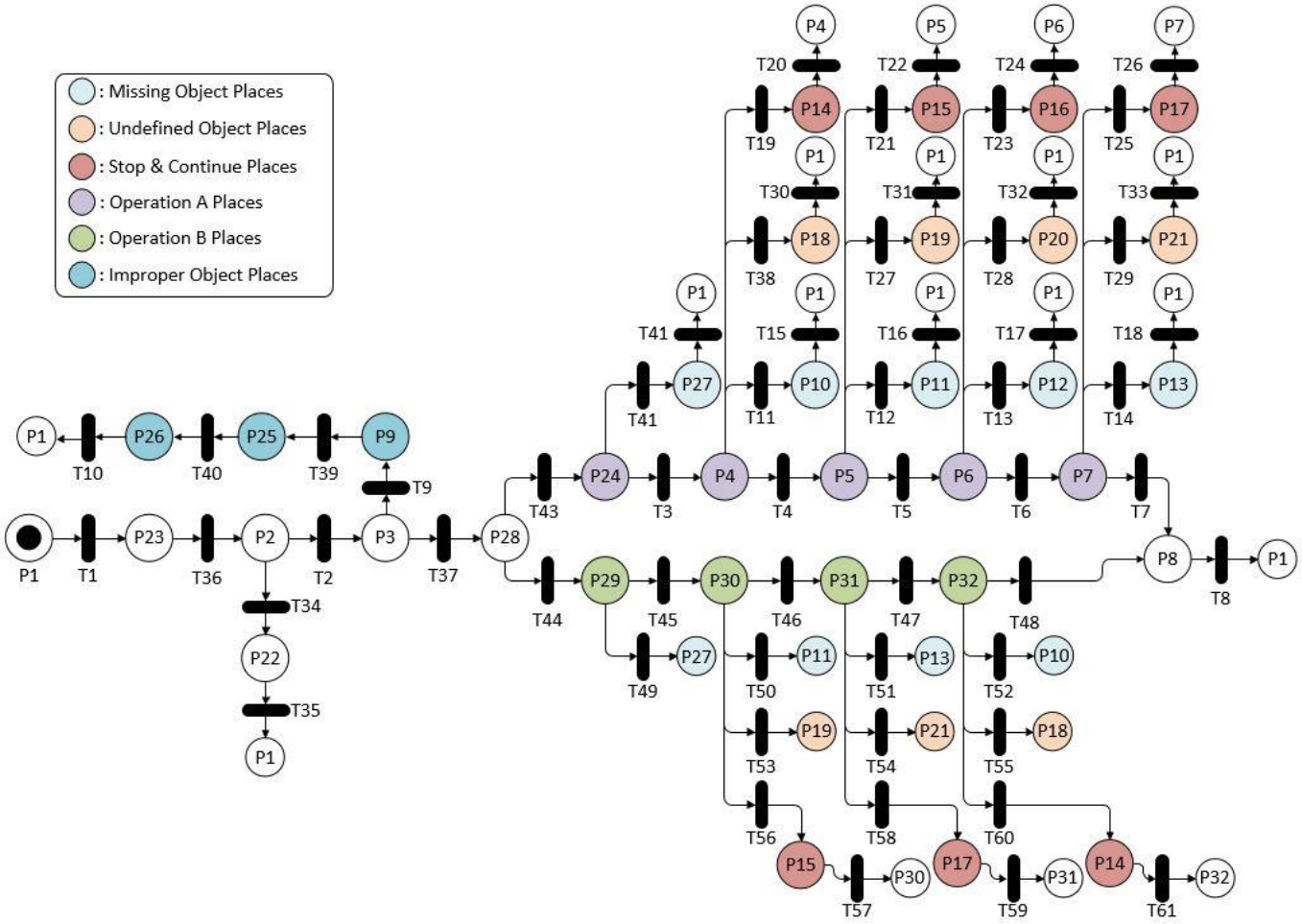


Figure 4.7 : Petri Net design of the conveyor belt system.

In conformity with the information given in the section 2.5, the Petri Net designed for the conveyor belt system satisfies the "Binary Petri Net" feature since all the weights of arcs are 1. Furthermore, the "Free Choice Net" property is fulfilled as well, since places like p_4, p_5, p_{24}, p_{30} and so on have more than one output transitions. For the sake of functionality, possible conflicts on triggering such transitions are prevented while programming. Otherwise speaking, more than one output transitions belonging to the same input place can not be triggered at the same time. Lastly, the "Safe Petri Net" feature is also met since it is impossible for any place in this Petri Net to contain more than one token.

Upon completion of the Petri Net design, it is possible to program the system in PLC using the information given in the section 4.2.2. With the flexibility feature of Petri net, it is very easy to make changes in the design. The fact that Petri Nets have a graphical structure and can be easily monitored helps the detection of the error in the event of an error in the system.

4.4 Establishing Redundant and Safe Configuration

In industrial automation, how control units of a system work has a great importance. For example, it is undesirable to stop the process in a 24/7 factory. If there is only one unit controlling, the process will stop in case of any failure. For this reason, backing up the control unit is a solution that eliminates this problem. Some control units have more than one backup in vitally important systems such as aircraft. As explained earlier, having more than one unit in a system that can do the same is the basis of the redundancy principle.

On the other hand, in some critical systems, a failure that may occur in a control unit can be harmful to the environment and even people. If there is only one unit controlling the system, these results may not be avoided. For example, if a unit controlling the refrigerant unit in a nuclear power plant performs erroneous calculations, it may cause large-scale explosions. For this reason, it may be a solution if two units do the same calculation at the same time. In this way, the outputs from the two controllers are compared and the system can be instantly switched into a safe state in case of inconsistency. In the scope of this thesis work, both redundancy and safety structure are obtained with the same configuration which consists of standard PLCs. As an

important point to note here, these structures are constructed only at logic solver level. In other words, any input or output units are not duplicated or established in a special structure. In Figure 4.8, the network of field equipments can be seen.

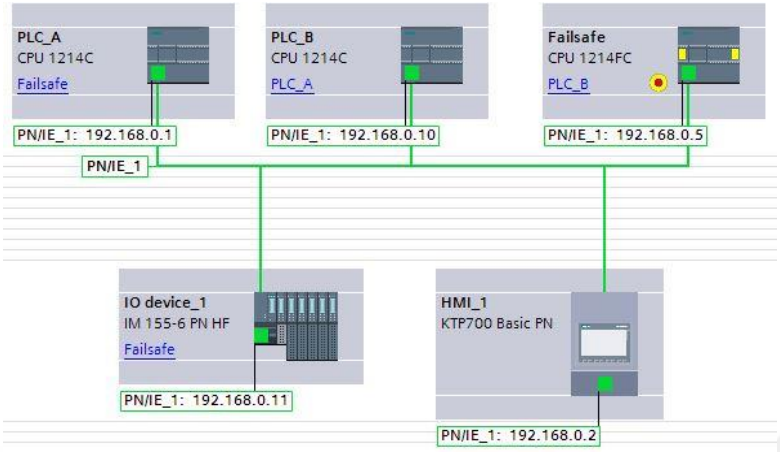


Figure 4.8 : Network of devices of the conveyor belt system.

In the given figure above, PN stands for PROFINET while IE represents Industrial Ethernet. PN/IE gives information about the communication protocol utilized and the physical interface of the protocol. For these devices to communicate each other, as the basis of ethernet communication, they have to be in the same sub-network. In other words, first three octets of their IP addresses have to be identical.

According to the purpose of this project, the Petri Net design of the conveyor belt system is carried out simultaneously in PLC A and PLC B. After deciding the system structure, PLC FS evaluates the outputs received from these two PLCs accordingly and transmits them to the system. If the redundant structure is decided, PLC FS transmits outputs from only one PLC directly to the system. However, the other PLC continues to do the same operations simultaneously. In case of a malfunction in the priority PLC, PLC FS immediately receives the outputs from the other PLC and transmits it to the system and possible disruptions are prevented. This established structure coincides exactly with the hot standby redundancy described in section 3.5. In Figure 4.9, the targeted redundant structure is given.

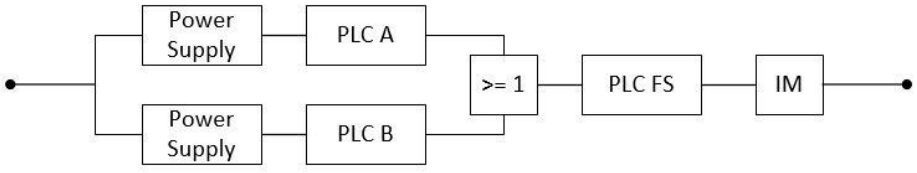


Figure 4.9 : Redundant structure to be established.

In the structure given above, PLC A is given priority over the other. This means that as long as this PLC is running, PLC FS takes the outputs produced by this PLC and transmit it to the system.

In case of the safe structure is selected for the system, PLC FS takes the outputs from both PLCs and makes comparisons between them. In order for the outputs to be transmitted to the system, they must be fully consistent with each other. Otherwise, the system does not operate. Additionally, in case of any power supply malfunction or connection error taking place in any PLC, since the agreement is spolied, the system is switched into a safe state and the operation is stopped. After errors are corrected, the system continues from where it left off with the approval of the operator concerned. In Figure 4.10, the targeted safe structure is given.

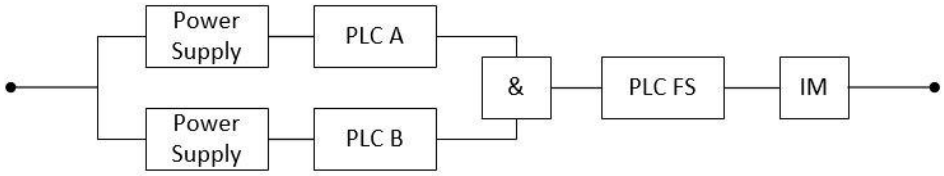


Figure 4.10 : Safe structure to be established.

Considering the information given in the section 3.4, it can be easily said that the structure in Figure 4.9 is a form of *1oo2* structure whereas the structure in Figure 4.10 represents *2oo2* structure.

4.4.1 Data transfer between PLCs

It is necessary for PLCs to exchange data with each other in order to establish the structures given in Figures 4.9 and 4.10. After configuring the devices in the system to be on the same sub-network, it is an important issue to exchange data between these devices. Communication between PLCs can be done with some function blocks that can be found under the "S7 Communication" library in TIA Portal. The most used of these are PUT & GET blocks.

In order to use these blocks effectively, a special data block which contains the data to be transferred should be created in each PLCs. Another important thing to consider is that the order of variables in each data block must be identical, since these blocks use data pointers. Data pointers play role of marking the beginning address value of data transfer. Starting from that labeled address, the data of predetermined byte-length is

read from the partner PLC or written to it. In Figure 4.11, an example of the "PUT" function block is given.

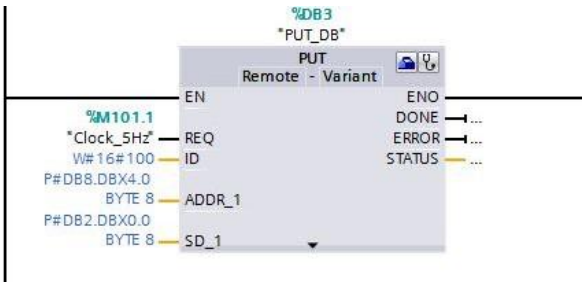


Figure 4.11 : PUT function block.

The "PUT" function block is utilized to write data to the partner PLC. As it is seen from the figure, this function block basically has 4 inputs. The "REQ" input, which means request, determines the frequency of data writing process. For this input, special clock bits of PLCs can be used as well as a bit associated with a push button. Thus, data writing can be performed continuously for a certain period, or it can be realized instantaneously when only one button is pressed. Next, the "ID" input takes the ID of the partner PLC as input. The use of "ADDR_1" and "SD_1" inputs are the same, they both take data pointers as input. Both are for pointing the starting address of data transfer. However, the pointer input of "ADDR_1" is related to the partner PLC whereas the one of "SD_1" is associated with the source PLC. In other words, the pointer input of "SD_1" indicates how many bytes of data to be sent starting from which address from which data block in the source PLC. The "ADDR_1" pointer input is to show how many bytes of data to be written starting from which address to which data block in the partner PLC. In order to complete the data transfer process, "GET" function block is used in the partner of the PLC, where the "PUT" function block is used. The aim of this block is to read data coming from the partner PLC and write them into related addresses. In Figure 4.12, an example regarding this function block can be seen.

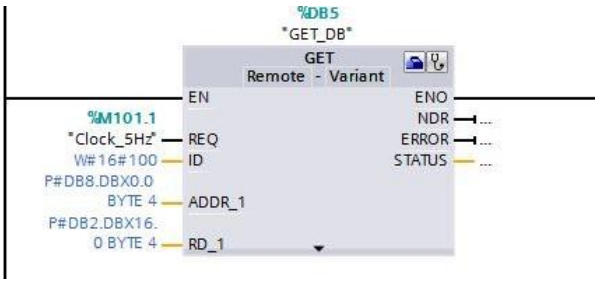


Figure 4.12 : GET function block.

As the single difference from the previous function block, the "GET" block includes "RD_1" as its input instead of "SD_1". The pointer of this input indicates how many bytes of read data to be written starting from which address to which data block in the source PLC.

In conclusion, PUT & GET blocks can be said useful for data transfer between PLCs. To establish a successful data transfer, these blocks have to be utilized as pairs between PLCs. Alternatively stating, if the source PLC uses the "PUT" function block to send data, the "GET" function block has to be employed in the partner PLC.

Another way to perform data transfer is to take advantage of the ability of PLCs operating as I/O controllers to work as I/O devices at the same time. This feature is activated under "Hardware Configuration > General > Operating mode > IO device", as shown in Figure 4.13.

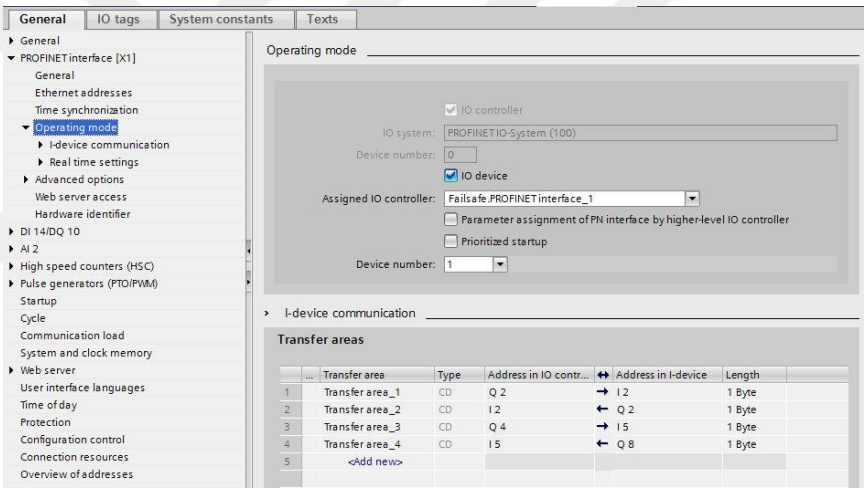


Figure 4.13 : Activation of I/O device feature of a PLC.

When activated, it is determined for the PLC which can also behave as an I/O device to be controlled by which PLC. After setting a controller, transfer areas between the I/O controller PLC and the I/O device PLC have to be identified. A transfer area consists of the starting address in I/O controller, the starting address in I/O device, and the data length. By this means, at the speed of PROFINET communication, data transfer between PLCs can be developed.

In the scope of this project, this special method is preferred for the communication between PLCs. This is due to an important drawback of using PUT & GET function blocks. Remembering the structures to be established, PLC FS has to decide whether PLC A or PLC B is running or not. To do so, a square wave test is applied between

PLCs. Square waves are sent non-stop to PLC FS by PLC A and PLC B in period of milliseconds. In case of any problem in a sender PLC, the square wave read by PLC FS will be stuck either at logic 1 or logic 0. The reason for such a problem can be: power supply failure, PLC failure, or failure in physical connection. It is very essential that such a problem is detected by PLC FS as soon as possible, even less than a second. This requires that the square waves to be sent and received are generated in precisely determined period and without being subject to any distortion. This is the case that using PUT & GET functions is inadequate. These function blocks are operated in "MAIN" block of the PLC. This block operates at between 0 and 150 ms, depending on the complexity of the program written in it. Using PUT & GET function blocks in "MAIN" block which has variable cycle time definitely leads to a synchronization problem. In this way, for example, it is possible for PLC FS to behave as if one of the PLCs was not working while actually it is working. Using the I/O device feature of PLCs for communication solves this problem and guarantees an unspoiled square wave test, as shown in Figure 4.14.

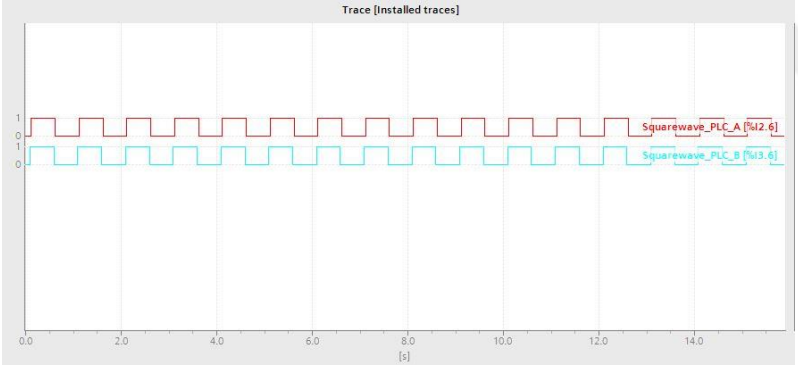


Figure 4.14 : Square wave test done in the project.

After deciding on the method to be used for communication, it was deemed appropriate to establish a structure similar to a ring network between PLCs. Thanks to this structure, all PLCs in the system take on both the I/O controller and I/O device role. Another advantage is that each PLC pair communicates directly between each other at the PROFINET speed. In Figure 4.15, the established communication network of the devices in the system is given.

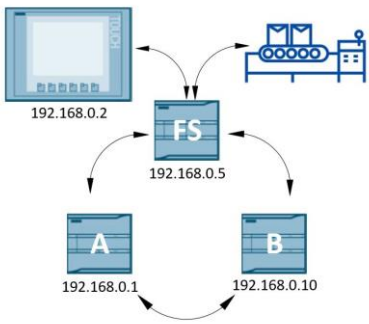


Figure 4.15 : Communication network of the devices.

According to the network given above, PLC A is the I/O device of PLC FS, whereas it is the I/O controller of PLC B. In addition, PLC B is the I/O controller of PLC FS. As it is also depicted in Figures 4.9 and 4.10, PLC FS is responsible for communicating with the conveyor belt. Taking advantage of the communication network built, also taking into account the requirements of the structures trying to be established in the system, the transfer areas required between PLCs are given in Table 4.3.

Table 4.3 : Transfer areas of data transfer between PLC pairs.

Pairs		Address in I/O Controller	Address in I/O Device	Length
I/O Con. : PLC FS I/O Dev. : PLC A	TA_1 TA_2 TA_3 TA_4	Q2 I2 Q4 I5	I2 Q2 I5 Q8	1 byte 1 byte 1 byte 1 byte
I/O Con. : PLC A I/O Dev. : PLC B	TA_1 TA_2 TA_3 TA_4	I3 Q3 Q4…7 I6…9	Q3 I3 I5…8 Q4…7	1 byte 1 byte 4 bytes 4 bytes
I/O Con. : PLC B I/O Dev. : PLC FS	TA_1 TA_2 TA_3 TA_4	I2 Q2 I4 Q8	Q3 I3 Q6 I6	1 byte 1 byte 1 byte 1 byte

As it is seen from the given table, 4 transfer areas for each PLC pairs are sufficient for executing the system as desired. For a better understanding, it is useful to give an example from the table. Considering the first pair and the first transfer area TA_1, an 8-bit data written to Q2 address in PLC FS is read from I2 address in PLC A in the same order.

To summarize the table, both commands given and feedback from the sensors on the conveyor are sent from PLC FS to PLC A and PLC B. In the opposite way, outputs

produced from executed programs are sent. Thus, according to the structure chosen for the system, PLC FS processes this incoming data and transfers it to the system accordingly. The 4-bytes data exchange between PLC A and PLC B is related to the Petri Net design that has been converted into the PLC program. In details, the information about which places are active in the designed Petri Net is shared instantly between these PLCs. In this way, after a problem in a PLC, when that PLC is working again, synchronization immediately takes place thanks to this continuous sharing.

The safe and redundant structure that is tried to be obtained with standard PLCs within the scope of this thesis work has been realized by means of the way of data transfer as explained above. If the redundant structure is selected for the system, and both PLCs are running, PLC A primarily transmits the outputs of the executed program to PLC FS. The same operations are done by PLC B at the same time; however, PLC FS does not use the outputs from PLC B. In case of a problem in PLC A, PLC FS starts to receive the outputs from PLC B momentarily. If the fault in PLC A is eliminated before the operation ends, that is, if it starts sending square waves to PLC FS again, PLC B is deactivated and the outputs sent by PLC A are used again. In Figure 4.16 the created function block in PLC FS in order for system to operate according to the redundant structure is given.

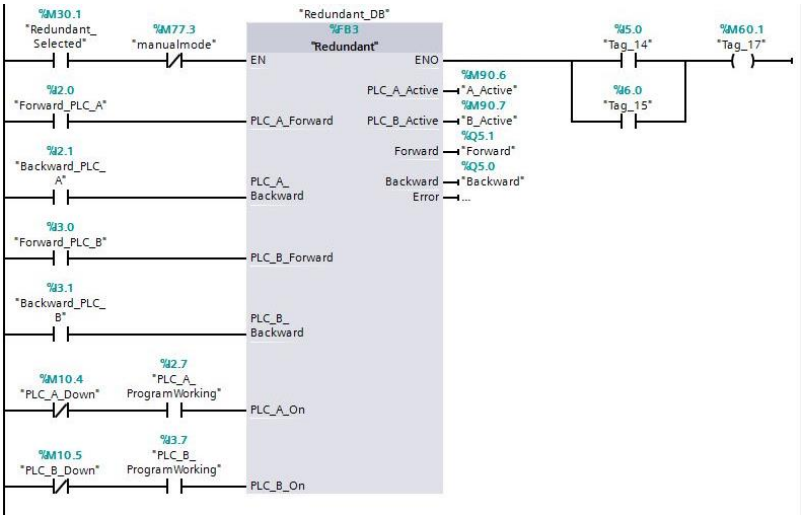


Figure 4.16 : The function block for the redundant structure.

When the given function block is investigated, it is seen that the DC motor related outputs coming from PLC A and PLC B are inputs of it. Additionally, by using the square waves sent from PLC A and PLC B, bits indicating whether these PLCs are working or not are the other inputs. According to the written program inside, the

outputs of this function block are: outputs indicating that PLC A and PLC B are working and outputs related to DC motor, to be transmitted directly to the system.

Alternatively, if the safe structure has been selected for the system, the condition of the same signals is sought from PLC A and PLC B as a result of the operation of this structure. In case of agreement between the outputs sent by these PLCs, the relevant output is directly transmitted to the system. In case of a power failure, CPU failure or a failure in physical connections in any PLC, the system is stopped immediately due to the inconsistency. The system is not allowed to start immediately from where it left off if all the conditions are appropriate after the relevant error is removed. The system restart is programmed to require the approval of an authorized operator. In Figure 4.17, the created function block in PLC FS in order for system to operate in conformity with the safe structure is given.

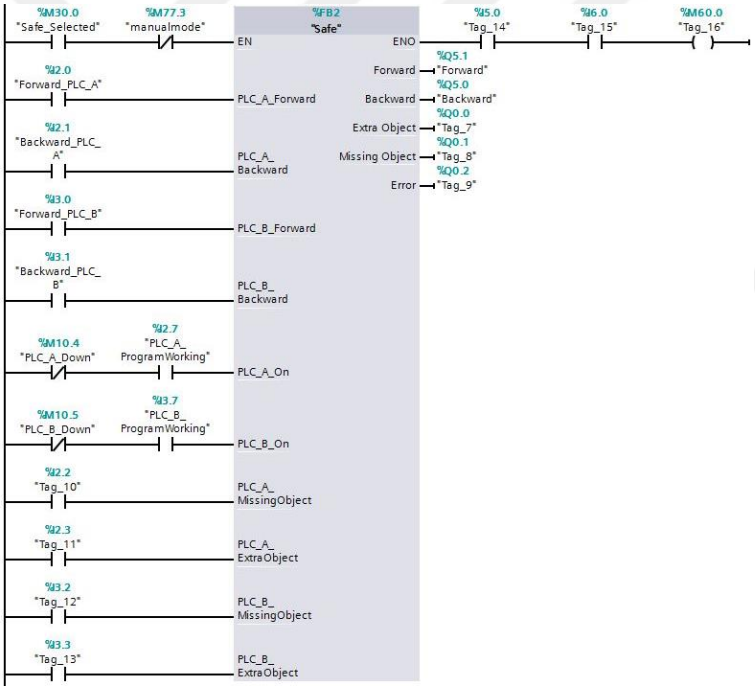


Figure 4.17 : The function block for the safe structure.

When the given function block is examined, it can be seen that there are the same inputs as the function block in Figure 4.16 and there are some additional inputs to them. The reason for the addition of these inputs is that both PLCs have to be in full compliance on account of the safe structure. For example, while the program is executed synchronously by two PLCs, if one PLC gives the missing object error while the other PLC does not give such an error, the system is stopped. The same applies to all outputs produced by two PLCs. For this reason, all of them are taken as the inputs

of this function block. On the output side of this function block, there are both outputs to be transmitted to the system related to DC motor and error outputs ,in case of compatibility with two PLCs, to be produced related to the conveyor belt.

4.5 Human Machine Interface

Human machine interface, shortly HMI, is used to control and monitor machines, processes and so on in industry. A very common HMI that is encountered in daily life is an ATM machine. In order to dispense or deposit certain amount of money, the screen and push buttons enable operation of the machine. In industry, HMI is an important member of having a great automated process. HMIs can either be form a screen, kind of like a computer screen, or touch screen mostly. An authorized operator can handle and oversee a process or machine through HMI. They may include information like temperature, pressure, sensor information etc. HMIs use special software so that engineering can be done properly. Different brands of panels use different software accordingly. The software allows engineer to design what the operator will actually see on the screen, what they can monitor on the screen, what buttons can bu pushed, or how the operator can manipulate the system. However, designing an HMI screen is not as simple as placing the necessary items on the screen with drag & drop. Each indicator or button in HMI screen has to be programmed to a specific input or output address of a PLC. The conclusion that can be drawn here is that HMI and PLC need to be compatible. In other words, they have to communicate each other. Like many existing protocols, PROFINET communication protocol can be used for such communication.

Within the scope of the thesis work, an HMI screen is designed for both control and monitoring of the conveyor belt system. At the same time, the working structure of the system, i.e. the redundant structure or safe structure, can be decided on the HMI screen. Under normal industrial conditions, the choice of the structure of the system is not left to the operator. The structure of the system is predetermined and the system is programmed accordingly. The reason for making such a design in this project is to show that the system can operate in both safe and redundant structure without changing the established communication structure given in Figure 4.15. The root screen of the designed HMI within the scope of this project is given in Figure 4.18.

Figure 4.18 : The root screen of the designed HMI.

When designing an HMI screen, it should be considered that the operator responsible for this does not know anything, so that it is detailed and understandable accordingly. When the root screen given in Figure 4.18 is examined, the operator has 3 things to choose: language, operation and system structure. After the system structure is selected, a new screen is opened according to the relevant structure and the conveyor is monitored on that screen. If the language or operation selection is forgotten by the operator and the system structure is selected, the relevant error message appears on the next screen and the operator is forced to return to the home screen. As a result, the HMI screen has been designed in proportion to possible errors that can be made by the operator. In this way, the effect of human error is aimed to be minimized. On the other hand, when a conveyor error occurs, the specially designed alarm screen appears immediately and directs the operator. Thereby, possible dangerous consequences are prevented.

.

5. RESULTS

In this thesis work, it is aimed that a conveyor belt system can work as redundant or safe when demanded within the framework of a structure established with standard PLCs. At the same time, as another important stage of the thesis work, the design of the mentioned conveyor belt system was carried out with Petri Nets. The main purpose of this study is to produce a solution for small businesses with cheap and compact standard PLCs that can perform the same tasks instead of the relatively expensive systems sold by many PLC manufacturers on the market.

Firstly, the subject of Petri Nets was introduced and the basic properties of Petri Nets were given. After that, necessary definitions were given in favor of a better understanding of the mathematical infrastructure. In order for coding the Petri Net design of the conveyor belt system with PLC, the previously proposed logical expressions method was preferred. The logical expressions from places and transitions obtained from the relevant Petri Net design were programmed with the LAD language defined in the IEC 61131-3 standard. As a next step, some definitions mentioned in some parts of the thesis have been explained in order to clearly understand the concepts of redundancy and safety. Subsequently, some architectural structures have been described in order to obtain the redundant and safe working structure intended for the conveyor belt system. Among these architectural structures generally known as *MooN*, the project has been realized based on *1oo2* and *2oo2* structures. In this project, the architectural structures utilized were realized only at the logic solver level. In other words, the way of connecting the input and output elements of the conveyor belt system has not been changed.

In this project, S7-1200 model PLC of Siemens brand was used as logic solver. The reason for choosing this PLC is that it is cheaper and more compact than other high-end products of the same brand. As an advantage of being produced recently, it can easily adapt to the developing technology and new application areas. On the other hand, the appealing and easy-to-use interface of the TIA Portal program where these PLCs are programmed is another advantage.

As a result of the studies carried out, fast and uninterrupted communication was provided between PLCs included in the structure and it was approved by square wave test. Then, benefiting from this communication, a special function block for processing the signals that travel between the PLCs was written and the targeted redundant structure was established firstly. As a result of the tests carried out on the conveyor belt, the system was observed to operate in accordance with the redundancy principle. Afterwards, a new function block was written and the system was enabled to operate according to the safety principle this time. As a result of the tests carried out, the targeted working principle for the system was successfully achieved. As another part of the project, an understandable and easy-to-use HMI screen was designed for both testing of targeted working principles of operation and control and monitoring of the conveyor belt system. When price comparison is made, it can be said, based on the prices of January 2020, that the system established within the scope of this project is 10 to 30 times cheaper than the high-end and bundle systems offered by the same brand. Therefore, it can be concluded that this is a cost saving solution for small businesses.

REFERENCES

[1] **Ji, K and Song, Z.** (2012). System and method for verification and validation of redundancy software in plc systems, uS Patent App. 13/415,897.

[2] **Alvarez, J., Marcos, J. and Fernandez, S.** (2005). Safe PLD-based programmable controllers, *International Conference on Field Programmable Logic and Applications*, 2005., IEEE, pp.559-562.

[3] **Kurtulan, S.** (2007). *Endüstriyel kumanda sistemleri,* Nobel Yayın Dağıtım.

[4] **Frey, G.** (2000). Automatic implementation of Petri net based control algorithms on PLC, *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, volume 4, IEEE, pp.2819-2823.

[5] **Lee, G.B. and Lee, J.S.** (2002). Constructing Petri net state equation for ladder diagram, *Journal of Intelligent Systems, 12*(2), 69-92.

[6] **Koch, I.** (2010). Petri Nets – A Mathematical Formalism to Analyze Chemical Reaction Networks, *Molecular Informatics, 29*, 838-843.

[7] **DiCesare, F., Harhalakis, G., Proth, J.M., Silva, M. and Vernadat, F.** (1993). *Practice of Petri nets in manufacturing,* Springer.

[8] **Fishwick, P.A.** (2007). *Handbook of dynamic system modeling,* CRC Press.

[9] **Cassandras, C.G. and Lafortune, S.** (2009). *Introduction to discrete event systems,* Springer Science & Business Media.

[10] **Moody, J.O. and Antsaklis, P.J.** (2012). *Supervisory control of discrete event systems using Petri nets*, volume 8, Springer Science & Business Media.

[11] **Hrúz, B. and Zhou, M.** (2007). *Modeling and control of discrete-event dynamic systems: With petri nets and other tools*, Springer Science & Business Media.

[12] **Lereverend, P.** (2008). Inside the standardization jungle: IEC 62061 and ISO 13849-1, complementary or competing?, *2008 5th Petroleum and Chemical Industry Conference Europe-Electrical and Instrumentation Applications*, IEEE, pp. 1-5.

[13] **Börcsök, J.** (2007). *Functional safety: basic principles of safety-related systems,* Hüthig.

[14] **Rausand, M.** (2014). *Reliability of safety-critical systems: theory and applications,* John Wiley & Sons.

[15] **Goble, W.M. and Cheddie, H.** (2005). *Safety Instrumented Systems verification: practical probabilistic calculations,* ISA-The Instrumentation, Systems and Automation Society.

[16] **ISO Central Secretary.** (2010). Guidance on the application of ISO 13849-1 and IEC 62061 in the design of safety-related control systems for machinery, **Standard PD IEC/TR 62061-1:2010**, British Standards Institution, London, UK.

[17] **Heuristic,** (n.d.). In *Wikipedia*. Retrieved March 20, 2020, from https://en.wikipedia.org/wiki/Redundancy_(engineering)..

[18] **Mok, Y.L., Goh, C.H. and Segaran, R.C.** (2013). Redundancy modeling for the X-sat microsatellite system, *2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS)*, IEEE, pp.1-6.

[19] **Url-1** <*https://support.industry.siemens.com/cs/pd/255094?pdti=td&dl=en&lc=en-TR*> date retrieved 26.03.2020.

[20] **Uzam, M. and Jones, H.** (1998). Discrete event control system design using automation Petri nets and their ladder diagram implementation, *The International Journal of Advanced Manufacturing Technology*, *14*(10), 716-728.

[21] **Uzam, M., Jones, H. and Yücel, I.** (2000). Using a Petri-net-based approach for the real-time supervisory control of an experimental manufacturing system, *The International Journal of Advanced Manufacturing Technology*, *16*(7), 498-515.

[22] **Başkocagil, C.** (2014). *Demiryolu Anklaşman Sistemlerinin Petri Ağları ile Tasarımı ve Gerçeklenmesi* (Ph.D thesis), Fen Bilimleri Enstitüsü, ISTANBUL.

CURRICULUM VITAE

Name Surname **:** Okan KONUK

Place and Date of Birth **:** Istanbul – 01.05.1993

E-Mail **:** konuko@itu.edu.tr

EDUCATION **:**

- **B.Sc.:** 2016, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control and Automation Engineering

PROFESSIONAL EXPERIENCE:

- Research Assistant, Istanbul Technical University, (2018 March – Ongoing)