People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Echahid Hamma Lakhdar University of El-Oued

FACULTY OF TECHNOLOGY

ELECTRICAL ENGINEERING DEPARTMENT

End of studythesis

Presented for obtaining the

ACADEMIC MASTER

Field: Sciences and Technologies

Specialty: Electrical machines

Thesis

Realization of PLC system based on Arduino Card

Board of Examiners

President: Dr.MAAMIR Madiha
Examiner:Dr.LAOUAMER Mosbah
Supervisor:Dr. CHERIF Hakima

Presented by

- Messai Mohammed larouci
- HassaniYoucef Houssam Eddin
- Bouzid Mohammed Fouad

2021-2022

Acknowledgement

We thank, Allah the Almighty for giving us the will and patience to carry out this modest work.

We would like to express our warm thanks to our mentor Dr. Hakima Cherif for her support, guidance and advice which were very precious to us.

We cannot forget to thank also Mr. President and Gentlemen of the jury, also all the staff of the Faculties of Technology and Exact Sciences of the Echahid Hamma Lakhdar University of El-Oued, in particular

Also, we thanks My School work team for various assistance and support

Dedication

Our dedication goes to our
beloved parents, who have
been along us through the road
leading to this moment.

Table of Content

List of figures i

List of Tables iii

Abbreviations iv

General introduction 1

Chapter I 2

- **I.1 Introduction** 2
- **I.2 The programmable logic controller** 2
- **I.3 PLCs Hardware** 3
- **I.4 Architecture of PLC** 4
  - **I.4.1 The CPU** 5
  - **I.4.2 The buses** 5
  - **I.4.3 Memory** 6
  - **I.4.4 Input/output unit** 6
  - **I.4.5 Sourcing and sinking** 8
- **I.5 Types of PLC** 8
  - **I.5.2 Compact PLC:** 9
  - **I.5.4 Sizes of PLC**: 10
- **I.6PLC Programming Languages** 12
  - **I.6.1 Ladder Logic PLC Programming** 13
  - **I.6.2 Function block diagram (FBD)** 14
  - **I.6.3 Sequential Function Charts (SFC)** 14
  - **I.6.4 Structured Text(ST)** 15
  - **I.6.5 Instruction List(IL)** 16
- **I.8 Conclusion** 19

Chapter II 20

II.1 Introduction 21

II.2 Open PLC Arduino Architecture 21

II.2.1 power supply 21

II.2.2 Microcontroller (Arduino). 21

II.2.3 Input 23

II.2.4 Output 23

II.3 Environment programming (IDE) 24

II.4 Realisation of OPENPLC arduino card 26

II.4.1 The design 26

II.4.2 Printing 27

II.4.3 Installing 28

II.4 Conclusion 29

Chapter III Application the OpenPLC Arduino to control the cold room system ... 30

III.1 Introduction 30

III.2 Generals on cold rooms 30

III.2.1 Definition 30

III.2.2 The method of work 32

III.2.3 The temperature range of a cold room 32

III.2.4 Uses of cold room 32

III.3 controlling the cold room system by OpenPLC Arduino 33

III.3.2 Grafcet 35

III.3.3 Equations: 36

III.3.3 Building the cold room 37

III.3.4 System result 39

III.5 Conclusion 40

General conclusion 40

References 42

List of figures

Figure I. 1: A programmable logic controller(PLC)...2
Figure I. 2: The PLC system...3
Figure I. 3: Basics communication model..4
Figure I. 4: Architecture of PLC...5
Figure I. 5: a - sourcing/ b - sinking...8
Figure I. 6: Modular PLC...9
Figure I. 7: Compact PLC...9
Figure I. 8: PLC product ranges...11
Figure I. 9: (a) Mitsubishi's smallest print size (b)PLC Direct DL105, (c)Giddings &Lewis PIC90, (d)Allen-Bradley's PLC 5/15, (e)Omron's C200H PLC, and (f)Allen-Bradley's PLC 5/80. 12
Figure I. 10: Most popular language of programming[6]...13
Figure I. 11: Ex for Ladder (LD)...14
Figure I. 12: Ex for FBD...14
Figure I. 13:SFC general diagram...15
Figure I. 14:SFC structure...15
Figure I. 15: Ex for ST language...16
Figure I. 16:ex instruction liste...17
Figure I. 17:A sample instruction list (IL) program illustrates the similarity to assembly language programming...17

Figure II. 1: Arduino uno...22
Figure II. 2: Optocoupler ……………………………………………………………23
Figure II. 3: Isolated input of OPENPLC arduino...23
Figure II. 4: Isolated output of OPENPLC Arduino...24
Figure II. 5: Arduino software...25
Figure II. 6: Editor preferences...25
Figure II. 7: Error message...26
Figure II. 8: Electrical circuit of the OPENPLC Arduino...27
Figure II. 9:PCB table of OPENPLC Arduino...27
Figure II. 10:OPENPLC Arduino card without component...28

Figure II. 11: The OPENPLC Arduino card 29

Figure III. 1: Cold room 30
Figure III. 2:cold room from inside 30
Figure III. 3: cold room storage 31
Figure III. 4: Installing of cold room 32
Figure III. 5: Grafcet of the system 36

List of Tables

Table I. 1: Difference between modular and compact PLC [3] 10
Table I. 2: The famous PLC in the world 18

Table II. 1: Structure of arduino 22
Table II. 2: Component of our OPENPLC Arduino card 28

Abbreviations

PLC	Programmable logic controller
CPU	Central processing unit
PCB	Printed circuit board

General introduction

General introduction

Early machines were controlled by mechanical means using cams, gears, levers and other basic mechanical devices. As the complexity grew, so did the need for a more sophisticated control system. This system contained wired relay and switch control elements. These elements were wired as required to provide the control logic necessary for the particular type of machine operation. The development and testing ground for this new means was the U.S. auto industry. The time period was the late 1960's and early 1970's and the result was the programmable logic controller, or PLC[16].

PLC systems dominate the smart controlling system and production lines in factories and institutions from all ranges including food industry, nuclear power plants, cold rooms...etc. In short, they dominate all technological aspects of industrial revolution. However, since it is expensive, and is considered an exclusive technology, it is hard for the weaker countries to possess it.

This became the inspiration for our project, titled "Open PLC Arduino", which seeks to provide a cost-effective industrial environment and available technology. This can be done by developing ATMega dependent board such as the ones used in Arduino. These boards are also compatible with multiple programming languages such as Arduino c, and Ladder programming which is used in PLC units.

The dissertation begins by giving an overview on PLC devices. After, an OPENPLC board combining was tackled in the first chapter. In the second chapter, an OPENPLC board was combined of an Arduino board.

Finally, chapter three comprised the application of the OPENPLC board in controlling a cold room model before moving to results

The dissertation is finally ended with a general conclusion.

Chapter I

The programmable logic controller

Overview

I.1 Introduction

In this chapter, firstly ,the definition of The programmable logic controller(PLC) is presented . Next, the Hardware and the architecture of PLC are described .Then, Types of PLC and their Programming Languages are introduced .Finally, the famous PLC companies are briefly recall.

I.2 The programmable logic controller

A programmable logic controller(PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes and are designed to be operated by engineers with perhaps a limited know ledge of computers and computing languages. They are not designed so that only computer programmers can set up or change the programs. Thus, the designers of the PLC have pre-programmed it so that the control program can be entered using a simple, rather intuitive, form of language. The term logic is used because programming is primarily concerned with implementing logic and switching operations, e.g. if A or B occurs switch on C, if A and B occurs switch on D. Input devices, e.g. sensors such as switches, and output devices in the system being controlled, e.g. motors, valves, etc., are connected to the PLC. The operator then enters a sequence of instructions, i.e. a program, into the memory of the PLC. The controller then monitors the inputs and outputs according to this program and carries out the control rules for which it has been programmed[1].

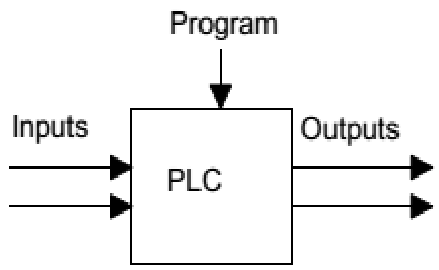


Figure I. 1: A programmable logic controller(PLC)

PLCs have the great advantage that the same basic controller can be used with a wide range of control systems. To modify a control system and the rules that are to be used, all that is necessary is for an operator to key in a different set of instructions. There is no need to rewire. The result is a flexible, cost effective, system which can be used with control systems which vary quite widely in their nature and complexity. PLCs are similar to computers but where as computers are optimized for calculation and display tasks, PLCs are optimized for control tasks and the industrial environment. Thus PLCs are[1]:

1) Rugged and designed to with stand vibrations, temperature, humidity and noise

2) Have interfacing for inputs and outputs already inside the controller.

3) Are easily programmed and have an easily understood programming language which is primarily concerned with logic and switching operations.

The first PLC was developed in 1969. They are now widely used and extend from small self-contained units for use with perhaps 20 digital inputs/outputs to modular systems which can be used for large numbers of inputs/outputs, handle digital or analogue inputs/outputs, and also carryout proportional-integral-derivative control modes[1].

I.3 PLCs Hardware

Typically, a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface and the programming device[2].Figure 2 shows the basic arrangement of the PLC hardware components.

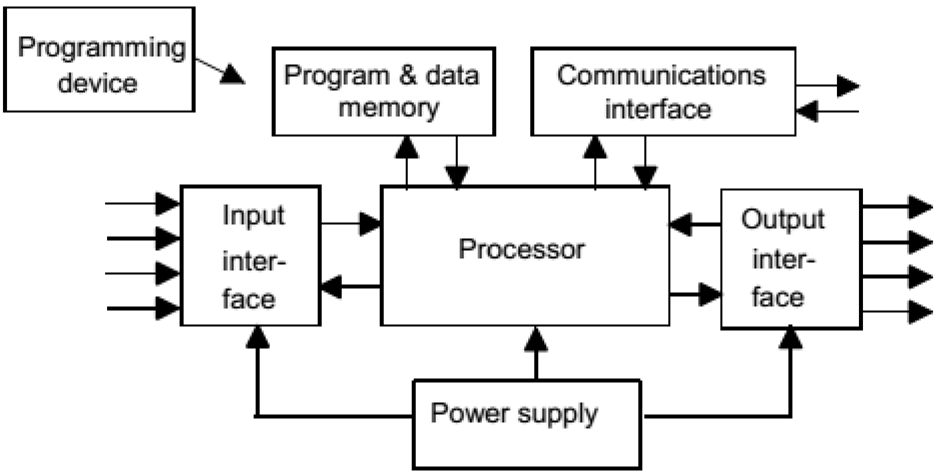


Figure I. 2: The PLC system

1. Stored the program to interpret the input signals and communicate between this inputs and the output decisions to carries out the control actions done in central processing unit (CPU).

2. The power supply convert AC voltage to DC voltage necessary for the processor and the circuits in the input and output interface modules.

3. The program enters into the memory of the processor by using the programming device and then transferred to the memory unit of the PLC.

4. The program is stored in the memory unit to be used for the control actions to exercised by the microprocessor and data stored from the input for processing and for the output for outputting.

5. The input and output sections are where the processor receives information from external devices and communicates information to external devices.

6. finally the communications interface is used to receive and transmit data by means of communication networks from or to other remote PLCs[2].

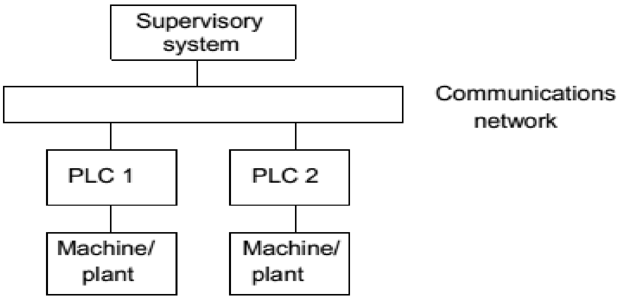


Figure I. 3: Basics communication model

I.4 Architecture of PLC

Figure 1.4 shows the basic internal architecture of a PLC. It consists of acentral processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronization for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called buses. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on aprinted circuit board or wires in a ribbon cable. The CPU uses the data bus for sending data between the constituent elements, the address bus to send the addresses of locations for accessing stored data and the control bus for signals relating to internal control actions. The system bus is used for communications between the input/output ports and the input/output unit [1].

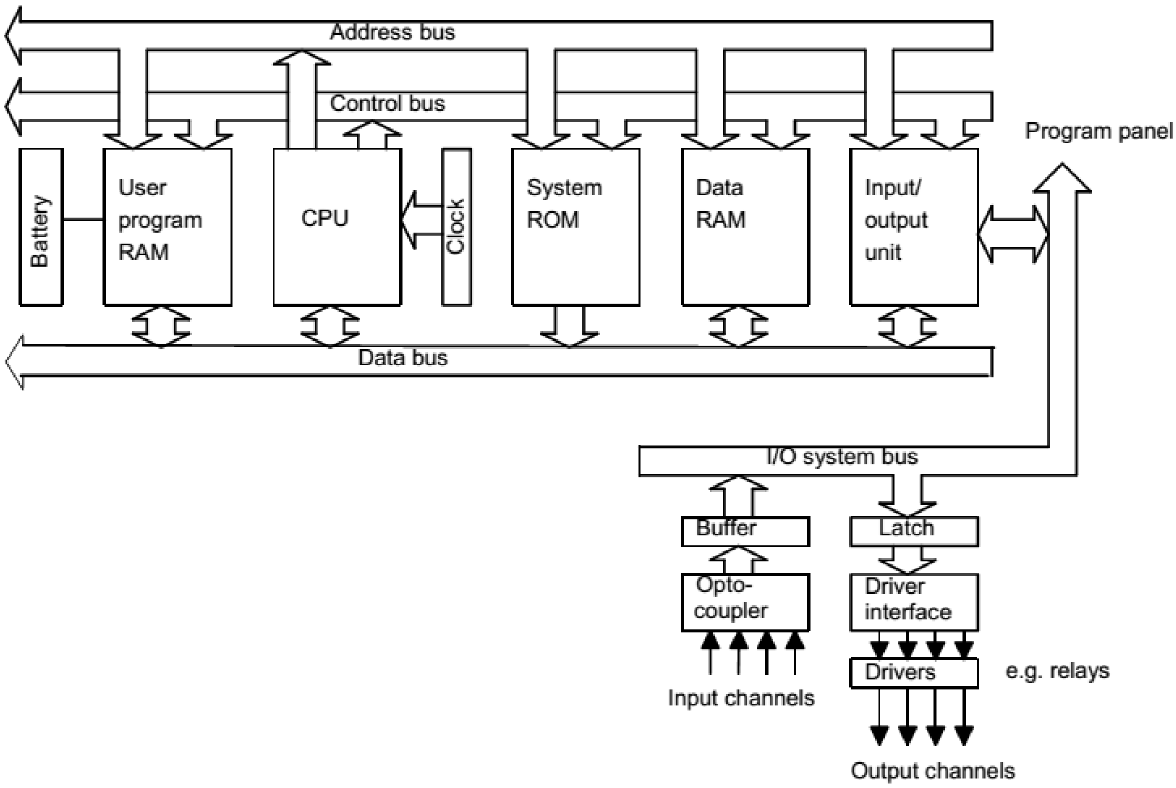


Figure I. 4: Architecture of PLC

I.4.1 The CPU

The internal structure of the CPU depends on the microprocessor concerned. In general they have [1]:

1. An arithmetic and logic unit(ALU) which is responsible for data manipulation and carrying out arithmetic operations of addition and subtraction and logic operations of AND, OR, NOT and EXCLUSIVE-OR.
2. Memory, termed registers, located within the microprocessor and used to store information involved in program execution.

I.4.2 The buses

The buses are the paths used for communication within the PLC. The information is transmitted in binary form, i.e. as a group of bits with a bit being a binary digit of 1 or 0, i.e. on/off states. The term word is used for the group of bits constituting some information. Thus an 8-bit word might be the binary number 00100110. Each of the bits is communicated simultaneously along its own parallel wire. The system has four buses [1]:

1. The data bus carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus which can handle 8-bit numbers.
2. The address bus is used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique address. Just like houses in a town are each given a distinct address so that they can be located.
3. The control bus carries the signals used by the CPU for control, e.g. to inform memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronize actions.
4. The system bus is used for communications between the input/output ports and the input/output unit.

I.4.3 Memory

There are several memory elements in a PLC system [1]:

1. System read-only-memory(ROM) to give permanent storage for the operating system and fixed data used by the CPU.
2. Random-access memory (RAM) for the user's program.
3. Random-access memory (RAM) for data. This is where information is stored on the status of input and output devices and the values of timers and counters and other internal devices. The data RAM is sometimes referred to as a data table or register table. Part of this memory, i.e. a block of addresses, will be set aside for input and output addresses and the states of those inputs and outputs. Part will be set aside for preset data and part for storing counter values, timer values, etc.
4. Possibly, as a bolt-on extra module, erasable and programmable read-only-memory (EPROM)for ROMs that can be programmed and then the program made permanent. The programs and data in RAM can be changed by the user. All PLCs will have some amount of RAM to store programs that have been developed by the user and program data.

I.4.4 Input/output unit

The input/output unit provides the interface between the system and the outside world, allowing for connections to be made through input/output channels to input devices such as sensors and output devices such as motors and solenoids. It is also through the input/output unit that programs are entered from a program panel. Every input/output point has a unique address which can be used by the CPU. It is like a row of houses along a road, number 10 might be the ‘house’ to be used for an input from a particular sensor while number ‘45’ might be the ‘house’ to be used for the output to a particular motor [1].

The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry. Electrical isolation from the external world is usually by means of optoisolators (the term optocoupler is also often used)[1].

The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel, with isolation, enables a wide range of input signals to be supplied to it. A range of inputs might be available with a larger PLC, e.g. 5 V, 24 V, 110 V and 240 V digital/discrete, i.e.on–off, signals [1].

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A a.c., or 240 V, 2 A a.c., from a triac output channel (Figure 1.10). With a small PLC, all the outputs might be of one type, e.g. 240 V a.c., 1 A. With modular PLCs, however, a range of outputs can be accommodated by selection of the modules to be used. Outputs are specified as being of relay type, transistor type or triac type [1]:

1.With the relay type, the signal from the PLC output is used to operate a relay and is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for a.c. and d.c. switching. They can with stand high surge currents and voltage transients.

2. The transistor type of output uses a transistor to switch current through the external circuit. This gives a considerably faster switching action. It is, however, strictly for d.c. switching and is destroyed by over current and high reverse voltage. As a protection, either a fuse or built-in electronic protection are used. Optoisolators are used to provide isolation.

3.Triac outputs, with optoisolators for isolation, can be used to control external loads which are connected to the a.c. power supply. It is strictly for a.c. operation and is very easily destroyed by over current. Fuses are virtually always included to protect such outputs.

I.4.5 Sourcing and sinking

The terms sourcing and sinking are used to describe the way in which d.c.devices are connected to a PLC. With sourcing, using the conventional current flow direction as from positive to negative, an input device receives current from the input module, i.e. the input module is the source of the current. If the current flows from the output module to an output load then the output module is referred to as sourcing. With sinking, using the conventional current flow direction as from positive to negative, an input device supplies current to the input module, i.e. the input module is the sink for the current. If the current flows to the output module from an output load then the output module is referred to as sinking[1].

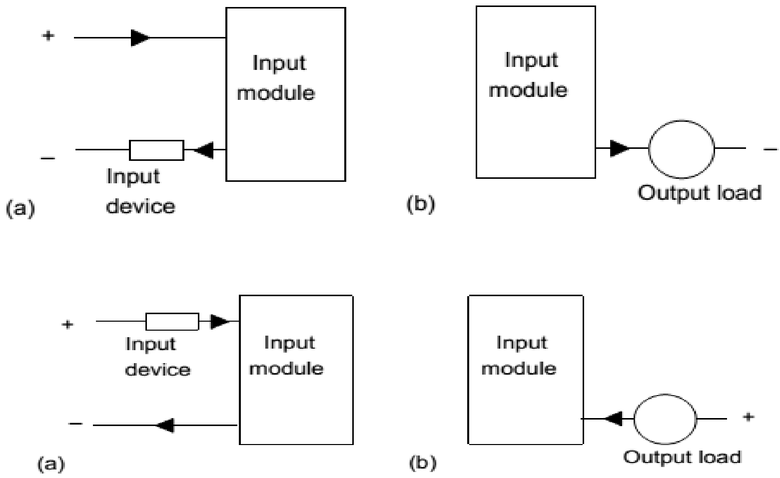


Figure I. 5: a - sourcing/ b - sinking

I.5 Types of PLC

Programmable Logic Controller available in two type, that is Modular and Compact. It's usually depends on the customer requirement and specification for their projects or manufacturing lines[3].

I.5.1 Modular PLC:

Is built with several components that are plugged into a common rack or bus with extendable I/O capabilities. It contains power supply module, CPU and other I/O modules that are plugged together in the same rack, which are from same manufacturers or from other manufacturers. These modular PLCs come in different sizes with variable power supply, computing capabilities, I/O connectivity, etc. Modular PLCs are further divided into small, medium and large PLCs based on the program memory size and the number of I/O features[3].

Figure I. 6: Modular PLC

I.5.2 Compact PLC:

In Compact PLC, I/O capability is fixed and determined by the manufacturer. They are used for small scale uses. A single case contains the multiple built-in modules. The integral PLC has integrated power supply, CPU, I/O interfaces and other components are housed in a compact chassis. Some of the integrated PLCs allow connecting additional I/Os[3].

Figure I. 7: Compact PLC

I.5.3 Difference between modular and compact PLC

Table I. 1: Difference between modular and compact PLC [3]

Compact PLC	Modular PLC
Cannot be easily repaired.	Easy to maintain compared to the compact PLC.
Fixed number of I/O.	Inputs and outputs can be added to the modular PLC systems by the user.
Integrated or Fixed PLC.	Rack PLC.
It has already fitted I/O with CPU.	Several parts are fitted with separate slots on chassis or rack or bus in this PLC.
It is helpful for smaller apps and is best suited for home use.	It is used for industrial purpose and also for future industrial expansion and growth.
Low cost.	High cost compared to compact PLC.
Small in size.	Larger in size, which includes power supply and other additional capabilities.

I.5.4 Sizes of PLC:

The PLC market can be segmented into five groups:

Micro PLCs are used in applications controlling up to 32 input and output devices, 20 or less I/O being the norm. The micros are followed by the small PLC category, which controls 32 to 128 I/O. The medium (64 to 1024 I/O), large (512 to 4096 I/O), and very large (2048 to 8192 I/O) PLCs complete the segmentation [4]. Figures(I.9) and(I.10) shows everal PLCs that fall into this category classification. The differences between PLCs in overlapping areas include I/O count, memory size, programming language, software functions, and other factors. An understanding of the PLC product ranges and their characteristics will allow the user to properly identify the controller that will satisfy a particular application.

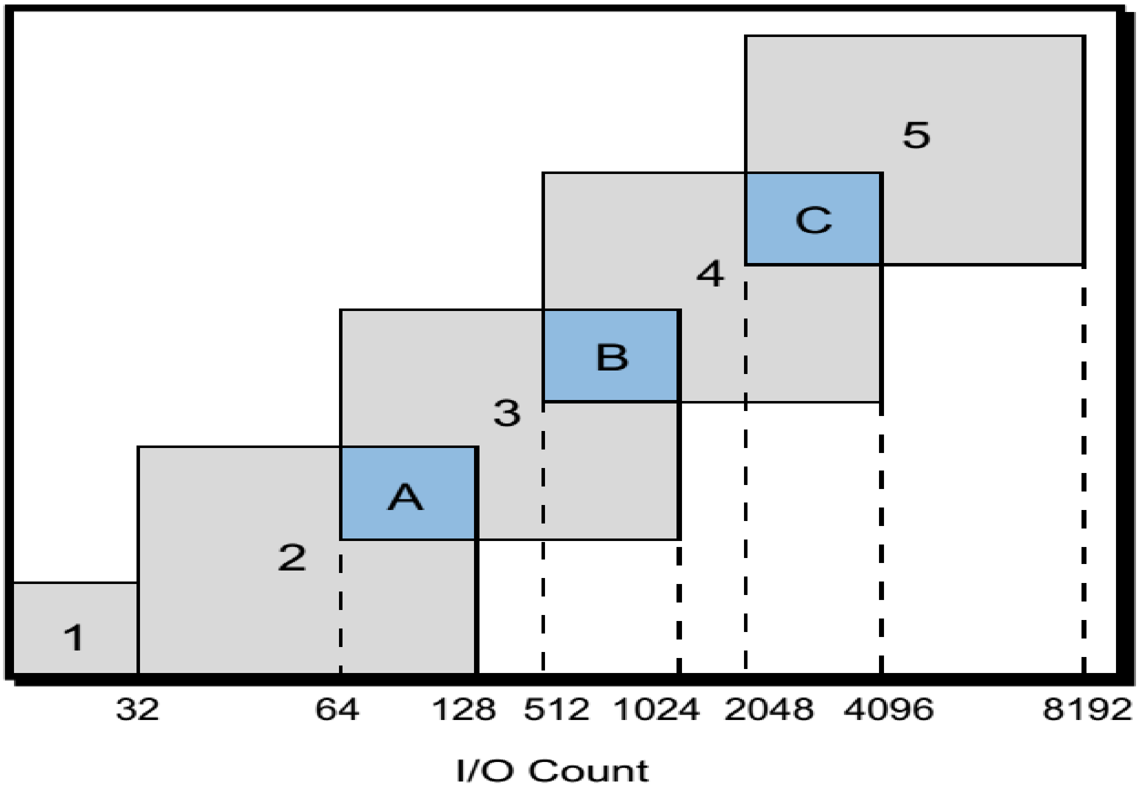
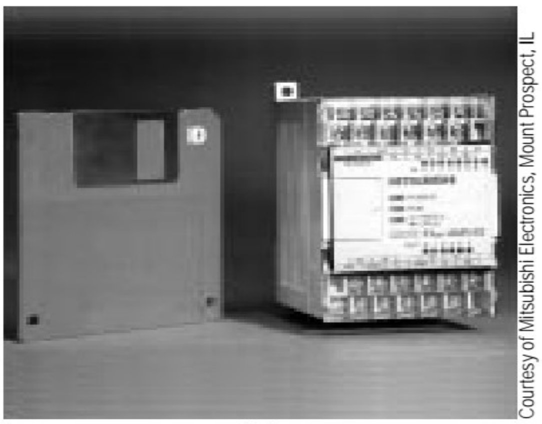


Figure I. 8: PLC product ranges



(a)

(b)

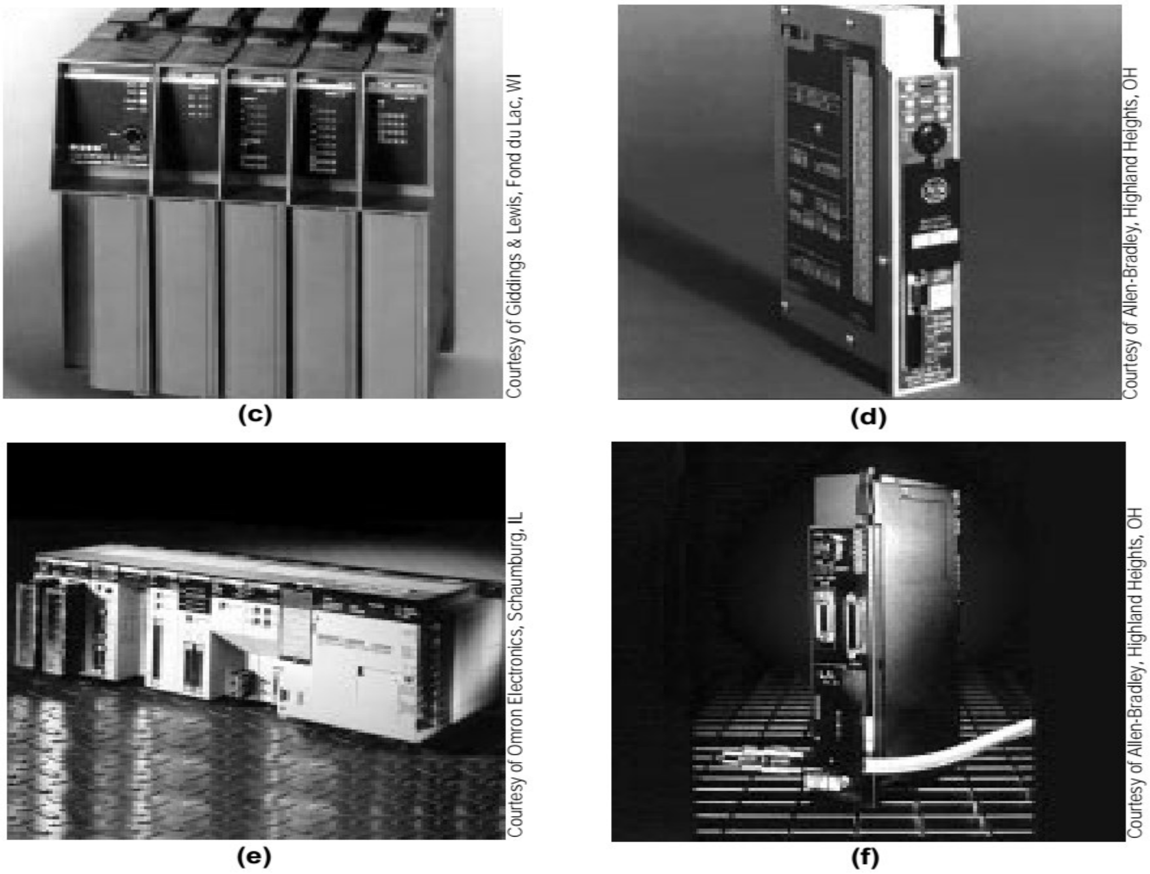


Figure I. 9: (a) Mitsubishi's smallest print size (b)PLC Direct DL105, (c)Giddings &Lewis PIC90, (d)Allen-Bradley's PLC 5/15, (e)Omron's C200H PLC, and (f)Allen-Bradley's PLC 5/80.

I.6PLC Programming Languages

The term PLC programming language refers to the method by which the user communicates information to the PLC. The standard IEC 61131was established to standard dize the multiple languages associated with PLC programming by defining the following five standard languages[5] as showing in Figure 1.11.

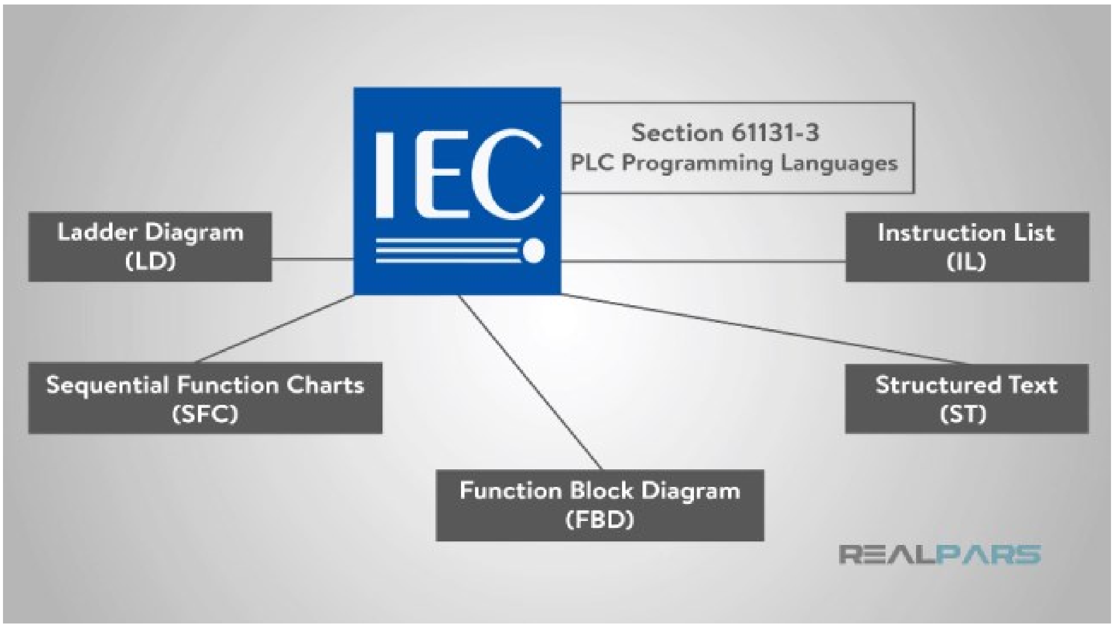


Figure I. 5: Most popular language of programming[6]

I.6.1 Ladder Logic PLC Programming

Before Programmable Logic Controllers became popular, relay-based controls were the norm at most manufacturing sites. Relays drove loads based on the simple logic that was implemented through the physical wiring of the devices. The wiring of these devices was specified in electrical drawings that assumed the layout resembling a ladder. As the most basic PLCs were introduced into the field, ladder logic PLC programming was designed to mimic the layout of relay-based circuits. In other words, ladder logic was one of the first PLC programming languages that's still used today due to simplicity.

Since its inception, ladder logic has evolved significantly. However, the basic principles of operation remain the same. Ladder logic PLC programming evaluates each rung of a ladder in sequential order assesses conditional instructions, and if the result evaluates to "TRUE," the output instructions are executed.

Ladder Logic is the most used PLC programming language around the world. It's easy to work with and maintain for those who don't have constant exposure to PLC programming [7].

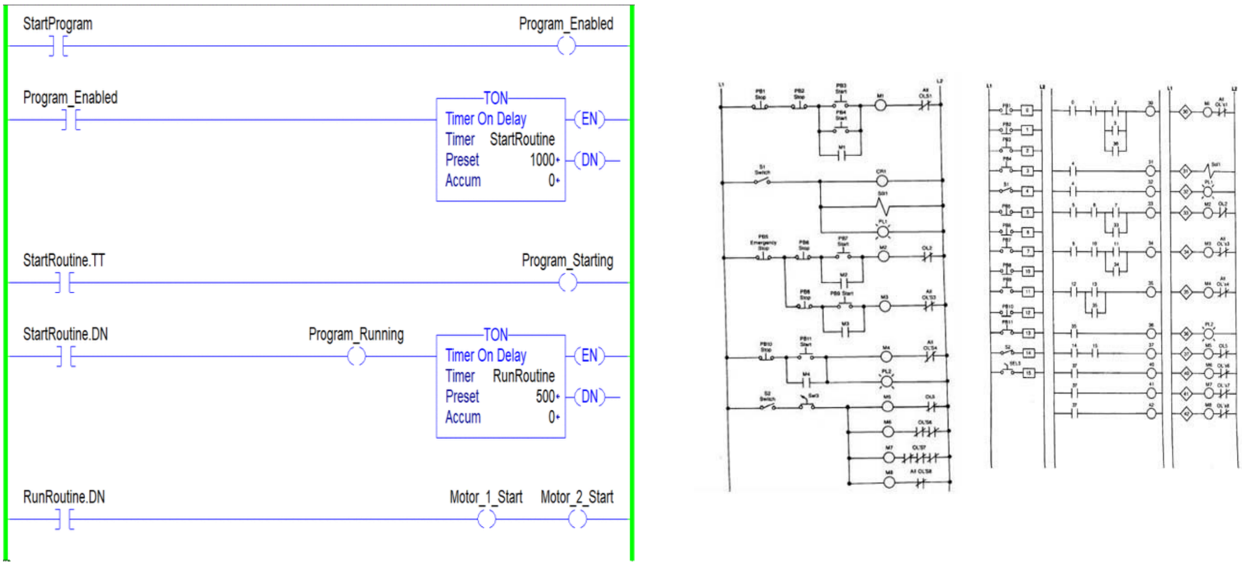


Figure I. 6: Ex for Ladder (LD)

I.6.2 Function block diagram (FBD)

Function Block Diagram programming is a language in which elements appear as blocks that are connected together resembling a circuit diagram. Function block diagrams show the relationship between the principal parts of a total system and are well-suited for process or drives control. Function (Instruction) Block is a graphical representation of a series of executable code that contains user-defined control algorithms [8].

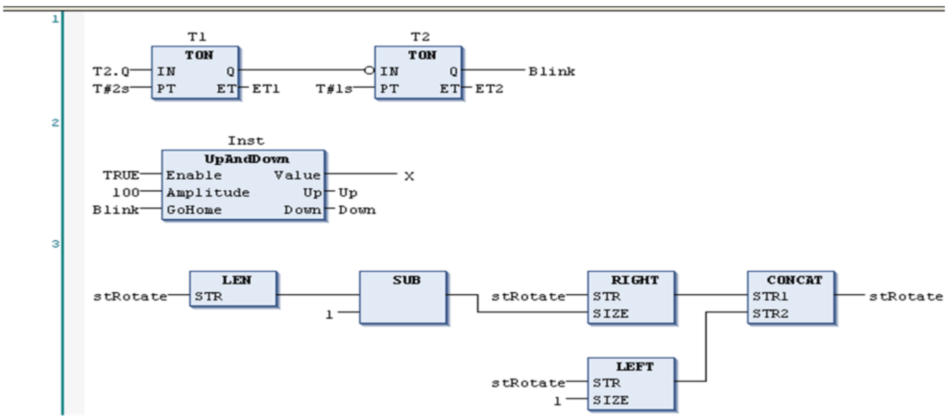


Figure I. 7: Ex for FBD

I.6.3 Sequential Function Charts (SFC)

The SFC was developed on the basis of GRAFCET, developed by Telemecanique. Similar to it is the GRAFTEC language developed by SAIA AG. GRAFTEC is a simplified version of the SFC language and can be easily translated to the other languages of the standard. Different versions of the SFC language are related to the PLC programming environment firmware. An exception is, inter alia, a software package IsaGRAF used by different PLC manufacturers.

The importance of SFC language is emphasized by other applications of the language. the synthesis of the sequential control program for FPGA based devices with use of SFC language is presented [9].

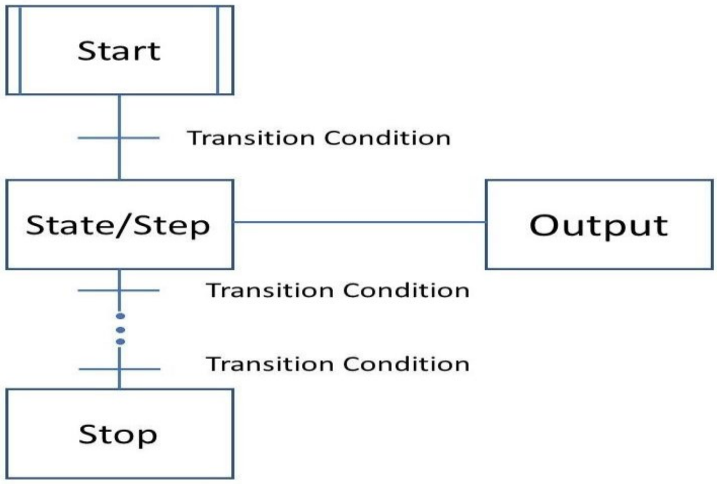


Figure I. 8:SFC general diagram

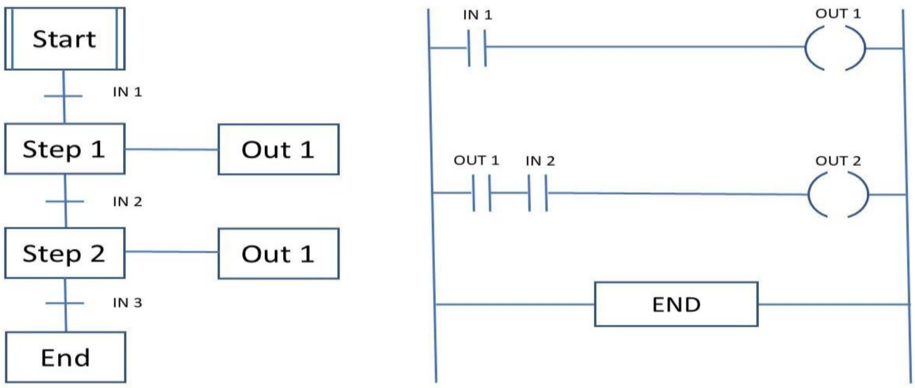


Figure I. 9:SFC structure

I.6.4 Structured Text(ST)

An ST "program" is a group of template definitions and a controller written in Java, C++, or Python that injects un typed data attributes as template argument sand instantiates template instances. The controller ultimately issues a render-to-text command to the top-level template instance to obtain the program output. An attribute is either a controller program object such as a string or VarSymbol object, a template instance, or sequence of attributes including other sequences. To enforce model-view separation, templates may not test nor compute with attribute values and, consequently, attributes have no need for type information. Templates may, however, know the data is structured in a particular manner such as atreestructure.[10]

```
PROGRAM_CYCLIC
	IF pbStart THEN
		MotorON := TRUE;
	ELSIF pbStop THEN
		MotorON := FALSE;
	END_IF
END_PROGRAM
```

Figure I. 10: Ex for ST language

Virtually any automation control scenario can be done with ST. State machines can be used for sequencing, FOR loops can be used for processing data, and WHILE loops can be used for cyclic operation. PLC manufacturers are developing hardware whose native language is ST, primarily because of the software used to program the PLC. Programmers can map inputs and outputs to variables accessed within the code, and sometimes complicated code can be easily represented with ST programming. In addition, robots typically use ST, or a variation, for their programming language [11].

I.6.5 Instruction List(IL)

IL is a simple typed assembly language, frequently used whenever it is necessary to have compact, time-critical code. The IL language it self provide it little structuring possibilities, in fact, goto-like jumps are the only ones. This makes IL programs difficult to read and difficult to manually analyze. Fur- there more, there are hardly any tools available for algorithmic analyses of IL programs. The situation is even worsened by the fact that the standard itself does not provide a formal semantics.

PLCs are reactive systems interacting in a cyclic manner with their environment. In each cycle inputs (sensor values) are read, computations take place and outputs are written (to actuators). It is possible that a number of IL programs are called sequentially within one cycle [12].

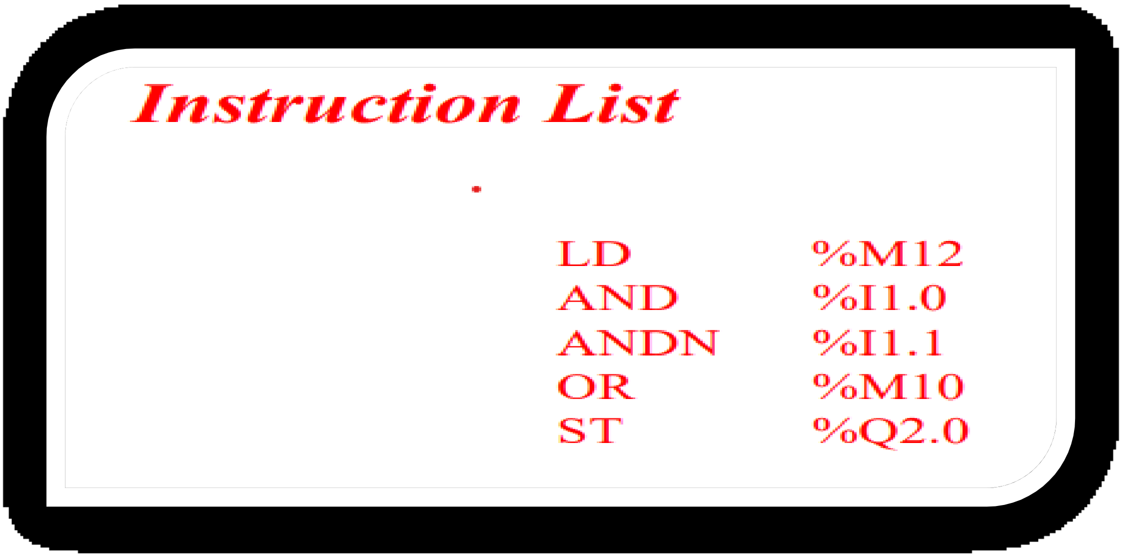


Figure I. 11:ex instruction liste.

Figure I. 12:A sample instruction list (IL) program illustrates the similarity to assembly language programming

I.7 Examples of some famousPLC companies

The following table represent some products of the famous PLC companies in the world with their Features, software used and price.

Table I. 2: The famous PLC in the world

Companies	PLC	Price (DA)	Features	Software
SIEMENS	S7-200 EM222	33,866.41	24V DC// int8/out4	Step 7, wincc, somatic manager, Tia portal, llot cloud.
	S7-300	123,180.48	24v DC// int 8/ out 8	
	S7-1200	138,952.53	120/240V AC//int 14/out 10	
OMRON	CP1E-NA20DT1-D	53,518.79	24vdc /int 12/out 8(transistor)	Cx programmer, Cx-one, symatic studio.
	CP1E-E20SDR-A	13,202.27	Ac 100-240v /int 12/out 8	
	CP1E-N40DR-A	23,153.26	100 to 240 vac/ int24/out24(relay) /	
MITSUBISHI	Fx1s-20MT-001	12,067.91	Ac 100~220v //int 12/out8	Gx developer, Gx works 2.
	Fx3SA-10MR	9,656.65	100 to 240v Ac// int6/out4	
	Fx3U-64MR	28,919.05	100~240VAC//int 32/out 32	
Schneider Electric	TM258LD42DT	63,989.86	24DCV// 42 I/O /	Zelio soft, Pro worx 32, SOmachineb asic&HMI: vijeo designer
	TM241CE40R	94,515.93	100 to 240v // 40 I/O	
	TM258LF66DT4L	97,439.10	24 Dc v// 66 I/O	

Within a few years, the PLC started to spread all over the automotive industry, replacing relay logic machines as an easier and cheaper solution, and becoming a standard for industrial automation. There is a strict relation between automation and development. In less developed countries, the greatest barriers are knowledge and cost. Industrial controllers are still very expensive. Companies don't provide detailed information about how these controllers work internally as they are all closed source as showing in the table I.2.

The OpenPLC was created to break these two barriers, as it is fully open source and open hardware. It means that anyone can have access to all project files and information for free. This kind of project helps spread technology and knowledge to places that need the most. Also, the OpenPLC is made with inexpensive components to lower its costs, opening doors to automation where it wasn't ever possible before[13].

I.8 Conclusion

This chapter is Overview of the programmable logic controller (PLC) and the associated to architecture and the multiple programming languages . We have divided this chapter into three mainly parts: the first concerns the constitution of the PLC , the second part introduces the Programming Languages (LD ,FBD, ...) and the third part deals the types and the famous companies that making the PLC.

The high prices of the PLC and the fact that they are closed systems are among the disadvantages and major obstacles to the development of the industry, especially in weak countries.

For that ,The OpenPLC was created in order to solve these problems, as it is fully open source and open hardware. It means that anyone can have access to all project files and information for free. This is what we will show in the second chapter**.**

Chapter II

Implementation of OPENPLC Arduino

II.1 Introduction

Companies are always looking for ways to increase production. The elevated consumerism pushes factories to produce more in less time. Industry automation came as the solution to increase quality, production and decrease costs. Since the early 70s, PLC (Programmable Logic Controller) has dominated industrial automation by replacing the relay logic circuits. However, due to its high costs, there are many places in the world where automation is still inaccessible[13].This chapter describes the creation of a low-cost open source PLC based on Arduino(UNO)card, comparable to those already used in industry automation, with a modular and simplified architecture and expansion capabilities.

II.2 Open PLC Arduino Architecture

In this section, we will implement the OpenPLC prototype, which has four parts will be built:

- Power Supply
- ArduinoUNO
- Inputs
- Outputs

II.2.1 power supply

In this project , an external alimentation is used (5V/ 24V)

II.2.2 Microcontroller (Arduino).

The Arduino Uno is a programmable microcontroller that allows, as its name suggests, to control mechanical elements: systems, lights, motors, etc. This electronic card therefore allows its user to easily program things and create automated mechanisms, without having any particular programming knowledge. It is a tool designed and intended for inventors, artists or amateurs who wish to create their own automatic system by coding it .Table II.1 represent the features of arduino UNO , which we used as a CPU in our open PLC.

Table II. 1: Structure of arduino

Microcontroller	ATmega 328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for	3.3V Pin 50 Ma
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

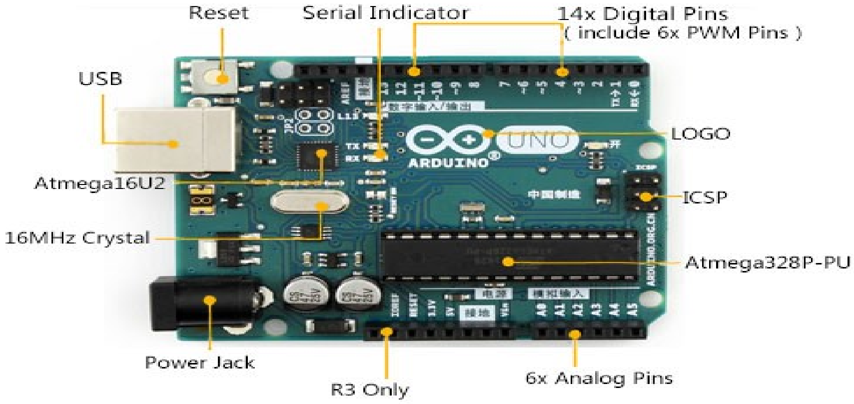


Figure II. 1: Arduino uno

II.2.3 Input

the open PLC that was proposed consist of four digital inputs. The input channels provide isolation and signal conditioning functions so that sensors can be often directly connected to them without other circuitry. Electrical isolation from the external world is usually by means of optoisolators ,in our project, the optocoupler (4N35) is used

Figure **II.3** shows the isolated input of OPENPLC arduino. When a digital pulse passes through the light-emitting diode, a pulse of infrared radiation is produced. This pulse is detected by the phototransistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the phototransistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit (Figure **II.2)**.

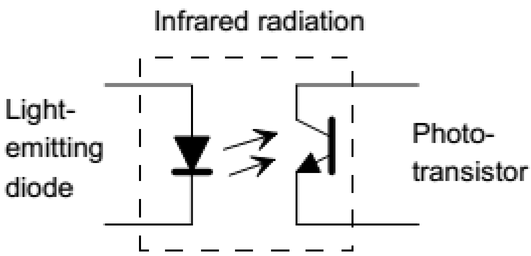


Figure II. 2: Optocoupler

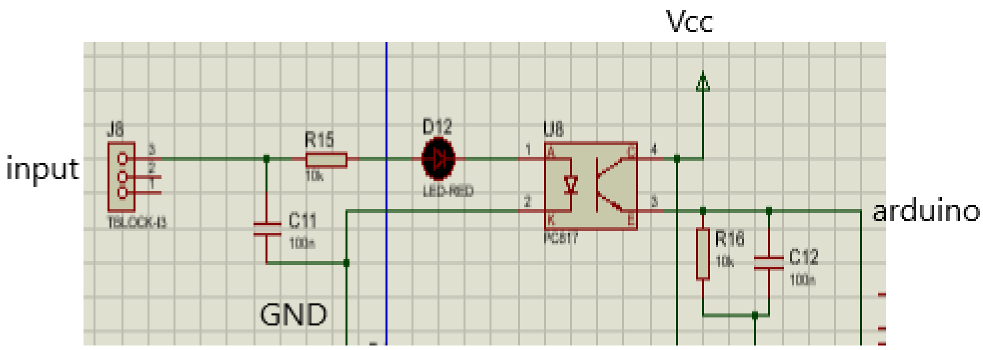


Figure II. 3: Isolated input of OPENPLC arduino

II.2.4 Output

the open PLC arduino proposed has four outputs with the relay type, which the signal from the PLC output is used to operate a relay and is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for a.c. and d.c. switching.

They can withstand high surge currents and voltage transients. The circuit of outputs has a double isolated ,as they are isolated by an optocoupler (like the input circuit) and the relay ,which give an additional layer of isolation, and other component to protect the optocoupler and the relay : resistors, led, capacitors, diode and transistor as shown in figure II.3

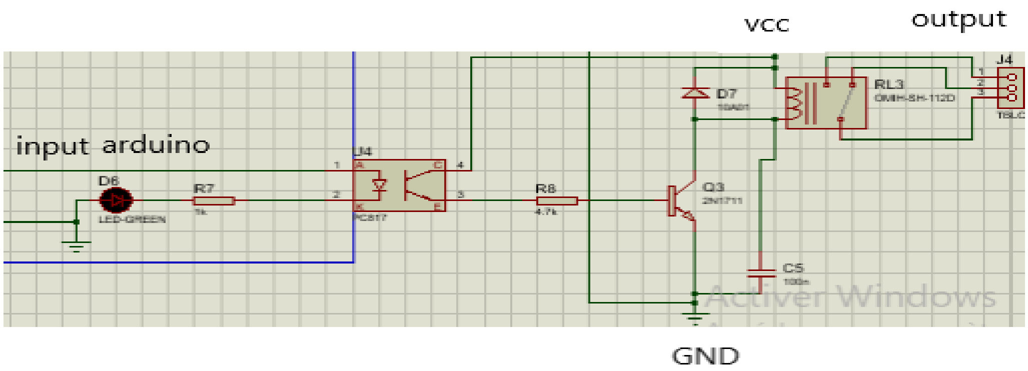


Figure II. 4: Isolated output of OPENPLC Arduino

II.3 Environment programming (IDE)

it is an official software introduced by Arduino.cc, that is mainly used for editing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go. In this article, we will introduce the Software, how we can install it, and make it ready for developing applications using Arduino modules.

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module [14].

In our project we use the IDE for programming the open PLC arduino .

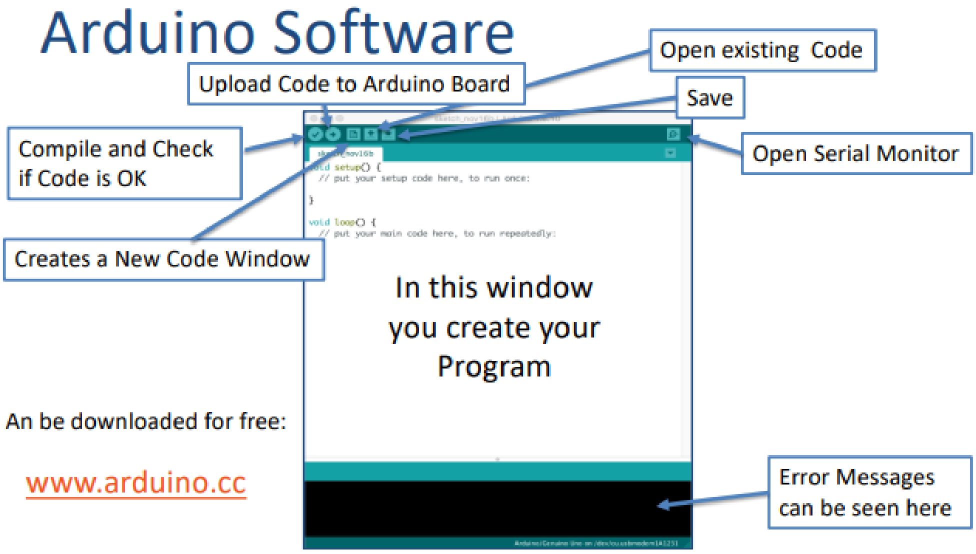


Figure II. 5: Arduino software

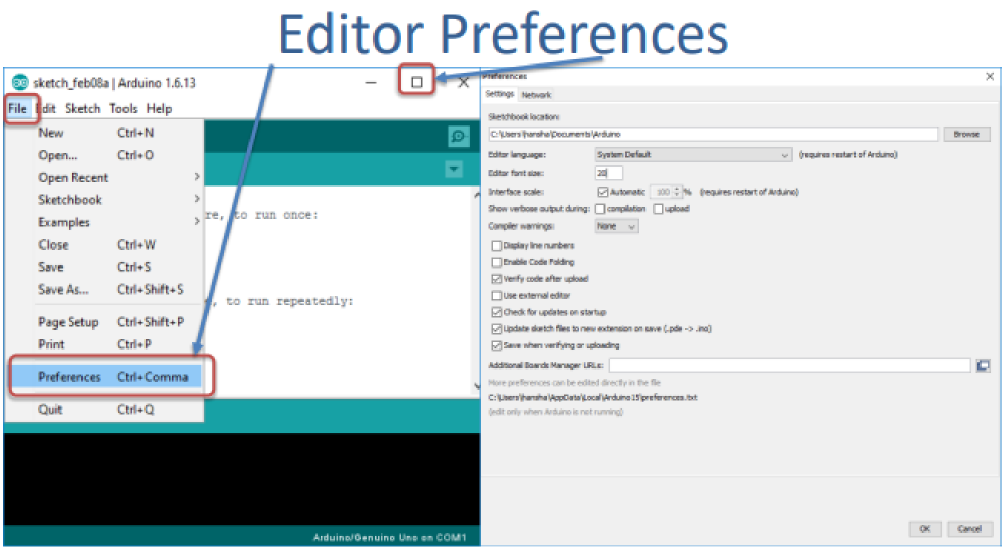


Figure II. 6: Editor preferences

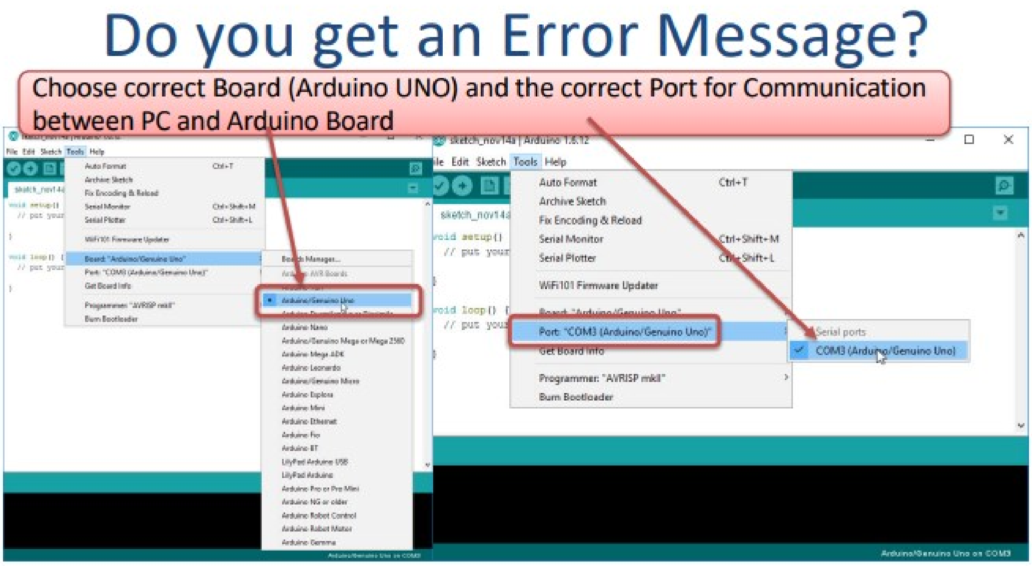


Figure II. 7: Error message

II.4 Realisation of OPENPLC arduino card

On realized the OPENPLC arduino card in three steps : designing, printing and installing.

II.4.1 The design

In order to design the OPENPLC arduino card, we will use the Proteus software(8,13 version) , which was considered from one of the best programs in designing electrical circuits, simulating microprocessors and designing PCB boards.

The figures **II.7** , **II.8 represent the** schemics and designing of PCBs that was implemented in proteus

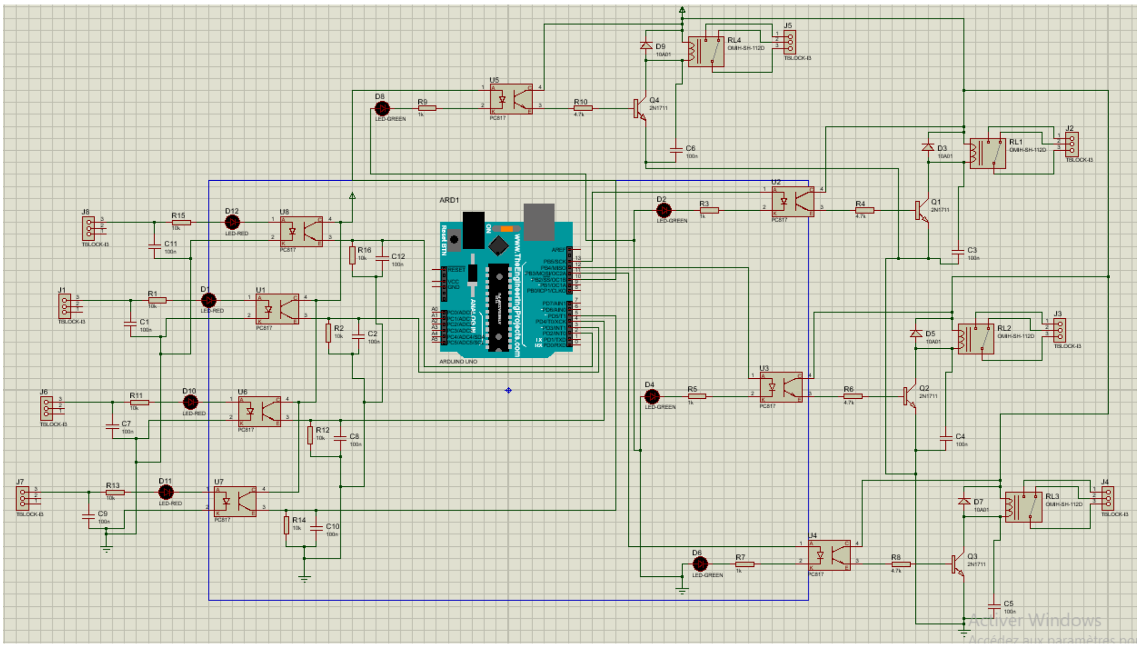


Figure II. 8: Electrical circuit of the OPENPLC Arduino

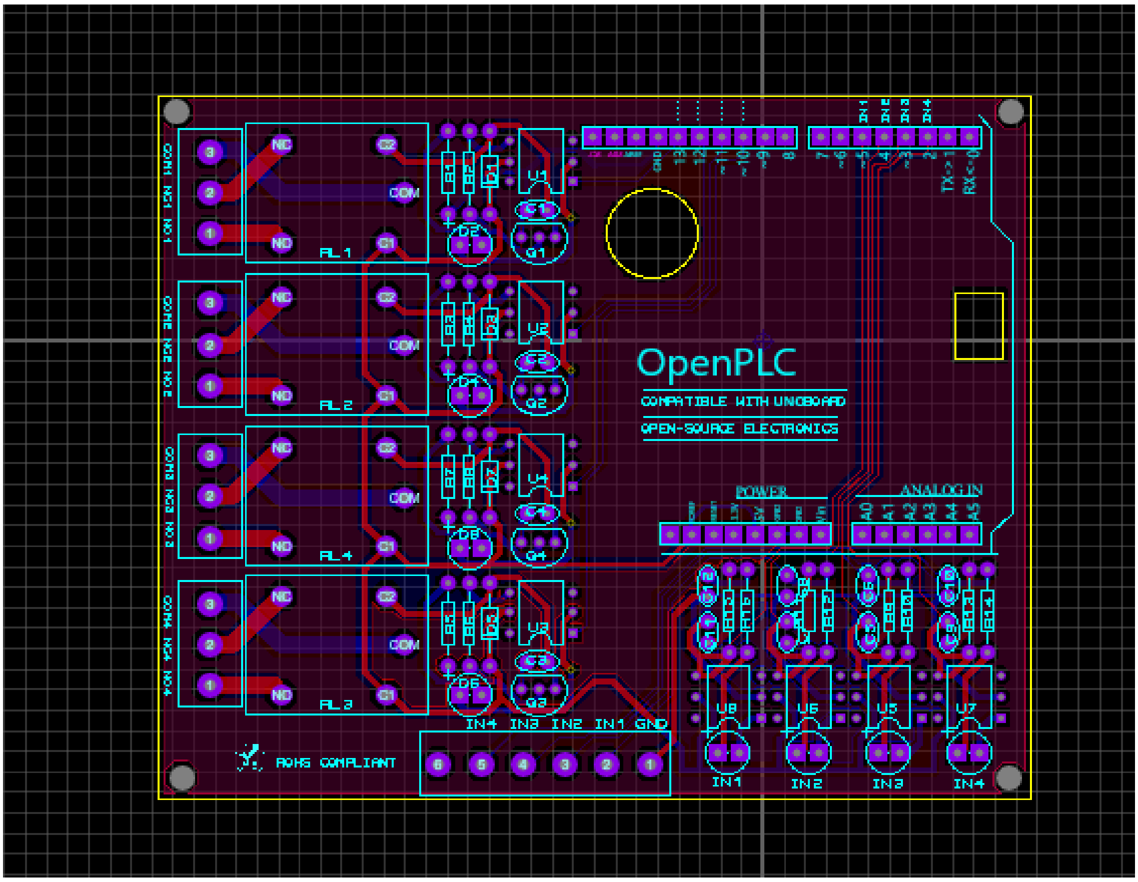


Figure II. 9:PCB table of OPENPLC Arduino

II.4.2 Printing

After we printed our design, we got the card shown in the figure II.9

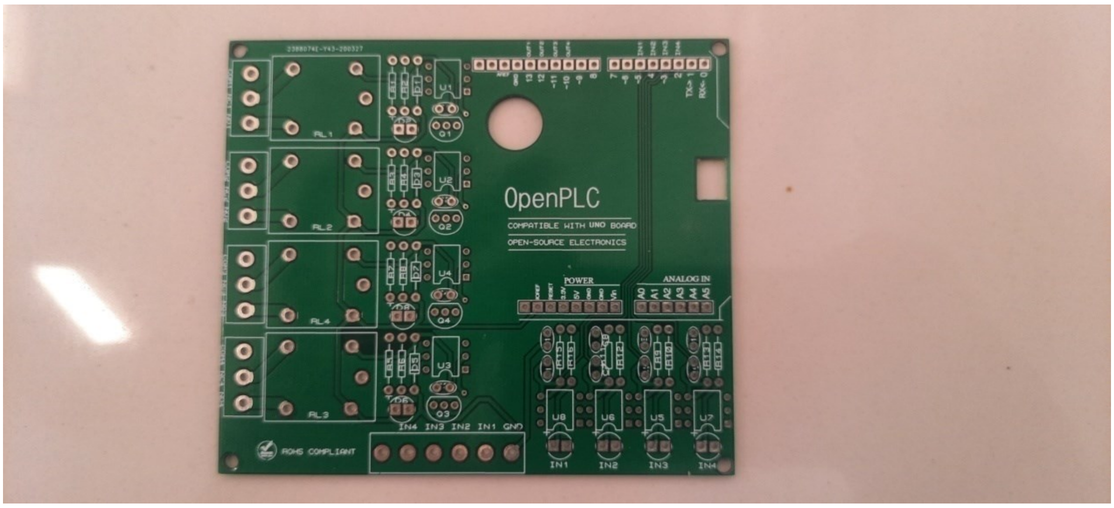


Figure II. 10:OPENPLC Arduino card without component

II.4.3 Installing

Table II. 2: Component of our OPENPLC Arduino card

Component	quantity	Price DA
Resistor	8 (10 k ohm)	5 DA
	4 (4.7 k ohm)	5 DA
	4 (1 k ohm)	5 DA
Led	4 (red led)	10 DA
	4 (green led)	10 DA
Optocouplor	8 (4N35)	80 DA
Capacitor	12 (100 Nf)	30 DA
Transistor	4(BC548)	80 DA
Diode	4(1N4007)	5 DA
Relay	4(SRD-05VDC-SL-C)	200 DA
Arduino uno	1(Uno R3 ATmega328P)	2200 DA
PCB card	1 (figure II.9)	1000 DA

- Total price : 5500 DA

When installing the components, we get the following card:

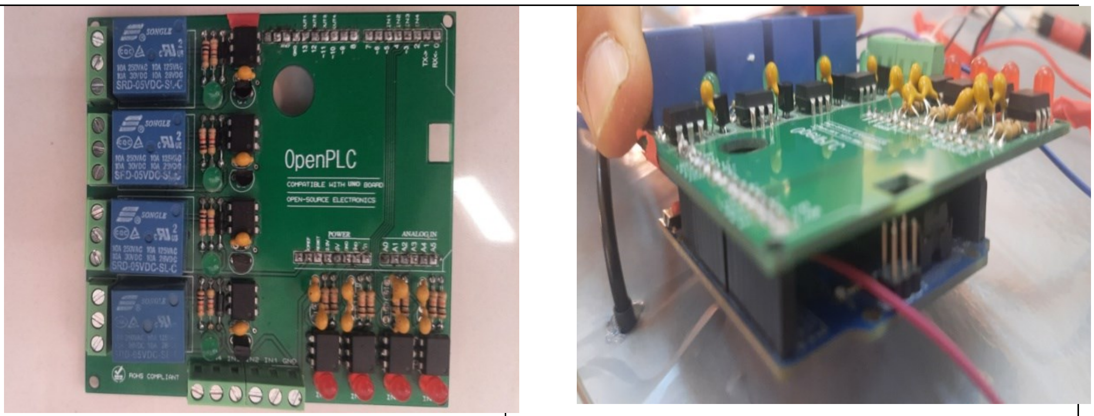


Figure II. 2: The OPENPLC Arduino card

II.4 Conclusion

In this chapter , we have made a OPENPLC from Arduino uno and some components table (II.2) are cheap compared to plc.

We designed this card with Proteus software (electrical circuit , PCB board).

In the next chapter we will try to control a cold room by this card (OPENPLC arduino).

Chapter III

Application the OpenPLC Arduino to control the cold room system

III.1 Introduction

In this chapter the OpenPLC Arduino board assembled in the previous chapter is used to control a cold room. An overview of cold room will be presented prior to making a demonstrative model by showcasing the required components and steps of combination.

III.2 Generals on cold rooms

III.2.1 Definition

A cold room is a type of refrigeration chamber or insulated space designed to maintain an artificially generated temperature or range of temperatures. Cold rooms are used for storing temperature-sensitive, perishable items, such as food items and pharmaceutical products like vaccines. Cold rooms can vary in size from very small walk-in rooms to very large warehouse storage[15].

Figure III. 1: Cold room

Figure III. 2:cold room from inside

Cold rooms provide precise temperature control for commercial facilities where consistent, powerful refrigeration or freezing is required. For food or chemical storage, this means long-term temperature regulation for perishable or unstable products, lowered deterioration rates, and peace of mind knowing that items are preserved in the optimal conditions. For pharmaceuticals, the FDA indicates current best practices for the warehousing of drugs and requires them to be stored "under appropriate conditions of temperature, humidity, and light so that the identity, strength, quality, and purity of the drug products are not affected." Cold rooms represent one of the above mentioned requirements for cold storage of these products[15].

Some other benefits of Parameter's cold rooms include[15]:

- **Energy Efficiency** – Reduce energy costs for storage of samples and other materials. Cold storage units are constructed to prevent fluctuations in temperature, meaning less energy is required to balance and adjust the temperature while in use. This holds true no matter your industry.
- **High Product Standards** – Guarantee potency of vaccines, drugs, and other sensitive items with cold storage. This can also help improve your company's reputation and reliability.
- **Safety and Security** – Medications and medical equipment are highly sensitive and expensive — ensure their safekeeping with locked, weather-proof cold storage units.

Cold storage is also used in transportation of industrial materials and other temperature-sensitive products across industries. In this case, the container has a cooling unit installed to ensure optimal temperature control. Cold rooms, in general, are valued for their precision and power and can maintain specific temperatures over long periods of time.

Figure III. 3: cold room storage

III.2.2 The method of work

The design behind cold rooms that allows them to function efficiently isn't so different from any other refrigeration system. Cold rooms use a compressor, condenser, fans, and an evaporator to maintain temperature within the unit. After a gas refrigerant gets compressed in the compressor it expands, and the gas absorbs energy. The hot gas flowing from the compressor passes over the evaporator coils and, after liquefying under high pressure, this cools the evaporator coils and the surrounding air. In order to maintain temperature, cold rooms must also be efficiently insulated[15].

Figure III. 4: Installing of cold room

III.2.3 The temperature range of a cold room

Most cold rooms will be able to accommodate temperatures between -2°C and 10°C. As for the specific temperature range of a cold room, that depends on what's being stored inside. For food items such as fresh produce, an ideal cold room temperature ranges between 2 and 8 degrees Celsius. Pharmaceutical ICH storage will require a 5°C with a tolerance of ±3°C. When colder conditions are required, freezer rooms will be the preferred option, able to tune down to -30°C for storage of pharmaceutical products or chemicals. Freezers are also able to provide ultra-low storage needs at -50°C to -80°C[15].

III.2.4 Uses of cold room

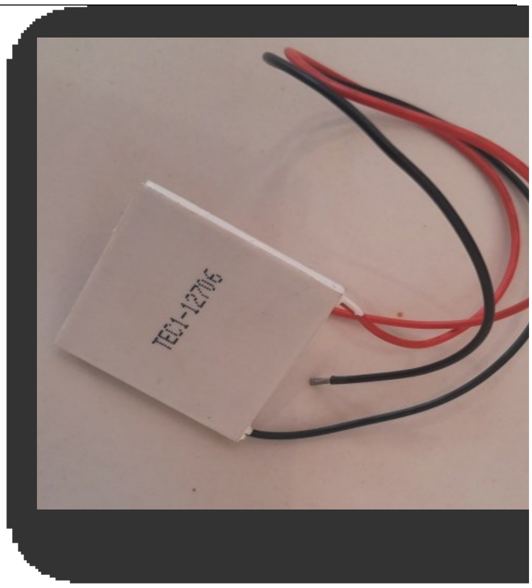
As for their application, cold rooms can be used to store a variety of items across industries. The most common items include biopharmaceutical and pharmaceutical products, textiles, tobacco products, perishable foods, flowers, delicate plants, artworks, and even rare books.

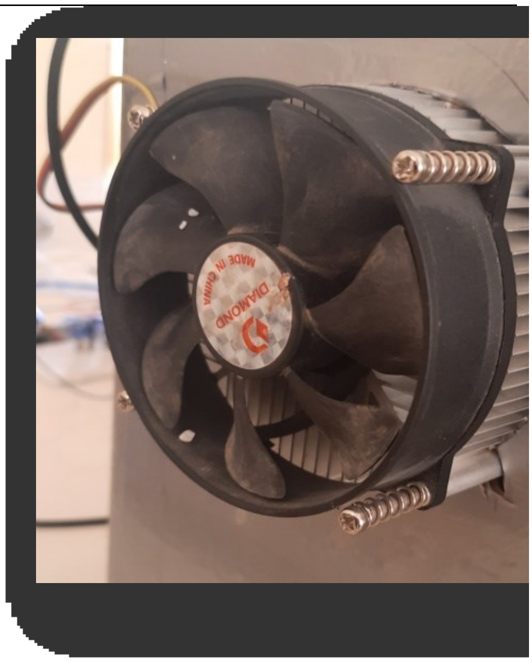
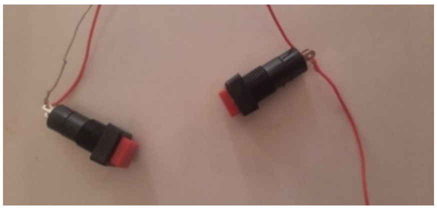
Precise temperature control of certain drugs and medical devices is especially important to prevent material corruption and guarantee usage safety. Laboratory reagents, dialyzers, disinfectant solutions, sterility, burn treatment products, and dental restorative materials, for example, are all highly temperature sensitive and require routine or consistent refrigeration to be effective. Cold rooms also can be utilized to process or slow down chemical reactions in engineering settings. And as mentioned above, if sensitive materials need to be temperature-regulated during transit, cold rooms can help there too [15].

III.3 controlling the cold room system by OpenPLC Arduino

On this part ,the cold room proposed is build by the component that represented in The table III.1

Table III.1 the component of the cold room

diymore TEC1-12706 Thermoelectric Cooler Cooling Peltier	Model: TEC1-12706 Size: 40mm x 40mm x 3.6mm Operation Temperature: -30°C-70°C Refrigeration Power: Qcmax 50-60W Working Current: 4.3-4.6 A (rated 12 v); Imax: 6A Price: 980 DA	
Temperature Sensor DS18B20	3.0-5.5V input voltage Waterproof -55°C to+125°C temperature range ±0.5°C accuracy from -10°C to +85°C 1 Wire interface Price: 1050 DA	

CPU fan LGA 775	○ Noise level: 20 dB(A) ○ Voltage : 12V ○ Price : 900 DA	
Push button	Price :100 DA	
Brushless Super Quiet Cooling Fan	Brand:HXS® Model#:HXS-4010 Bearing Type: Sleeve Bearing Dimension (mm): 40 x 40 x 10 Rated Voltage: DC 12V Current: 0.16A Speed (RPM): 3,900 (±5%) Airflow (CFM): 5 Noise (dBA): 18~20 External Material: Plastic	

	Connector: 2-pin Cable Length: 130mm (5"inches) Weight: 13g Color: Black Package Contents: - 1x 40mm Price: 350 DA	
Heat Sync north GB	Material : almunium We get it from computer	

III.3.2 Grafcet

In the initial case, the red light is glowing, and when pressing Push button P1 the first stage is achieved, in which the green light glows and the red light turns off, and here the sensor starts reading the temperature value. If the temperature is greater than 25 degrees, the second stage is achieved where the green light remains glowing in addition to that the fan starts working, as well as the radiator and cooling fan. But if it is less than 20 degrees, it goes back to state 1 or it goes back to the starting point and it stops everything working with the red light glowing by pressing the Push button stop.

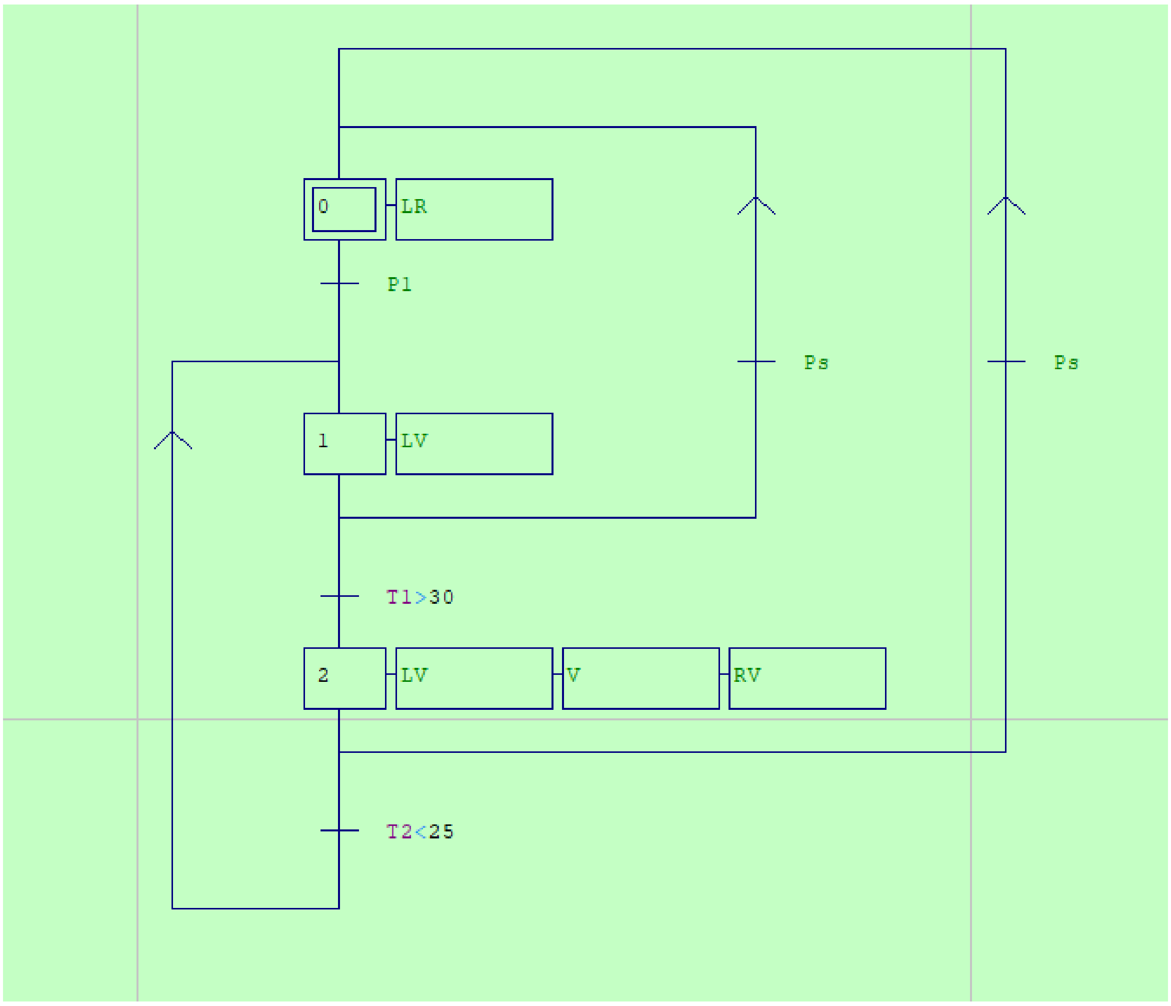


Figure III. 5: Grafcet of the system

III.3.3 Equations:

Set :

$$Sx0 = x1 * Ps + x2 * Ps + (\overline{x0} * \overline{x1} * \overline{x2} * \overline{x3})$$

$$Sx1 = x0 \ * P1 + x2 * T2$$

$$Sx2 = x1 * T1$$

Reset:

$$Rx0 = x0 * P1$$

$$Rx1 = \ x1 * T1 + (x2 * Ps)$$

$$Rx2 = x2 * T2 + (x2 * Ps)$$

States equations:

$$x0 = sx0 + x0 * \overline{Rx0}$$

$$x1 = sx1 + x1 * \overline{Rx1}$$

$$x2 = sx2 + x2 * \overline{Rx2}$$

Outputs:

$$LR = x0$$

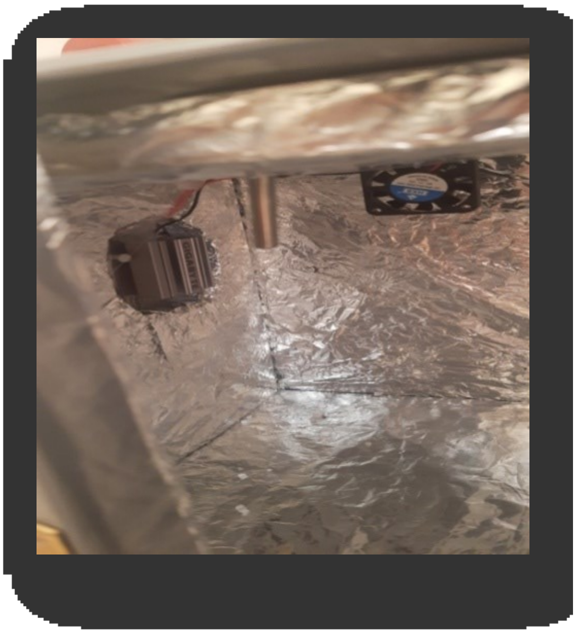
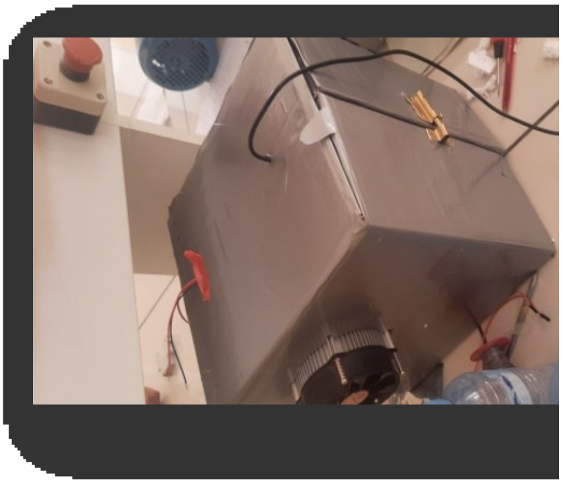
$$LV = x1 + x2$$

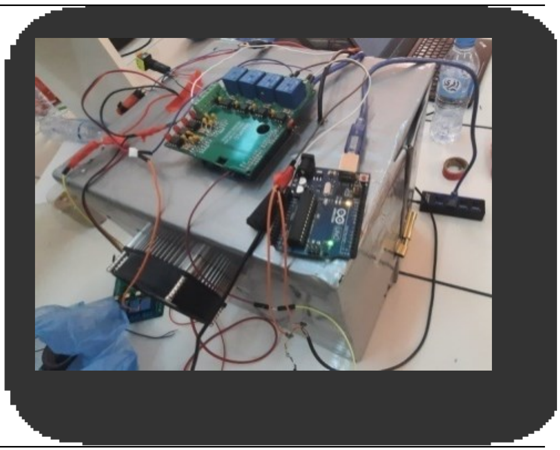
$$V = x2$$

$$RV = x2$$

III.3.3 Building the cold room

We bring polyester and then cover it with aluminum	
The aluminum-covered polyester is installed on the carton walls until it resembles a model of a cold room.	

We install the components mentioned previously in the table (III.1) in our model of the cold room	  

Setting up a cold-room-control program according to the herein equations in IDE, programmed using the arduino c language. Ordering inputs and outputs according to the program.	

III.3.4 System result

After combining the cold room mode, it was connected to the OPENPLC cardboard. The system was launched through the Arduino C software on installed on a computer. The figure III.6 demonstrates the system function where it decreased the room temperature from 22 degrees Celsius to 20 degrees Celsius within a short period of time. The system stops when the temperature is at that degree, then it is reactivated when it reaches 25 degrees Celsius (as programmed in the graphs set).

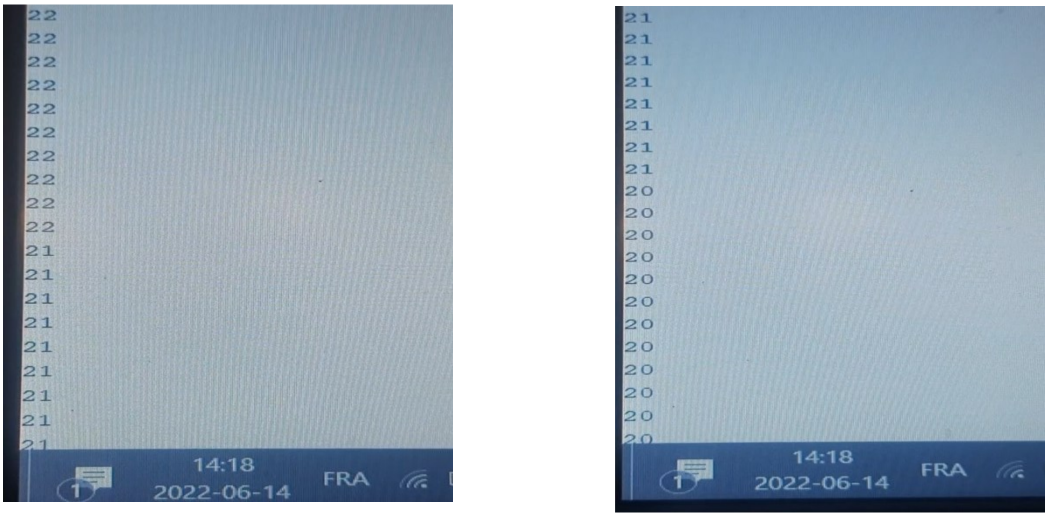


Figure III.6 drop in temperature inside the cold room

III.5 Conclusion

In this chapter, we applied the OPENPLC arduino card to control a prototype of a cold room that we have made , where we did a Grafcet and its equations by using Arduino C language.

In the end, we could say that the OPENPLC card worked as perfect and effectively as the PLC devices**.**

General conclusion

General conclusion

Nowadays, industrialization represents an indicator to nations' prosperity. However, controlling these smart industrial systems by means of PLC is not available to most countries due to high cost and being a closed technology. Consequently, opting for OPENPLC has become a necessity.

As such, OPENPLC system is an alternative that is characterized by ease of programming and use, coming at costs lower than the ones available at the market with same functioning reliability. It depends on Open source microcontroller like Arduino (uno, nano, mega,...etc) and Rispry-Pi.

In this work, a 4 24v inputs, 4 relay outputs, 6 analogue inputs, and five 5v programmable inputs Arduino Uno-dependent OPENPLC was constructed.

The cost of this card was much less than the smallest PLC available in the market with the addition of it being programmable by IDE language according to the graph set. This card has been used in controlling a cold room model. The OPEN PLC card controlled the suggested system effectively.

Perspective:

- Increasing the number of inputs and outputs.
- Developing software (environment programming) throw that we could program it with ladder language or other languages.

References

References

[1]W.Bolton ,“programmable logic controller forth edition”,Published by Elsevier Newnes ,2006,ISBN-13: 978-0-7506-8112-4

[2] Lashin, Maha M. "Different applications of programmable logic controller (PLC)." International Journal of Computer Science, Engineering and Information Technology(IJCSEIT) 4.1 (2014): 27-32.

[3] https://www.asasmy.com/modular-plc-vs-compact-plc-malaysia.html,13/2/ 2022 20:49

[4] L.A.Bryan, E.A.Bryan, “programmable controllers theory and implementation”,2nd edition Published by Industrial Text Company ,1997 ,ISBN 0-944107-32-X

[5] Petruzella, Frank D. Programmable logic controllers fifth edition, Published by Tata McGraw-Hill Education, 2005, ISBN 978-0-07-337384-3.

[6] https://realpars.com/plc-programming-languages/(25/05/2022) 9:30

[7] https://www.solisplc.com/tutorials/how-to-read-ladder-logic (25/05/2022) 10:08

[8]https://www.brightonk12.com/cms/lib/MI02209968/Centricity/Domain/517/Tutorial%2006%20Function%20Block%20Diagram%20Programming%20with%20PLC%20Tutorial.pdf 25/5/2022 12:04

[9] Wcislik, M., K. Suchenia, and M. Łaskawski. "Programming of sequential control systems using functional block diagram language." IFAC-PapersOnLine 48.4 (2015): 330-335.

[10] Parr, Terence. "A functional language for generating structured text." URL http://www. cs. usfca. edu/parrt/papers/ST. pdf (2006).

[11] https://control.com/technical-articles/comparing-programming-languages-structured-text-vs-ladder-logic/ 25/05/2022 11:00

[12] Huuck, Ralf. "Semantics and analysis of instruction list programs." Electronic Notes in Theoretical Computer Science 115 (2005): 3-18.

[13]Alves, Thiago Rodrigues, et al. "OpenPLC: An open source alternative to automation." IEEE Global Humanitarian Technology Conference (GHTC 2014). IEEE, 2014.

[14]https://www.researchgate.net/profile/Mohamed-Fezari
2/publication/328615543_Integrated_Development_Environment_IDE_For_Arduino/links/5b
d8c6d24585150b2b9206df/Integrated-Development-Environment-IDE-For-Arduino.pdf
06/05/2022 22:03

[15] https://humiditycontrol.com/what-is-a-cold-room/ 29/05/2022 13:55

[16]Hackworth, John R., and Frederick D. Hackworth. Programmable logic controllers programming methods and applications. New Jersey: Pearson, 2004.

Abstract

PLC systems are dominating most smart industrial systems. The weaker countries and the followers of this field often suffer from high costs and its closed technology. This work seeks to provide an effective, reasonable-priced, and easy solution to using an equivalent PLC alternative that is available at the market. The work recommends combining an OPENPLC Arduino card, based on the ATMega chip found in the Arduino board. Then, a cold room was built and was controlled by OPENPLC Arduino to account for its efficiency.

Key words : PLC ; open PLC ; controlling ;Arduino ;Cold room ; Smart industrial System .

Résumé

Les contrôleurs logiques programmables ou les Automates programmables industriels contrôlent la plupart des systèmes génie-industriels, cependant, les pays développés et les amateurs de ce domaine souffrent à cause de ses prix excessifs et de son système fermé.

L'objectif de notre travail est de présenter une solution efficace, abordable et facile à utiliser qui équivaut aux API disponibles sur le marché.

Nous avons créer une carte API ouvert sourceà partir d'une puce AT Mega qui existe dans l'Arduino.

Nous avons ensuite construit une chambre froide et nous l'avons contrôlé avec la carte API ouvert source à fin de confirmer son efficacité.

Mots clés :automate programmable industriel ; API ouvert source ; contrôler ; Arduino ;chambre froid ; system industriel intelligent

الملخص

تعتبر أنظمة التحكم المنطقي القابل للبرمجة هي المتحكمة في أغلب الأنظمة الصناعية الذكية. تعاني الدول الضعيفة و هواة مجال التحكم من غلاء أسعارها و تقنيتها المغلقة. الهدف من عملنا هو تقديم حل غير مكلف و سهل يكافئ المتحكمات المنطقية الموجودة في السوق.

قمنا بصناعة بطاقة متحكم منطقي مفتوحة المصدر وقابلة للبرمجة انطلاقا من شريحة الأردوينو اينوا. ثم قمنا بإنشاء غرفة تبريد و التحكم فيها بواسطتها للتأكد من فعاليته.

الكلمات المفتاحية :المتحكم المنطقي القابل للبرمجة، بطاقة متحكم منطقي مفتوحة المصدر، التحكم، اردوينو، غرفة تبريد، الأنظمة الصناعية الذكية.