GIAD STEEL FACTORY – MELTING HOUSE
CONTROL SYSTEM: ANALYSIS AND DEVELOPMENT

By
Mohammed Babiker Mohammed El-Obaid

Index NO. 064062

A thesis submitted to
Fulfillment of the
Requirement for Degree of
B.Sc (HON)

IN
Electrical and Electronic Engineering

Under Supervisor
Dr: Kamal Ramadan

Electrical and Electronic Engineering Department
Faculty of Engineering
University of Khartoum
July-2011
Declaration of originality:

I declare that this report entitled “Giad Steel Factory Melting House: Case Study and Development” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature: --------------------------------------------------

Name:---------------------------------------------------------

Date:----------------------------------------------------------
Dedication:

To my mother, with gratitude.
Acknowledgement:
I am indebted mostly to my project pair who had been a great help in accomplishing our project goal and to many individuals who continued to encourage us and eased the access to information for us. I want to mention particularly Osman - the melting house engineer- for his time and help, also Haydar who helped in the access to the factory which is the topic of the project.
Special cordial thanks to our project supervisor Dr. Kamal Ramadan for his time and dedication.
Finally to my family, from whom came the greatest support of all.
# Contents

Declaration of originality........................................................................................................... i  
Dedication .................................................................................................................................... ii  
acknowledgement .......................................................................................................................... iii  
table of contents............................................................................................................................. iv  
List of tables.................................................................................................................................. vi  
List of figures................................................................................................................................. vi  
List of abbreviations........................................................................................................................ viii  
abstract........................................................................................................................................... ix  
المستخلص....................................................................................................................................... ix  

CHAPTER 1: introduction........................................................................................................... 1  
1.1Introduction ............................................................................................................................... 2  
1.2problem statement....................................................................................................................... 2  
1.3project objectives......................................................................................................................... 2  
1.4motivation and justifications ...................................................................................................... 2  
1.5project activities........................................................................................................................... 2  
1.6thesis layout ............................................................................................................................... 3  

CHAPTER 2: industrial automation............................................................................................ 4  
2.1Introduction............................................................................................................................... 5  
2.2Definitions................................................................................................................................. 5  
2.3Historical briefing ..................................................................................................................... 5  
2.4Tools of industrial automation.................................................................................................. 6  
2.4.1Plant design ......................................................................................................................... 6  
2.4.2Process control ..................................................................................................................... 6  
2.4.3Industrial data communication ........................................................................................... 6  
2.4.4Other factors......................................................................................................................... 7  
2.5Industrial automation vs. industrial control............................................................................. 8  
2.6Automation advantages .......................................................................................................... 8  

CHAPTER 3: PLC and HMI......................................................................................................... 9  
3.1Introduction............................................................................................................................... 10  
3.2PLC systems............................................................................................................................. 10  
3.2.1definition ............................................................................................................................. 10  
3.2.2historical background .......................................................................................................... 10  
3.2.3specifications of the early PLCs .......................................................................................... 11  
3.2.4development and future ...................................................................................................... 11  
3.2.5architecture of PLCs .......................................................................................................... 12  
3.2.6communication protocols ................................................................................................... 16  
3.2.7programming languages ...................................................................................................... 16  
3.2.8KOYO DL450 PLC ............................................................................................................. 17  
3.2.9SLC 500 PLC ..................................................................................................................... 24  
3.3Human Machine Interface ...................................................................................................... 24  
3.3.1definition............................................................................................................................ 24  
3.3.2historical briefing ............................................................................................................... 24  
3.3.3HMI concept and standards ............................................................................................... 25  
3.3.4Wonder Ware InTouch ....................................................................................................... 26  

CHAPTER 4: case study, development and simulation.............................................................. 27
CHAPTER 4: Giad Steel Factory

4.1 Giad steel factory ................................................................. 28
   4.1.1 Introduction ............................................................... 28
   4.1.2 Brief description ....................................................... 28

4.2 General process description .................................................. 29
   4.2.1 PFD process flow diagram .......................................... 29
   4.2.2 List of components .................................................... 30

4.3 Electrical arc furnace .......................................................... 31
   4.3.1 Components .............................................................. 31
   4.3.2 Process description ..................................................... 33
   4.3.3 Logic analysis .......................................................... 36

4.4 Ladle furnace ......................................................................... 38
   4.4.1 Components .............................................................. 38
   4.4.2 Process description ..................................................... 39
   4.4.3 Logic analysis .......................................................... 40

4.5 Continuous casting machine ................................................... 41
   4.5.1 Components .............................................................. 41
   4.5.2 Process description ..................................................... 42
   4.5.3 Logic analysis .......................................................... 44

4.6 Developments ......................................................................... 45
   4.6.1 Introduction .............................................................. 45
   4.6.2 Automatic oxygen injection system ................................ 45
   4.6.3 Automatic steel composition system in LF .................... 47
   4.6.4 CCM tundish and mold level control system ............... 49
   4.6.5 Human machine interface design ............................... 50

4.7 Simulation ............................................................................... 52
   4.7.1 Introduction .............................................................. 52
   4.7.2 Tools ....................................................................... 52
   4.7.3 Simulation procedure ............................................... 53

CHAPTER 5: Conclusion and Future Work ........................................... 58

5.1 Conclusion ............................................................................. 59
5.2 Problems and Obstacles ......................................................... 59
5.3 Future Work and Recommendations ....................................... 60

References .................................................................................. 61

Appendix A: Visits Log ................................................................. 62
Appendix B: EAF HMI, Development and Simulation ....................... 63
   EAF Human Machine Interface and graphical notations ........ 64
   EAF Script ................................................................. 67

Appendix C: LF HMI, Development and Simulation ......................... 69
   LF Human Machine Interface and graphical notations ......... 69
   LF Script ................................................................. 70

Appendix D: CCM HMI, Development and Simulation ..................... 72
   CCM Human Machine Interface and graphical notations .... 72
   CCM Script ................................................................. 74

Appendix E: Electricity Status Simulation ....................................... 77
Appendix F; hydraulic status simulation..........................78

Appendix G; Ladder logic.............................................79
  EAF ladder logic..................................................79
  LF ladder logic...................................................82
  CCM ladder logic...............................................83

Appendix H; sensors....................................................84

Appendix I; simulation testing......................................86
List of tables:

Table 3.1                              DL450 memory map.           [19]
Table 3.2                              Some instructions used in DirectSoft 5 [21]

List of figures:

Figure 2.1                              Open System interconnection "OSI". [7]
Figure 2.2                              Industrial Control Component. [7]
Figure 3.1                              Siemens Simatic 200 PLC. [10]
Figure 3.2                              PLC block diagram [12]
Figure 3.3                              operations performed by a typical PLC [12]
Figure 3.4                              typical PLC’s CPU major components [13]
Figure 3.5                              CPU scan time concept. [13]
Figure 3.6                              categories of rack enclosure [15]
Figure 3.7                              KOYO DL450 CPU arrangement [18]
Figure 3.8                              STR, OUT and END instructions [18]
Figure 3.9                              logic instructions [20]
Figure 3.10                             PID start address initiation [22]
Figure 3.11                             PID setup [23]
Figure 4.1                              Giad steel factory [28]
Figure 4.2                              process flow diagram of the melting house [29]
Figure 4.3                              Electrical arc furnace [31]
Figure 4.4                              EAF control system flow chart [37]
Figure 4.5                              Ladle furnace [38]
Figure 4.6                              LF control system flow chart [40]
Figure 4.7  Continues casting machine  [41]
Figure 4.8  CCM control system flow chart  [44]
Figure 4.9  Panel configuration  [45]
Figure 4.10  PLC configuration to input PV and CV  [47]
Figure 4.11  Automatic Oxygen Injection System Configuration  [47]
Figure 4.12  Code for automatic steel composition adjustment  [48]
Figure 4.13  Hardware for automatic steel composition adjustment  [48]
Figure 4.14  High temp Level Sensors  [49]
Figure 4.15  (a) Ethernet module used in KOYO PLC.  [51]
Figure 4.16  (b) CCM connection to PC  [51]
Master control station  [51]
Figure 4.17  a snap-shot of the ladder logic used to configure the CCM  [53]
Figure 4.18  Communication Configurations for RSlogix  [53]
Figure 4.19  Topic Configurations for Wonder ware Intouch  [54]
Figure 4.20  EAF Interface  [54]
Figure 4.21  Script for tapping order.  [55]
Figure 4.22  LF movements.  [55]
Figure 4.23  Casting Process  [56]
Figure 4.24  (a) Strand straighter/ withdrawal movement  [57]
(b) Cutting Process

Figure 4.25 The CCM interface. [57]

List of abbreviations:

CCM  Continues Casting Machine
DCS  Distributed Control System
DDE  Dynamic Data Exchange.
EAF  Electrical Arc Furnace
HMI  Human Machine Interface
KVM  Keyboard, Visual display and Mouse
LF  Ladle Furnace
PAC  Programmable Automation Controller
PLC  Programmable Logic Controller
Abstract:
This project is focused in studying and improving of the control system cited in one of the major steel producing facilities in the country - Giad Steel Factory – through detailed analysis of the process and the hardware/software components and logic, the development of the monitoring process through deploying of a PC-based HMI, and a solution to the problems of the control system through modification both to the ladder logic and the hardware arrangement was proposed.

The proposed developments includes an automatic oxygen lancing system; which controls the flow and amount of oxygen fed to the Electrical Arc Furnace, The auto steel composition adjustment; which improves the quality of the product by precisely adjusting the additives ratios through PLC controlled valves, The CCM tundish and mold level control; for adjusting the casting and solidification process, and a PC-based human machine interface; which is the main contribution of our project to the enhancement of the factory overall control.

Several visits were conducted to the factory as indicated in the log-sheet to the assurance of accuracy of the information in this report and to closely study the factory environment and its working system. Also the actual ladder logic exists in the factory was obtained.

Finally the proposed system was simulated and a way of testing and calculating of reliability also proposed using the random variable generator method.

المستخلص:
هذا المشروع يركز على دراسة وتطوير نظام التحكم الموجود بوحدة من أهم منشآت إنتاج الحديد والصلب في البلاد – مصنع جياد للحديد والصلب- من خلال تحليل تفصيلي لعملية إنتاج الحديد ومتناقلاتها من عداد ونظام برمجيه، كما أن الدراسة تعطي حلول تطويرية لعملية مراقبة النظام من خلال إدخال وحدة تحكم تفاعليه مبنية على الكمبيوتر، وحلول لبعض المشاكل الأساسية المتعلقة بنظام التحكم من خلال تعديلات على مستوى الكود والاعتاد.

إن التعديلات المدخلة تشمل بالضرورة نظام آلی لحف الأكسجين يتحكم في معدل انسيب الأكسجين الي فرن الصهر الكهربائي و كذلك في جمعية الأكسجين المحقون. كما تشمل التعديلات نظام آلی لضادة المواد الكيميائيه لتحسين جودة الحديد المنتج من خلال التحكم في نسب المكونات الكيميائيه. بالإضافة الي نظام آلی لمراعية منسوب الحديد المصهور في وعاء الصهر والقواعد التابعه له لضبط عمليه التصدئ.

إن التطوير الأساسي يكمن في إضافة نظام مراقبه وتحكم من خلال اجهزة الكمبيوتر مما يسمح للعاملين و المهندسين من اصدار الأوامر عن بعد.

أخيرا، تمك منحكا استقلال النظام المقترح وتم ادخال طريقة لحساب كفاءة النظام من خلال استعمال دالة مولد القيم العشوائيه بمساعدة برنامج الماتلاب.
Chapter One: Introduction
1.1 Introduction:
Automation in steel industry had been a great contribution to the enhancement of economy and hence been the subject of many researches worldwide. This project was aimed to study and analyze an existing steel producing facility form the control system point of view and also to recommend the necessary modifications where needed.

1.2 Problem statement:
The safety, quality and economical issues related to the steel industry made automation essential, hence our project is directed to study the application of automation concepts in Giad steel factory. In spite of using automatic controllers < PLCs >, many operations in the factory are still conducted manually, causing serious performance degradations, these include:
- Loading of scrap and lime stone into the EAF.
- Oxygen injection into EAF.
- Loading of treatment additives into LF.
- Monitoring of tundish level.
These problem are somewhat serious, however, the lack of a sophisticated monitoring mechanism < PC-based HMI > is more serious, hence, most of the effort was dedicated to the design of an appropriate HMI.

1.3 Project objectives:
The project aims at achieving the following objectives;
1- Run a case study for the melting house existing control system.
2- Introduce developments if any to the control system.
3- Design an HMI for the melting house
4- Establish a simulated melting house with the development introduced.

1.4 Motivation and justification:
The power of the case study based projects comes from the fact that it actually relates the researcher and the research centre to the real industry problems, hence helping in elevating all the three parties involved in the process. Documentation and analysis conducted during a case study will help as a guiding reference in the future.
As for the developments proposed, the motivation is directly related to the enhancement of performance, quality and safety factory in the studied facility i.e. Giad steel factory.

1.5 Project activities:
The project was conducted on several levels;
1- Case study: a visits log appendix A describes the visits and rounds and interviews conducted with the factory personnel. Documentation, logic analysis, I/Os listing and flow charts were constructed.
   ❖ Note: the I/Os listing and ladder logic are in appendices B and C respectively, whilst flow charts is distributed among the case study chapter.
2- Developments and simulation: several steps were conducted for development and simulation; these include:
   - Problem statement.
- Logic <software > and hardware specifications and design.
- Assumptions, scripting and ladder logic redesigning.

3- Knowledge sharing: this includes;
• Presentation for the conference of “The Contribution Of Engineering Community To promote society”.
• www.automationproject.blogspot.com.
• Brief presentation on ”Thursday 23.june”.

1.6 Thesis layout:
As a graduation project, this report is intended for readers with a solid background on control systems theories and technologies; however the report follows a simplified approach to make it easy for the ordinary readers to follow the main ideas.

The first chapter of this report is an introduction gives a briefing on the problem statement, the project objectives, and the activities conducted to completion. The second chapter is a literature chapters intended to review main automation and control concepts and history. In chapter three a description of PLC and HMI is given in term of definition and programming methodology.

The fourth chapter, Case Study, Development and Simulation, gives the established work in the project with a full reporting of the case study, a statement of the development designed and the simulation procedure. Finally in chapter Five a conclusion and future work are stated.
Chapter Two:
Industrial Automation
2.1- Introduction:
This chapter provides a briefing on the historical background of the industrial automation and the body of knowledge required to establish an industrial automation infrastructure.  
This chapter is made as an introduction to the next literature chapter “PLCs and SCADA” which further discuss tow topics in detailed manner.

2.2- Definition:
According to Oxford Dictionary the word Automation in language means to convert (a process or facility) to be operated by largely automatic equipment. The adjective industrial refers the automation under consideration to the industry in its broad aspect. Technically industrial automation can be defined as the technology concerned with application of the electrical, mechanical and computer systems in the operation and control of manufacturing systems \([1]\). According to IEEE , automation is the use of information technologies and control systems to reduce the need for human labor in the production of goods and services\([2]\).

2.3 Historical Brief:
The word automation was coined in 1947 \([3]\) by Del Harder the Ford Motor Co. vice president of production. In that year Ford Co. applied the concept of automation to machine processes in automobile manufacture. In 1952 \([4]\) the writer and consultant John Diebold used the term as a title for his well know book Automation: The advent of the Automatic Factory.

Despite the late appearance of the word automation in the industry the concept of automation existed in the very beginning of the industrial revolution. In 1801 the Jacquard loom was invented by Joseph Marie Jacquard. the Jacquard loom was a mechanical loom controlled by punched holes. It was the first machine to use punched cards to control a sequence of operation\([5]\). Later in the 19\textsuperscript{th} century automation was applied using cams. Several machines was automated based on cams such as gun-stock-copying lathes built by Thomas Blanchard ,an American inventor, in 1820s-30s\([6]\). This machine was developed by another American inventor ,Christopher Miner Spencer, to become the screw machine in 1870\textsuperscript{s}\([7]\). All the automated machines in the 19\textsuperscript{th} century was an introduction to the computing concept in general and in industry. It’s all clear now that the automation concept even without referring to industry or manufacturing was produced by those aspects.

Other types of early automation appeared in the beginning of the 20\textsuperscript{th} century, the automotive industry in USA adopted relay logic based automation and applied semiautomatic mechanisms to the assembly lines used by the industry. In parallel with the initiative automatic systems developed by the automobile and aircraft industries the engineers in chemical industries developed control systems based on different sensors and a centralized control , in 1930\textsuperscript{s} \([8]\) ,especially in petroleum refineries. In the 1940\textsuperscript{s} the Numerical Control came to surface based on the work of john parsons who conceived of using punched cards containing position data to control the axes of a machine tool \([9]\)[10]. The previously mentioned Jacquard Loom can be considered the precursors of the NC machine tool as both operated using a form of punched paper tape as a program to control the actions of the respective machines \([9]\).

The 1940\textsuperscript{s} witnessed not only the birth of NC machine and the word automation, but also the invention of transistor in 1947\([11]\) by John Bardeen and Walter Brattain at AT&T\textapos;s Bell Labs. This invention was the beginning of the micro revolution. In the 1950\textsuperscript{s} computer industry was heading to replace vacuum tubes with transistors. The first industrial computer system was built in an oil refinery in Port Arthur USA in 1959\([12]\). Also in the 1950\textsuperscript{s} a robotic device was introduced by a the British inventor Cyril Walter Kenward. In 1961 the U.S patent for a device,
called "Programmed Article transfer" invented by the American inventor George C. Devol, was issued. The U.S patent describes the device as follows: "Apparatus having automatic control means.." \[9\][13]. Devol's work established the foundation for the modern industrial robot.

In the 1960's the automotive industry in U.S required some flexibility in the automation used. The relay control in the industry was costly and not flexible to changes in production. In response to a GM Co. request for an electronic replacement of the hard-wired system, an invention of a programmable controller was held by a company called Modicon. Later in 1974 the term PLC, Programmable logic controller, was coined by Allen Bradley company \[14\].

Another major technology of the industrial automation had its roots in the 1970's, the technology of DCS (distributed control systems). In 1975, two companies independently produced DCS's, Honeywell and Yokogawa. In the same year US-Based Bristol introduced a DCS.

A qualitative improvement in the automation field was a result of the invention of the microprocessor in the late 1970's. In the 1980's and 1990's the microprocessor along with the introduction of networking to industry lead to certain complexity and system integration to produce high performance industrial automation facility. Industrial automation in the late years experienced broad development forced by the competition factor in industry. Specification and requirements have changed from only cost efficient automation product to necessity of reliability and efficiency. The products for automation establishment such as PLC's and DCS's have become more resemble to each other as the broad meaning of integrated systems was adopted by vendors. HMI's (human machine interface) were introduced by different vendors and software companies.

Along the evolution of the controllers in automation and industry developed. According to the requirements of the industries institutes developed sensors with different features.

2.4- Tools of Industrial Automation:

to begin with, the concept of industrial automation drags many components that forms the needed body of knowledge, these components may be summarized as:

2.4.1- plant design: which refers to the automation technologies, work practices and business rules supporting the design and engineering of process and power plants \[15\].

This is done through a systematic approach; process description > process flow diagrams > instrumentation > electrical design and documentation.

2.4.2- process control: most basic process control systems consists of a control loop i.e.

- A measurement of the state or condition of a process
- A controller calculating an action based on this measured value against a pre-set or desired value (set point)
- An output signal resulting from the controller calculation which is used to manipulate the process action through some form of actuator
- The process itself reacting to this signal, and changing its state or condition.

Also several considerations are notified while designing a control system; these includes system’s stability and reliability and other performance considerations.

2.4.3- Industrial data communication: the favorite industrial data communication model is the OSI (open system interconnection) model; it provides a universal framework though it doesn’t specify the protocol used in each of its layers, the fig(2.1) represents this model.
The means by which various devices communicate is known as the protocol. Some of the famous industrial protocols are; profibus, modbus plus, Ethernet/IP and deviceNet. The physical layer is usually presented by a RS 232 cable or ETHERNET slot.

2.4.4- Other factors: These include:
- Monitoring and interfaces design and implementation.
- Safety and environment issues.
- Control panel design: this is a pure technical aspect regarding the wiring and installation of the electrical components of the controller and how it interacts with the various elements in the factory floor.
- Automation justification: from every point of view; i.e. economical, social ..Etc

Figure(2.2) illustrates the industrial control concept and components:
2.5- Industrial automation VS. Industrial control:
The main idea is that industrial automation is a higher level, thus it includes industrial control and other aspects (information technology, production planning ...Etc).
So industrial automation is the philosophy that defines the manufacturing and production process path, equipments and tools and the technical specifications involved, thus controller specifications and process modeling and case studies are conducted under its umbrella.
The simple hierarchy proposed is as follows:
- Sensors and actuators “factory floor”: in this level the actual process takes place, the controller acquires data through the sensors and addresses the process through actuators.
- Controller level: in this level inhabits the process station “operation station” which performs a predefined logic to manipulates the process, typical process stations is composed from a server and a field controller.
- Engineering station: whereby the logic design and modifications takes place, and actual equipment design and implementations modifications are made.
- Network: for data collection and transfer among various levels.
- Operation station: whereby operators monitor the process and the controller/ sensors, actuators performance.
- Production planning unit: responsible of defining the production strategies and hence the set points and other factors regarding the production control to the engineering station, and also monitor the performance of the whole unit.
- Enterprise control: this unit is responsible from the monitoring of the strategic goals of the facility and issues the required modifications to the lower levels.

2.6- Automation Advantages:
This section is to provide a set of incentives that made automation a very charming approach in industry worldwide;

1- Economical factors:
Regarding this, automation proved to be a very powerful tool in increasing productivity and enhance the products quality.
A very important issue is the unemployment relation to the automation, this was discussed in many academic papers, a result is that the automation may cause a seasonal unemployment but it also help creating a new job market which help solving the structural unemployment issue.

2- Safety factors:
Removal of human beings from hazardous environment helps enhance the work conditions and decrease work accidents and hence the money wasted for this matter.

3- Environmental factors:
By implementing automation standards, flew gas treatment and additives conservation concepts are guaranteed, hence ensuring a green industry.
Chapter Three:
PLC and HMI
3.1- Introduction:

The control system of Giad Steel Factory- Melting House ,which is the subject of this case study, is built on PLC's. This chapter states all the necessary information about PLC's in general and description of two PLC brands Koyo and AB. Koyo PLC's are described in term of CPU specifications and the programming methodology because they are the main controllers. In contrast AB PLC's are described only in term of programming methodology because they are used in simulation.

HMI is also defined in this chapter and the design methodology is given by the description of the design software , Wonderware Intouch 9.5.

3.2- PLC Systems:

3.2.1- Definition:

PLC , stands for programmable-logic-controller , which can be described as a special digital computer mostly dedicated for automation purposes of an electromechanical process \([15]\). A PLC offers a large number of input/output arrangements, can operates at a large range of temperature, resistance and immunity to the industrial environment of vibration and impact. The National Electrical Manufacturers Association - US (NEMA) has defined PLC as follows:

" A PLC is a digitally operated electronic system, designed for use in an industrial environment, which uses a programmable memory for the internal storage of user-oriented instructions for implementing specific functions such as logic sequencing, timing counting and arithmetic to control ,through digital or analog inputs and outputs, various types of machines or processes "\([16]\).

Fig (3.1) shows the Siemens Simatic -200 PLC.

3.2.2- Historical Background:

Before PLC control, sequencing, and safety interlock logic for manufacturing was accomplished using hundreds or thousands of relays, cam timers, and drum sequencer and dedicated closed –loop controllers. the main disadvantage of this type of automation is that it's not flexible and the change over for in the process is both time and finance consuming \([15][17]\).

The invention of the PLC was a response to the needs of the American automotive manufacturing industry. The PLC was initially adopted where software revision replaced the re-wiring of hard-wired control panels when production models changed. In 1968 \([15][17]\), the automatic transmission division of General Motors issued a request for proposal for an electronic replacement of hard-wired systems. The electronic replacement specification were stated as follow \([17]\):

- The new control system had to be price competitive with the use of relay systems.
- The system had to be capable of sustaining an industrial environment.
- The input and output interfaces had to be easily replaceable.
- The controller had to be designed in modular form, so that subassemblies could be removed easily for replacement or repair.
- The control system needed the capability to pass data collection to a central system.
• The system had to be reusable.
• The method used to program the controller had to be simple, so that it could be easily understood by plant personnel.

Four companies took on the challenge\[^19\]:
1. Information Instruments, Inc. (fully owned by Allen-Bradley a year later).
2. Digital Equipment Corp. (DEC).

### 3.2.3- Specification of the early PLCs:

The first PLCs offered relay functionality, thus replacing the original hardwired relay logic, which used electrically operated devices to mechanically switch electrical circuits. They met the requirements of modularity, expandability, programmability, and ease of use in an industrial environment. These controllers were easily installed, used less space, and were reusable. The controller programming, although a little tedious, had a recognizable plant standard: the ladder diagram format. In a short period, programmable controller use started to spread to other industries. By 1971\[^17\], PLCs were being used to provide relay replacement as the first steps toward control automation in other industries, such as food and beverage, metals, manufacturing, and pulp and paper.

The 1st programmable controllers formed the conceptual design of the PLC even though they were more or less just relay replacers. Their primary function was to perform the sequential operations that were implemented by relays. Those controllers were, unlike relays and cam timers, easily installed and used less space energy. They were easier in trouble shooting and reusable if the project was scrapped.

As a new invention of course the Programmable Controller faced problems. The problems were mainly about the response time of the Programmable controller (which had to be less than the relay response time), the price and reliability. Those problems were solved by introducing a PLC consists of 3 main units which are a processor board, memory and a logic solver\[^19\].

### 3.2.4- PLC development and future:

The four companies mentioned earlier (Modicon, AB, GE and OMRON) continued producing different models of PLCs. The enhancement in the models was basically concentrated on the reliability and speed of the PLC. In 1979, both Modicon company and AB introduced networking capabilities for their PLCs. In 1980's, two companies participated in the race. That was Siemens company offering the famously known Simatic PLC, in 1983, which later became a leading brand in PLC industry, and the Japanese company Mitsubishi with the "A series PLCs"\[^18\]. The general electric company became GE Fanuc Automation Company in 1986 and continued the production PLCs\[^18\].

In the 1990's, Modicon offered the well known Profibus and Ethernet capabilities for their PLC. AB followed that by the innovation of the "device network" which offered open network capabilities for AB PLCs\[^18\].

The last few years a PLC alike systems such as DCS (distributed control system) and PAC (Programmable automation controller) were adopted by the manufacturers of the PLC. The resemble between DCS, which was a very different technology started in the 1970's, and the PLC became huge due to the microprocessor technology. The vendors of DCS are mostly the vendors of DCS and the combination of those two automation means is now very common.

Indeed PLC innovation was a highlight in industry, and today wherever there's industry there's PLC. The future of PLC relies on the integration of PLCs with other control and factory management equipments. In fact today, PLCs are being incorporated through networks into...
computer integrated manufacturing (CIM) systems combining their power and resources with numerical controls, robots, CAD/CAM systems, personal computers, management information systems, and hierarchical computer-based systems [17].

3.2.5- Architecture of PLC:
A typical block diagram of a PLC is shown in fig(3.2).
The basic sections of the PLC are:
1- The central processing unit.
2- The output/input interface system.

Those two sections show that a PLC has the basic minicomputer structure. The main difference is the number of input/output that can be added to the PLC, which is much more than a general-purpose computer.

The CPU (Central Processing Unit) has three major components: the processor, the memory system, and the power supply. These components along with the I/O interface system perform the programmable controller operation which is relatively simple. The I/O interface system is physically connected to the field devices. The field devices usually sensors, actuators and any other component of control. Those field devices can be discrete or analog such as limit switches, pressure transducers, push buttons, motor starters, solenoids, etc.

As a typical processor system the PLC processor during operation performs three basic operations: read, execute, write, shown in fig(3.3). This process of sequentially reading the inputs, executing the program in memory, and updating the outputs is known as scanning.

The input/output system forms the interface by which field devices are connected to the controller. The main purpose of the interface is to condition the various signals received from or sent to external field devices. Incoming signals from sensors (e.g., push buttons, limit switches, analog sensors, selector switches, and thumbwheel switches) are wired to terminals on the input interfaces. Devices that will be controlled, like motor starters, solenoid valves, pilot lights, and position valves, are connected to the terminals of the output interfaces. The system power supply provides all the voltages required for the proper operation of the various central processing unit sections [17].

Although not generally considered a part of the controller, the programming device, usually a personal computer or a manufacturer’s miniprogrammer unit, is required to enter the control program into memory. The programming device must be connected to the controller when entering or monitoring the control program.

3.2.5.1- CPU (Central Processing Unit):
As seen fig(3.4) the CPU of a PLC consists of three major components: a processor, memory and a power supply. CPU architecture may differ from one manufacturer to another, but in general, most CPUs follow this typical three-component organization. Although this diagram shows the power supply inside the CPU block enclosure, the power supply may be a separate unit that is mounted next to the block enclosure containing the processor and memory. In general, the processor executes the control program stored in the memory system in the form of
ladder diagrams, while the system power supply provides all of the necessary voltage levels to ensure proper operation of the processor and memory components.

![Typical PLC's CPU major components](image)

**Chapter 3: PLC and HMI**

*a- Processor:*

They perform the mathematical operations, data handling, and diagnostic routines that were not possible with relays or their predecessor, the hardwired logic processor. In the PLC the principle function of the processor is to command and govern the activities in the system. This function is performed by the microprocessor using interpreting and executing of collection of system programs known as the executive \(^{[17]}\). The executive is a group of supervisory programs that are permanently stored in the processor. By executing the executive, the processor can perform all of its control, processing, communication, and other housekeeping functions. The executive also performs the communication between the PLC system and the user through a programming device. It also supports other peripheral communication.

The microprocessor in a PLC is responsible of reading all the I/O, execute an originally specified program then update the I/O. This processor is called Scanning Process. The repetition of this process makes the operation of the process sequential from the top to the bottom. Fig(3.5) shows the concept of processor scanning.

The program execution is called the program scan by the PLC manufacturers. The read and updating of the I/O is called I/O update scan. The internal process signal which indicates that the program scan is done is called end-of-scan signal (EOS). The time that takes the processor to perform a single scan is called scan time. The time scan time generally depends on 2 factors\(^{[16][17]}\): the amount of the memory taken by the control program and the types of instructions used in the control program. Time scans for different PLC may vary from a few tenths of a millisecond to 50 milliseconds\(^{[15][16][17]}\).

The changes in I/O during the program scan in the model of processor scanning earlier. An immediate input/output feature is common with some types of PLC. That ensure the changes in I/O during the program scan will be noticed and processed.

The processor is responsible of detecting failures that may occur during system operation and produce an alert to warn the user. To perform this task, the processor make diagnostics
during its operation. Typical diagnostics include memory OK, processor OK, battery OK and power supply OK. Different types of PLCs may contain extra diagnostics.

b- System Power Supply:

The power supply is responsible of supplying well regulated power and the protection for the other PLC components. Usually power supplies require an AC input voltage, however, some power supplies accept DC also. Most controllers require 120 V/AC or 220 V/AC. Only few controllers require 24 v/DC. A line voltage tolerance of ±10% is necessary specification needed in the power supply. In some plants a problem of unstable may appear due to heavy equipment start up or any other industrial circumstances. Usually a very good supply will not handle more than ±10% tolerance go to shut down, this shut down may lead to production loss. For this reason a constant voltage transformer is usually interfaced with plant network to give good regulation for the Power Supply. The PLC may b placed in an environment where an electromagnetic interference may occur. This interference may result in intermittent misoperation of the controller. An isolation transformer [17] can be installed between the controller and the AC power circuit to reduce the electromagnetic interference.

The power supply is supposed to provide the DC power required by the logic circuits of the CPU and the input/output circuits. A power supply has a maximum amount of current it can provide at a given voltage depending on the type of the power supply. A problem of undercurrent may occur if the OLC has an excessive number of I/O module. An auxiliary power supply can be added or the power supply can be replaced with another power supply that can provide higher power for the I/O modules.

c- Memory System:

The memory system is the area in the PLC’s CPU where all of the sequences of instructions, or programs, are stored and executed by the processor to provide the desired control of field devices. The memory sections that contain the control programs can be changed, or reprogrammed, to adapt to manufacturing line procedure changes or new system start-up requirements [17]. The memory system consists two types of memory: the executive memory and the application memory.

The executive memory is a collection of permanently stored programs that are considered part of the PLC itself. These supervisory programs direct all system activities, such as execution of the control program and communication with peripheral devices. The executive section is the part of the memory where the system’s available instruction software is stored (i.e., relay instructions, block transfer functions, math instructions, etc.). This area of memory is not accessible to the user.

The application memory provides a storage area for the user-programmed instructions that form the application program. The application memory area is composed of several areas, each having a specific function and usage.

The storage and retrieval requirements for the executive and application memory sections are not the same; therefore, they are not always stored in the same type of memory. For example, the executive requires a memory that permanently stores its contents and cannot be erased or altered either by loss of electrical power or by the user. This type of memory is often unsuitable for the application program. Usually ROM is used for the executive memory while RAM is used for the application memory.

d- Programming Devices:

This unit is not built in the PLC, and hence not a part of the CPU but it's considered as a very important external unit. The programs that the PLC meant to perform to automate a given process is provided through this terminal. most of PLCs are programmed using very similar
instructions. The only main difference may be in the mechanics associated with entering the program into the PLC. This mechanics may vary from a vendor to another.

A programming device may be a mini programmer or a personal Computer. The mini-programmer are mainly used for the small PLCs while PCs are used for programming all kinds of PLCs and the PLC vendors provide the necessary software for programming and the ability of monitoring.

3.2.5.2- The I/O interface:

Input/output (I/O) systems put the “control” in programmable controllers. These systems allow PLCs to work with field devices to perform programmed applications.

An I/O module is a plug-in–type assembly containing circuitry that communicates between a PLC and field devices. All I/O modules must be placed or inserted into a rack enclosure, usually referred to as a rack, within the PLC. The rack holds and organizes the programmable controller’s I/O modules, with a module’s rack location defining the I/O address of its connected device. The I/O address is a unique number that identifies the input/ output device during control program setup and execution. Several PLC manufacturers allow the user to select or set the addresses (to be mapped to the I/O table) for each module by setting internal switches.

A rack, in general, recognizes the type of module connected to it (input or output) and the class of interface (discrete, analog, numerical, etc.). This module recognition is decoded on the back plane (i.e., the printed circuit board containing the data bus, power bus, and mating connectors) of the rack. The controller’s rack configuration is an important detail to keep in mind throughout system configuration. Remember that each of the connected I/O devices is referenced in the control program; therefore, a misunderstanding of the I/O location or addresses will create confusion during and after the programming stages.

Generally speaking, there are three categories of rack enclosures:

a- master racks, fig(3.6,a).
b- local racks, fig(3.6,b).
c- remote racks, fig(3.6,c).

\[ \text{The term master rack refers to the rack enclosure containing the CPU or processor module. This rack may or may not have slots available for the insertion of I/O modules. The larger the programmable controller system, in terms of I/O, the less likely the master rack will have I/O housing capability. A local rack is an enclosure, which is placed in the same area as the master} \]
rack, that contains I/O modules. If a master rack contains I/O modules, the master rack can also be considered a local rack. In general, a local rack (if not a master) contains a local I/O processor that sends data to and from the CPU. This bidirectional information consists of diagnostic data, communication error checks, input status, and output updates. The I/O image table maps the local rack’s I/O addresses. As the name implies, remote racks are enclosures, containing I/O modules, located far away from the CPU. Remote racks contain an I/O processor (referred to as a remote I/O processor) that communicates input and output information and diagnostic status just like a local rack. The I/O addresses in this rack are also mapped to the I/O table. The rack concept emphasizes the physical location of the enclosure and the type of processor (local, remote, or main CPU) that will be used in each particular rack. Every one of the I/O modules in a rack, whether discrete, analog, or special, has an address by which it is referenced. Therefore, each terminal point connected to a module has a particular address. This connection point, which ties the real field devices to their I/O modules, identifies each I/O device by the module’s address and the terminal point where it is connected. This is the address that identifies the programmed input or output device in the control program.

There are two types of I/O interfaces: Discrete I/O systems and Analog I/O systems. The discrete input/output (I/O) system provides the physical connection between the central processing unit and field devices that transmit and accept digital signals. Digital signals are non-continuous signals that have only two states—ON and OFF. Through various interface circuits and field devices (limit switches, transducers, etc.), the controller senses and measures physical quantities (e.g., proximity, position, motion, level, temperature, pressure, current, and voltage) associated with a machine or process. Based on the status of the devices sensed or the process values measured, the CPU issues commands that control the field devices. In short, input/output interfaces are the sensory and motor skills that exercise control over a machine or process. Analog input modules are used in applications where the field device’s signal is continuous. Unlike discrete signals, which possess only two states (ON and OFF), analog signals have an infinite number of states. Temperature, for example, is an analog signal because it continuously changes by infinitesimal amounts. Consequently, a change from 70°F to 71°F is not just one change of 1°F, but rather an infinite number of smaller changes of a fraction of a degree.

3.2.6- Communication protocols:
PLCs have built-in communications ports, usually 9-pin RS-232, but optionally EIA-485 or Ethernet. Modbus, BACnet or DF1 is usually included as one of the communications protocols. Other options include various fieldbuses such as DeviceNet or Profinet. Other communications protocols that may be used are listed in the List of automation protocols.

Most modern PLCs can communicate over a network to some other system, such as a computer running a SCADA (Supervisory Control And Data Acquisition) system or web browser.

3.2.7- PLC Programming Languages:
The programming languages used in programmable controllers have been evolving since the inception of the PLC in the late 1960s. As PLCs have developed and expanded, programming languages have developed with them. Programming languages allow the user to enter a control program into a PLC using an established syntax. Today’s advanced languages have new, more versatile instructions, which initiate control program actions. These new instructions provide more computing power for single operations performed by the instruction itself. For instance, PLCs can now transfer blocks of data from one memory location to another while, at the same time, performing a logic or arithmetic operation on another block. As a result of these new, expanded instructions, control programs can now handle data more easily. In addition to new programming instructions, the development of powerful I/O modules has also changed existing
instructions. These changes include the ability to send data to and obtain data from modules by addressing the modules’ locations. For example, PLCs can now read and write data to and from analog modules. All of these advances, in conjunction with projected industry needs, have created a demand for more powerful instructions that allow easier, more compact, function-oriented PLC programs.

The three types of programming languages used in PLCs are: Ladder, Boolean and Grafcet. The ladder and Boolean languages essentially implement operations in the same way, but they differ in the way their instructions are represented and how they are entered into the PLC. The Grafcet language implements control instructions in a different manner, based on steps and actions in a graphic oriented program. The subsections below gives an over view of the Relay ladder Logic (RLL) and Grafcet.

3.2.7.1- Relay Ladder Logic (RLL) :

The programmable controller was developed for ease of programming using existing relay ladder symbols and expressions to represent the program logic needed to control the machine or process. The resulting programming language, which used these original basic relay ladder symbols, was given the name ladder language. Figure 9-1 illustrates a relay ladder logic circuit and the PLC ladder language representation of the same circuit.

The evolution of the original ladder language has turned ladder programming into a more powerful instruction set. New functions have been added to the basic relay, timing, and counting operations. The term function is used to describe instructions that, as the name implies, perform a function on data that is, handle and transfer data within the programmable controller. These instructions are still based on the simple principles of basic relay logic, although they allow complex operations to be implemented and performed.

New additions to the basic ladder logic also include function blocks, which use a set of instructions to operate on a block of data. The use of function blocks increases the power of the basic ladder language, forming what is known as enhanced ladder language shows enhanced functions driven by basic relay ladder instructions.

3.2.7.2- Grafcet

Grafac (Graphe Fonctionnel de Commande Étape Transition) is a symbolic, graphic language, which originated in France, that represents the control program as steps or stages in the machine or process. In fact, the English translation of Grafcet means “step transition function charts. Grafcet is the foundation for the IEC 1131 standard’s sequential function charts (SFCs), which allow several PLC languages to be used in one control program.

Few programmable controllers may be directly programmed using Grafac. However, several Grafac software manufacturers provide off-line Grafac programming using a personal computer. Once programmed in the PC, the Grafac instructions can be transferred to a PLC via a translator or driver that translates the Grafac program into a ladder diagram or Boolean language program. Using this method, a Grafac software manufacturer can provide different PLCs that use the same language.

3.2.8- KOYO DL450 PLC:

Koyo Electronics is major manufacturers of PLC's. In fact KOYO introduced its first PLC in 1982 and called "SR21 PLC". From then more PLC were introduced until 1994 when KOYO designed the well known series of PLC's called Direct Logic "DL". The DL series have a range from small PLC's "DL05" to powerful PLC's "DL405". Giad Steel Factory - Melting House control system use five PLC's of DL405 specifically with DL450, shown in fig(3.7), which is latest PLC of DL405 series. In this section a description of the KOYO DL450 PLC will be given. The description is given in term of the system components (CPU , Bases , I/O configuration and I/O Modules) and programming methodology.
3.2.8.1- DL450 Hardware:

DL450 is a modular CPUs which are installed in either 4, 6, or 8 slot bases. It has a maximum of 30.8K of program memory comprised of 15.5K of ladder memory and 15.3K of V-memory (data registers). It supports a maximum of 2048 points of local and local expansion I/O, and 1536 points of remote I/O. It includes an additional internal RISC--based microprocessor power. The DL450 has 210 instructions. It gives a scan time of 4-5 ms The DL450 has a total of four communications ports: two RS232C interface, RS232C/RS422 interface, RS-422 interface, shown in fig(3.7). Table (5.1) states the memory mapping of DL450 CPU.

3.2.8.2- DL450 Programming software "DirectSOFT 5":

DirectSOFT 5 is a windows based program used to construct and download the ladder logic to the PLC. The DirectSOFT 5 offers online monitoring for PLC beside a set of instructions. The instructions used in the Project are stated below in this section.

Fig(3.8), shows three main instruction necessarily used in every ladder program. Those instructions are STR, OUT and END. The STR (store) instructions is drawn as a relay in the ladder logic. It's used to store the value of (X0) as it's shown in fig(3.8) to the contact beneath it to use it for a logical purpose such as AND, OR. OUT is to set the result of the logic arrangement in the rung to the output (Y0). END statement is used to indicate the end of the program. Notice that each of those instructions can be drawn as it shown in fig(3.8) or can be written as follows:

STR X0
OUT Y0
END

To use the STR instruction for logic, the contacts can be connected in parallel to for OR operation or in series for AND operation. STRN is used as NOT operation. as it's shown in fig(3.9). Also in fig(3.10) the instructions needed for Timers, Counters and Memory manipulation are stated.
<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Discrete Memory Reference (octal)</th>
<th>Word Memory Reference (octal)</th>
<th>Qty. Decimal</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Points</td>
<td>X0 – X1777</td>
<td>V40400 – V40477</td>
<td>1024</td>
<td>X0</td>
</tr>
<tr>
<td>Output Points</td>
<td>Y0 – Y1777</td>
<td>V40500 – V40577</td>
<td>1024</td>
<td>Y0</td>
</tr>
<tr>
<td>Control Relays</td>
<td>C0 – C3777</td>
<td>V40600 – V40777</td>
<td>2048</td>
<td>C0</td>
</tr>
<tr>
<td>Special Relays</td>
<td>SP0 – SP777</td>
<td>V41200 – V41237</td>
<td>512</td>
<td>SP0</td>
</tr>
<tr>
<td>Timers</td>
<td>T0 – T277</td>
<td>V41100 – V41117</td>
<td>256</td>
<td>T0</td>
</tr>
<tr>
<td>Timer Current Values</td>
<td>None</td>
<td>V00000 – V00377</td>
<td>256</td>
<td>V0</td>
</tr>
<tr>
<td>Timer Status Bits</td>
<td>T0 – T377</td>
<td>V41100 – V41117</td>
<td>256</td>
<td>T0</td>
</tr>
<tr>
<td>Counters</td>
<td>CT0 – CT377</td>
<td>V41140 – V41157</td>
<td>256</td>
<td>CT0</td>
</tr>
<tr>
<td>Counter Current Values</td>
<td>None</td>
<td>V01000 – V01377</td>
<td>256</td>
<td>V0</td>
</tr>
<tr>
<td>Counter Status Bits</td>
<td>CT0 – CT377</td>
<td>V41140 – V41157</td>
<td>256</td>
<td>CT0</td>
</tr>
<tr>
<td>User Data Words</td>
<td>None</td>
<td>V14000 – V17377 V10000 – V38777</td>
<td>3072</td>
<td>None specific, used with many instructions</td>
</tr>
<tr>
<td>Stages</td>
<td>S0 – S1777</td>
<td>V41000 – V41077</td>
<td>1024</td>
<td>S0</td>
</tr>
<tr>
<td>Remote In / Out</td>
<td>GX0 – GX3777</td>
<td>V40000 – V40177</td>
<td>2048</td>
<td>GX0</td>
</tr>
<tr>
<td></td>
<td>GY0 – GY3777</td>
<td>V40200 – V40377</td>
<td>2048</td>
<td>GY0</td>
</tr>
<tr>
<td>System parameters</td>
<td>None</td>
<td>V7000 – V7777 V7400 – V7777 V37000 – V37777</td>
<td>832</td>
<td>None specific, used with many instructions</td>
</tr>
</tbody>
</table>

Table 3.1: DL450 Memory Map
Fig(3.9): Logic instructions (a) Not (b) AND (c) OR (d) Combined Logic
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR</td>
<td>a 0.1 second single input timer that times to a maximum of 999.9 seconds. will be enabled if the input logic is true (on) and will be reset to 0 if the input logic is false (off).</td>
<td><img src="image" alt="TMR Diagram" /></td>
</tr>
<tr>
<td>TMRA</td>
<td>a 0.1 second two input timer that times to a maximum of 9999999.9. have two inputs, an enable and a reset.</td>
<td><img src="image" alt="TMRA Diagram" /></td>
</tr>
<tr>
<td>CNT</td>
<td>a two input counter that increments when the count input logic transitions from off to on. When the counter reset input is on the counter resets to 0.</td>
<td><img src="image" alt="CNT Diagram" /></td>
</tr>
<tr>
<td>LD</td>
<td>a 16 bit instruction that loads the value (Aaaa) (either a V-memory location or a 4 digit constant) into the lower 16 accumulator bits. The upper 16 accumulator bits are set to 0.</td>
<td><img src="image" alt="LD Diagram" /></td>
</tr>
<tr>
<td>LDD</td>
<td>a 32 bit instruction that loads the value (Aaaa), which is either two consecutive V—memory locations or an 8 digit constant value, into the accumulator.</td>
<td><img src="image" alt="LDD Diagram" /></td>
</tr>
<tr>
<td>OUT</td>
<td>a 16 bit instruction that copies the value in the lower 16 bits of the accumulator to a specified V—memory location (Aaaa).</td>
<td><img src="image" alt="OUT Diagram" /></td>
</tr>
<tr>
<td>ADD</td>
<td>a 16 bit instruction that adds a BCD value in the accumulator with a BCD value in a V—memory location (Aaaa). The result resides in the accumulator.</td>
<td><img src="image" alt="ADD Diagram" /></td>
</tr>
<tr>
<td>SUB</td>
<td>a 16 bit instruction that subtracts the BCD value (Aaaa) in a V—memory location from the BCD value in the lower 16 bits of the accumulator The result resides in the accumulator.</td>
<td><img src="image" alt="SUB Diagram" /></td>
</tr>
<tr>
<td>STRE</td>
<td>normally open comparative contact. The contact will be on when Aaaa = Bbbb</td>
<td><img src="image" alt="STRE Diagram" /></td>
</tr>
<tr>
<td>STRGE</td>
<td>Normally open comparative contact. The contact will be on when Aaaa &gt;= Bbbb</td>
<td><img src="image" alt="STRGE Diagram" /></td>
</tr>
<tr>
<td>STRLT</td>
<td>normally closed comparative contact. The contact will be on when Aaaa &lt; Bbbb.</td>
<td><img src="image" alt="STRLT Diagram" /></td>
</tr>
</tbody>
</table>

Table(3.2): Some Instructions used In DirectSOFT 5
3.2.8.3- PID control in DL450:

A PID (proportional–integral–derivative) controller controls a continuous feedback loop that keeps the process output (control variable) flowing normally by taking corrective action whenever there is a deviation from the desired value (set point) of the process variable (PV). An “error” occurs when an operator manually changes the set point or when an event or a disturbance changes the load, thus causing change in the process variable.

The PID controller receives signals from sensors and computes corrective sum of all previous errors (Integral) and the rate of change of the error (Derivative). This can be stated in equation (5.2):

\[ M(t) = K_c \left[ e(t) + \frac{1}{T_i} \int e(x) \, dx + T_d \frac{d}{dt} e(t) \right] + M_0 \]

where:

- \( K_c \) = proportional gain
- \( T_i \) = Reset or integral time
- \( T_d \) = Derivative time or rate
- \( SP \) = Setpoint
- \( PV(t) \) = Process Variable at time “\( t \)”
- \( e(t) = SP - PV(t) \) = PV deviation from setpoint at time “\( t \)” or PV error.

Then: \( M(t) \) = Control output at time “\( t \)”

DL450 supports the PID loop operation through specifying a memory space (32 words). There's no function block that can be used, instead the following setup is established:

1- Using the programming software used to program the Directlogix DL 450, DirectSOFT 5 Programming, go to PLC > setup > PID.
2- The window shown in Fig(3.11) will appear requesting the starting address of the PID table address.
3- The setup window, shown in fig(3.11,a), will appear requesting some configuration. The Algorism is set to Velocity (Due to the required Flow Control).
4- In the SP/PV tap, Fig (3.11,b) and the output taps fig(3.11,c) the addresses of the SP, PV and the output are to be specified
5- The setup is saved after the previous steps.
Chapter 3: PLC and HMI

FIG(3.11): PID setup (a) configuration (b) SP/PV (C) output
3.2.9- SLC 500 PLC:
SLC 500 PLC is designed by AB and used extensively in industry. The SLC 500 PLC is programmed using the RSlogix 500 programming software. The software is used in constructing the ladder logic and communicate with connected PLC’s through the communication package associated with it "RSlinx". The RSlogix 500 comes with an emulator unlike DirectSOFT 5. For that it's used to redesign the original code in the factory and simulate it. The programming is similar to the directSOFT 5 with simple differences

3.3- Human Machine Interface HMI:
3.3.1- Definition:
It is the user interface in manufacturing or process control. It mostly provides a graphics-based visualization of an industrial control and monitoring system. According to Schneider Electric: “Human Machine Interface (HMI) equipment provides a control and visualization interface between a human and a process, machine, application or appliance”.
The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions.

3.3.2- Historical briefing:
There is no quite obvious time-line for the evolvement of HMIs, but we will try to distinguish certain time eras of evolvement according to the dominant vendor and related technology;
1- 1920s through 1940s: Telemecanique invented the first contactor in 1924, and hence a strong reputation for powerful control and power solutions.
2- 1940s-1960s: Founded in 1947, EAO has a global reputation for being an expert partner for Human Machine Interfaces across a range of industries including: transportation, machinery, telecommunications, process control, lifting and moving, and automotive. EAO manufactures high-quality Human Machine Interface products and solutions, from switches, keypads and keyboards to complete custom built control panels.
3- 1980s: witnessed the establishment of the well known world-dominant HMIs vendor Wonder Ware. Wonderware, a business unit of Invensys plc, is the world’s leading supplier of industrial automation software. Founded in 1987, Wonderware's first and best-known product is InTouch(r) human-machine-interface (HMI) software, the first object-oriented software product based on Microsoft(r) Windows(r) for the plant floor in our project we used InTouch.
4- 1990s and 2000: the evolution of PC-based HMIs and their graphical and archiving capabilities and networking features.
The above arrangement referred to all HMI kinds, which may typically be categorized as follows:
1- Panel based HMIs: this type includes a numerous technologies, ranging from small touch screens to huge mimic boards or bench boards < with lots of switches and light indicators >.
2- PC-based HMIs: which are now tending to be the typical abbreviation of the term HMI i.e. commonly used to the degree that when referring to HMI they will be the first thing to think of. From their name, its obvious that they make use of the computer technology available today.
3- Handheld small devices that are used remotely to control machinery and process, these includes cell phones and a other devices.
3.3.3- HMI concept and standards:

In order to design an HMI there are several features that this design should include, and several standards that it should follow on both hardware and software levels;

3.3.3.1- Hardware standards and choices:

A selection criteria is an indication of how to choose the appropriate hardware for the HMI deployment; several factors affects the selection;

1- The complexity of the process i.e. the amount of data and the need for archiving.
2- The placement of the HMI whether it is needed locally, i.e. embedded with the process or a remote location is used for supervision.
3- Economical feasibility.
4- The control of access and safety issues.

In our project we chose the second type i.e. PC based HMI, considering all the above factors.

In this category there are several issues; considering a PC based HMI;

1- The database issues; i.e. the memory amount needed.
2- Networking capabilities; i.e. if the computer used would be connected to other HMIs on a peer-to-peer basis or on a master/slave base.
3- The processing capabilities, if some of the calculations load is transferred to the computer from the field controllers.
4- The display unit used, which may impose some restrictions upon the hardware used, i.e. if to use one large display unit and several other units each showing one window and all connected to the same server, thus some sort of a port or switch would be needed e.g. KVM.

And if to use a different arrangement, in our case we used different arrangement that would be explained in the Chapter 4.

Note: a KVM is a switch that is used to connect several terminals, particularly keyboards, mice and display unit to one server or vice versa, this is obvious from the abbreviation Keyboard, Visual display and Mice. It works on a daisy chain basis for pulling the terminals, i.e. the request is received by all the peripherals and the appropriate one issues the response upon checking a certain addressing signal.

3.3.3.2- Software standards and choices:

There are several software packages that are used to develop graphical based HMIs, these can be categorized as follows;

1- Propriety software: theses work with a certain type of hardware, mostly issues by the same vendor, e.g. WinCC for Siemens.
2- Third party applications: theses work with a wide spectrum of hardware configurations and vendors e.g. Wonder ware InTouch.

Any HMI development software should include at least the following features;

1- A GUI for creating of screens and windows: a graphical user interface is inevitable and necessary for the configuration of graphical elements that represents the process.
2- Window maker or workspace tools and features; these include the symbols used to represent the actual external and internal tags, e.g. symbol factory in InTouch and other drawing utilities.
3- A tags data base; this maybe further divided into tow kinds; a runtime database and configuration time database.
Note: a tag is a memory element that represent a certain I/O to the controller and periodically read and displayed in the PC based graphical HMI by an element. A tag is a variable, that may be eternal or internal i.e. represents an actual I/O or an I/O effect respectively.

4- Alarms and events management system: this is the most important function that the HMI is designed for, to be able to acknowledge alarms and archive them.

5- Recipe management system: a powerful common feature to all new HMI development software that manages the amount of additives in a process automatically.

6- A runtime machine: real time configures that runs the configurations and controls the process.

7- Simulation mode: to be able to simulate the control process for testing and troubleshooting/redesigning.

3.3.4- Wonder ware InTouch:

Wonderware is the market leader in real-time operations management software, founded in 1987. It is a widely used application for creating and running a Microsoft windows based human machine interfaces, InTouch is a component of the Wonderware FactorySuite™.

The application is composed of tow main components:

1- Window maker: the development environment, where object-oriented graphics are used to create animated, touch-sensitive display windows. These display windows can be connected to industrial I/O systems and other Microsoft Windows applications.

2- Window viewer: the runtime environment used to display the graphic windows created in WindowMaker. WindowViewer executes InTouch QuickScripts, performs historical data logging and reporting, processes alarm logging and reporting, and can function as a client and a server for both DDE and SuiteLink™ communication protocols.

Note: DDE and suiteLink are communication protocols heavily used by Microsoft windows based application to exchange data, they use a TCP/IP based standards, DDE stands for dynamic data exchange.

In window maker a designer can configure a PC based HMI by using several helping necessary features, these may be summarized as follows:

1- Tagname dictionary: which represents the variables database and the I/Os representations.

2- Alarms and events management system: to configure the alarms and their conditions, priorities, handling, and logging.

3- Script: to create quick scripts for animations features creations and tags manipulation.

4- Real time/ historical trending: to view certain characteristics of the process in a graphical representations < charts >.

5- SQL access management: to access remote databases and to be able to cross-reference some values.

We used InTouch for the powerful features mentioned above and the fact that it is a third party application that guarantees high performance.
Chapter Four:  
Case Study, Development and Simulation
4.1 Giad steel factory:

4.1.1 Introduction:

Giad industrial area is one of the leading industrial complexes in Sudan with multiple nature activities ranging from steel and cables manufacturing to automobiles and agricultural equipments, with highly standardized deployment of experiences.

Giad steel factory, shown in fig(4.1), is located in GAZEERA state Kamlin province at a distance of 50 KM south of Khartoum, the total area is 250,000 m² with all the future projects covered, the present factory and its accessories located in an area of 100,000 m².

Giad steel factory started with a melt-shop and a barmil in order to enable the country to make use of its iron deposits by erecting-blast furnaces.

![Giad Steel Factory](image)

Fig(4.1): Giad Steel Factory.

4.1.2 brief description:

The main products provided by the factory are:

- Ribbed bars with different sizes ranging from 8mm to 32mm, with annual production capacity of about 180000 ton.
- Billet with annual production capacity of 90000 ton.
- Melt shop: scrap is melted by electrical arc furnace then a ladle arc furnace in which alloying elements are added to produce the required alloy according to the international standards, the liquid steel is then casted into billets with cross-section area ranging from 100*100 mm up to 130*130 mm, with a length of 6m.
- Melt shop consumables: the main consumables include oxygen, limestone and other additives.
- Barmil: in the barmil billets are reheated in a furnace by using residual oil or crude oil then rolled into ribbed bars, during this process regular samples are taken to the lab to determine the physical properties of the product and the degree to which it complies with the expected results, the bars are then passed to the cooling bed and sheared into the required lengths and automatically bundled in two tons bundles and taken the quality dept. this is strictly controlled.
- The metal shop accessories:
  - Fume dedusting unit.
  - Oxygen and nitrogen producing plant with a capacity of 250 m³/hour.
  - SVC unit.
4.2: general process description:

It was aimed to describe the process concerning the phases that the product undergoes and the equipments and components of the system that participate in achieving that.

4.2.1: PFD “process flow diagram “:

PFDs are very powerful tool to describe a process in general without getting involved in the deep technical aspects of the process.

Shown in fig(4.2) a PFD describing the process of casting the scrap to produce billets and the main parts and components participating.

The preceding arrangement gives us a glance over the melting process in the factory which is the topic of our project:

a) The row material (usually scrap) is fed into the electrical arc furnace “EAF” along with the appropriate amount of limestone “5 tankards of scrap for each one tankard of lime stone, 1 tankard = 5 tons “. 
b) The Electrical arc furnace has a group of conditions for its opening:
- Furnace is emptied.
- The EBT is closed.
- Electrodes are at high position.
- Door in low position.

c) It is useful here to mention the type of PLC used which is KOYO a Japanese industrial automation related components manufacturer, The application with which ladder logic is introduced is DIRECTSOFT.

d) It is also useful to say that most of the inputs to the EAF PLC are sensors (limit switches and proximate switches, level sensors, thermostors,…etc) and instrumentation devices (mostly ampere meters and voltmeters from the transformer).

e) Oxygen is provided manually in our case in order to accelerate scrap melt-down.

f) After forming the molten iron, it is transferred to the ladle furnace via “the ladle car”.

g) In the ladle furnace the molten iron is reheated and samples are taken (typically three) and a chemical treatment is done through the additives shown in the above figure with the appropriate ratios; typically:

h) After producing the expected steel “molten”, it undergoes a continues casting process producing a well shaped product “billet typically”.

i) The CCM is formed from:
- A tundish “huge container”.
- A mold “for shaping purposes”.
- Rollers.
- Pusher arm.
- AC motors for pulling the steel.
- A water cooling system “optionally also a secondary cooling system”.

Fig(4.3) shows the map of the Melting House in Giad Steel Factory.

4.2.2: List of components:
This section is to provide an overview of the parts and components of the casting system, typically most of this parts will be further discussed in details in the upcoming sections.

1- SVC: The SVC “i.e. static VAR compensator” is an automated impedance matching device, designed to bring the system closer to unity power factor.

2- Transformers: the main role that these transformers is to provide the required high current to the electrodes for melting purposes ‘18 MW for EAF, and 5 MW for ladle furnace’.

3- Electrodes: their role is to melt the scrap via deployment of high current.

4- EAF: the first step in the casting process, it is fed with the row material and produces molten iron, further details about this furnace will be discussed later.

5- Ladle furnace: after EAF further heating and chemical treatments are applied through this furnace, further details about this furnace will be discussed later.

6- Continues casting machine: its aimed to mostly shape and cool the product, further details about this phase will be discussed later.
7- Hydraulic system: the valves, pumps, oil and water cooling systems involved in the process, every phase of the process has its own hydraulic system feedings, a brief description will be granted later while discussing the three phases of casting.
8- Chemical treatment accessories: includes preservation tanks and valves to control the addition of the substances and pumps.
9- Motors and motor drives: as noticed the manufacturing process involved motion almost in every level, this is done through large AC/DC motors, the control of these motors through PLCs is achieved through typical motor drives e.g. ABB CS 600.
10-PLCs and control panels: these are the brains controlling the whole process, they communicate with the process via a complex system of sensors and actuators, they do also gives indications of the process status.

4.3- Electrical Arc Furnace (EAF):

Generally the electric arc furnace (EAF), shown in fig(4.3), is a furnace used in steel making process by melting the scrap by means of electric arc using graphite electrodes. The melting process is performed to purify the steel and allow it to be processed and casted in the further parts if the steel making.

4.3.1-Components:
The electric arc furnace system existed in the Giad steel factory consists of the following components:
1- Section of sphere shaped, water cooled, refractory-lined roof. The roof is moved with a hydraulic system in two segments: roof up/down and roof swing in/out.
2- A shell, consists of sidewalls and lower steel bowl. The side walls has a door in it called slag door which is used to eject the slag of impurities formed during the melting operation.
3- A hearth, which consists of the refractory that lines the lower bowl.
4- eccentric bottom tap-hole (EBT), by which the molten steel is poured to a ladle car for further process in the ladle furnace.
5- 18 MW-10taps transformer with a vacuum switch and a set of alarms. Also with a On Load Tap Changer(OLTC).
6- Three Graphite electrodes, connected to the 18 MW transformer. The high current (in kilo amps) from transformer when flowing through the electrodes generates an electric arc.

7- Controller(PLC), a Koyo, Directlogic,405 series, DL450 CPU, used in the expansion mode with 3 racks. Connected to a set of input , coming from different buttons and sensors, and a set outputs through relays and contactors. The controller is used to supervise the process by performing a sequencing logic. The logic senses/controls the following:  
   i- Different valves/pumps of the hydraulic system which actuate the movement of the roof, the electrodes, the tilting and taping.
   ii- Different alarms and trips of the Hydraulic system 
   iii-Different alarms and trips of the 18 MW transformer. 
   iv-Signals to the vacuum switch,  
   v- Water cooling on/off,  
   vi-Commands from the control panel.

8- Control Panel, consist of a set of buttons and lamps. Used to initiate different phase of the EAF operation.

9- Sensors and instruments, placed in different parts of the systems to sense the alarms and inputs to the controller. The sensors used are limit switches, approximate switches, thermocouples, level sensors, flow indicator. Instruments include over current relays, earth fault trip relay and different voltimeters and ammeters.

10- Panels, or instrument panels are the panels in which the controllers and relays are placed.

11- Hydraulic system, consists of several valves and pumps to actuate the motion of the roof (up/down, swing in/out), the electrodes (up/down, clamp/unclamp, regulation), furnace body angular movement (+15o, +3o, 0o, -3o, -10o). This hydraulic system operate according to the signals coming from the controller.

12- Electrode regulation system, used to conserve the distance between the electrode and the under melting steel. This distance is conserved to protect the electrodes and extend the life time of them. The regulation system measures the voltage and the current input to the electrodes and through high sampling modules those measures are introduced to a controller. The controller calculate the impedance and the cross distance. Due to this calculation the hydraulic system is controlled to lift or lower one of the electrodes.

13- Cooling system, used for cooling roof, shell and hearth.

14- Manual Oxygen injection system, to provide the oxygen needed by the EAF to de-slag the impurities.

15- Thermocouple, mentioned here as a separate component because the system has a special thermocouple used to measure high temperature , up to 1600o, the thermocouple is placed inside the molten steel at the end of each phase manually by an operator. The measures appear on special display inside the control room.

16- Fume Dedusting system: consists of an array of filters used to dust out the gases produced during the electric arc furnace.
4.3.2 Process Description:

In Giad steel factory the operation of melting using the electric arc furnace follows the following sequence:

1- Furnace charging.
2- Melting.
3- Refining.
4- De-slagging.
5- Tapping.

The steel is molten in batches called "heats". For each heat the above sequence of processes, which further described later, is performed. This operating cycle is called tap-to-tap cycle. The tap-to-tap cycle in the electric arc furnace Giad steel factory takes in average about 60 minutes.

In the following sections, each operation of the tap-to-tap cycle is described as a process controlled by a PLC, i.e in term of inputs and outputs.

1- Furnace Charging:

To begin the furnace charging, the roof should be up, swung out and the electrodes are lifted. This is accomplished by an operator pressing a button in the control panel. Lifting roof, swinging out, electrodes lifting are performed by the hydraulic system after receiving run signal from the controller. When the operator press the lift and swing button, the controller check the general condition for lifting and swinging back the roof and the electrodes. The conditions are:

i- EAF in vertical position.
ii- Furnace interlock in switch is closed
iii- The vacuum switch is open, i.e no power injected from the transformer to the electrodes.
iv- Hydraulic system is ok.
v- Emergency button is not pushed.

If those conditions are fulfilled the controller send a signal to the hydraulic system according to the orders given by the operator. Limit switches are dedicated to sense the completion of this operation and give signals to the controller. The controller lights lamps as indication. Once the operator sees those indicators he gives another operator placed in the loading bay the admission to start charging. Using a crane the loading bay operator charges the furnace with 5 baskets 5 ton each of scrap. The scrap is layered carefully in the baskets to enhance the melting operation. A basket of lime stone is charged with the scrap to manipulate the carbon percentage in the steel and help in the formation of the slag layer. The baskets are buckets with a moveable bottom. The operator of the crane places the basket over the furnace hole and opens the bottom to empty the basket.

As a final phase of the furnace charging, the roof and electrodes are swung in and/or lowered. By the end of the furnace charging the controller indication completion of lowering roof and electrodes the steel is ready to be melted.

2- Melting:

The melting operation is the main operation in the electrical arc furnace. This operation is performed by supplying the furnace interior with energy. The energy can be chemical or electrical. The chemical energy is given by the oxygen injection system manually. The heat from this energy is a product of the chemical reactions happen in the furnace. As mentioned previously this system is...
manually controlled i.e an operator is dedicated for this purpose and the controller does not involve in this operation. However the controller is involved in the supplying of the electrical energy by sending the signal to the vacuum switch to close and supply the electrodes with the high current to establish arc. When an operator place the vacuum switch control selector on ready the controller checks the following conditions:

i- EAF between +3 and -3,
ii- Roof is not in the high position/ roof in the low position,
iii- Roof is not swung back / roof is swung in,
iv- Electrodes clamping is ok,
v- Voltages are ok:
   - 220 voltage is ok,
   - 24 voltage DC for inputs is ok,
   - 24 voltage DC for solenoids is ok.
vi- EBT device closed.
vii- No transformer trip:
   - No Buchloz tank relay trip,
   - No winding temp. trip,
   - No transformer oil temp. trip,
   - No oil tank level trip,
   - No OLCT oil level trip,
   - No pressure relied device trip,
   - No transformer water flow indication trip,
   - No OLTC protective relay trip,
   - No over current relay trip,
   - No earth fault trip.

x- Hydraulic is OK.
   - Hydraulic temperature is below at max,
   - (Pump1 is running and poppet valve 1 is open) or pump2 is running and poppet valve 2 is open),
   - Accumulator oil level is above minimum.

xi- Emergency button not pushed.

The controller sends an indication the vacuum switch is ready if the above conditions are fulfilled. The operator spots the indicator lamp and change the selector from ready to on. According to this a signal is sent to the controller which in turn sends a signal to the vacuum switch and once the vacuum switch is closed it sends a signal back to the controller as an indicator of the success of closing the vacuum switch. The controller notify the operator by lighting a lamb.

By switching on the vacuum switch, the power is successfully injected to the electrodes. The high current injection to the three electrode causes the arc required to heat the steel. By establishing this arc the melting phase is initiated. During the melting an operator will observe the measures of currents and voltages to lower or raise the transformer tap using the OLTC "On Load Tap
Changer". Using one of two buttons (tap lower button, tap raiser button) the operator sends orders to the controller. The controller check the current tap and moves up or down according to the order given by the operator. The tap changing is necessary because of the variation of the power requirements as the temperature is increased in the electric arc furnace.

Also during the melting an additional controller is used as an electrode regulation system. The system used the measures of the voltages and currents to control the arc length of each electrode. The operation is performed by a separate controller due to the use of high sampling speed analog input/output modules and the powerful CPU.

Once the steel is completely melted, a bath temperature and sample will be taken. The temperature is taken using a simple data acquisition system consist of a thermocouple, transmitter and a display unit. Due to the very high temperature no continuous reading of the temperature will be given. Instead an operator will place the thermocouple in the molten steel almost at the end of each phase. The sampling is established by the operator when pressing on "sample collection position" button. This button interrupt the auto operation of the electrodes regulation system and lift them allowing the operator to collect samples. According to Lab testing results of the sample the supervisor determines the amount of oxygen to be blown during refining phase.

3- Refining:

In the refining phase involved the removal of phosphorus, sulfur, aluminum, silicon, manganese and carbon from the steel. This operation is performed by the manual ignition of Oxygen. The controller does not contribute in this phase of operation except sending signals to the Electrode regulation system and continuously check the condition of vacuum switch closed and sends signals to the OLTC according to the operator orders. Also the controller is responsible of executing the sampling order given by the operator at the end of this phase.

4- De-slagging:

During melting and refining operations, some of the undesirable materials within the bath are oxidized and form a layer on the top of the molten steel. It's required to remove this layer.

It's necessary to mention here that the electric arc furnace can controlled from 2 rooms, Main room and EBT room. The main room is the room in which all the previously mention order come from. The EBT room has a control panel to optionally control the de-slagging and the taping operation. It's also necessary to mention that the control is transferred to the EBT room by pushing a button existed in the main room. The controller is responsible of this transition of control.

According to an operator, in the main room or the EBT room, observation and by the end of the refining operation a "slag out" button is pushed. By giving this order the electric arc furnace will be tilted up to an angel of -10o. It's obvious from the previous description that the vacuum switch won't be closed unless the electric arc furnace angular position is between +30 and -30o, so automatically the switch will open once the electric arc furnace is tilted for more than -30o. The system gives two options for slag out tilting, low speed tilting and normal speed tilting. This two options have a selector. Each option is selected according the mechanical requirement observed by an operator. Of course before tilting for slag out conditions should be fulfilled:
i- Hydraulic OK.
ii- Roof in low position/roof not in high position.
iii- Roof is swung in/roof not swung out.
iv- Electrode clamping is
v- Slag door is open.
vi- Furnace interlock out.

The angular position of the electric arc furnace is sensed using approximate switches. The maximum tilting for slag out is -100. After performing this operation the electric arc furnace is returned to the vertical position and sampling collection is performed.

5- Tapping:

The final phase of the electric arc furnace tap-to-tap cycle. When tapping the furnace is tilted and the EBT is open to pour the molten steel in the ladle car place under the furnace. The tilting for tapping requires the fulfilling of the same condition of the tilting for slag out, however the EBT opening require the following conditions:

i- Electrodes 1,2 and 3 are in the intermediate position,
ii- The furnace is not at the vertical position,
iii- Vacuum switch is closed,
iv- Hydraulic is ok,
v- Ladle Car at EAF,

By the end of taping the tap-to-tap cycle is completed and the ladle car is driven manually to the ladle furnace for further processing. Once every several heats the electrodes and roof are raised, the furnace lining is inspected for refractory damage, the EBT is filled with sand and If necessary, repairs are made.

4.3.3 Logic analysis:

The flow chart, shown in fig (4.4) gives a quick overview upon the ladder logic downloaded in the controller for EAF controller.

Key factors about the flow chart:

- the diamond shapes represent check operation and conditional instructions (i.e. to check whether the FUB ON conditions are fulfilled, if yes follow a certain path, if not ; the other ).
- Parallelogram shapes represents execution of a certain logical sequence or task, this task may be a sub-routine, a function block, or a procedure.
- The flow chart used here is too abstract for simplicity reasons, thus comments about each element are provided.
- The comments refers the reader to the rungs range this particular element occupies, the ladder logic is provided in the appendices for more convenience.
- In contrast to the ordinary programming, the ladder logic is excited sequentially and simultaneously, thus the logic spends great deal of the time in the middle region in the flow chart, i.e. the start and End instructions are for convenience purposes, and this is not performed repetitively as may rise to the readers mind.
The flow chart is meant to represent a reference for typical applications, thus the same arrangement may be used regardless of the controller specifications and the process modifications.

Initialization includes setting internal registers with exp. Date, then checking furnace pos. and display this pos. for appropriate action. Rungs 1 through 36.

FUB ON conditions includes voltages check, electrodes pos. roof condition, hydraulic and transformer trips check. Rungs 37 through 43.

Tap changing is performed according to the operators order and needed temp. this is done in rungs 44 through 101.

Three samples are taken, conditions for sampling are electrodes pos. and FUB off ...etc, the de slaging is performed through tilting, tapping conditions require opening for EB and ladle car existence, hydraulic system is revised continuesly, general check includes state indicators and emergency checks.

Fig(4.4): EAF Control System Flow Chart.
4.4 Ladle Furnace (LF):

The main purpose of ladle furnace, shown in fig (4.5) treatment is to ensure that the molten steel has the required temperature when the ladle is taken over at downstream secondary metallurgy units or at a continuous caster.

4.4.1 Components:
The ladle furnace system existed in the Giad steel factory consists of the following components:

1- Ladle car, a car hold the ladle bucket, moved manually by the hydraulic system,
2- Roof, water cooled with holes for the electrodes and the enclosure if additives.
3- 5 MW-5 taps transformer with a vacuum switch and a set of alarms. Without OLTC "On load tap changer.
4- Three Graphite electrodes, connected to the 5 MW transformer.
5- Controller(PLC), a Koyo, Directlogic ,405 series, DL450 CPU , used in the expansion mode with 2 racks. Connected to a set of input ,coming from different buttons and sensors, and a set outputs through relays and contactors. The controller is used to supervise the process by performing a sequencing logic. The logic senses/controls the following:
   i- Different valves/pumps of the hydraulic system which actuate the movement of the roof, the electrodes and the ladle car.
   ii- Different alarms and trips of the Hydraulic system.
   iii- Different alarms and trips of the 18 MW transformer.
   iv- Signals to the vacuum switch,
   v- Water cooling on/off,
   vi- Commands from the control panel.
6- Control Panel, consist of a set of buttons and lamps. Used to initiate different phase of the LF operation.
7- Sensors and instruments, placed in different parts of the systems to sense the alarms and inputs to the controller. The sensors used are limit switches, approximate switches, thermocouples, level sensors, flow indicator. Instruments include over current relays, earth fault trip relay and different voltimeters and ammeters.
8- Panels, or instrument panels are the panels in which the controllers and relays are placed.
9- Hydraulic system, there's no separate hydraulic system for the ladle furnace however the movement is actuated using the electric arc furnace hydraulic system. The system is responsible of movement of the ladle car across the rail, the electrode clamping, roof up/down swing back/out and the electrode vertical movement.
10- Electrode regulation system, same as the one used in EAF.
11- Cooling system, used for cooling roof, and the ladle body.
12- Manual Nitrogen injection system, connected with pipes and valves to the bottom of the ladle bucket. Used to move the create dynamic in the molten steel
13- Manual Oxygen injection system, to provide the oxygen needed by the EAF to de- slag the impurities.
14- Thermocouple,

4.4.2- Process Description :

After the melting the steel in the electric arc, the temperature of the molten steel is around 1600oC. a buffer operation is carried before sending the molten steel to the CCM "Continuous Casting Machine". This buffer operation is the Ladle furnace heating. The steel needed to be processed in the CCM unlike the steel come directly from the EAF has to be homogenous in temperature, this characteristic is prepared by the ladle furnace. Steel grading, which means set the composition of the steel by certain percentage of additives, is also set in the ladle furnace.

The sequence of the ladle furnace is stated as follows:
1- The ladle car is driven by an operator for EAF to the furnace position. The position is detected using 2 limit switches. When the operator press the forward button in the control panel the controller examines the following conditions:
   i. LF vacuum switch is off/ not on,
   ii. Electrodes are high,
   iii. Roof is high,
   iv. Ladle car stop switch is not pressed,
   v. Ladle car is not out of service,
   vi. Ladle car is not overloaded,
2- Once the ladle car hit the limit switches, an indicator lights. An according to the operator command the controller lowers the roof after examining the following conditions:
   i. Hydraulic exists,
   ii. Electrodes are high,
3- The electrodes are now lowered, either manually or automatically.
4- The electrodes and the furnace in this step is prepared to the power injection, allowing the high current injection is done after checking the following conditions:
   i- Voltages are OK.
   ii- Alarm2 not triggered, the alarm is triggered by one of the following transformer trips:
      • Buchhloz  tank relay trip,
      • Oil temperature trip,
      • Pressure relief device trio,
      • Over current trip,
      • Transformer winding temperature trip,
      • Water flow trip
   iii- Tap alarm is not trigger, the triggering occurs when the transformer tap is mistakenly placed.
   iv- Hydraulic exists,
   v- Ladle car at LF,
   vi- No Over current trip,
   vii- No earth fault,
   viii- Electrode clamping is ok,
5- After starting the ladle furnace operation, like the eaf, the controller waits for the operator commands of tap changing. During the operation Nitrogen is injected using the manual injection system, additives are added in the desired level.

6- After about 30 minutes a sample is taken and the temperature is measured. If those to characteristics are probably set, then the roof and electrodes are raised, the ladle car is driven to the casting side.

4.4.3- Logic analysis:
The flow chart, shown in fig(4.6), is constructed according to the key factors mention before in section (4.4.4)

![Flowchart](image-url)
4.5 Continues casting machine:
The third stage through casting process, where by the molten steel acquires a pre-specified shape (billet), through rolling and cooling, is the Continuous Casting, which is performed by the Continuous Casting Machine (CCM), shown in fig(4.7).

4.5.1 Components and main structure:
The continues casting machine can be configured in different arrangements, which include:
- Vertical configuration.
- Vertical with bending configuration.
- Curve type configuration.

Giad steel factory adapts the last configuration; in this configuration the molten steel poured from the ladle undergoes a pre-specified set of actions performed by various components; these are:

1- Tundish: the shape of the tundish is typically rectangular; tow nozzles are located in the bottom of the tundish for steel distribution among the two strands. The main job is to provide a continues flow of liquid during ladle exchange, and hence keeping the casting speed constant.

2- Mold: it is basically an open-end box structure; its function is to establish a sufficient shell to contain its liquid upon entry to the secondary spray cooling zone. The primary cooling inside the mold is a water-cooled inner lining, the lining is a copper alloy, mold heat transfer is critical in the CCM design, computer modeling programs are used for this purpose.

3- Mold oscillation system: the purpose is to minimize friction and sticking of the shell and avoid shell tearing, and liquid steel breakouts, which can wreak havoc on equipment and machine downtime due to clean up and repairs, it is achieved through motor-driven cams, the motors are PLC controlled.

4- Secondary cooling system: a combination of air and water is sprayed through a set of zones upon the steel whilst progressing through the CCM, three forms of heat transfer occurs i.e. radiation, conduction and convection. This cooling system enhances and controls the rate of solidification.
5- Straightener / withdrawal system: a set of induction motors responsible of unbending the CCM, the solid shell is subjected to tension, hence a careful design is considered, otherwise product defects will occur.

6- Shears arrangement: after straightening, the strand is transferred on roller tables to a cut off machine, which cuts the product into ordered lengths. The shears have a moving arrangement, this is done through typical DC motors, the controller is responsible for timing adjustment, when the billet pre-specified length passes through the sensor the moving arrangement carries the shear forward while it cuts the billet slowly in order to prevent product edges defects.

7- Rolls: a group of rolls perform the containment of the strand, it is the extension to the secondary cooling region, In order to avoid compromises in product quality, careful consideration must be made to minimize stresses associated with the roller arrangement and strand unbending. Thus, roll layout, including spacing and roll diameters are carefully selected to minimize between-roll bulging and liquid/solid interface strains.

8- Carry-over tables: the billet approaches the pusher while completely solidifies, the carrying arrangement have a set of motors for motion control, external drive arrangement connected to the controller, an economic cooling system is embedded in the table, also a sufficient lubrication system.

9- Cross-transfer car: this car is for lifting purpose, it is implemented with the pusher arm arrangement, its job is to lift the billet for the pusher’s level, this is done through a DC motor.

10- Pusher arm: the arm places the billet in a loading bay, for rolling stage’s Crain to lift it to the next shaping stage.

11- Controller: 3 KOYO DirectSoft PLCs 405 series one is common for specific tasks shared by the tow strands i.e. pusher arm, cross-transfer car, tundish car, billet loading, the other two are one for each strand. The controller used in expansion mode. A set of sensors and actuators is used by the controller to address and control the desired process.

The above arrangement may vary in other steel manufacturing facilities due to technical feasibility and considerations.

4.5.2- Process description:
A predefined sequence is followed in the factory for continues casting, the main purpose is to solidify the molten treated steel, this is done through the following sequence;

- Liquid steel transfer.
- Primary mold cooling.
- Secondary cooling.
- Extraction of billet < dummy bar >
- Shear cutting.
- Loading bay.

In the following section, each operation is described in details, the CCM is located at the end of the ladle car trail, when the limit switch indicated that the ladle is in casting position, a signal provided to a crain operator to lift it to the tundish level, then the first step is initiated;
Chapter 4: Case Study, Development and Simulation of Melting House, Giad steel Factory

1- liquid steel transfer:
The ladle should be exactly above the tundish, two operators are located near the tundish, their job is to monitor the molten-steel pouring from the ladle to the tundish, also the level of the molten steel in the tundish is monitored by these operators (the lack of a level sensor in the quite high temp. in the tundish), the operators have a panel to control the opening of the tundish nozzles through the mold to maintain a constant rate of pouring and level.
The tundish has tow nozzles, for two molds and strands, the order of the opening of these nozzles comes from the operator directly, and the controller’s involvement is halted here.
The controller automatically activates the primary cooling system (a water pump, and mold lubricants pump) and starts the tow mold oscillation motors, after checking the hydraulic status and aux. relays of the motors, a negative strip is used whereby a stroke pattern in which the downward stroke of the cycle enables the mold to move down faster than the section withdrawal speed, this is done after receiving a signal from the limit switch indicating that the ladle is in the casting pos.

2- primary cooling:
The primary cooling copper inner lining is supplied with water through a water pump, the poppet valve controlling the flow has a proximate switch for that purpose, this is controlled directly by the controller which holds the flow constant, the cooling is achieved through the mold to form a shell.
The controller checks the lubrication pump for mold lubrication (lubrication is used to reduce friction and partially for cooling).
As mentioned the controller’s involvement is only in the level of cooling pump, no further participation but in the mold oscillation.

3- Secondary cooling:
A series of sprays are distributed in the rail as the steel progresses through the machine, with setting the mold oscillation system the controller activates these sprays, through appropriate relay connection.

4- Extraction of billet:
When the dummy bar (a bar used to pull the steel through the strand) is in place, a sensor acknowledges the controller, it runs the straightener/ withdrawal induction motors to push the dummy bar upwards the bending configuration and then downwards to pull the beginning-to-solidify steel, the induction motors average speed is 900 RPM, the idea behind using induction motors is to be able to configure more than one speed, to manipulate the rate of casting.
The carrying-table is mounted upon a set of DC motors to progress the billet through the cutting shear and up to the pusher arm.
The controller sets the priority for a specific strand according to the billet presence in the strand.

5- Cutting shear:
The cutting shear operates due to a timer set in the controller for a pre-specified length, the moving arrangement considering the DC motor is firstly checked by the controller, oil level and over temperature trips are checked in the shear hydraulic, then a signal is applied to the DC motor, a constant movement forward is achieved.
The cutting is done and the shear car moves backwards (to home position).

6- pusher arm:
the pusher arm is composed of two arrangements, a lifting car and a pusher, the lifting car has a limit switch attached at its end, when the billet arrives at the loading bay the controller signals the car to lift the billet upwards to the pusher level, then the controller issues a command to the pusher to unload the billet.

The pusher has a lubrication pump as well, the controller feeds the pump appropriately.

**4.5.3- Logic analysis:**

A flow chart is given in fig(4.8), it clarifies the logical flow of the process and how the controller acts;

**Start**

- Steam exhauster

- Tundish cars 1,2

- Rec. pump

- Shear hydro.

- Priority control

- Billet loading

- Pusher arm

- Emergency hydro

**End**

Two sets of speed and locations: low and high, and parking position and casting posting, at the beginning of the casting the controller must make sure that it is in the casting pos. and check the aux. relay and thermal faults of the motor running the car. two tundish cars are present for work continuity.

These two ops. Simultaneously, the recirculation pump is responsible for the steam in the mold cooling chamber, the shears operates with a moving arrangement, a motor driver ABB ACS 500 is present for controller interfacing.

The priority is determined according to the billet presence in each strand and the timing, the loading of the billets is controlled with the cross transfer car, after the dummy bar is withdrawn and the straightener/withdrawal motors run the semi-cooled billets through the rollers. the cross-transfer car lifts the billet to the pusher level for discharging.

The pusher arm function is to discharge the strand line, a DC motor is responsible for the movement.
4.6- Developments:

4.6.1- Introduction:

The case study of Giad Steel Industry – Melting House Control system was successfully accomplished. According to the case study along with some interviews with the technical staff of the factory, Four enhancement were introduced to the factory. The enhancements are stated below:

1- Automatic Oxygen Injection System to EAF.
2- Automatic Steel Composition System in LF.
3- CCM Tundish and Mould level Control system.
4- PC-Based Human Machine Interface "HMI".

This Chapter states the four enhancements in term of problem statement, system specifications, design concept, hardware arrangement.

4.6.2 – Automatic Oxygen Injection System:

5.6.2.1- Problem Statement:

As stated in Chapter 4, Case Study Of Giad Steel Factory Melting House Control System, during the operation of the EAF and as a part of the steel refining process it's required to inject a certain amount of oxygen to refine the steel from impurities. This injection is originally done in the factory using a manual injection system. An operator is required to approach to the slag door and
open a manual valve. The amount of injected Oxygen in the existed system is estimated by the operator. From this description of the existed system the following disadvantages can be stated:

i- Safety violation, due to 1600°C molten steel temperature and the explosion hazard of oxygen Cylinder.

ii- Inexact injection leads to wastage of oxygen and more down time of the furnace due to the possibility of not giving the required purity of steel.

The system to be designed is required to perform the following:

i- Give a volume of 34000 liter of oxygen with flow of 1200 Nm³/h. This is the standard volume needed to be injected for 30 tons of scrap in the electric arc furnace.

ii- After the injection of the standard volume it's required to enable the operator to inject more Oxygen with the same flow but different volume.

iii- The optional injection should not require the operator to approach to the furnace. i.e. an HMI for this system is required.

4.6.2.2- Design Concept:

The flow rate of the Oxygen is required to be maintained as constant. The flow rate is a function of the pressure and the radius of the outlet tube according to the equation:

\[
Q = \frac{\pi \times r^4 \times \Delta P}{8 \times n} \quad \text{equ}(5.1)
\]

\[
P_2 = \rho \times g \times H \quad \text{equ}(5.2)
\]

Where:
Q: Flow rate.
r: radius of the outlet pipe.
\( \Delta P \): Differential Pressure (between the cylinder and the outlet pipe) = \( P_2 - P_1 \)
n: viscosity for oxygen = 0.00020

\( P_1 \): outlet pipe pressure (assumed to equal to the ambient pressure)
\( P_2 \): Pressure in the storage cylinder (assuming all the Oxygen in the tank is liquid and no vapor pressure).

Equation(5.1) shows that the flow rate of oxygen can be controlled with either varying \( r \) or varying \( P \). The pressure can be controlled using a compressor while the radius of the pipe can controlled using a control valve. Controlling the pressure requires a compressor that means extra components to install and more power consumption. In contrast controlling the radius of pipe through a valve is much simpler specially if the right valve is found. A closed loop control system is set up using PID controller.

4.6.2.3- Implementation:

The PID controller used to implement the design concept mention in previous article is the same PLC used to control the operation in "EAF", which is Directlogix DL 450. This PLC supports the PID loop operation through specifying a memory space (32 words). There's no function block that can be used. The setup was described in article (3.2.8.3).
- Analog modules are needed to input the PV and PC. As shown in fig(4.10)
- The analog input is received from the flow meter, the analog output is given to the control valve, shown in fig(4.11).

4.6.3- Automatic Steel Composition System in LF:
4.6.3.1- Problem Statement:

The main purpose of the ladle furnace is to adjust the composition of the molten steel coming from the EAF. Originally the addition of the substances is done by loading the powders in the ladle manually. The disadvantages of the existed system are listed below:

i- Requires hand calculations based on lab results – wastage of time.
ii- Requires operators to carry heavy loads of powders.
iii- Inexact weighting of the components.
iv- More downtime of the system.

The proposed system is required to do the following:

i- Based on a lab results and recipe selection given through an HMI, it calculates the required weight of each substance.
ii- Add the calculated amount based on the enable of an operator through an HMI order.
iii- Give proper storage of the powder and indicate the unavailability of one of the substances.
4.6.3.2- Design Concept:
The design is based on accumulating the mass flow meter until the desired quantity is fulfilled. The HMI due the calculation for the desired amount of substance needed. The equation of flow will be as follows:

\[ M = M_0 + (\rho \times Q \times t) \]

Where:
- \( \rho \): Substance Density.
- \( Q \): Average volumetric flow rate in period (0 - t).
- \( t \): time.
- \( M_0 \): Initial Mass of the substance.
- \( M \): mass of the substance at time \( t \).

4.6.3.3- Implementation:
The following code in fig(4.12) was designed, the inputs, outputs, coils and memory words are defined as follows:
- \( X_{210} \): enable of the addition of a substance.
- \( Y_{200} \): order to the powder valve of the substance tank.
- \( T_{100} \): Timer resets itself every 1 sec.
- \( V_{4000} \): memory pointer to the analog input from the flow meter.
- \( V_{5000} \): memory word saves the accumulating flow rate.
- \( V_{5020} \): memory word contains the desired

The Hardwar arrangement is shown in fig(4.13)
4.6.4- CCM Tundish and Mould level Control system:

4.6.4.1- Problem Statement:

The Mould level control is to ensure that the molten steel does not spill out the mould. The tundish level control maintaining a constant level in the tundish enhance the quality of steel. The existed system has the following disadvantages:

i. An operator is supposed to stand near to the tundish, about 2 feet, observing the mould level by direct eye contact and change the speed using a potentiometer - Safety issues, less quality.

ii. The tundish level control does not exist in the factory.

The designed system is supposed to have the following specification:

i. The system should run fully automatic without any interference to ensure high productivity.

ii. Reliable level sensors work with 1600°C,

4.6.4.2- Design Concept:

The Design concept is the closed loop PID control stated in section (3.2.8.3).

4.6.4.3- Implementation:

Three PID loops are required to control the valves in the ladle bucket and the speed of the CCM strands. Note that there’re are three PLC’s to operate the CCM. In the common PLC the Tundish level PID loop is implemented. For each of the strands PLC’s a PID loop are setup. The same setup of PID used in section (3.2.8.3)

Level sensors that operates at 1600°C are needed shown in Fig(4.14). see appendix G;
4.6.5- Human Machine Interface design:

4.6.5.1- Problem statement:
The existing user interface is a conventional mimic board, thus operators should focus on several lamp indicators, pushing manual buttons and to be close to the process location thus maintaining eye contact with the process variations.
Thus we proposed a different system, our system have the following set of requirements;
- Design of an easy-to-use PC-based HMI.
- The design of the HMI should include most of the process aspects in a generalized form.
- The designed HMI should be restricted to the process description taking into consideration the existing control system hardware components.

4.6.5.2- Design concept:
The HMI used is PC-based, connected to direct-logic 405 PLC, thus a third party HMI designing software package (specifically wonderware Intouch) was used.
Furthermore the design was implemented in two levels:
- Hardware level: by which we mean the arrangement of display units, physical connection media between the PLC and the HMI and so on.
- Software level: by which we mean the arrangement and standard used to implement the graphical display, the tagging and scripting process, and the arrangement of data collection i.e. whether through direct physical connection or through indirect collection using OPC server.
In our implementation we used serial communication through RS 485 for communication, and a display unit towards each process i.e. EAF, LF, CCM regarding the hardware level.
In the software level, we used an indirect method with KEP direct as an OPC server for implementation and RS linx in simulation, the reason why we used the indirect method because of its simplicity.

4.6.5.3- implementation:
A) hardware level:
To configure the hardware level a low cost approach was proposed in which:
- A computer is presented to handle the interface operation of each unit i.e. one computer to serve as a server for EAF, one for LF and one for the CCM.
- The communication media proposed was the Ethernet for speed and availability considerations, Direct Logic 405 uses DirectNet network/ Modbus through H4-ECOM module.
- For CCM the 3 PLCs are connected via a hub to the CCM HMI computer.
Figure (4.15,a) gives the Ethernet module used with KOYO PLCs, to connect several PLCs we use a HUB as indicated above.
**Note: a hub is a device to connect multiple Ethernet devices (mostly twisted pair or fiber optic based) together and make them act as a single network segment.**

Then to obtain a master control station with full authority we use a D-LINK to connect all three computers as suggested by the Fig(4.16).

**Note: D-link is a network adapter, the name comes from the Taiwan based Datex systems inc[1].**
B) **Software level:**
In this level we used the Wonder Ware Intouch software package to design the elements and distribute them among the process oriented windows and to obtain the necessary tagging.
As mentioned earlier we used the indirect method of communication to the machines (PLCs) for simplicity purposes.
The OPC server used was KEP Direct software package, it resides in the computer connected to the PLC to perform the data collection.
The reader may refer to the appendices (appendix xx) for more information about KEP Direct and the OPC concept in general.

### 4.7 Simulation:

#### 4.7.1 Introduction:
Modeling and simulation is often seen as a sub-set of Systems Theory, Control Theory, Numerical Analysis, Computer Science, Artificial Intelligence, or Operations Research.
Recently, modeling and simulation has been slated to become the computing *paradigm* of the future. As a paradigm, it is a way of representing problems and thinking about them, as much as a solution method.
This chapter discusses the simulation that was done, in the first section the tools used was mentioned, where in the second section the simulation procedure was emphasized taking all parts of the process into consideration.
This implies testing of our simulation performance by the well known *random variable generator* method is proposed in the end. See *appendix H*.

#### 4.7.2 Tools:
In our simulation we used a certain approach:

1. For ladder logic configuration we used Rockwell software package < RS logix 500 > for its easiness and simplicity.
2. After writing the ladder logic we downloaded it in RS emulator 500, this software emulates the behavior of an Allen Bradley PLC, so we can be able to analyze the PLC response to various changes applied to its inputs by the OPC client.
3. The OPC server for data collection used is RS linx classic, which provides a powerful mean of data exchange through OPC/DDE standards.
4. For human machine interface configuration and simulations purposes we used Wonderware Intouch 9.5. this software package provides a powerful tools of editing graphical objects to represent the actual I/Os and enhanced animations and recipes and databases management abilities.
5. For simulation testing we used the MATLAB OPC tool box, MATLAB can provide a direct connection to the RS logix, with the power of simulink random generator.
4.7.3- Simulation procedure:
4.7.3.1- Ladder logic editing:

As clarified in the tools sub-section we used RS logix 500 to edit the *redesigned ladder logic*. This copy of the ladder logic is attached with the appendices and included with the CD that contains the simulation files.

The reason of redesigning the ladder logic is to fit our project purposes i.e. many modifications and enhancements was done, and subsequently many parts of the original ladder logic was omitted for simplicity purposes.

All the modifications and enhancements are mentioned in the *developments chapter*, the reader may refer to them individually where due. a snap-shot of the ladder logic used to configure the CCM is shown in fig(4.17).

![Figure (4.17): a snap-shot of the ladder logic used to configure the CCM](image)

4.7.3.2 Connection configuration:

After editing the ladder logic we configure the connections as follows:

1- First we downloaded the ladder logic in our PLC emulator, this is done through the comms menu in the toolbar of RS logix, after that we chose the node, the processor and download the ladder logic in the emulator, shown in fig(7.18).

![Figure (4.18): Communication Configuration for RSlogix](image)
2- after that we set up the OPC/DDE connection using RS linx, this is also quite simple from the topic configurations menu bar.

3- From the special menu in Intouch menu bar, you chose the prespesified topic name and application name.

the window aside gives the configuration dialog box, the suitlink option is very much of a fast DDE link, shown in fig(4.19).

![Figure (4.19): Topic Configuration for Wonderware Intouch](image1)

4.7.3.3- HMI and simulation configuration:

A) Electrical Arc Furnace: “see appendix B”;

In the electrical arc furnace we configured the main monitoring screen as in the figure, with the ability to access other windows.

The animations features of the object on the screen, shown in fig(4.20), is achieved through the use of internal tags, an EAF status button indicates the general conditions of the EAF e.g. electricity (380 AC voltage, and power transformer conditions), besides tilting, tapping, EBT … etc.

Note: by internal tags we mean variables (memory elements) that are allocated within the developing interface to provide calculation capabilities, counting, timing and animation features to the HMI objects but has no external connection to any of the PLC I/Os (maybe initiated upon an I/O activation, but that’s all).
The script, shown in fig(4.21), checks for operator’s order TAP and do the tapping for certain amount of time by tilting the furnace;

\[
\begin{align*}
\text{IF tapping} &= 1 \text{ AND eafangle} < 190 \text{ AND tapback} = 0 \text{ AND electrodeupstatus} = 1 \text{ THEN eafangle} = \\
&\quad \text{eafangle} + 1; \text{ENDIF}; \\
\text{IF eafangle} &= 190 \text{ AND counta} < 20 \text{ THEN counta} = \text{counta} + 1; \text{ENDIF}; \\
\text{IF counta} &= 20 \text{ THEN tapback} = 1; \text{ENDIF}; \\
\text{IF eafangle} &> 180 \text{ AND tapback} = 1 \text{ THEN eafangle} = \text{eafangle} - 1; \text{ENDIF}; \\
\text{IF eafangle} &= 180 \text{ AND tapback} = 1 \text{ THEN tapfinish} = 1; \text{counta} = 0; \text{tapping} = 0; \text{ENDIF}; \\
\text{IF electrodedownstatus} &= 1 \text{ THEN tapfinish} = 0; \text{tapback} = 0; \text{ENDIF}; \\
\end{align*}
\]

Figure (4.21): Script for Tapping order.

B) Ladle Furnace:”see appendix C”;
Same procedure as above was used here to simulate the ladle furnace interface and operation (including the automatic additives system)
the ladder logic and simulation files are attached in the appendices (Appendices B through F) for the EAF, LF, CCM.
LF movement script shown in fig(4.22)

\[
\begin{align*}
\text{IF ladleforward} &= 1 \text{ AND lfposition} <> 1 \text{ THEN ladlecar} = \text{ladlecar} + 5; \text{ENDIF}; \\
\text{IF swinglfin} &= 1 \text{ AND swingfinstatus} <> 1 \text{ THEN swinglf} = \text{swinglf} + 4; \text{ENDIF}; \\
\text{IF rooflfdown} &= 1 \text{ AND roofldownstatus} <> 1 \text{ THEN rooflfup} = \text{rooflfup} + 4; \text{ENDIF}; \\
\text{IF elecrodelfdown} &= 1 \text{ AND electrodelfdownstatus} <> 1 \text{ THEN electrodelf} = \text{electrodelf} + 5; \text{ENDIF}; \\
\end{align*}
\]

Figure (4.22): LF movement.

Note: in the above code, the Intouch checks for the external tagged limit switches ladle_position, roofdownstatus ..etc and performs the swinging and opening of the door, after motion, this methodology is almost typical to any vertical/horizontal movements configured in our simulation.
C) **Continues Casting Machine:**”see appendix D”

According to the process description, we configured the two main components of the CCM as follows:

**a) The casting process:**

During initialization, all position indicators limit switches were set to initial value, see fig(4.23).

The first conditional algorithm (IF conditioning ) tests the limit switches to verify the ladle car position, hence if the operator clicks on the (ON) button in the CCM main screen the crain will automatically conveys the ladle to the casting position above the tundish, a limit switch designated (ladle_casting) will give the signal to the operator to start the casting process by clicking on another button designated (cast ).

When casting; the PLC automatically activates the straightener/withdrawal motor set, this initiates the rollers, and hence a timer is set for the right amount of time (3 seconds in simulation) to start cutting.

With the initiation of cutting shear movement, the strand conditions are latched on , an important development rise here, the control of Tundish/Mold level, which was discussed earlier.

The next code , shown in fig(4.24,a) illustrates the strand and straightener/withdrawal arrangement;

also is a piece of code , shown in fig(4.24,b) represents the cutting mechanism considering one shear, this is typical to the second shear.

- **Note: this code is directly inseparable from the ladder logic configured earlier, many of the tags are internal i.e. cutter1 for the cutting effect, but also these tags, and hence animated movements are initiated only upon external tags toggling e.g. enable_shear.**

1- the pusher arrangement:

The pusher is the final loading elements who’s Function is to load the produced billets into the Loading-bay, this is done through two different entities, a **cross transfer car** and **A Pusher**.

When the billet hits the limit switch located in the end Of the transfer car side, the PLC automatically issues An order to the car’s motor set to rotate, pushing the Car towards the appropriate strand line to lift the Billt to the pusher arm level which automatically Discharges it into the loading-bay.
Chapter 4: Case Study, Development and Simulation of Melting House, Giad steel Factory

Fig(4.25) gives a quick glance (snap-shot) at the CCM interface, the right side of the interface represents the pusher arrangement, whilst the left side represents the casting machine along with the two strands, the tundish and the straightener/withdrawal configuration.
Chapter Five: Conclusion and future work
5.1 Conclusion:

To relate the academic researches in universities to actual industry problems has been a very strong method in developing both fields. In the steel industry; technological improvements in the control of the process have allowed better productivity and quality which enhanced the overall country’s economy.

Several visits were conducted to the factory as will be stated in the log-sheet to the assurance of accuracy of the information within this report and to closely study the factory environment and the system working in it. Also the actual ladder logic exists in the factory was obtained.

For simplification purposes the; the project was divided into two main components; case study and development and interface, these two levels were further divided into sup-levels , three sup-levels each ; Electrical Arc Furnace, Ladle Furnace and Continues Casting Machine, then the case study, analysis , development and simulation was conducted to each sub-level individually.

The case study was conducted with direct reviews and interviews with site engineer and technicians to deduce the overall system characteristics.

Regarding the interface, Wonder Ware In Touch to create and configure a PC based Human Machine Interface was used.

Regarding existing system’s drawbacks; several developments was proposed which covered the basic defects; on the basis of safety and performance, these developments include an automatic oxygen ignition system, auto composition adjustment, and a Continues Casting Machine tundish and mold level control.

A simulation was conducted using Allen-Bradley emulator and software package for its powerful features and simplicity, and a method of testing was proposed using the random generator – poissonly distributed – variable with the aid of MatLab.

Finally it is worth mentioning that we are to continue testing our new system < interface and developments > and to further readjust the ladder logic used in the factory in the upcoming days – a test is supposed to be held in the downtime of the factory soon – .

5.2 Problems and obstacles:

I used the word obstacles deliberately for it describes what is to be listed here more adequately; these obstacles mainly are due to Giad complex regulations and routines that governs the access to information; these maybe listed as follow;

1- The access problem: to access the factory we used to be halted down in the gates for long periods by the security team, although they got used to us, which is something sarcastic, and the entrance permissions were granted in a per-visit basis.

2- No photographing and materials extracting outside the factory complex.

The problems related to development tools and coding is The editor of the PLC used (RS logix 500) doesn’t provide running capability for PID loops, like all other sofwares.
5.3 **Future work and recommendations:**

It should be obvious to the reader by now that this project constitutes a very good start for a series of possible projects that may take place and achieves the value of relating the academic researches directly to the industry.

Concerning Giad steel factory, the project was focused on the melt-shop phase, i.e. the typical future works recommended are:

1- Rolling house automation development.
2- Interface and monitoring development to connect other factories in Giad complex and to introduce SCADA with a master control station.
3- A comprehensive study on industrial automation benefits and drawbacks from both engineering and economical points of view.
References:


Appendix A; Visits Log:

<table>
<thead>
<tr>
<th>NO.</th>
<th>DATE</th>
<th>ACHIEVEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13th NOV.</td>
<td>General overview of rolling line</td>
</tr>
<tr>
<td>2</td>
<td>2nd DEC.</td>
<td>General overview of iron melting and continuous casting stage</td>
</tr>
<tr>
<td>3</td>
<td>6th FEB.</td>
<td>Detailed description of EAF controller and I/Os.</td>
</tr>
<tr>
<td>4</td>
<td>8th FEB.</td>
<td>Detailed description of LF controller and I/Os</td>
</tr>
<tr>
<td>5</td>
<td>10th FEB.</td>
<td>Detailed description of CCM control and I/Os</td>
</tr>
<tr>
<td>6</td>
<td>3rd MAR</td>
<td>Review of process stages</td>
</tr>
<tr>
<td>7</td>
<td>5th APRIL</td>
<td>Review of ladder logic</td>
</tr>
<tr>
<td>8</td>
<td>17th APRIL</td>
<td>Interviews with the staff</td>
</tr>
<tr>
<td>9</td>
<td>3rd MAY</td>
<td>Interviews with the staff</td>
</tr>
<tr>
<td>10</td>
<td>17th MAY</td>
<td>Interviews with the staff</td>
</tr>
<tr>
<td>11</td>
<td>30th MAY</td>
<td>Review of EAF and LF</td>
</tr>
<tr>
<td>12</td>
<td>8th JUNE</td>
<td>Review of CCM</td>
</tr>
<tr>
<td>13</td>
<td>21st June</td>
<td>Appointment of interface testing.</td>
</tr>
</tbody>
</table>
Appendix B; EAF HMI, Development and simulation

EAF Muman Machine Interface, and graphical notations:
**Issue the EAF discharge order**

**Action:**

(IF tapping == 1 AND eafangle < 190 AND tapback == 0 AND electrodeupstatus == 1 THEN eafangle = eafangle + 1;ENDIF)

**Tag:** tapping I/O discrete I:1/11 plc1

---

**Issue the discharge of slag order**

**Action:**

(IF slagout == 1 AND eafangle > 173 AND slagback == 0 AND electrodeupstatus == 1 THEN eafangle = eafangle - 1 ; ENDIF ;)

**Tag:** slagout I/O discrete I:1/9

---

**Loads a basket to the EAF**

**Action:** (tnoapp = 1 ;)

**Tag:**

---

**Introduce power to the electrodes**

**Action:** (Show "Powe In Status Check" ; powerreadyor = 1 ;)

**Tag:** powerreadyor I/O discrete I:3/8

---

**Opens the EAF status window**

**Action:** show window

**Tag:**

---

**Opens the LF window**

**Action:** show window

**Tag:**

---

**Opens the CCM window**

**Action:** show window

**Tag:**

---

**Lift the electrodes upon sampling**

**Action:** (electrode = electrode + 5 ; ENDIF ;)

**Tag:**

---

**Starts the EAF process**

**Action:** (horizontal movement, start loading)

**Tag:** start I/O discrete I:1/0 , loading memory real

---

**Determine number of bascket to be loaded.**

**Action:** (tnoapp = 0 ; start = 1 ;loadingfinish = 0 ;)

**Tag:** tnoapp memory discrete.
**Oxygen injection system, a valve and oxygen tanks.**

**Action:** PID loop is controlling the amount and flow of oxygen, a separate window is used for this action.

**Tag:** PID internal control variable and process variable I/O real

---

**Displays the lap test results considering the composition of steel**

**Action:** display given element’s percentage.

**Tag:** EAFc, EAFmn, EAFp, EAFs

Memory real

---

**Manifests the electrodes actions, i.e. electrodes movement vertically and power in status and so on.**

**Action:** (*IF electrodeup == 1 THEN electrode = electrode + 5 ; ENDIF;  
* IF electrodedown == 1 THEN electrode = electrode - 5 ; eafstatus =" LOWERING ELECTRODES " ;ENDIF;  
* up in tapping, slagout)

**Tag:** pinelectrode memory discrete, electrode memory real

---

**Manifests the roof actions i.e. vertical and horizontal movements.**

**Action:** *IF begin==1  THEN roofup= roofup + 5 ;ENDIF;  
* IF swingback == 1 THEN swing = swing + 5 ; ENDIF;  
* IF swingin == 1 THEN swing = swing - 5 ; eafstatus = "SWINGING IN ROOF" ;ENDIF;  
* IF roofdown == 1 THEN roofup = roofup -5; eafstatus = "LOWERING ROOF" ; ENDIF ;

**Tag:** roofup, swing memory real.

---

**Manifests the EAF body and associated movements**

**Action:** (*IF tapping == 1 AND eafangle < 190 AND tapback == 0 AND electrodeupstatus == 1  THEN  
eafangle = eafangle + 1;ENDIF;  
IF eafangle == 190 AND counta < 20 THEN counta = counta+1 ; ENDIF;  
IF counta == 20 THEN tapback = 1; ENDIF;  
IF eafangle >180 AND tapback== 1 THEN eafangle = eafangle -1 ; ENDIF;  
IF eafangle == 180 AND tapback==1 THEN tapfinish =1; counta = 0; tapping =0 ;ENDIF;  
IF electrodedownstatus == 1 THEN tapfinish = 0 ;  
tapback = 0 ; ENDIF ;)

*(same goes for slag out process )

**Tag:** eafangle memory real,
Notifies a sampling request for LF
Action: --------------
Tag: Samplerequest memory discrete

Notifies a sampling request for LF
Action: --------------
Tag: Samplerequest memory discrete

Manipulates the lap-results display window values in EAF
Action: EAFc = labc; EAFmn = labmn; EAFp = labp; EAFs = labS;
Tag: --------------

Manipulates the lap-results display window values in LF
Action: lfc = labc; lfmn = labmn;
Tag: --------------

Reset results to initial values i.e. 0.
Action: labc = 0; labmn = 0; labp = 0; labS = 0;
Tag: --------------

Changes the amounts of the load.
Action: tonapp = 1; Sapp = 0;
Tag: Sapp memory discrete
Indicates the total amount in tons, manipulates it.
Action: tonapp = 0; Sapp = 1; Tf = Ti;
Tag: tonapp

Manifests the power status and controls the application of power
Action: Hide "Powe In Status Check";
switchonpower = 1; powerreadyor = 0;
Tag: --------------

Manifests the EAF status values
Action: hide window
Tag: --------------
**EAF script:**

IF electrode == 150 THEN electrodeupstatus = 1 ; ENDIF ;

IF electrodeup == 1 THEN eafstatus = "LIFTING ELECTRODES"; ENDIF;

IF electrode <150 THEN electrodeupstatus = 0 ; ENDIF ;

IF electrode == 0 THEN electrodedownstatus = 1 ; ENDIF;

IF electrode > 0 THEN electrodedownstatus = 0 ; ENDIF ;

IF electrodeup == 1 THEN electrode = electrode + 5 ; ENDIF ;

IF roofup == 200 THEN roofupstatus = 1 ;ENDIF;

IF begin == 1 THEN eafstatus = " LIFTING ROOF " ; ENDIF;

IF roofup < 200 THEN roofupstatus = 0 ; ENDIF ;

IF roofup == 0 THEN roofdownstatus = 1; ENDIF;

IF roofup > 0 THEN roofdownstatus = 0 ; ENDIF ;

IF begin==1 THEN roofup= roofup + 5 ;ENDIF;

IF swingback == 1 THEN eafstatus = " SWINGING BACK ROOF "; ENDIF;

IF swing == 250 THEN swingbackstatus = 1 ; ENDIF;

IF swing< 250 THEN swingbackstatus = 0 ; ENDIF;

IF swing == 0 THEN swinginstatus = 1; ENDIF;

IF swing > 0 THEN swinginstatus = 0 ; ENDIF ;

IF swingback == 1 THEN swing = swing + 5 ; ENDIF;

IF startloading == 1 AND cnumber < Tno AND back == 0 AND loading <450 THEN loading = loading + 10 ; ENDIF ;

IF startloading == 1 THEN eafstatus = " LOADING SCRAP";ENDIF;

IF loading == 450 AND  flip < 90 AND count==0 THEN simc1= 0 ; simc2 = 1; flip = flip + 5; ENDIF ;

IF flip == 90 AND count < 5 THEN count = count + 1 ; ENDIF ;

IF flip > 0 AND count == 5 THEN flip = flip - 5 ; ENDIF ;

IF flip == 0 AND count == 5 THEN simc1 = 1; simc2 = 0 ; back = 1 ; ENDIF ;
IF back == 1 AND loading > 0  THEN loading = loading - 10 ; IF loading == 0 AND cnumber < Tno  THEN
cnumber = cnumber +1 ; ENDIF; ENDIF ;

IF loading == 0 THEN  back = 0 ; count = 0; ENDIF ;

IF cnumber == Tno THEN loadingfinish = 1; loading = 0; start = 0 ; cnumber = 0; startloading = 0 ; ENDIF ;

IF swingin == 1 THEN swing = swing - 5 ; eafstatus = " SWINGING IN ROOF " ;ENDIF;

IF roofdown == 1 THEN roofup = roofup -5; eafstatus = "LOWERING ROOF" ; ENDIF ;

IF electrodedown == 1 THEN electrode = electrode - 5 ; eafstatus = " LOWERING ELECTRODES " ;ENDIF;

IF powerin == 1 THEN pinelectrode = 1; ENDIF;

IF powerin == 0 THEN pinelectrode = 0 ; ENDIF;

IF eafangle >= 177 AND eafangle <= 183 THEN efa3 = 1 ; ENDIF ;

IF eafangle <= 177 OR eafangle >= 183 THEN efa3 = 0 ; ENDIF ;

IF slagout == 1 AND eafangle > 173 AND slagback == 0 AND electrodeupstatus == 1  THEN eafangle =
eafangle - 1 ; ENDIF;

IF eafangle == 173 AND counta < 20 THEN counta = counta+1 ; ENDIF;

IF counta == 20 THEN slagback = 1; ENDIF;

IF eafangle <180 AND slagback== 1 THEN eafangle = eafangle +1 ; ENDIF;

IF eafangle == 180 AND slagback==1 THEN slagfinish =1; counta =- 0; slagout =0 ;ENDIF;

IF electrodedownstatus == 1 THEN slagfinish = 0 ; slagback = 0; ENDIF ;

IF tapping == 1 AND eafangle < 190  AND tapback == 0 AND electrodeupstatus == 1  THEN eafangle =
eafangle + 1;ENDIF;

IF eafangle == 190 AND counta < 20 THEN counta = counta+1 ; ENDIF;

IF counta == 20 THEN tapback = 1; ENDIF;

IF eafangle >180 AND tapback== 1 THEN eafangle = eafangle -1 ; ENDIF;

IF eafangle == 180 AND tapback==1 THEN tapfinish =1; counta =- 0; tapping =0 ;ENDIF;

IF electrodedownstatus == 1 THEN tapfinish = 0 ; tapback = 0 ; ENDIF ;
Appendix C; LF HMI, Development and simulation

LF Muman Machine Interface, and graphical notations:

- **Power In**
  - Controls the application of power to LF electrodes
  - Action: "-------------"
  - Tag: startladle memory discrete

- **To CCM position**
  - Controls the motion after LF treatment to CCM
  - Action: "-------------"
  - Tag: startladle memory discrete

- **To Ladle Position**
  - Control the motion of ladle car to LF position to begin the LF treatment
  - Action:
  - Tag: startladle memory discrete

- **Enable Ladle Car Control**
  - Further control of the ladle car motion, in case of limitswitches failure.
  - Action:
  - Tag: "-------------"
Manifests the supply of nitrogen to the LF
Action:
Tag:

Manifests the supply of carbon and manganese for steel composition adjustment
Action:
Tag: lfc memory real, lfmn memory real

Ladle furnace script:

IF ladlecar == 0 THEN eafposition = 1; ENDIF;

IF ladlecar <> 0 THEN eafposition = 0; ENDIF;

IF ladlecar == 350 THEN lfposition = 1; ENDIF;

IF ladlecar <> 350 THEN lfposition = 0; ENDIF;

IF ladlecar == 720 THEN ccmposition = 1; ENDIF;

IF ladlecar <> 720 THEN ccmposition = 0; ENDIF;

IF electrodelf == 0 THEN electrodelfupstatus = 1; ENDIF;

IF electrodelf > 0 THEN electrodelfupstatus = 0; ENDIF;

IF electrodelf == 200 THEN electrodelfdownstatus = 1; ENDIF;

IF electrodelf < 200 THEN electrodelfdownstatus = 0; ENDIF;

IF rooflfup == 0 THEN rooflfupstatus = 1; ENDIF;

IF rooflfup > 0 THEN rooflfupstatus = 0; ENDIF;

IF rooflfup == 48 THEN rooflfdowstatus = 1; ENDIF;

IF rooflfup < 48 THEN rooflfdowstatus = 0; ENDIF;
IF swinglf == 0 THEN swinglfbackstatus = 1 ; ENDIF ;
IF swinglf > 0 THEN swinglfbackstatus = 0 ; ENDIF;
IF swinglf == 108 THEN swinglfstatus = 1; ENDIF;
IF swinglf < 108 THEN swinglfstatus = 0 ; ENDIF ;

IF ladleforward == 1 AND Ifposition <> 1 THEN ladlecar = ladlecar + 5 ; ENDIF;
IF swinglf == 1 AND swinglfstatus <> 1 THEN swinglf = swinglf + 4; ENDIF;
IF rooflfdown == 1 AND rooflfdownstatus <> 1 THEN rooflfup = rooflfup + 4 ; ENDIF;
IF elecrodelfdown == 1 AND electrodelfdownstatus <> 1 THEN electrodelf = electrodelf +5 ; ENDIF ;
Appendix D; CCM HMI, Development and simulation

CCM Muman Machine Interface, and graphical notations:

**Controls the beginning of operation by moving the ladle to casting position**

**Action:**

```
IF load ==1 THEN
    IF ladlecar < 100 THEN
        ladlecar = ladlecar + 10;
    ELSE
        ladlecar=100;
    ENDIF;
ENDIF;
```

**Tag:** load I/O discrete I:1.0/0

**Begins the operation by controlling the strand**

**Action:**

```
cast=1;
```

**Tag:** cast I/O discrete I:1.0/3
**Manifests the ladle, tundish, straightener/withdrawal.**

Action: *IF load == 1 THEN
    IF ladlecar < 100 THEN
        ladlecar = ladlecar + 10;
    ELSE
        ladlecar = 100;
    ENDIF;
ENDIF;
*IF withd_1 == 5 THEN strand_1 = 1; ENDIF;
IF withd_2 == 5 THEN strand_2 = 1; ENDIF;
Tag: ladlecar, withd_1, withd_2, str_1, str_2 memory real

**Shears monitoring and visualization**

Action: IF shear_1 < 20 THEN
    shear_1 = shear_1 + 1; after checking the ladder for certain condition.
Tag: shear_1, shear_2 memory real

**Manifest and visualize the strand1(2)**

Action: IF strand_1 == 1 THEN
    IF billet01 <> 100 THEN billet01 = billet01 + 1; ENDIF;
    IF shear_enable == 1 THEN IF billet01 > 50 THEN cut_1 = 1; ENDIF; ENDIF;
    IF cut_1 == 1 THEN IF cutter_1 <> 275 THEN cutter_1 = cutter_1 + 5; ELSE IF cutter_1 == 275 THEN cutter_1 = 0; ENDIF; ENDIF;
Tag: billet01, billet02, cutter_1, cutter_2 memory real, cut_1, cut_2 memory discrete

**Visualize and controls the pusher arrangement**

Action: IF billet01 >= 100 THEN IF pusher >= 0 THEN pusher = pusher + 5; IF pusher >= 195 THEN pusher = 0; ENDIF; ENDIF;
    IF pusher == 0 THEN billet_01on = 1; billet_02on = 1; IF pusher >= 15 THEN billet_01 = billet_01 + 5; IF pusher >= 60 THEN billet_02 = billet_02 + 5; IF billet_01 >= 175 THEN billet_01 = 0; IF billet_02 >= 110 THEN billet_02 = 0;
    ENDIF; ENDIF; ENDIF; ENDIF;
Tag: pusher, billet_01, billet_02 memory real
**Visualize and controls the tundish/mold level.**

**Action:** PID control trigering  
**Tag:** LEVEL_TUNDISH I:4.0, valve_LADLE O:4.0/0, discharge O:4.1/0 mold_level I:4.2/0 >>I/O real

**CCM SCRIPT:**

```plaintext
IF ladlecar==0 THEN ladle_home=1 ;ENDIF;
IF ladlecar <> 0 THEN ladle_home =0;ENDIF;
IF ladlecar <> 100 THEN ladle_casting=0;ENDIF;
IF ladlecar == 100 THEN ladle_casting=1;ENDIF;
IF load ==1 THEN
   IF ladlecar < 100 THEN
      ladlecar = ladlecar + 10;
   ELSE
      ladlecar=100;
   ENDIF;
ENDIF;
ENDIF;
IF casting == 1 THEN
   IF str_1 < 5 THEN
      str_1=str_1 + 1;
   ELSE
      str_1 = 5;
   ENDIF;
   IF withd_1 < 5 THEN
      withd_1=withd_1 + 1;
   ELSE
```

74
withd_1 = 5;
ENDIF;

IF str_2 < 5 THEN
str_2=str_2 + 1;
ELSE
str_2 = 5;
ENDIF;

IF withd_2 < 5 THEN
withd_2=withd_2 + 1;
ELSE
withd_2 = 5;
ENDIF;

IF withd_1 == 5 THEN strand_1 =1 ; ENDIF;

IF withd_2 == 5 THEN strand_2 =1 ; ENDIF;

IF shear_enable == 1 THEN

IF strand_1 == 1 THEN

IF shear_1 < 20 THEN
shear_1 = shear_1 + 1;
ENDIF;

IF strand_2 == 1 THEN

IF shear_2 < 20 THEN
shear_2 = shear_2 + 1;
ENDIF;

ENDIF;ENDIF;ENDIF;

IF strand_1 == 1 THEN
IF billet01 <> 100 THEN billet01 = billet01 + 1; ENDIF;

IF shear_enable == 1 THEN IF billet01 > 50 THEN cut_1 = 1; ENDIF; ENDIF;

IF cut_1 == 1 THEN IF cutter_1 <> 275 THEN cutter_1 = cutter_1 + 5; ELSE IF cutter_1 == 275 THEN cutter_1 = 0; ENDIF;ENDIF;ENDIF;

IF billet01 >= 100 THEN IF pusher >=0 THEN pusher = pusher + 5 ; IF pusher >= 195 THEN pusher = 0; ENDIF;ENDIF;

IF pusher >= 0 THEN billet_01on = 1; billet_02on = 1;IF pusher >= 15 THEN billet_01 = billet_01 + 5 ; IF pusher >= 60 THEN billet_02 = billet_02 + 5 ; IF billet_01 >= 175 THEN billet_01 = 0; IF billet_02 >= 110 THEN billet_02 = 0; ENDIF;ENDIF;ENDIF;ENDIF;ENDIF;

ENDIF;

IF strand_2 == 1 THEN

IF billet02 <> 100 THEN billet02 = billet02 + 1; ENDIF;

IF shear_enable == 1 THEN IF billet02 >= 50 THEN cut_2 = 1; ENDIF;ENDIF;

IF cut_2 == 1 THEN IF cutter_2 <> 275 THEN cutter_2 = cutter_2 + 5; ELSE IF cutter_2 == 275 THEN cutter_2 = 0; ENDIF;ENDIF;ENDIF;

ENDIF;
Appendix E; electricity status simulation

Muman Machine Interface, and graphical notations:

**Monitoring and controlling of the transformer tapping**

**Action:** (IF tapmov >= 0 THEN tapmov = tapmov - 32 ; ENDIF;)
(IF tapmov <= 388 THEN tapmov = tapmov + 32 ; ENDIF;)

**Tag:** tapmov memory real.

**Visualization of the electricity status**

**Action:** shows typical values

**Tag:** -----------
Appendix F; hydraulic status simulation

Muman Machine Interface, and graphical notations:

Monitoring and controlling of the 2 pumps
Action: manipulates the ladder logic.
Tag: pumptwosw I:3/13, pumponesw I:3/12
>> I/O discrete.

Visualization of the hydraulic status
Action: shows typical values
Tag: p1runind, popetv1ind, p1thermalok, p2runind, popetv2ind, p2thermalok,
hydooltemp, hydacclevel memory discrete
Appendix G: Ladder Logic:

Electric Arc Furnace ladder logic:
## Ladle Furnace Ladder Logic:

<table>
<thead>
<tr>
<th>R011</th>
<th>R012</th>
<th>R013</th>
<th>R014</th>
<th>R015</th>
<th>R016</th>
<th>R017</th>
<th>R018</th>
<th>R019</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

---

**Note:** The ladder logic is a graphical representation used in electrical engineering and automation to show the control logic of a system. Each rung represents a logical step in the control process, with inputs and outputs indicated by various symbols and connections.
Continuous Casting Machine Ladder Logic:
Appendix F; Flow meter sensor specifications;

<table>
<thead>
<tr>
<th>Measuring ranges</th>
<th>Measuring ranges for air/nitrogen</th>
<th>Measured medium conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal diameter</td>
<td>Maximum measuring ranges kg/h Nm3/h(^1) Nl/min(^2) lb/h SCFM</td>
<td>Measured medium temperature, operating temperature</td>
</tr>
<tr>
<td>1&quot;</td>
<td>105  125  2100  300  70</td>
<td>-26 ... 150 °C (-13 ... 302 °F)</td>
</tr>
<tr>
<td>1 1/2&quot;</td>
<td>430  330  5500  940  150</td>
<td></td>
</tr>
<tr>
<td>2&quot;</td>
<td>740  570  9500  1000  330</td>
<td>Measured medium pressure, maximum</td>
</tr>
<tr>
<td>3&quot;</td>
<td>1775 1375 22800 3900 800</td>
<td>Standard 1 MPa (10 bar [145 psi])</td>
</tr>
<tr>
<td>6&quot;</td>
<td>7500 5800 97000 16500 3400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nominal diameter</th>
<th>Maximum measuring ranges kg/h Nm3/h(^1) Nl/min(^2) lb/h SCFM</th>
<th>Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&quot;</td>
<td>130  95  1800  280  55</td>
<td>Weight in kg (lbs)</td>
</tr>
<tr>
<td>1 1/2&quot;</td>
<td>310  200  4300  680  150</td>
<td>Threaded pipe component</td>
</tr>
<tr>
<td>2&quot;</td>
<td>510  390  6500  1120 230</td>
<td>Flange pipe component</td>
</tr>
<tr>
<td>3&quot;</td>
<td>1200 920 15000 2640 540</td>
<td>Nominal diameter kg (lbs)</td>
</tr>
<tr>
<td>6&quot;</td>
<td>5700 4400 73000 12500 2500</td>
<td>1&quot;</td>
</tr>
</tbody>
</table>

\(^1\) Notation also m\(^3\)/h - m\(^3\)/min
\(^2\) Notation also m\(^3\)/min - l/min

All volume flow rate specifications are referenced to 0 °C / 1013.25 hPa (32 °F/14.696 psi).

Output

Analog output signal
0/4 ... 20 mA, switchable

Load
~ 750 Ω, electrically isolated

Characteristics

Measurement deviation
Air, nitrogen  
<1.5 % of rate plus ± 0.05 % of end value
Biogas  
<1.8 % of rate plus ± 0.1 % of end value under calibration conditions in specified flow rate

Reproducibility
< ± 0.25 % of rate, \( t_{mean} = 10 \) s

Response time
\( T_{63} \approx 500 \) ms

Influences

Temperature effect
< 0.05 % / K of measured value

Pressure effect
± 0.2 % / 100 kPa (bar [14.5 psi]) of measured value

Pressure drop
< 1 kPa (10 mbar [0.145 psi]) at full scale decreasing quadratically for smaller flow rates

Ambient conditions

Ambient temperature for flowmeter sensor
-25 ... 70 °C (-13 ... 158 °F)

Ingress protection
IP 65, NEMA 4X

Storage temperature
-25 ... 85 °C (-13 ... 185 °F)

Materials, process connection

Flowmeter sensor  CrNi steel, e.g. 1.4301

Pipe component with external threads
R 1" ... 3"  Galvanized steel

Pipe component with connection flanges
DN 25 ... DN 80  CrNi steel, e.g. 1.4301

Weld-on adapter  CrNi steel, e.g. 1.4301

Connection flanges to EN1092-1 Form B1, PN10

Supply power

Voltage
24 V AC / DC ± 25 %

Power consumption
< 15 W

Power consumption
< 600 mA, recommended fuse of at least 2 A, slow-blow

Cable gland M20 x 1.5

Communication interface

LCI adapter

Accessories (optional)

– Power supply unit
– Display unit
– Integrator with indicator (current pulse transformer)

Parameterize

The output signal of the Sensflo Flow FMT200-D flowmeter can be toggled between 0 ... 20 mA and 4 ... 20 mA. Additionally, a measuring range window can be expanded in such a way that a smaller span corresponds to a 20 mA current signal. A current < 3.5 mA or > 22 mA can be selected for the failure signal.

An LCI adapter is used to parameterize the device. It is possible to change the output signals or the settings of the measuring ranges and signals by using a standard PC or laptop.
KFD-2 MICROWAVVW TYPE FLOW METER

- Flow Meter with continuous output (4-20mA DC)
- This can retrieve such a minute change in concentration that KFD-1 has not been able to do.

**SPECIFICATIONS**

- Model: KFD-2
- Supply voltage: 105/210VAC±10%
- 50/60Hz
- Signal output: 4-20mA DC
- Acceptable temp: -10 ~+55°C
- Color: Gold
- Allowable pressure: 490kPa
- Load resistance: less than 300Ω

**WIRING**

**OUTLINE DRAWING**
Appendix E; Simulation performance evaluation method:

For running a stochastic simulation only method was given, due to the unavailability of the statistics. The method requires connection to Matlab to generate the stochastic variables according to specified distribution. That connection was successfully held.

one may also use a random variable generator through visual basic for applications embedded within the RSLogix 500, the code of the random variable generator is given as follows;

```vbnet
Public Class Form1
    
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim rannumber As Random
        Dim number As Integer
        rannumber = New Random
        number = rannumber.Next(1, 1000)
        TextBox1.Text = number.ToString
    End Sub
End Class
```

The output value of this is viewed in a textbox as obvious, the user may configure this value to manipulate a certain I/O coil value in the ladder logic