Palestine Polytechnic University

College of Information Technology and Computer Engineering

Department of Computer System Engineering

Smart Pool System

Shaymaa Abuzaneh

Linda Hasasneh

Marwa Aldawdeh

Supervisor: Eng. Elayan Abu-Gharbiyeh

2021-2022

Acknowledgment

First, all the praises are to Allah Almighty and thank him for his blessings throughout our work on our project. He gave us strength, knowledge, and he helped us succeed in this project.

We would like to express our sincere gratitude to our supervisor Eng. Elayan Abu-Gharbiyeh, for guidance, direction, encouragement, and absolute patience throughout the project; presenting their knowledge and experiences to us and urging us to make the project distinct. Also, we would like to extend our greetings to Eng. Wael Takrouri for his dedication, support, and assistance to complete this project.

We would like to thank all the people who helped us, and those who had direct or indirect contributions to our project. Special thanks to our parents who always encourage, support, and care about us. Thanks to Palestine Polytechnic University and its staff for all their teaching and guidance efforts. We are also grateful to our friends in the Computer Systems Engineering program, College of Information Technology and Computer Systems Engineering at Palestine Polytechnic University.

Abstract

Due to the increase in drowning cases, the lack of pure water, and increasing or decreasing the temperature of water in the pool. So, monitoring of swimming pool has become one of the daily routine tasks that may consume time and effort and may be dangerous at times in deep pools. The need for an automated system arose to monitor the swimming pool and the ability to know data remotely through the webpage.

Our project seeks to provide a system that performs several operations, including checking the quality of water. Also, this system checks the temperature of the water in the pool to avoid some problems that may occur, especially when the water freezes. Also, the system measures the temperature of pool water then the system will turn on the heater. This system also deals with drowning cases that may occur. Therefore, this system provides the inhabitant with all the information that it may need from anywhere he is located through a webpage without time or effort and without the need to manually check the pool.

The system has been built and verified to meet the requirements of this project, and it was found that it performs as it should and meets the needs of the project.

Contents

Table of Figures

List of Tables

Chapter 1: Introduction

1.1 Overview

The Smart Pool System is an internet of thing (IoT) based device that is capable of automating the pool processes by Drowning alert (warning) for the pool, knowing the water quality and water temperature.

The project aims to design a system that we will be commanded the Arduino microcontroller through a webpage to control the problem of drowning, water quality, and water temperature.

Where the system finds solutions to the following problems: temperature drop below 18 degrees Celsius, Low water quality (the degree of pollution of the swimming pool water exceeds 700), and the case of drowning in the water as follow:

- If the temperature drops below 18 degrees Celsius, the system turns on the heater which raises the temperature of the pool.
- If the water quality drops so that the water pollution exceeds 700 degrees, the system withdraws all the water from the pump to replace it with clean water.
- To avoid drowning, the person dressed a hat, if the water touches this hat for 15 continuous seconds the buzzer will start working and beep.

1.2 Motivation and importance

One of the most important motives for this project in the current era is that one of the biggest problems the world faces is drowning (especially for children), also low water temperature which leads to dangerous diseases like respiratory diseases and low water quality that affects the health of people who swim in swimming pools where it leads to serious skin diseases especially water that contains chlorine, so a system is required to ensure that the pool is used wisely.

It will also have many advantages for swimming pools. Pool control will become much easier and more convenient, We have chosen this project to make swimming pools easier to care for and to handle most efficiently, to ensure a safe environment for the human body, and to eliminate the difficulties involved in the system by minimizing human intervention as much as possible.

1.3 System objectives

The main objectives of this system are:

- The system will be sent a notification (warning) to the webpage in drowning cases.
- The system will be able to check the quality of water.
- The system will be able to check the water temperature at a certain threshold.

1.4 Problem statement

In this section, we're going to talk about problem analysis, a list of requirements, expected results, and definitions.

1.4.1 Problem analysis

Many problems and dangers are related to swimming pools, people were drowned while swimming especially children, pollution of the pool water and the presence of sediments inside it which makes it unsuitable for swimming and taking into account the ideal water temperature appropriate for each category age.

The main objective of our project is to design a simple, easy to install, microcontroller-based circuit to monitor the pool. It communicates with the various sensor modules in real-time to control the important parameter. Communication will be wirelessly to make the system expandable in the future. The user has only to start the system with one click on a webpage.

1.4.2 List of requirements

The system requirements are summarized as:

1. Control the pool remotely by using the Internet of Things.

2. The system must provide integrated Information about the pool swimming.
3. The software must be able to communicate with microcontrollers and other devices.
4. Build webpage to control swimming pool.
5. The system must be able to expand in the future.

1.4.3 Expected results

Expect to build an integrated system with the following specifications:

1. The system must be able to check the water quality, measure the temperature of the water inside the pool, as well as and be alert in case of drowning.
2. The system must send all data related to the pool to the webpage and display it to the user.
3. System will be built with its final form of simplicity to accomplish our needs.
4. Build interface as a webpage with microcontroller.
5. Build an HTML page.

1.5 Short description of the system

This system works as follows:

This system controls drowning by Wearing a hat before entering the pool, if the hat is exposed to water continuously for 15 seconds, the system will turn on the buzzer to make a sound that there is a drowning person. This system checks the percentage of turbidity in water and if the turbidity percentage is high, the system will draw all the water in the pool using a pump to empty it elsewhere and fill the pool with other clean water. This system also measures the temperature of the water inside the pool as this process is very important in the winter season, when the temperature is low than 18 degrees Celsius heater is turned on which raises the water temperature. The system will use sensors to carry out the previous operations, and these sensors will be connected with the microcontroller, where the data will be downloaded to it and sent via the Internet to a webpage so the user can see all the data.

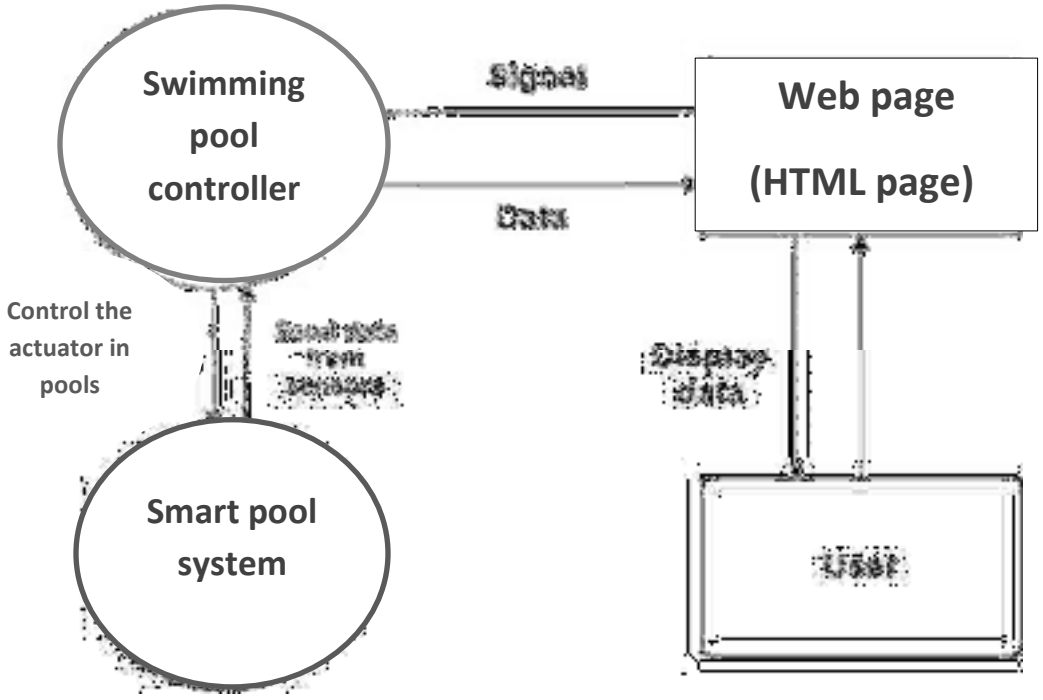


Figure 1: Context diagram of the smart pool system

1.6 Overview of the report

The report is organized as follows: Chapter 2 presents a theoretical background of the project; a description of the hardware and software components is discussed in addition to the system specification and design constraints. Chapter 3 discuss detailed design, block diagrams, flowcharts. Chapter 4 contains an introduction to the software and the platform used for project programming. Chapter 5 contains the expected results and validation for the project. The last chapter contains a conclusion about our work, then, we have a summary, references, and appendices.

Chapter 2: Background

2.1 Overview

This chapter introduces a theoretical background of the system, some descriptions of the hardware and software components used in the system. Specifications of the design system and constraints are discussed too.

2.2 Theoretical background

The project's main idea is to design and implement a smart pool system using an ESP32 mega based on Arduino programming language. Sensors are used to collect data from the water pool then send it to ESP32 that controls the valves of the pool, after that the ESP32 will send it to the web page which will show the data and notifications to the user.

To achieve the goals of our project, we need several components:

- To achieve the water temperature calculation, we need a DS18B20 water temperature sensor, and we need a heater to heat the temperature in case the temperature drops below 18 degrees Celsius by a pump.
- To achieve the water quality calculation, we need a water quality sensor to determine the dissolved chemicals in the water (TDS), and we need a pump to get rid of the water in case the water pollution rate exceeds 700 and replace it with a clean one.
- We need the water drowning sensor, in this part, we need a hat that contains the Arduino-nano board to secure the information between the transmitter, and receiver, transmitter that sends the information from the water (TX), and a socket (acts as a switch and contain 3 pins) where when the pins touch water, the information is sent by TX to a receiver (RX)which is located on the ESP32 mainboard and received to start counting up to 15 seconds and after the end of 15 seconds it starts to beep to indicate a drowning.

To check whether this water is safe to swim in or not we need to measure the quality of water. Quality is the degree or level of cloudiness or haziness of a liquid. This happens due to the presence of large numbers of invisible particles with the naked eye similar to white smoke in the air. When light passes through liquids, light waves get scattered Due to the presence of these tiny particles. The quality of a liquid is directly proportional to the free suspended particles, that is if the number of particles increases, quality will also increase. So, if the quality is less than 700, we conclude that the water is pure, and if it is greater than 700, we conclude that it is not pure.

To find a solution to the problem of water quality, we have installed a pump that is operated automatically through a relay if the water quality rate exceeds 700, where the polluted water is initially drawn to an empty place, then clean water is pumped in its place.

To check the water temperature, we used a temperature sensor and adopted a temperature of 18 degrees Celsius to assess the rise and fall of the temperature so that if the temperature drops below 18 degrees, the system automatically operates a heater by a relay that raises the water temperature to be suitable for swimming.

For the drowning problem, we have relied on making the drowning sensor, which is a socket that acts as a switch and contains three pins, which connected with: transceiver that sends and receives data between the sensor and the esp32 board in the form of digital signals, and Arduino -nano board to secure the transmission and reception in the form of digital signals, Where if the pins come into contact with water, the information will be sent by the transmitter via the internet to the receiver on the esp32 board as digital signals, as if the value of TX is 0, the RX signal remains 0, but after the contact of the water with the pins exceeds 15 seconds, it sends TX Signal 1 to RX In this case, RX automatically triggers the buzzer to alert the pool owner that someone is drowning. The drowning sensor (socket) with transmitter TX and Arduino-nano will be placed inside a hat that the person wears before entering the pool as shown in Figure 2.

((11) Sender and receiver:, 2021)

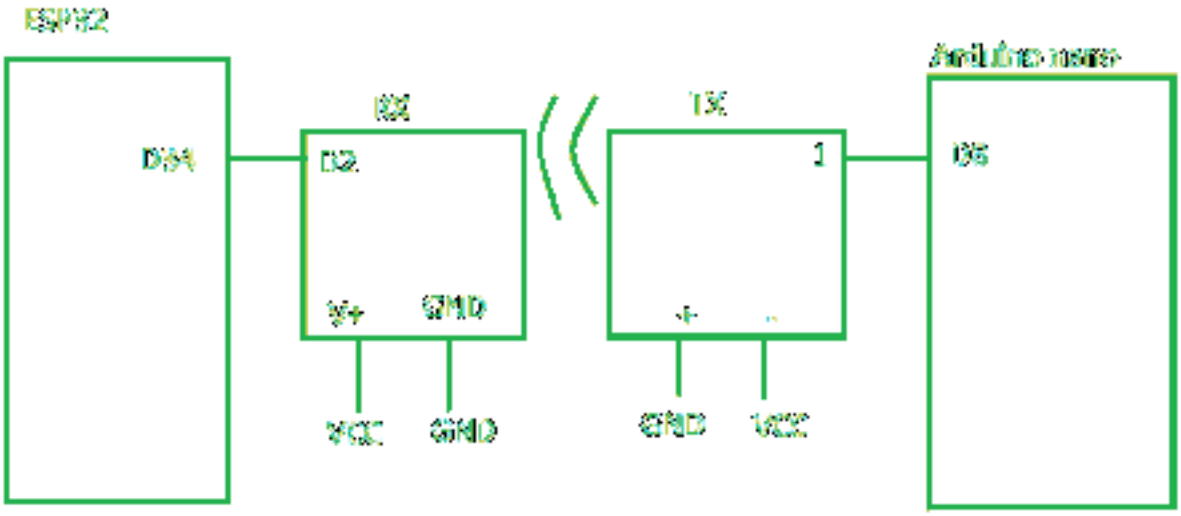


Figure 2: Connection of sensor drowning water

2.3 Hardware component of the system:

This part describes the components and devices used in the system:

2.3.1 Node MCU ESP32

The ESP32 is a dual-core 160MHz to 240MHz CPU, whereas the ESP8266 is a single-core processor that runs at 80MHz. These modules come with GPIOs that support various protocols like SPI, I2C, UART, ADC, DAC, and PWM. The best part is that these boards come with wireless networking included, which makes them apart from other microcontrollers like the Arduino. This means that you can easily control and monitor devices remotely via Wi-Fi or Bluetooth for a very low price.

We will program this controller to deal with the data coming from the connected sensors and send it to the application for display to the user, and use it to connect the system with the Internet.

2.3.2 Water Temperature Sensor

The DS18B20 is a temperature sensor, manufactured by Dallas Semiconductor Corp. Wire encounter requires only one digital pin for bidirectional communication with the microcontroller.

The DS18B20 temperature sensor usually comes in two forms. One that comes in the TO-92 package is just like a regular transistor. The other is in waterproof probe style which can be more useful when you need to measure something far away, underwater or underground.

We will use this sensor to measure the temperature of the water in the pool system.

Figure 3: Water Temperature Sensor

2.3.3 TDS (Total Dissolved Solids) water quality measuring sensor

It detects total levels of dissolved solids (TDS) in the water which can be used to indicate the quality of the water. The TDS water quality sensor can be applied in water quality applications and it Supports 3.3 / 5V input voltage and 0 ~ 2.3V Output Voltage making it easy to be compatible with all Arduino boards. The sensor also provides a waterproof probe, which makes the testing process much easier.

We will use this sensor to measure the quality of the water in the pool system, calculate the total dissolved solids in the water.

Figure 4: Water quality sensor

2.3.4 Transistor as a switch

It used as sensor drowning system that contains 3 pins when Touching water for 15 seconds continuously sends an alarm to a receiver that turns on the buzzer.

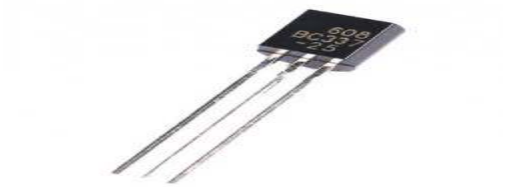


Figure 5: Transistor as a switch

2.3.5 RF transmitter and receiver

The RF transmitter receives serial data and transmits it wirelessly through its RF antenna. The transmission occurs at the rate of 1 Kbps – 10 Kbps. RF receiver receives the transmitted data and it is operating at the same frequency as that of the transmitter.

They will be used to send information from the drowning sensor to be received by the RX on the ESP32 board to turn on the buzzer in the event of a drowning. ((11) Sender and receiver:, 2021)

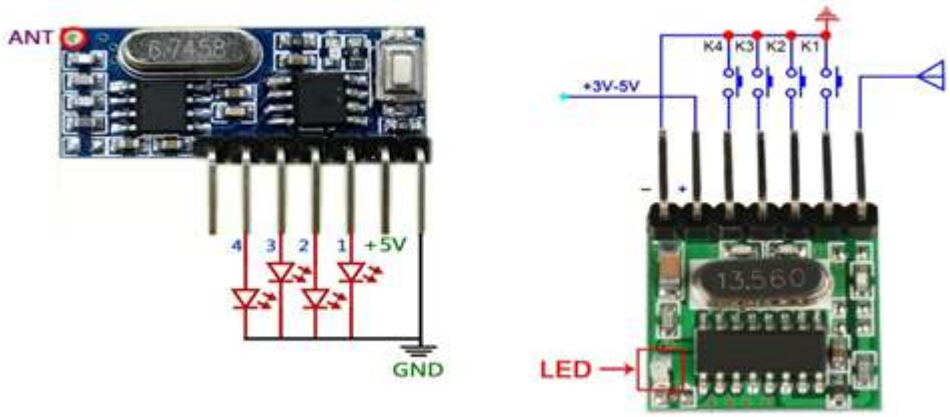


Figure 6: RF transmitter and receiver

2.3.6 Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3. x). It has more or less the same functionality as the Arduino Duemilanove but in a different package. It lacks only a DC power jack and works with a Mini-B USB cable instead of a standard one. ((10)Water sensor with Arduino nano:, 2021)

It will be used to secure the transmission between the transmitter and receiver in the drowning sensor.

Figure 7: Arduino Nano

2.3.7 Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

It will be used to beep if water touches the sinking sensor for more than 15 seconds.

Figure 8: Buzzer

2.3.8 Heater (TEC1-12706 Thermoelectric)

Thermoelectric Peltier cooler TEC1-12706, consists mainly of semiconductor material sandwiched between two ceramic plates is are used in many applications from CPU coolers to other power sources. A thermoelectric cooling module (TEC) is an electronic semiconductor component that functions as a small heat pump. By applying a DC power source to a TEC, heat

will be transferred from one side of the module to the other. This creates a cold and warm side. They are widely used in industrial areas, e.g. computer CPUs, CCDs, portable refrigerators, medical instruments, etc.

It will be used to heat the water if the pool temperature drops below 18°C.

Figure 9: Heater (TEC1-12706 Thermoelectric)

2.3.9 Pump

A pump is a device that moves fluids (liquids or gases), or sometimes slurries, by mechanical action, typically converted from electrical energy into hydraulic energy. Pumps can be classified into three major groups according to the method they use to move the fluid: direct lift, displacement, and gravity pumps.

In our project, a pump will be used to pump clean water into the pool after emptying it of the polluted water.

Figure 10: Pump

2.3.10 Relay

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

In our project, two pumps will be used, one to operate the heater and the second to operate the pump.

Figure 11: Relay

2.4 Software component of the system

2.4.1 Arduino IDE

The Arduino Integrated Development Environment or Arduino Software (IDE) li is open-source, 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. It connects to the Arduino and Genuine hardware to upload programs and communicate with them. It runs on Windows, Mac OS X, and Linux, written in the Java programming language.

2.4.2 HTTP Protocol

2.4.3 XAMPP control panel

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and use.

Where this program will be used to run the code and control the database. (Websites references, 2021)

2.5 Specification and design constraints.

2.5.1 Design Specification

Our project should be built in a simple form and must be ready to receive data from Arduino to start controlling the swimming pool.

2.5.2 Constraints

- Permanent need for the internet
- This system works for one pool
- Our project should be comfortable and accurate with specific measurements in order to send data without mistakes at the same time

- Arduino Uno must be used because it doesn't require-shield like Arduino mega

2.5.3 Design alternatives.

This subsection illustrates the alternative for NodeMCU ESP32 and explains why it is not used.

2.5.3.1 Raspberry pi

Raspberry Pi is a single-board computer. It is a credit-card-sized computer with low cost, which plugs into a computer monitor or TV and to operate it, a user can use a standard keyboard and mouse to operate it. The single-board consists of a fully functional computer with its dedicated memory, processor and it runs an operating system.

The table below shows the main difference in specifications for ESP32 and Raspberry Pi

Table 1: RASBERRY PI VS ESP32

FEATURE	ESP 32	RASPBERRY PI
Core count	Single/dual-core	Dual-core
Microcontroller	TensilicaXtensa LX6	RP2040
Clock frequency	160kHz / 240kHz	48kHz / 133kHz
Internal Flash Memory	4 MB	2 MB
SPI	4	2
I²C	2	2
PWM	16	16
ADC	18(12-bits)	3(12-bits)
GPIO(total)	34	26
UART	3	2
Programming Language	C, MicroPython with limited support	Micro Python, C, C++
Wi-Fi	802.11 b/g/n	Not Supported
Bluetooth	V4.2 (Supports Classic Bluetooth and BLE)	Not Supported
Ethernet	10/100 Mbps	Not Supported
Price	$4 – $12	$4

2.5.3.2 ESP 8266

All ESP8266 variants have an ESP8266EX core processor and a Tensilica L106 32- bit microcontroller unit. This is a low cost, high performance, low power consumption, easy to program, wireless SoC(System-On-Chip).

This table shows the main difference in specifications for ESP8266 and ESP32.

Table 2: ESP8266 VS ESP32

FEATURE	ESP32	ESP8266
MCU	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS	Xtensa Single-core 32-bit L106
802.11 b/g/n Wi-Fi	HT40	HT20
Bluetooth	Bluetooth 4.2 and BLE	X
Typical Frequency	160 MHz	80 MHz
SRAM	✓	X
Flash	✓	X
GPIO	34	17
Hardware /Software PWM	None / 16 channels	None / 8 channels
SPI/I2C/I2S/UART	4/2/2/2	2/1/2/2
ADC	12-bit	10-bit
CAN	✓	X
Ethernet MAC Interface	✓	X
Touch Sensor	✓	X
Temperature Sensor	✓(old versions)	X
Hall effect sensor	✓	X
Working Temperature	-40ºC to 125ºC	-40ºC to 125ºC
Price	$$ ($6 - $12)	$ (3$ - $6)

2.5.3.3 Arduino mega

Arduino Mega 2560 is a microcontroller board based on the ATmega2560, it is designed for projects that require more I/O lines, more sketch memory, and more RAM. Connect the board to the computer using the USB cable to load the program and power it up.

This table shows the main difference in specifications for Arduino Mega 2560 and ESP32 Node MCU.

Table 3: ESBNODEMCU VS Arduino mega

FEATURE	Arduino mega	ESP32 NodeMCU
Price	36$ - 39$	$11
Processor	Atmega 2560	ESP32
Clock speed	16MHz	80 MHz / 160 MHz
Digital I/O Pins	54	36
Digital I/O Pins with PWM	15	36
Analog Input Pins	16	15
SPI/I2C/I2S/UART	1/1/1/4	4/2/2/2
WIFI	No	Yes
Ethernet MAC Interface	No	Yes
Bluetooth	No	Yes

2.6 Additional information

We can manage the entire pool with the help of a webpage. enjoy the liberty that new technology has to offer ensuring the fact that total control of the pool.

The entire pool can be managed with an Arduino Uno and Arduino Software (Arduino IDE), so that the warning from drowning, water quality, and temperature using appropriate sensors.

Chapter 3: System Design

3.1 Overview

The following section has a description of the system, detailed design, and necessary information about the design.

3.2 Detailed description of the system

3.2.1 Hardware Design

The system consists of the ESP32, TDS water quality measuring sensor, temperature sensor, and transistor as a switch to drowning case.

Our system will be designed to track the smart pool and detect any abnormalities. When the system is activated, The system receives water signals touching the drowning sensor, and then it will send it to the ESP32 to prepare to run the buzzer automatically after 15 seconds in case there is a drowning person, Also, the temperature sensor will measure the temperature of the water inside the pool and if temperature down than 18 0C the ESP receives signals to turn on a pump that turns on the heater to raise the water temperature.

There is also a water purity sensor that will check the water turbidity, as if the water turbidity is poor, the ESP32 will send a signal to the relay which turn on a pump that allows water to enter and exit the pool system, informing the user of the status.

3.2.1.1 IIOT device options

- First Design Option:

  Use ESP32 to build cloud page and met information then interpretation.

- Second Design Option:

Use Bolt IoT platform it is a hardware and software platform to build cloud page .

- Chosen Design Option:

We have chosen the first approach because it can combine more than one work at the same time like the Microcontroller option replace Arduino board and IOT board, also the usage of the device is very simple robust communication is equipped with industry-standard protocols to ensure secure and fast communication of the device data with the cloud, more security, ESP32 has built-in safeguards to secure all user data from unwanted third party instruction and hacks, can easily connect it to any pins, but using Bolt IoT needs an extra shield to connect with Arduino UNO, there will be more cost and more complexity.

3.2.1.2 Sensors options

• Chosen Design Option:

The Type of chosen sensor is taken into account by a simple method of measurement, delivers the results immediately, and is very low in cost.

3.2.1.3 Relay options

- First Design Option:

  Use solid-state relay (SSR), a relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate as a switch.

- Second Design Option:

  ULN2003A is used wherever it is necessary to control high power or high voltage circuits with a low power circuit and relay can work either as switches or as an amplifier.

- Chosen Design Option:

  We have chosen the second approach, because; they are simple in construction, they are reliable, the values can be easily set. No special programming device is required.

3.2.1.4 Water Pumps option

- First Design Option:

Use AC water pump, it is used to control the pipe of water, and pump water to the right place.

- Second Design Option:

  Use a 12-volt pump, it is used to control the pipe of water, and pump water to the right place.

- Chosen Design Option:

  We have chosen the second approach, because; safety: The main reason for this is the fact that it normally uses a low voltage power supply. This means that electrical shocks are something that you should not worry about, easy to control, cost-effective.

3.2.2 Software Design

We use one programming software for programming the Arduino as described below: Arduino Software: The open-source Arduino Software (IDE) makes it easy to write codes and upload them to the board. It runs on Windows. The environment is written in java and based on processing and other open-source software. It will be used to write codes for driving the component of the kit to be able to function.

The reason for choosing the Arduino IDE program; it is suitable for Arduino devices and easy to use.

3.3 System diagrams

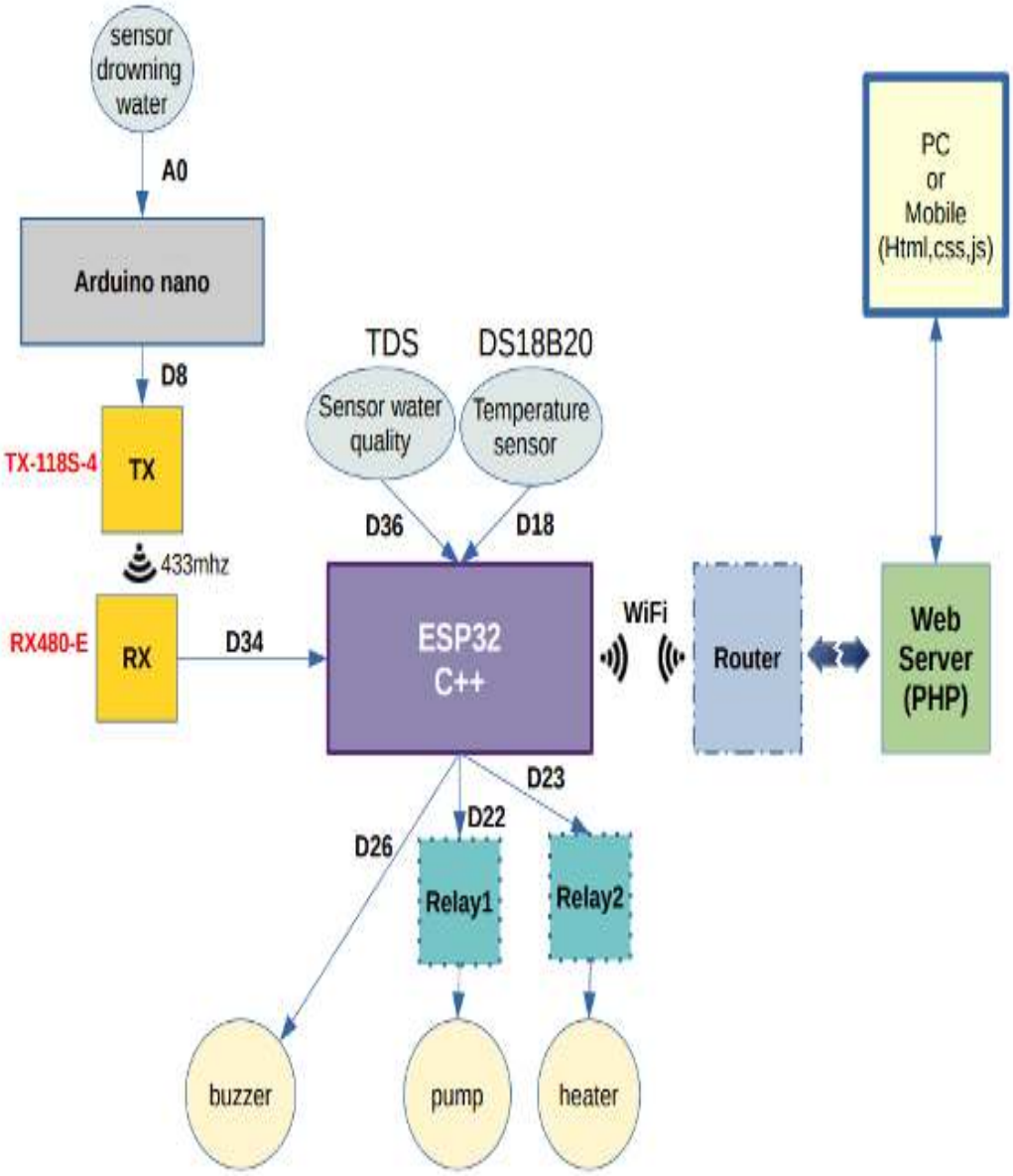
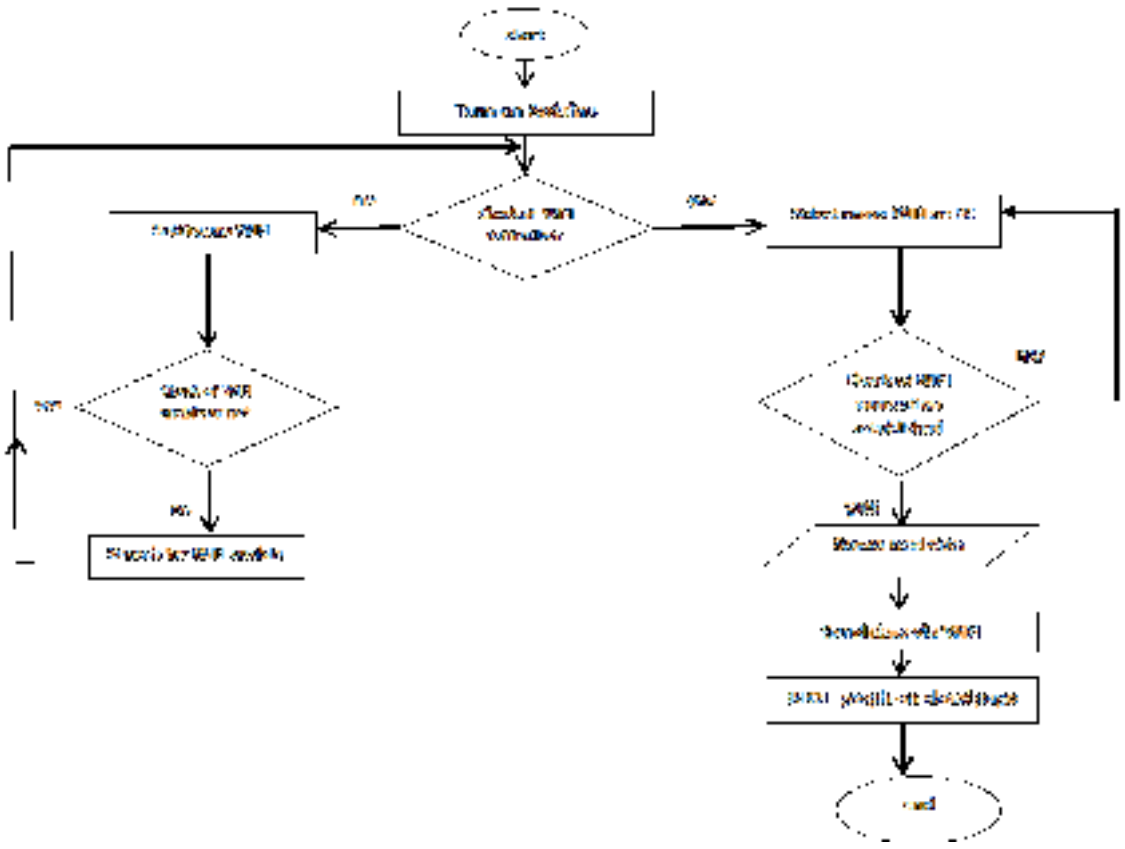


Figure 12: Block diagram

The Block diagram in Figure 13 shows how reading sensors will be transformed to ESP32 in form of data via the internet The ESP32 should receive data and send signals to the cloud page to analyst data and control the relay, Buzzer remotely.

3.4 Flow charts



flowchart system13 Figure

We built the software that controls our system as shown in the following flowchart (Figure14)

1. Turn power the Arduino.

2. Check the WIFI connection: if WIFI is enabled connect with the name available to start transfer data. If not enable set up WIFI, and search for WIFI enable.

3. Start reading data from the sensor that connects to the Arduino.

4. Send data via WIFI to the cloud page.

5. data graph in cloud page and analyze data.

6. Control motor by webpage.

3.5 Pseudo code of the system

The following subsections have a description of the pseudo code used in the system:

connect to internet

connect to database

while connecting to the internet and database

get data from the database

get data from sensors

calculate the amount quality of water

if quality data greater than or equal 700

Sending a signal to pump to be open

if temperature data is less than 18

Sending a signal to the heater to turn on

if the sensor drowning water touch the water for 15 seconds straight

Sending a signal to the buzzer to turn on

Then sending signals to the pump to open

if data completed

Read data

else

update this system to read after connecting with the internet

3.6 Schematic diagram

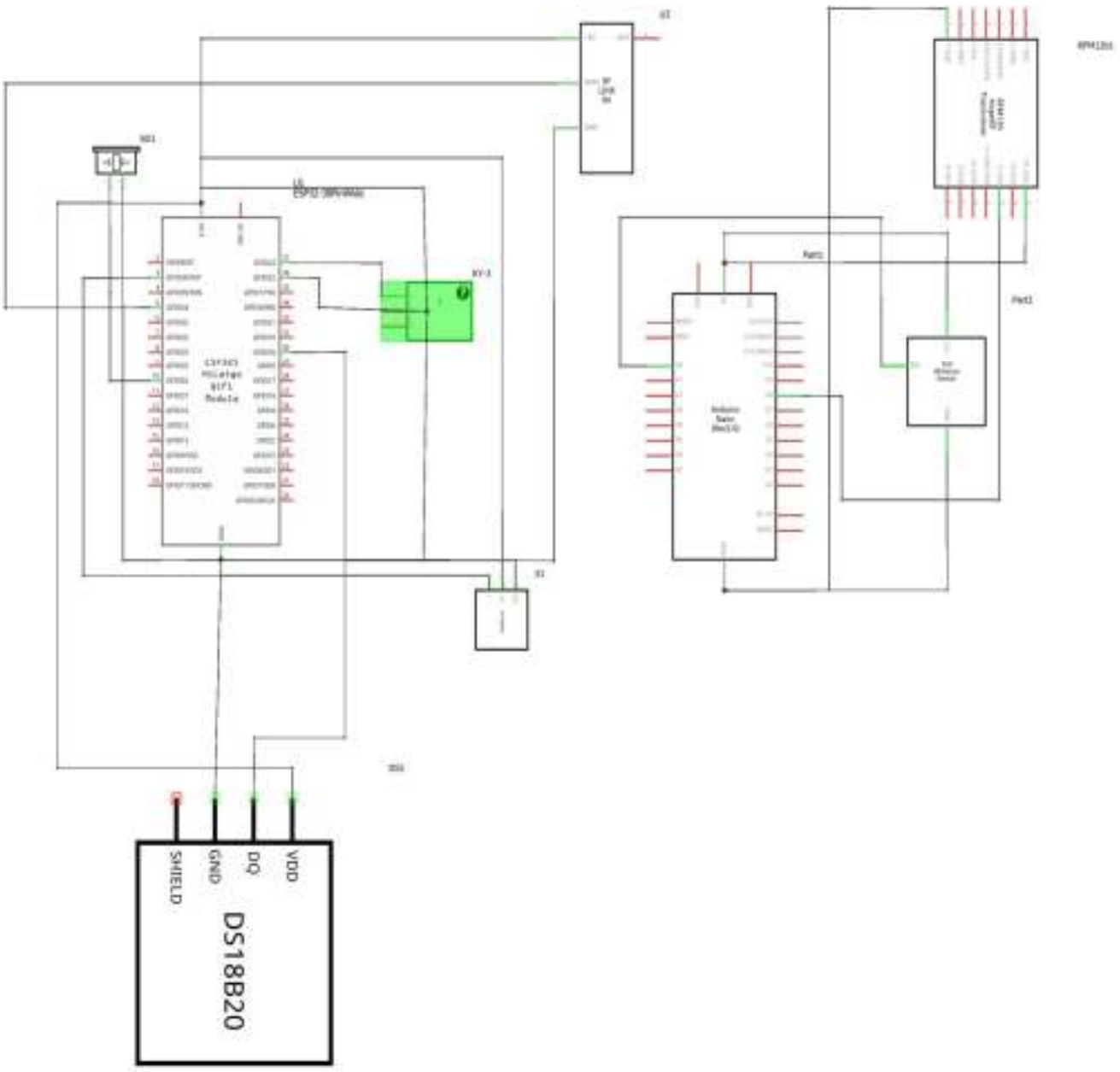


Figure 15: Schematic diagram of smart pool system

1. ESP32: The main component in the system, which will be linked to the other components. Represent the inlet and outlet of the signals and information. It will also control the valves connected to it based on the data it receives from the senators connected to it, and it will also send the data to the cloud to display on the web application.

2. Purity sensor (water quality sensor): It is used to find the water turbidity rate, as if the percentage is high, it will send a signal to the ESP32 to turn on the pump that pumps clean water.
3. Temperature sensor: It is used to measure the temperature of the water, as if it is down than 18ºc, it will send a signal to the ESP32 to turn on the heater. ((6)Temperature sensor:, 2021)
4. Drowning water sensor: It is used to sense water that touches the sensor, as if the water touch sensor over 15 seconds, it will send a signal to the ESP32 to turn on the buzzer.

3.7 Detailed design

In this section we will talk about Hardware and Software be used in our project that summarized below:

3.7.1 Hardware

1. NodeMCU we will choose ESP32, specifications for this:

   ESP32 is a low-cost, low-power system on a chip (SoC) series with Wi-Fi & dual-mode Bluetooth capabilities. At its heart, there's a dual-core or single-core TensilicaXtensa LX6 microprocessor with a clock rate of up to 240 MHz ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics, and IoT applications, ESP32 achieves ultra-low power consumption through power-saving features including fine resolution clock gating, multiple power modes, and dynamic power scaling.

3.7.2 Software

- Arduino

  The Arduino Integrated Development Environment - or Arduino Software (IDE) - makes it easy to write code and upload it to the board. This software can be used with ESP32 boards. And this 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. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. And the IDE.

: Arduino IDE logo16 Figure

Chapter 4: System implementation, testing, and discussion

4.1 Overview

This chapter introduces a description of the implementation, implementation issues, implementation challenges, description of the method used to validate the system, validation results include an analysis and discussion about the result and recommendations based on the result.

4.2 Description of the implementation

4.2.1 Software Implementation tools

We used Arduino IDE in our project, and for a Programming language, The Arduino has its own IDE where you write the program. C code is just used as a partial interface between humans and the Microcontroller to instruct the Microcontroller so as to perform a particular operation. After the code is written, it is compiled. Compiling is basically converting the code written in English words into a hex file which is a series of l's and 0's. The Microcontroller only understands binary language and hence the compilation process is required. So, the code is implemented using C and is executed by a hex code. HTML code is just used to a built webpage, and JavaScript to display data in graphical form.

4.2.2 Hardware Implementation

4.2.2.1 Communication

To connect Arduino UNO with the cloud a Wi-Fi (ESP32) connection is needed before we start printing data of sensors.

4.3 Implementation issues and challenges

1. Finding the appropriate sensors.

2. Sending a signal to the cloud pool and analyzing this signal.

3. Experience the best way to connect the devices via WIFI.

4. Errors during connection and learning how to connect the ESP32 to the internet.

4.4 Validation result

4.4.1 Hardware testing

A NodeMCU ESP32 was tested with all sensors and connected directly to the laptop as well as the 12V power supply all of them pass. ((5)Hardware references and programming: Internet connection, 2021)

This image represents the project that was built in the university lab and that we did the tests on.

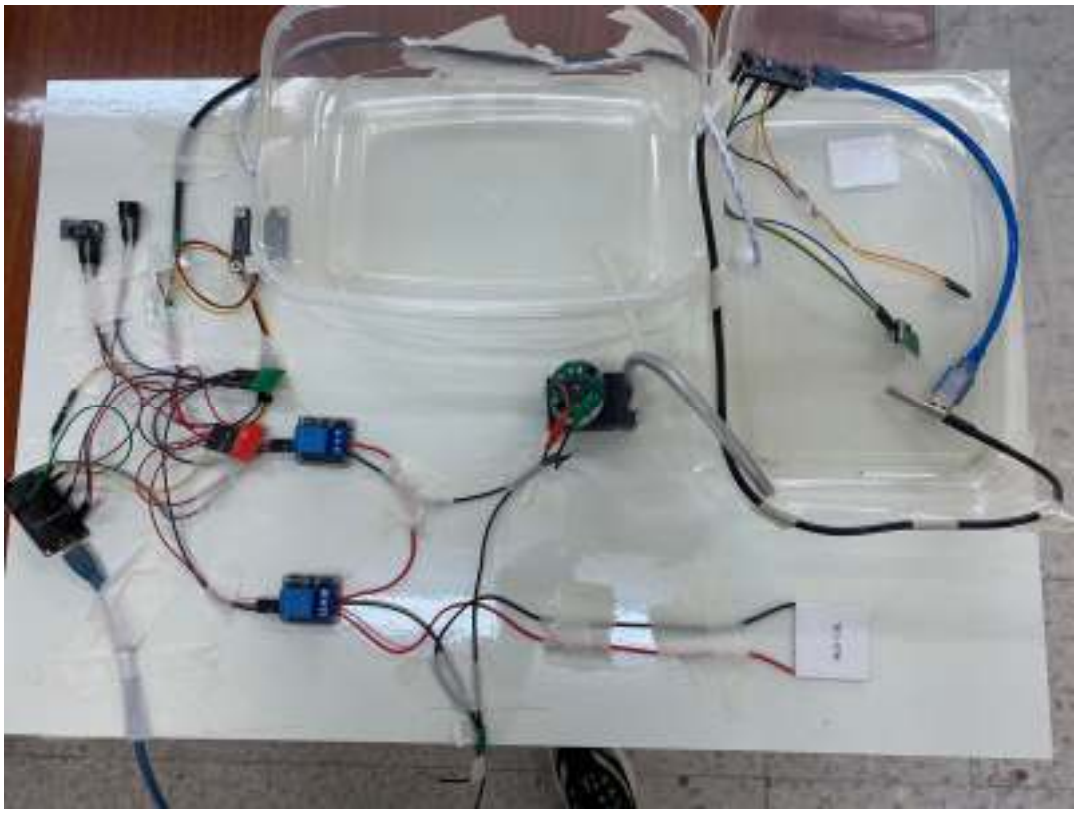


Figure 17 Hardware Components

This picture represents the assembly of wires on the breadboard and the controller.

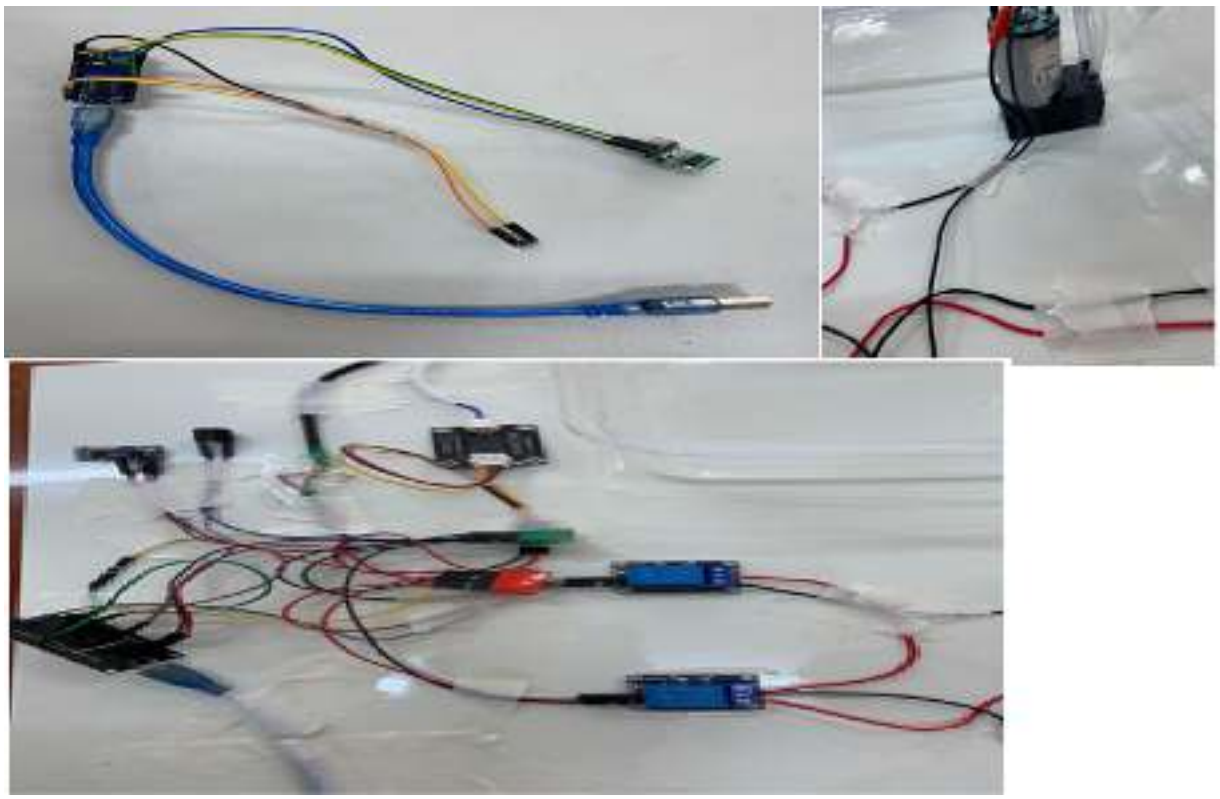


Figure 18 Assebly of wires

The data is stored in the Database in the fields shown in the following figures on the Firebase Real-time Database website: ((2) Webpage, 2021)

((1) Webpage, 2021)

Figure 19: All data Query

4.4.2 Software testing

The system was fully checked and ensured how it worked and the results of the Testing were successful, and the following tables are a review of some of the tests that we have carried out.

Table 4 Software Testing

Pass/Fail	Obtained Output	Expected Output	Case	number
Pass	Connect to the internet successfully	Connect to the internet	Connect to the internet	1
Pass	Get data successfully	Get data from DB	Get necessary data from the database	2
Pass	Get data successfully	Get data from all sensors	Get data from sensors	3
Pass	Read and write successfully	Read and write data from TX	Get and store data from TX	4
Pass	Send data successfully	Send data to DB	Send data to DB	5

Table 5 Sensors Testing

Pass/Fail	Obtained Output	Expected Output	Case	Number
Pass	Open pump successfully	Open pump if the water is not pure	Open pump if the water is not pure	1
Pass	Turn on heater successfully	Turn on heater if the temperature is low	Turn on heater if the temperature is low	2
Pass	Turn on pump successfully	Turn on the pump if the Buzzer is on	Turn on the pump if the Buzzer is on	3
Pass	Turn on buzzer successfully	Turn on the buzzer when drowning case	Turn on the buzzer when drowning case	4

Table 6 Webpage Testing

Pass/Fail	Obtained Output	Expected Output	Case	Number
Pass	show data successfully	show data in a user interface	Webpage returns data	1
Pass	show data successfully	Show current pool status	Show current pool status	2
Pass	show data successfully	show text in a user interface	Show if there is a problem	3

- The user is asked to connect to the "graduate" network. After the connection, he can enter the webpage, where he is asked to enter the user name and password. ((1) Webpage, 2021)

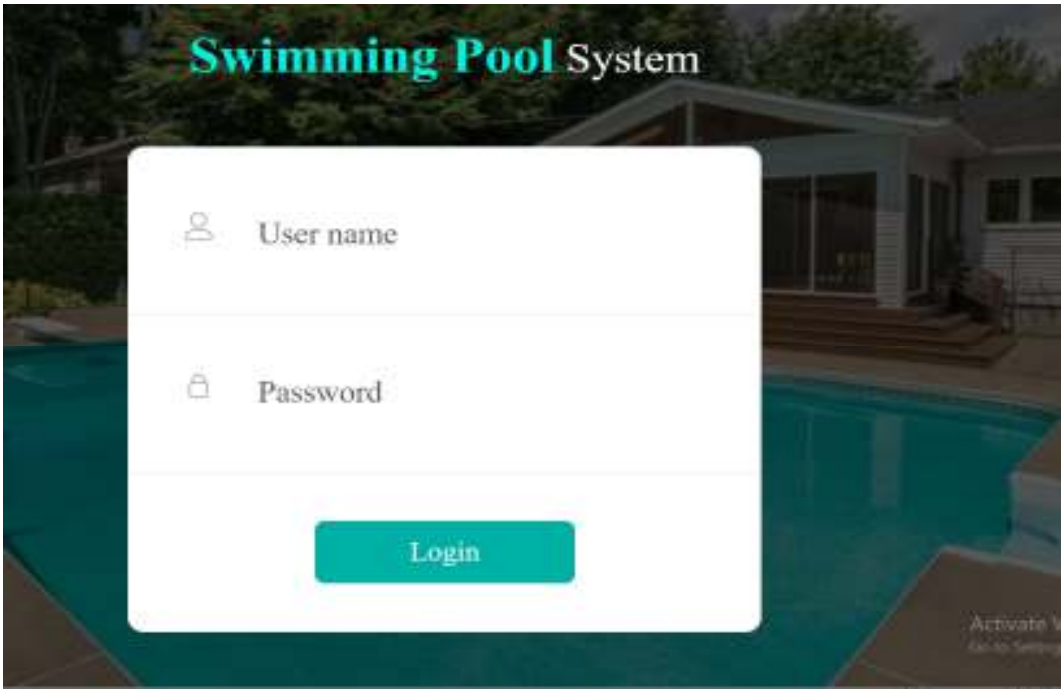


Figure 20: Webpage interface

- When the user opens the webpage, then the user goes to the home page, which contains data on the pump, heater, and buzzer witch all can turn on or off by clicking him.

Figure 21 Webpage Switches

- Then shows the value of temperature and water quality. Users can see the value of water quality with its interpretation of whether it is greater than 700 Tds and value of temperature witch less than 18^0c.
  - Finally, if the user wants to log out from the system he clicks sign-out.

Figure 22 Webpage Interface

4.5 Discussion

In the beginning, there were several goals in this project to solve many problems. Work has been done to achieve this through this system, which consists of many sensors and pumps. Where these components have been programmed to achieve the requirements of this system, and after building this system and doing the necessary tests to ensure that it works in the required manner and whether it achieves the goals of its construction or not, we can say that it achieved the desired goals and that it worked as required for a simple system. As this system took the data from the sensors correctly and the controller processed it and performed the calculations on the obtained data and controlled the pumps based on the obtained values. The process of sending and receiving data from the database was done correctly, and it was displayed and controlled as required from the web page.

Chapter 5: Conclusion and Future work

5.1 Overview

Tack in this part about the summary of all parts, future directions, and future work.

5.2 Conclusion

A smart pool system can solve many of the problems facing people, such as checking the quality of water, which if it is not pure, affects the health of individuals, and solving the problems that may occur in the smart pool in terms of low water temperatures, drowning case, which saves a lot of time and effort and eliminates the danger that may occur. In this system, users can view all their data through the web page, and they can control the pump on the pool by making it on or off. The control also deals with some things that may lead to low temperature, water quality, which may sometimes occur without the knowledge of the user, which reduces time, effort, and cost.

This system has been built and verified to work as required to achieve the requirements of this project, and it was found that it works as required and meets the needs of this project.

5.3 Future work

In the future, we look forward to adding important features to the system, the most important of which are:

- Develop the system to deal with more than one pool.

- Developing and enlarging the pump system inside the pool so that the pool can be emptied during the drowning process and polluted water inside the pool can be purified faster.

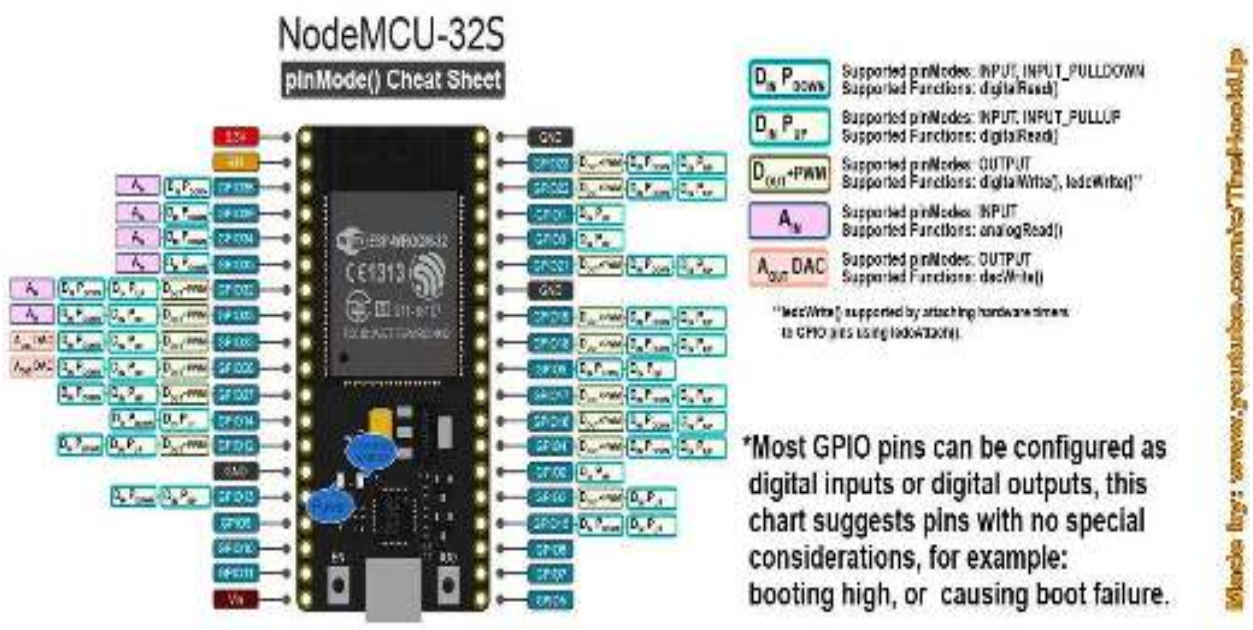
- Connecting the system with the authorities responsible for managing the swimming pools so that it can be controlled better and faster.

Appendices

Appendix A

Datasheets for needed components

((8)esp32 datasheet:, 2021):



NodeMCU32S Datasheets23 Figure

References

Websites references

(1) ((4) Webpage, 2021)

(2) https://www.w3schools.com/js/js_json_php.asp?fbclid=IwAR3mOw8W1PE2Uk-9SWKsy88nrHVGhBjW5FVFP6Lv_s1tLSAzb9vo0qNiL4g

(3) ((3) Webpage, 2021)

(4) https://www.w3schools.com/php/php_mysql_connect.asp?fbclid=IwAR1zXPqaOvjFiKy5NUTLX7bMPxNyPIVcka--XVOpDQ-CWQK0w5W0j8BupCg

(5) ((2) Webpage, 2021)

(6) https://www.w3schools.com/php/?fbclid=IwAR10xYg3SdOvSwsoNdxQWHy5Ya7e9fYm7xPOI0NSIXrdh0RspgJDaeSDUr0

(7) ((1) Webpage, 2021)

(8) https://www.w3schools.com/css/?fbclid=IwAR3pTdX_dhDquvFNZpQEk9tX5Px75CitBPe7TYvkF8vlUkiWUvSQAXNSqqA

((5)Hardware references and programming: Internet connection, 2021) https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/?fbclid=IwAR1edMBskIAbcRYdA1A1Sb5YXtfvmRUYUePOaVI_DVlQEMGtXkbxx4IvtQY

((6)Temperature sensor:, 2021) https://www.electronicshub.org/esp32-ds18b20-tutorial/?fbclid=IwAR2QjysuPXqTLWYB0gvB5rIGvWTgRZZQM2ud5WbhiuUpmR_3I3nPcX-5HEs

((7)Analyze the data retrieved from the site:, 2021)

https://randomnerdtutorials.com/decoding-and-encoding-json-with-arduino-or-esp8266/?fbclid=IwAR2QjysuPXqTLWYB0gvB5rIGvWTgRZZQM2ud5WbhiuUpmR_3I3nPcX-5HEs

((8)esp32 datasheet:, 2021)

https://www.studiopieters.nl/esp32-pinout/?fbclid=IwAR3Us7bAw4MJnCz3maq3YQHFf_jqdD8YVddLfT9NfgSJbRM3EcIHVWvgzEY

((9)Temperature and water quality sensor:, 2021)

https://how2electronics.com/iot-based-tds-meter-using-esp8266-for-water-quality-monitoring/?fbclid=IwAR10J5FLSD4TnWRw-EikwRlj0J95rbD1oWV7uY1vJlk6qlD5VRV8QCwBFBU

((10)Water sensor with Arduino nano:, 2021)

https://create.arduino.cc/projecthub/muchika/how-to-use-soil-moisture-sensore-using-arduino-af1365?fbclid=IwAR3maX_Z9rwUmCa943OLzYhDynJ_Dt3PGMMXIWglRsGswdjUsoGKDUpqEK4

((11) Sender and receiver:, 2021)

https://acoptex.com/project/11234/basics-project-086f-esp8266-esp-12e-module-rf-315330433-mhz-xd-rf-5v-receiver-led-control-at-acoptexcom/