IOT BASED SMART WATER FOUNTAINS…

Date: 16.10.2023

PHASE 3: Development Part

To build the IOT smart water fountain systems.

INTRODUCTION:

A water fountain or animated fountain is widely used these days to decorate city parks and squares. It can either spray water into the air or create a waterfall effect.

Water fountains mainly need three parts to operate: water source, water pump and delivery channel . All outdoor water fountains have a water source or reservoir from which they draw their water and to which the water returns, creating a closed circuit. The power to move the water is supplied by a pump. The pump is usually submerged in the water reservoir. A spinning impellor in the pump draws water in and is spinning at such a speed that the water is forced out of the pump by centrifugal force. The water forced from the pump is delivered to the delivery channel or fountain head where it is sprayed into the air through a fine nozzle or where it is allowed to flow down the outside of the structure of the fountain, The water that is sprayed into the air will fall back into the fountain and drain back into the reservoir.

COMPONENTS:

1. Arduino UNO
2. Transistor as switch
3. Water pump
4. 12v power supply
5. Jumper wires
6. Breadboard
7. Solenoid valve

Physical Design:

A pictorial representation of your project that puts your solution in context. Not necessarily restricted to your design. Include other external systems relevant to your project (e.g. if your solution connects to a phone via Bluetooth, draw a dotted line between your device and the phone). Note that this is not a block diagram and should explain how the solution is used, not a breakdown of inner components.

His excellent build is outlined here, including a surprisingly simple circuit that controls the pump and solenoid valve illumination via a trio of MOSFETs.

Requirements:

- Wokwi
- ThingSpeak
- ESP32
- Relay
- GND

- VCC
- Resistor
- Ultrasonic sensor
- LED

Wokwi:

Wokwi is an embedded systems and IoT simulator supporting ESP32, Arduino, and the Raspberry Pi Pico. Your code never leaves Your computer – Wokwi runs the simulation inside VS Code, using The firmware binaries from your project.

ThingSpeak:

ThingSpeak is an IoT analytics platform service that allows You to aggregate, visualize, and analyze live data streams in the Cloud. You can send data to ThingSpeak from your devices, create Instant visualizations of live data, and send alerts using web Services like Twitter and Twilio .

Procedure:

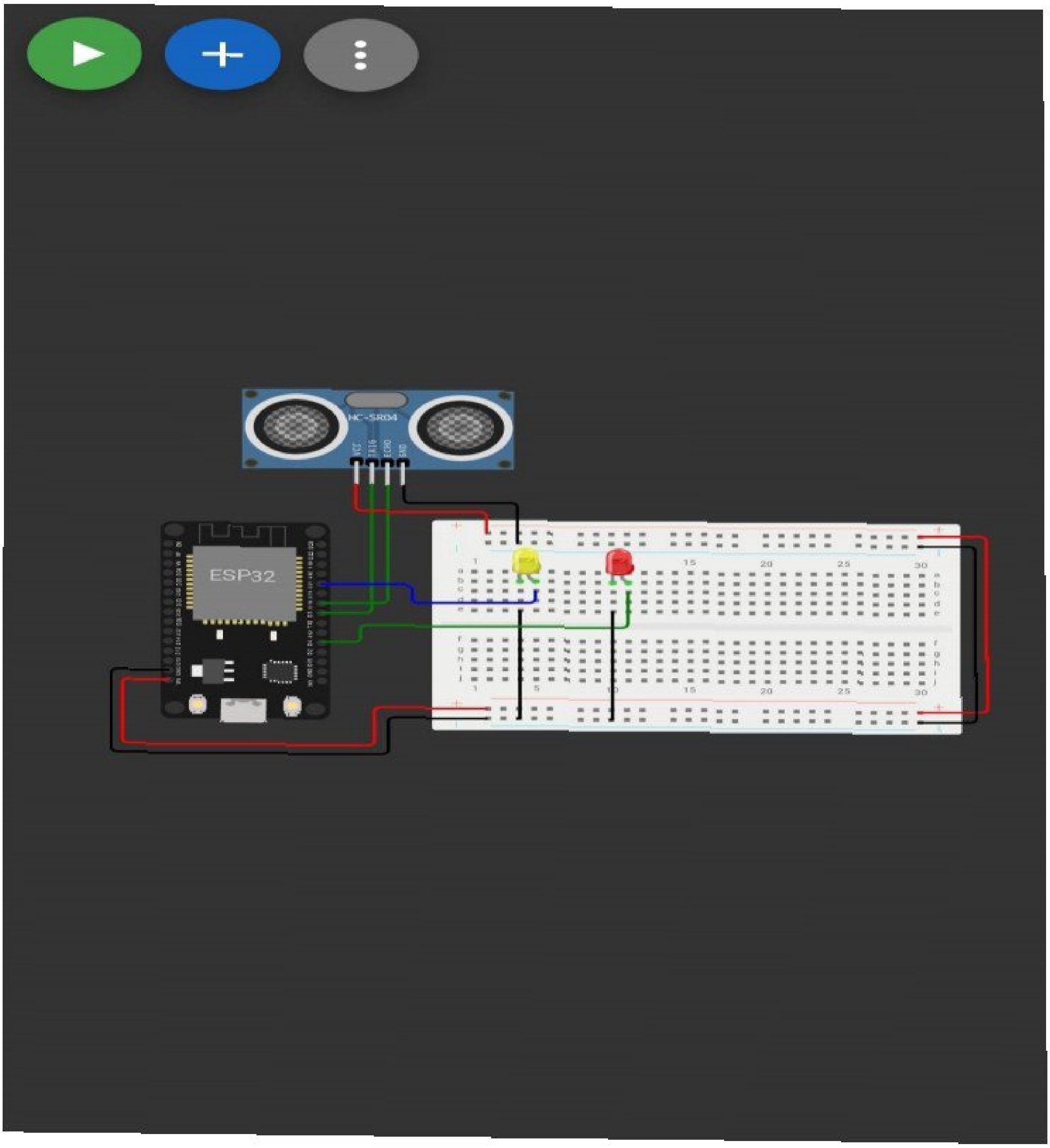
Step 1:Connect the ESP32 microcontroller with the ultrasonic sensor.

Step 2: Connect the LED's and the relay with the GPIO pins of ESP32.

Step 3: Setup each step in the code and connect Wi-Fi to the ESP32.

Step 4: Declare and initialize each component of the smart water Management. For consumption the water level "Flow sensor" is Not available in the wokwi so, instead of this we used "Ultrasonic Sensor".

Step 5: The condition for glowing LED's and motor is based on the level Of the water using the ultrasonic sensor.



HC-SR04
VCC
TRIG
ECHO
GND
ESP32
1
5
10
15
20
25
30
a b c d e
f g h i j

Step 6: Run the simulation and check the distance of the water level Through LED's glow, Motor run and push up messages.

Step 7: View the real-time data transmitting in the sensor through think speak.

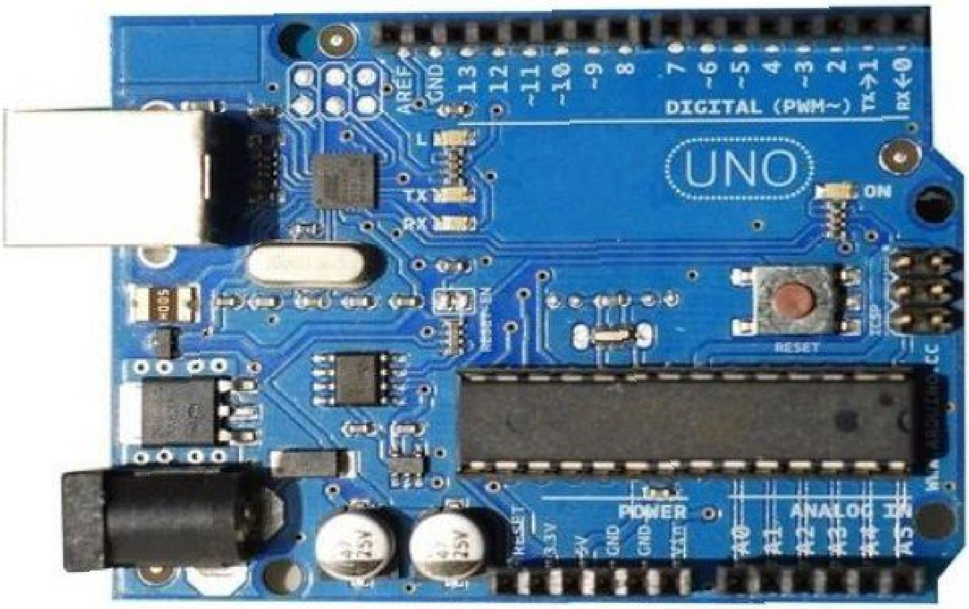
PROGRAM:

```
import time  # Import the time module for time delays
# Define GPIO pin numbers
TRIG_PIN = 2  # GPIO pin number for the ultrasonic sensor's trigger
ECHO_PIN = 3  # GPIO pin number for the ultrasonic sensor's echo
PUMP_PIN = 4  # GPIO pin number for the water pump
LED_PIN = 5   # GPIO pin number for the LED
# Initialize components (virtual components for Wokwi)
ultrasonic_sensor = Ultrasonic(TRIG_PIN, ECHO_PIN)  # Create an ultrasonic sensor
pump = Motor(PUMP_PIN)  # Create a water pump
led = LED(LED_PIN)  # Create an LED
while True:
    # Measure distance
    distance = ultrasonic_sensor.distance_cm  # Measure distance in centimeters

    if distance > 200:  # Water level is above 200 cm
        # Make the LED blink
        led.blink(on_time=0.5, off_time=0.5)  # LED blinks with 0.5 seconds on and
off time
        pump.on()  # Water pump is turned on
    else:
        # Water level is below 200 cm
        # Turn off the LED and the pump
        led.off()
        Pump.off()
    Time.sleep(0.1)
```

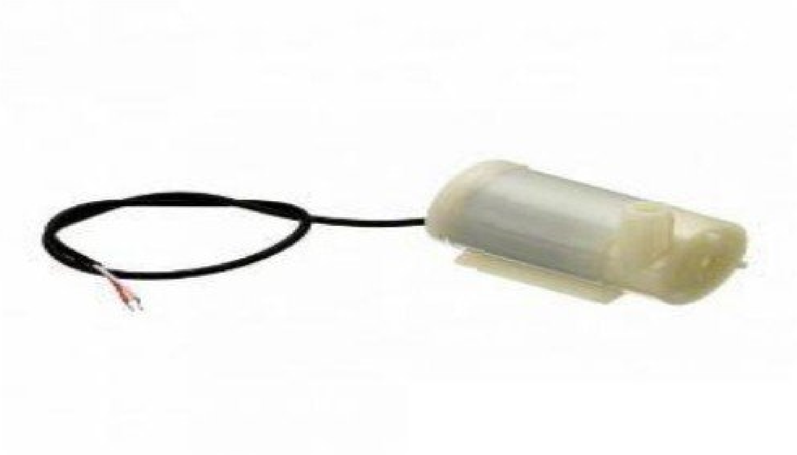
ARDUINO UNO:

Arduino UNO Arduino is an open source platform that is used in many projects . It consists of a programmable circuit board (called a microcontroller), as well as a programmable part of an integrated development environment (IDE [20]) that runs on the computer and is used to write and load code (which is a simplified version of C++) from the computer to the Arduino panel . Arduino Uno was used in this project (see Figure 4) which contains 14 digital ports (input / output) and analog inputs .



Water Pump:

Water pump or DC (direct current) powered pumps use direct current from motor, battery, or solar power to move fluid in a variety of ways. Motorized pumps typically operate on 6, 12, 24, or 32 volts of DC power (12 volt was used as shown in Figure 8). The main advantage of DC pumps is that they can operate directly from a battery , making the more convenient and portable . They are easier to operate and control.



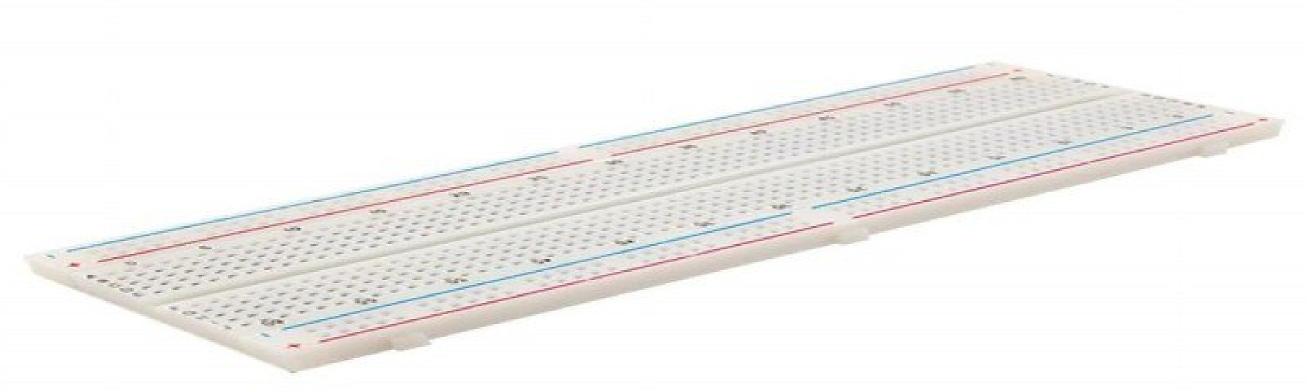
Solenoid valve :

A solenoid valve (see Figure 9) is an electromechanical device in which the solenoid uses an electric current to generate a magnetic field and thereby operate a mechanism which regulates the opening of fluid flow in a valve. The valve can use a two-port design to regulate a flow or use a three or more port design to switch flows between ports. Multiple solenoid valves can be placed together on a mainfold.

Other assisted components : These components are shown in Figure and include a breadboard, jumper wires, (12v) power supply and resistors (560Ω) .

Jumper wires

12v power supply



Breadboard