

RAPPORT DE PROJET:

SMART AQUARIUM : CONCEPTION ET IMPLÉMENTATION D'UN SYSTÈME INTELLIGENT POUR LA GESTION ET LA MAINTENANCE D'AQUARIUM

Travail réalisé par :

KABADOU NERMINE

OTHMANI SOUMAYA

Aguel Arij

Guermazi Yosr

Professeur encadrant:

DAMERGI EMIR

I. ETUDE PRELIMINAIRE:

1. Besoins & public cible :

1.1. Résumé exécutif :

Introduction au Smart Aquarium :

Le smart aquarium est un système d'aquarium connecté qui intègre des technologies avancées telles que la surveillance en temps réel, l'automatisation de certaines fonctionnalités et la connectivité avec des applications mobiles.

Opportunités identifiées et Défis Potentiels :

- Opportunité :
 - Satisfaire la demande croissante des produits hautement technologiques pour animaux de compagnie à savoir :
 - Réduire le temps d'entretien grâce à l'automatisation
 - Améliorer la santé des poissons grâce à une surveillance constante
 - Expérience utilisateur améliorée avec une interface conviviale
- Défi :
 - Éduquer les consommateurs sur les avantages du smart aquarium par rapport aux aquariums traditionnels.

1.2. Analyse du marché:

Taille du marché des Aquariums :

Selon le rapport de marché "Pets and Aquatics Global Report", le marché mondial des produits pour animaux de compagnie, y compris les aquariums, était estimé à 75 milliards de dollars en 2022. Cette valeur est soutenue par une augmentation constante du pouvoir d'achat des consommateurs et une préférence croissante pour des solutions innovantes dans le domaine des animaux de compagnie.

Tendances du marché des produits pour animaux de compagnie :

L'évolution des attentes des consommateurs en matière de produits pour animaux de compagnie est un facteur clé. Selon une étude de marché menée par "PetTech Insights, les consommateurs sont de plus en plus enclins à adopter des technologies pour faciliter la vie avec leurs animaux de compagnie.

Analyse de la demande potentielle pour les Smart Aquariums :

Une enquête en ligne auprès de 1500 propriétaires d'aquariums dans les grandes villes a révélé les points suivants :

- Intérêt pour la technologie : 80 % des répondants ont indiqué être intéressés par des technologies facilitant la maintenance de l'aquarium.
- Préoccupations de santé des poissons : 65 % ont exprimé des préoccupations liées à la santé des poissons et seraient prêts à investir dans des solutions de surveillance en temps réel.
- Âge et Démographie : Les consommateurs de 25 à 40 ans représentent la majorité des personnes intéressées par les smart aquariums, indiquant un attrait particulier pour les nouvelles technologies.

1.3. Segmentation du marché :

1.3.1. Segments de clients potentiels & évaluation de leurs besoins spécifiques :

Amateurs de poissons tropicaux :

❖ Caractéristiques :

Les passionnés de poissons tropicaux sont souvent des propriétaires expérimentés d'aquariums, attachés à la maintenance précise des conditions de l'eau.

❖ Besoins :

Un contrôle précis des paramètres de l'eau, des alertes instantanées en cas de changements anormaux et une interface conviviale pour ajuster les conditions de l'aquarium.

Familles avec enfants :

❖ Caractéristiques :

Les familles cherchent souvent des solutions d'aquarium qui combinent éducation et divertissement pour les enfants.

❖ Besoins :

Un smart aquarium avec des fonctionnalités éducatives intégrées, des jeux interactifs pour les enfants, et une conception sécurisée.

Entreprises (Restaurants, Bureaux) :

❖ Caractéristiques :

Les entreprises recherchent des éléments décoratifs attrayants avec une maintenance minimale.

❖ Besoins :

Un smart aquarium esthétique, facile à entretenir, et capable d'impressionner les clients ou les visiteurs. La fonction de contrôle à distance est également un avantage.

Nouveaux propriétaires d'aquariums :

❖ Caractéristiques :

Les débutants dans le monde des aquariums peuvent rechercher des solutions simples pour commencer.

❖ Besoins :

Un smart aquarium avec des guides intégrés, une assistance automatisée pour l'entretien de base, et une application conviviale pour les débutants.

Collectivités et établissements d'éducation :

❖ Caractéristiques :

Des écoles, musées, ou autres établissements éducatifs peuvent être intéressés par des solutions éducatives liées à l'aquaculture.

❖ Besoins :

Des fonctionnalités éducatives interactives, la possibilité de partager des données en temps réel, et des options de personnalisation pour des projets éducatifs spécifiques.

1.3.2. Concurrence :

Les principaux concurrents dans le domaine des aquariums intelligents incluent des marques établies telles que Smart Aquarium Solutions et TechFish Innovations.

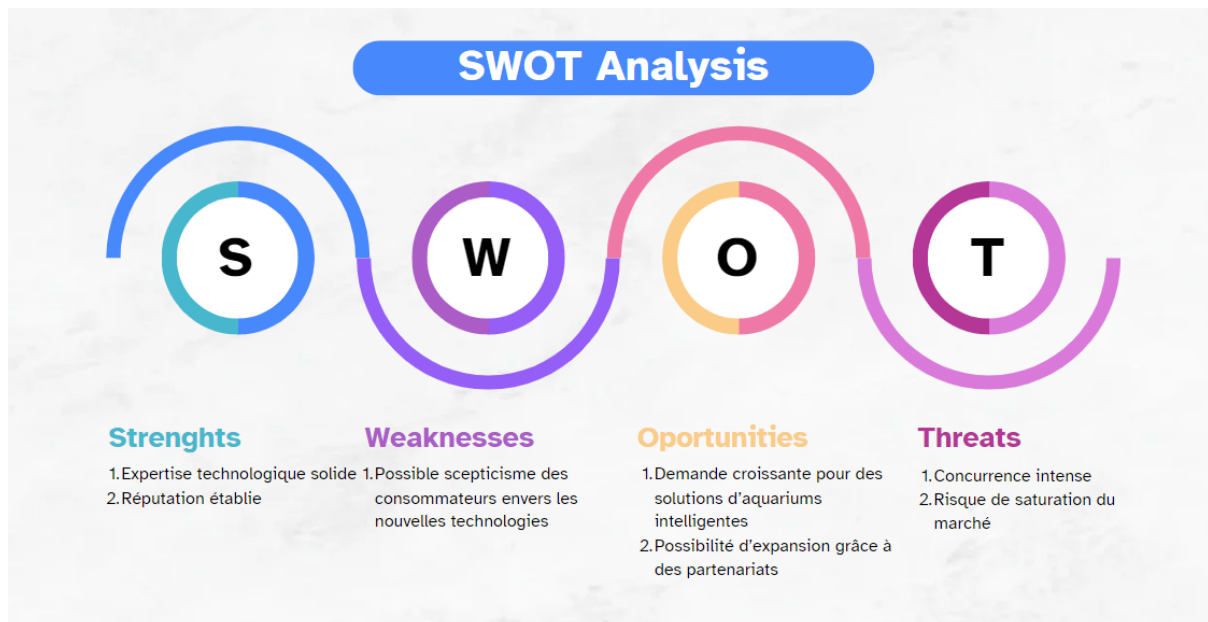
- SmartAquarium Solutions :
 - Forces : Une forte présence dans le marché des produits pour animaux de compagnie, une technologie bien établie, et des fonctionnalités de pointe telles que la reconnaissance vocale pour interagir avec les poissons.
 - Faiblesses : Les prix élevés peuvent être un obstacle pour certains segments de marché.
- TechFish Innovations :

- Forces : Des solutions innovantes avec une approche axée sur l'expérience utilisateur, une connectivité avancée, et une forte réputation pour la durabilité de leurs produits.
- Faiblesses : uniquement disponible via des canaux de distribution en ligne, ce qui pourrait limiter la visibilité en magasin.

Bien que la concurrence soit présente, il existe un espace significatif pour la différenciation en mettant l'accent sur la convivialité et les fonctionnalités innovantes.

- ❖ Opportunités pour se démarquer :
 - Innovation continue
 - Stratégies de prix compétitives
 - Collaborations et Partenariats
 - Service client exceptionnel

1.3.3. Analyse SWOT :



1.3.4. Analyse de la demande :

Motivations d'achat des consommateurs :

- -Facilité d'entretien : des solutions simplifiant l'entretien des aquariums
- -Bien-être des animaux de compagnie : Préoccupation croissante pour la santé des poissons
- -Expérience utilisateur : les fonctionnalités intelligentes et l'interface conviviale

Facteurs influençant la décision d'achat :

- Prix

- Fiabilité et Durabilité
- Avis et Témoignages

Prix Approximatif

Le coût total du Smart Aquarium sera influencé par les tarifs des composants, des matériaux et des dépenses liées à la production. Pour une approximation générale, le système du Smart Aquarium est estimé à être vendu à peu près à 300.190 DT.

2. Présentation de l'Aquarium Intelligent:

L'aquarium intelligent décrit dans le cadre de notre projet IoT comporte plusieurs aspects et fonctionnalités dédiés principalement à la surveillance et à la gestion de différents paramètres d'un aquarium.

2.1. Fonctionnalités du Smart Device :

a. Contrôle de l'Éclairage :

- Mesure du niveau de luminosité dans l'aquarium par des capteurs de lumière
- Ajustement de la couleur de LED RGB .

b. Gestion de la température :

- Mesure de la température de l'eau via un capteur de température (DS18B20)
- Envoi d'alerte si la température de l'eau dépasse ou descend en dessous de certains seuils définis.

c. Surveillance du pH :

- Mesure du pH de l'eau de l'aquarium

d. Gestion du niveau d'eau :

- Surveillance du niveau d'eau pour maintenir un niveau optimal.
- Envoi d'alertes en cas de niveau d'eau bas.

e. Distribution de nourriture:

-Garantir l'alimentation du poisson en utilisant un distributeur de nourriture

2.2. Architecture materielle:

a. Microcontrôleur (SOC - System on Chip):

- Le choix du SOC s'est porté sur l'ESP 32 en raison de sa polyvalence, de sa connectivité Wi-Fi intégrée et de sa capacité à gérer plusieurs capteurs et actionneurs.

b. Capteurs :

- Capteur de température numérique : Mesure la température de l'eau de l'aquarium.
- Capteur de niveau d'eau numérique : Surveille le niveau d'eau pour éviter les débordements.
- Capteur de luminosité : Ajuste l'éclairage en fonction de la luminosité ambiante.
- Capteur de niveau de pH : Mesure le niveau de pH de l'eau pour assurer un environnement sain pour les poissons.

c. Actionneurs :

- Système d'éclairage LED contrôlé : Permet d'ajuster l'éclairage de l'aquarium pour imiter les conditions naturelles.
- Servomoteur : Utilisé pour la distribution automatique de nourriture, offrant une alimentation régulière aux poissons.
- Ventilateur : Régule la température en activant le refroidissement lorsque nécessaire.

2.3. Moyens de Communication sans Fil :

- Wi-Fi : Assure la connectivité avec l'application mobile, permettant le contrôle à distance et la surveillance en temps réel.
- MQTT : Protocole de communication léger pour l'échange de données entre le smart aquarium et le serveur MQTT.

2.4. Communication avec l'Utilisateur Final :

Notre système intelligent offre une expérience utilisateur complète par le moyen d'une application mobile dédiée. L'application permet à l'utilisateur de visualiser les données en temps réel telles que la température, le niveau d'eau, la luminosité et le pH. De plus, elle offre des fonctionnalités interactives telles que la modification des paramètres, le déclenchement manuel de la distribution de nourriture et la réception des alertes en cas de d'anomalies détectées à l'intérieur de l'aquarium. La connectivité Wi-Fi joue un rôle très important pour l'interaction entre l'application et le smart aquarium, offrant ainsi un contrôle total et une surveillance continue et à distance.

2.5. Coût Unitaire Total :

Le coût unitaire total du Smart Aquarium dépendra des prix des composants, des matériaux, et des coûts de fabrication. Voici une estimation générale :

Composant	Coût unitaire(DT)
ESP32	55
Capteur de Température DS18B20	9.200
Capteur à ultrasons HC-SR04	13.990
Bande à LEDs Neopixel	16.000
Capteur de pH	175.000
Servomoteur	16.000
Ventilateur	15.000
TOTAL:	300.190

II. ETUDE DETAILLEE:

1. Consultation et Mise à jour des Paramètres variables :

Notre Smart Device aura besoin d'un certain nombre de paramètres variables nécessaires pour la configuration et la personnalisation du système intelligent. Ces paramètres pourraient être mis à jour en temps réel par l'utilisateur final.

Afin de pouvoir envoyer des messages MQTT au courtier MQTT, le dispositif intelligent doit être connecté à Internet. Ainsi, l'utilisateur final est tenu de configurer la connexion Internet.

[Paramètres pour l'établissement de la connexion Internet : SSID, Password ?]

*SSID : Nom du point d'accès

*Password : Mot de passe du point d'accès

```
const char* ssid = "Wokwi-GUEST";  
const char* password = "";
```

[Informations sur le serveur MQTT : Port, URL ?]

```
/**Set server**/  
const char* mqttServer = "broker.hivemq.com";  
int port = 1883;
```

Format des données : Chaîne de caractères

Consultation et mise à jour de la configuration de la connexion Internet :

L'utilisateur peut consulter la configuration actuelle de la connexion Internet ou/et la mettre à jour via une interface mobile.

2. Acquisition et contrôle :

Voici une brève illustration de notre conception pour l'appareil intelligent, mettant en lumière la manière dont les pins et les divers composants sont interconnectés.

2.1. Les capteurs:

2.1.1. Capteur de Température:

Reference: Capteur de temperature DS18B20



C'est une sonde étanche basée sur un DS18B20 permettant de mesurer une température de -55 à + 125 °C. Elle se raccorde sur un microcontrôleur via une entrée numérique.

Ça nous permet de mesurer la température ambiante dans notre smart aquarium pour la contrôler, c'est-à-dire se stabiliser entre les 2 valeurs déjà mentionnées (20° et 30°).

Ce capteur est connecté à l'ESP 32 via le tempPin (32).

```
int tempPin = 32;
```

On affiche au fur et à mesure la valeur de la température sur l'écran LCD. S'il détecte une anomalie dans la température, le Smart Device va allumer le buzzer (l'avertisseur sonore) pendant 1 seconde, avertir le client par MQTT et mettre en marche le ventilateur (puisque le ventilateur n'est pas disponible dans le simulateur wokwi, on a opté pour un diode Led qui s'allumera au déclenchement).

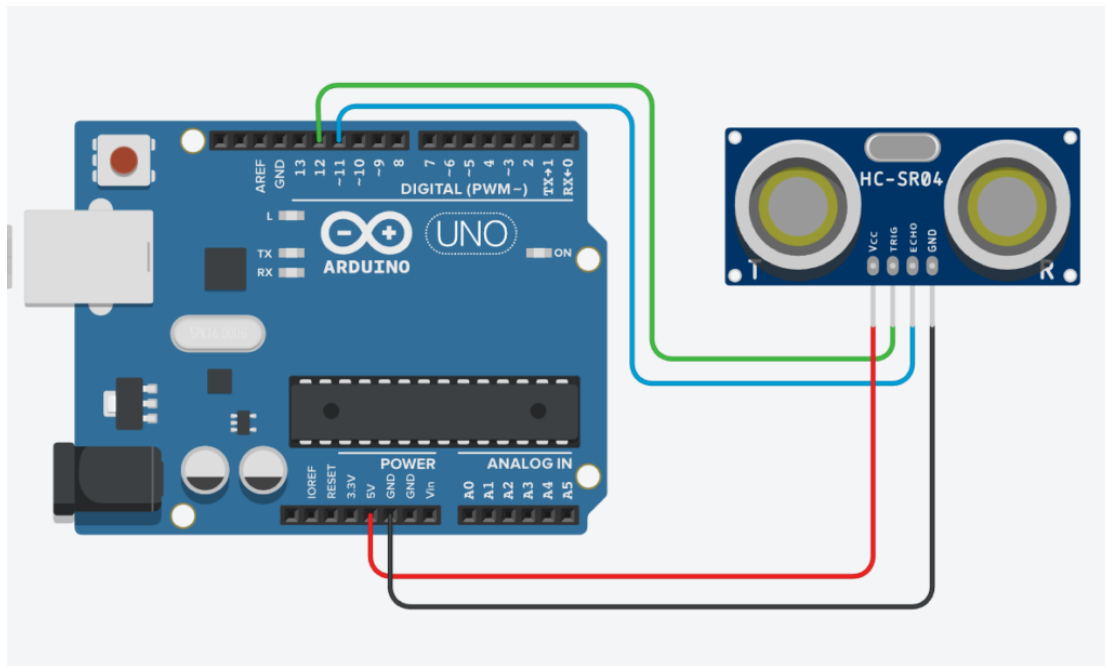
Avantages et caractéristiques:

- Alimentation,: 3 à 5 vcc
- Précision :0.5 % de -10 à +85°C
- Plage de mesure: -55 à +125°C
- Dimension du sonde: 06 x 35mm
- Largeur cordon: 90 cm
- Mise en oeuvre automatique
- Se raccorde sur une entrée digitale

2.1.2. Capteur de niveau d'eau:

Type: Capteur à ultrasons

Référence: Capteur à ultrasons HC-SR04



Étant donné que les ondes ultrasonores peuvent se refléter sur une surface liquide et revenir à la tête du capteur, le capteur à ultrasons est la meilleure option pour notre application.

De plus, la détection n'est pas affectée par l'accumulation de poussière ou de saleté.

Le capteur à ultrasons utilise deux broches principales:

- **trigUltraPin(23)** : c'est un pin d'entrée responsable d'initier la mesure de distance. Un signal envoyé à la broche déclenche l'émission d'une impulsion ultrasonique par le capteur.
- **echoUltraPin(22)** : c'est un pin de sortie qui génère une impulsion de largeur variable en fonction du temps que prend l'onde ultrasonique pour revenir après sa réflexion sur la surface

```
int trigUltraPin = 23;  
int echoUltraPin = 22;
```

Finalement, le calcul de la distance se fait à partir de la durée pendant laquelle la broche echo reste à HIGH, cela correspond au temps que prend l'onde pour revenir après toucher la surface de l'eau.

```
long duration = pulseIn(echoUltraPin, HIGH);  
long distanceCm = duration * 0.034 / 2;  
//distance = vitesse son dans l'eau * durée aller retour / 2
```

Ça nous permet de mesurer le niveau d'eau dans notre smart aquarium pour le contrôler.

Ce capteur est connecté à l'ESP 32 via la broche waterLevelPin (35) par le code suivant:

```
int waterLevelPin = 35;
```

On affiche au fur et à mesure la valeur du niveau d'eau sur l'écran LCD. S'il détecte une anomalie (<200 cm), le Smart Device va allumer le buzzer (l'avertisseur sonore) pendant 1 seconde, avertir le client par MQTT et mettre en marche la pompe à eau(puisque la pompe n'est pas disponible dans le simulateur wokwi, on a opté pour un diode Led qui s'allumera au déclenchement).

Avantages et caractéristiques:

La technologie utilisée par les capteurs à ultrasons est sans contact, ce qui confère à ce type de capteur la plupart des avantages dont ils disposent.

On peut citer les bienfaits suivants:

- ➔ Sans usure : Les capteurs à ultrasons n'ont pas de pièces mobiles ; cela les rend complètement inusables et leur donne une durée de vie plus longue que les alternatives de contact.
- ➔ Très précis : En raison de leur mode de fonctionnement, les capteurs à ultrasons sont très précis et peuvent être utilisés pour détecter de très petites modifications de position. Ils peuvent également mesurer l'épaisseur d'un objet ainsi que la profondeur de la surface parallèle.
- ➔ Détection de forme : D'autres types de capteurs de position peuvent détecter la position d'un objet, mais là où un capteur à ultrasons excelle, c'est qu'ils peuvent également détecter la taille et la forme des objets. Cela les rend beaucoup plus flexibles en termes d'utilisation de leurs applications.
- ➔ Détecter une gamme de matériaux : Les capteurs de position à ultrasons peuvent détecter et mesurer des objets indépendamment de leur surface ou de leur couleur.
- ➔ Facile à utiliser : ces capteurs peuvent facilement s'interfacer avec un microcontrôleur et ils ne sont pas dangereux à utiliser.

- Alimentation : 5 vcc

- Dimension: 60 x 21 x 7mm
- Distance de mesure théorique : 2 cm à 450 cm
- Distance de mesure pratique : 2 cm à 80 cm
- Angle de mesure couvert : <15°
- Courant de fonctionnement : < 15 mA
- Fréquence de fonctionnement : 40 Hz

2.1.3. Capteur de niveau du pH :



Description:

- Puissance du module: 5,00 V
- Taille du module: 43 x 32 mm (1,69 x 1,26 ")
- Plage de mesure: 0 - 14 pH
- Mesure de la température: 0 - 60 °C
- Précision: $\pm 0,1$ pH (25 °C)
- Temps de réponse: ≤ 1 min
- Capteur de pH avec connecteur BNC
- Interface pH2.0 (patch de 3 pieds)
- Potentiomètre de réglage du gain
- Voyant d'alimentation

2.2. Les actionneurs:

2.2.1. Servomoteur pour le Distributeur de Nourriture:

Type: MICRO SERVO SG90 9G

Reference: TPSG90



Avantages et caractéristiques:

Le Micro Servo Module est composé d'un moteur électrique mécaniquement lié à un potentiomètre. On utilise la librairie [Servo Library](#) pour pouvoir piloter facilement ce module avec un kit Arduino.

L'électronique à l'intérieur du servo-moteur transforme une largeur d'impulsion PWM en position physique : Quand le servo est commandé, le moteur sera actionné jusqu'à atteindre la valeur du potentiomètre correspondante à la position demandée

- Modulation : Analogique
- Force : 4.8V (1.6 kg-cm)
- Vitesse : 4.8V 0.1 sec/60°
- Poids : 9g
- Dimensions : 23mm x 12.2mm x 29 mm
- Angle de rotation : 180°
- Connectique : Connecteur 3 points

Le servomoteur est utilisé pour contrôler le distributeur de nourriture. Il est activé pour une certaine durée et peut également être activé manuellement si le bouton correspondant est enfoncé et est connecté à l'ESP via le `fishFeederPin(33)`.

```
int fishFeederPin = 33;
```

2.2.2. Ventilateur :

Type: Ventilateur DC

Reference: Ventilateur DC 12V



Avantages et caractéristiques:

- Tension d'alimentation: 12V DC
- Consommation: 1.58 w
- Performance: 44.62 m3/h
- Poids : 35.5g
- Dimensions : 60mm x 60mm x 15 mm
- Température de travail : [-10 °, 70°]
- Vitesse de rotation: 4700 (±20%) rot/min

Le ventilateur est utilisé pour ajuster la température de l'aquarium. Il est activé au cas où la température dépasse une valeur bien spécifiée (30°C) et est désactivée lorsqu'elle atteint 20°C.

```
int turnOnFanPin = 15;
```

2.2.3. Eclairage à LED :

Type: Led neopixel RGB

Reference: Bande à LEDs ADA1376



Avantages et caractéristiques:

Ruban NeoPixel RGB 1 mètre équipé de 30 LEDs basées sur le WS2812B avec protection en silicone et communiquent avec un microcontrôleur type Arduino ou compatible via une sortie série 1 broché .

- Tension d'alimentation: 5 vcc
- Consommation: 60 mA par LED (3 couleurs au maxi)
- Nombre de LEDs : 30
- Raccordement : pastilles à souder
- Poids : 35.5g
- Longueur : 1 mètre
- Largeur: 12 mm

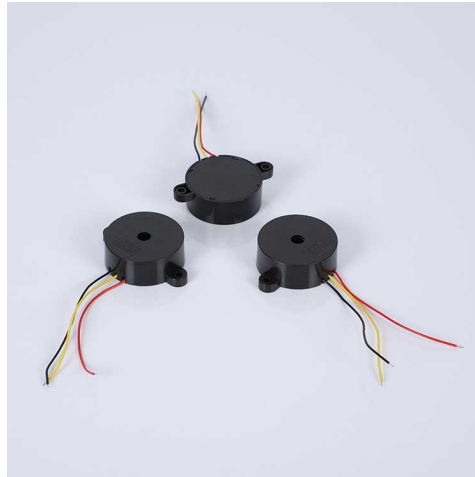
Cette bande est utilisée pour afficher des effets lumineux. Elle s'allume en rouge en cas d'alerte de température ou de niveau d'eau anormal.

```
//Led
int LED_COUNT = 16;
int LED_PIN = 19;
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(LED_COUNT, LED_PIN, NEO_GRB+NEO_KHZ800);
```

2.2.4. Buzzer :

Type: Led neopixel RGB

Reference: Bande à LEDs ADA1376



Avantages et caractéristiques:

Le buzzer est utilisé pour émettre un signal d'alerte lorsque des conditions anormales apparaissent dans l'aquarium. Il émet un signal sonore pendant une seconde

- Tension: 5V
- Diamètre : 11,9 mm/0,47
- Poids : 1g
- Fréquence : 2KHz
- Hauteur : 8,4 mm / 0,33
- Sortie sonore minimum: 85 db à 10 cm
- Fréquence de résonance: 2500 hz \pm 300 Hz

Le buzzer est connecté à l'ESP via le buzzerPin(4):

```
int buzzerPin = 4;
```

3. Traitement

Notre projet IOT est essentiellement un système intelligent conçu pour la gestion et le contrôle d'un aquarium. En fonction des différentes données acquises en temps réel, notre système couvre plusieurs traitements et fonctionnalités pour résoudre les potentiels problèmes et par conséquent maintenir un état stable à l'intérieur de l'aquarium.

Voici ci-dessous ces traitements:

- Mesure de température à l'intérieur de l'aquarium par un capteur de température qui renvoie une valeur analogique convertie du Kelvin en

degré Celsius. Ceci est implémenté dans notre code de la manière suivante:

```
float readTemperature() {
    const float VCC = 3.3;           // NodeMCU on board 3.3v vcc
    const float R2 = 10000;          // 10k ohm series resistor
    const float adc_resolution = 4096; // 12-bit adc
    const float A = 0.001129148;     // thermistor equation parameters
    const float B = 0.000234125;
    const float C = 0.0000000876741;

    float Vout, Rth, temperature, adc_value;

    adc_value = adc_resolution - analogRead(tempPin) + 0.5; // switch direction
    Vout = (adc_value * VCC) / adc_resolution;
    Rth = (VCC * R2 / Vout) - R2; // Formula for R2 as Pull-down: Vcc-Rth-R2-GND

    temperature = (1 / (A + (B * log(Rth)) + (C * pow((log(Rth)), 3)))); // Temperature in kelvin

    temperature = temperature - 273.15; // Temperature in degree celsius
    Serial.print(" Temperature = ");
    Serial.print(temperature);
    Serial.println(" degree celsius");
    return temperature;
}
```

- Mesure du niveau d'eau par un capteur à ultrasons qui calcule la distance entre la surface d'eau dans l'aquarium et le capteur. Cette fonctionnalité est implémentée de la façon suivante:

```

long getDistance() {
    //distance par rapport à un obstacle (changement de milieu)
    digitalWrite(trigUltraPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigUltraPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigUltraPin, LOW);
    |
    long duration = pulseIn(echoUltraPin, HIGH);
    long distanceCm = duration * 0.034 / 2;
    //distance = vitesse son dans l'eau * durée aller retour / 2

    return distanceCm;
}

```

- Détecter les anomalies et activer le buzzer. Ceci est assuré par la boucle loop qui surveille continuellement si les conditions de l'aquarium ne sont pas en dehors des plages normales. Si la température est inférieure à 20°C ou supérieure à 30°C ou le niveau d'eau est inférieur à 200 cm notre système déclenche le buzzer qui serve d'alerte pour signaler les irrégularités.

```

// Check temperature conditions for the buzzer alert
if ((celcius < 20 || celcius > 30) ) {
    // Turn on the buzzer for 1 second
    tone(buzzerPin, 300, 1000);
    Serial.println("Buzzer alert! : the temperature must be between 20 and 30 Celsius");
    colorWipe(pixels.Color(255, 0, 0), 50); // Red
}
else
    // Check distance conditions for the buzzer alert
    if ( (distanceCm < 200)) {
        // Turn on the buzzer for 1 second
        tone(buzzerPin, 300, 1000);
        Serial.println("Buzzer alert! the distance must be greater than 200 cm");
        colorWipe(pixels.Color(255, 0, 0), 50); // Red
    }
    else {
        colorWipe(pixels.Color(0, 255, 0), 50); // Green
    }
}

```

- Activer le servomoteur du distributeur de nourriture à des intervalles de temps réguliers et cette fonction peut être aussi bien manuelle que programmée pour des moments spécifiques.

```

fishFeeder.write(90);
delay(1000);
fishFeeder.write(0);
if (digitalRead(turnOnFishFeederBtnPin) == LOW) {
  fishFeeder.write(150);} // Operate the fish feeder

```

- Donner des mise à jour et des notifications en temps réel via MQTT pour toujours tenir l'utilisateur final alerté de tout changement qui se déroule dans son aquarium et lui permettre de maintenir un bon environnement pour son animal de compagnie.
- On a implémenté cela dans notre code de cette manière:

```

// Function to publish data to MQTT topics
void publishData(float temperature, float pH, float distance, float waterLevel) {
  mqttClient.publish("ESP32Client-/temperature", String(temperature).c_str());
  mqttClient.publish("ESP32Client-/pH", String(pH).c_str());
  mqttClient.publish("ESP32Client-/distance", String(distance).c_str());
  mqttClient.publish("ESP32Client-/water_level", String(waterLevel).c_str());

  if ((temperature < 20 || temperature > 30) ) {
    mqttClient.publish("ESP32Client-/AlertTemperature", "Alert! :T° n'est pas entre 20 et 30°");
  }

  if ( (distance < 200)) {
    mqttClient.publish("ESP32Client-/AlertDistance", "Alert! Distance< 200cm! ");
  }
}

// Publish data to MQTT topics
mqttClient.loop();
if (mqttClient.connected()) {
  Serial.print("connected to mqtt");
  publishData(celcius, pH, distanceCm, waterLevel);
}

```

Une fois le message MQTT reçu par le dispositif IoT, le système affiche le contenu du message sur la console série. La visualisation des informations reçues de manière claire et compréhensible est une approche qui assure le débogage des échanges entre le smart aquarium et le système MQTT et fournit

conséquent la visibilité nécessaire pour le suivi et le contrôle du dispositif. Voici le code correspond à cette fonction:

```
//callback: appelée lorsqu'un message MQTT est reçu.  
//Elle affiche le sujet du message et le contenu du message dans la console.  
void callback(char* topic, byte* message, unsigned int length) {  
    Serial.println(topic);  
    String strMsg;  
    for(int i=0; i<length; i++) {  
        strMsg += (char)message[i];  
    }  
    Serial.println(strMsg);  
}
```

4. Communication:

Les utilisateurs de notre projet ont évidemment besoin d'accéder à l'Aquarium Intelligent à distance dans le but d'envoyer des ordres et/ou bien récupérer les données enregistrées tels que la température, le niveau d'eau, le pH.

Dans ce contexte, la communication est établie par le biais des deux protocoles Wi-Fi et MQTT.

4.1. Configuration :

Notre code arduino démarre avec les bibliothèques nécessaires pour ce projet. Dans notre cas, la bibliothèque WiFi.h est utilisée pour gérer les fonctions qui aident à connecter la carte ESP32 au réseau Wi-Fi. Les bibliothèques ESPAsyncWebServer et AsyncTCP sont nécessaires pour construire le serveur Web HTTP asynchrone. La bibliothèque PubSub Client.h est utile pour implémenter le protocole MQTT et assurer la communication avec le broker et donc l'échange entre l'ESP32 et les clients MQTT. La bibliothèque HTTPClient.h permet d'effectuer les requêtes HTTP

```
#include <WiFi.h>  
#include <PubSubClient.h>  
#include <ESPAsyncWebServer.h>  
#include <AsyncTCP.h>  
#include <HTTPClient.h>
```

L'objet AsyncWebServer sera utilisé pour configurer le serveur Web ESP32. Nous passerons le port HTTP par défaut qui est 80, comme entrée au constructeur. Ce

sera le port sur lequel le serveur écoutera les requêtes HTTP entrantes au port numéro 80.

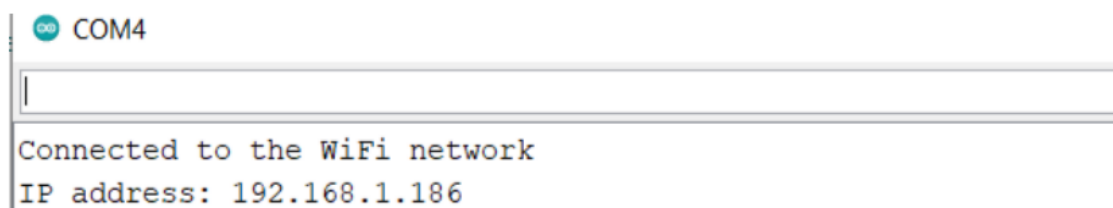
```
WiFiClient wifiClient;  
PubSubClient mqttClient(wifiClient);  
// Create AsyncWebServer object on port 80  
AsyncWebServer server(80);
```

Ensuite, on a défini les variables globales pour chaque champ d'entrée.

```
const char* ssid = "Wokwi-GUEST";// Nom du PA  
const char* password = "" //mot de passe du réseau
```

Maintenant, on a bien vérifié que l'ESP32 se connecte avec succès avec les informations d'identification réseau enregistrées. Ceci est fait grâce à ce bout de code qui permet de générer l'adresse IP en cas de connexion.

```
void wifiConnect() {  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    Serial.print("Connecting to WiFi..");  
    delay(500);  
  }  
  Serial.println(" Connected to the WiFi network");  
  // Print ESP Local IP Address  
  Serial.println("IP Address: ");  
  Serial.println(WiFi.localIP());  
}
```



The screenshot shows a serial monitor window with a title bar that includes a green circle icon and the text "COM4". The window has a scrollable text area. The output text is as follows:

```
Connected to the WiFi network  
IP address: 192.168.1.186
```

Maintenant arrivons à la configuration du serveur MQTT mais tout d'abord expliquons le principe de ce protocole.

MQTT(Message Queuing Telemetry Transport) est un protocole de transport de données qui est extrêmement léger. Il aide à minimiser les besoins en ressources pour l'appareil IoT, et peut gérer les réseaux non fiables. MQTT fonctionne sur de grandes réseaux de petits appareils qui doivent être surveillés à partir d'un serveur principal.

4.2. Pourquoi MQTT ?

MQTT collecte des données à partir de divers appareils électroniques et prend en charge la surveillance des appareils à distance. C'est un protocole d'abonnement/publication qui s'exécute sur Transmission Control Protocol (TCP), ce qui signifie qu'il prend en charge l'échange de messages événementiels via les réseaux sans fil.

MQTT est léger et se caractérise par une taille d'en-tête de 2 octets par message. Il nécessite moins d'énergie et de ressources et est conçu pour les appareils à faible bande passante. Un broker est un logiciel qui fonctionne pour recevoir toutes les informations et messages des clients éditeurs. Il filtre les messages et envoie les informations aux clients abonnés.

MQTT adapte un broker central où il peut réduire la quantité de paquets à être envoyé sur Internet et réduit l'utilisation de la mémoire requise par les clients. Ce protocole offre le plus haut niveau de qualité de services avec le moins d'interopérabilité. Il définit trois QoS niveaux :

- 0- au plus une fois (garantie TCP uniquement),
- 1- au moins une fois (garantie MQTT avec confirmation),
- 2-exactement une fois (garantie MQTT avec poignée de main).

Une fois qu'on a implémenté la connexion Wifi, on passe à la configuration du client MQTT pour l'ESP32. Pour cette tâche, on a eu recours à la bibliothèque PubSubClient.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
```

On définit le MQTT Broker avec une adresse et un numéro de port que nous allons spécifier

```
/**Set server**/
const char* mqttServer = "broker.hivemq.com";
int port = 1883;
```

Nous allons initialiser partiellement l'instance client avec les setters de propriété :

```
WiFiClient espClient;
PubSubClient client(espClient);
// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
```

puis en utilisant la fonction `setServer`, notre client est prêt à l'emploi. Ensuite, nous testons notre connectivité. Une fois connecté, un message sera publié sur le sujet `esp32/esp32test` pour informer le client.

```
void mqttConnect() {
    //Start server
    server.begin();
    mqttClient.setServer(mqttServer, mqttPort);
    while(!mqttClient.connected()) {
        Serial.println("Connecting to MQTT...");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);
        if(mqttClient.connect(clientId.c_str())) {
            Serial.println("connected");
        }
        else {
            Serial.println("try again in 5 seconds");
            Serial.println(mqttClient.state());
            delay(5000);}}
    mqttClient.publish("esp32test", "Hello from your smart aquarium");
}
```

Maintenant, les valeurs des capteurs seront publiées sur le Broker. Le client doit tout simplement entrer le nom du Broker et le port utilisé, puis s'abonner à `esp32test`, et il obtiendra toutes les informations partagées par les capteurs de l'aquarium.

5. Consommation:

L'évaluation de la consommation d'énergie est cruciale dans la conception de systèmes IoT alimentés par batterie pour assurer une autonomie satisfaisante. Cependant divers paramètres tels que les fréquences d'échantillonnage peuvent empêcher la mesure précise de la consommation d'énergie de chaque composant.

Composants principaux :

La carte ESP32 :étant le microcontrôleur principal, il a une consommation d'énergie variable en fonction de son état (activé ou non).

Capteurs : Les capteurs, tels que le capteur de température, le capteur de pH et le capteur de niveau d'eau, consomment de l'énergie lors de la lecture des données.

Actionneurs :tel que le servomoteur pour la distribution de nourriture des poissons, consomme de l'énergie lorsqu'il est activé.

Interfaces de communication : Le Wi-Fi utilisé pour la communication MQTT peut être l'un des principaux consommateurs d'énergie, en particulier lors de la

transmission de données. La fréquence de communication peut être ajustée pour économiser de l'énergie.

6. Synthèse:

Le fonction global du système est représenté par le diagramme d'activité suivant:

