



**An-Najah National University**

**Faculty of Engineering and Information Technology**

**Telecommunications Engineering department**

**Graduation Project 2**

**Colorful Musical Dancing Fountain**

Submitted in partial fulfillment of the requirements for Bachelor degree in

Telecommunication Engineering

**Supervisor by: Dr. Saed Tarapiah**

**Prepared by:**

Isra Sarrawi 11820562

Tuqa mosallam 11819474

Tabarak Salmiyah 11820985

Raghad Bdear 11821417

Academic year 2022 / 2023

## Acknowledgments

First of all, thanks to the families for all the support they have provided for us. Thanks, the honorable Dr. Saed Tarapiah who supervised this project from the beginning. Thanks to anyone who helped to complete this project, whether with information or in any way.

## Disclaimer Statement

Students from An-Najah National University's Faculty of Engineering and IT's Telecommunications Engineering Department wrote this study. It may have grammatical and content mistakes because it has not been amended or corrected, save from editorial adjustments, as a consequence of the assessment. The opinions expressed in it, along with any conclusions and advice, are all purely the students. An-Najah National University disclaims all liability and responsibility for the results of using this study for purposes other than those for which it was commissioned.

## Table of Contents

Acknowledgments .....	ii
Disclaimer Statement .....	iii
Abstract .....	vi
Chapter1: Introduction.....	1
1.1: Overview about fountains .....	1
1.2: Fountains Progress timeline .....	2
1.3: Current situation on the Market.....	3
Chapter2: Literature Review .....	4
2.1: Literature Review and Related Work.....	4
Chapter3: Component and Programming .....	8
3.1: Fountain Main Hardware Component .....	8
3.2: Fountain software Component .....	14
3.3: Fountain Design .....	15
3.4: Software Programming .....	20
3.5: Block Diagram for Fountain System .....	21
3.6: Flow Chart .....	22
3.7: Relationship between music and water .....	24
Chapter4: Methodology .....	25
Chapter 5: Earlier course work .....	26
Chapter 6: Conclusion .....	27
References.....	28
Appendix .....	30

## List of Figures

Figure1: Fountain history timeline .....	2
Figure2: Different Types of Nozzles .....	8
Figure3: Different Types of Water Pumps .....	9
Figure4: Different Types of H-Bridges .....	10
Figure5: Different Types of Servo Motors .....	11
Figure6: Different Types of Microphones Sensor .....	12
Figure7: RGB digital led .....	13
Figure8: Different Types of Arduinos .....	14
Figure9: Our Type of Nozzle we used .....	15
Figure10: Dry pump .....	16
Figure11: H-Bridge .....	16
Figure12: Servo motor, MG996R .....	17
Figure13: Arduino Mega .....	18
Figure14: Jumper's wire .....	19
Figure15: Power supply .....	19
Figure16: Arduino program .....	20
Figure17: Block Diagram .....	21
Figure18: Water pump control part .....	21
Figure19: Flow chart .....	23
Figure20: Our Fountain test .....	25

## Abstract

The flawless fusion of music, light, and water in a Dancing or Musical Fountain is one enthralling example of how art has always played a significant role in human society. The goal of this study is to build an autonomous musical fountain system that uses a microcontroller unit (MCU) to extract, amplify, and manipulate audio signals from a laptop. For additional processing, the MCU transforms analog signals into digital signals using an analog-to-digital converter (ADC).

An Arduino Mega 2560 microprocessor, a breadboard, water pumps, an H-Bridge, nozzles, a control servo motor (MG996), and an RGB digital LED strip are just a few of the parts that make up the musical fountain system. The C programming language is used to program the Arduino Microcontroller (Mega 2560) to produce enthralling water shows and coordinate water and lighting scenarios with music files.

The goal of this capstone project report is to provide a thorough overview of the development of the automated musical fountain system. The study will go into the system's technical specifics, including the hardware parts, how they are connected, and the programming methods used. The paper will also showcase the several creative arrangements and sequences made possible by the system's capabilities.

## Chapter1: Introduction

### 1.1 Overview about fountains

When it comes to outdoor furniture, there has been a noticeable change in people's choices in recent years. In order to design comfortable, distinctive, and livable rooms, aesthetic sense, comfort, and attention to detail have become essential. Fountains, a typical component of outdoor furniture, have also been impacted by this trend. People now desire greater visual impact and appeal, and a fountain's plain water flow is insufficient.

Dancing fountains, also known as musical fountains, have been designed to satisfy this demand by designers who specialize in fountain design. These fountains have light and water shows that are timed to a recorded song. This ground-breaking idea has been quite popular, creating a market niche for fountains that is in high demand.

In the past, fountains have been used for a variety of things, including evaporative cooling in arid climates, enticing water shows, serving as focal points in urban areas, muffling traffic noise, and creating calming noises, among other things. [1]

Some of the most amazing fountain designs have included elements like unexpected water effects, moving statues propelled by water flow, dynamic water jets, spectacular water and fire displays, and incorporating music and lighting effects.

A fountain's fundamental idea is to manage water to produce enthralling water displays. The water source, display effects, and the pump are the three primary things that need to be taken into account. The first step is to locate a water source that is accessible. The fountain needs to be artificially fed and refilled from a central conduit when a natural water source is not available. The next step after establishing the water source is to start the water moving and choose the effects you want to see in the water display. The selection of the proper pump comes after assessing the water jets' display effects and pattern. [2]

Previously restricted to using water jets to create patterns, fountains have developed into multi-sensory shows that use music, light, and other elements. These intriguing displays, which operate as sources of entertainment and interest, can be found in parks, buildings, and offices. This idea is expanded upon by a musical fountain, which synchronizes light and water jets with the tempo and pitch of the music to produce a mellow and visually spectacular sight.

## 1.2: Fountains Progress timeline

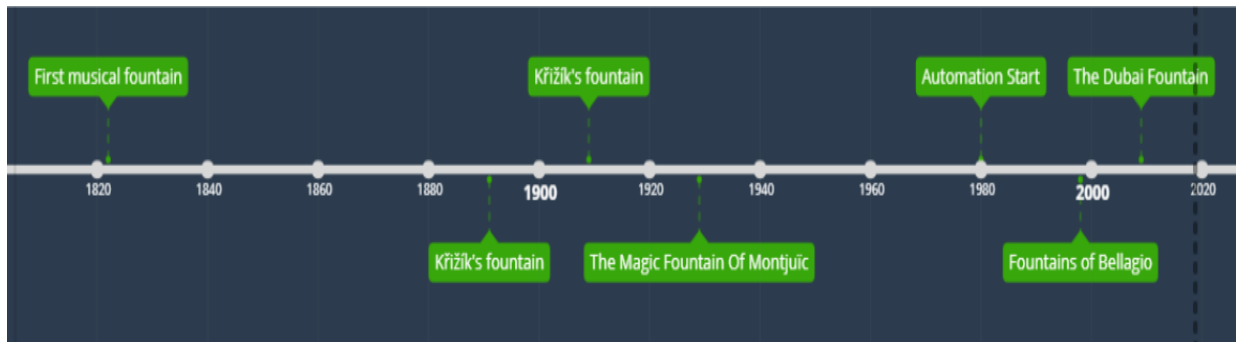


Figure1: Fountain history timeline

Years	Descriptions
1820-1822	The inaugural musical fountain was constructed by Peter Bodor in Marosvásárhely (now Târgu Mureș, Romania) and operated using hydraulic mechanisms.
1891	František Křižík created Křižík's fountain for the World Exhibition. The fountain featured a bottom plate with 1300 multicolored reflectors and water circuits comprising nearly 2 kilometers of pipes and approximately 3000 nozzles.
1909	Denver, Colorado's Prismatic Fountain dazzled viewers with its eleven columns of vividly colored light and dynamic water patterns. F.W. Darlington, a pioneer in electrical fountain control and water design, required an operator to modify water effects and lighting.
1929	The Magic Fountain of Montjuic in Barcelona incorporated music into its light show during the 1980s, synchronizing the fountain's display with the rhythm of songs.
1980	The revolution of fully automated Dancing Fountains commenced primarily in the United States.
1998	The Fountains of Bellagio, known for their grandeur, boasted over 1,200 nozzles and more than 4,500 lights, enabling coordinated fountain displays. The construction cost of the fountains was estimated to be \$40 million.
2009	The Dubai Fountain emerged as the largest fountain in the world, with a staggering price tag of USD \$218 million.
2012	Kolkata witnessed the inauguration of India's most captivating musical fountain, the CESC Fountain of Joy.

Table1: Evolution of Musical Fountains.

Quite many musical fountains, what we can observe now, were created in mid or early ages of 20th century. Most of them were lacking musical or lightning part in the beginning and were redesigned already at the end of 20th century.

### **1.3: Current situation on the Market**

Almost all dancing fountains in the world, and especially the most famous ones, are using pre-programmed style of choreography for dancing fountains. There are many dancing fountain control systems in the market, but most of them are proprietary development. There is no common standard on the market.

Leading key players in the Dancing Fountain market are - Delta Fountains, Hall Fountains, Suzhou Gold Ocean, Gzfenlin, Fontana Fountains, Safe-Rain, OASE Living Water, Aquatique Show, Lumiartecnia Internacional, PREMIERWORLD Technology Ltd., Yixing Sea Fountain Equipment.

#### **Most common installation**

Even if there is no common standard in the Dancing fountain market, the components of the system are more or less the same from one system to another.

#### **Most common installation for dancing fountains includes:**

Pump, Nozzles, Electronic Valves, Controllers, Computer with Fountain control software, Lights.

## Chapter2: Literature Review

### 2.1 Literature Review and Related Work

In a study by [3], the design of a musical fountain was presented, utilizing various components including an Arduino board, five transistors, a sound detector sensor, a water pump, solenoid valves, a breadboard, jumper wires, resistors, and LEDs. The Arduino board was employed to control the water valves and create captivating effects synchronized with music, such as timed pulses of water. Transistors served as switches to activate and deactivate the fountain based on the music, with each LED lighting up in sync. A sound detector sensor was utilized to detect different sounds. Water pumps were responsible for fluid movement, while solenoid valves regulated the opening of fluid flow. Although the circuit connection details were provided, the code was unfortunately not available.

Another search [4], focused on dancing fountains that could synchronize water and lighting scenarios with mp3 music files. The system employed a PIC microcontroller, a decoder, a Serial Control Interface (SCI) module, water pumps, LED-based lights, drivers, an audio amplifier, audio outputs, water jets, anemometers, and a tank level sensor. The PIC microcontroller reproduced music files corresponding to specific scenarios and drove the system in a synchronized manner. The decoder was responsible for playing the music files, while the SCI module facilitated the PIC's interfacing and decoding processes. The search included an architecture scheme of the control system developed for musical fountains, as well as a screenshot of the firmware file.

In another paper [5], the design and construction of a musical water fountain system driven by Arduino using Fourier Transform technology was presented. Audio signals were sourced from devices such as MP3 players, smartphones, and laptops. These signals were first amplified using an LM386 IC and then processed through the Arduino's analog input using Fourier Transform processing. Digital output pins were utilized to control the DC pump motors, which enabled the creation of synchronized lighting and water spray effects based on various frequency bands of the music. The study analyzed the system's response for different audio frequency ranges and emphasized the control of the pump motors. The fountain model was designed for domestic usage.

In search [6], Master's thesis main objective was to plan and prototype the self-choreographed dancing fountain system, which is reacting on real-time audio signal and automatically synchronizes water movements with music to achieve the artistic effect.

After several iterations and major change of concept in the middle of research it is possible to say that this study has measurable outcomes:

- It was proved that the color organ approach in self-choreographed dancing fountain system could be implemented and is fully operational from technical point of view, but has no much artistic value.
- Considering all the artist requirements, the top-level architecture of the self-choreographed dancing fountain system was planned. The major functionality of the system is audio pattern recognition (with use of different in-house or external technology) and hydraulics control pattern library to synchronize the incoming audio signal with water movements.
- Prototyping of the product with minimum viable functionality was initiated and several example modules were developed. At the same time, the possibility of using external libraries for specific system modules was analyzed and specific technology presented.
- One of the important outcomes of this study is the understanding that it is not possible yet to fully rely on machines in some areas. Even if the fountain will work autonomously without human interaction, all the pre-configuration of the fountain behavior has to be done by human.
- Embed the audio fingerprinting external library into the system container.
- Involve an external expert to develop the machine learning part of audio pattern recognition module.
- Enrich the audio pattern library with defined by artist patterns per each audio pattern recognition technology.
- Find or develop the real-life hydraulic part of the dancing fountain and add control patterns defined by artist to the library.

In a review conducted in [7], various attempts to create dancing fountains were summarized, highlighting the necessary steps to achieve the technical and commercial development of a dancing fountain. These steps included defining the physical setting of the fountain, creating 3D dancing water fountain animations, and incorporating DMX technology in the control panel design. Important considerations for a musical fountain project were also outlined, such as determining the number, type, location, height, and departure angle of water jets, as well as selecting between dry pumps or submersible pumps and solenoid valves. The choice between DMX and PLC protocols for the control panel design was also discussed, with DMX being preferred for musical fountains due to its integration with computers or tablet PCs.

In another search [8], a fountain system consisting of nine nozzles, a microcontroller (Basic Stamp 2), pumps, a radio frequency (RF) receiver, and an RF remote was described. The outlets were arranged in a three-by-three matrix, with small submersible pumps attached to plastic tubing. A squared-shaped plate was used to hold the tubing in an upright position. The Basic Stamp 2 microcontroller, powered by a 12VDC adapter, controlled the system's operations. The wet side of the fountain included a plastic container acting as the basin. It was noted that the programming language, PBASIC, required for the microcontroller might pose some difficulty and would need to be studied thoroughly.

In an article [9], FIR filters were utilized to decompose the original sound signal into four signals, each representing a narrow band of frequencies. These filter outputs were used to drive variable speed drives (VSDs) that controlled the speed of water pumps and the intensity of colored lamps. The analog control signals produced by the FIR filters were applied to the VSD unit to regulate the frequency and magnitude of the supplied voltage, thereby controlling the water jet heads and light intensity. The study employed a TI TMS320C31 processor, ADC, and DAC with a high sampling frequency of 44.1 kHz to ensure audio quality. Previous works mentioned in the article included a water fountain that displayed letters using a BS2 microcontroller, an intelligent musical fountain authoring system based on Bayesian networks, and the design of a real-time audio equalizer and fancy fountain control system using PLC and AC drive.

In a study conducted in [10], the addition of a motor in the form of a fountain was mentioned to enhance the dynamic movement of the fountain. The following components and modules were discussed:

1. Arduino Mega 2560: This microcontroller board based on the ATmega 2560 is commonly used for control purposes. It simplifies working with microcontrollers by using simplified C language and Arduino libraries. It can be integrated with Android, a Linux-based operating system used on mobile phones, allowing users to build their own applications.
2. Module HC-05: The HC-05 module enables wireless serial communication by converting the serial port to Bluetooth. It facilitates communication between devices.
3. Light Dependent Resistor (LDR): LDR is a photosensitive electrical component that varies its electrical properties based on the light it receives. It is used for light sensing applications.
4. Real Time Clock (RTC): RTC is an electronic component used as a timer. It accurately calculates and stores the time in real-time. It can be used in various applications that require precise timing.
5. Water Pump: A motor pump used to raise fluid from a lower to a higher level. It operates by reducing the vacuum pressure to create suction and raise the fluid.

6. MP3 Shield: An Arduino module used to emit group voice commands in MP3 format. It plays audio based on predefined coding conditions.
7. Loudspeaker: A transducer that converts electrical signals into sound frequencies using membrane components. It produces sound waves that can be heard.
8. Servo Motor: A motor with three wires used for precise control. It is commonly used for controlling movement in various applications.

In a research paper [11], an automatic musical fountain system controlled by audio frequency analysis using the FFT algorithm was presented. The system used audio signals from various sources, amplified them, and processed them using a Microcontroller Unit (MCU). The MCU converted analog signals to digital signals using an Analog-to-Digital Converter (ADC) and employed the FFT algorithm for audio frequency analysis. The system included modules such as a Power Supply Module, Signal Amplification Module, Audio Signal Filtering Module, ADC Module, Display Module, Delivery Module, and an MCU.

Another research paper [12], described the design of an intelligent musical fountain authoring system (Imfas) that generates scenarios for musical fountain shows based on any piece of music. Imfas utilized a Bayesian network trained on sample scenarios and employed audio analysis techniques to synchronize the music with changing water shapes. It also allowed programmers to modify automatically generated scenarios through a simple interface. The water pumps, solenoid valves, lights, and nozzle movement equipment were generally controlled by a programmable logic controller (PLC).

In a different study [13], a simple and inexpensive controller for a musical fountain was designed, featuring real-time processing and automatic control. The system included a user interface to select audio input from various sources, pre-processing and equalizing the audio signal, digital signal processing using an ADC and DSP, and the application of the Fast Fourier Transform (FFT) to transform the signal from the time domain to the frequency domain. The obtained signal was transmitted using the DMX 512 protocol for controlling RGB LED lamps and electro-valves.

In a paper [14], a falling water system for an indoor fountain with special effects was presented. The system utilized a microcontroller (Arduino MEGA 2560) and was programmed to create and sequence interesting arrangements of water displays. The hardware system included relays, water valves, power supply, and an iron structure for presenting shapes. The software part consisted of a Visual Basic program on a PC device and C language on the Arduino device to control the falling water system. The experimental results were tested with different alphabetic words and graphical shapes.

## Chapter3: Component and Programming

### 3.1: Fountain Main Hardware component

#### 3.1.1 Nozzles

A nozzle is a tool used to change the characteristics of a fluid flow as it leaves (or enters) a closed chamber or conduit, particularly to enhance velocity.

A nozzle can be used to guide or modify the flow of a fluid (liquid or gas), and it is frequently a pipe or tube with a variable cross-sectional area. Nozzles are widely used to control the stream that emerges from them in terms of flow rate, speed, direction, mass, form, and/or pressure. The fluid's pressure energy is lost as its velocity rises in a nozzle.

There are several types of nozzles depending on the shape of the water that comes out of them, such as: Jet, Propelling, Swirl, Rotary, Spherical and Spray.



Figure2: Different Types of Nozzles.

### 3.1.2 water pumps

Fountain pumps are made to provide water flow to fountains and other devices used in residential, commercial, and industrial applications for aesthetic and aeration purposes. Fountain pumps can be set up for use indoors or outdoors, and they can be distinguished by the size of the pool or pond, as well as by whether they are being used for decorative or practical purposes. Some fountain pumps, which can be used in conjunction with a well pump to produce unique garden or landscape elements like waterfalls, are even solar-powered.

The majority of fountain pumps are centrifugal pumps, which pressurize water by employing centrifugal force generated by the rotation of an impeller. A fountain pump works similarly to a sump pump, except instead of pumping water out of the pond or reservoir, it feeds it back in.

There are two type of water pumps depending on where will put the pump, such as: Submersible or dry pump [15].



Figure3: Different Types of Water Pumps.

### 3.1.3 Motor driver, H-Bridge, PWM Bridge.

Motor driver controls the speed of DC motor in the Water pump, which follows with changes of the water height. Specifications:

- Work mode: H bridge drive (double)
- Logical voltage: 5V
- Logical current: 0mA-36mA
- Storage temperature: - 20 °C to + 135 °C
- Weight: 30 g
- Main control chip: L298N
- Drive voltage: 5v-35v
- Driving current: 2A (MAX single bridge)
- Maximum power: 25W
- Peripheral dimensions: 43\*43\*27m

When driving voltage is less than 12V, on-board 5V power supply can be used to power the Microcontroller or other board. When the driving voltage is higher than 12V, 5V power supply can't be used. Use board jumper to control these modes. [16].

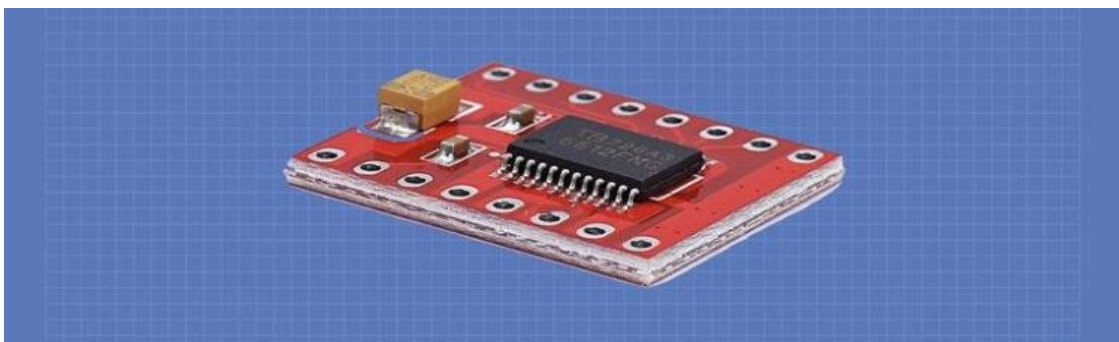
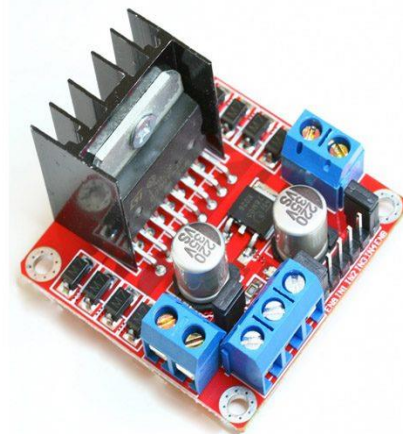
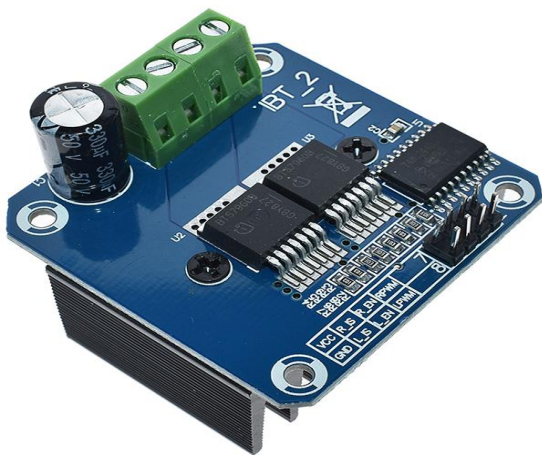


Figure4: Different Types of H-Bridges.

### 3.1.4 Servomotor

The rotary actuator or linear actuator known as a servomotor (or servo motor) enables precise control of angular or linear position, velocity, and acceleration. It consists of an appropriate motor connected to a position feedback sensor. It also needs a rather sophisticated controller, frequently a special module created just for use with servomotors.

Although the word "servomotor" is frequently used to describe a motor appropriate for use in a closed-loop control system, servomotors are not a particular sort of motor.

Power, ground, and signal are the three wires that servo motors have. Usually red, the power cable needs to be attached to the Arduino board's 5V pin. The ground wire, which is normally brown or black, needs to be attached to a ground pin on the circuit board. The PWM pin on the board should be connected to the signal pin, which is commonly yellow or orange. [17].



Figure5: Different Types of Servo Motors.

### 3.1.5 microphone sensor

You're probably referring to a microphone module or sensor when you say "microphone sensor," which is used to identify and transform sound waves into electrical signals. In many different applications, including audio recording, voice recognition systems, communication devices, and more, microphone sensors are frequently employed.

- Types of Microphones:

Electret Condenser Microphones (ECM), Dynamic Microphones, MEMS (Micro-Electro-Mechanical Systems) Microphones.

Here are some general characteristics about microphone sensors: Sensitivity, Frequency Response, Connection Interface.

- Applications:

Microphone sensors are used in a wide range of applications, including audio recording and broadcasting, telecommunication devices, voice-controlled systems (such as virtual assistants), hearing aids, surveillance systems, and more.

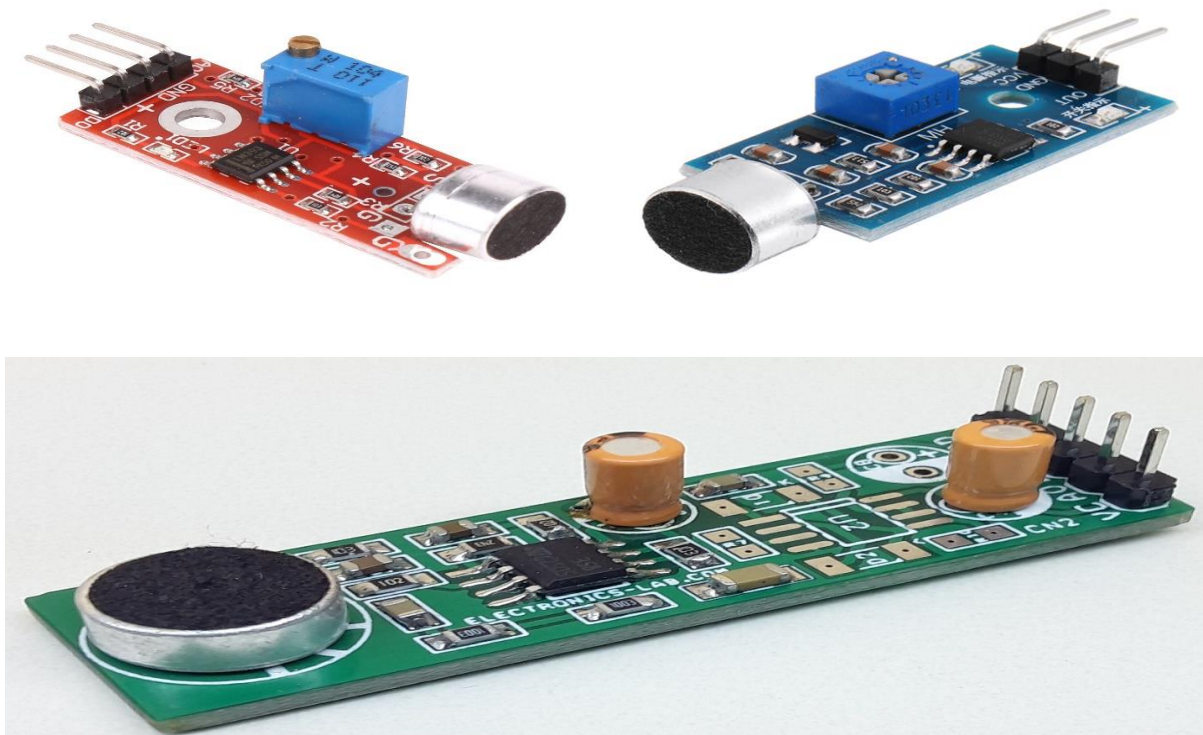


Figure6: Different Types of Microphones Sensor.

### 3.1.6 Serial RGB digital LEDs

Serial RGB LEDs are a form of LED module that enables independent control of each individual LED inside the module. They are also referred to as addressable RGB LEDs or individually addressable RGB LEDs. Compared to conventional RGB LEDs, this enables more sophisticated and dynamic lighting effects.

- Applications:

Serial RGB LEDs are widely used in applications where dynamic and programmable lighting effects are desired. They are commonly used in decorative lighting, signage, gaming peripherals, home automation, wearable electronics, art installations, and visual displays.

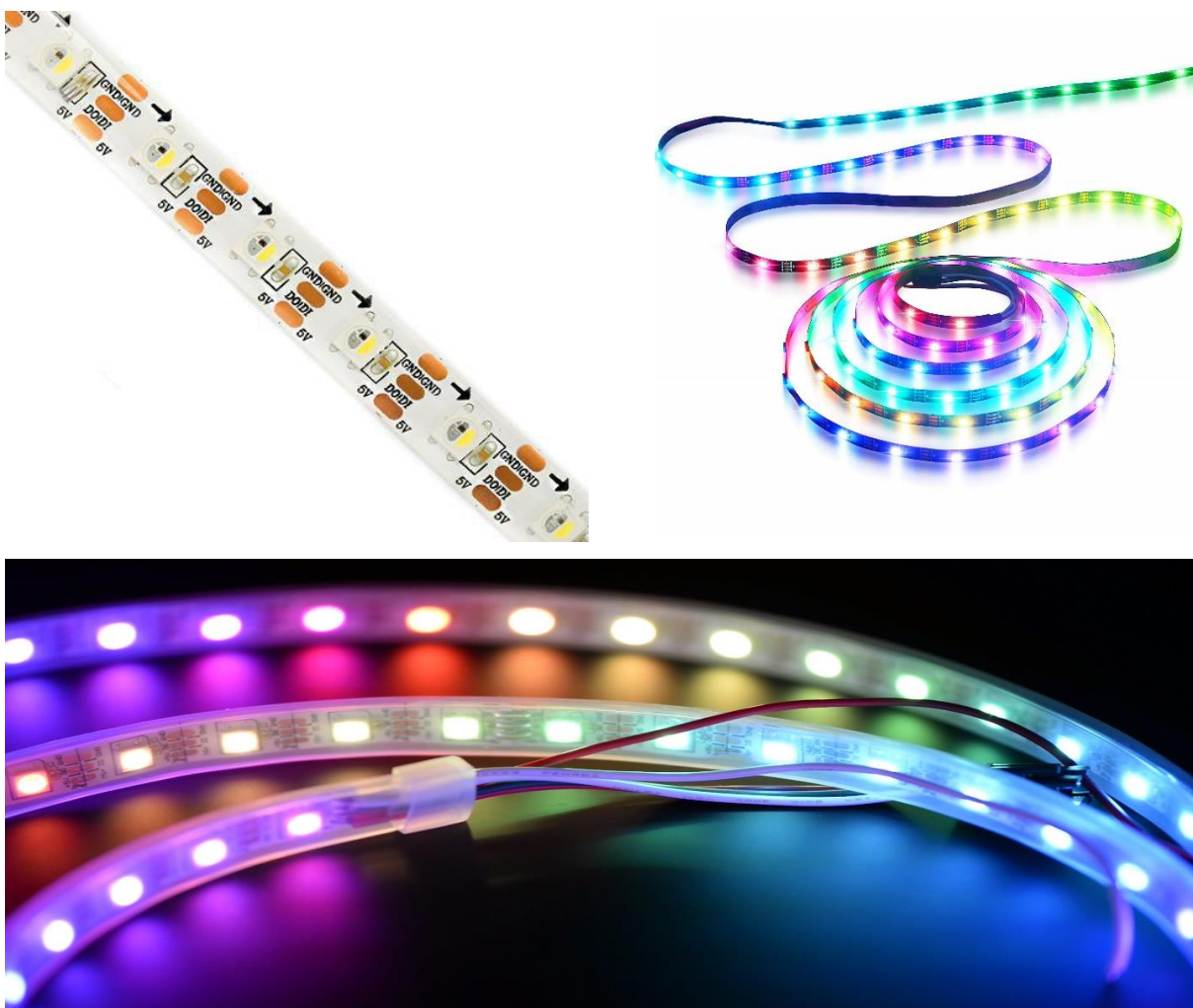


Figure7: RGB digital led.

## 3.2 Fountain Software Component

### 3.2.1 Arduino

The open-source Arduino platform is used to create electrical projects. With Arduino, you can write and upload computer code to a physical programmable circuit board (commonly called a microcontroller) using IDE (Integrated Development Environment), a program that runs on your computer.

You can just use a USB cable to upload fresh code to the Arduino instead of a separate hardware component known as a programmer.

Additionally, the Arduino IDE uses a condensed form of C, making programming simpler to learn. Last but not least, Arduino offers a standard form factor that separates the microcontroller's functionality into a more usable packaging [19].

There are many types of Arduinos like: Arduino UNO, Arduino Nano, Arduino Micro, Arduino Leonardo, Arduino PRO Mini, Arduino Mega2560 Rev3.

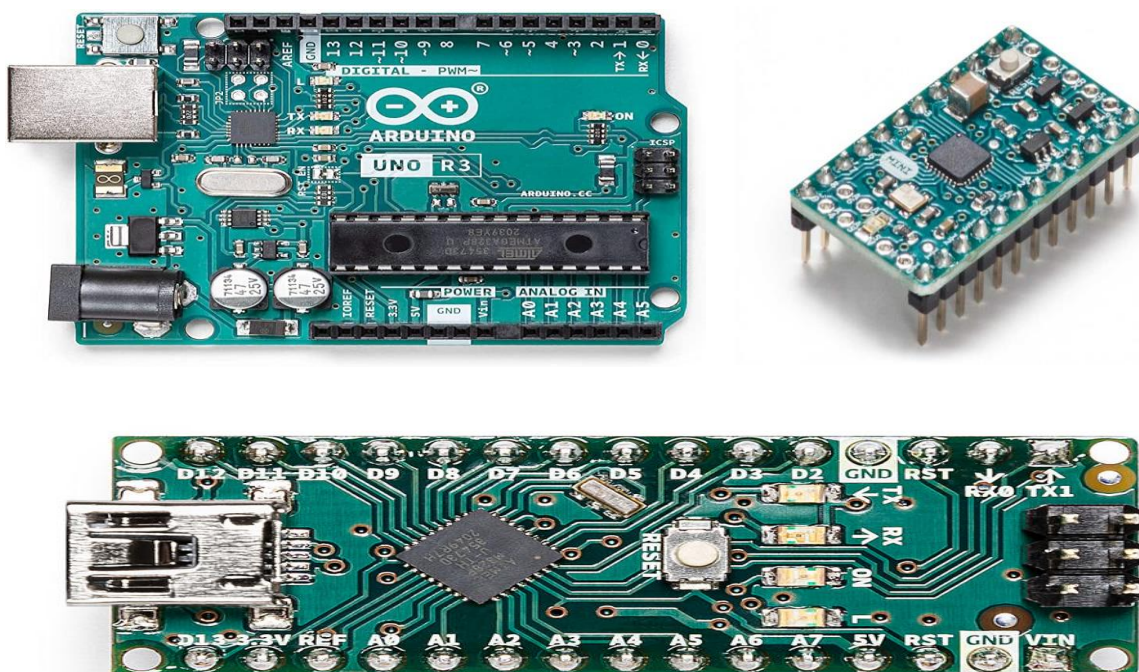


Figure8: Different Types of Arduinos.

### 3.3 Fountain Design

#### 3.3.1 shape and dimensions of the architectural fountain

In this project, we chose the shape of the fountain to be square in shape with a width of 35 cm, height of 20 cm and the thickness 0.5 cm.

#### 3.3.2 Number of Nozzles:

The number of nozzles that we decided is 3 nozzles.

As shown in figures:



Figure9: Our Type of Nozzle we used.

### 3.3.3 Number of Water Pumps and Type:

The number of water pumps that we decided is one pump.

In our project we will use one type of pumps: dry pump.

As shown in figure:



Figure10: Dry pump.

### 3.3.4 Number of H-Bridge(driver) and Type:

The number of Driver that we decided is one.

Every one driver contains 2 pumps.

As shown in figure:

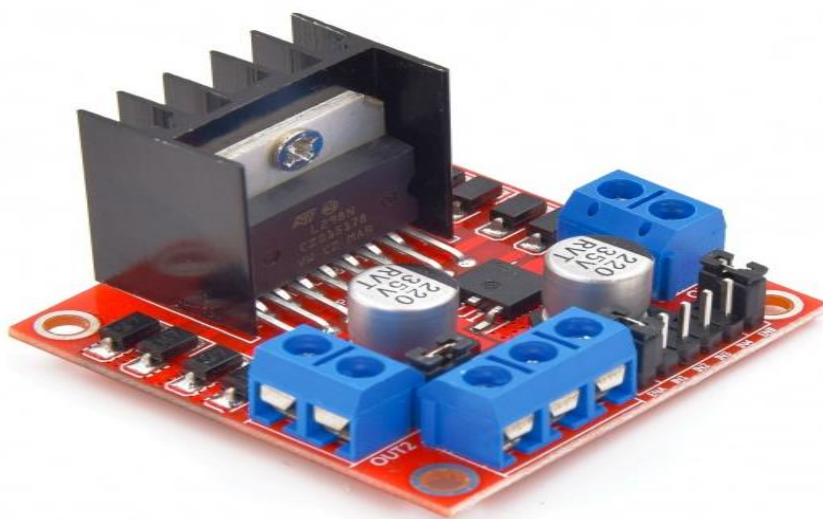


Figure11: H-Bridge, L298N chip

### 3.3.5 Number of Servomotor and Type:

The number of Servomotor that we decided is 3.

The Type we used: MG996R, so that it is waterproof.

As shown in figure:



Figure12: Servo motor, MG996R.

### 3.3.6 Number of Arduino and Type

The number of Arduino that we decided is one.

In our project we will use one type of Arduino: Arduino Mega 2560.

#### 3.3.6.1 Arduino Mega 2560

An electronic development board called the Arduino Mega 2560 is based on the Atmega2560 microcontroller. This board is a good choice for projects that need more GPIO pins and memory space because it has 16 analog inputs, 54 digital input/output pins (of which 15 can be used as PWM outputs), 5V operating voltage, 50 mA DC current for 3.3V pins, 20 mA DC current per I/O pin, 4 UARTs (hardware serial ports), 256 KB, of which 8 KB are used by bootloader flash memory, 8 KB SRAM Everything required to support the microcontroller is included in it [20] [21].

As shown in figure:

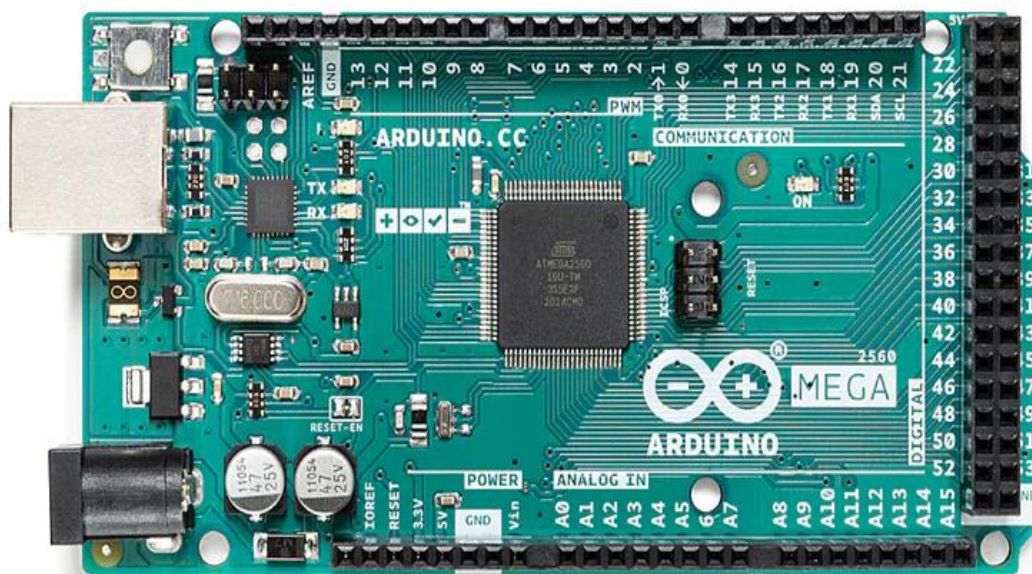


Figure13: Arduino Mega



### 3.4 Software Programming

In our project we used software Programming: C programing for Arduino.

The code we used shown in Appendix.

#### 3.4.1 C programming

Dennis Ritchie developed the general-purpose computer language C at Bell Laboratories in 1972. Despite being an ancient language, it is extremely popular.

Since the UNIX operating system was created using C, the two are closely related. It is extremely quick compared to other programming languages, such as Python and Java, and it is very adaptable; it can be used for both technology and applications [20].

The C programming language is a machine-independent language that is primarily used to develop numerous applications and operating systems like Windows, as well as more complex programs like the Git repository, the Oracle database, the Python interpreter, and games. It is regarded as a programming foundation when learning any other programming language. Examples of such applications include operating systems and other application software for computer architectures ranging from supercomputers to PLCs and embedded systems [21].



Figure16: Arduino program.

### 3.5 Block Diagram for Fountain System

In this section we shown how we design the fountain process and how we connected the component with each other.

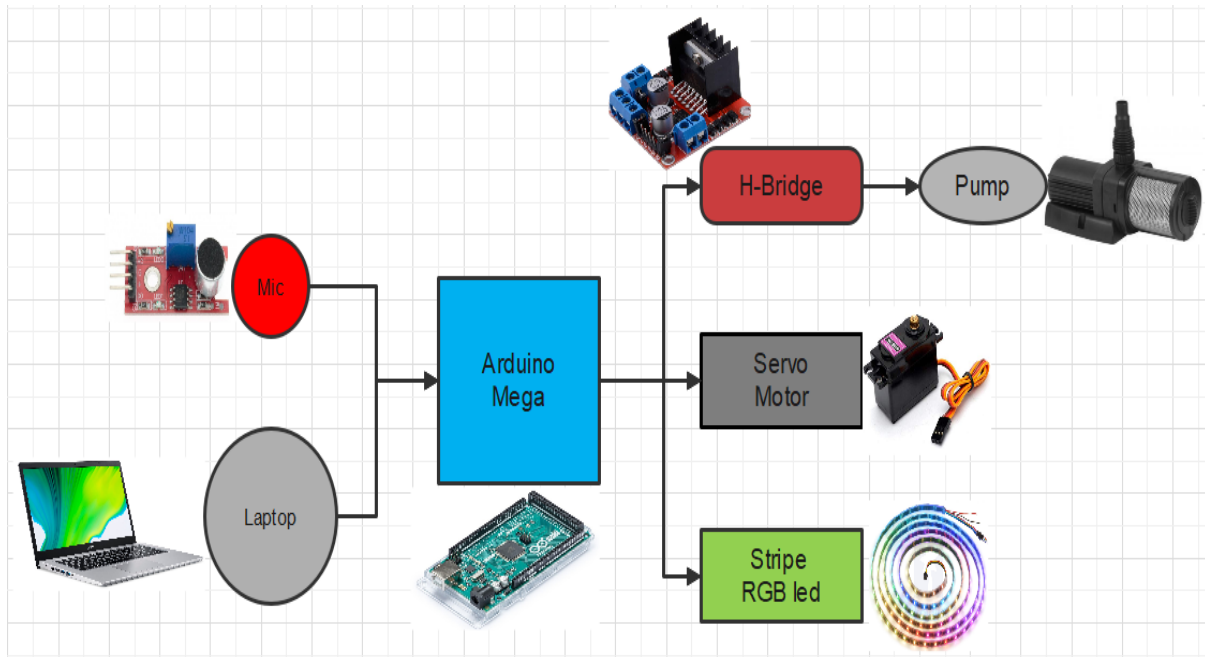


Figure17: Block Diagram

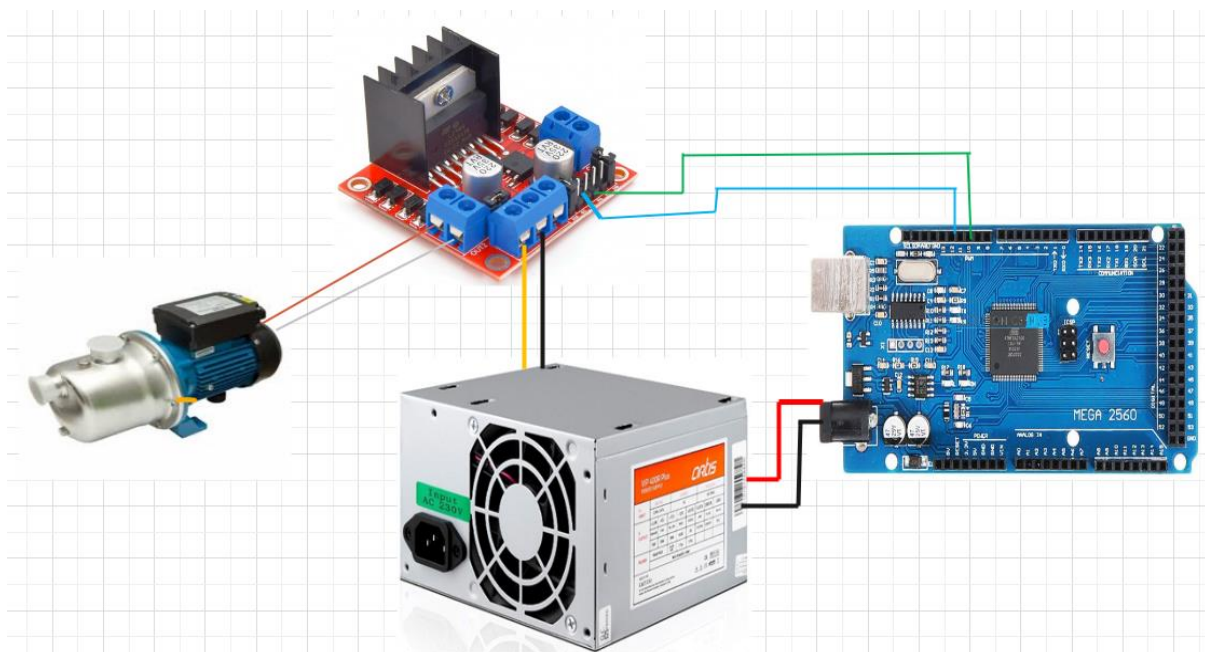
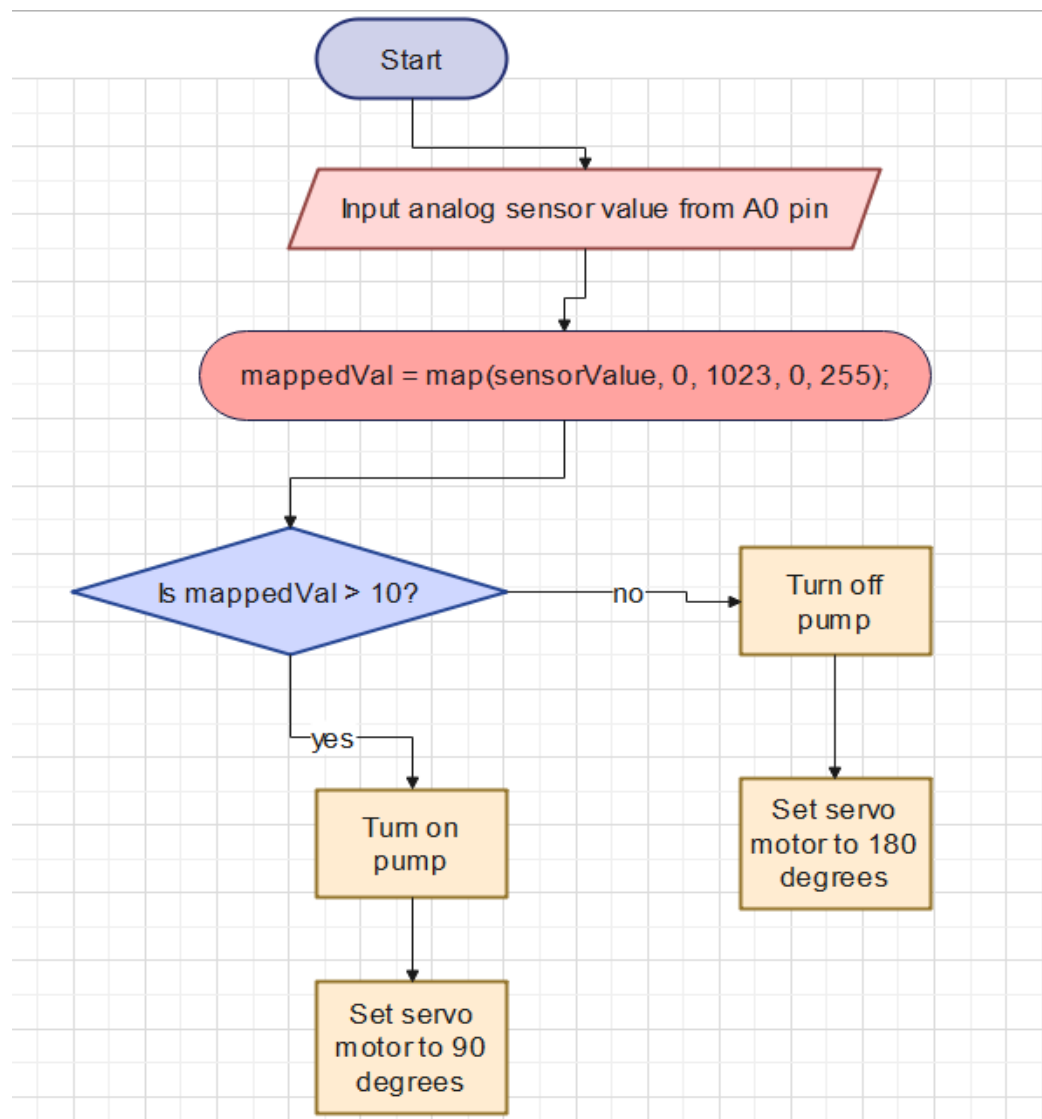


Figure18: Water pump control part

### 3.6 Flow Chart



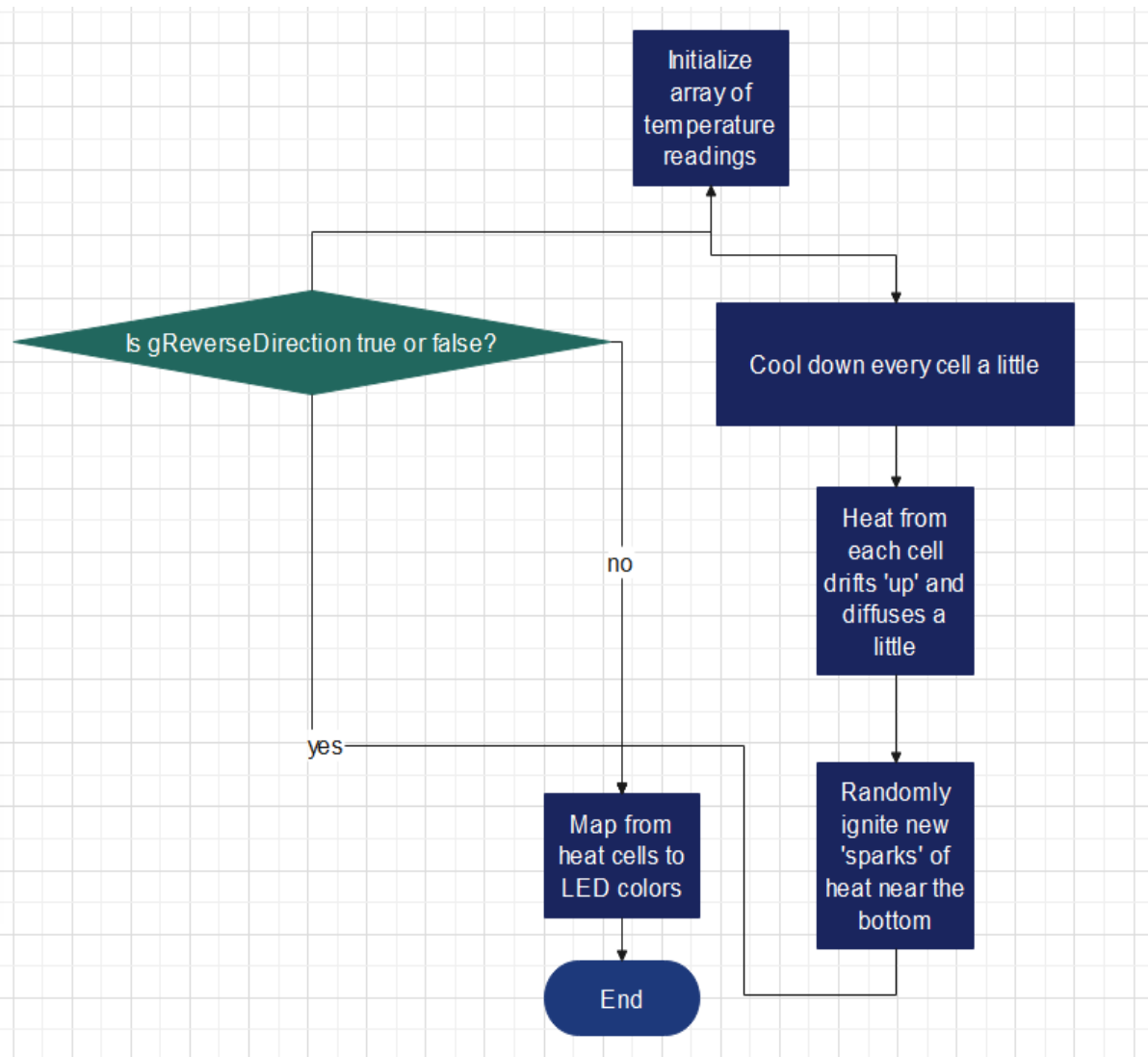


Figure19: Flow chart

### 3.7: Relationship between music and water

The relationship between music and the movement of water in a dancing fountain is achieved through a carefully orchestrated technological setup. The process involved capturing the sound of the music through a microphone, amplifying it, and then transferring it to a microcontroller unit (MCU). The MCU in this case, an Arduino Mega 2560 plays a crucial role in processing the audio signals.

The analog audio signals received from the microphone are converted into digital signals using an analog-to-digital converter (ADC) within the MCU. This digital representation allows for precise analysis and manipulation of the sound data. By programming the Arduino using C, you can create and sequence various arrangements of water shows, synchronized with the music files.

To translate the music into water movements, the MCU controls various components of the fountain instrumentation system. This system includes water pumps, an H-Bridge for controlling the pumps' direction and speed, nozzles for water dispersion and a control servo motor (such as the MG996) for additional water effects.

The MCU receives the analyzed audio data and interprets it to generate corresponding commands for the water pumps, H-Bridge, and servo motor. These commands coordinate the flow and pressure of water, as well as the positioning and movement of the nozzles. By precisely timing these actions based on the rhythm, tempo and intensity of the music, the water can rise, fall, twist and sway in harmony with the musical composition.

Additionally, the color-changing aspect of the fountain is accomplished using an RGB digital LED strip. The MCU also controls these LEDs, allowing them to change colors, fade or pulse according to the music.

In summary, the relationship between music and the movement of water in your dancing fountain is facilitated by the integration of various components and technologies. The MCU, along with the Arduino Mega 2560, serves as the central control system, receiving audio signals, processing them, and generating commands for the water pumps, H-Bridge, servo motor and RGB LED strip. This intricate coordination results in a mesmerizing display where the water and lighting effects beautifully respond and synchronize with the music, captivating audiences and creating a truly immersive sensory experience.

## Chapter4: Methodology

At the beginning of the project, we read several research papers related to the topic of our project and summarized them to see the difference between the researches, and then decided how we would design the fountain according to what we saw fit.

Then we began writing the report, dividing the tasks between us, and researching the topic of the fountain in detail.

We chose the pieces that we will use and their number based on what is best for the fountain, and we also chose the programming languages that we will use in the project (C language), and as a result, we will operate the fountain in terms of water movement and music.

We also made a block diagram for how the pieces are intertwined inside the fountain, the location of each part, the number and type of Arduino that we will use.

And we studied how to turn on the music and lights in the fountain, according to which the water moves, all this for the theoretical part.

As for the practical part, we purchased the necessary parts and calculated the cost, then we tried to interlock the pieces with each other and make a water network for the basin.

We had several problems at first, but we solved them thanks to God.

Then we wrote the necessary code to operate the fountain in terms of water movement.

We hope you liked it and that we did our job successfully.

### 5.1 Test Model



Figure20: Our Fountain test.

## Chapter 5: Earlier course work

Through the five years too many courses had been taken, some of them have been used in this project in direct way or not directly.

- English 102.
- English in Workspace contents were employed in writing professional report and preparing a perfect presentation.
- Electrical installation and safety system.
- C++ used to write the code on the Arduino to operate the fountain.
- microprocessors and microcontrollers used it to learn how to control the motors, RGB digital strip led, Arduino and pumps.
- Control systems to deal with pumps and drivers.

## Chapter 6: Conclusion

The project's primary goal was to enhance the nation's economic standing in the tourism sector. The dancing fountain plays a significant role in luring visitors to the area because it has a stunning aesthetic appeal in addition to a gorgeous lighting setup that blends in beautifully with the music.

The major goal of the project is to develop an integrated system that will process and divide the audio signal into different stages in order to account for water and illumination movements.

The students' degree of communication, preparation for the job market, and inventiveness in their area of specialization are all demonstrated in the project.

## References

- [1] S. A. I. D. Shakerin, "Microcontrolled water fountain: a multidisciplinary project.," SAID SHAKERIN, Stockton, California, 2004.
- [2] J. J. P. S. J. Hukka, "Precious and treasured water: the role of fountains and water in Islam and Tunis.," in *Water Fountains in the Worldscape*, Tunis, International Water History Association and Kehrä Media Inc., 2012, pp. 85-90.
- [3] S. R. a. S. Y. Qasim, "Mini and Cost-Effective Musical Water Fountain," Sara Raad , Baghdad, Iraq,, September 2019..
- [4] P. P. C. a. G. C. Visconti, "Smart electronic system for dancing fountains control capable to create water and lighting scenarios synchronized with a music track.," Visconti, Paolo Visconti, Italy, May 2016..
- [5] K. P. a. W. Z. H. Soe, "Design and Construction of Musical Lighting Water Fountain.," Published in International Journal of Trend in Scientific Research and Development," Soe, , Mandalay, Myanmar, August 2019.
- [6] D. E. Haldre, "DEVELOPMENT OF EXPERIMENTAL SELF-CHOREOGRAPHED DANCING FOUNTAIN," Juri Šatov,, Tallinn, May 2019.
- [7] M. A.Velasco, "How to make a dancing fountain," Miguel A. Velasco in Water show, Spain, May 2014.
- [8] S. SHAKERIN, "Microcontrolled water fountain: a multidisciplinary project.," SAID SHAKERIN, International Journal of Engineering Education, Britain, November 2004.
- [9] J. M. R. S. A. A.-W. Ali M. Jasim, "Sound Signal Analysis Using FIR Filters for Musical Fountain Operation," *Basrah Journal for Engineering Sciences*, pp. 21-27 , Saturday June 2016.
- [10] R. H. Z. T. S. Retno Devita, "PENGONTROLAN POLA DANCING FOUNTAIN BERIRAMA MUSIC MENGGUNAKAN ANDROID BERBASIS MIKROKONTROLER ARDUINO," *Jurnal Teknologi Informasi dan Pendidikan*, p. P. 104 – 110, Friday March 2020.
- [11] L. D. Vu, "AUTOMATIC MUSICAL FOUNTAIN WITH AUDIO FREQUENCY ANALYSIS USING FFT ALGORITHM," Lung Duc Vu, Chi Minh , August 2011.
- [12] I.-K. L. Min-Joon Yoo, "Creating Musical-Fountain Shows," The IEEE Computer Society, California, September/October 2009.
- [13] D. G. A. B. R. J. P. T. H. Mejia Ugalde Ignacio, "Design of an inexpensive Ethernet/Arduino-based DMX-control system via network applied to musical fountain," International conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), Mexico, 2016.
- [14] B. A. I. ,. T. R. A. T. Israa Subri Alfurat, "Design and Construction a Falling Water Digital Display System," International Journal of Computer Applications, Basrah, Iraq, November 2019.

- [15] "Fountain Pumps Information," Engineering360, [Online]. Available: [https://www.globalspec.com/learnmore/flow\\_control\\_flow\\_transfer/pumps/fountain\\_pumps#:~:text=Fountain%20pumps%20are%20typically%20centrifugal,rather%20than%20out%20of%20it..](https://www.globalspec.com/learnmore/flow_control_flow_transfer/pumps/fountain_pumps#:~:text=Fountain%20pumps%20are%20typically%20centrifugal,rather%20than%20out%20of%20it..)
- [16] "Solenoid Valve," Guangzhou Water Art Fountain Irrigation Garden, [Online]. Available: <http://www.waterartfountain.com/solenoid-valve-products.html>.
- [17] Arduino, "Servo Motor Basics with Arduino," Arduino, 10 5 2022. [Online]. Available: <https://docs.arduino.cc/learn/electronics/servo-motors>.
- [18] B\_E\_N, "What is an Arduino?," sparkfun, [Online]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>.
- [19] "Arduino UNO R3," hobby electronics, [Online]. Available: <https://www.hobbytronics.co.uk/arduino-uno-r3>.
- [20] "Arduino Uno Rev3," arduino, [Online]. Available: <https://store-usa.arduino.cc/products/arduino-uno-rev3>.
- [21] "C Introduction," W3Schools, [Online]. Available: [https://www.w3schools.com/c/c\\_intro.php](https://www.w3schools.com/c/c_intro.php).
- [22] "Use of C Language," simplilearn, 15 Mar 2022. [Online]. Available: <https://www.simplilearn.com/tutorials/c-tutorial/use-of-c-language>.
- [23] "What Is MATLAB?," mathworks, [Online]. Available: <https://www.mathworks.com/discovery/what-is-matlab.html>.
- [24] P. P. C. a. G. C. Visconti, "Smart electronic system for dancing fountains control capable to create water and lighting scenarios synchronized with a music track.," Paolo Visconti, Italy, May 2016.
- [25] K. P. a. W. Z. H. Soe, "Design and Construction of Musical Lighting Water Fountain.," Published in International Journal of Trend in Scientific Research and Development (ijtsrd)., Mandalay, Myanmar, August 2019.
- [26] D. Eero Haldre, "DEVELOPMENT OF EXPERIMENTAL SELF-CHOREOGRAPHED DANCING FOUNTAIN," Juri Šatov, Tallinn, May 2019.
- [27] S. SHAKERIN, "Microcontrolled water fountain: a multidisciplinary project.," SAID SHAKERIN, International Journal of Engineering Education, Britain, November 2004.
- [28] "DC Motors," IQS Directory , [Online]. Available: <https://www.iqsdirectory.com/articles/electric-motor/dc-motors.html>.

## Appendix

```
#include <Servo.h>

#include <FastLED.h>

#define in1 6

#define in2 7

Servo myservo; // create servo object to control a servo

#define LED_PIN 5

#define COLOR_ORDER GRB

#define CHIPSET WS2811

#define NUM_LEDS 30

#define BRIGHTNESS 200

#define FRAMES_PER_SECOND 60

bool gReverseDirection = false;

CRGB leds[NUM_LEDS];

void setup() {

    Serial.begin(115200);

    myservo.attach(9);

    pinMode(in1, OUTPUT);

    pinMode(in2, OUTPUT);

    delay(3000); // sanity delay
```

```
FastLED.addLeds<CHIPSET, LED_PIN, COLOR_ORDER>(leds,  
NUM_LEDS).setCorrection(TypicalLEDStrip);
```

```
FastLED.setBrightness(BRIGHTNESS);
```

```
myservo.write(90);
```

```
delay(15);
```

```
}
```

```
void loop() {
```

```
int sensorValue = analogRead(A0);
```

```
int mappedVal = map(sensorValue, 0, 1023, 0, 255);
```

```
Serial.println(mappedVal);
```

```
myservo.write(170);
```

```
delay(100);
```

```
Fire2012(); // run simulation frame
```

```
FastLED.show(); // display this frame
```

```
FastLED.delay(1000 / FRAMES_PER_SECOND);
```

```
if (mappedVal > 10)
```

```
{
```

```
  pumb_on();
```

```
  myservo.write(90);
```

```
} else
```

```
{
```

```

    pumb_off();

    myservo.write(180);

}

}

#define COOLING 55

#define SPARKING 120

void Fire2012() {

    // Array of temperature readings at each simulation cell

    static uint8_t heat[NUM_LEDS];

    // Step 1. Cool down every cell a little

    for (int i = 0; i < NUM_LEDS; i++) {

        heat[i] = qsub8(heat[i], random8(0, ((COOLING * 10) / NUM_LEDS) + 2));

    }

    // Step 2. Heat from each cell drifts 'up' and diffuses a little

    for (int k = NUM_LEDS - 1; k >= 2; k--) {

        heat[k] = (heat[k - 1] + heat[k - 2] + heat[k - 2]) / 3;

    }

    // Step 3. Randomly ignite new 'sparks' of heat near the bottom

    if (random8() < SPARKING) {

        int y = random8(7);

```

```

    heat[y] = qadd8(heat[y], random8(160, 255));

}

// Step 4. Map from heat cells to LED colors

for (int j = 0; j < NUM_LEDS; j++) {

    CRGB color = HeatColor(heat[j]);

    int pixelnumber;

    if (gReverseDirection) {

        pixelnumber = (NUM_LEDS - 1) - j;

    } else

    {

        pixelnumber = j;

    }

    leds[pixelnumber] = color;

} }

void pumb_on() {

    digitalWrite(in1, LOW);

    digitalWrite(in2, HIGH); }

void pumb_off() {

    digitalWrite(in1, LOW);

    digitalWrite(in2, LOW); }

```