لجمهورية الجزائرية الديمقراطية الشعبية

زارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Télécommunication

Spécialité : Réseaux et Télécommunication, Système et Télécommunication

Mémoire

Présenté en vue de l’obtention du Diplôme de Master

Thème:

Elaboration d’un serveur web à moindre coût

Présenté par: *Debabi Rym*

Bouguenna Mohamed

Encadrant : *Hafs Toufik* *M.C.A* *UBMA*

Jury de Soutenance :

Nasri Seif Allah el mesloul	M.C.B	UBMA	Président
Hafs Toufik	M.C.A	UBMA	Encadrant
Kouadria Nacereddine	M.C.A	UBMA	Examinateur

Acknowledgement

First and foremost, we would like to thank our Allah (SWT) for giving us the strength and patience throughout all this journey. For without Him, we could've never accomplished anything at all.

Our sincere and profound gratitude goes to our parents, we would like to thank them so much for all the support, understanding and care they gave us during the making of our thesis. For without them, their prayers and motivation we could have not been able to finish this project successfully.

We would like to express our deep and sincere gratitude to our supervisor Dr Hafs for providing support, guidance, and all the equipment that we needed to complete this project. As well as helping us with his immense knowledge and experience. Without him finalizing this project within a limited time frame would have been difficult.

We wish to show our appreciation and acknowledge the help of Lahmar Hocine who offered valuable information, tips and support.

We would like to extend our special thanks to our friends and family who contributed with the moral support and have been always there for us during our time of need.

Finally, we would like to warm heartedly thank all the teachers who have gave us a good education, collaborated in our training and knowledge from the first cycle of study until the end of university cycle.

Dedication

This thesis is dedicated to God Almighty my creator, he has been the source of my strength throughout this period. For without his graces and blessings I wouldn't have been able to finish this study.

I dedicate this work to my beloved parents who continually provide their moral, spiritual, emotional and financial support. I would like to thank them wholeheartedly because they gave me the tools and values necessary to be where I am standing right now. I present this thesis and hope that I can make you proud the same way that I am proud of having both of you as my parents.

I also dedicate this work to my beloved ones and family: to my three precious sisters whom I appreciate and love unconditionally and my best friends Dounia and Selma who were always there for me to support me and lift me up throughout this process. I will always appreciate you and your presence in my life.

I special dedication ton my best friend and binomial Bouguenna Mohamed who was always there to support me and lift me up and help me when I face any issue. I wouldn't have been able to finish this thesis alone without his help and encouragement.

And lastly I dedicate this to my fellow students and friends: Rayene, Aymen, Nouh and Riad. I am glad that I have shared all these years with you from the first cycle of study to the end of the university cycle you were always kind, supportive and funny. It has been a long journey that I wouldn't have had the effort to complete if it wasn't for your support.

Debabi Rym

I dedicate my thesis to my family. A special feeling of gratitude to my loving parents, who gave me words of encouragement and supported me morally and financially. My sisters and my brothers who have never left my side and are very special to me.

I also dedicate this dissertation to my friends who supported me throughout the process of making this project. I will always appreciate all they have done, especially Hocine for helping, advising and supporting me.

And finally i dedicate this work and give special thanks to my best friend and my binomial in this work Rym Debabi for being there for me throughout the entire study program. And i wish you all the best for the rest of your life.

Bouguenna Mohamed

Abstract:

A server is the central node in the network domain; it can be programmed to operate numerous services to customers at the same time, like storing, sending, and receiving data, and providing resources, programs to other computers, as well as other things. The server has physical and technical features, such as capacity and dimensions which control its performance, use, and cost. There are multiple types of servers, each one serves a particular function. Almost all networks have one or more of the common server types. In this project, we will make a pocket web server and program it to serve several services at a very symbolic cost.

Key words: Server, webserver, network, programing, low-cost.

Résumé:

Un serveur est le nœud central du domaine du réseau ; il peut être programmé pour assurer simultanément de nombreux services aux clients, comme le stockage, l'envoi et la réception de données, et la fourniture de ressources, de programmes à d'autres ordinateurs, ainsi que d'autres choses. Le serveur possède des caractéristiques physiques et techniques, telles que la capacité et les dimensions, qui contrôlent ses performances, son utilisation et son coût. Il existe plusieurs types de serveurs, chacun remplissant une fonction particulière. Presque tous les réseaux possèdent un ou plusieurs des types de serveurs courants. Dans ce projet, nous allons fabriquer un serveur web de poche et le programmer pour qu'il serve plusieurs services à un coût très symbolique.

Mots clés: Serveur, serveur web, réseau, programmation, faible coût.

ملخص:

الخادم هو العقدة المركزية في مجال الشبكة؛ فمن الممكن برمجتها بحيث تعمل على تشغيل العديد من الخدمات للعملاء في نفس الوقت، مثل تخزين البيانات وإرسالها وتلقيها وتوفير الموارد والبرامج لأجهزة الكمبيوتر، فضلا عن أمور أخرى. يتمتع الخادم بميزات مادية وتقنية، مثل السعة والأبعاد التي تتحكم في أدائه واستخدامه وتكلفته. هناك أنواع متعددة من الخوادم، يعمل كل منها على وظيفة معينة. تحتوي جميع الشبكات تقريبا على واحد أو أكثر من أنواع الخوادم الشائعة. في هذا المشروع، سنصنع سيرفر جيبي ونبرمجه ليخدم عدة خدمات بتكلفة رمزية جدا.

كلمات مفتاحية: خادم، خادم ويب، شبكة، برمجة، تكلفة منخفضة

LIST OF ACRONYMS

WAN	Wide Area Network
LAN	Local Area Network
RAM	Random-access Memory
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
IMAP	Internet Message Access Protocol
FTP File	Transfer Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
OS	Operating System
JSP	Java Server Pages
IDE	Integrated Development Environment
ARM	Advanced RISC Machines
USB	Universal Serial Bus
AC	Direct Current
DC	Alternating Current
PC	Personal Computer
SD	Secure Digital
VPN	Virtual Private Network
MAC ADRESSE	Media Access Control Address
API	Application Programming Interface
DNS	Domain Name System
Wi-Fi	Wireless Fidelity
CPU	Central Processing Unit

SUMMARY

List of Figure:

Chapter I:

Chapter II:

Chapter III:

LIST OF TABLES

General Introduction

Introduction

The Internet have grown rapidly in recent years and are now primed and ready to provide a wide range of services. Computers are now and will proceed to be an important part of the daily life of each individual. Consumers, institutions, and companies are highly depending on networks to send and receive data.

A web server powers all Internet operations that are in control of serving the user's requests. Furthermore, the second necessary tool for displaying these web pages is a web browser that will translate lines of code into a more readable format of texts and images. Web servers, such as consumer software, are immensely powerful applications installed on computers that process various web page languages. Not all web servers on the industry are the same; they vary in terms of architectural style as well as distribution model, and according to that they can be very expensive. That is why in this thesis, we will figure a way to adjust our site to our server at the lowest cost possible while preserving its high performance.

To achieve a good price/quality ratio, we will build a server with an Arduino board, which is a microcontroller, and an Ethernet Shield to connect it to the internet. And we'll program it to work with any network, ADSL or mobile. We will also deal with the issue of port blocking by local businesses.

Chapter I: Introduction to Web Servers and Arduino Boards

1.1. Introduction:

In network terminology, a client is a piece of hardware or software used to communicate with a data provider (server). Normally, only one user uses a specific client at a time. A client connects to a server to send and receive information. Think of a client as a program that gets information from somewhere else. A server is usually a large computer capable of providing data to many clients at the same time. The word server can mean the physical computer or piece of hardware, or it can refer to the actual server software or daemon running on that machine. A daemon is a program that offers a service to other programs, usually over a network. It accepts requests from clients, processes the requests, and returns the results to the requesting client. Although the client and server can be on the same machine, they are usually on separate machines connected by some kind of network [1]

1.2. What is a server?

A server is a dedicated computer that provides services on behalf of clients, such as ordinary desktop computers or work stations. So it's a centralized machine where multiple clients connect to, either over the internet or in a local area network, and they connect to a server for a specific service for example that service could be to retrieve a website or access data or send an email and so on. A server could be dedicated to handle one of these services only (where you would have one server dedicated for website, one server dedicated for data storage, and a server for email). And this model is what larger organizations use. Or you can also set up a server to handle each of these services on the same machine, which is what typically happens in smaller organizations. So depending upon which setup is used, it all depends on the needs of an organization.

A server is not just an actual computer, it is actually a role that a computer can take. Any ordinary desktop computer can be set up as a server, and it doesn't necessarily have to be a powerful computer. However desktop computers do have their limitations because they are not designed to handle a large workload and they cannot handle a lot of incoming connections from users. It's not only because of their inferior hardware but it's also their software.

That is why servers needs to be running 24/7 and if one of them breaks down, all the web sites or data or whatever service it is providing will become inaccessible and this makes customers unhappy. [2]

Figure 1. Servers

1.3. Characteristics of a server:

Processor: it is the nerve center of the server. The speed and number of processors impact its ability to support applications [3]

Figure 2. Intel Xeon Processor

Numbers of cores: which is the number of physical processors contained in the processor. Servers usually use Intel Xeon processors because it can support a multi-processor environment.

Cache memory size: A large cache memory reduces the frequency of data retrieval by the processor. Processors with multiple cores and that are running at a high frequency usually have a large cache size to provide optimal performance [4].

RAM: The amount of RAM memory available is proportional to the number of operations that the server is able to perform simultaneously without having to access the hard disks [5]. For example a Xeon processors work with ECC RAM (Error Correcting Code).

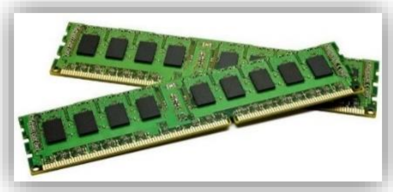


Figure 3. ECC RAM

Hard drive: A server should have a hot swappable hard drives in a RAID configuration. Because if a hard drive were to fail then no data loss would happen and the server would still be up and running because of RAID [5].

Figure 4. Server Hard

1.4. Different types of servers:

There are different types of servers, we mention some of them in the following figure:

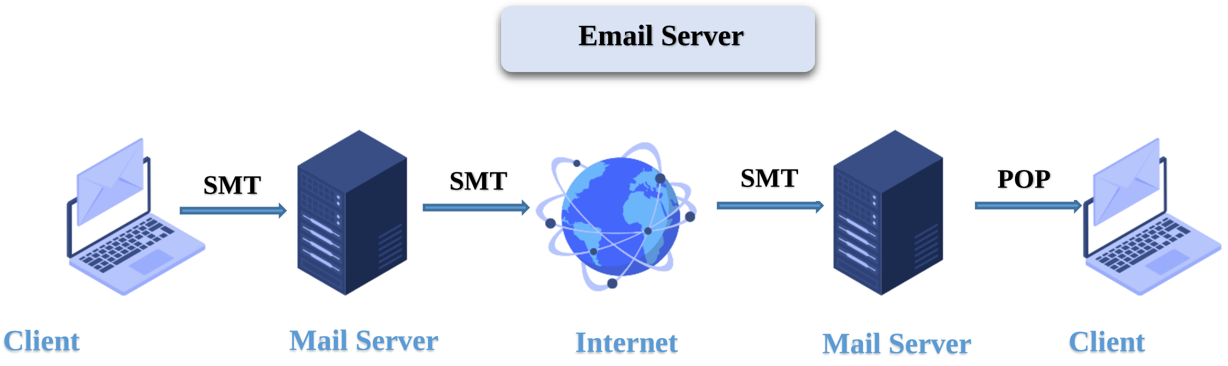


Figure 5. Email Server

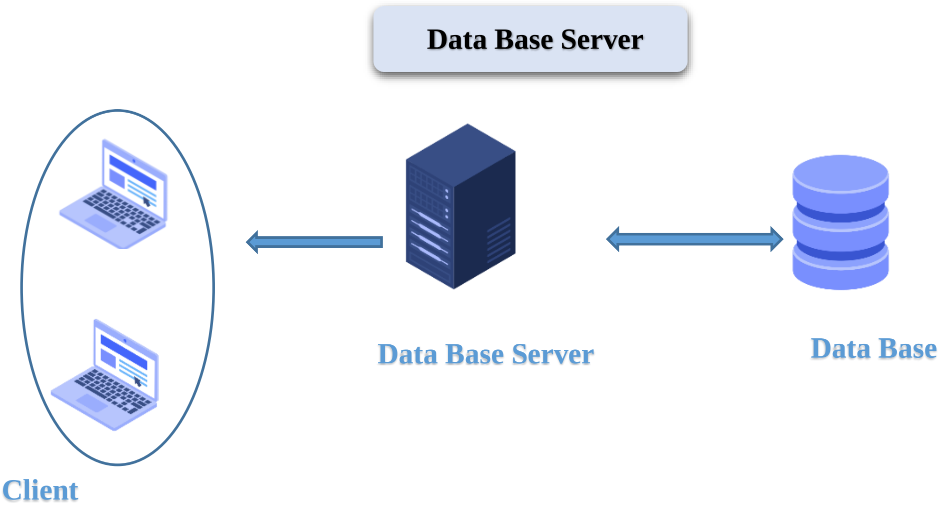


Figure 6. Data Base Server

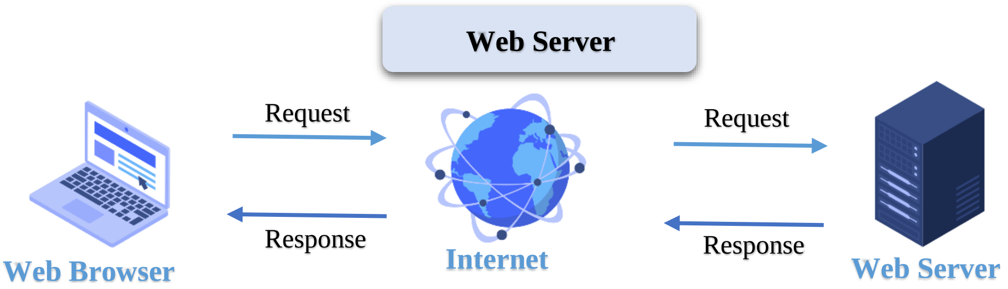


Figure 7. Web Server

1.5. Web Servers:

1.5.1. What is a web server?

A web server can be referred to as either the hardware (the computer) or the software (the computer application) that helps to deliver content that can be accessed through the Internet.

- On the hardware side, a web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files). A web server connects to the Internet and supports physical data interchange with other devices connected to the web [6].
- On the software side, a web server includes several parts that control how web user's access hosted files. At a minimum, this is an HTTP server. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device. [7]

A web server is what makes it possible to be able to access content like web pages or other data from anywhere as long as it is connected to the internet. The hardware houses the content, while the software makes the content accessible through the internet [8].

The most common use of web servers is to host websites but there are other uses like data storage or for running enterprise applications. There are also different ways to request content from a web server. The most common request is the Hypertext Transfer Protocol (HTTP), but there are also other requests like the Internet Message Access Protocol (IMAP) or the File Transfer Protocol (FTP). [9]

1.5.2. How does a web server work?

To publish a website, you need either a static or a dynamic web server:

A static web server: consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser. [10]

A dynamic web server: consists of a static web server plus extra software, most commonly an application server and a database. We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.[11]

At the most basic level possible, the following figure shows the steps that brings a web page to your screen:

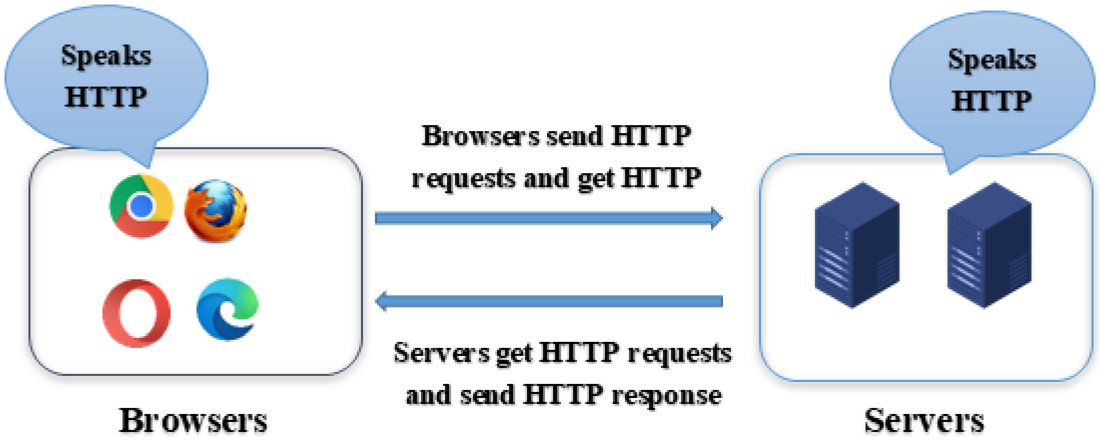

Speaks HTTP
Speaks HTTP
Browsers send HTTP requests and get HTTP
Servers get HTTP requests and send HTTP response
Browsers
Servers

Figure 8. Static Web server

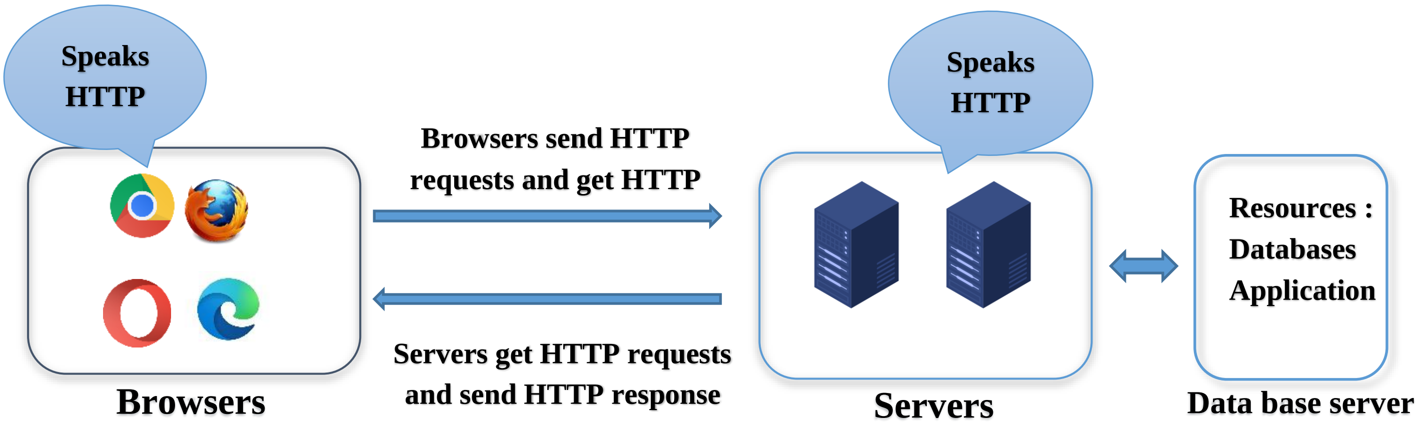

Speaks HTTP
Speaks HTTP
Browsers send HTTP requests and get HTTP
Servers get HTTP requests and send HTTP response
Resources :
Databases
Application
Browsers
Servers
Data base server

Figure 9. Dynamic Web server

1.5.3. A web server's loop of an HTTP request:

Since HTTP is layered on TCP/IP layer so a web server is going to loop forever doing as it is shown in the following figure:

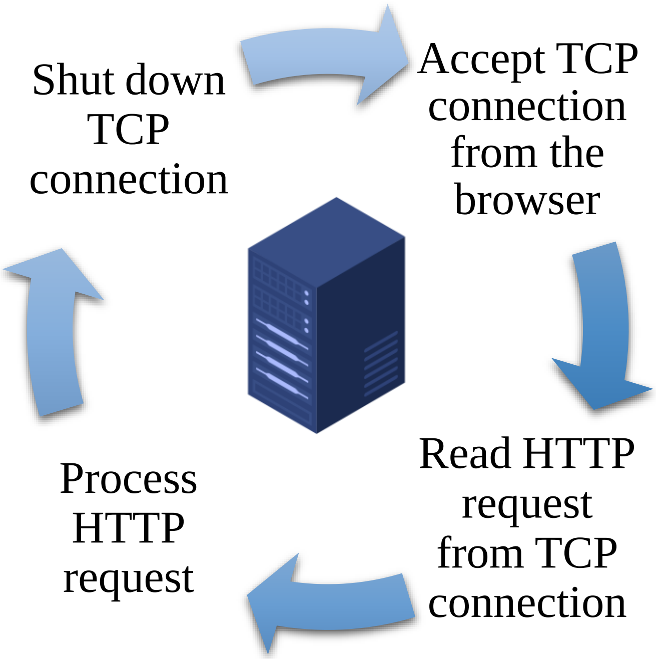

Accept TCP connection from the browser
Read HTTP request from TCP connection
Process HTTP request
Shut down TCP connection

Figure 10. A web server's loop of an HTTP request

1.6. HTTP Protocol:

Hypertext Transfer Protocol (HTTP) is a method for encoding and transporting information between a client (such as a web browser) and a web server. HTTP is the primary protocol for transmission of information across the Internet.

Information is exchanged between clients and servers in the form of Hypertext documents, from which HTTP gets its name. Hypertext is structured text that uses logical links, or hyperlinks, between nodes containing text. Hypertext documents can be manipulated using the Hypertext Markup Language (HTML). Using HTTP and HTML, clients can request different kinds of content (such as text, images, video, and application data) from web and application servers that host the content. https://www.nginx.com/resources/glossary/http/

1.7. Load limits:

A Web serve has defined load limits, because it can handle only a limited number of concurrent client connections (usually between 2 and 80,000, by default between 500 and 1,000) per IP address (and TCP port) and it can serve only a certain maximum number of requests per second depending on:

- its own settings
- the HTTP request type

- content origin (static or dynamic)
- the fact that the served content is or is not cached
- the hardware and software limitations of the OS where it is working

When a Web server is near to or over its limits, it becomes unresponsive.

1.8. Overload causes:

At any time web servers can be overloaded because of:

- **Too much legitimate web traffic:** thousands or even millions of clients connecting to the web site in a short interval, e.g., Slashdot effect;
- **Distributed Denial of Service:** attacks
- **Computer worms:** that sometimes cause abnormal traffic because of millions of infected computers (not coordinated among them);
- **Internet (network) slowdowns:** so that client requests are served more slowly and the number of connections increases so much that server limits are reached
- **Web servers (computers) partial unavailability:** this can happen because of required or urgent maintenance or upgrade, hardware or software failures, back-end (e.g., database) failures, etc.; in these cases the remaining web servers get too much traffic and become overloaded.

1.9. Example of web servers:

These are the most popular open source web servers that are available for everyone on the internet:

1.9.1. **Apache HTTP Server:** This is the most popular web server in the world developed by the Apache Software foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more.

1.9.2. **IIS Windows Web Server**: It has been developed by the software giant, Microsoft. It offers higher levels of performance and security than its predecessors. It also comes with a good support from the company and is the second most popular server on the web.

1.9.3. **The lighttpd**: Is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power.

1.9.4. **NGINX:** Free open source popular web server including IMAP/POP3 proxy server. Nginx is known for its high performance, stability, simple configuration and low resource usage.

1.9.5. **Apache Tomcat:** It has been developed to support servlets and JSP scripts. Though it can serve as a standalone server, Tomcat is generally used along with the popular Apache HTTP web server or any other web server.

Figure 11. Open source web servers

- **There are several webserver that we can classify by type, the following table shows the different types:**

TYPE	SERVICE
SHARED SERVERS	If you opt for a shared hosting offer, your site will be placed on a server managing several websites at the same time (perhaps a hundred, perhaps more). It is based on the fair sharing of resources, namely RAM memory, CPU, disk space and bandwidth. This is the cheapest offer.[1]
VIRTUAL SERVERS	The server only manages very few web sites (generally less than ten). This offer is generally suitable for sites which on the one hand can no longer fit on shared hosting because they have too much traffic (too many visitors), but which on the other hand cannot afford dedicated hosting.[1]
DEDICATED SERVERS	The server only manages your website and no others, meaning it is available to a single client by a service provider. However, it is quite expensive and it is better to have knowledge of Linux to administer the server remotely. [1]

Table 1 – Types of a web server

1.10. Microcontrollers:

Nowadays, microcontrollers are everywhere around us. For many years, doing something with them took a lot of knowledge that had to be accumulated over a fairly long period of time. Using microcontrollers was also relatively expensive, so most of the time they were used by trained professionals in various industries.

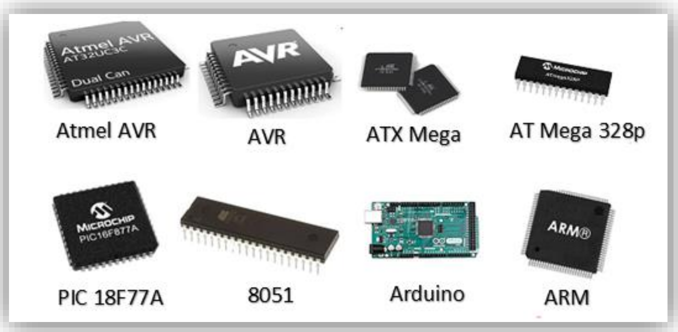


Figure 12. Common Microcontrollers

This changed a lot back in the 2005 when the Arduino board came out. The whole idea behind Arduino is that it can provide hobbyists around the world with a relatively inexpensive means for building their electronics projects. A lot could be said about the Arduino, and the microcontrollers in general, but this is a succinct book so we'll keep it short and to the point. Keep in mind that there are a lot of varieties of Arduino boards out there.

1.11. What is Arduino?

Arduino is an open source electronics creation platform, which is based on free, flexible and easy to use hardware and software for creators and developers. This platform allows you to create different types of single-board microcomputers to which the community of creators can give different types of use.

It is also capable of receiving and sending information over the internet with the help of various Arduino shields. Arduino uses a hardware known as the Arduino development board and software for developing the code known as the Arduino IDE (Integrated Development Environment). Built up with the 8-bit Atmel AVR microcontroller's that are manufactured by Atmel or a 32-bit Atmel ARM, these microcontrollers can be programmed easily using the C or C++ language in the Arduino IDE.

Figure 13. Arduino board

1.11.1. How does it work?

The Arduino is a board based on an ATMEL AVR microcontroller. Microcontrollers are integrated circuits where instructions can be recorded, which you write with the programming language that you can use in the Arduino IDE environment. These instructions allow you to create programs that interact with the circuitry on the board it has communication ports and input / output ports with which we can connect different types of peripherals on the board. The information of these peripherals that you connect will be transferred to the microcontroller, which will be in charge of processing the data that comes through them.

1.11.2. Why the Arduino?

Arduino simplifies the work with microcontrollers and offers the following advantages:

- Cheap
- Multiplatform
- Simple programming environment
- Free and extensible software through C ++ libraries
- Free and extensible hardware.

When working with Arduino, different technology concepts are handled with priori that have nothing to do with each other but unify them. For example digital and analog electronics, electricity, programming, microcontrollers, signal processing, and so on.

For someone who wants to do a project, the process is to download and install the IDE, search the internet a bit and simply “cut and paste” the code that interests us and upload it. Then make the corresponding wiring with the peripherals and we already have the software interacting with the Hardware. All this with a minimal economic investment: the cost of the Arduino and the peripherals.

Arduino is a project and not a specific model of board, which means that sharing its basic design you can find different types of boards.

1.11.3. Different types of Arduino boards:

Leonardo Board: The first developed board of an Arduino is Leonardo Board. This board uses one microcontroller. This board is very simple and cheapest board. This board handles USB directly.

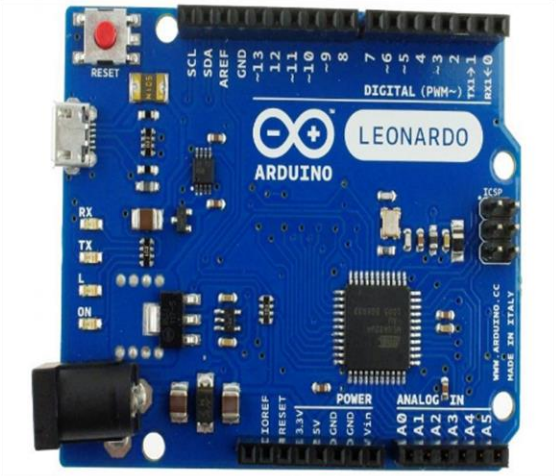


Figure 14. Leonardo Board

Red Arduino Board: this microcontroller board will work on window 8. This is very easy or simple to utilize in the project designing. this board can be programmed by using USB cable using Arduino IDE , Just plug the board, select the menu option to chosen an Arduino Uno and reading to upload the program. Board jack used for control the Red Board over USB cable.

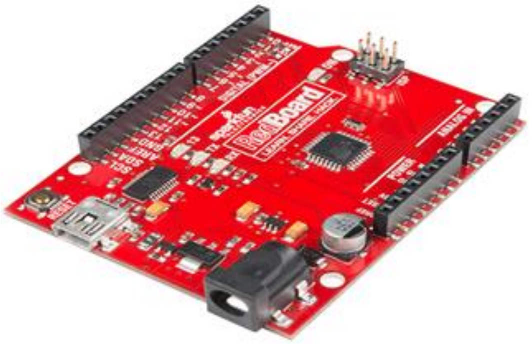


Figure 15. Red Arduino Board

Lily Pad Arduino Board: this board is an etextile technology which also comprise of I/O power and also sensor Board. These are even washable.

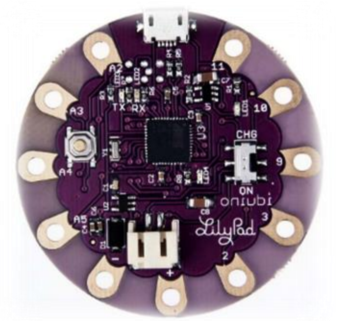


Figure 16. Lily Pad Arduino Board

Arduino Mega: is a modified form of Arduino Uno, Arduino Mega consist of lot of digital I/O pins. Among them14 pins can be used as PWM (pulse width modulation) outputs signal. 6 pins as analog input pins, a reset button, a power button, a power jack and USB connection. The huge number of pins makes the Arduino board very helpful for designing the project. It required holding up the microcontroller simple connected Arduino board with PC with the help of USB cable and giving the supply to get started with AC to DC battery.

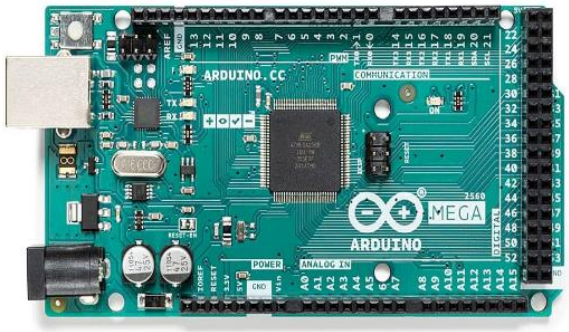


Figure 17. Arduino Mega Board

Arduino Uno: The Arduino Uno is consist of 14 digital I/O pins where 6pin can be used for Pulse Width Modulation and 6 analog input pins, a reset button, power jack, USB connection. It require to hold up the micro controller. With the help of USB cable Arduino is simply attach to PC and give the supply started with AC to DC battery.

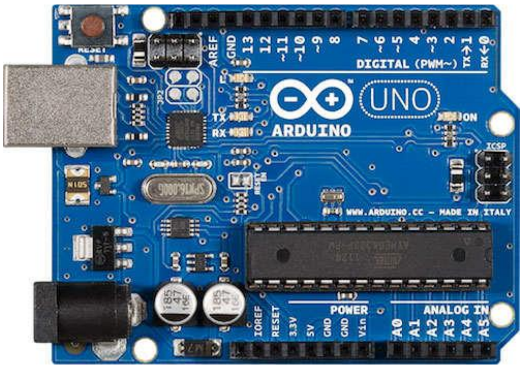


Figure 18. Arduino UNO Board

1.11.4. Why Arduino UNO?

We chose to work with the Arduino UNO board for different reasons, and the main reason is because we already possess a basic knowledge of it, since we have examined it profoundly in our 3rd year (license). In addition to that, this board is relatively inexpensive and doesn’t require any soldering or special power source to get you started. Basically, all you need is an Arduino Uno board, USB cable, and a computer.

1.11.5. Basic elements of Arduino UNO board:

Elements of Arduino UNO Board are divided into two different categories. These are: Hardware and Software

1.11.5.1. HARDWARE: Arduino board consists of many components that make Arduino board to work together (figure 19), we list them down below:

Microcontroller: microcontroller is the heart of Arduino board. Microcontroller is work as minicomputer which sending and receiving the information or command to device. The device is connecter to the Arduino board .microcontroller differ from board to board.

External Power Supply: this power supply is used to give power to board with voltage range from 9-12 Volts.

Analog Pins: analog pins ranging from A0 to A7. These pins are used for analog input / output pins. Board to Board the number of analog pins is differing.

Digital Pins: Arduino microcontroller consists of some digital pins also ranging from 2 to 16 pins. Board to board the number of digital pins is differing. These pins are used for the digital input/.output pins.

Power and Ground Pin: these pins provide 3.3, 5Volts and ground through them.

Reset Button: Reset button present on the board. This button used to reset the Arduino microcontroller.

USB Plug: USB Plug is most important part in the Arduino board .USB plug is used to upload program programming to the microcontroller or using USB cable. When the external power supply is absent then this plug provide regular power of 5 Volts.

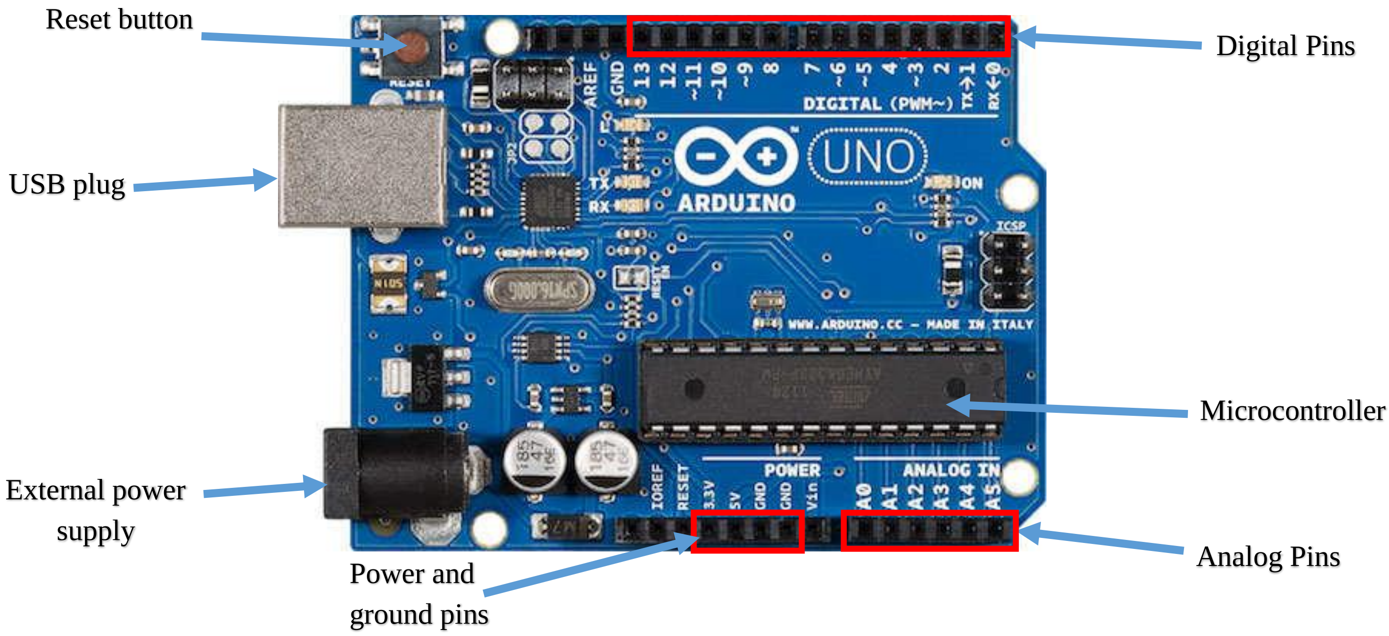


Figure 19. Arduino UNO components

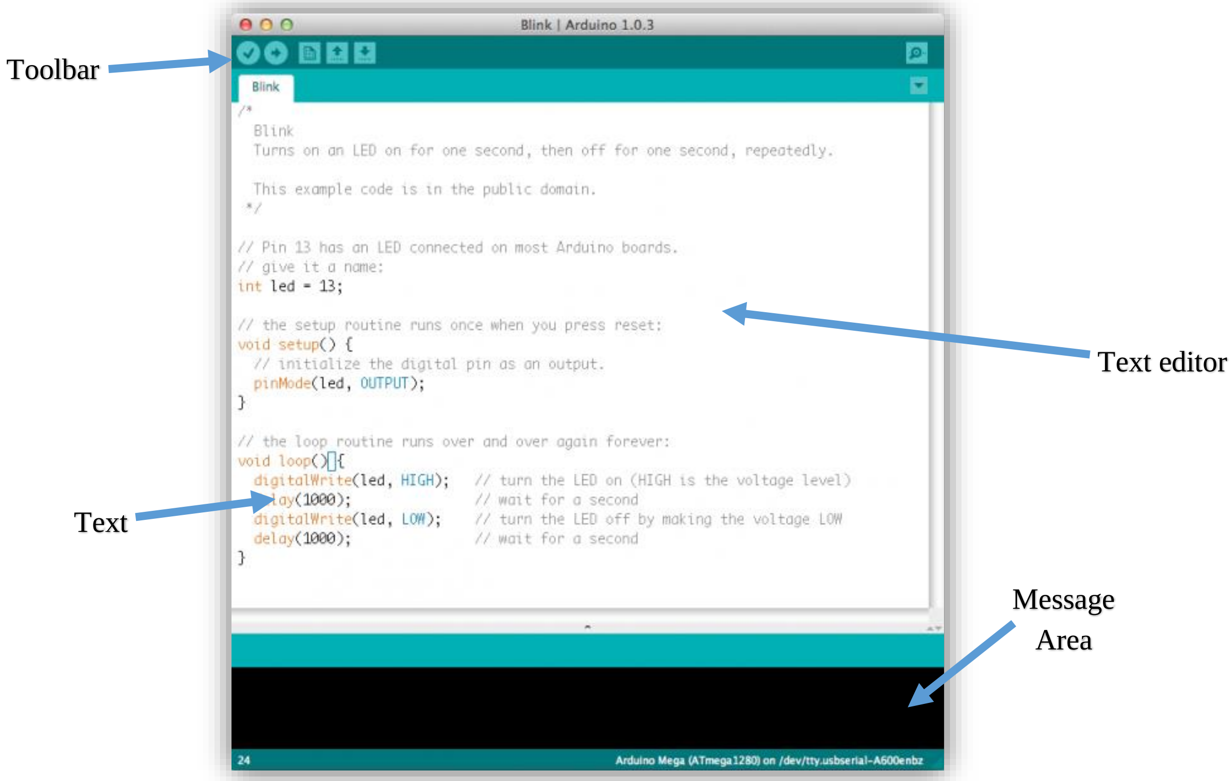
1.11.5.2. SOFTWARE: The program code written for Arduino is known as sketch. The software used for program coding is Arduino IDE software (figure 20). IDE software consists of:

Text Editor: Simplified code is written in simplified version of C++ programming language.

Message Area: - Display the error.
- Saving and exporting the code.

Text: Output by area Arduino including complete error message and other information.

Toolbar: verify, upload, new, open, save and serial monitor buttons are include in toolbar.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

24 Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz

Figure 20. Arduino Software

1.12. Types of Arduino shields:

Arduino boards are available with many different types of built-in modules in it. Boards such as Arduino BT come with a built-in Bluetooth module, for wireless communication. These built-in modules can also be available separately which can then be interfaced (mounted) to it. These modules are known as Shield. Some of the most commonly used Shields are listed in the table below:

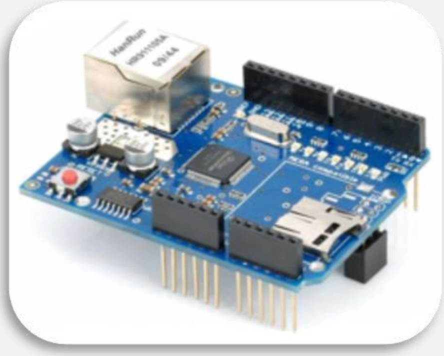
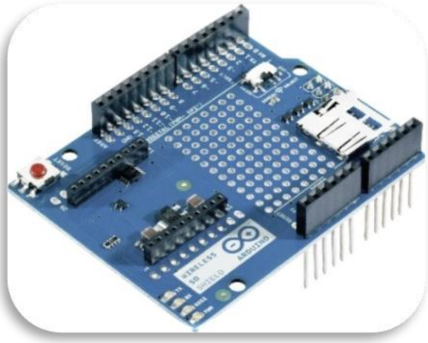
SHIELD	CHARACTERISTICS
ARDUINO ETHERNET SHIELD 	It that allows an Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library.[12]
ARDUINO WIRELESS SD SHIELD 	It allows your Arduino board to communicate wirelessly using ZigBee. It is an ideal choice when it comes to linking a radio module to your Arduino.[12]
ARDUINO MOTOR DRIVER SHIELD 	It allows your Arduino boards to interface with driver of a motor etc.[12]

Table 2 – Arduino Shields

1.13. VPN :

VPN (Virtual private network) is a software who protects your information by masking your device's IP address, encrypting your data and routing it through secure networks to servers in faraway states or even other countries.

1.14. Proxy :

A proxy server is a system or router that provides a gateway between users and the internet. Therefore, it helps prevent cyber attackers from entering a private network.

1.14.1. Why do we need a proxy server?

Privacy: allows us to surf the internet anonymously, because it hides our IP address.

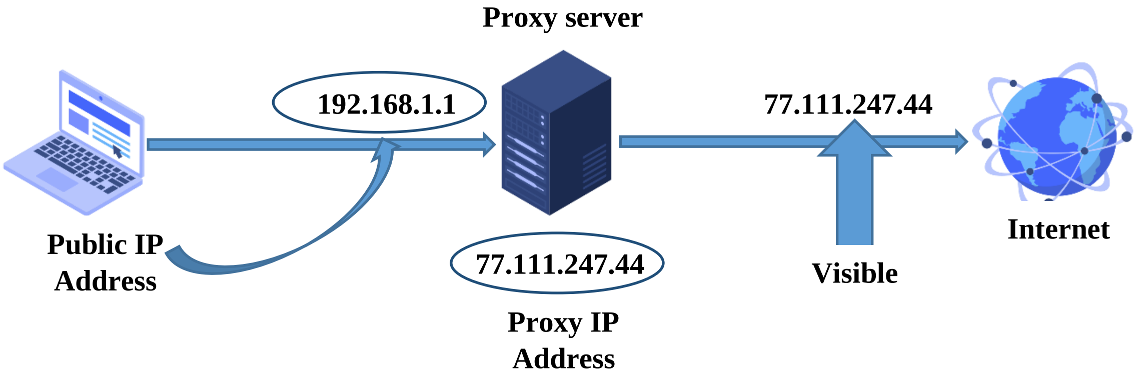


Figure 21. Proxy server

Speed: when a user wants to retrieve a webpage from the internet a proxy server will retrieve it on behalf of the user, and then it will store that page in a centralized cache data base.

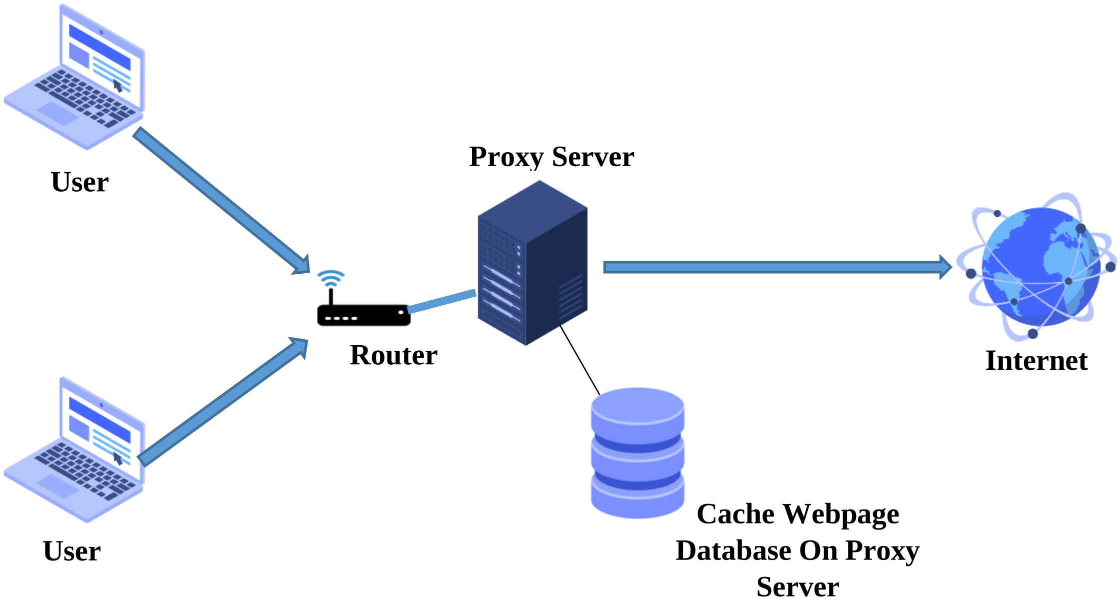


Figure 22. Proxy server

Bandwidth: a proxy server reduces the need to go out on the internet to retrieve data because it already has that data stored in its data base.

Activity Logging: a proxy server can keep track of what websites a user is looking at, because the proxy will keep a record of what websites the user is visiting and how long he was on the website. In addition to that, you can also configure the proxy to block certain pages to keep users from visiting them.

1.15. The difference between Proxy and VPN:

A VPN secures all your network traffic, while a Proxy works on an application level. They both hide your IP address, but only a VPN redirects your internet data through an encrypted tunnel.

1.16. Conclusion :

In this chapter we talked about servers in general and webservers in particular. We explained what the role of a webserver is and how it functions. In addition to that, we have shown the different types as well as the utilities of each one. After that, we addressed the Arduino microcontroller and showed the various kinds of this board. Then we explained how the Arduino UNO works and talked about both of the software and hardware side of it.

In the next chapter, we will discuss how to build an actual performing web server using an Arduino UNO board and go into further details about how to fix problems in case we face any.

Chapter II: Operating System

2.1. Introduction :

In this chapter we are going to discuss the creation and the development of our web server and go through all the steps, step by step in detail until we finally build a functioning web server. We are also going to develop a website, using different programs and programming languages, so that we can eventually host it in our server. The organizational chart below reviews the steps we will go through during the making of our web server.

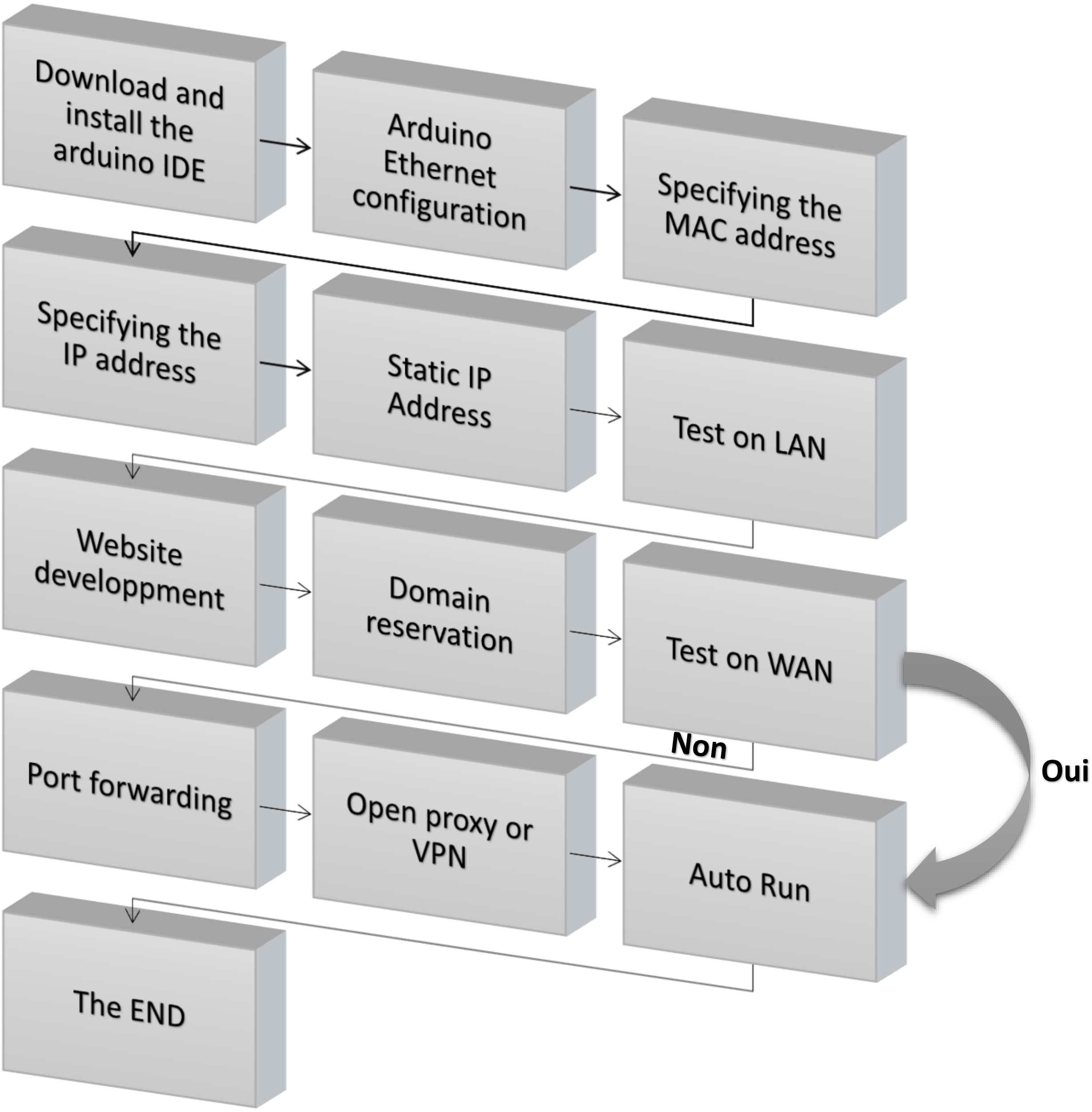


Figure23. Organizational Chart Of The Web Server

2.2. Arduino IDE :

The Arduino Integrated Development Environment or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus (as shown in the previous chapter (**figure 20**). It connects to the Arduino hardware to upload programs and communicate with them.

The open-source Arduino Software (Arduino IDE 1.8.19) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

2.2.1. Future Version of the Arduino IDE:

Arduino IDE 2.0 RC is the new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features auto completion, code navigation, and even a live debugger. [13]

2.2.2. Arduino Ethernet Shield configuration :

To control the Ethernet shield, you use the *Ethernet.h library*. The shield must be assigned a MAC and IP address using the *Ethernet.begin()* function. For a particular device, a MAC address is a globally unique identifier. Current Ethernet shields come with a sticker indicating the MAC address. For older shields, a random one should work, but one should not use the same one for many boards. Validity of IP addresses depends on the configuration of one’s network. If DHCP is used, it may dynamically assign an IP to the shield.

This example doesn't require an SD card. If an SD card is inserted but not used, it is possible for the sketch to hang, because pin 4 is used as SS (active low) of the SD and when not used, it is configured as INPUT by default. Two possible solutions:

- Remove the SD card;
- Add these lines of code in the setup()

```
pinMode(4, OUTPUT);
digitalWrite(4, HIGH);
```

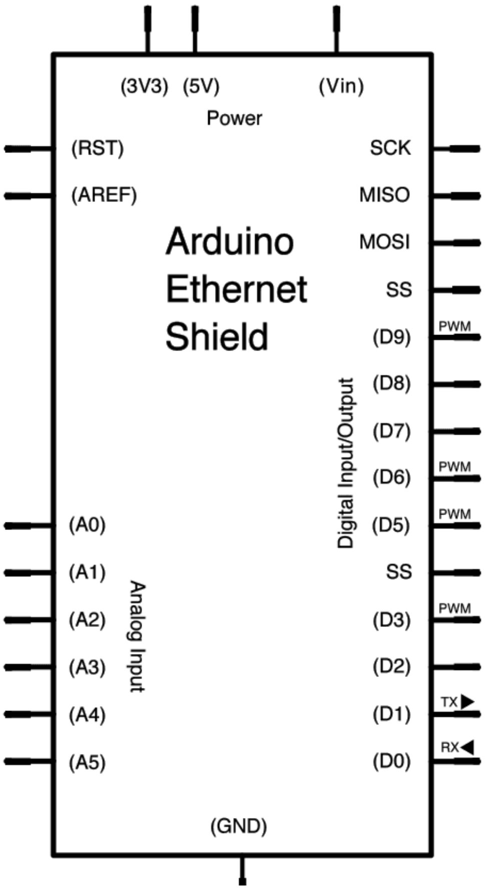


Figure 24. Schematic Of The Arduino Ethernet Shield

2.3. Specifying the MAC Address:

MAC address (media access control address) is a unique identifier assigned to each device participating in a physical network. Each piece of networking equipment has a unique serial number to identify itself over a network and this is normal hard-programmed into the equipment's firmware. However, with Arduino, we can define the MAC address ourselves. As for the Ethernet shield we will find the MAC address in the back of the board.

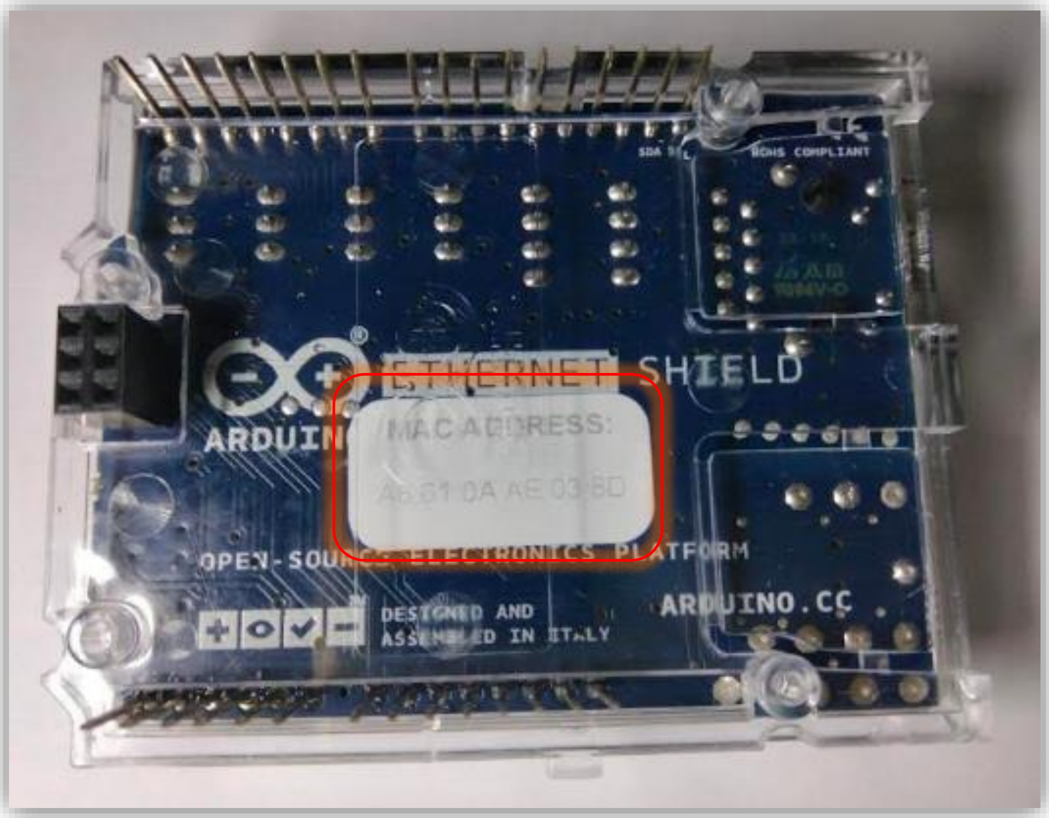


Figure 25. MAC address of the Ethernet Shield

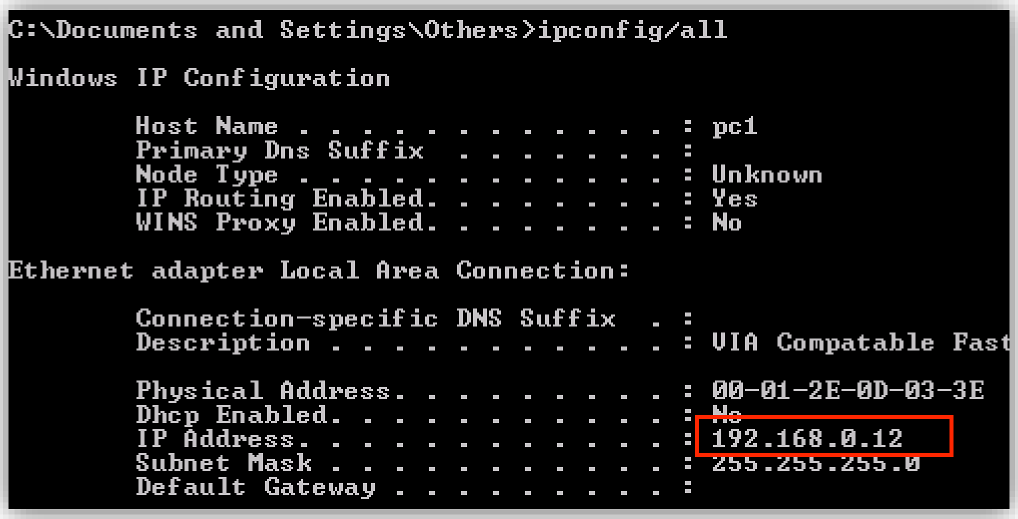
2.4. Specifying the IP Address:

IP address (Internet Protocol address) is a numerical label assigned to each device participating in a computer network that uses the Internet Protocol for communication. Specifying the IP address is done by writing the line *(IP example 192 169 0 112)*:

```
Byte ip[] = { 192, 168, 0, 112 };
```

And change it to match one own setup. For example, to assign the IP of Ethernet shield to 192.168.1.105, write the line:

```
Byte ip[] = { 192, 168, 1, 105 };
```



```
C:\Documents and Settings\Others>ipconfig/all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : pc1
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : Yes
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : VIA Compatable Fast

        Physical Address. . . . . . . . . : 00-01-2E-0D-03-3E
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.0.12
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

Figure 26. IPconfig Command

2.5. Static IP Address:

Our Arduino UNO board has a dynamic IP address by default, but this may cause a problem to us because we are going to have to change the program each time the address changes, as well the no-ip configuration (that we are going to see in the next chapter). That is why we decided to give a static IP address to the board in order to avoid any error.

2.6. Test on LAN:

In order to make a LAN (Local Area Network) test, we will type the IP address of our Arduino UNO in the URL bar of any browser of our choice. Then the page of our website should appear, if not then probably we have an Internet Service Provider that blocks port 80, the standard http port. To solve this problem, we will try one of the following ports 8080 or 8081. Then we have to find the line that reads: **Ethernetserver server(80);** in our webserver sketch. Now we will change 80 to 8080 or 8081. Finally upload. To get on our site, however we will need to type: http://*ArduinoIP@*:8080 or http://*ArduinoIP*:8081.

2.7. Development of the website:

Our web server is mainly created for hosting a website that we are going to develop using several programming languages such as HTML, CSS, and JAVASCRIPT to make our website functional and responsive.

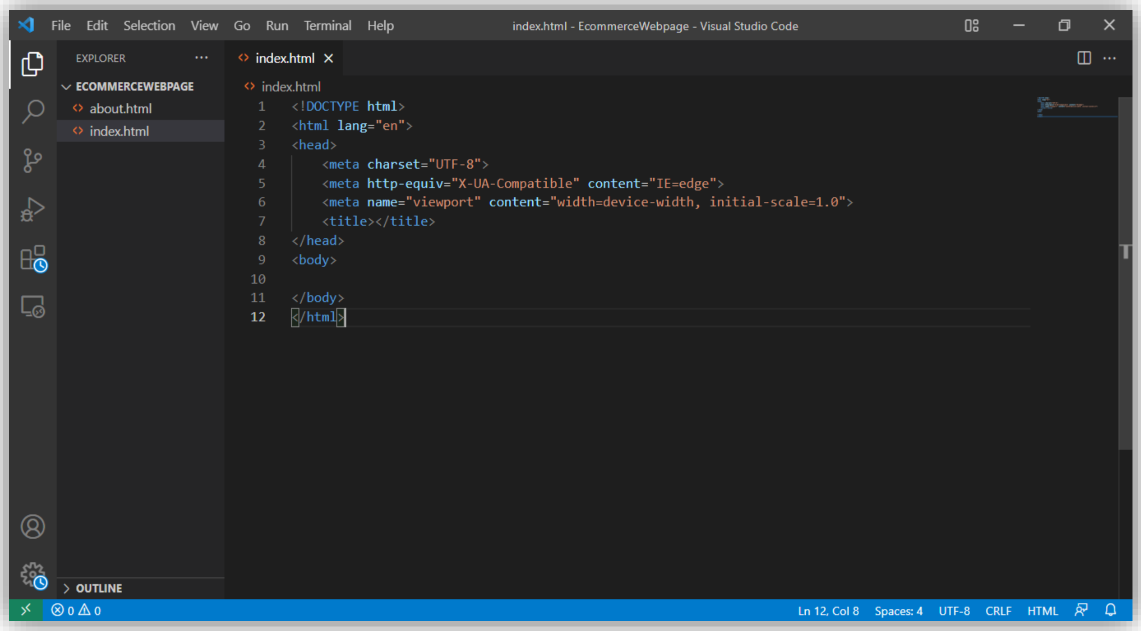
2.7.1. Website activation:

In order to place the website file in our Arduino UNO, we are going to need an SD card to copy all the files in it. But before that we have to format our SD card because the devices memory cards are installed on may use a variety of operating systems, such as Windows, Linux and so on. Sometimes unpredictable issues may occur during data read/write. To avoid such issues, we strongly suggest you format the memory card before using it for the first time on any new device. After that we can copy the file in the card and then place it in the Ethernet Shield. Finally we add the SD card library and add some changes in the program. By that the website activation is complete, meaning our website is stored in our server successfully.

2.7.2. Visual Studio:

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. **[14]**

Figure 27. Visual Studio Icon



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title></title>
</head>
<body>

</body>
</html>
```

Figure 28. Visual Studio Interface

2.7.3. Website theme:

We chose to create a weather app that will display information which it gets from an API (Application Programming Interface) and then shows it on our page. The user can also search for location by typing the name of any city in the search bar and then it will display the weather (description, temperature, humidity, wind) of the typed location.

2.8. Domain reservation:

Every website on the internet has a specific IP address that belongs only to that website. However, people unlike machines cannot memorize all those IP addresses to access to their desired website. That is why we have to give a domain name to each one to make it possible to remember. And for that we are using www.no-ip.com to replace our dynamic IP address.

2.9. No IP:

The No-IP Free Dynamic DNS service takes your dynamic IP address and makes it act as though it is static by pointing a static hostname to it and checking every 5 minutes for changes to your IP address. If your IP address changes, our Dynamic Update Client updates your hostname with the current IP address. [15]

2.10. Test on WAN:

For The WAN (Wide Area Network) test, we will open our browser then type in the URL bar the domain name. Our website should appear, if it doesn't then we will have to use port forwarding to forward a port number.

Do not use port numbers below 1024 other than 389 or 636 for directory services as they will conflict with other services. Additionally, port numbers below 1024 are accessible by root only. Make sure the ports you choose are not already in use. [16]

2.11. Port Forwarding:

2.11.1. What is port forwarding?

Port forwarding allows you to communicate with devices behind a router (or firewall) by instructing the router to forward traffic on certain ports to specific IP address on its LAN. This is often necessary when configuring equipment at remote transmitter locations to communicate with studio equipment, or to allow access from other locations. [17]

The definition of the different terms we used and will use are shown in the table below:

TERM	DEFINITION
LAN	Local Area Network. A computer network comprising systems in a single location that do not require a router to communicate with each other
WAN	Wide Area Network. A computer network that uses routers to link multiple networks together.
PORT NUMBER	A port number is used to identify the service requested by an IP connection. When a connection is established on a specific port, the device will provide the appropriate service to the connecting computer. Port numbers range from 0 to 65,535.
TCP	Transmission Control Protocol. TCP is used for reliable communications over an IP link.
UDP	User Datagram Protocol. UDP is typically used for streaming data where timeliness is more important than reliability
HTTP	Hypertext Transfer Protocol. This is the protocol used by web servers. HTTP typically uses TCP port 80
WELL-KNOWN PORTS	Ports 0 – 1023 are considered to be "well-known" ports. These ports are reserved for common services, including HTTP, SMTP, telnet, etc.

Table3. Term and Definition

2.11.2. TCP UDP ports:

TCP ports and UDP ports use the same range of numbers (0 – 65,535). However, these ports are typically defined (and forwarded) separately. Any given port number may be used for TCP and UDP simultaneously. Depending on your router's capabilities, a single port number may be forwarded to different IP addresses for TCP and UDP. [18]

2.11.3. Why we need port forwarding?

Port forwarding is critical for remote access to items on private networks. Since firewalls exist to keep unwanted visitors out, the visitors you want to get in are going to need a way to do so. Knowing the IP address isn't enough: Requests need to be directed to the correct port as well. This extra required information helps keeps unwanted visitors out and adds a further layer of security against DDoS (direct denial of service) attacks. [19]

Sometimes an ISP (Internet Service Provider) blocks a port such as port 80 for residential internet service so you cannot run your own web server without paying extra for a business account that would open port 80, 443, 21, 22, 23, 53, etc.

After making some port tests we will find that the problem is due to the IP address provided by Algérie Télécom which is probably shared, and blocks port 80.

2.11.4. Port 80:

Port 80 is the port number assigned to commonly used internet communication protocol, Hypertext Transfer Protocol (HTTP). It is the default network port used to send and receive unencrypted web pages.

2.11.5. How do you set up a port forwarding?

First we type our router's IP address in the URL bar of our browser, then log into it and go to the port forwarding section. Here it depends on your router because each router brand is slightly different from the other so you will need to look in your router's settings for the port forwarding option. Next, enter the port numbers and your device's IP address. Finally choose a forwarding protocol and save your changes.

2.11.6. Why port 8080?

Port 8080 is an alternative to port 80 and is used primarily for http traffic. It is commonly used for web servers and proxy servers, it can also run a Web server as a nonroot user.

2.12. Conclusion:

In this chapter we have discussed the steps that we have to go through in order to create a responsive web server. We talked about the issues and errors we might encounter during the programming and configuration of the server and gave a solution to each and every one of them. In the next chapter we will set the theoretical part into practice. Also we will illustrate our work and show more details of each step.

Chapter III: Results and Discussion

3.1. Introduction:

In this chapter we will go into further details about our practical realization of the webserver. We will begin by configuring our Arduino UNO and Ethernet shield, then show the results of the different programs and tests that we have done on both LAN (Local Area Network) and WAN (Wide Area Network) as well as solving all the possible problems. The next diagram shows the steps we are going to follow.

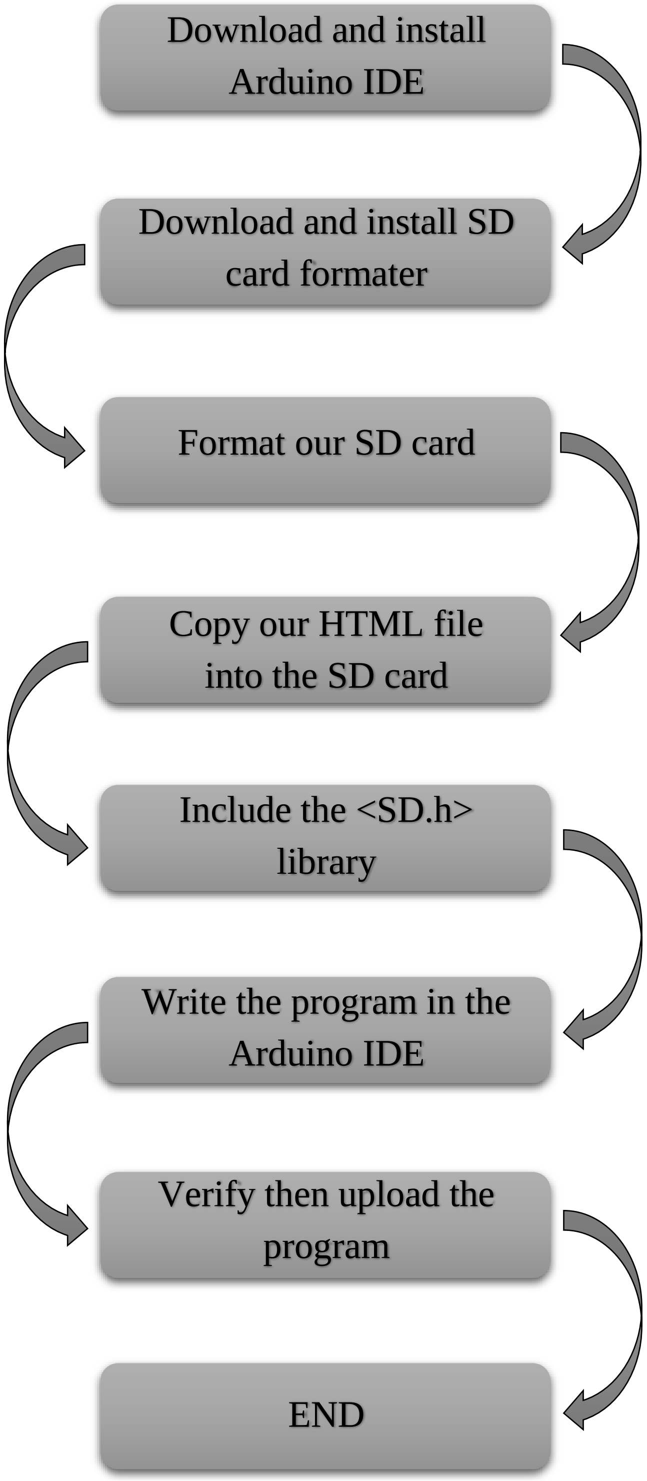

Download and install Arduino IDE
Download and install SD card formater
Format our SD card
Copy our HTML file into the SD card
Include the <SD.h> library
Write the program in the Arduino IDE
Verify then upload the program
END

Figure.29 Organizational chart of our system

3.2. Equipment:

Hardware	Software
Arduino Uno	Arduino IDE
Ethernet shield	Visual Studio
SD card	Port Forwarding
PC	
USB cable	
Ethernet Cable	
WIFI Router	

Table4. Equipment

3.3. Installation of the Arduino IDE:

Step1:

In order to be able to upload programs that we have on our Arduino IDE, first we are going to need an Arduino board (*in our case we are using the Arduino UNO board*). In addition to that we must have an Ethernet Shield and some other necessary equipment to build our webserver. The following figure explains in further detail what we require.

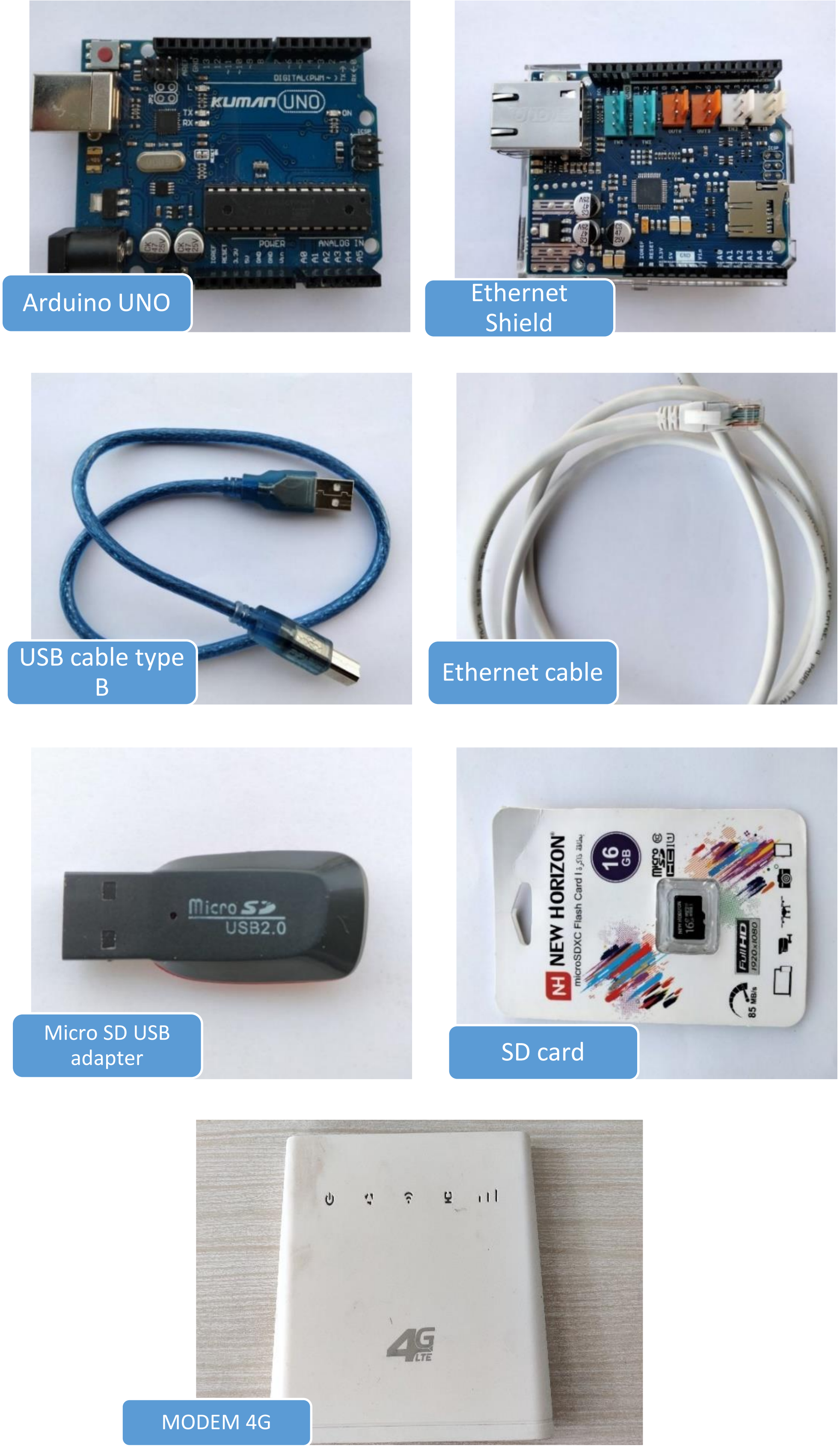
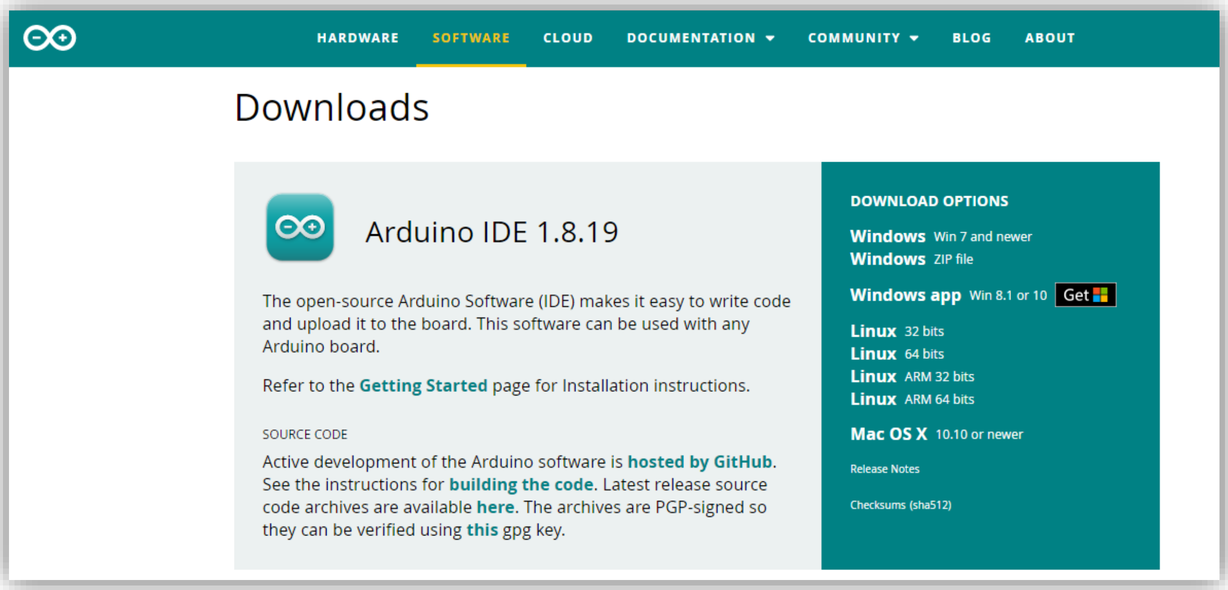


Figure30. Equipment

Step2:

To download the Arduino IDE files we need to go to www.arduino.cc/en/software on the Arduino official website. First, we must select our software, which is compatible with our operating system (Windows, IOS, or Linux). After the file finished downloading, unzip the files.



Downloads

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

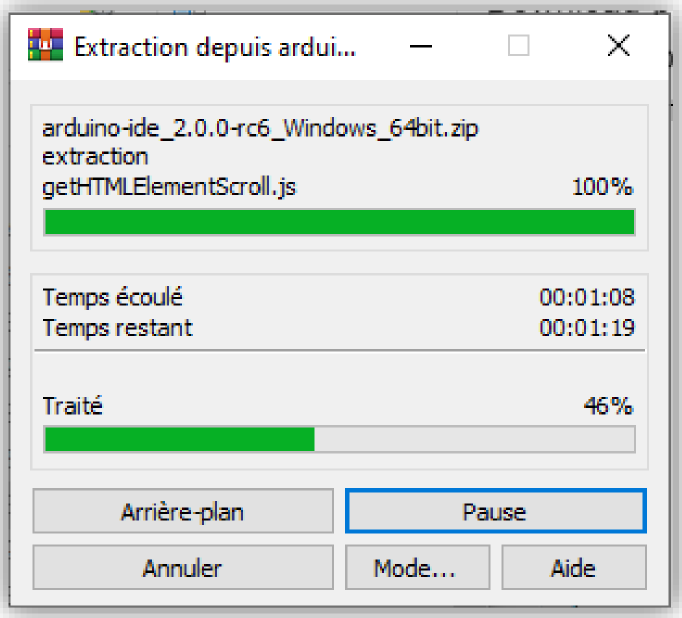
Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer
Release Notes
Checksums (sha512)

Figure31. Arduino IDE download page

We unzip the files using WinRAR:


Extraction depuis ardui...

arduino-ide_2.0.0-rc6_Windows_64bit.zip
extraction
getHTMLElementScroll.js 100%

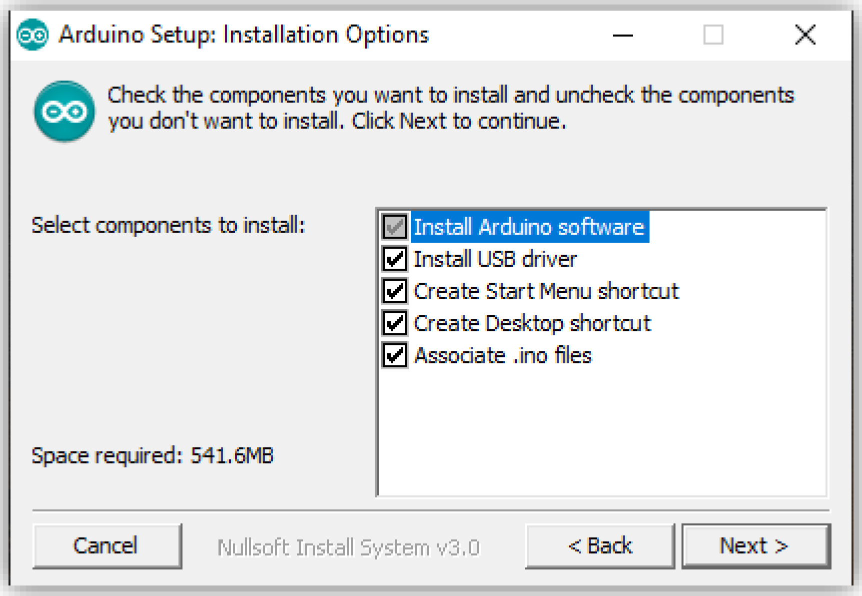
Temps écoulé 00:01:08
Temps restant 00:01:19

Traité 46%

Figure32. Unzipping files

Step3:

After unzipping the folder, we go to the Arduino Setup and select the components we need to install and click next, then select which folder we want the Arduino software to be in. And finally click install.



Arduino Setup: Installation Options

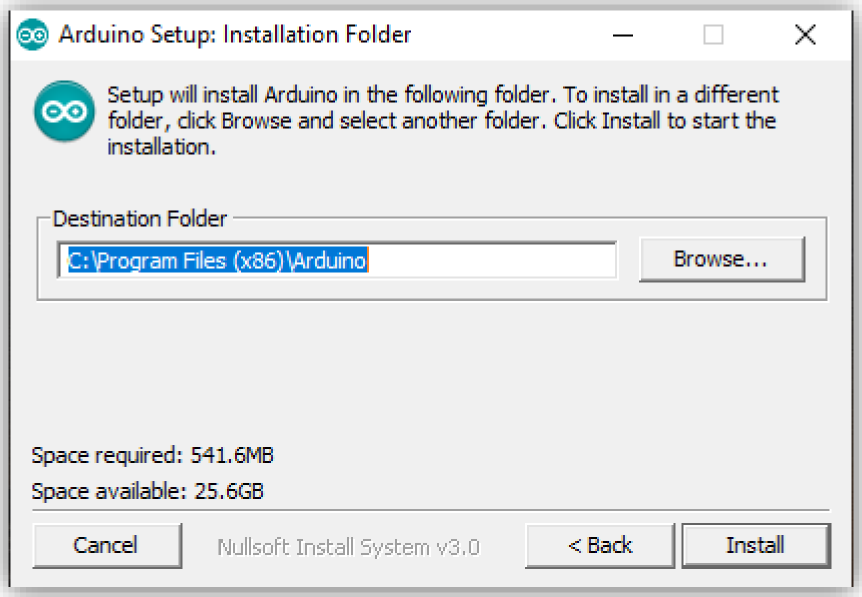
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Install Arduino software
☑ Install USB driver
☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Associate .ino files

Space required: 541.6MB

Figure33. Select components



Arduino Setup: Installation Folder

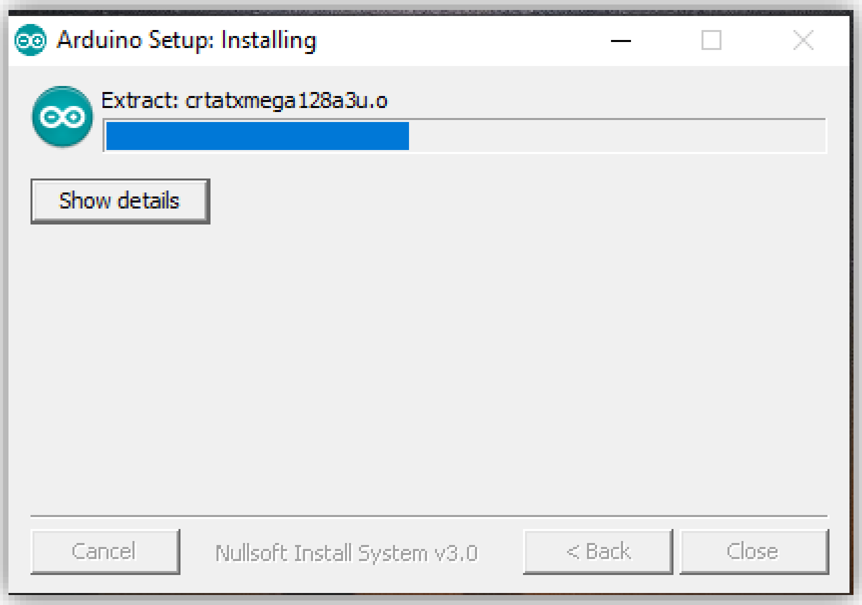
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino

Space required: 541.6MB
Space available: 25.6GB

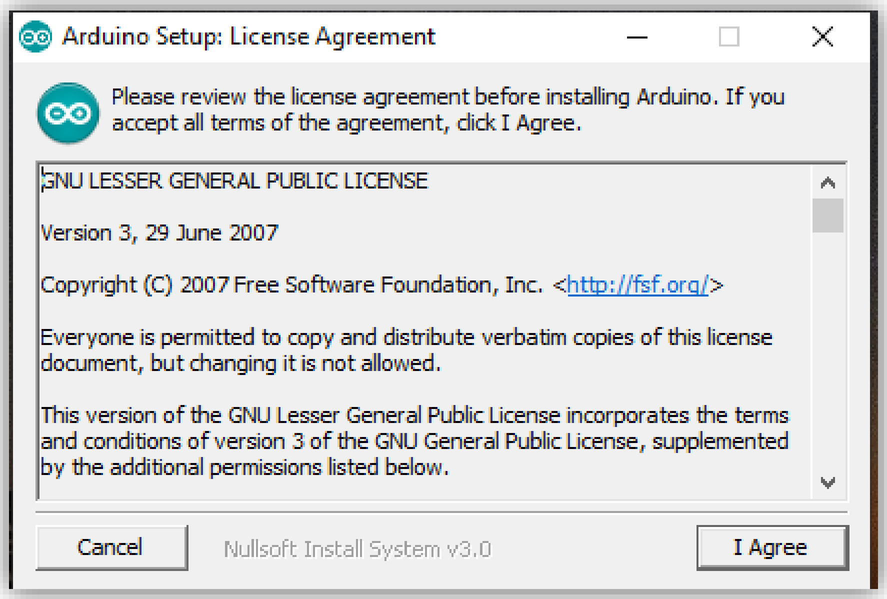
Figure34. Folder destination

We wait few minutes until the software is installed then we agree to the terms and conditions of the license agreement.


Arduino Setup: Installing

Extract: crtatxmega128a3u.o

Figure35. Installation


Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

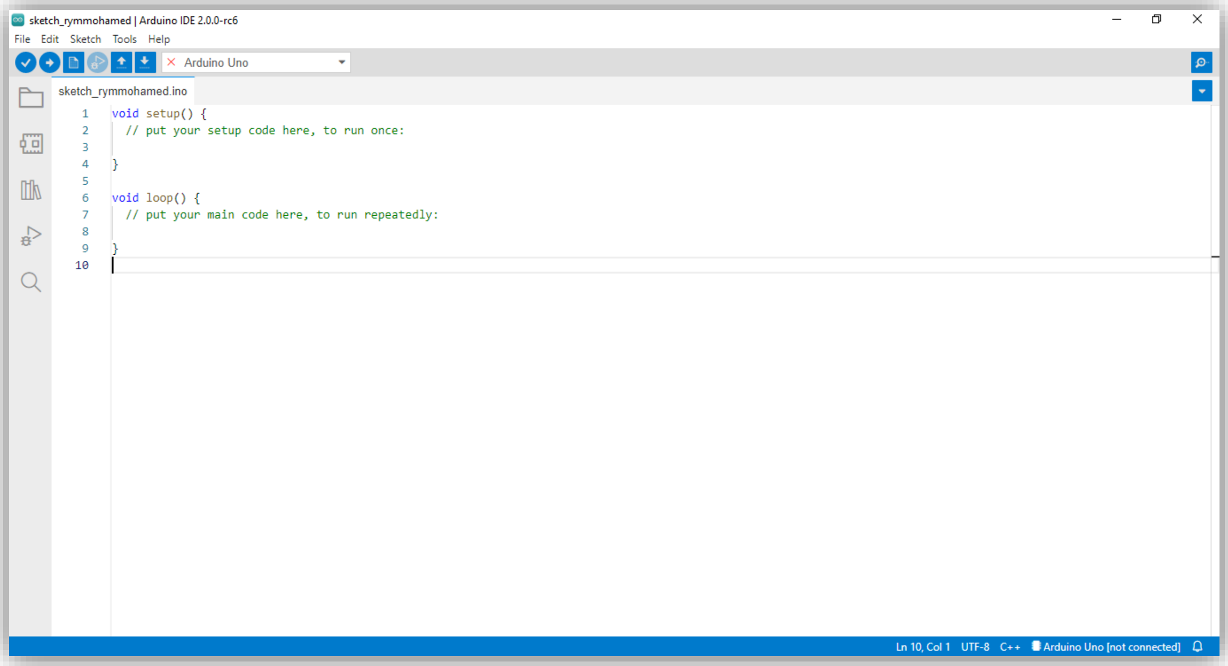
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Figure36. License agreement

Step4:

Now we launch our software. Once it starts we have two options: either create a new project or open an existing project example.


sketch_rymmohamed | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno
sketch_rymmohamed.ino
Ln 10, Col 1 UTF-8 C++ Arduino Uno [not connected]

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure37. Arduino IDE interface

In our case we are going to open an existing project example to test on our LAN. We select file, examples, Ethernet, and finally select WebServer.

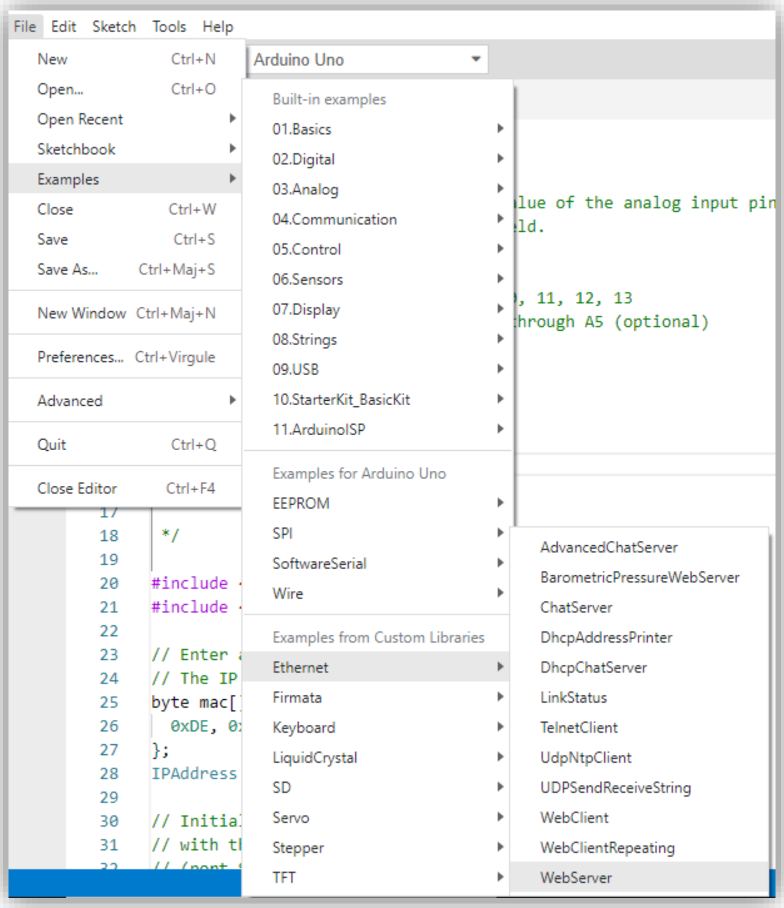
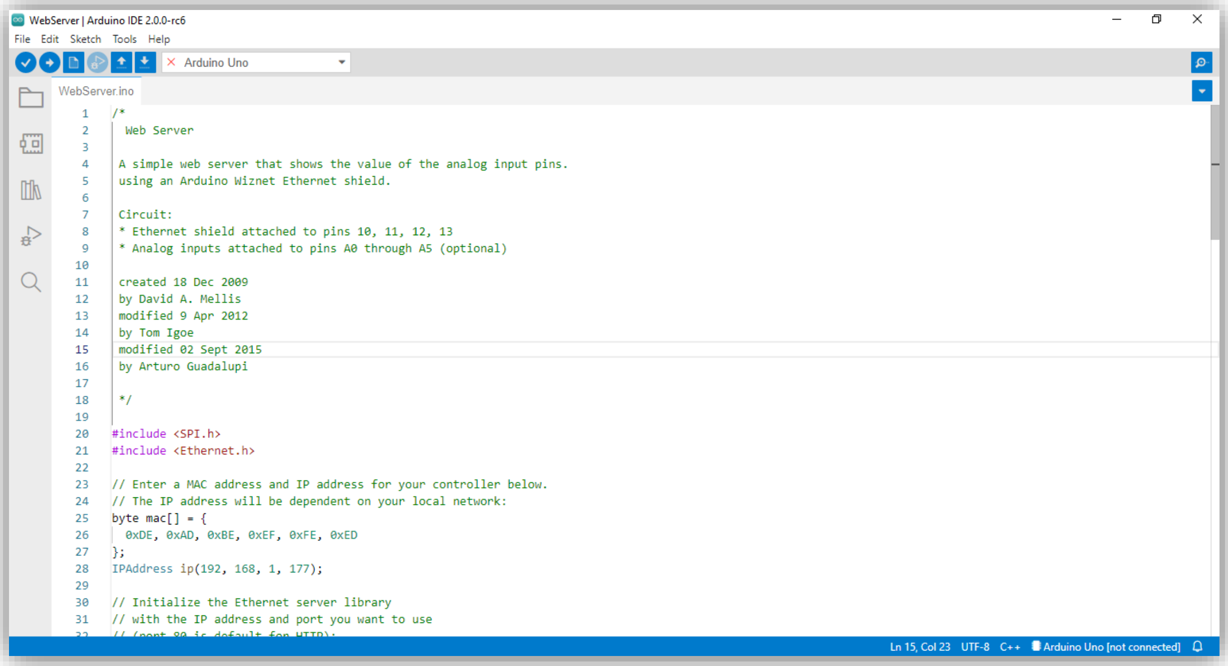


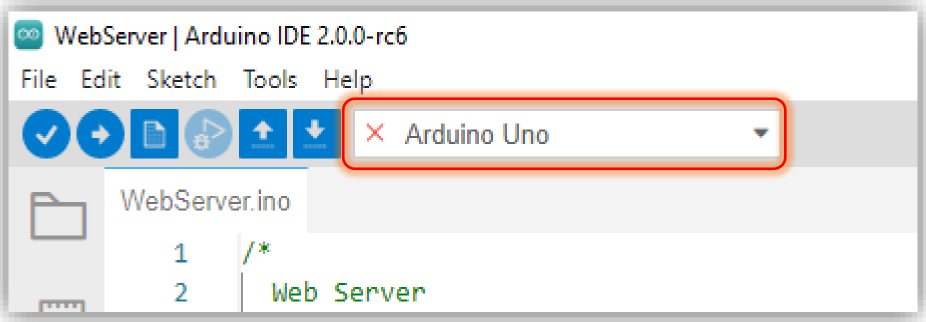
Figure38. Selecting the WebServer example


WebServer | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno
WebServer.ino
Ln 15, Col 23 UTF-8 C++ Arduino Uno [not connected]

```
/*
 Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:
* Ethernet shield attached to pins 10, 11, 12, 13
* Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe
modified 02 Sept 2015
by Arturo Guadalupi

*/

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
```

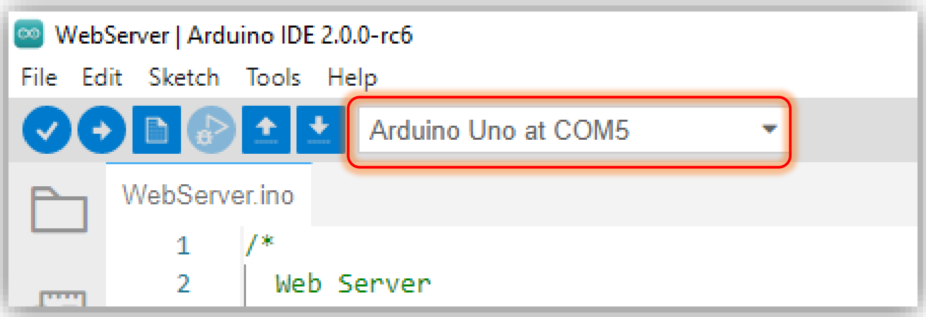
Figure39. WebServer Sketch Example

We notice that our Arduino board is not selected because it is not connected yet. Once we connect it, it appears like this:


WebServer | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno
WebServer.ino

```
/*
 Web Server
```

Figure40. Arduino UNO not connected


WebServer | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno at COM5
WebServer.ino

```
/*
 Web Server
```

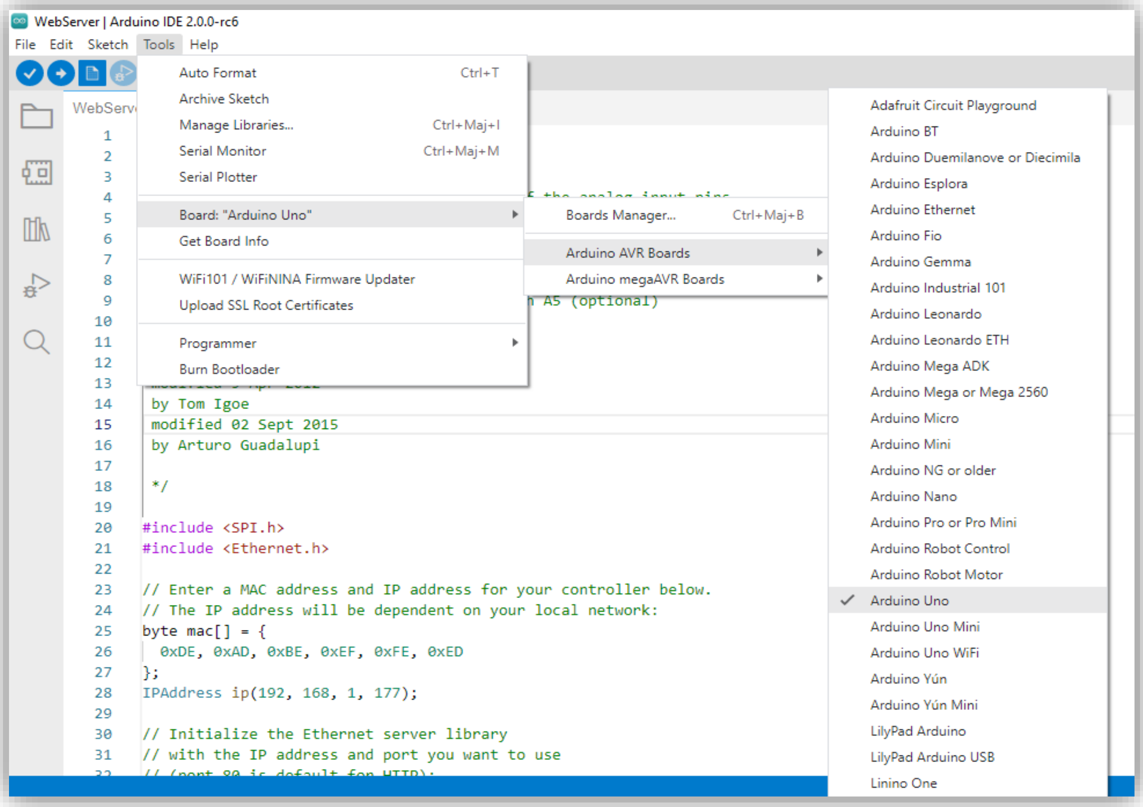
Figure41. Arduino UNO connected

Now we have successfully connected our Arduino UNO board and configured our port.

3.4. Arduino UNO board configuration:

Step5:

We need to select our Arduino UNO to avoid any error while uploading our program to the board. We must select the correct Arduino board name, which matches with the one connected to our computer.

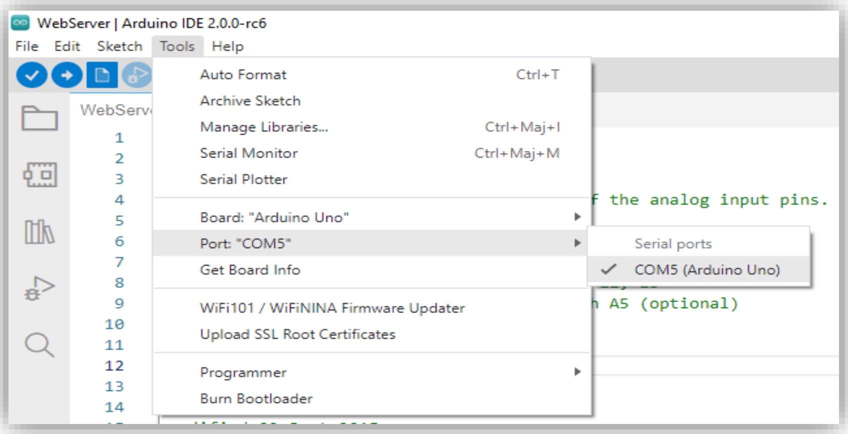


```
f the analog input pins
n A5 (optional)
by Tom Igoe
modified 02 Sept 2015
by Arturo Guadalupi

*/

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
```

Figure42. Selecting Arduino UNO board

Step6:

Next we select the right port. To do so, we go to Tools then Serial port menu. To make sure you selected the right port you can disconnect the Arduino board and re-open the menu, the entry that disappears should be of the Arduino UNO. Reconnect the board and select the serial port. In our case it is at Port: "COM5".



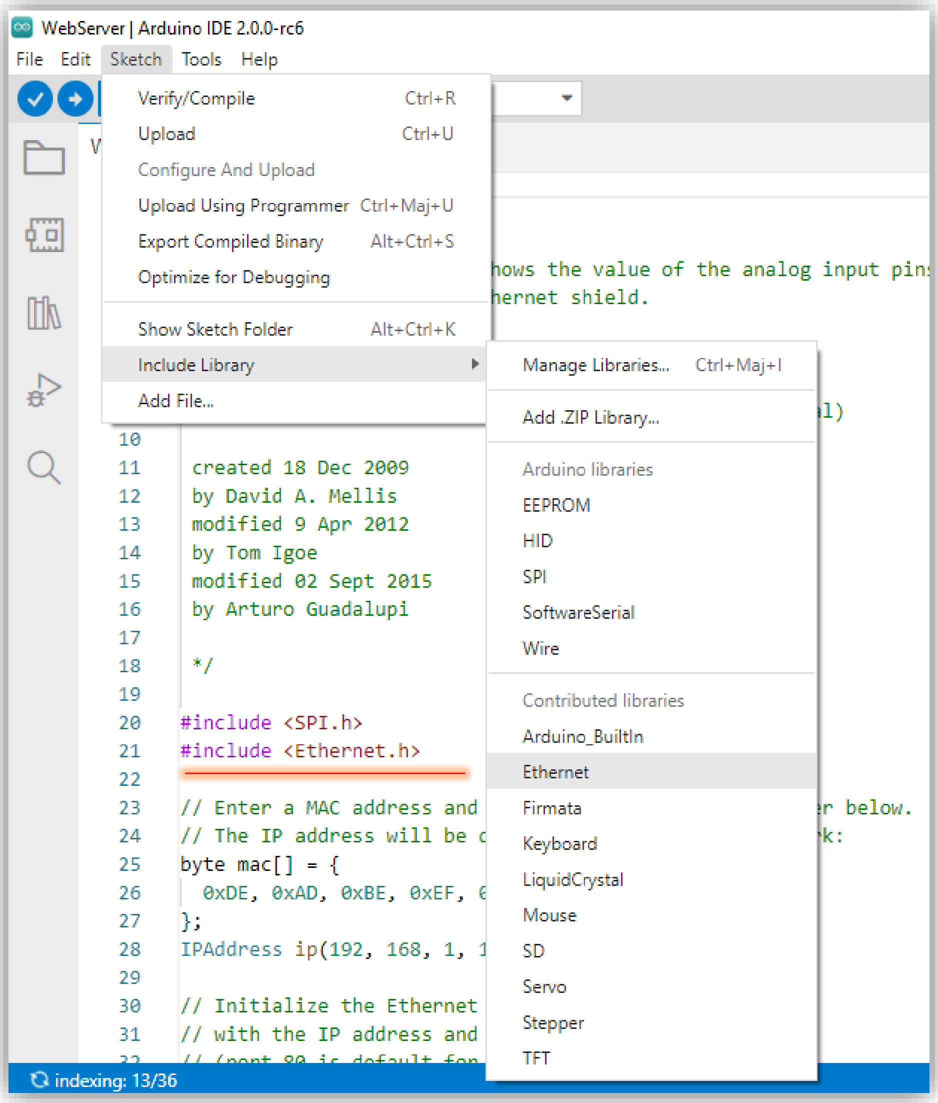
```
f the analog input pins.
h A5 (optional)
```

Figure43. Selecting port COM5

3.5. Ethernet Shield configuration:

Step7:

In this step we have to add the Ethernet shield library to our program. After that we need to change the MAC address with the one of the Ethernet shield as well as the IP address.

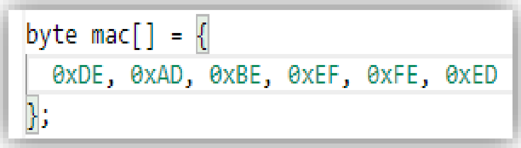


```
hows the value of the analog input pin
hernet shield.

l)

 created 18 Dec 2009
 by David A. Mellis
 modified 9 Apr 2012
 by Tom Igoe
 modified 02 Sept 2015
 by Arturo Guadalupi

 */

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and                    r below.
// The IP address will be c                   k:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0
};
IPAddress ip(192, 168, 1, 1

// Initialize the Ethernet
// with the IP address and
// (port 80 is default for
```

Figure44. Including Ethernet Shield library

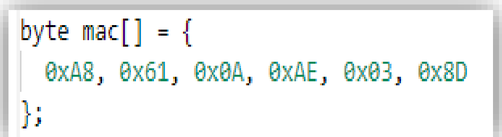
3.6. Specifying the MAC address:

We turn our Ethernet shield on its back and write down the MAC address that’s on it.



```
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
```

Figure45. MAC Address by default

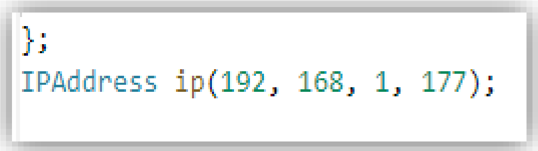


```
byte mac[] = {
  0xA8, 0x61, 0x0A, 0xAE, 0x03, 0x8D
};
```

Figure46. Ethernet Shield MAC Address

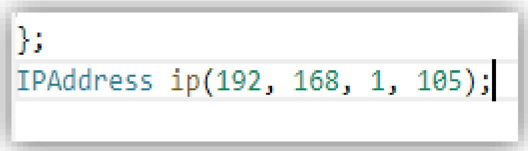
3.7. Specifying the IP address:

In our browser we type the address: 192.168.1.1 (*public IP address of our router*) then we go to Interface Setup, LAN and then scroll down to find a DHCP table. We copy the Arduino UNO IP address and paste it in our program.



```
};
IPAddress ip(192, 168, 1, 177);
```

Figure47. IP Address by default

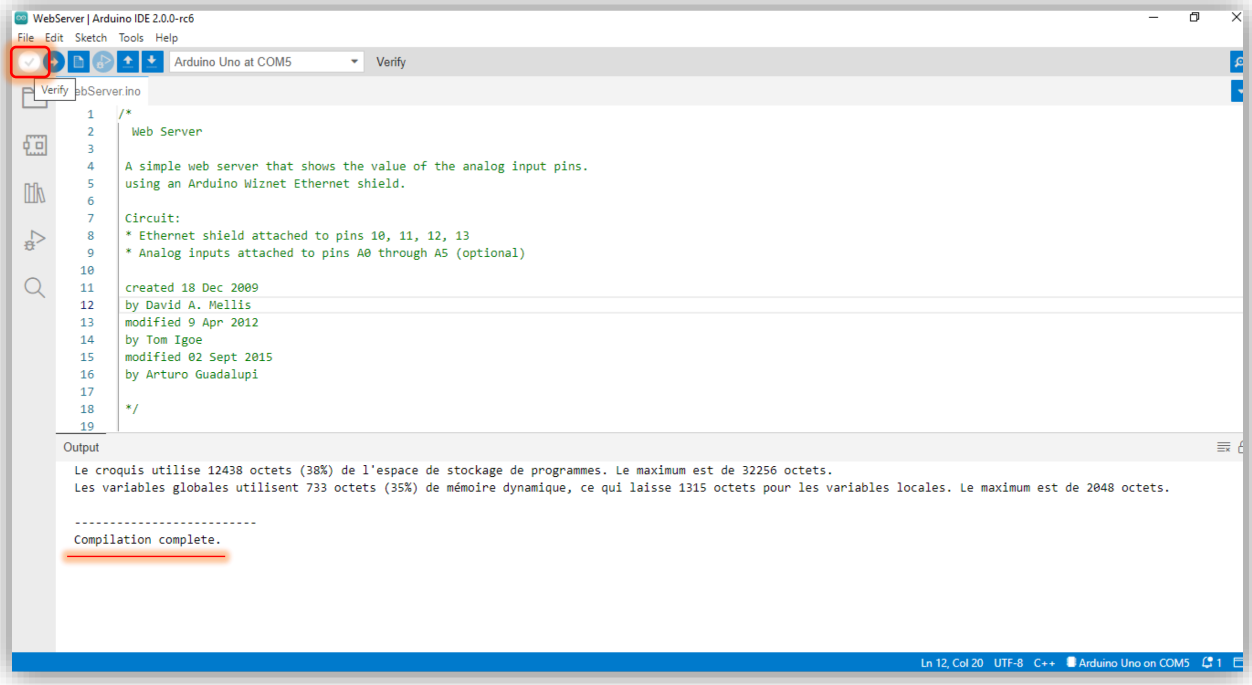


```
};
IPAddress ip(192, 168, 1, 105);
```

Figure48. Arduino UNO IP address

Step8:

Finally we upload the program to our Arduino UNO board. But before we need to verify our program.

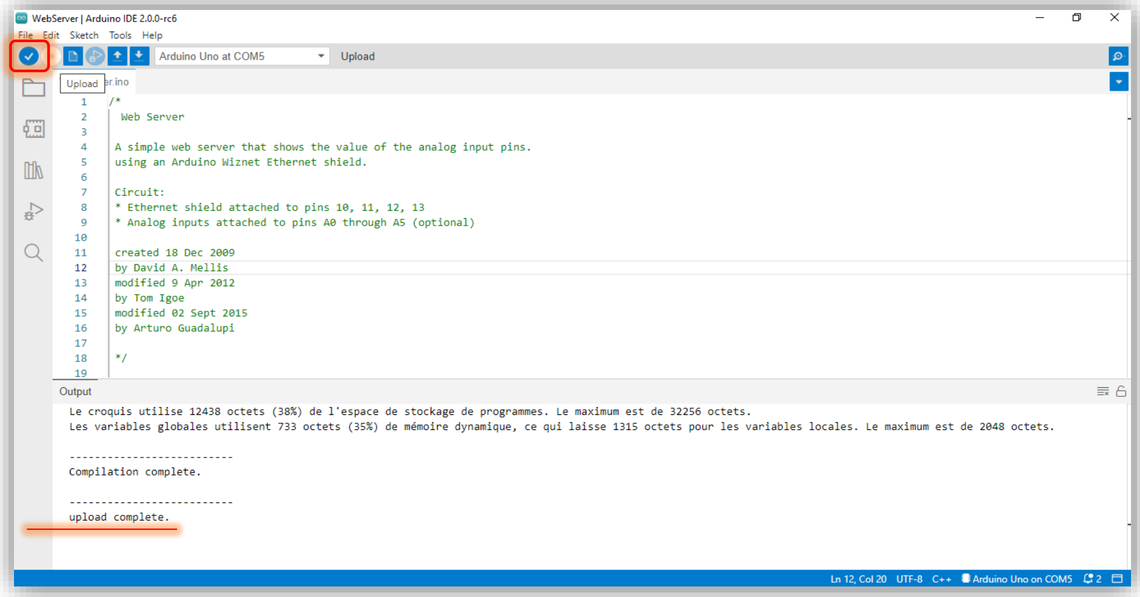


WebServer | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno at COM5
Verify
Verify
ebServer.ino
Output
Ln 12, Col 20 UTF-8 C++ Arduino Uno on COM5 1

```
/*
 Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:
* Ethernet shield attached to pins 10, 11, 12, 13
* Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe
modified 02 Sept 2015
by Arturo Guadalupi

*/
```

```
Le croquis utilise 12438 octets (38%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 733 octets (35%) de mémoire dynamique, ce qui laisse 1315 octets pour les variables locales. Le maximum est de 2048 octets.

--------------------------
Compilation complete.
```

Figure49. Compilation



WebServer | Arduino IDE 2.0.0-rc6
File Edit Sketch Tools Help
Arduino Uno at COM5
Upload
Upload
er.ino
Output
Ln 12, Col 20 UTF-8 C++ Arduino Uno on COM5 2

```
/*
 Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:
* Ethernet shield attached to pins 10, 11, 12, 13
* Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe
modified 02 Sept 2015
by Arturo Guadalupi

*/
```

```
Le croquis utilise 12438 octets (38%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 733 octets (35%) de mémoire dynamique, ce qui laisse 1315 octets pour les variables locales. Le maximum est de 2048 octets.

--------------------------
Compilation complete.

--------------------------
upload complete.
```

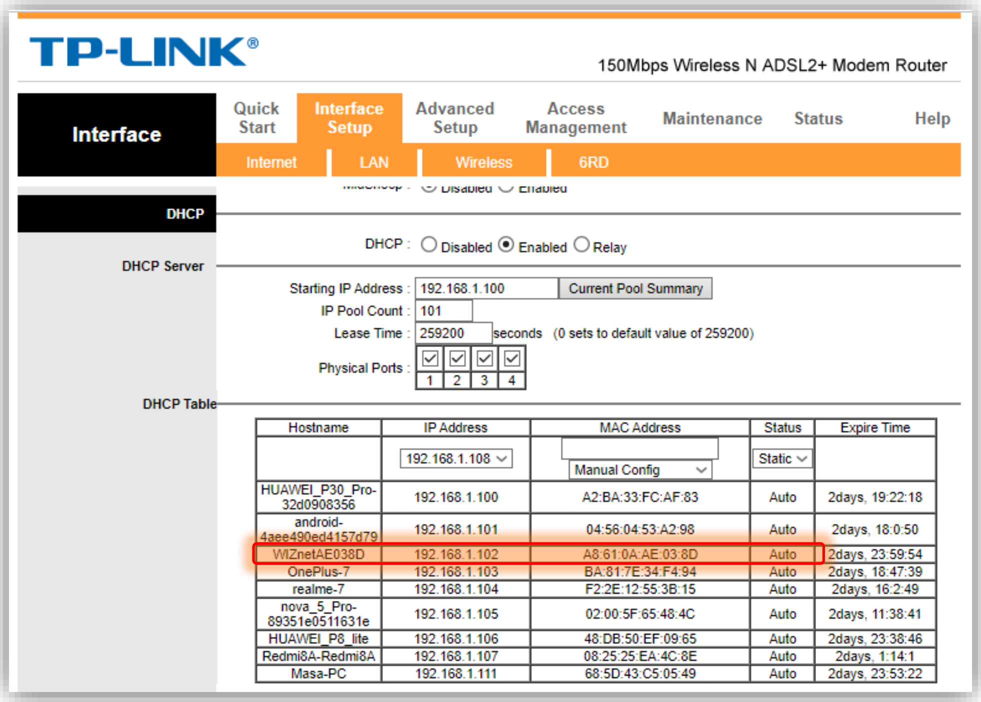
Figure50.Upload

Note:

If there was an error while uploading the program, then you should change the port you are using and it will be fixed.

3.8. Static IP address:

The IP address of our Arduino UNO is dynamic by default, and that will cause a problem with the program every time it changes so in order to avoid that we will give the board a static IP address.



DHCP : ○ Disabled ● Enabled ○ Relay

Starting IP Address : 192.168.1.100 Current Pool Summary

IP Pool Count : 101

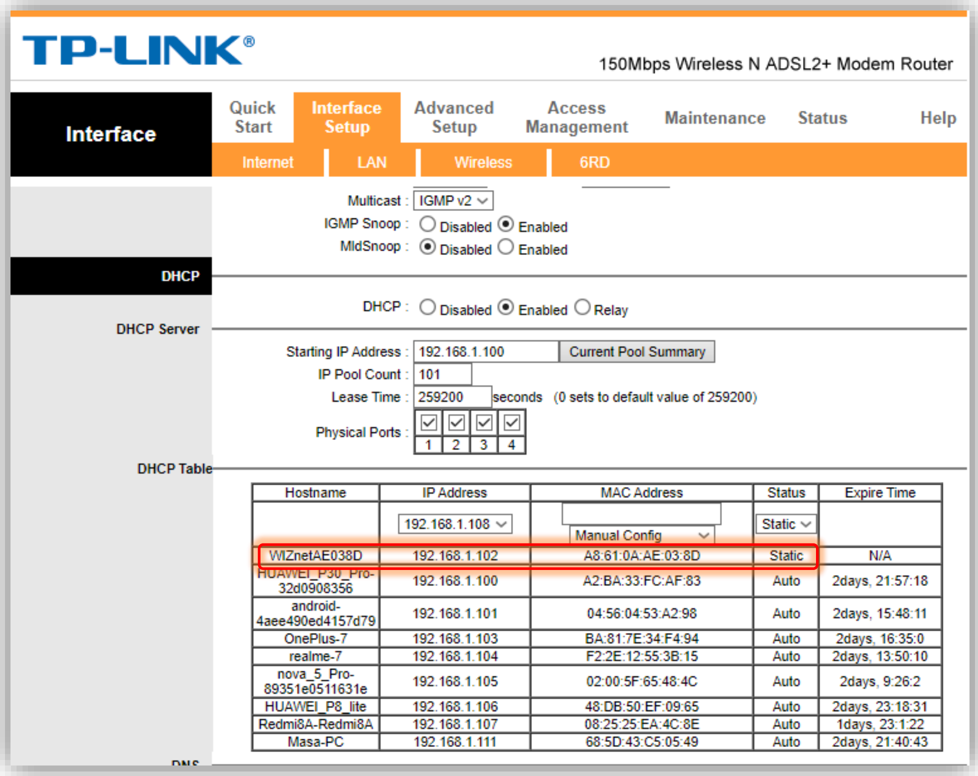
Lease Time : 259200 seconds (0 sets to default value of 259200)

Physical Ports : ☑ 1 ☑ 2 ☑ 3 ☑ 4

Hostname	IP Address	MAC Address	Status	Expire Time
	192.168.1.108	Manual Config	Static	
HUAWEI_P30_Pro-32d0908356	192.168.1.100	A2:BA:33:FC:AF:83	Auto	2days, 19:22:18
android-4aee490ed4157d79	192.168.1.101	04:56:04:53:A2:98	Auto	2days, 18:0:50
WIZnetAE038D	192.168.1.102	A8:61:0A:AE:03:8D	Auto	2days, 23:59:54
OnePlus-7	192.168.1.103	BA:81:7E:34:F4:94	Auto	2days, 18:47:39
realme-7	192.168.1.104	F2:2E:12:55:3B:15	Auto	2days, 16:2:49
nova_5_Pro-89351e0511631e	192.168.1.105	02:00:5F:65:48:4C	Auto	2days, 11:38:41
HUAWEI_P8_lite	192.168.1.106	48:DB:50:EF:09:65	Auto	2days, 23:38:46
Redmi8A-Redmi8A	192.168.1.107	08:25:25:EA:4C:8E	Auto	2days, 1:14:1
Masa-PC	192.168.1.111	68:5D:43:C5:05:49	Auto	2days, 23:53:22

.

Figure51. Static IP configuration



Multicast : IGMP v2

IGMP Snoop : ○ Disabled ● Enabled

MldSnoop : ● Disabled ○ Enabled

DHCP : ○ Disabled ● Enabled ○ Relay

Starting IP Address : 192.168.1.100

IP Pool Count : 101

Lease Time : 259200 seconds (0 sets to default value of 259200)

Physical Ports : ☑ 1 ☑ 2 ☑ 3 ☑ 4

Hostname	IP Address	MAC Address	Status	Expire Time
	192.168.1.108	Manual Config	Static	
WIZnetAE038D	192.168.1.102	A8:61:0A:AE:03:8D	Static	N/A
HUAWEI_P30_Pro-32d0908356	192.168.1.100	A2:BA:33:FC:AF:83	Auto	2days, 21:57:18
android-4aee490ed4157d79	192.168.1.101	04:56:04:53:A2:98	Auto	2days, 15:48:11
OnePlus-7	192.168.1.103	BA:81:7E:34:F4:94	Auto	2days, 16:35:0
realme-7	192.168.1.104	F2:2E:12:55:3B:15	Auto	2days, 13:50:10
nova_5_Pro-89351e0511631e	192.168.1.105	02:00:5F:65:48:4C	Auto	2days, 9:26:2
HUAWEI_P8_lite	192.168.1.106	48:DB:50:EF:09:65	Auto	2days, 23:18:31
Redmi8A-Redmi8A	192.168.1.107	08:25:25:EA:4C:8E	Auto	1days, 23:1:22
Masa-PC	192.168.1.111	68:5D:43:C5:05:49	Auto	2days, 21:40:43

Figure52. Static IP configuration

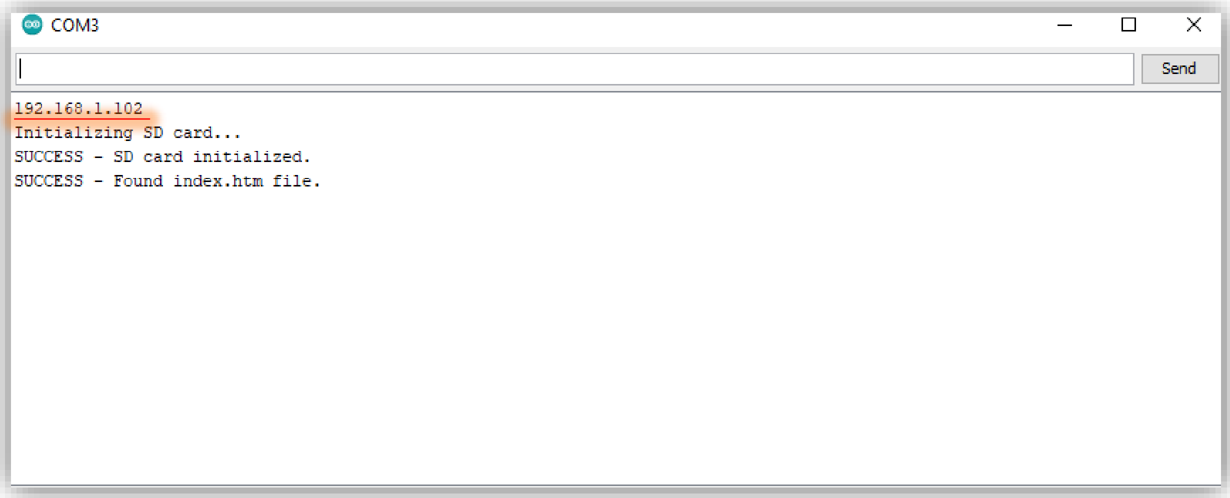
Now we can see that in the status column it says Static instead of Auto and by that we can say that we have successfully changed our IP address from dynamic to static.

3.9. Test on LAN:

Step9:

After the upload, we copy our Arduino UNO IP address and paste it in the URL bar.

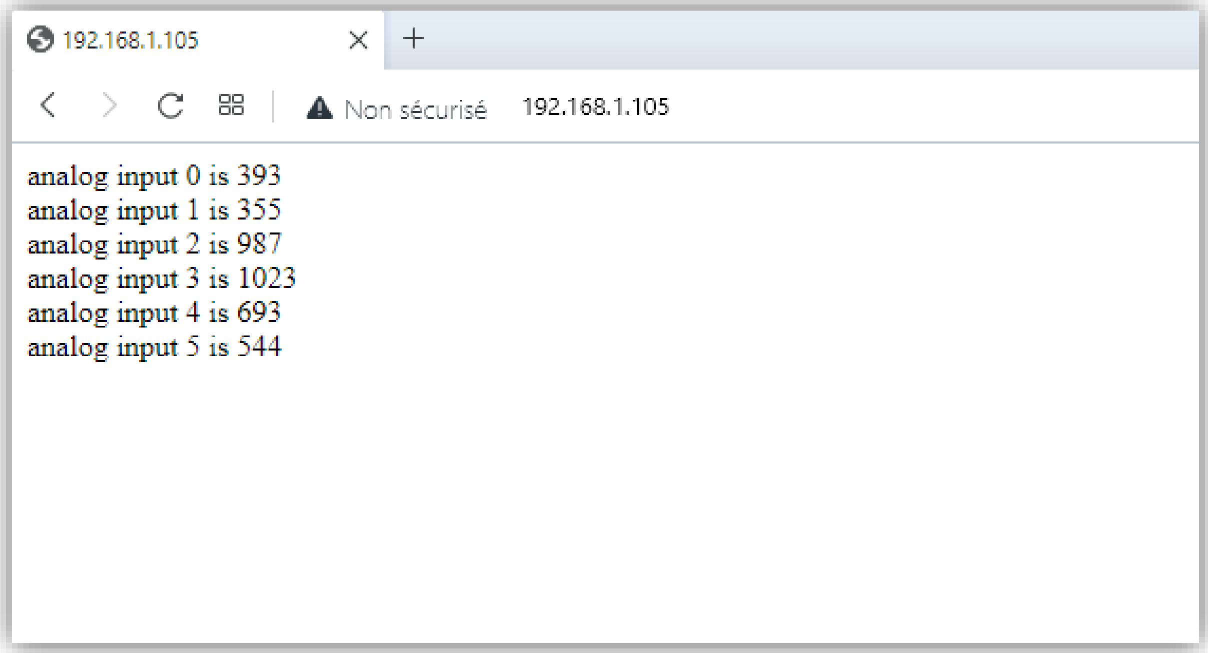
We go to serial monitor



```
192.168.1.102
Initializing SD card...
SUCCESS - SD card initialized.
SUCCESS - Found index.htm file.
```

Figure53. Arduino Uno IP adresse

The following should appear.



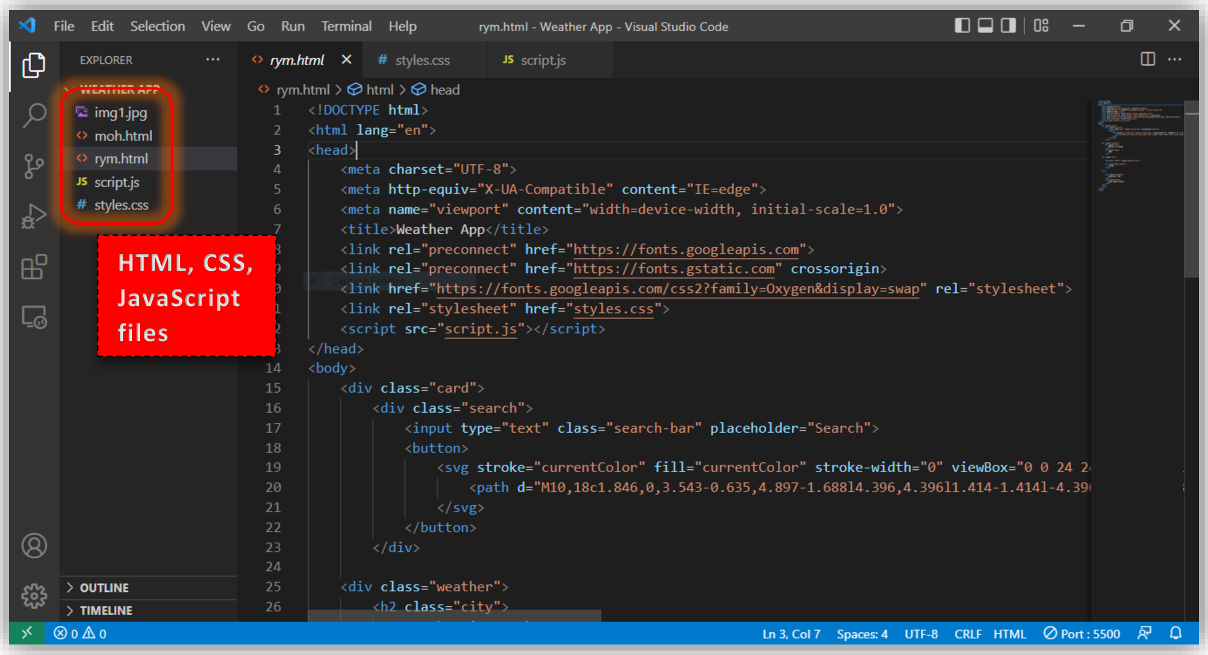
```
analog input 0 is 393
analog input 1 is 355
analog input 2 is 987
analog input 3 is 1023
analog input 4 is 693
analog input 5 is 544
```

Figure54. Test on LAN

3.10. Development of the website:

To create our website we will use Visual Studio which is an integrated development environment from Microsoft. We will include multiple languages like HTML, CSS, and JavaScript.

3.10.1. HTML: or HyperText Markup Language, allows web users to create and structure sections, paragraphs, and links using elements, tags, and attributes.

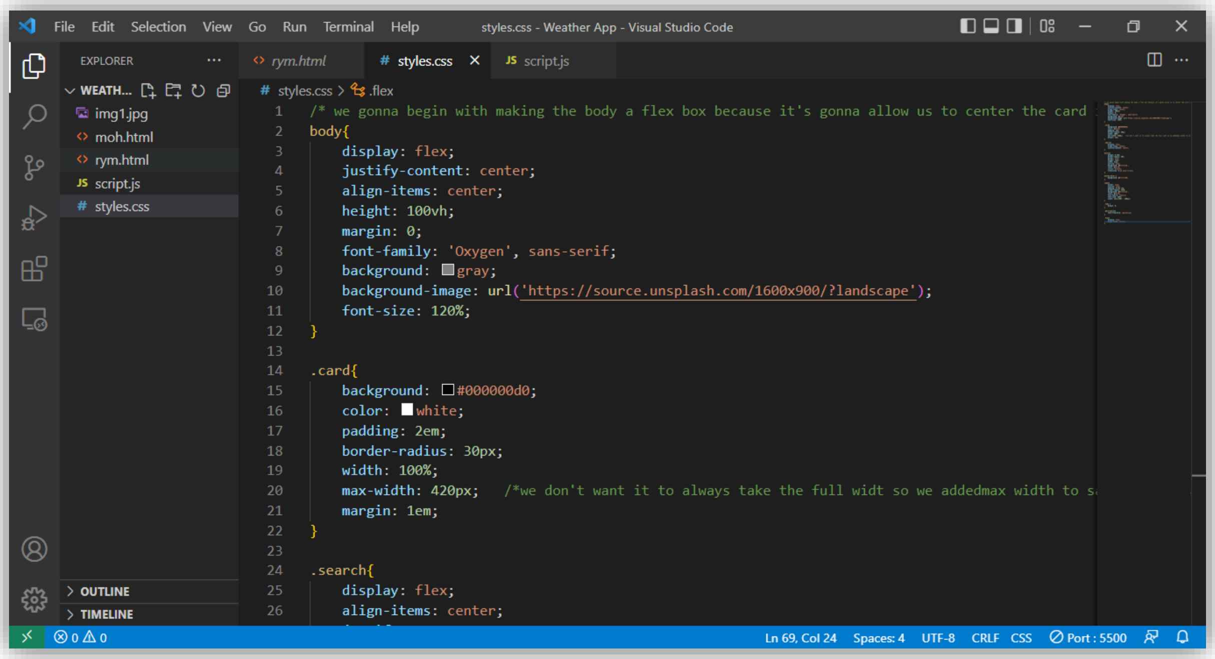


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Oxygen&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="styles.css">
    <script src="script.js"></script>
</head>
<body>
    <div class="card">
        <div class="search">
            <input type="text" class="search-bar" placeholder="Search">
            <button>
                <svg stroke="currentColor" fill="currentColor" stroke-width="0" viewBox="0 0 24 2
                    <path d="M10,18c1.846,0,3.543-0.635,4.897-1.688l4.396,4.396l1.414-1.414l-4.39
                </svg>
            </button>
        </div>

    <div class="weather">
        <h2 class="city">
```

Figure55. HTML

However, it’s worth noting that HTML is not considered a programming language as it can’t create dynamic functionality. That is why we also need CSS

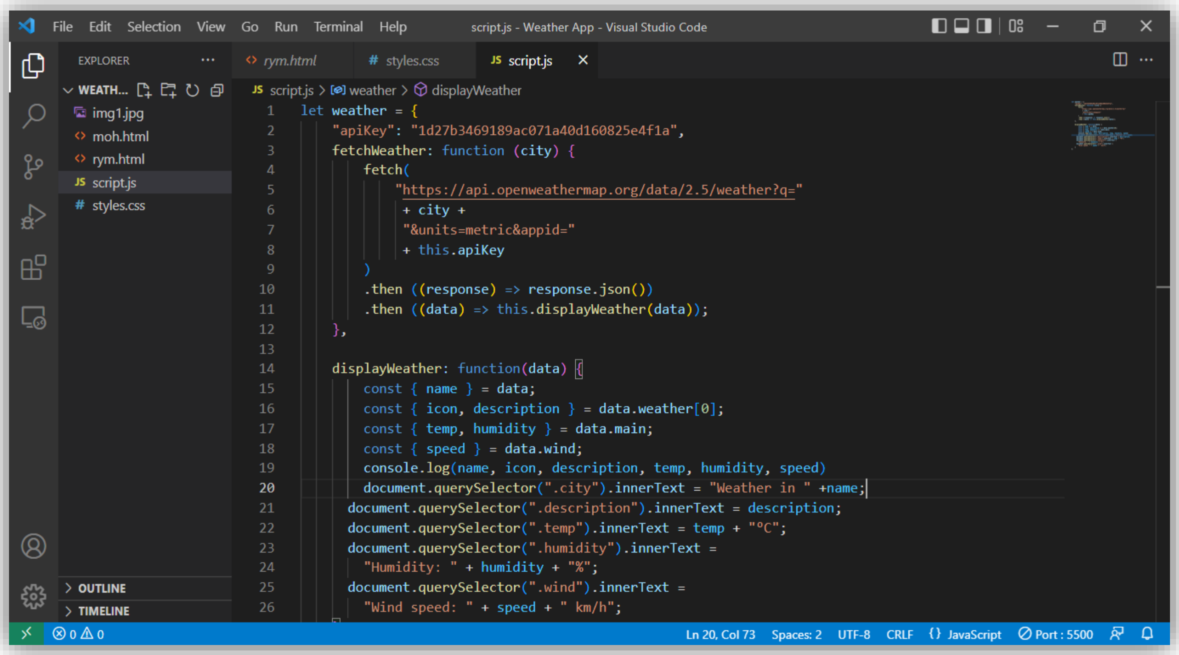
3.10.2. CSS: (Cascading Style Sheets) is used to style and layout web pages for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features


File Edit Selection View Go Run Terminal Help
styles.css - Weather App - Visual Studio Code
EXPLORER
WEATH...
img1.jpg
moh.html
rym.html
script.js
styles.css
rym.html
styles.css
script.js
styles.css > .flex
OUTLINE
TIMELINE
⊗ 0 ⚠ 0
Ln 69, Col 24 Spaces: 4 UTF-8 CRLF CSS Port : 5500

```css
/* we gonna begin with making the body a flex box because it's gonna allow us to center the card
body{
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    font-family: 'Oxygen', sans-serif;
    background: gray;
    background-image: url('https://source.unsplash.com/1600x900/?landscape');
    font-size: 120%;
}

.card{
    background: #000000d0;
    color: white;
    padding: 2em;
    border-radius: 30px;
    width: 100%;
    max-width: 420px;   /*we don't want it to always take the full widt so we addedmax width to s
    margin: 1em;
}

.search{
    display: flex;
    align-items: center;
```

Figure56. CSS

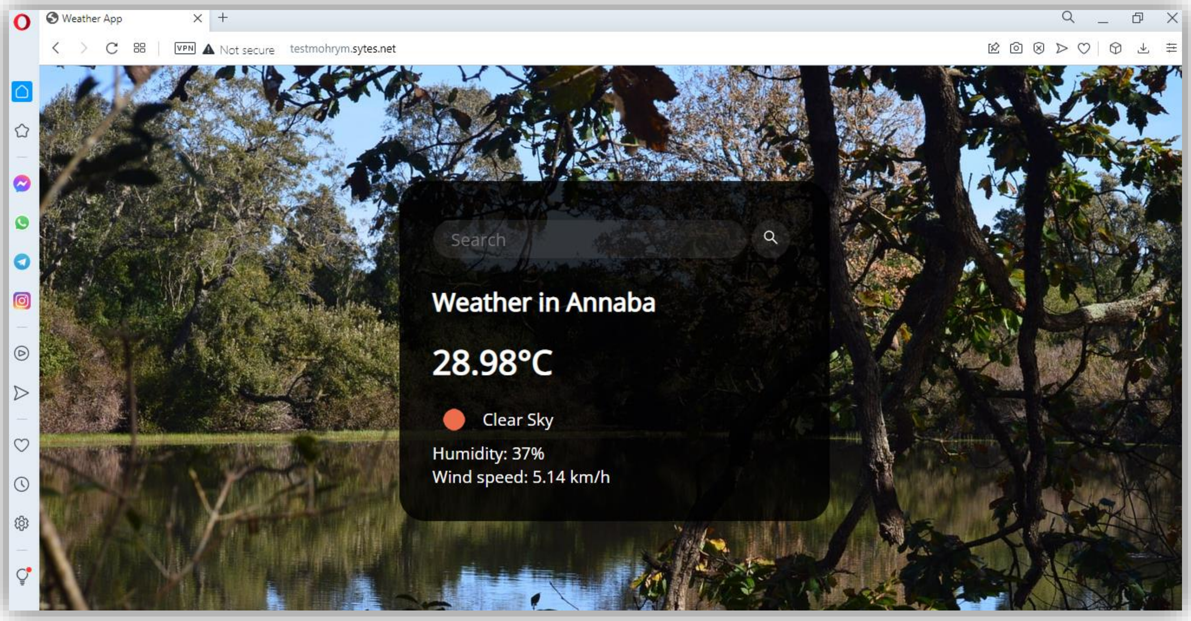
3.10.3. JavaScript: it is used to create dynamic and interactive web content like applications and browsers. JavaScript is so popular that it's the most used programming language in the world.


File Edit Selection View Go Run Terminal Help
script.js - Weather App - Visual Studio Code
EXPLORER
WEATH...
img1.jpg
moh.html
rym.html
script.js
styles.css
rym.html
styles.css
script.js
script.js > weather > displayWeather
OUTLINE
TIMELINE
⊗ 0 ⚠ 0
Ln 20, Col 73 Spaces: 2 UTF-8 CRLF {} JavaScript Port : 5500

```javascript
let weather = {
  "apiKey": "1d27b3469189ac071a40d160825e4f1a",
  fetchWeather: function (city) {
    fetch(
      "https://api.openweathermap.org/data/2.5/weather?q="
       + city +
       "&units=metric&appid="
       + this.apiKey
    )
    .then ((response) => response.json())
    .then ((data) => this.displayWeather(data));
  },

  displayWeather: function(data) {
    const { name } = data;
    const { icon, description } = data.weather[0];
    const { temp, humidity } = data.main;
    const { speed } = data.wind;
    console.log(name, icon, description, temp, humidity, speed)
    document.querySelector(".city").innerText = "Weather in " +name;
   document.querySelector(".description").innerText = description;
   document.querySelector(".temp").innerText = temp + "°C";
   document.querySelector(".humidity").innerText =
    "Humidity: " + humidity + "%";
   document.querySelector(".wind").innerText =
    "Wind speed: " + speed + " km/h";
```

Figure57. JavaScript

Finally we will open our website in any chosen web browser to see the final result. Type **testmohrym.sytes.net** in the URL bar and the following figure will be displayed. We can see that it will show the weather of Annaba city by default.



Weather in Annaba

28.98°C

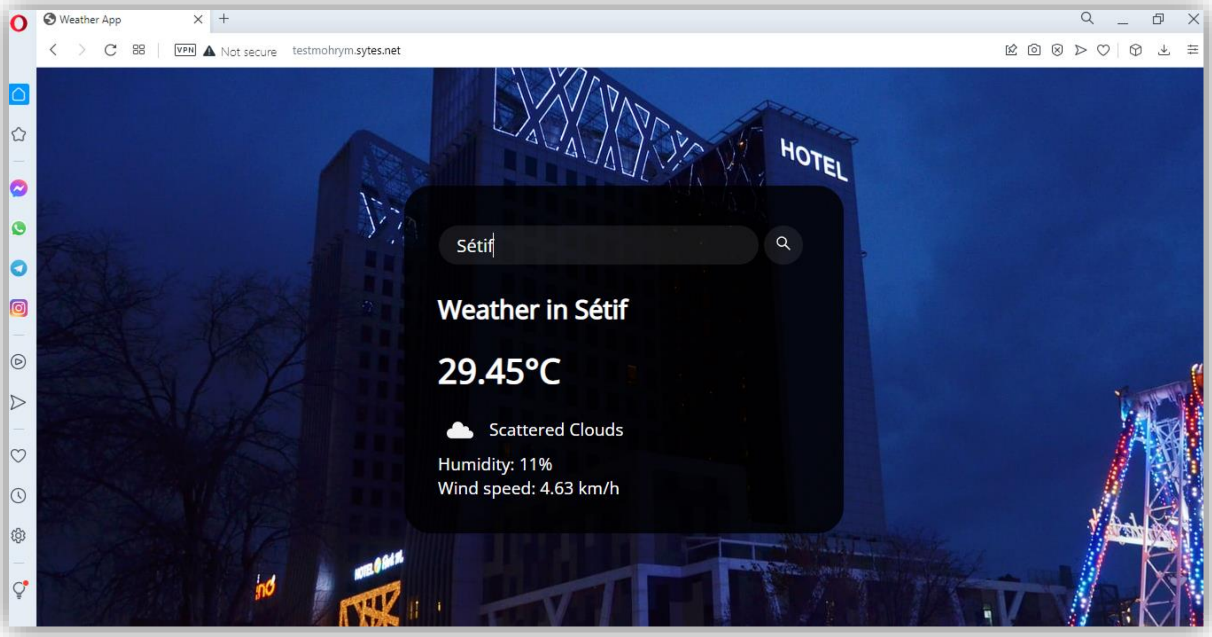
Clear Sky

Humidity: 37%

Wind speed: 5.14 km/h

Figure58. Website Interface

Now we will type any city name in the search bar and weather will be given to us as well as other details (weather description, humidity, wind speed).



Sétif

Weather in Sétif

29.45°C

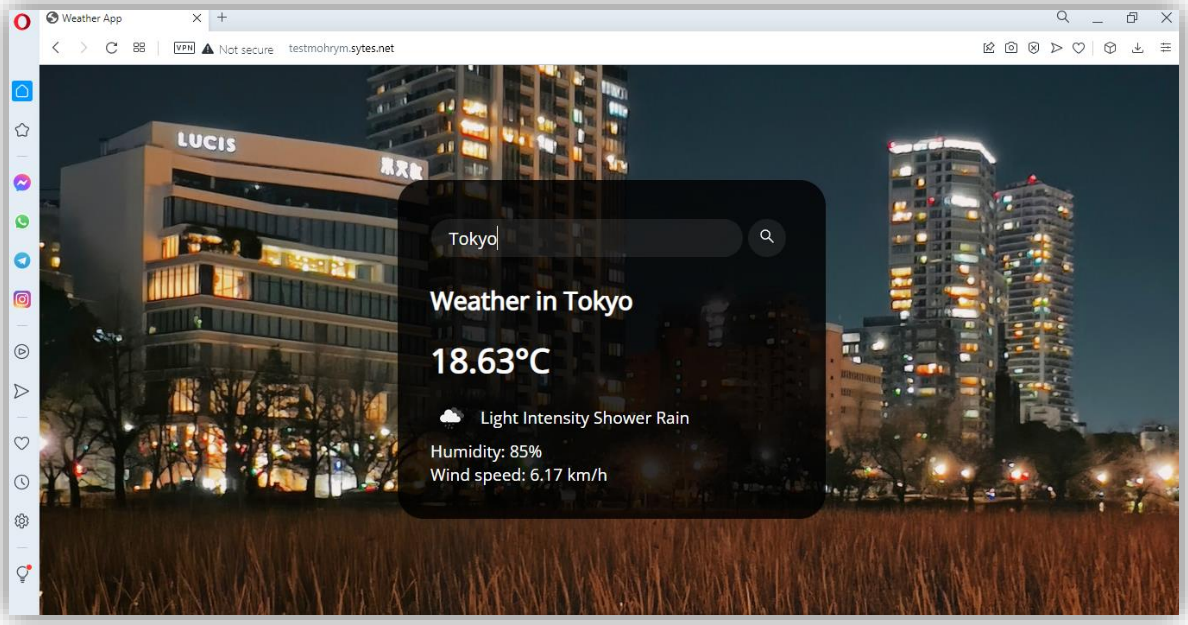
Scattered Clouds

Humidity: 11%

Wind speed: 4.63 km/h

Figure59. Website

We type again another city name in the search bar and you can notice that the background changes, and each time you make a new search or click on refresh a new image will be displayed.



Tokyo

Weather in Tokyo

18.63°C

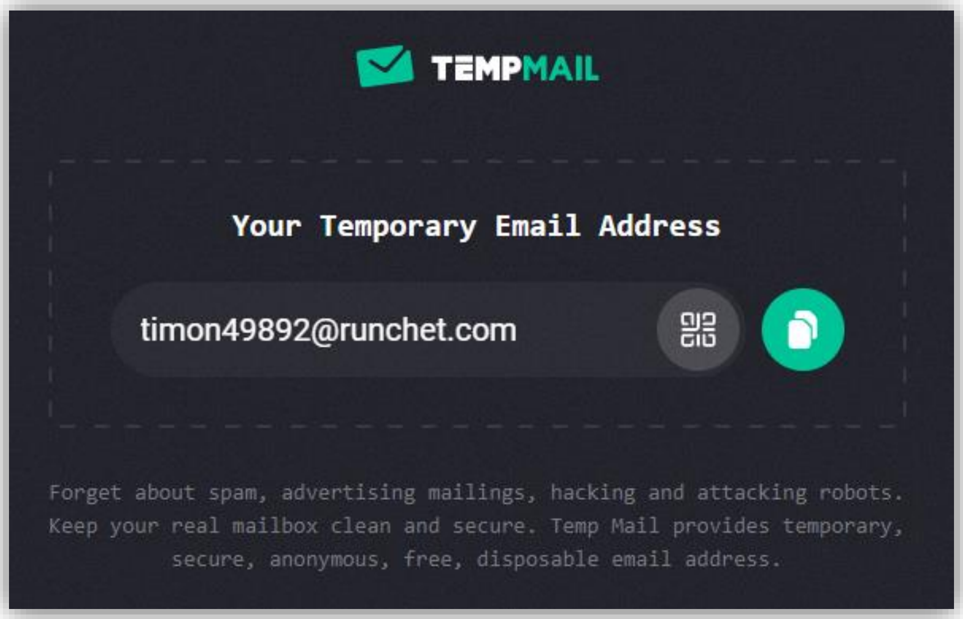
Light Intensity Shower Rain

Humidity: 85%
Wind speed: 6.17 km/h

Figure60. Website

3.11. Domain reservation:

To create our domain name we go to the no-ip page and click on sign up. To create our account we will use a temporary email address, and for that we will use www.temp-mail.org

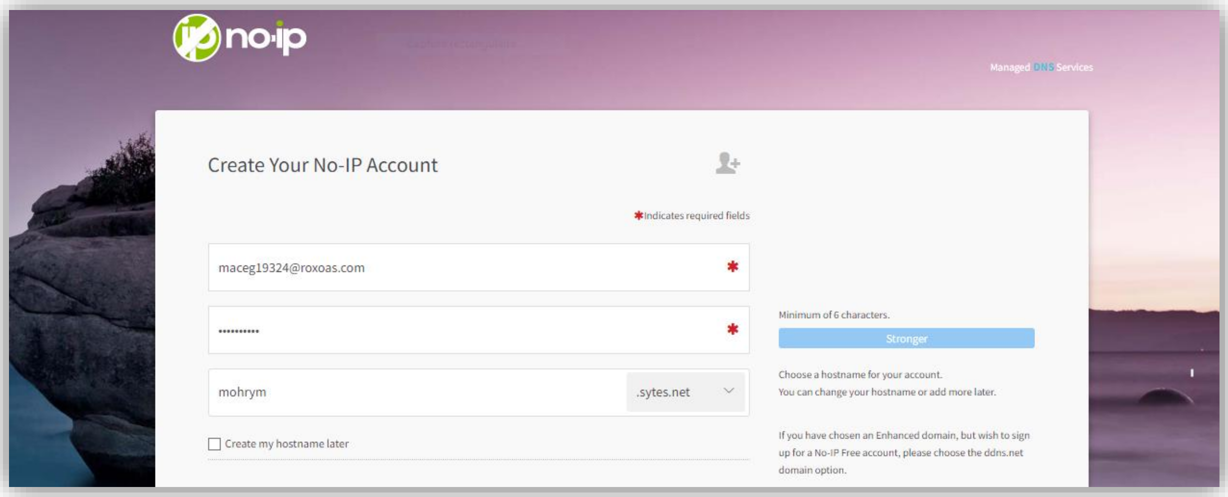


Your Temporary Email Address

timon49892@runchet.com

Forget about spam, advertising mailings, hacking and attacking robots. Keep your real mailbox clean and secure. Temp Mail provides temporary, secure, anonymous, free, disposable email address.

Figure61. Temp mail



Managed DNS Services

Create Your No-IP Account

*Indicates required fields

maceg19324@roxoas.com *

•••••••••• *

Minimum of 6 characters.

Stronger

mohrym .sytes.net

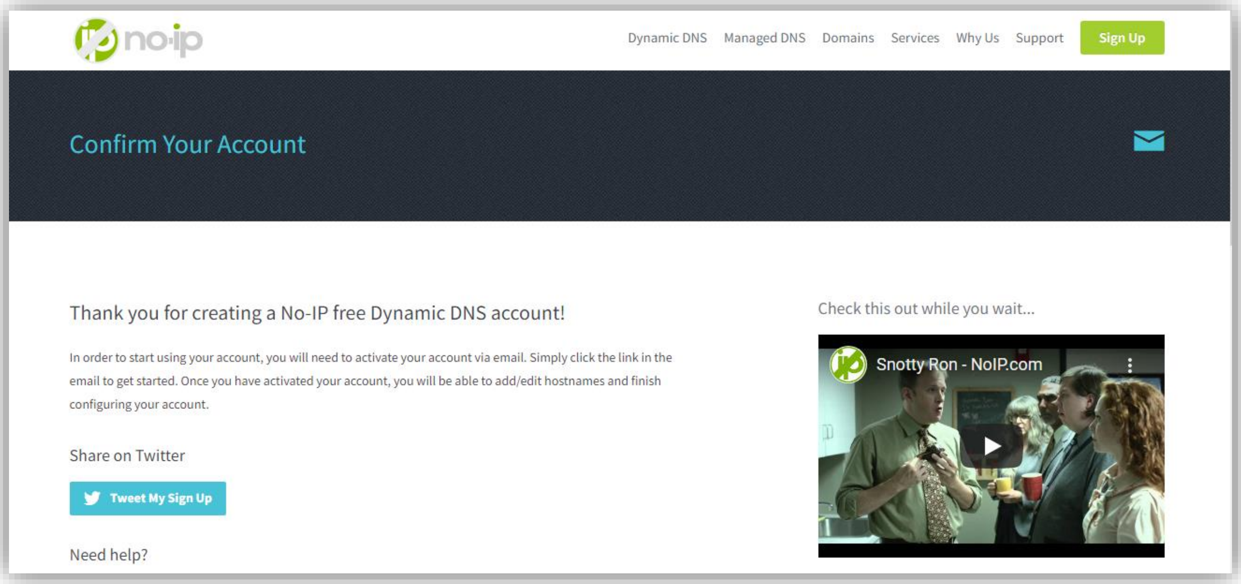
Choose a hostname for your account.
You can change your hostname or add more later.

□ Create my hostname later

If you have chosen an Enhanced domain, but wish to sign up for a No-IP Free account, please choose the ddns.net domain option.

Figure62. Sign up page

Then we check our temp mail to confirm our account.



Confirm Your Account

Thank you for creating a No-IP free Dynamic DNS account!

In order to start using your account, you will need to activate your account via email. Simply click the link in the email to get started. Once you have activated your account, you will be able to add/edit hostnames and finish configuring your account.

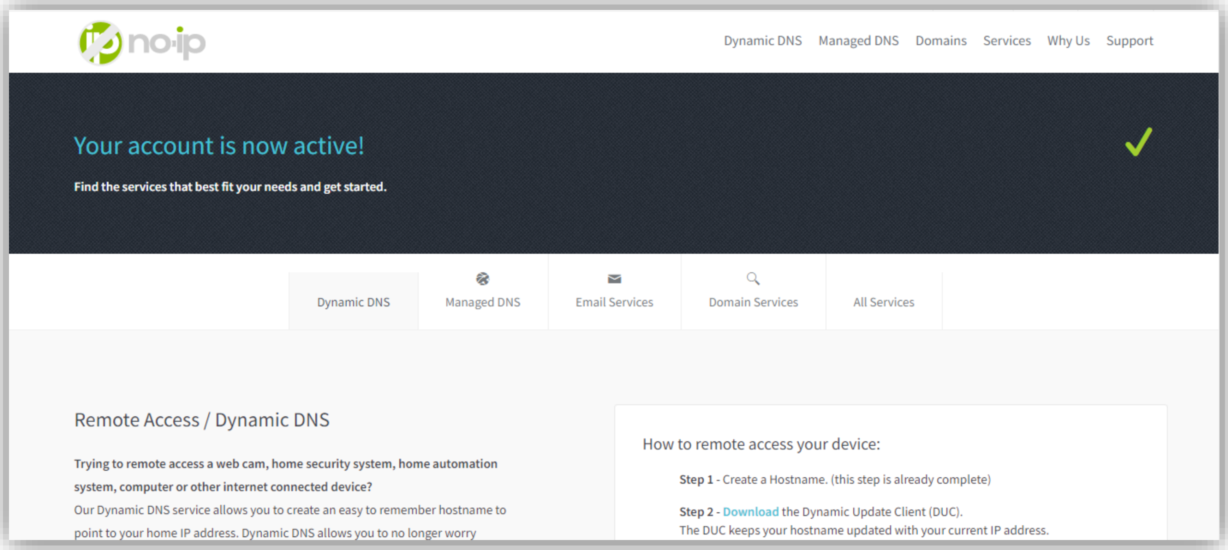
Share on Twitter

Tweet My Sign Up

Need help?

Check this out while you wait...

Figure63. Confirm Account



Your account is now active!

Find the services that best fit your needs and get started.

Remote Access / Dynamic DNS

Trying to remote access a web cam, home security system, home automation system, computer or other internet connected device?

Our Dynamic DNS service allows you to create an easy to remember hostname to point to your home IP address. Dynamic DNS allows you to no longer worry

How to remote access your device:

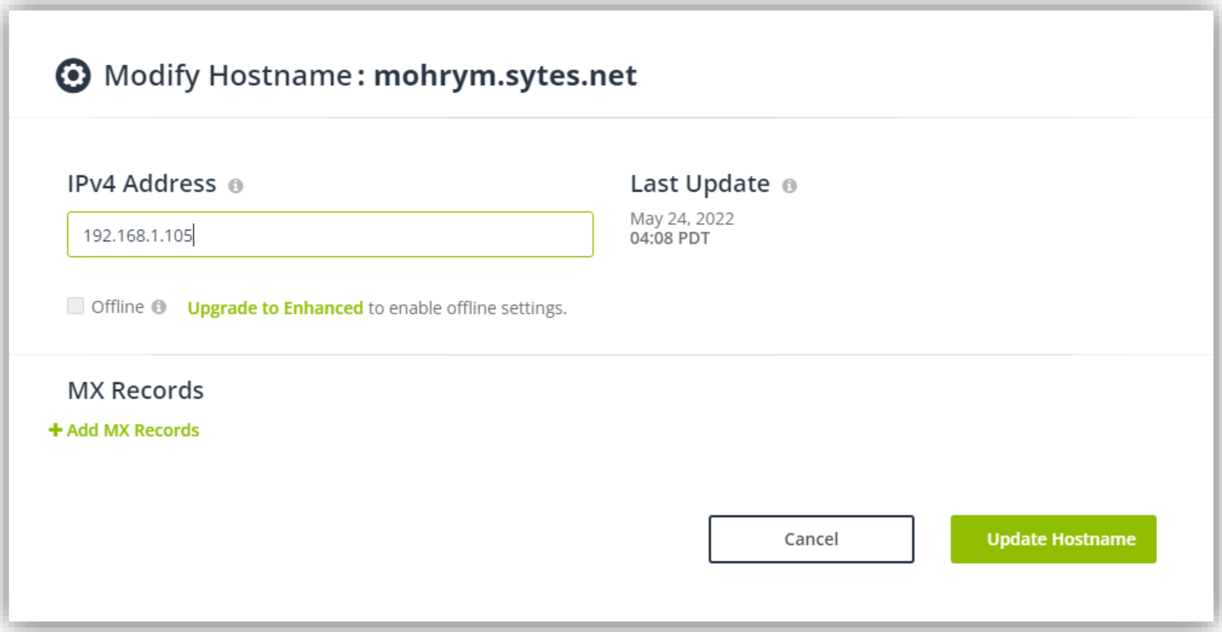
Step 1 - Create a Hostname. (this step is already complete)

Step 2 - Download the Dynamic Update Client (DUC).
The DUC keeps your hostname updated with your current IP address.

Figure64. Account Activation

We sign in then go to my account, click on active hostname, and finally click on modify to change the IP address.

We write down the IP address of our Arduino UNO board.



Modify Hostname : **mohrym.sytes.net**

IPv4 Address: 192.168.1.105

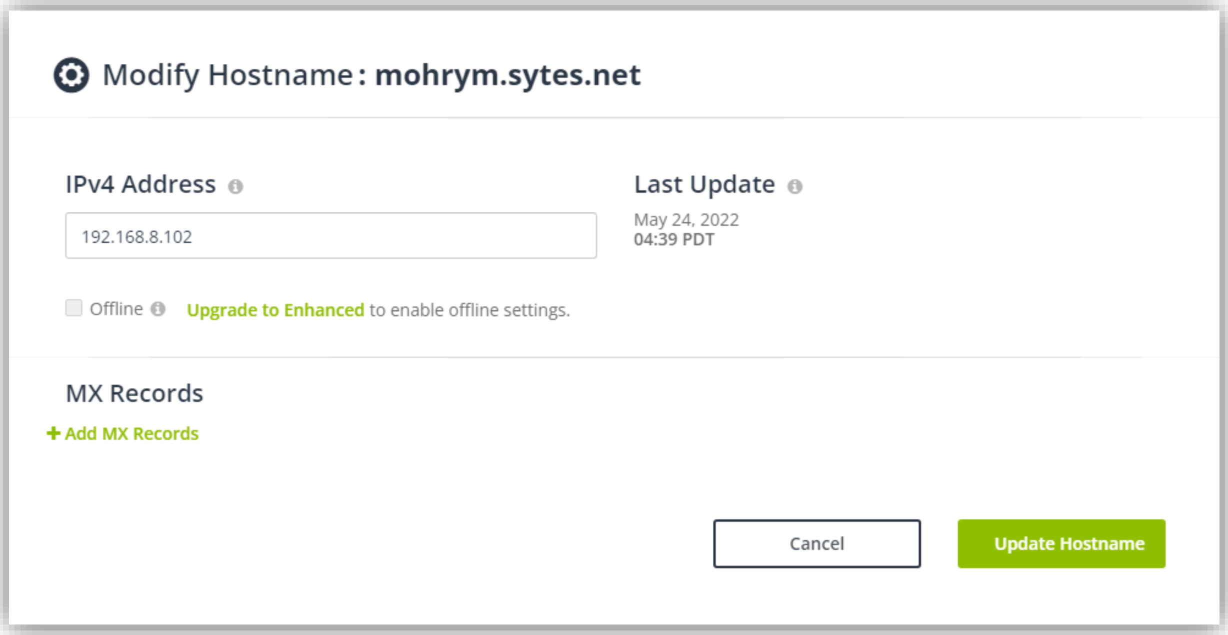
Last Update: May 24, 2022 04:08 PDT

□ Offline — Upgrade to Enhanced to enable offline settings.

MX Records

Figure65. Attachement of the IP

Then click on update hostname.



Modify Hostname : mohrym.sytes.net

IPv4 Address: 192.168.8.102

Last Update: May 24, 2022 04:39 PDT

□ Offline — Upgrade to Enhanced to enable offline settings.

MX Records

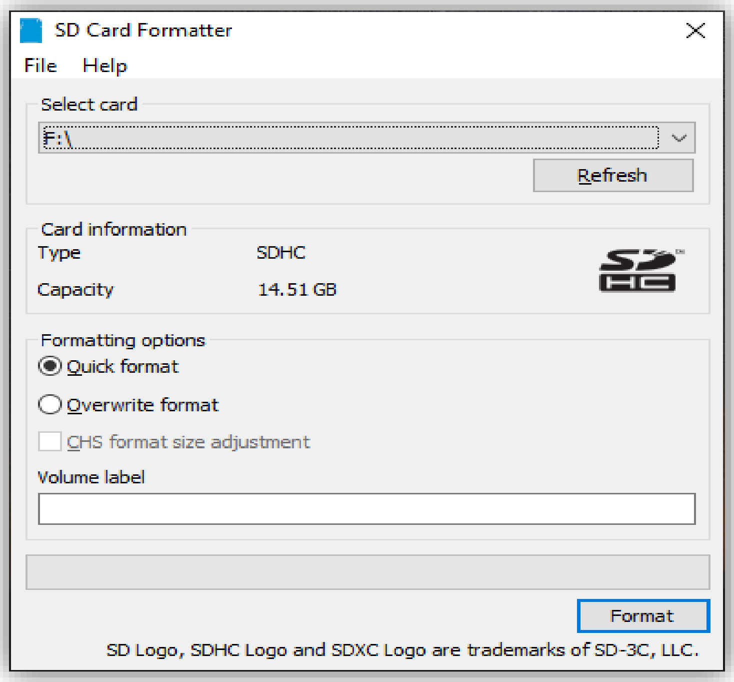
+ Add MX Records

Figure66. Attachement of the IP

3.12. Website activation:

3.12.1. Formatting SD card:

Now we have to store our website in the SD card but before doing that we have to make sure that it is formatted using the SD Card Formatter software. We are going to use an original microSDXC Flash Card 16GB.



SD Card Formatter

Select card: F:\

Card information

Type: SDHC

Capacity: 14.51 GB

Formatting options

● Quick format

○ Overwrite format

□ CHS format size adjustment

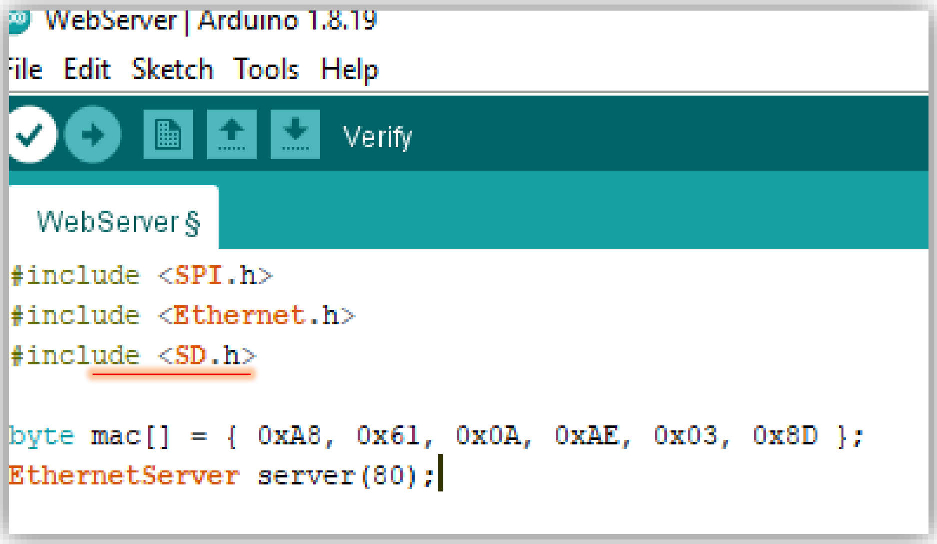
Volume label:

SD Logo, SDHC Logo and SDXC Logo are trademarks of SD-3C, LLC.

Figure67. SD Card Formatter

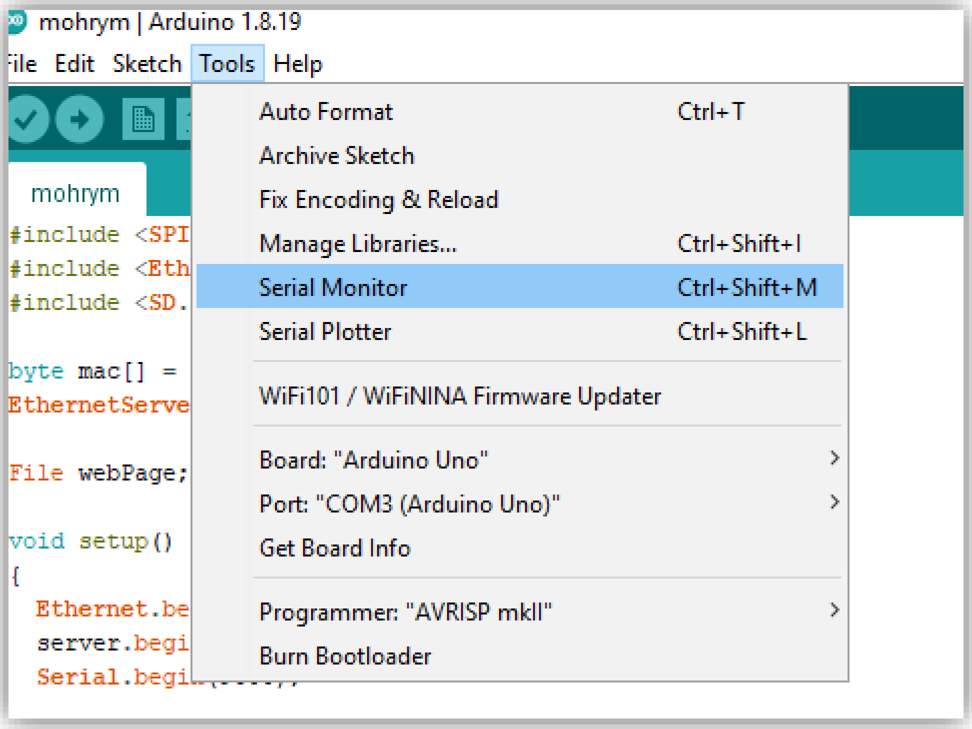
3.12.2. Adding the SD card library:

After that we insert the SD card in our Ethernet Shield and add the SD library to our program. Then go to tools and click on serial monitor to check if the SD card is initialized and if the website files are found in it.



```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0x03, 0x8D };
EthernetServer server(80);
```

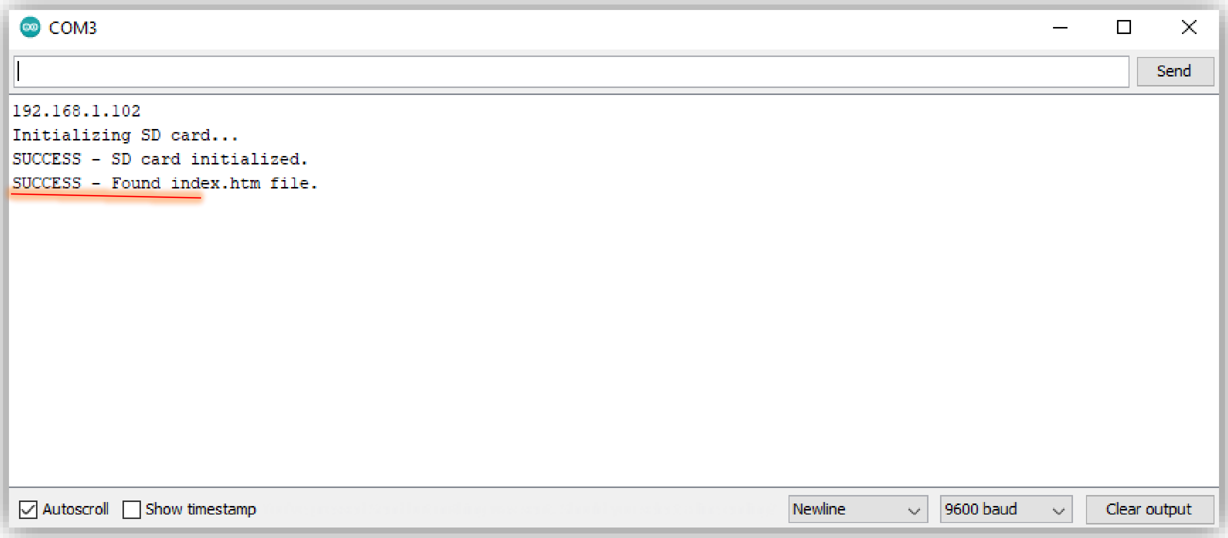
Figure68. Include SD card library



mohrym | Arduino 1.8.19
File Edit Sketch Tools Help
mohrym
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM3 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

```
#include <SPI
#include <Eth
#include <SD.

byte mac[] =
EthernetServe

File webPage;

void setup()
{
  Ethernet.be
  server.begi
  Serial.begi
```

Figure69. Serial Monitor

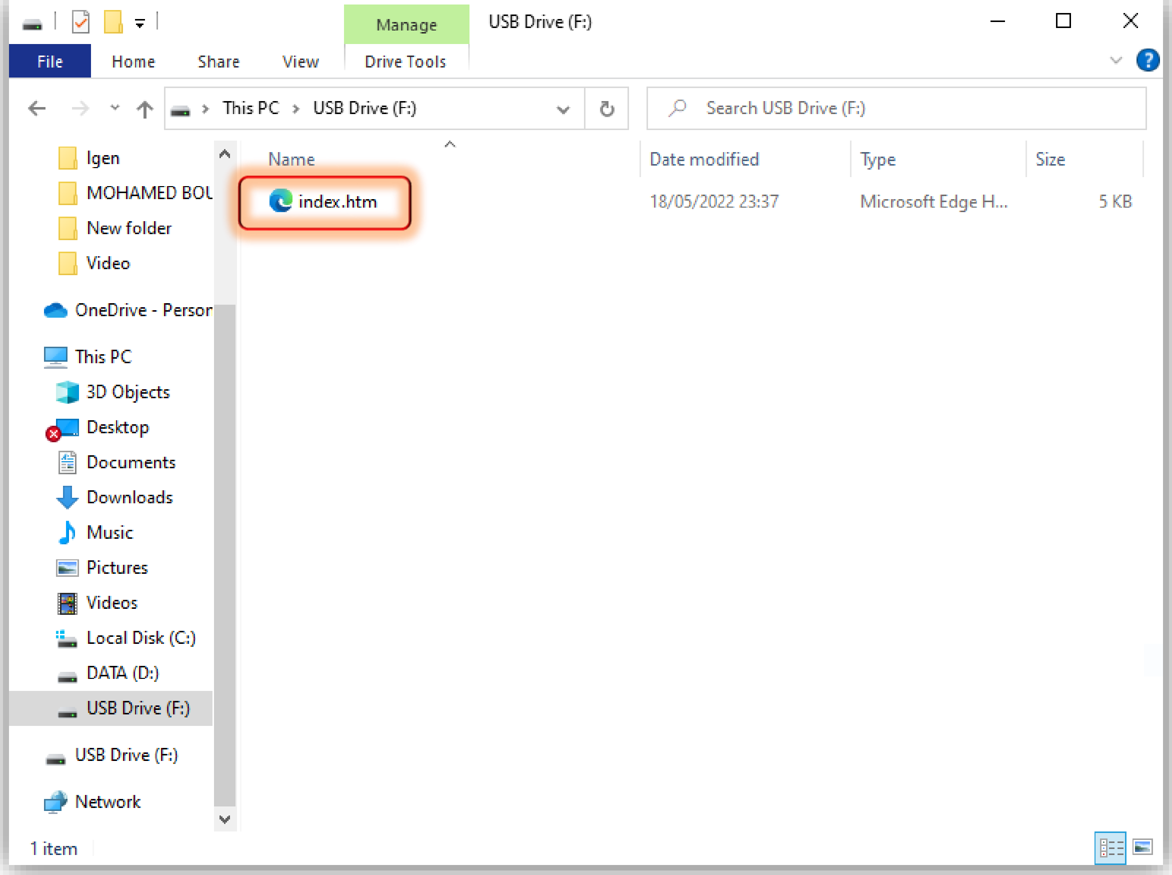


```
192.168.1.102
Initializing SD card...
SUCCESS - SD card initialized.
SUCCESS - Found index.htm file.
```

Figure70. Initializing SD card

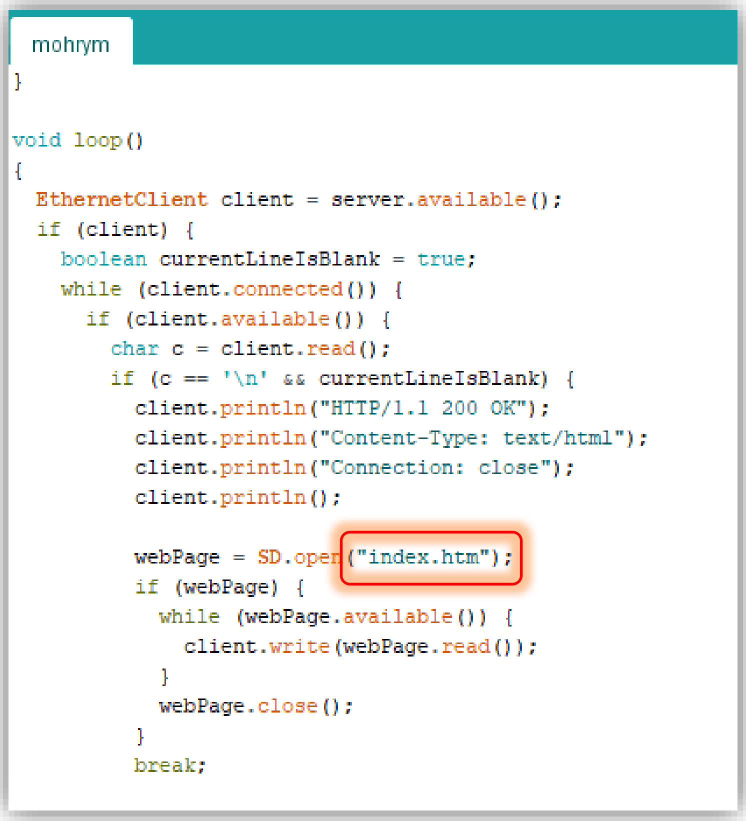
Note:

If you change your file name then you should also change it in the program. For example our file name is index.htm so we have to write in the program index.htm



Name	Date modified	Type	Size
index.htm	18/05/2022 23:37	Microsoft Edge H...	5 KB

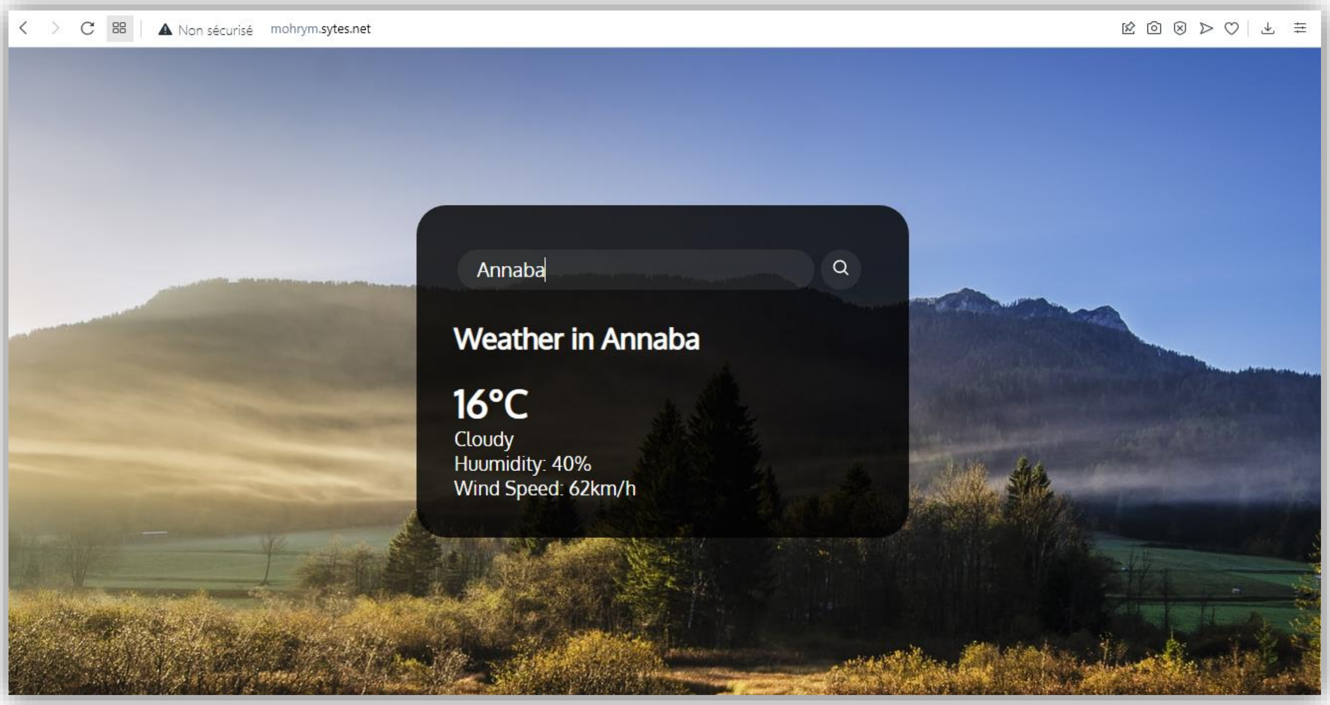
Figure71. Index file stored in the SD card



```
}

void loop()
{
  EthernetClient client = server.available();
  if (client) {
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        if (c == '\n' && currentLineIsBlank) {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println();

          webPage = SD.open("index.htm");
          if (webPage) {
            while (webPage.available()) {
              client.write(webPage.read());
            }
            webPage.close();
          }
          break;
```

Figure72. Index.htm

The names must be the exact same or else you will have an error in your program and the files will not be found. We upload the program to our Arduino UNO board, then we test again on our LAN, but this time we will use our domain name instead of our local Arduino UNO address.



Annaba

Weather in Annaba

16°C

Cloudy

Huumidity: 40%

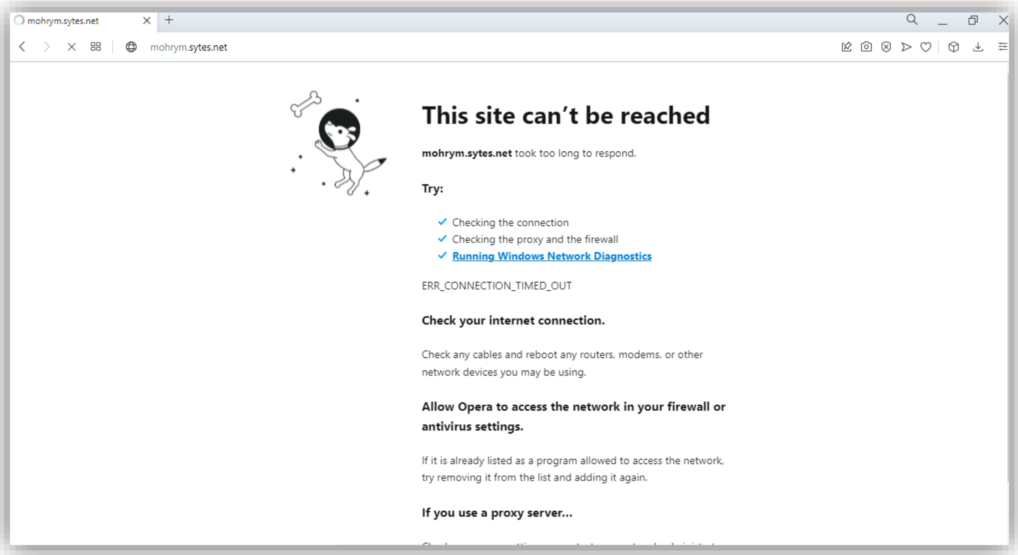
Wind Speed: 62km/h

Figure73. Test on LAN with domain name

As you can see our website is working but it’s not on the internet yet because it’s only on our LAN. Now we need to broadcast it on WAN.

3.13. Test on WAN:

We type our domain name or the public IP address using another computer that is connected to a different router.



This site can't be reached

mohrym.sytes.net took too long to respond.

Try:

- ✓ Checking the connection
- ✓ Checking the proxy and the firewall
- ✓ Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Check your internet connection.

Check any cables and reboot any routers, modems, or other network devices you may be using.

Allow Opera to access the network in your firewall or antivirus settings.

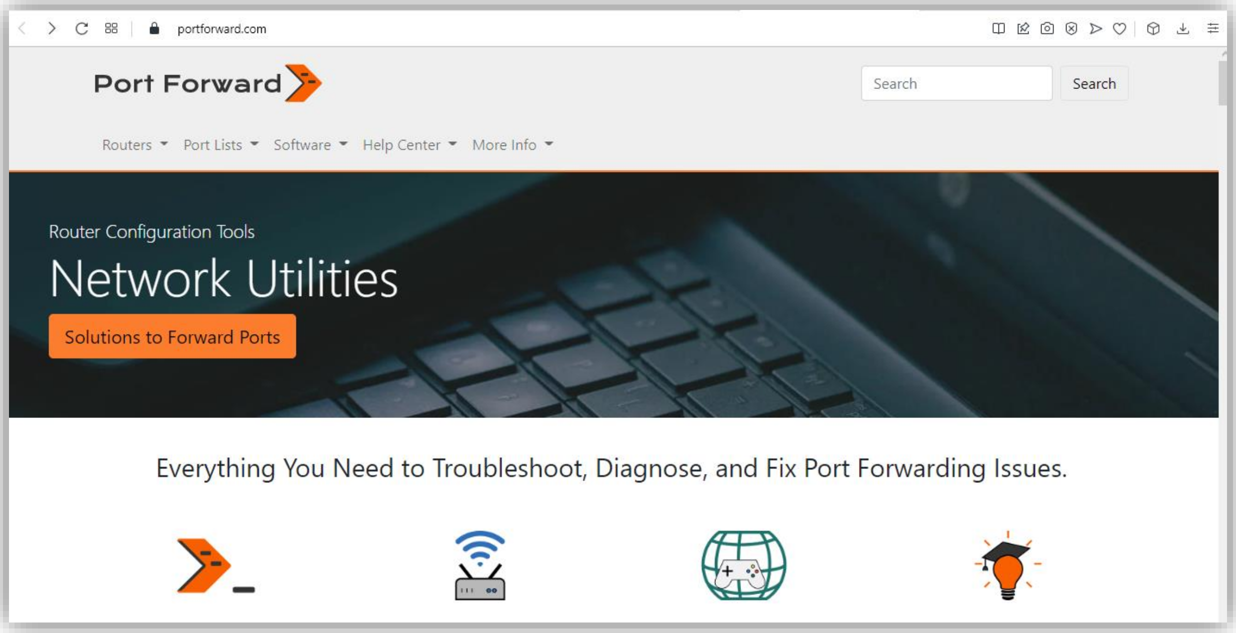
If it is already listed as a program allowed to access the network, try removing it from the list and adding it again.

If you use a proxy server...

Figure74. Test on WAN

The webpage didn't appear because our ISP (Internet Service Provider) blocks common ports like (80, 22, 21, 53, etc). Using Port Forwarding software app, we are going to check if the common port 80 and 22 are blocked by our ISP (Algérie Télécom) or not.

To install the software we are going to their official webpage www.portforward.com , click on software, Network utilities, then scroll down to download the free trial.

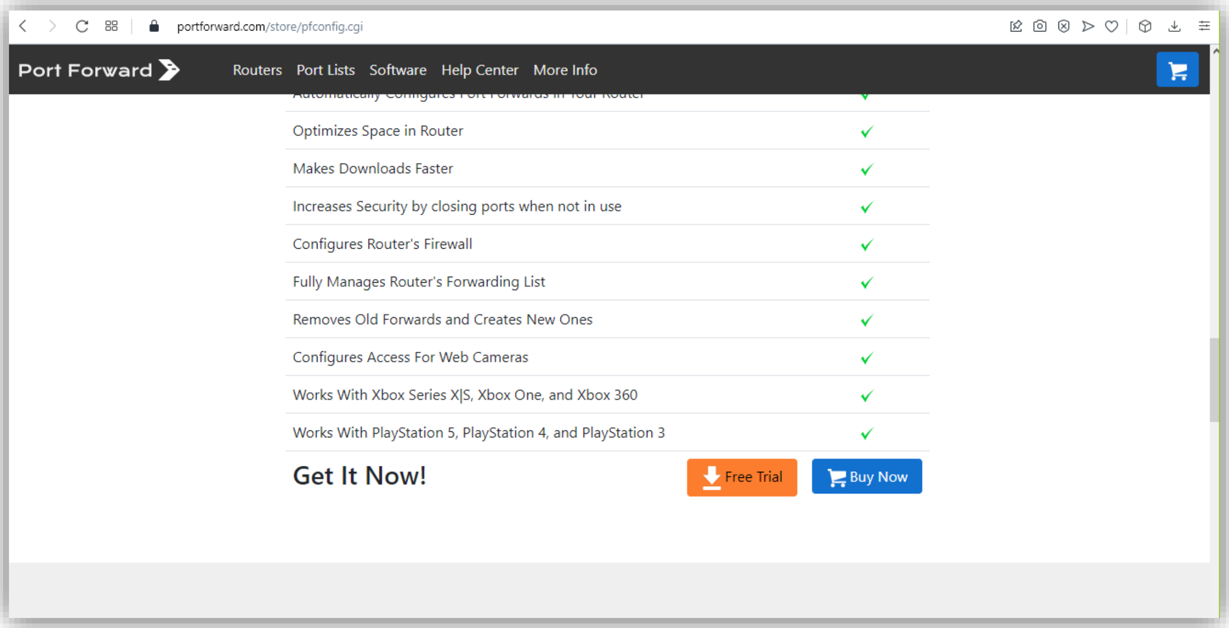


Router Configuration Tools

Network Utilities

Everything You Need to Troubleshoot, Diagnose, and Fix Port Forwarding Issues.

Figure75. Network Utilities Page

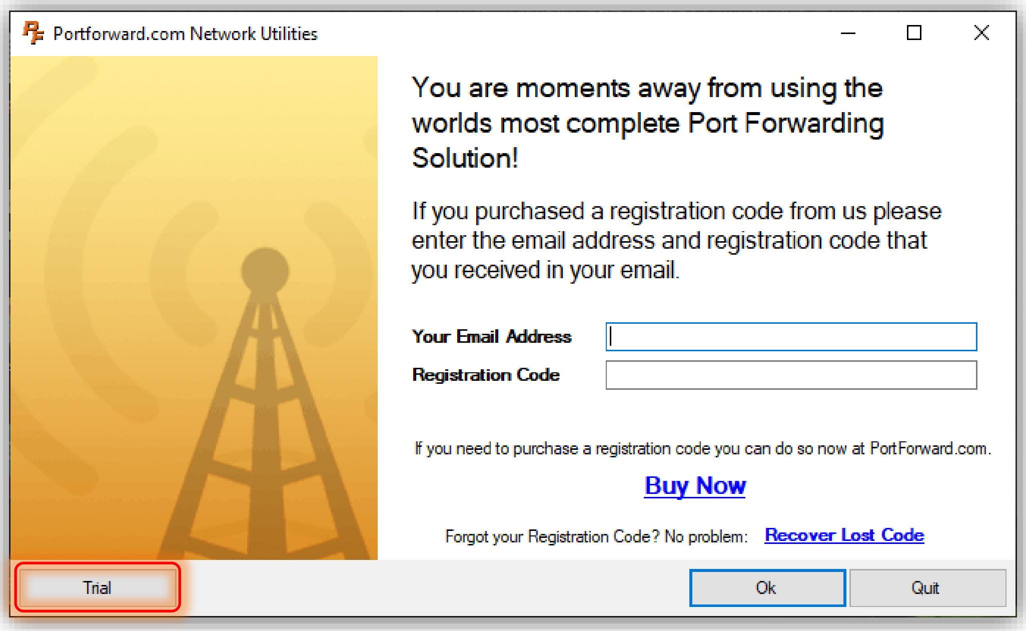


Automatically Configures Port Forwards in Your Router	✓
Optimizes Space in Router	✓
Makes Downloads Faster	✓
Increases Security by closing ports when not in use	✓
Configures Router's Firewall	✓
Fully Manages Router's Forwarding List	✓
Removes Old Forwards and Creates New Ones	✓
Configures Access For Web Cameras	✓
Works With Xbox Series X|S, Xbox One, and Xbox 360	✓
Works With PlayStation 5, PlayStation 4, and PlayStation 3	✓

Get It Now!

Figure76. Free Trial Download

After the download is finished, we launch the Port Forwarding software app and click on trial, then choose Port Checker.



You are moments away from using the worlds most complete Port Forwarding Solution!

If you purchased a registration code from us please enter the email address and registration code that you received in your email.

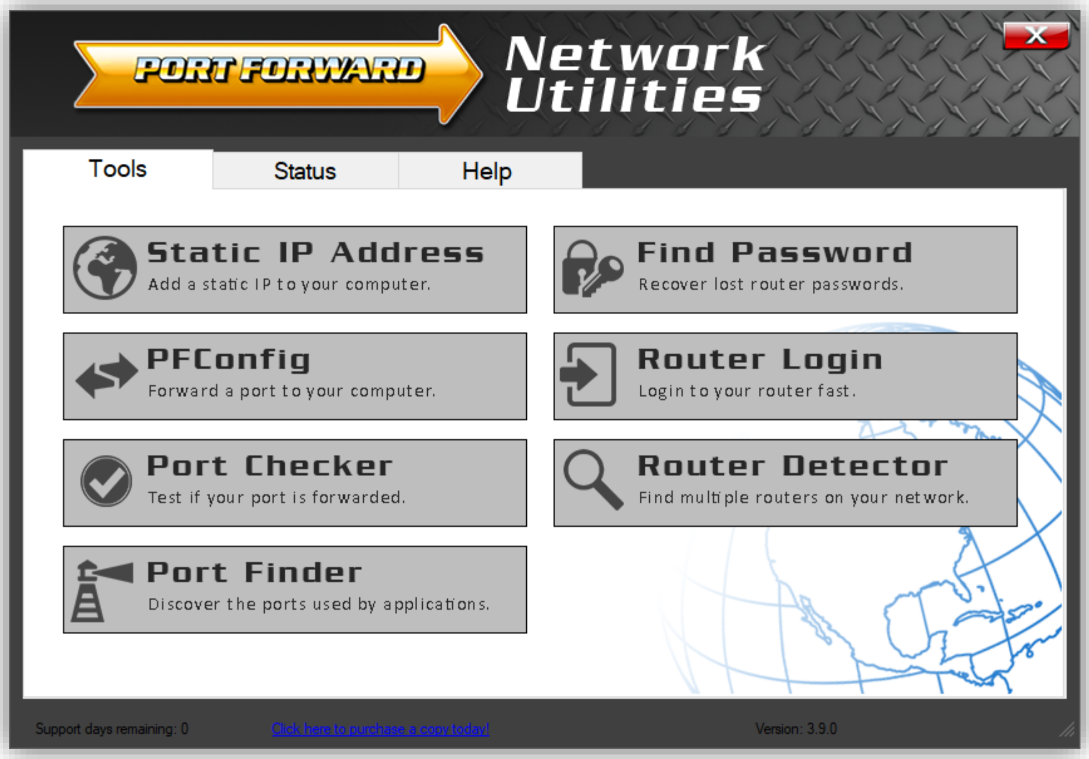
Your Email Address

Registration Code

If you need to purchase a registration code you can do so now at PortForward.com.

Forgot your Registration Code? No problem: Recover Lost Code

Figure77. Portforward software



Static IP Address
Add a static IP to your computer.

Find Password
Recover lost router passwords.

PFConfig
Forward a port to your computer.

Router Login
Login to your router fast.

Port Checker
Test if your port is forwarded.

Router Detector
Find multiple routers on your network.

Port Finder
Discover the ports used by applications.

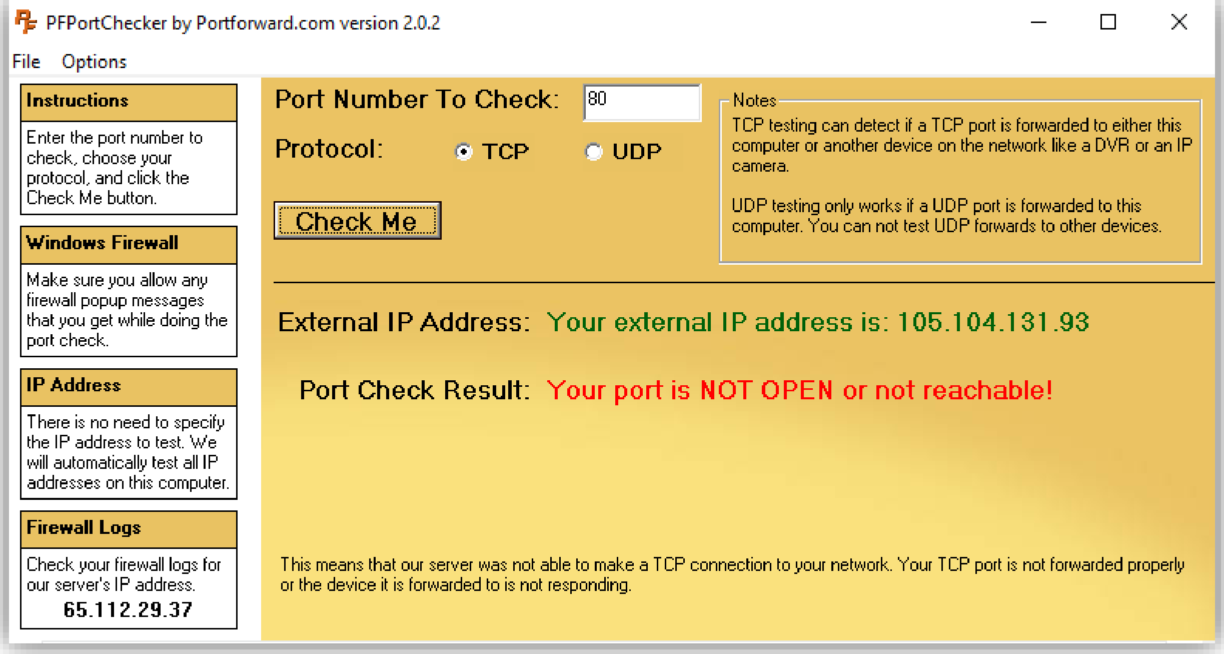
Support days remaining: 0

Click here to purchase a copy today!

Version: 3.9.0

Figure78. Portforward software interface

After clicking on Port Checker, we insert our port number in the box then click on Check me. Finally the port check result will be displayed and we can see that port 80 and 22 are blocked "Your port is NOT OPEN or not reachable".



Instructions

Enter the port number to check, choose your protocol, and click the Check Me button.

Windows Firewall

Make sure you allow any firewall popup messages that you get while doing the port check.

IP Address

There is no need to specify the IP address to test. We will automatically test all IP addresses on this computer.

Firewall Logs

Check your firewall logs for our server's IP address.

65.112.29.37

Port Number To Check: 80

Protocol: ● TCP ○ UDP

Notes

TCP testing can detect if a TCP port is forwarded to either this computer or another device on the network like a DVR or an IP camera.

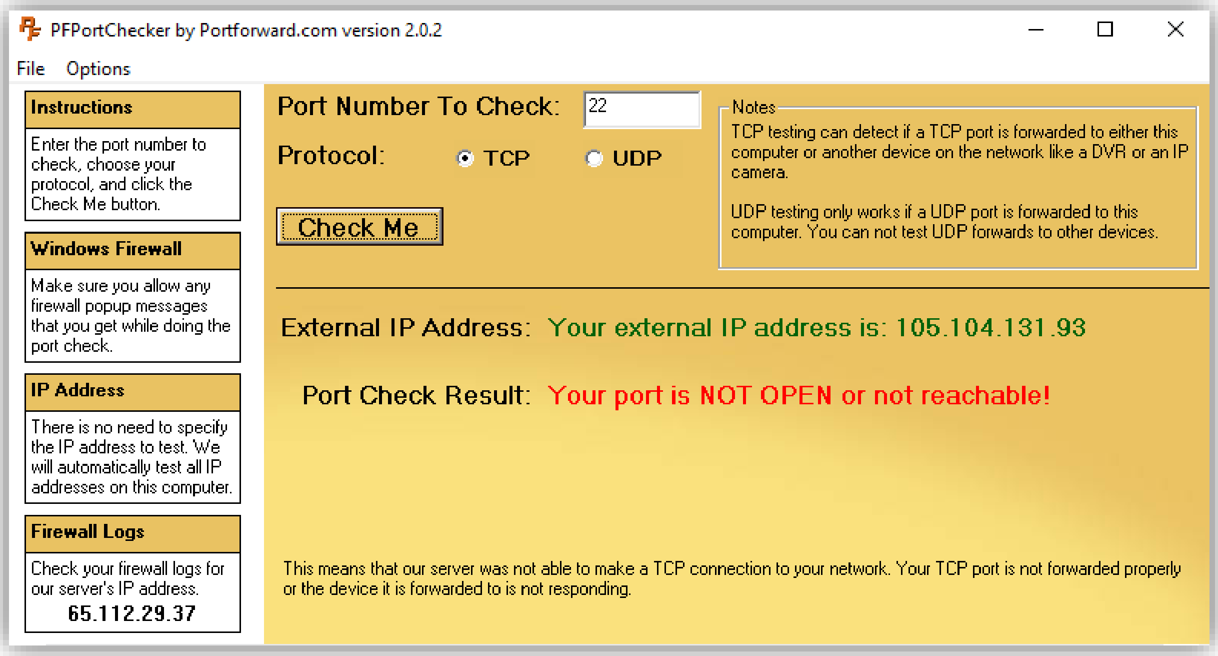
UDP testing only works if a UDP port is forwarded to this computer. You can not test UDP forwards to other devices.

External IP Address: Your external IP address is: 105.104.131.93

Port Check Result: Your port is NOT OPEN or not reachable!

This means that our server was not able to make a TCP connection to your network. Your TCP port is not forwarded properly or the device it is forwarded to is not responding.

Figure79. Checking port 80



PFPortChecker by Portforward.com version 2.0.2

File Options

Instructions

Enter the port number to check, choose your protocol, and click the Check Me button.

Windows Firewall

Make sure you allow any firewall popup messages that you get while doing the port check.

IP Address

There is no need to specify the IP address to test. We will automatically test all IP addresses on this computer.

Firewall Logs

Check your firewall logs for our server's IP address.
65.112.29.37

Port Number To Check: 22

Protocol: ● TCP ○ UDP

Check Me

Notes

TCP testing can detect if a TCP port is forwarded to either this computer or another device on the network like a DVR or an IP camera.

UDP testing only works if a UDP port is forwarded to this computer. You can not test UDP forwards to other devices.

External IP Address: Your external IP address is: 105.104.131.93

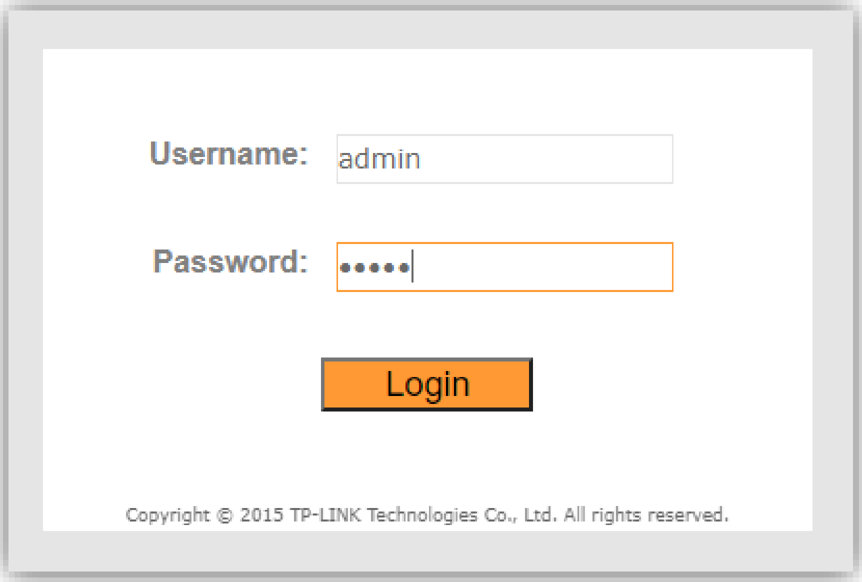
Port Check Result: Your port is NOT OPEN or not reachable!

This means that our server was not able to make a TCP connection to your network. Your TCP port is not forwarded properly or the device it is forwarded to is not responding.

Figure80. Checking port 22

3.14. Port Forwarding:

Step1: We open our web browser and type the public IP address of our router 192.168.1.1 then log in with the username of: admin ; and password: admin (by default).



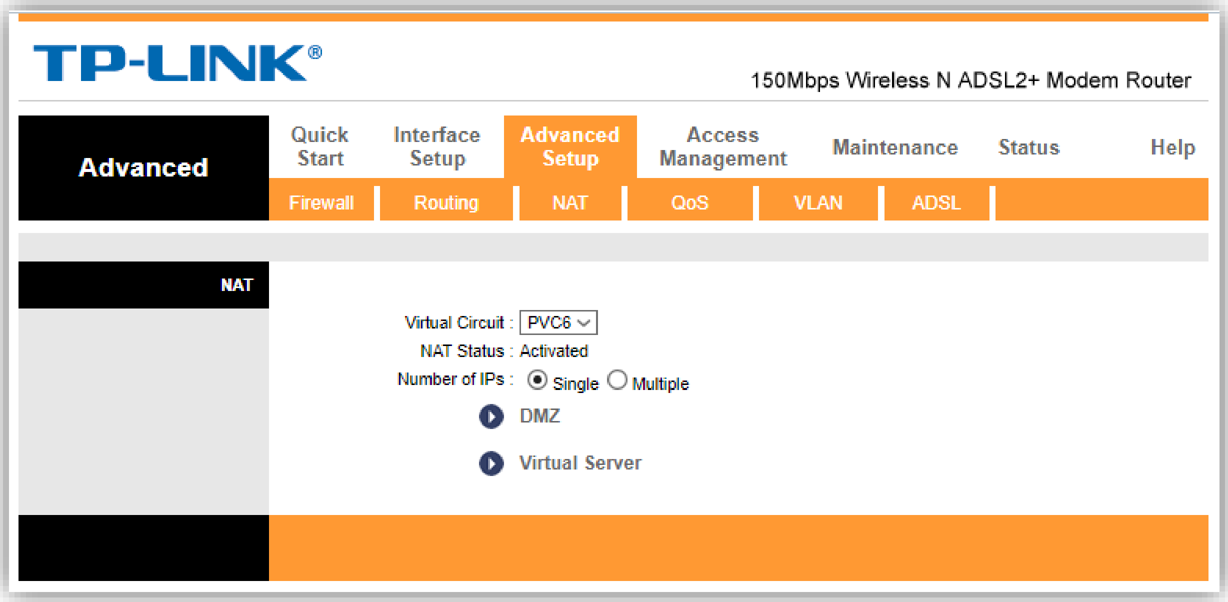
Username: admin

Password: •••••

Copyright © 2015 TP-LINK Technologies Co., Ltd. All rights reserved.

Figure81. Login

Step2: we click on advanced setup, NAT, then click on virtual server



Virtual Circuit : PVC6

NAT Status : Activated

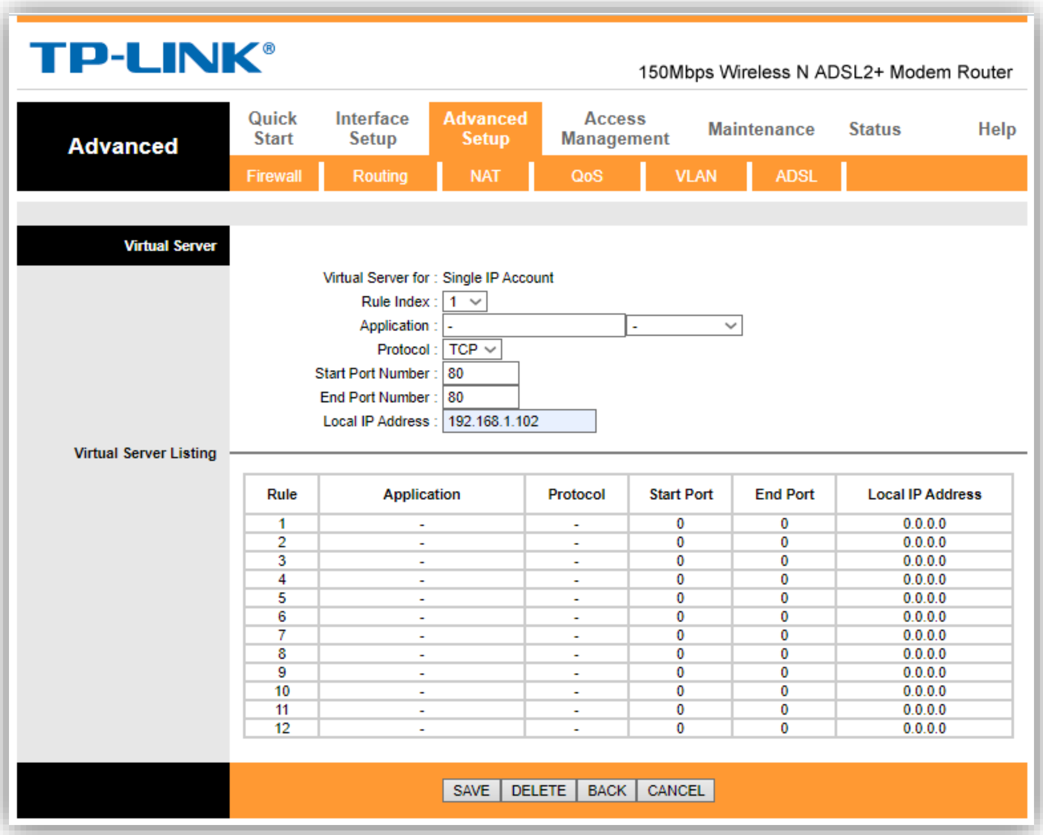
Number of IPs : ● Single ○ Multiple

DMZ

Virtual Server

Figure82. Login

Step3: we start by configuring our virtual server. Protocol: TCP, Start Port Number: 80, End Port Number: 80, finally we insert our arduino UNO IP address 192.168.1.102 then click save.



Virtual Server for : Single IP Account

Rule Index : 1

Application : - -

Protocol : TCP

Start Port Number : 80

End Port Number : 80

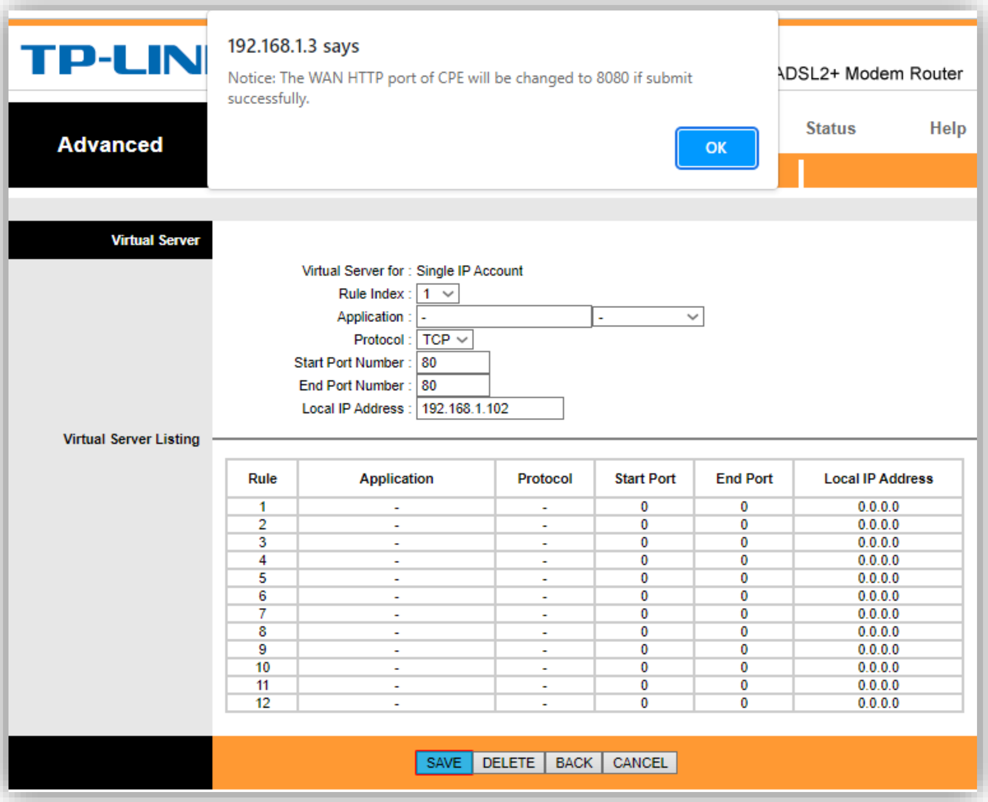
Local IP Address : 192.168.1.102

Virtual Server Listing

Rule	Application	Protocol	Start Port	End Port	Local IP Address
1	-	-	0	0	0.0.0.0
2	-	-	0	0	0.0.0.0
3	-	-	0	0	0.0.0.0
4	-	-	0	0	0.0.0.0
5	-	-	0	0	0.0.0.0
6	-	-	0	0	0.0.0.0
7	-	-	0	0	0.0.0.0
8	-	-	0	0	0.0.0.0
9	-	-	0	0	0.0.0.0
10	-	-	0	0	0.0.0.0
11	-	-	0	0	0.0.0.0
12	-	-	0	0	0.0.0.0

Figure83. Configuration

After we hit save a message will be displayed we click on OK.



192.168.1.3 says

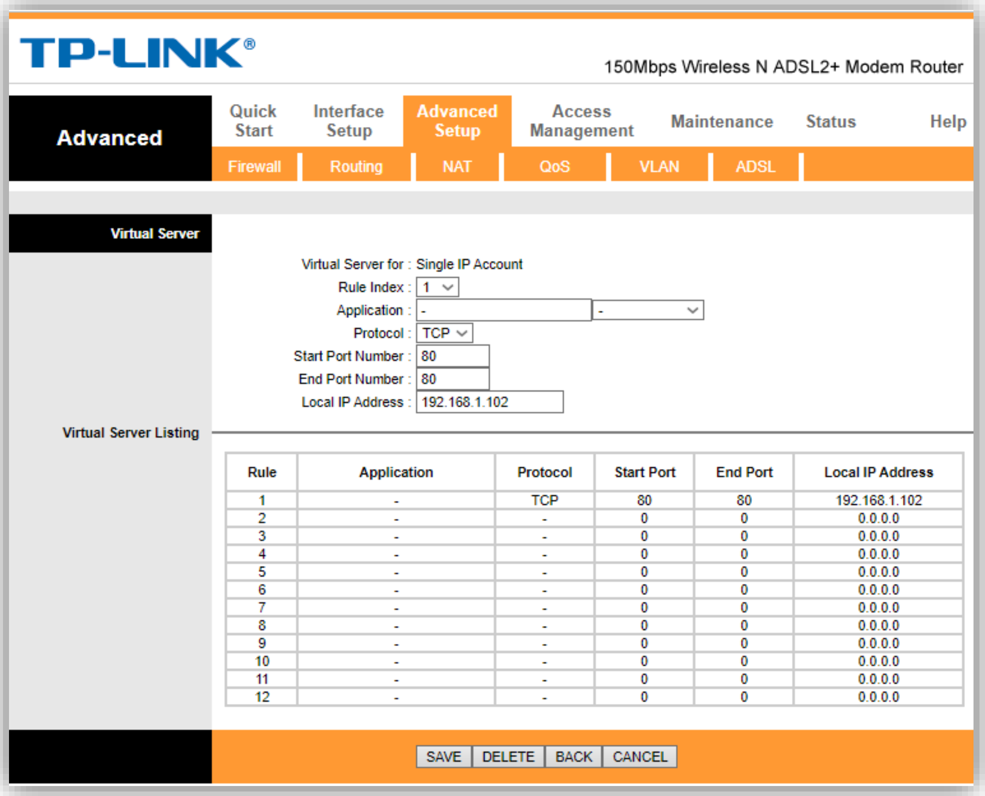
Notice: The WAN HTTP port of CPE will be changed to 8080 if submit successfully.

Virtual Server for : Single IP Account
Rule Index : 1
Application : - | -
Protocol : TCP
Start Port Number : 80
End Port Number : 80
Local IP Address : 192.168.1.102

Rule	Application	Protocol	Start Port	End Port	Local IP Address
1	-	-	0	0	0.0.0.0
2	-	-	0	0	0.0.0.0
3	-	-	0	0	0.0.0.0
4	-	-	0	0	0.0.0.0
5	-	-	0	0	0.0.0.0
6	-	-	0	0	0.0.0.0
7	-	-	0	0	0.0.0.0
8	-	-	0	0	0.0.0.0
9	-	-	0	0	0.0.0.0
10	-	-	0	0	0.0.0.0
11	-	-	0	0	0.0.0.0
12	-	-	0	0	0.0.0.0

Figure84. Port forwarding

Now we have finally forwarded our port.

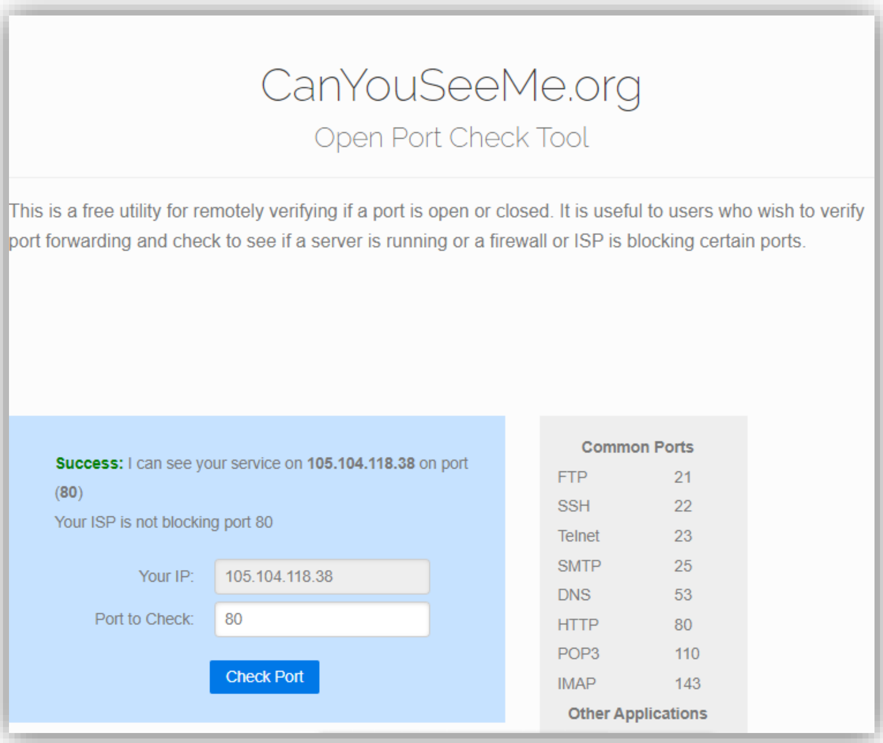


Virtual Server for : Single IP Account
Rule Index : 1
Application : - | -
Protocol : TCP
Start Port Number : 80
End Port Number : 80
Local IP Address : 192.168.1.102

Rule	Application	Protocol	Start Port	End Port	Local IP Address
1	-	TCP	80	80	192.168.1.102
2	-	-	0	0	0.0.0.0
3	-	-	0	0	0.0.0.0
4	-	-	0	0	0.0.0.0
5	-	-	0	0	0.0.0.0
6	-	-	0	0	0.0.0.0
7	-	-	0	0	0.0.0.0
8	-	-	0	0	0.0.0.0
9	-	-	0	0	0.0.0.0
10	-	-	0	0	0.0.0.0
11	-	-	0	0	0.0.0.0
12	-	-	0	0	0.0.0.0

Figure85. Port forwarding

Step 4: we go to www.canyouseeme.org and check if the port is open



CanYouSeeMe.org

Open Port Check Tool

This is a free utility for remotely verifying if a port is open or closed. It is useful to users who wish to verify port forwarding and check to see if a server is running or a firewall or ISP is blocking certain ports.

Success: I can see your service on **105.104.118.38** on port (**80**)
Your ISP is not blocking port 80

Your IP: 105.104.118.38

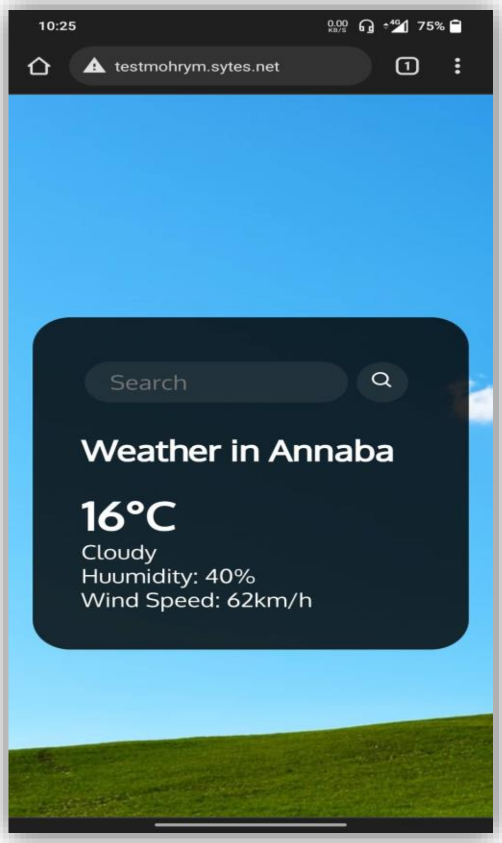
Port to Check: 80

Common Ports	
FTP	21
SSH	22
Telnet	23
SMTP	25
DNS	53
HTTP	80
POP3	110
IMAP	143

Other Applications

Figure86. CanYouSeeMe Test

Now our ISP is no longer blocking port 80. We go to our browser and type our domain name



Weather in Annaba

16°C

Cloudy
Huumidity: 40%
Wind Speed: 62km/h

Figure87. Test WAN

3.15. Discussion:

Webpages are now a platform and a medium for almost every domain, but web servers on the other hand are pretty expensive especially if you want a special server that will host only your website and has reliable, high performances. However our server will save a good amount of money, particularly if the site does not manage a lot of data and people. The cost will be symbolic, considering that we have not spent any DA for services such as Port Forwarding and no-IP which are paid services. As for the no-IP, it changes every two to seven days, so we must open a disposable IP account, and therefore we must renew the domain every month. That was one of the problems we have faced during the making of this web server. Another issue we have encountered was the blocked ports, but we managed to find a solution by forwarding a port number that was not blocked by our ISP.

3.16. Conclusion:

In conclusion, we can say that we have successfully built a web server at the lowest cost possible, and that all issues concerning the address and ports were resolved, with the server being tested both inside and outside the network. Also, we have created a functioning and a responding website that displays data which it gets from an API. Finally we finish the task by having the server run automatically when it is turned on.

General Conclusion

Conclusion

During the making of this dissertation, we have relied on several modules that we have studied in previous years, including:

- Routage IP
- Technologies du web
- Réseaux haut débit
- Recherche documentaire et conception de mémoire

Also we have faced cases that can only be theoretically correct but outside the ideal case of networks it is completely different. We have seen this with closed ports and the dynamic address. In addition to that there were some issues that are somewhat outside the networks domain, like programming and working with Arduino IDE and configuring the board and the Ethernet Shield. But thanks to the thorough research and the study of all the possibilities, we were able to find solutions without resorting to additional expenses and maintain good performance.

We began by preparing the required programs and configuring our Arduino UNO board as well as the Ethernet Shield, then we built the website from the ground up and copied it to the SD card. We also used several services to overcome the network obstacles, and we completed the job by running our server successfully without any problem.

In conclusion, a webserver can be programmed and configured for almost every domain application and it is possible for us to build one at the least total price, thanks to the services like no-IP and Port Forwarding that helped overcome difficulties, but that was only valid for a short period of time so in order to have a permanent domain name for example you will have to purchase a subscription.

Bibliography

[1] ptgmedia.pearsoncmg.com what is a web server

[2]Chapter 1 course: introduction to the web by Doctor Hafs.

[3] Les Serveurs by Guillaume Burel Nancy University 2008-2009

[4] What is a webserver from developer.mozilla.com?

[5] Introduction to web servers by SAN DIEGO COMMUNITY COLLEGE DISTRICT

[6] www.uotechnology.edu.iq by Ahmed Saeed

[7] Internet technologies webservers pdf from tutorialspoint.com

[8] What is Arduino, how it works by Circuit Schools Staff May 30, 2020

[9] Leo Louis, Department of Electronics and Communication Engineering, Gujarat Technological University, Ahmedabad, India

[10] International Journal of Control, Automation, Communication and Systems

[11] (IJCACS), Vol.1, No.2, April 2016

[12] Arduino Succinctly, By Marko Švaljek, Foreword by Daniel Jebaraj, by Syncfusion Inc.

[13] Suresh R. Borkar, in LPWAN Technologies for IoT and M2M Applications, 2020

[14] What is server Palash mondalFebruary 12, 2022?

[15] By Andy Myers, Electrical and Computer Engineering at Carnegie Mellon University December, 1998

[16] Arduino IDE from www.arduino.cc Apr 24, 2022

[17] Visual Studio IDE For the latest documentation on Visual Studio 2017 RC

[18] The Book of Visual Studio .NET—A Guide for Developers by Robert B. Dunaway San Francisco

[19] Technical Bulletin: Port forwarding BURK technology july 2010