

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 study thesis
Presented for obtaining the
ACADEMIC MASTER

Field: Sciences and Technologies
Specialty: ELECTRICAL CONTROL

Thesis

Study and preparation of PLC Arduino

Board of Examiners

President: LAOUAMER Mosbah

Examiner: MAHMOUDI Abdelkader

Examiner: LABIOD Chouaib

Supervisor: CHEMSA Ali

Co-supervisor: CHERIF Hakima

Presented by

- GHARIB Abdelhamid

- BACI Nadhir

- DIFALLAH Mohammed Bachir

-SAHRAOUI Mohammed

2021-2022

ملخص

في هذه المذكرة، قمنا بانجاز متحكم منطقي مبرمج مفتوح المصدر أطلقنا عليه اسم open plc arduino و هو يتكون من أربعة مداخل منطقية و ستة مداخل تناظرية و أربعة مخارج عن طريق مرحلات ثم قمنا بتطبيقها في التحكم في نظام الري بالررش المحوري, واشتغل النظام بكفاءة وفعالية جيدة و منه كانت هذي اللوحة open plc arduino قد حققت النتائج جيدة و بتكلفة اقل مقارنة مع المتحكمات المنطقية المبرمجة المتوفرة في السوق .

Abstract

In this thesis, we have implemented an open source programmable logic controller which we called open plc arduino, and it consists of four logic inputs, six analog inputs and four outputs via relays, and then we applied them to control the system of pivot sprinkler irrigation, and the system worked effectively and efficiently. This open plc arduino board has achieved good results and lower cost compared to PLCs available in the market.

Résumé

Dans cette mémoire, nous avons implémenté un automate programmable open source que nous avons appelé open plc arduino, et il se compose de quatre entrées logiques, six entrées analogiques et quatre sorties via des relais, puis nous les avons appliqués pour contrôler le système d'irrigation par aspersion à pivot, et le système a fonctionné de manière efficace et efficiente. Cette carte open plc arduino a obtenu de bons résultats et à un coût inférieur par rapport aux automates disponibles sur le marché.

Contents

List of figures	I
List of tables	II
List of abbreviations	III
Introduction general	1
Chapter I :Overview about programmable logic controllers	
I.1 Introduction.....	2
I.2 definition	4
I.3 Principle of operation.....	5
I.4 Architecture.....	8
I.4.1.Input and Output (I/O) Modules	9
I.4.2 Power Supply	9
I.4.3Central Processing Unit (CPU).....	9
I.4.4 Memory System	9
I.4.5 Programming Device	11
I.5 GRAFCET	11
I.6 Programming languages.....	11
I.6.1 Ladder diagram (LD)	12
I.6.2 Instruction List	13
I.6.3 Structured Text (ST)	13
I.6.4 Function Block Diagram (FBD)	13
I.6.5 Sequential Function Charts (SFC)	13
I.7 Input/output unit.....	14
I.8 Types of plc.....	15
I.8.1 Modular PLC	15
I.8.2 Compact PLC	15
I.8.3 Difference between modular and compact PLC	15
I.8.4 Sizes of PLC	16
I.9 The most famous companies of programmable logic controllers in the world.....	16

I.10 Example of some PLC companies.....	17
I.11 Conclusion	17
Chapter II: Realisation of OPENPLC Arduino	
II. Introduction.....	19
II.2 Generality about Open PLC Arduino.....	19
II.3 Architecture of Open PLC Arduino	19
II.3.1 Power supply.....	20
II.3.2 Arduino Uno	20
II.3.3 Input.....	20
II.3.4 Output.....	21
II.3.5 Protections	22
II.4 Open PLC Arduino characteristics.....	22
II.5 Implementation the Open PLC Arduino board.....	23
Drawing the circuit and printing it on PCB:.....	23
Installing the electronic parts and welding them on the PCB	25
II.6 language and environment programming	26
II.7 Conclusion	27
Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system	
III.1 Introduction	29
III.2 General Information about agricultural irrigation systems	29
III.2.1 Definition.....	29
III.2.2 Agricultural irrigation techniques	29
III.3 Controlling the Sprinkler irrigation system by OpenPLC Arduino.....	31
III.3.1 Grafcet	31
III.4 Used equipment	33
III.5 Conclusion.....	35
Conclusion general.....	37

List of figures

Figure 1.1 PLC conceptual application diagram	4
Figure 1.2 Programmable controller block diagram.....	5
Figure 1.3 Block diagram of major CPU components.....	5
Figure 1.4 Illustration of a scan.	6
Figure 1.5 Input/output interfaces.....	7
Figure 1.6 (a) Personal computer used as a programming device and (b) A mini programmer unit.....	7
Figure 1.7: PLC hardware depiction	8
Figure.1.8 Most popular language of programming.....	12
Figure 1.9 Ladder diagram (LD)	12
Figure 1.10 Optocoupler	14
Figure 1.11 PLC product ranges.....	16
Figure2.1 Arduino Uno Card	20
Figure2.2 Isolated digital input circuit	21
Figure.2.3Isolated output circuit	22
Figure.2.4 The schemics of Open PLC Arduino.....	24
Figure.2.5 The PCB designing of Open PLC Arduino.....	24
Figure.2.6 Open PLC Arduino board before and after installation the components.....	25
Figure.2.7 Arduino IDE description.....	27
Figure 3.1Pivot irrigation.....	30
Figure 3.2Grafcet Sprinkler Irrigation System	32

List of tables

Table 1.1 Difference between modular and compact PLC	15
Table 1.2 The most famous companies of programmable logic controllers in the world	16
Table 1.3 Example of some PLC companies	17
Table 2.1 Open PLC Arduino features	23
Table 2.2 The components used in implementation the Open PLC Arduino	25
Table 3.1 Used Equipment.....	33

List of abbreviations

ADC: Analog to Digital Converter
BM: Button Start
BS: Button Stop
CADCAM: computer-aided design / computer-aided manufacturing
CPU: Central Processing Unit
FBD: Function Block Diagram
GND: Ground
H: Moisture sensor
I/O: Input and Output
IDE: Integrated Development Environment
IL: Instruction List
LD: Ladder Diagram
LR: Red led indicator light
LV: Green led indicator light
M: Water pump
MP: Motor to rotate the sprinkler shaft
ODB: Open data base
PCB: Printed Circuit Board
PLC: Programmable Logic Controller
PWM: Pulse Width Modulation
RAM: Random Access Memory
RLL: Relay Ladder Logic
ROM: Read Only Memory
SFC: Sequential Function Charts
ST: Structured Text
USA: United States of America
USB: Universal Serial Bus

Introduction

Introduction

In the industrial field where competition is fierce, companies are faced with a requirement to improve the quality and quantity of their products by minimizing prices and expenses. To do this, they must have flexible and efficient production chains, with less human intervention. This is how the idea of automated processes in industry got immersed. To achieve this task, a Programmable Logic Controller has been introduced.

Industrial programmable logic controllers appeared in the USA around 1969 in the automotive assembly line. They appeared in France in 1971, after there becomes a global industrial revolution.

A programmable logic controller is a programmable electronic machine, capable of controlling a system with a simple and easy programming language with a very precise and real-time response time [16].

Our work consists in realizing a programmable logic controller on based an Arduino Uno (open plc Arduino), we then speak of our programmable logic controller (open plc Arduino) designing and construction. The latter will be able to carry out most of the applications that will be required of it by simply modifying the software while keeping the same hardware structure.

The open plc Arduino that we will realize will meet the same overall requirements expected of a classic programmable logic controller

We will a practical a project of smart irrigation using open plc Arduino, Our work is organized into three chapters:

Chapter 01: we represent a global view on programmable logic controllers and their method of operation with these different programming languages.

Chapter 02: we determine the design of the important parts of our PLC board, and we give the hardware description of open plc Arduino and software using in our project, we represent the programming compiler of the open PLC Arduino with a general view.

Chapter 03: this is the part of application on smart irrigation testing and discussion of the results of our work.

Finally, we will end this thesis with a general conclusion and perspectives.

Chapter I

**Overview about programmable logic
controllers**

Chapter I: Overview about programmable logic controllers

I.1 Introduction

The hard-wired nature made it difficult for design engineers to alter the automation process. Changes would require rewiring and careful updating of the documentation. If even one wire was out of place or one relay failed, the whole system would become faulty. Often technicians would spend hours troubleshooting by examining the schematics and comparing them to existing wiring. When general-purpose computers became available, they were soon applied to control logic in industrial process. These early computers were unreliable and required specialist programmers and strict control of working conditions, such as temperature, cleanliness, and power quality. PLC originated in the late 1960s in the automotive industry in the USA and was designed to replace relay logic systems. Before, control logic for manufacturing was mainly composed of relays, cam timers, drums sequencers, and dedicated closed-loop controllers. This chapter is an introduction to the programmable logic controller (PLC) [19].

I.2 definition

Programmable logic controllers, also called programmable controllers or PLCs, are solid-state members of the computer family, using integrated circuits instead of electromechanical devices to implement control functions. They are capable of storing instructions, such as sequencing, timing, counting, arithmetic, data manipulation, and communication, to control industrial machines and processes.

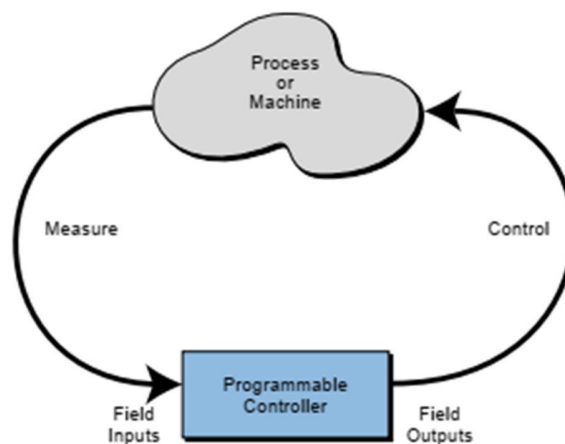


Figure 1.1 PLC conceptual application diagram [1]

Chapter I: Overview about programmable logic controllers

Programmable controllers have many definitions. However, PLCs can be thought of in simple terms as industrial computers with specially designed architecture in both their central units (the PLC itself) and their interfacing circuitry to field devices (input/output connections to the real world)[1].

I.3 Principle of operation

A programmable controller, as illustrated in Figure 1.2, consists of two basic sections:

- The central processing unit
- The input/output interface system

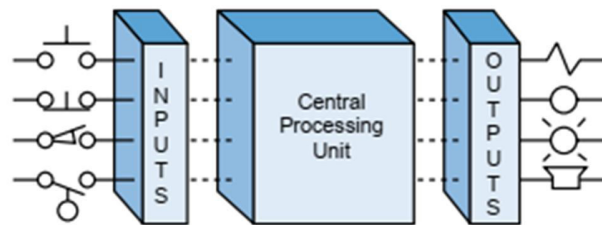


Figure 1.2 Programmable controller block diagram[1]

The central processing unit (CPU) governs all PLC activities. The following three components, shown in Figure 1.3, form the CPU:

- the processor
- the memory system
- the system power

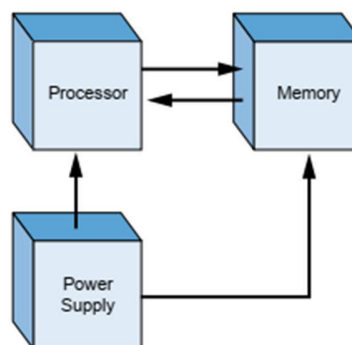


Figure 1.3 Block diagram of major CPU components [1].

Chapter I: Overview about programmable logic controllers

The operation of a programmable controller is relatively simple. The input/output (I/O) system is physically connected to the field devices that are encountered in the machine or that are used in the control of a process. These field devices may be discrete or analog input/output devices, such as limit switches, pressure transducers, push buttons, motor starters, solenoids, etc. The I/O interfaces provide the connection between the CPU and the information providers (inputs) and controllable devices (outputs). During its operation, the CPU completes three processes: (1) it reads, or accepts, the input data from the field devices via the input interfaces, (2) it executes, or performs, the control program stored in the memory system, (3) it writes, or updates, the output devices via the output interfaces. This process of sequentially reading the inputs, executing the program in memory, and updating the outputs is known as scanning, Figure 1.4 illustrates graphic representation of a scan.

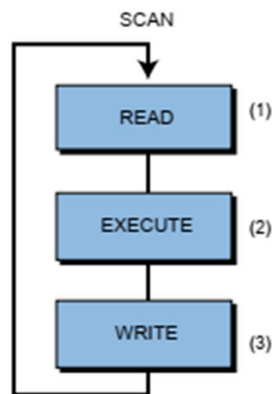


Figure 1.4 Illustration of a scan[1].

The input/output system forms the interfaces by which field devices are connected to the controller (see Figure 1.5). 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.

Chapter I: Overview about programmable logic controllers

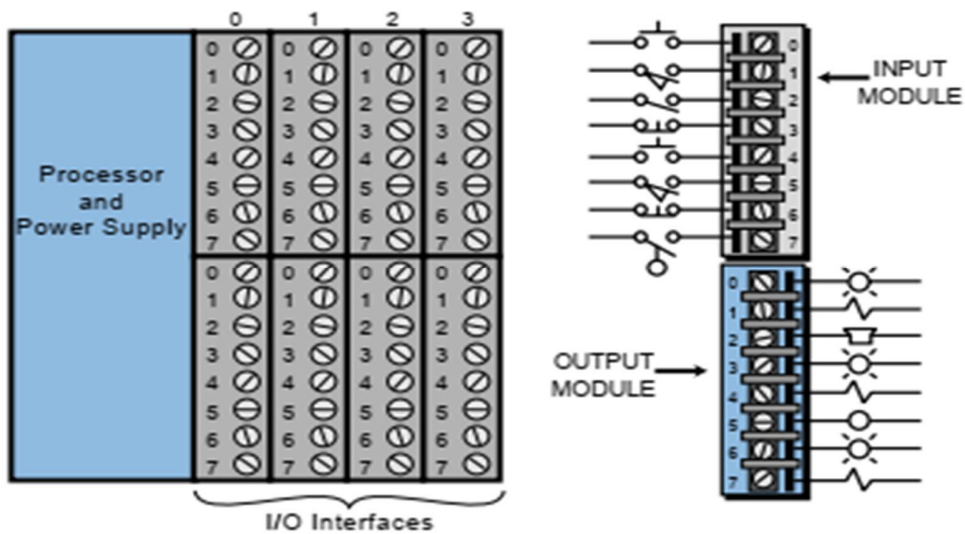


Figure 1.5 Input/output interfaces [1].

Although not generally considered a part of the controller, the programming device, usually a personal computer or a manufacturer's mini programmer unit is required to enter the control program into memory (see Figure 1.6). The programming device must be connected to the controller when entering or monitoring the control program [1].

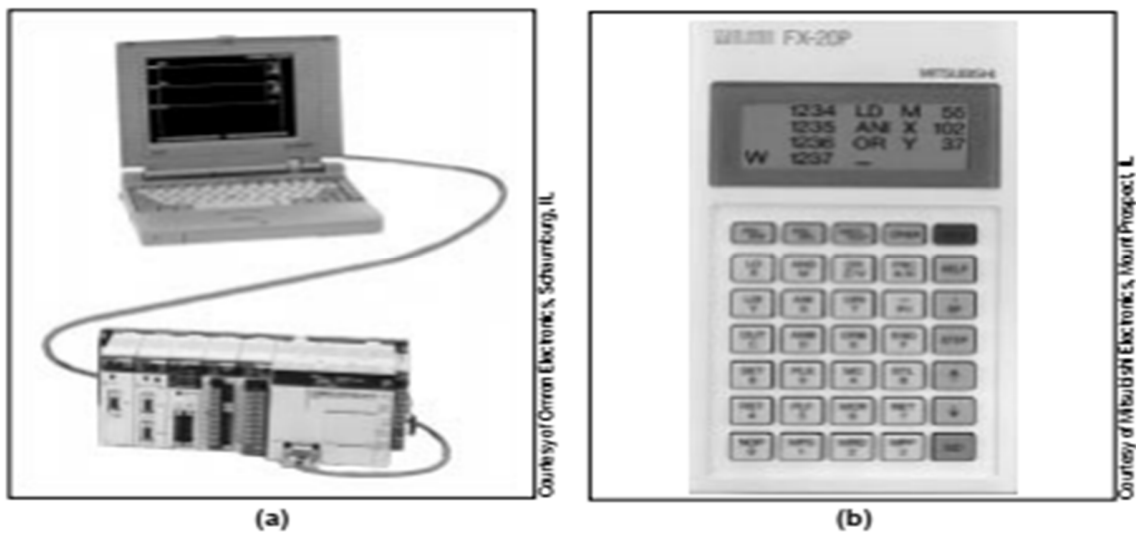


Figure 1.6 (a) Personal computer used as a programming device and (b) a mini programmer unit [18].

Chapter I: Overview about programmable logic controllers

I.4 Architecture

Figure 1.7 shows the basic internal architecture of a PLC. It consists of a central 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 a printed 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.

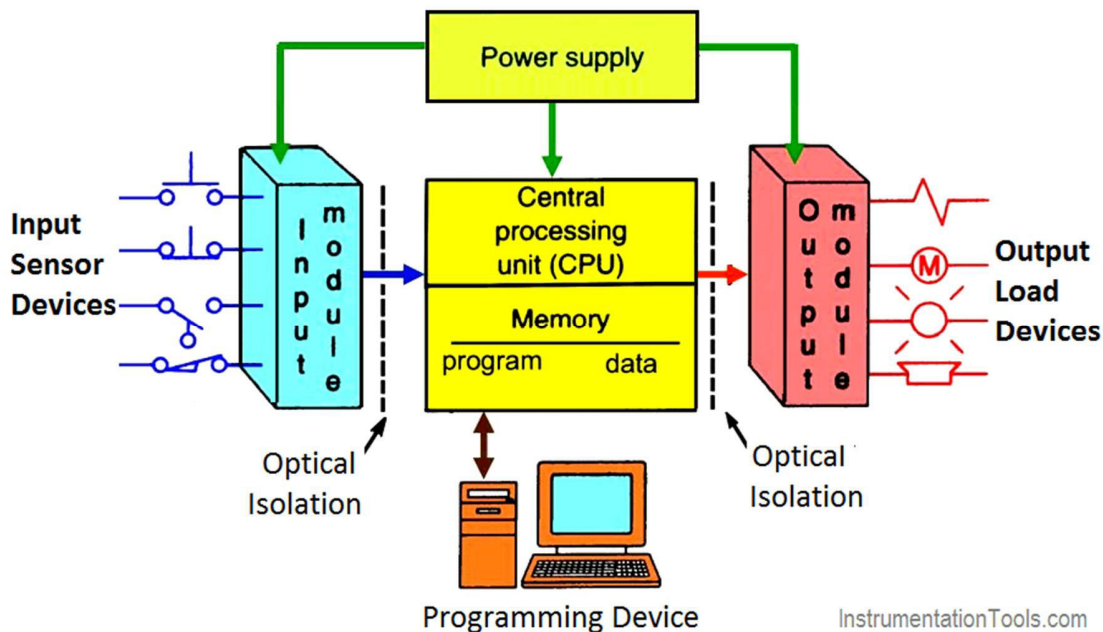


Figure 1.7: PLC hardware depiction[14]

That has 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

Chapter I: Overview about programmable logic controllers

conductors along which electrical signals can flow. It might be tracks on a printed 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 [2].

I.4.1. Input and Output (I/O) Modules

The input/output modules in PLC are of two types. It can be either digital or analog. Just like any other device or machine, we have to provide input to the PLC controller. It yields output. For example, when the user presses a button, the motor starts. Here the switch button is input. The motor is an output device. In PLC, to take the input and to return the output, there are I/O modules. The input modules are used for providing an interface for input devices like different types of switches (push button switch, selector switch, and limited switch), sensors, etc. The output modules are used for providing an interface for output devices like motor, fan, relay, light, lamp, electric heater, solenoid valve, buzzer, etc.

I.4.2 Power Supply

For PLC, we need an electrical power supply. The power supply provides power to all other components to operate. It provides power to the input/output modules, memory system, and processor. The function of the power supply is to provide the DC or AC power to operate the PLC. Most of the PLCs work at 220VAC or 24VDC.

I.4.3 Central Processing Unit (CPU)

The central processing unit is the heart of the PLC system. The function of the processor is store and executes PLC software programs. It helps to perform basic arithmetic, the logic, control, and input/output operations specified by the instructions. This consists of three sub-parts which are memory, processor and power supply.

I.4.4 Memory System

A memory system is responsible for storing and retrieving data and information. To operate the PLC system there is a need for it to access the data to be processed and instructions, that is, the program, which informs it how the data is to be processed. Both are stored in the PLC memory for access during processing. There are several memory elements in a PLC system:

Chapter I: Overview about programmable logic controllers

- System read-only-memory (ROM) gives permanent storage for the operating system and fixed data used by the CPU.
- Random-access memory (RAM) is used for the user's program. Programmable Logic Controllers
- Random-access memory (RAM) is used 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, that is, 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, and the like. • Possibly, as a bolt-on extra module, erasable and programmable read-only-memory (EPROM) is used to store programs permanently. 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. However, to prevent the loss of programs when the power supply is switched off, a battery is used in the PLC to maintain the RAM contents for a period of time. After a program has been developed in RAM it may be loaded into an EPROM memory chip, often a bolt-on module to the PLC, and so made permanent. In addition, there are temporary buffer stores for the input/output channels. The storage capacity of a memory unit is determined by the number of binary words that it can store. Thus, if a memory size is 256 words, it can store $256 \times 8 = 2048$ bits if 8-bit words are used and $256 \times 16 = 4096$ bits if 16-bit words are used. Memory sizes are often specified in terms of the number of storage locations available, with 1K representing the number 2¹⁰, that is, 1024. Manufacturers supply memory chips with the storage locations grouped in groups of 1, 4, and 8 bits. A 4K 1 memory has 4 × 1024 bit locations. A 4K 8 memory has 4 × 8 × 1024 bit locations. The term byte is used for a word of length 8 bits. Thus the 4 K 8 memories can store 4096 bytes. With a 16-bit address bus we can have 2¹⁶ different addresses, and so, with 8-bit words stored at each address, we can have 2¹⁶ × 8 storage locations and so use a memory of size $2^{16} \times 8 = 64K \times 8$, which might be in the form of four 16K 8-bit memory chips [2].

Chapter I: Overview about programmable logic controllers

I.4.5 Programming Device

A programming device is a tool that is used to enter programs or instructions into the memory of the processor of PLC. The program is entered using a programming language called relay ladder logic. As we learned previously, relay ladder logic is the most popular programming languages. Rather than using words, relay ladder logic uses graphic symbols to illustrate their intended outcome. The programs used by the programming devices are similar to a schematic for a relay control circuit[9].

I.5 GRAFCET

The GRAFCET is a functional diagram whose purpose is to describe graphically the different behaviors of a sequential automatism. Created by the AFCET (French Association for Economic and Technical Cybernetics), the Grafcet is the synthesis of about twenty description systems originally proposed (1976). Its promotion was made by the ADEPA (Agency for the Development of the Automated Production) and was then accepted by the international standardization bodies, in particular by the International Electro technical Committee in its publication 848 of 1988, from which large extracts of this course (IEC 848) [8].

I.6 Programming languages

The term PLC programming language refers to the method by which the user communicates information to the PLC. The standard IEC 61131 was established to standardize the multiple languages associated with PLC programming by defining the following five standard languages [3].

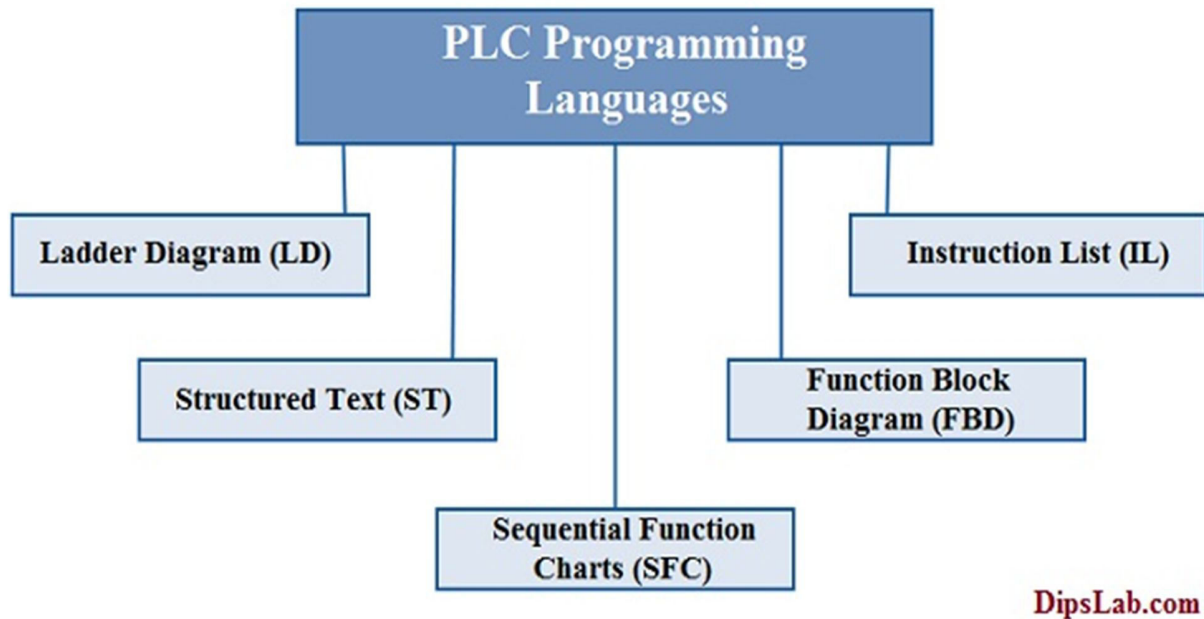


Figure.1.8 most popular language of programming[3]

I.6.1 Ladder diagram (LD)

Ladder Diagram (LD) Programming The most common language used to program PLCs is Ladder Diagram (LD), also known as Relay Ladder Logic (RLL). This is a graphical language showing the logical relationships between inputs and outputs as though they were contacts and coils in a hard-wired electromechanical relay circuit [10].

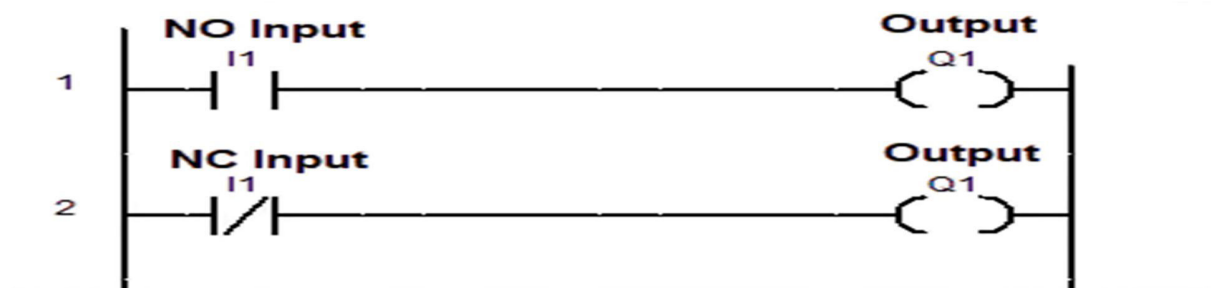


Figure 1.9ladder diagram (LD) [10]

I.6.1.1 Advantages of LD

- LD is simple logic construction and more reliable than an electronic circuit controller.
- Easy to learn and read the program.

Chapter I: Overview about programmable logic controllers

- Every programming symbol performs specific actions.
- It having good representation for discrete logic.
- Easy to troubleshoot.
- Shut down the power without the switch (i.e. hardware devices) [11].

I.6.2 Instruction List

Instruction List (IL) is another type of PLC programming language. It uses the mnemonic code. So the syntax of this programming language is easy to remember. In general, AB PLC brand works on the Instruction List (IL) programming language [11].

I.6.2.1 Advantages of IL

- It has a high execution speed.
- It takes less memory as compared to other programming languages [11].

I.6.3 Structured Text (ST)

The “ST” and “STX” are also other abbreviations used for Structured Text PLC language. It uses high-level programming language syntax. The syntax of ST is similar to the syntax of a high-level programming language with loops, variables, conditions, and operators [11].

I.6.3.1 Advantages of ST

- ST is very easier to understand for both novice and experienced programmers.
- Because of its standard coding format, it is easy to edit and modify program written in ST language [11].

I.6.4 Function Block Diagram (FBD)

Function block diagram (FBD) is a popular and easy way to write a program like a Ladder Diagram. FBD is represented like box which consists of a number of lines of code for putting different programming functions. It is a graphical language for programming logic controller. So, it makes your job easy to describe a system [11].

I.6.5 Sequential Function Charts (SFC)

Sequential function charts (SFC) is also a graphical programming language. It is not a text base. It has become a popular method of accurately specifying sequential control requirements.

Chapter I: Overview about programmable logic controllers

The benefit of SFC is easy to understand. Because you can visualize what is happening and when it is happening in the procedure of the code. The main function of SFC is only the active parts of the code are executed. Due to this, it makes easier to troubleshoot and to change the code if problems occur. These languages are accepted internationally. Among all of them, mostly the Ladder diagram (LD) programming language is used in the industry [11].

1.7 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. 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 optoisolator (the term optocoupler is also often used). Figure 1.8 shows the principle of an optoisolator. 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 [2].

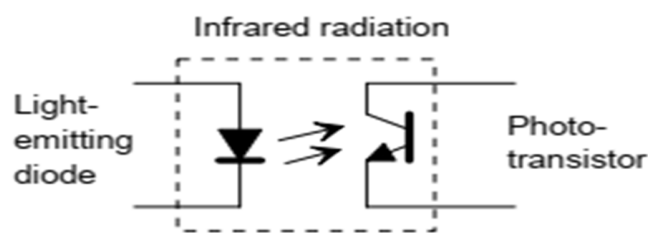


Figure 1.10 Optocoupler[2]

I.8 Types of plc

I.8.1 Modular PLC

- Is built 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 [12].

I.8.2 Compact PLC

- In compact PLC, I/O capability is fixed and determined by the manufacturers.
- 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 [12].

I.8.3 Difference between modular and compact PLC

Table1.1 Difference between modular and compact PLC

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

I.8.4 Sizes of PLC

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

Chapter I: Overview about programmable logic controllers

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. Figure 1-14 shows several PLCs that fall into this category classification [1].

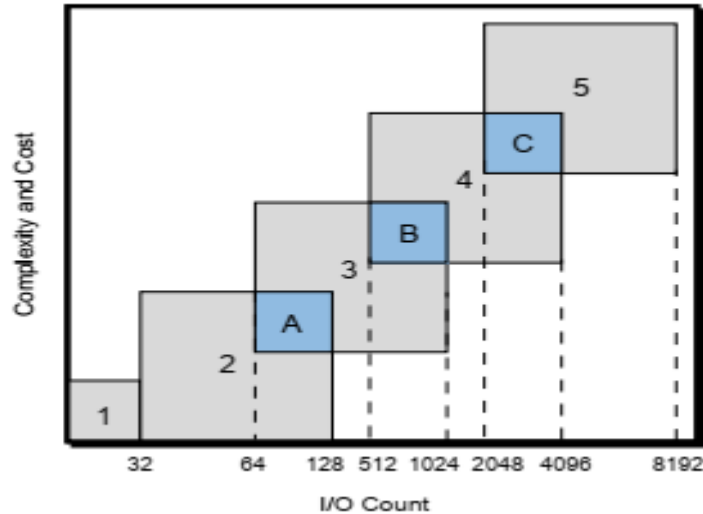


Figure 1.11 PLC product ranges[1].

I.9 The most famous companies of programmable logic controllers in the world

Table1.2 The most famous companies of programmable logic controllers in the world

Types of companies	SIEMENS	Schneider	AB(Allen Bradley)	ABB	OMRON	MITSUBISHI	HITACHI
The most Important its product	S200 S300 S400 S1200 S1500	MODICON M238 M241 M258 M340 M580	Compactlogix Micrologix1000 Micrologix1100 Micrologix1400 Contrologix	ABB AC500 ABB AC800 ABB AC800M	ZEN CPM1A CPM2A SRM1 CS1 CJ1	MELSEC MODULAR PLC A/L-Family FX	MICRO-EH SERIES H SERIES EH150 EHV HX
Company Nationality	German company	French company	American company	Swiss company	Japanese company	Japanese company	Japanese company

Chapter I: Overview about programmable logic controllers

I.10 Example of some PLC companies

Table1.3 Example of some PLC companies

Companies	PLC	Price(DA)	Structure
SIEMENS	S7-200 EM222	33,866.41	24V DC//int8/out4
	S7-300	123,180.48	24v DC//int8/out8
Schneider	TM241CE40R	94,515.93	100 to 240v //40 I/O
	TM258LF66DT4L	97,439.10	24 Dc v// 66 I/O
MITSUBISHI	Fx1s-20MT-001	12,067.91	Ac 100-220v/ in 12/out8
	Fx3SA-10MR	9,656.65	100 to 240v Ac/ in 6/out4
OMRON	CP1E-NA20DT1-D	53,518.79	24vdc /int12/out8(transistor)
	CP1E-N40DR-A	13,202.27	Ac 100-240v /int12/out8

I.11 Conclusion

This chapter was an introduction to the programmable logic controller (PLC) and its general function, hardware forms, internal architecture, and the most used languages for programming. This overview will be followed by more detailed discussion in the following chapters.

Chapter II

Realisation of OPENPLC Arduino

II. Introduction

As we discussed in the first chapter 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.

So there is a problems in using the PLC that is currently in the market, which is the high price, and also requires a program for programming that is not free and also needs a special cable for programming according to the type of PLC. For that, an idea came to have an open source PLC, capable offer excellent service and technical support and the most competitive price compared to those already used in industry automation.

In this chapter, we introduce the Implementation Industrial PLC with of open source technology based on Arduino Uno,

II.2 Generality about Open PLC Arduino

The Open PLC Arduino is an electronic card based on Arduino Uno and electronic components to simulate the role of PLC, it has digital& analog inputs and relay outputs, it can be used in the industrial field

The Open PLC Arduino 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 Open PLC Arduino is made with inexpensive components to lower its costs, opening doors to automation where it wasn't ever possible before.

II.3 Architecture of Open PLC Arduino

The Open PLC Arduino was implemented based on the architecture of actual PLCs on the market. To create the Open PLC prototype, the board has:

- Power supply
- Arduino Uno
- Inputs

- Outputs

II.3.1 Power supply

In the Open PLC Arduino we are using an external power supply 24VDC and 5 VDC, this power supply fed Arduino Uno and sensors.

II.3.2 Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable [13].

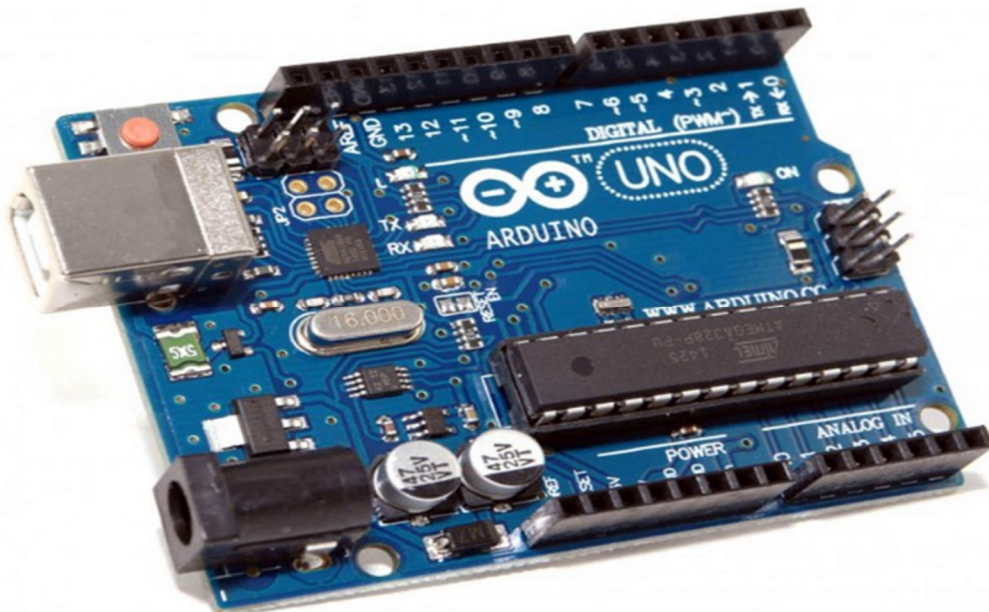


Figure2.1 Arduino Uno Card

II.3.3 Input

Inputs are used to read signals of sensors installed at the field. There are many types of inputs, depending on the sensor to be read, but they can generally be split into two categories:

- **Digital Input** can handle discrete signals, generated by devices that are either on or off; our Open PLC Arduino has 4 isolated input circuits.

The input signal conditioning circuit is composed mainly by an optocoupler (4N35) used to isolate the input signals and the control signals. The circuit of each input can be seen on **Figure. 2.2**.

When a stimulus is made between $v+$ and $v-$, a current flow through the input resistor and activates the internal LED of the optocoupler. The photons emitted by the internal LED are sensed by the phototransistor, which creates a path for the current from 5VDC to ground, sending logic 1 to input pin of Arduino Uno.

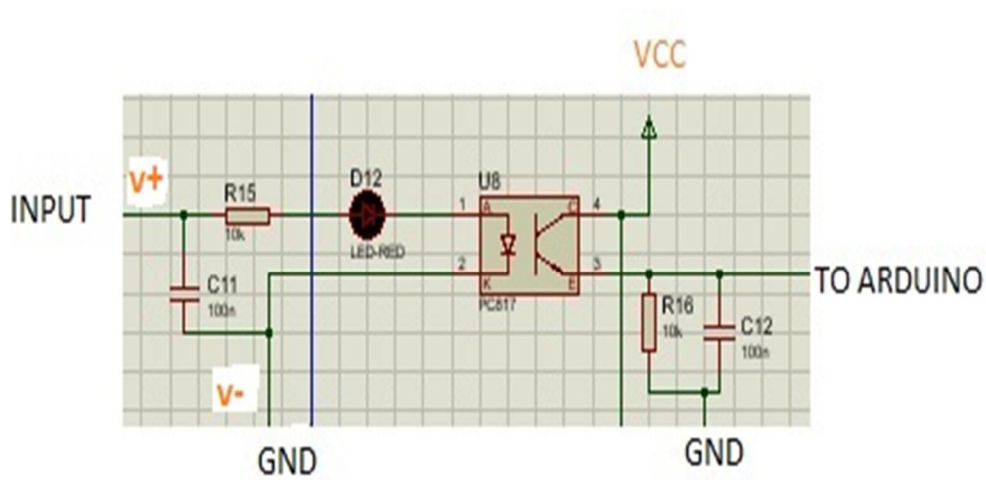


Figure2.2 Isolated digital input circuit

▪ Analog input

Convert a physical quantity to a digital number. This process of conversion is usually made by an ADC [Analog to Digital Converter] inside the analog input Arduino Uno (A0, A1, A2, A3, A4, A5) the type of the physical quantity to be read determines is voltage 0 to 5VDC.

II.3.4 Output

Digital output can control devices installed in the field as if on-off switches.

Our Open PLC Arduino has 4 relay-based outputs driving up to 4 loads at the same time. It has double isolated outputs, as they are isolated by an optocoupler (4N35) (just like the Input) and the relay itself, which gives an additional layer of isolation. Figure.2.3 shows the circuit of one isolated output.

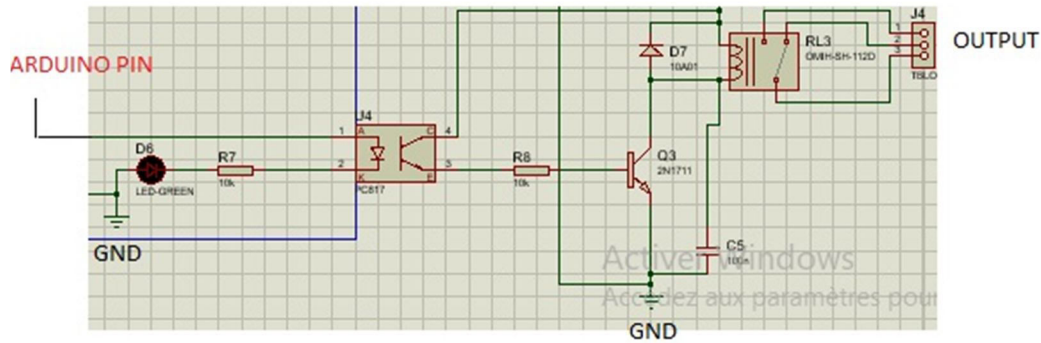


Figure.2.3 Isolated output circuit

As digital processors are better sinking current than sourcing, the cathode of the optocoupler's internal LED is connected to an output pin on the Arduino Uno. While the output pin remains with logic 0, no current flows through the LED.

If the output pin goes to logic 1, a current is drawn on that pin, activating the optocoupler's internal LED.

The internal phototransistor is connected to an external BC 548 transistor, when photons are sensed by the internal phototransistor, both transistors are polarized, energizing the relay's coil. Without photons, there isn't any current flowing through the coil, and the relay remains off.

II.3.5 Protections

The protections used in the Open PLC Arduino input and output:

- ✓ Optocouplers and relays were used to isolate high power circuits from control logic. The filled zones were connected to ground and only the low power zones of the board were filled.
- ✓ To protect against reverse polarity on inputs, diodes were connected in series to allow current flow in only one direction. Also, capacitors were used in parallel to ground to filter noise from sensitive devices.

II.4 Open PLC Arduino characteristics

The general characteristics of Open PLC Arduino proposed are shown in the following table

Table 2.1 Open PLC Arduino features

power supply	24VDC / 5VDC
Microcontroller	Microchip ATmega328P
Programming cable	USB cable type B
Inputs	4 Digital / 24 V DC 6 Analog / 0 to 5V DC without protection
Outputs	4 Relays
Supplement	4 Digital / 5V DC without protection can use as input or output

II.5 Implementation the Open PLC Arduino board

After we have done the theoretical work to complete Open PLC, we will move on to the practical, which is divided into two parts:

Drawing the circuit and printing it on PCB:

In this part we worked on the software Proteus (8.13 version), it is a used for electronic design automation; mainly used for preparing schemas and designing PCBs. It has many tools which are not only easy to use but are helpful in PCB designing. It takes to use of an integrated auto-router, and allows for the full schematic capture, configurable design rules, supports power planes, has an interactive circuit stimulator, follows industry standard CAD/CAM and ODB++ output, and allows 3D viewing[15].

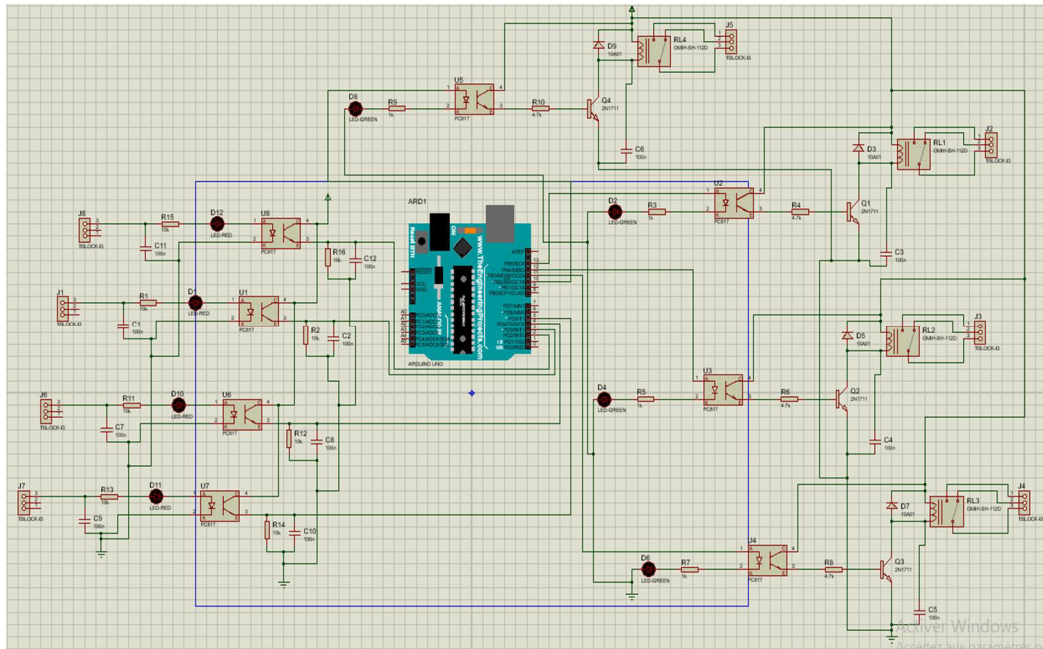


Figure.2.4 The schemas of Open PLC Arduino

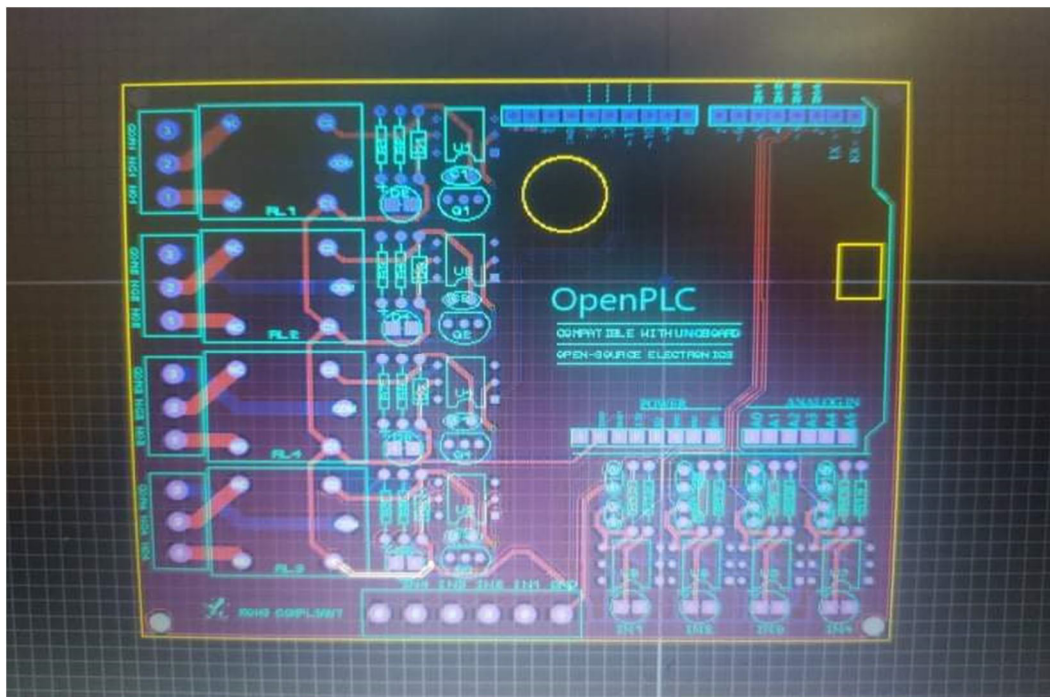


Figure.2.5 The PCB designing of Open PLC Arduino

Installing the electronic parts and welding them on the PCB

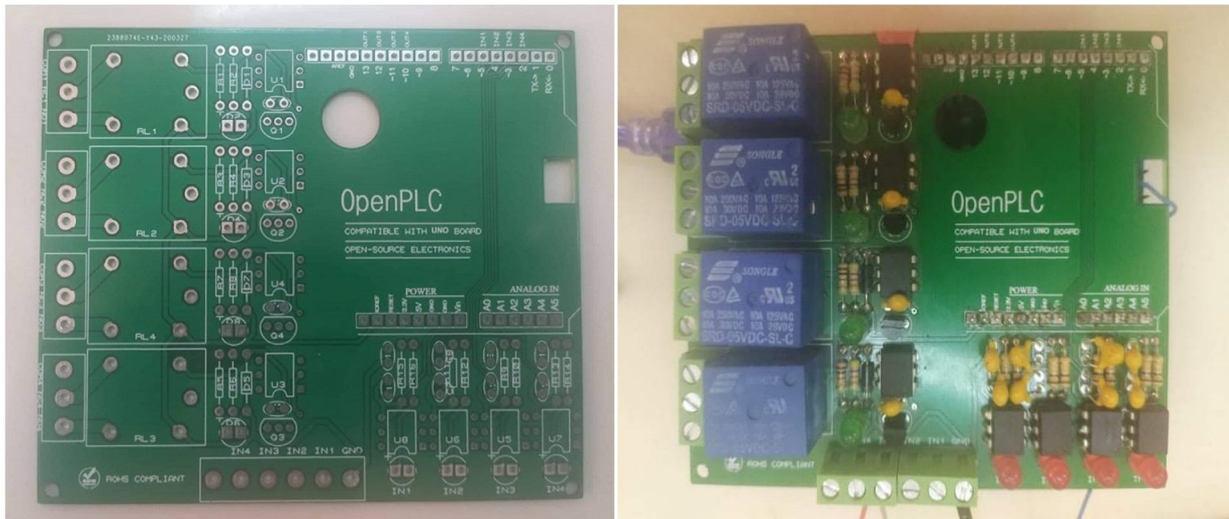


Figure.2.6 OpenPLC Arduino board before and after installation the components

After purchased the parts and the construction of open plc Arduino was completed, we wrote down the type of parts with the quantity and purchase price and that is shown in the **Table 2.2**. We note that the cost of open plc Arduino is very low compared to PLC.

Table 2.2 the components used in implementation the Open PLC Arduino

Components	Reference	Quantity	Price
Optocoupler	4N35	8	80 DA
Transistor	BC 548	4	80 DA
Diode	1N4007	4	5 DA
Capacitor	100Nf	12	30 DA
Resistance	10 K Ohm	8	5 DA
Resistance	1 K Ohm	4	5 DA
Resistance	4.7 K Ohm	4	5 DA
LED	GREEN	4	10 DA
LED	RED	4	10 DA
RELAY	SRD-05VDC-SL-C	4	200 DA
Arduino Uno	Uno R3 ATmega328P	1	2200 DA
Board PCB	17 cm/12 cm	1	400 DA
TOTAL PRICE			4900 DA

II.6 language and environment programming

The programming is the set of development and implementation activities that make it possible to command and control an automated system. A program is a set of instructions and information intended to be executed in order to apply a certain task.

In Open Plc Arduino, we can use the application Arduino IDE to programming and load the code to our open PLC board.

The Arduino IDE (integrated development environment) is an open-source software, designed by Arduino.cc and mainly used for writing, compiling & uploading code to almost all Arduino Modules. It is official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is available for all operating systems i.e. MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing and compiling the code [16].

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

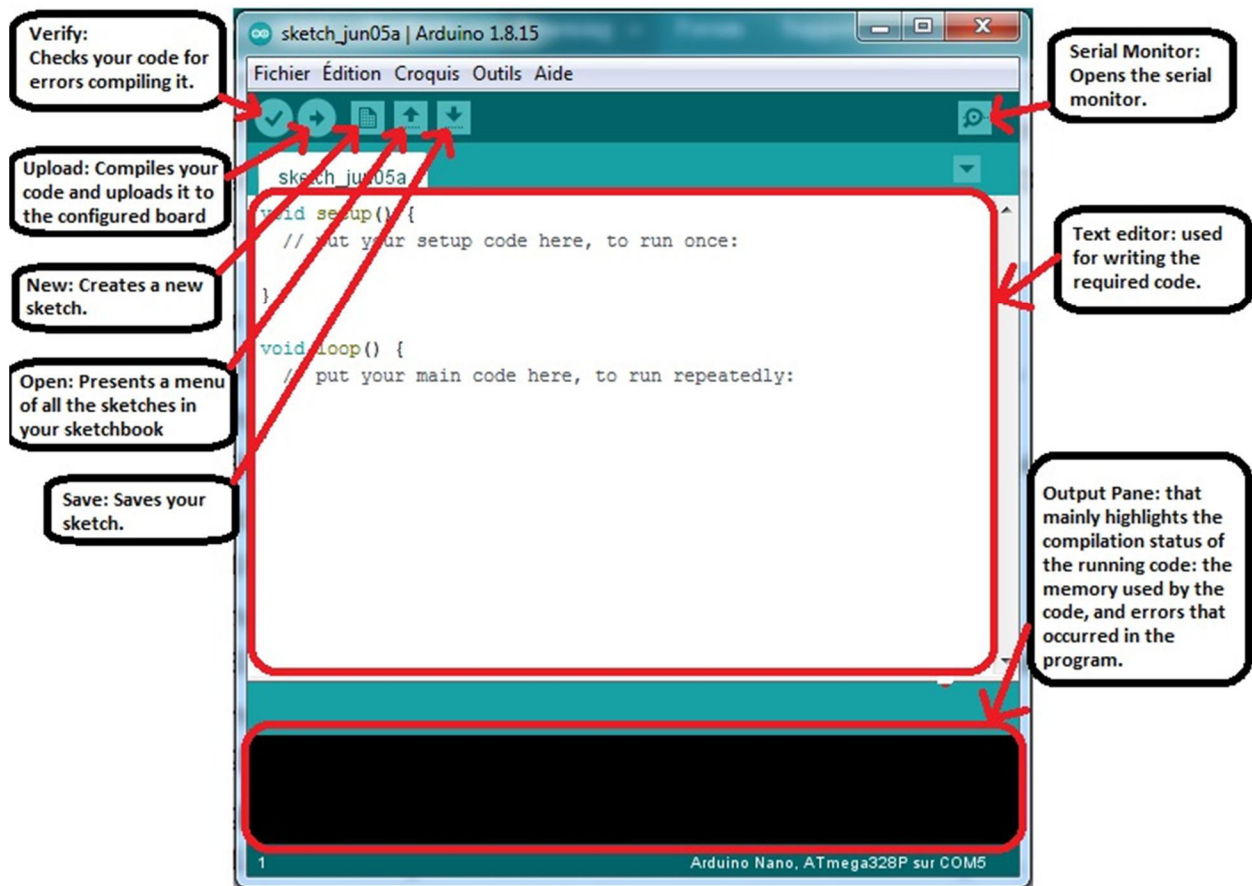


Figure.2.7 Arduino IDE description

II.7 Conclusion

This chapter presents the most important part and the basis of this project which is the design of a programmable automaton based on an Arduino card.

The chapter describes the main diagram of the open PLC Arduino board which is based on the Arduino Uno and the coordination with the input and the output, and then the Construction and assembly board.

We got very good and encouraging results in terms of cost compared to PLC available in the market, add to that the free programming software.

In the next chapter we will choose an automation project to achieve and use our open plc Arduino board.

Chapter III

Application the Open PLC Arduino to control the sprinkler irrigation system

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

III.1 Introduction

Based on our study in the second chapter, which depends on implemented the Open PLC Arduino that can be applied to control different system in various fields; in this chapter we will apply it to control the agricultural irrigation system.

III.2 General Information about agricultural irrigation systems

III.2.1 Definition

Irrigation is an effective management tool against the vagaries of rainfall and is crucial for the world's food resources [4].

III.2.2 Agricultural irrigation techniques

Agricultural irrigation techniques are methods to bring water to crops to increase their production, and to allow their normal production, and to allow their normal development in case of water deficit induced by rainfall and are classified and are classified as surface irrigation, sprinkler irrigation and micro irrigation. Deciding to select an irrigation technique or to switch to a more efficient one is complicated. Efficient technique is complicated. From a water conservation standpoint, the choice is simple, as water savings increase when water savings increase when switching from surface irrigation to sprinkler and from sprinkler to from sprinkler to micro irrigation. However, the success of an irrigation technique will be highly. However, the success of an irrigation technique will be highly dependent on the site, situational factors as well as the level of management used. The existing irrigation technique must be evaluated very carefully before switching to another technique [17].

Among the most widely used types of irrigation systems in the world, there are three:

- ❖ Surface irrigation (by gravity).
- ❖ Sprinkler irrigation.
- ❖ Localized irrigation (drip irrigation) [5].

We choose among these types spray irrigation Sprinkler irrigation for controlling by Open PLC Arduino which implemented in the second chapter.

III.2.2.1 Sprinkler irrigation

Pivot and front boom systems are installations that are used mainly on large farms used mainly in large-scale operations. They have a programmable drive mechanism to move the

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

elements. The swivel boom system consists of a pipe with sprinklers with sprinklers, supported at one end by a central pivot tower, a series of wheeled towers and a motor. Towers with wheels and an electric (or hydraulic) motor. The pipe can measure from 100 to 500m, and can irrigate up to 75 ha. The whole system allows irrigating a circular area, but requires a high requires a high investment capital. The flow rates are in the order of 250 to 850 m³/h for a pressure of 6 bars.

The front boom system differs from the pivoting boom by the fact that all the towers are mobile and the movement is done laterally. The water supply is done either by a ditch dug in the middle or at the edge of the fielding the middle or at the edge of the field, or by a flexible pipe. It requires an investment as important if not or even higher than the pivoting boom system. The energy consumption of both systems is high. Both systems are high.

Other types of installations are: the towed or traileed wing, rotating arm or giant sprinkler, and the self-propelled cannon (self-propelled irrigation machine with flexible hose: reels) [6].



Figure 3.1Pivot irrigation [20]

Advantages

Does not require any prior leveling of the soil.

- Can be used whatever the nature of the watered soil
- It causes a strong oxygenation of the water projected in rain.
- It ensures a good homogeneity of the distribution of the contributions [5].

Disadvantages

- It would promote evaporation which is all the more intense as the droplets are finer and the air drier.
- Difficulty of use and reduced efficiency in windy regions.
- Wetting of the leaves favors cryptogamic diseases in certain plant species.
- Plant species.
- Reduced possibilities for watering with wastewater (formation of aerosols).
- Difficulty in moving the equipment in areas with high crops [5].

III.3 Controlling the Sprinkler irrigation system by Open PLC Arduino

In this part, the sprinkler irrigation system is controlled using the Open PLC Arduino that we touched on, in the second chapter. The system is controlled via a humidity sensor, through which we estimate the amount of water needed to feed the plant.

III.3.1 Grafset

The sprinkler irrigation system proposed is controlled according to grafset (Figure 3.2). Our system consists of: three inputs, and four outputs.

Inputs:

- Start button (BM)
- Stop button (BS)
- moisture sensor (h)

Outputs:

- Green led indicator light (LV)
- Red led indicator light (LR)
- Motor to rotate the sprinkler shaft (MP)
- Water pump (M)

Firstly, the system starts working by pressing the BM button, where the green led glows. Next, the sensor starts to read the value of humidity, if the humidity value is less than 10% (which means that the soil is dry and we need watering), it sends a signal by Open PLC Arduino to the

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

water pump (M) and motor to rotate the sprinkler shaft (MP) for starting watering, when the humidity value is greater than 90%, the system stops. The system can be stopped manually by the stop button BS, where the red led glowing, as shown in Figure3.2

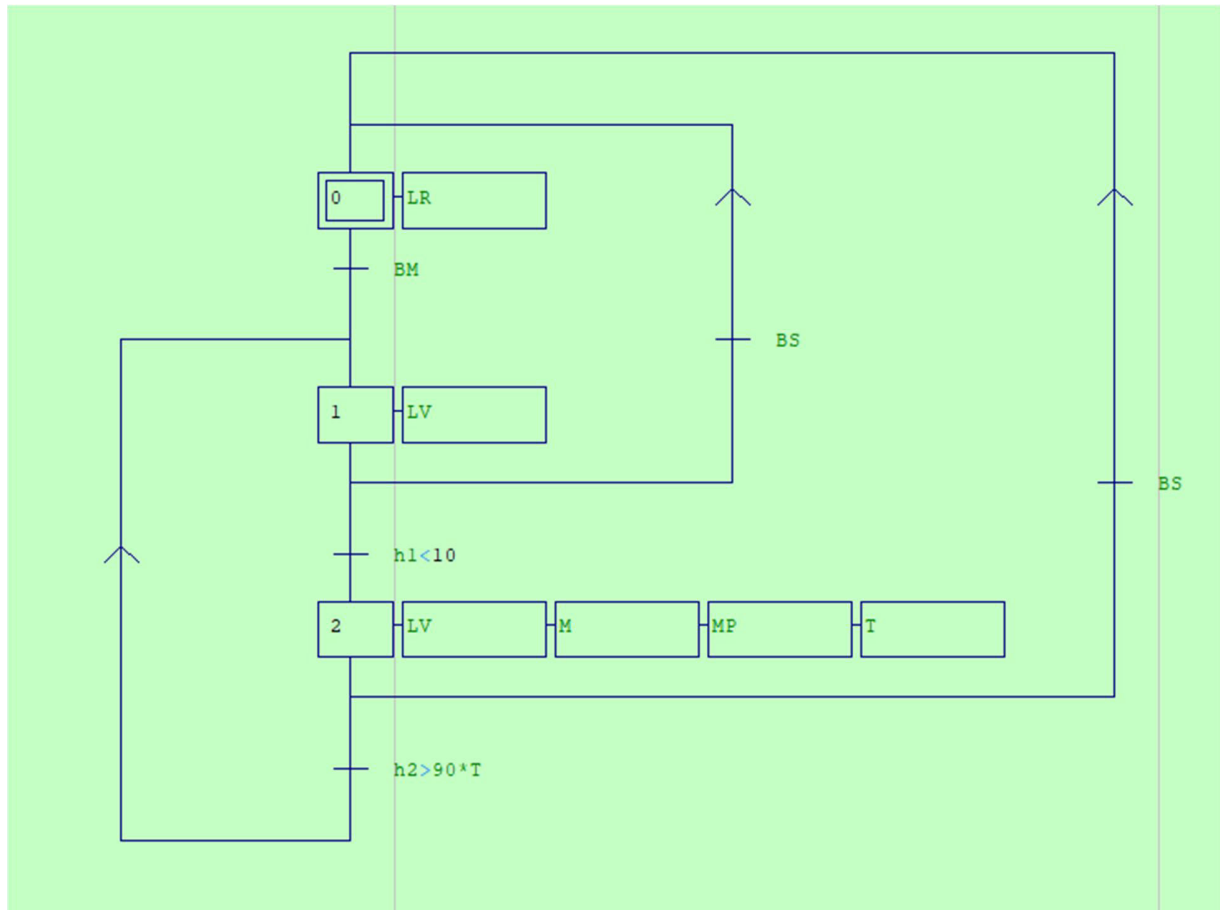


Figure 3.2 Grafcet of the sprinkler Irrigation System

From Figure 3.2, set, reset and output equations are extracted, as follows:

❖ Set equations

$$sx0 = x1 * BS + x2 * BS + (\overline{x0} + \overline{x1} + \overline{x2})$$

$$sx1 = x0 * BM + x2 * h2$$

$$sx2 = x1 * h1$$

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

❖ Reset equations

$$rx0 = x0 * BM$$

$$rx1 = x1 * h1 + x1 * BS$$

$$rx2 = x2 * BS + x2 * h2$$

❖ Output equations

$$LV = x1 + x2$$

$$LR = x0$$

$$M = x2$$

$$MP = x2$$

❖ States equations

$$x0 = sx0 + x0 * \overline{rx0}$$


$$x1 = sx1 + x1 * \overline{rx1}$$

$$x2 = sx2 + x2 * \overline{rx2}$$

III.4 Used equipment

In this table 3.3, all the elements needed to build a sprinkler irrigation model with pictures of each of them are shown.



Table 3.1 Used Equipment

<p>Wheel + Tyre + Motor DC 3V-6V: A DC motors with gear, single axis, with DC 6V operating voltage and an RPM of 100R / minute. This gear box is ideal for robotic car or line-tracing robot. With plastic construction and colored in bright yellow, the DC gear motor measures approx. 6.5cm long, 2.5cm wide and 1.8cm thick.</p>	
--	--

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

<p>Car windshield washer pump: Car windshield washer pump, which can resist -25 degrees to above 80 degrees. Specifications: Material: ABS Color: Black Displacement: Approx. 450ml / 10s Resistance to Low Temperature: -25 degrees Resistance to High Temperature: Above 80 degrees.</p>	
<p>Electrical wires: Material: Copper Length: 150 cm Diameter: 2*0.5</p>	
<p>White plastic tubes + Tee + cover: Tube length: 60 cm</p>	

Chapter III: Application the Open PLC Arduino to control the sprinkler irrigation system

<p>A device that helps transmits electricity and prevents wires from sticking to the irrigation axis.</p>	
<p>An iron arm helps rotate the sprinkler axis and allows water to pass through it and is located between the water pump and the sprinkler axis.</p>	

III.5 Conclusion

In this chapter, we have applied the Open PLC Arduino to control the sprinkler irrigation system prototype via the humidity sensor through which we determine the amount of water needed for irrigation. The result is that the Open PLC Arduino board works efficiently and at a lower cost compared to used PLC in industry.

Conclusion

Conclusion

Nowadays, the use of programmable logic controllers is not only in the industrial field, they are found in home automation, in embedded systems, agriculture....

But there are problems in using the PLC that is currently in the market, which is the high price, and also requires a program for programming that is not free and also needs a special cable for programming according to the type of PLC

During this work of end of study, we have realized our idea and our goal of creating a programmable automation based on an Arduino card, with low cost and open source

We did the main diagram of the open PLC board which is based on the Arduino Uno microcontroller and the coordination with the input and the output, and then the Construction and assembly the board.

We are developing an application with our PLC board which is control the sprinkler irrigation system as a general example to express the functionality and ability of our PLC board to control systems and perform tasks and different productive projects.

After we have implemented a mini-central pivot irrigation system where the control of the water pump and the rotating wheel drive is been by humidity sensor so that the watering process is done automatically, we used our panel in the control process and we made a program that corresponds to the proposed specifications, and we got satisfactory results

The final results are a success since the objective is achieved, PLC with low cost, good efficacies and free programming environment.

This work can possibly be improved and further developed, we propose the following ideas:

- Build our compiler to program our Open Plc Arduino by ladder or FB language.
- Expand the panel and increase the number of inputs and outputs
- Create other PLC card as extension and make communication by different methods with Open PLC Arduino.

Finally, we hope that this modest work will serve as a valuable document for future promotions.

Bibliography

- [1] L.A.Bryan. E.A.Bryan , "programmable logic controllers theory and implementation", 2nd edition published by industrial text company , 1997, Isbn 0_944107_32_X.
- [2] W.Bolton"programmable logic controllers "4th edition/p 6 to 10 .
- [3] programmable logic controller frank d petruzella 2017//Page81
- [4] Salhi Ammar et Boukerker Hatem,"Etude et réalisation d'un système d'irrigation intelligente",Master thesis,15 septembre 2021,Université de mohamedel-bachirel-Ibrahimi - Bordj bouarreridj,Page 3
- [5] Taibi hamida kastali fatima,"Dimensionnement d'un reseau d'irrigation du perimetre hessaine wilaya de mostaganem", Master thesis, 2017/2018, University djillali bounaama – khemis miliana.
- [6] Aouata Ibrahim, "Étudeet developpement de l'irrigation souterraine en algerie", Mastertheses, Jan – 2015, Ecole nationale superieure d'hydraulioue -Arbaoui abdallah- Departement d'irrigation et drainage.
- [7] Makouf Sabrina," Conception d' un automate programmable à base d' arduino", Master theses, 2019/2020, University Mohamed Khider de Biskra.
- [8] <https://www.specialautom.net/automatism/Grafcet.pdf> 06/07/2022
- [9] <https://learn.automationcommunity.com/plc-programming-devices/> 06/08/2022
- [10] <http://www.bing.com/search?q=ladder+diagram&src=IE-SearchBox&FORM=IE8SRC> 06/08/2022
- [11] <https://dipslab.com/different-types-plc-programming-languages/> 06/08/2022
- [12] <https://testbook.com/question-answer/-has-a-fixed-number-of-io-modules-and-exter--615c02ec007cdf514761e14e> 06/08/2022
- [13] https://en.wikipedia.org/wiki/Arduino_Uno 06.03.2022

[14] <https://electricala2z.com/motors-control/plc-programmable-logic-controller-hardware-components-plc-hardware-basics/>16/7/2022

[15] <https://www.electronicshobby.com/2018/09/design-your-own-pcb-by-using-proteus.html>/06.03.2022

[16] <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>06.04.2022

[17] <https://irrigazette.com/fr/articles/les-techniques-dirrigation-agricole>6/7/2022

[18] <http://pielsi.blogspot.com/2007/08/principles-of-operation.html> 16/7/2022

[19] <https://www.emerson.com/documents/automation/control-valve-handbook-en-3661206.pdf>

[20] <https://irrigazette.com/fr/articles/inverstir-dans-un-pivot-ou-une-rampe-frontale>18/6/2022