

# Smart Pet Water Dispenser

## Introducere

Soluție inovatoare concepută pentru a asigura confortul și sănătatea animalelor de companie prin furnizarea automatizată a apei proaspete. Acest dispozitiv utilizează tehnologii avansate pentru a monitoriza nivelul apei, a reumple rezervorul la nevoie și a oferi notificări în timp real către proprietari. Plecând de la ideea de a simplifica și automatiza procesul de aprovizionare cu apă pentru animalele de companie, proiectul își propune să ofere o soluție eficientă și inteligentă, atât pentru proprietari, cât și pentru animalele lor de companie.

## Descriere generală



## Hardware Design

Lista de piese:

- Microcontroller (Arduino Uno sau Arduino Nano)
- Senzor de nivel de apă
- Pompă de apă submersibilă
- Modul WiFi (ESP8266)
- Modul RTC (Real-Time Clock)
- Baterie sau sursă de alimentare
- Senzor ultrasonic
- Rezervor pentru apă
- Conectori și fire pentru conexiuni



Conexiuni:

Senzor ultrasonic

- VCC la 3.3V de la Arduino
- GND la GND de la Arduino
- Trig la pinul digital 9 de la Arduino
- Echo la pinul digital 10 de la Arduino

Pompă de apă (controlată prin tranzistor)

- E (emitter) de la tranzistor NPN la GND de la Arduino
- B (bază) de la tranzistor NPN printr-o rezistență de 220Ω la pinul digital 3 de la Arduino (PWM)

Senzor de nivel apă

- VCC la 3.3V de la Arduino
- GND la GND de la Arduino
- S (semnal) la pinul analogic A0 de la Arduino trecut prin rezistente pentru a schimba din 3.3V → 1V

## Software Design

**Mediu de Dezvoltare (IDE) :** Arduino IDE

**Librării Utilizate:**

- **ESP8266WiFi.h:** Permite conectivitatea WiFi pentru modulul ESP8266.
- **PubSubClient.h:** Permite comunicarea MQTT pentru publicarea și abonarea la topicuri.

**Structura:**

- **Librării și Constante:**
  - Include librăriile necesare.
  - Definiște credențialele WiFi, detaliile broker-ului MQTT, pinii senzorilor și variabilele globale.
- **Funcții:**
  - **setup\_wifi():** Conectează ESP8266 la rețeaua WiFi specificată.
  - **callback(char topic, byte payload, unsigned int length):** Funcție de tip callback pentru procesarea mesajelor MQTT primite.
  - **reconnect():** Reconectează la broker-ul MQTT dacă conexiunea este pierdută.
  - **setup():** Inițializează comunicarea serială, WiFi, MQTT și pinii senzorilor.
  - **loop():** Bucla principală care verifică conexiunea MQTT și verifică periodic prezența animalului de companie.
  - **checkPetPresence():** Folosește senzorul ultrasonic pentru a detecta prezența unui animal și verifică nivelul apei dacă nu este detectat niciun animal.
  - **readSensor():** Citește senzorul de nivel al apei și returnează citirea curentă.

**Algoritmi Implementați**

- **Conexiunea WiFi:**
  - Încearcă să se conecteze la rețeaua WiFi specificată.
  - Afișează starea conexiunii și adresa IP odată conectat.

- **Conexiunea și Reconectarea MQTT:**
  - Se conectează la broker-ul MQTT cu un ID de client unic.
  - Se abonează la topicul MQTT specificat.
- **Citirea Senzorului Ultrasonic:**
  - Trimite un puls de declanșare și măsoară durata pulsului de ecou.
  - Calculează distanța bazată pe durata pulsului.
- **Verificarea Nivelului de Apă:**
  - Citește valoarea analogică de la senzorul de nivel al apei.
  - Pornește pompa dacă nivelul apei este prea scăzut.
  - Trimite un mesaj MQTT dacă pompa a fost activată de mai multe ori.

## Concepte Folosite

### 1. GPIO (General Purpose Input/Output)

- **Pini de Sensori și Actuatori:** Utilizarea pinilor GPIO pentru a controla senzorii și pompa de apă.
- **Pini Definiți:**
  - `sensorPower`` (pin 13): Controlul alimentării senzorului de nivel al apei.
  - `trigPin`` (pin 12) și `echoPin`` (pin 14): Controlul senzorului ultrasonic pentru măsurarea distanței.
  - `pumpPin`` (pin 15): Controlul pompei de apă.
- **Operații GPIO:**
  - `pinMode()``: Setarea modului pinilor ca intrare sau ieșire.
  - `digitalWrite()``: Scrierea valorilor HIGH sau LOW pe pini pentru a controla senzorii și pompa.

### 2. UART (Universal Asynchronous Receiver-Transmitter)

- **Comunicare Serială:** Utilizarea interfeței seriale pentru a debuga și monitoriza activitatea sistemului.
- **Operații UART:**
  - `Serial.begin(9600)``: Inițializarea comunicării seriale la o viteză de 9600 bps.
  - `Serial.print()`` și `Serial.println()``: Trimiterea mesajelor și datelor către monitorul serial pentru diagnosticare.

### 3. ADC (Analog-to-Digital Converter)

- **Citirea Senzorului de Nivel al Apei:** Utilizarea pinului analogic pentru a citi valorile de la senzorul de nivel al apei.
- **Operații ADC:**
  - `analogRead(sensorPin)``: Citirea valorii analogice de la pinul senzorului pentru a determina nivelul apei.

### 4. Timere

- **Managementul Timpului:** Utilizarea funcțiilor de delay și millis pentru a gestiona temporizarea în sistem.
- **Operații cu Timere:**
  - `delay()``: Introducerea unor întârzieri pentru stabilizarea senzorilor și activarea pompei.
  - `millis()``: Verificarea timpului scurs pentru a efectua verificări periodice ale senzorilor (de

exemplu, verificarea prezenței animalului de companie la fiecare 5 secunde).

## 5. WiFi

- **Conectivitate la Rețea:** Utilizarea modulului WiFi integrat în ESP8266 pentru a conecta dispozitivul la o rețea WiFi și pentru a comunica cu broker-ul MQTT.
- **Operații WiFi:**
  - `WiFi.begin(ssid, password)`: Inițierea conexiunii la rețeaua WiFi specificată prin SSID și parolă.
  - `WiFi.status()`: Verificarea stării conexiunii WiFi.
  - `WiFi.localIP()`: Obținerea adresei IP alocate dispozitivului în rețeaua locală.

## 6. MQTT (Message Queuing Telemetry Transport)

- **Protocol pentru Comunicarea M2M:** Utilizarea MQTT pentru transmiterea datelor între dispozitivele IoT și broker-ul MQTT.
- **Operații MQTT:**
  - `client.setServer()`: Configurarea serverului MQTT.
  - `client.setCallback()`: Setarea funcției de callback pentru gestionarea mesajelor primite.
  - `client.connect()`: Conectarea la broker-ul MQTT.
  - `client.subscribe()`: Abonarea la un anumit topic MQTT.
  - `client.publish()`: Publicarea unui mesaj pe un topic MQTT.

## Rezultate Obținute

Codul trebuie modificat pentru a schimba refill-ul cu formula 10 umpleri/L.

### Video cum funcționează:

[https://drive.google.com/file/d/1RldX3eXlpAgX2A6bU9fj-tDBX\\_Kzxtoz/view?usp=sharing](https://drive.google.com/file/d/1RldX3eXlpAgX2A6bU9fj-tDBX_Kzxtoz/view?usp=sharing)

## Concluzii

Săptămâna	Sarcină	Descriere
21 Apr – 5 Mai	Documentare inițială	Cercetare privind componentele și soluțiile similare.
21 Apr – 5 Mai	Achiziționare componente	Achiziționarea microcontrolerului, senzorilor, pompei, etc.
21 Apr – 5 Mai	Prototipare hardware	Asamblarea circuitului pe breadboard.
13 Mai – 17 Mai	Programare inițială	Scrierea codului pentru citirea senzorilor și controlul pompei.
13 Mai – 17 Mai	Integrare senzori	Implementarea funcționalităților pentru senzorii de nivel și calitate a apei.
13 Mai – 17 Mai	Control PWM și întreruperi	Configurarea PWM pentru pompa de apă și întreruperilor pentru nivelul scăzut de apă.
17 Mai – 22 Mai	Testare și calibrare	Testarea sistemului și ajustarea parametrilor pentru funcționare optimă.
17 Mai – 22 Mai	Documentare finală	Redactarea documentației pentru proiect.

17 Mai - 22 Mai	Prezentare și revizuire	Prezentarea proiectului și revizuirea finală pe baza feedback-ului.
-----------------	-------------------------	---

## Download

<https://github.com/YANNICKKO/SmartWaterDispenser>

## Bibliografie/Resurse

Biblioteci:

- <https://github.com/knolleary/pubsubclient>
- <https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/src/ESP8266WiFi.h>

Software:

- <https://ocw.cs.pub.ro/courses/iotthings/laboratoare/lab7>
- <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
- <https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/>

Hardware:

- <https://www.adafruit.com/product/2821>

[Export to PDF](#)

From:  
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/yannick.koning>



Last update: **2024/05/26 18:11**