**BỘ CÔNG THƯƠNG**



**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞  
A blue and white circle with text and a book

AI-generated content may be incorrect.

**ĐỒ ÁN**

**ĐỀ TÀI: MÁY CHO CÁ ĂN TỰ ĐỘNG**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | Sinh viên thực hiện | Mã số sinh viên |
| Võ Hoàng Hải | Nguyễn Văn Dũng (NT) | 2033220685 |
|  | Giang Tuấn Kiệt | 2001221896 |
|  | Nguyễn Lương Khanh | 2033222044 |
|  | Lê Mẫn | 2001222603 |
|  | Nguyễn Ngọc Sang | 2001224156 |

TP. HỒ CHÍ MINH – 2025

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞  
A blue and white circle with text and a book

AI-generated content may be incorrect.

**ĐỒ ÁN**

**ĐỀ TÀI: CẢM BIẾN CHUYỂN ĐỘNG PIR**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | Sinh viên thực hiện | Mã số sinh viên |
| Võ Hoàng Hải | Nguyễn Văn Dũng (NT) | 2033220685 |
|  | Giang Tuấn Kiệt | 2001221896 |
|  | Nguyễn Lương Khanh | 2033222044 |
|  | Lê Mẫn | 2001222603 |
|  | Nguyễn Ngọc Sang | 2001224156 |

TP. HỒ CHÍ MINH – 2025

Mục lục

[LỜI MỞ ĐẦU 4](#_Toc212288056)

[1. Giới thiệu đề tài 5](#_Toc212288057)

[2. Mục tiêu đề tài 5](#_Toc212288058)

[3. Nội dung thực hiện 5](#_Toc212288059)

[3.1. Xác định bài toán 5](#_Toc212288060)

[3.2. Thiết kế hệ thống 5](#_Toc212288061)

[3.3. Xây dựng hệ thống 6](#_Toc212288062)

[3.4. Mô phỏng & kiểm thử 6](#_Toc212288063)

[4. Kế hoạch thực hiện 6](#_Toc212288064)

[5. Kết quả đạt được 7](#_Toc212288065)

[6. Kết luận & hướng phát triển 7](#_Toc212288066)

## LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0 hiện nay, việc ứng dụng Internet of Things (IoT) vào các hoạt động sản xuất và đời sống hằng ngày đang ngày càng trở nên phổ biến. Các hệ thống IoT giúp con người có thể giám sát, điều khiển và tự động hóa các quy trình một cách thông minh, tiết kiệm thời gian và công sức.

Trong lĩnh vực nuôi trồng thủy sản, đặc biệt là nuôi cá cảnh, cá thương phẩm hoặc cá trong hồ, việc cho cá ăn đúng thời điểm, đủ lượng là yếu tố rất quan trọng, ảnh hưởng trực tiếp đến sự phát triển của cá và chất lượng nguồn nước. Tuy nhiên, công việc này thường được thực hiện thủ công, tốn thời gian và dễ bị quên hoặc cho ăn không đều, dẫn đến lãng phí thức ăn hoặc ô nhiễm môi trường nước.

Vì vậy, nhóm em đã lựa chọn đề tài “Máy cho cá ăn tự động IoT”, với mong muốn tạo ra một mô hình tự động, thông minh và dễ sử dụng, giúp người nuôi cá có thể điều khiển và theo dõi quá trình cho ăn từ xa qua Internet, đồng thời giám sát các thông số môi trường như nhiệt độ, độ ẩm, pH của nước để đảm bảo điều kiện sống tối ưu cho cá.

## 1. Giới thiệu đề tài

Hệ thống Máy cho cá ăn tự động IoT được xây dựng dựa trên vi điều khiển ESP32, kết hợp với servo motor để điều khiển nắp thức ăn, cảm biến DHT11 để đo nhiệt độ – độ ẩm môi trường, và cảm biến pH để theo dõi chất lượng nước. Ngoài ra, hệ thống được kết nối với nền tảng Adafruit IO, cho phép người dùng điều khiển cho ăn thủ công hoặc tự động từ xa thông qua mạng Wi-Fi.

Hệ thống còn được tích hợp nút bấm vật lý để cho cá ăn trong trường hợp không có Internet, đảm bảo tính linh hoạt và tiện dụng.

Dữ liệu đo từ các cảm biến sẽ được gửi lên dashboard trực tuyến, giúp người dùng có thể giám sát hồ cá mọi lúc, mọi nơi.

Đề tài không chỉ mang tính ứng dụng cao trong thực tế mà còn là ví dụ điển hình cho việc tích hợp giữa phần cứng (hardware) và phần mềm điều khiển IoT (software) trong môi trường học tập, nghiên cứu.

## 2. Mục tiêu đề tài

- Xây dựng một hệ thống IoT tự động cho cá ăn, có thể điều khiển từ xa qua Internet (Adafruit IO).

- Thu thập dữ liệu từ các cảm biến nhiệt độ, độ ẩm và pH nước, hiển thị realtime trên dashboard.

- Tích hợp servo và LED để điều khiển cơ khí và báo trạng thái.

- Thiết kế, lập trình và mô phỏng toàn bộ trên nền tảng Wokwi.

- Tối ưu giao diện điều khiển trên Adafruit IO.

## 3. Nội dung thực hiện

3.1. Xác định bài toán

**-** Vấn đề: Người nuôi cá thường quên cho ăn đúng giờ hoặc cho ăn quá nhiều, gây ô nhiễm nước.  
- Giải pháp: Xây dựng hệ thống tự động cho cá ăn theo lịch định sẵn, đồng thời có thể điều khiển từ xa qua Internet, giám sát môi trường nước (nhiệt độ – độ ẩm pH).

3.2. Thiết kế hệ thống

Phần cứng:

- ESP32 DevKit V1

- Cảm biến DHT11 (nhiệt độ, độ ẩm)

- Cảm biến pH (analog)

- Servo SG90 (mở cửa thức ăn)

- LED báo trạng thái

Phần mềm:

- Arduino IDE (lập trình ESP32)

- Adafruit IO (Dashboard Cloud)

- Wokwi (mô phỏng phần cứng online)

3.3. Xây dựng hệ thống  
- Lập trình ESP32 để đọc cảm biến DHT11 và cảm biến pH.

- Gửi dữ liệu lên Adafruit IO qua WiFi.

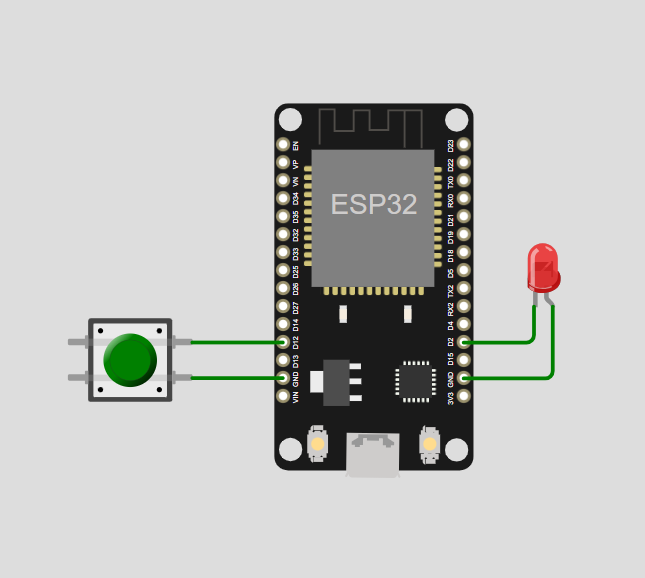
- Nhận lệnh từ feed “fish\_feeder” để điều khiển servo mở cửa thức ăn.

- LED sáng khi servo hoạt động.

- Tạo Dashboard trên Adafruit IO gồm các feed:

* fish\_feeder – điều khiển cho ăn
* temperature – hiển thị nhiệt độ
* humidity – hiển thị độ ẩm
* ph\_value – hiển thị độ pH

1. Sơ đồ thiết kế phần cứng



Hình 3.1 – Mô hình kết nối phần cứng mô phỏng trên Wokwi.

1. Thư viện sử dụng

1. ESP32Servo.h

* Chức năng: Điều khiển **động cơ servo SG90** (mở – đóng nắp thức ăn).
* Thư viện này giúp lập trình viên dễ dàng tạo xung PWM chính xác cho servo, tương thích với vi điều khiển **ESP32**.

2. AdafruitIO\_WiFi.h

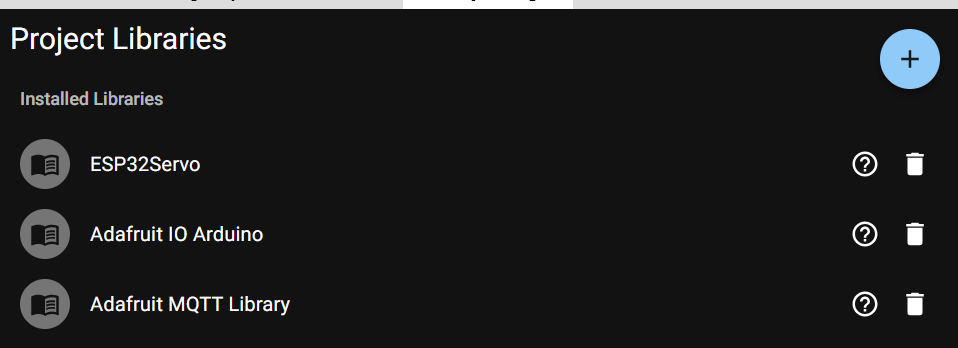
* Chức năng: Kết nối ESP32 với nền tảng Adafruit IO thông qua mạng Wi-Fi, cho phép gửi và nhận dữ liệu IoT (feed control, cảm biến).
* Sử dụng giao thức MQTT và REST API của Adafruit IO.

3. WiFi.h (thư viện mặc định của ESP32)

* Cung cấp các hàm khởi tạo và quản lý kết nối Wi-Fi (SSID, password, trạng thái mạng).
* Là thành phần cơ bản để module ESP32 kết nối Internet.

4. DHT.h (nếu có dùng cảm biến nhiệt độ/độ ẩm)

* Hỗ trợ đọc dữ liệu từ cảm biến DHT11 hoặc DHT22.

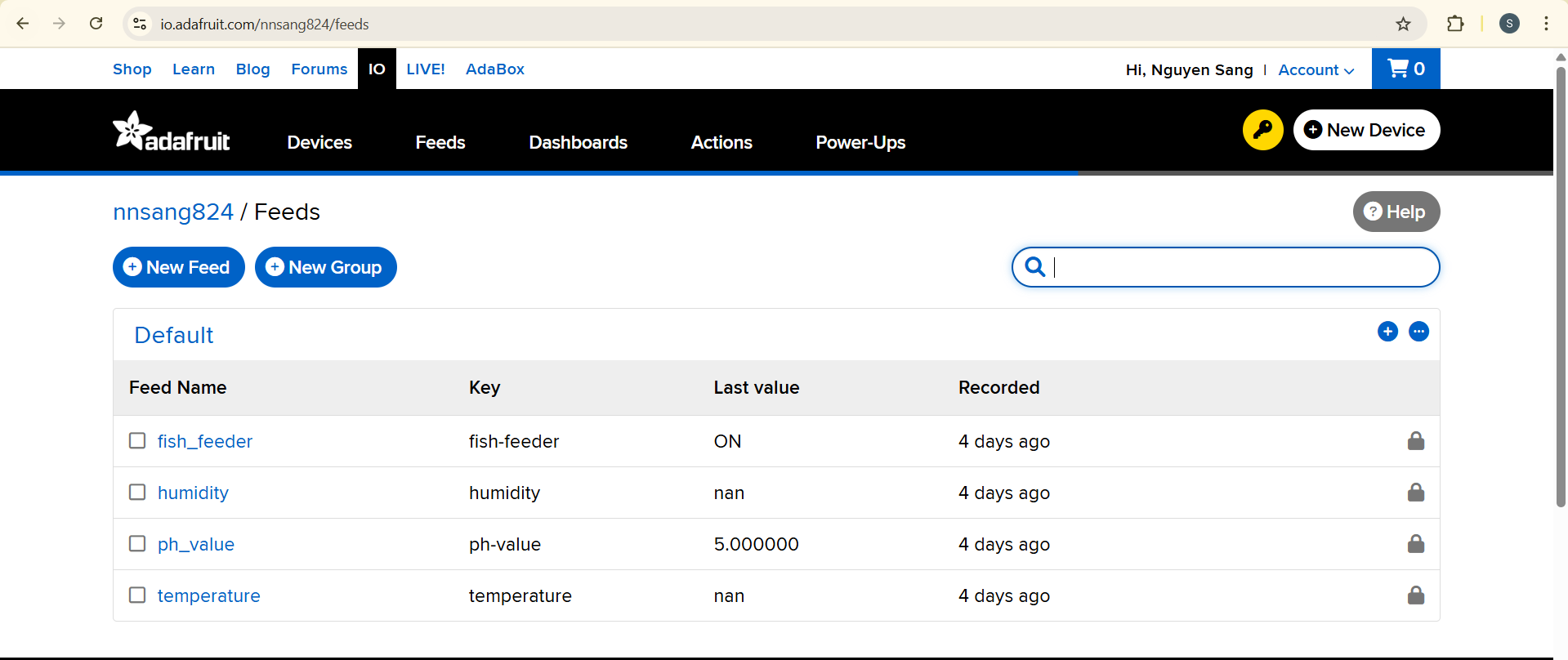


Hình 3.2 – Danh sách các thư viện được cài đặt và sử dụng trong

1. Cấu hình các Feed trên Adafruit IO

Danh sách các feed sử dụng trong dự án:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên feed** | **Chức năng** | **Kiểu dữ liệu** | **Ghi chú** |
| **fish\_feeder** | Điều khiển cho cá ăn (ON/OFF) | Boolean | Kích hoạt servo quay và LED sáng khi cho ăn |
| **temperature** | Hiển thị nhiệt độ môi trường | Float | Đọc từ cảm biến DHT11 |
| **humidity** | Hiển thị độ ẩm môi trường | Float | Đọc từ cảm biến DHT11 |
| **ph\_value** | Theo dõi độ pH nước | Float | Giả lập cảm biến pH analog |



Hình 3.3 – Giao diện dashboard điều khiển máy cho cá ăn tự động trênAdafruit IO

1. Code chương trình chính

#include <WiFi.h>

#include "AdafruitIO\_WiFi.h"

#include <ESP32Servo.h>

#include "DHT.h"

// ====== Cấu hình Adafruit IO ======

#define IO\_USERNAME  "nnsang824"

#define IO\_KEY       "aio\_YepX15sUNZjHmRhim60G9GTcOkj9"

// ====== WiFi ======

#define WIFI\_SSID "Wokwi-GUEST"

#define WIFI\_PASS ""

// ====== Khởi tạo Adafruit IO ======

AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, WIFI\_SSID, WIFI\_PASS);

// ====== Feed ======

AdafruitIO\_Feed \*feeder = io.feed("fish\_feeder");

AdafruitIO\_Feed \*tempFeed = io.feed("temperature");

AdafruitIO\_Feed \*humFeed  = io.feed("humidity");

AdafruitIO\_Feed \*phFeed   = io.feed("ph\_value");

// ====== Servo + LED ======

Servo servo;

const int servoPin = 13;

const int ledPin = 2;

// ====== Nút bấm thủ công ======

const int buttonPin = 12;S

// ====== DHT11 ======

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// ====== Cảm biến pH (giả lập analog) ======

const int phPin = 34;

// ====== Thời gian cho ăn tự động ======

unsigned long lastFeedTime = 0;

const unsigned long feedInterval = 60 \* 1000; // 1 phút/lần

// ====== Gửi dữ liệu cảm biến mỗi 15s ======

unsigned long lastSensorTime = 0;

const unsigned long sensorInterval = 15 \* 1000;

void setup() {

**Serial**.begin(115200);

  servo.attach(servoPin);

  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT\_PULLUP); // nhấn = LOW

  dht.begin();

**Serial**.println("🔌 Kết nối tới Adafruit IO...");

  io.connect();

  // Khi Adafruit IO gửi lệnh

  feeder->onMessage(handleMessage);

  while (io.status() < AIO\_CONNECTED) {

**Serial**.print(".");

    delay(1000);

  }

**Serial**.println("\n✅ Đã kết nối Adafruit IO!");

  feeder->get();

}

void loop() {

  io.run(); // duy trì kết nối IoT

  unsigned long currentTime = millis();

  // ====== Tự động cho ăn ======

  if (currentTime - lastFeedTime >= feedInterval) {

    feedFish();

    lastFeedTime = currentTime;

  }

  // ====== Cho ăn thủ công ======

  if (digitalRead(buttonPin) == LOW) {

**Serial**.println("🔘 Nút thủ công được nhấn!");

    feedFish();

    delay(1000); // tránh lặp

  }

  // ====== Gửi dữ liệu cảm biến ======

  if (currentTime - lastSensorTime >= sensorInterval) {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    int phAnalog = analogRead(phPin);

    float phValue = map(phAnalog, 0, 4095, 0, 14);

**Serial**.printf("🌡️ %.2f°C | 💧 %.2f%% | 🧪 pH: %.2f\n", t, h, phValue);

    tempFeed->save(t);

    humFeed->save(h);

    phFeed->save(phValue);

    lastSensorTime = currentTime;

  }

}

// ====== Khi có lệnh từ Adafruit IO ======

void handleMessage(AdafruitIO\_Data \*data) {

  String cmd = data->toString();

**Serial**.print("📡 Lệnh từ Adafruit IO: ");

**Serial**.println(cmd);

  if (cmd == "ON" || cmd == "1") {

    feedFish();

  }

}

// ====== Hàm cho cá ăn ======

void feedFish() {

**Serial**.println("🐟 Đang cho cá ăn...");

  digitalWrite(ledPin, HIGH);

  servo.write(90);

  delay(2000);

  servo.write(0);

  digitalWrite(ledPin, LOW);

**Serial**.println("✅ Đã cho ăn xong!\n");

  delay(500);

}

3.4. Mô phỏng & kiểm thử

**-** Mô phỏng toàn bộ hệ thống trên nền tảng Wokwi để kiểm tra hoạt động của servo, cảm biến và LED.

- Kết nối thực tế Adafruit IO để quan sát dữ liệu và điều khiển từ xa qua Internet.

## 4. Kế hoạch thực hiện

|  |  |  |
| --- | --- | --- |
| Tuần | Công việc | Thời gian (giờ) |
| Tuần 1 | Nghiên cứu, thiết kế sơ đồ, lập trình cơ bản | 6h |
| Tuần 1 | Mô phỏng servo + LED + cảm biến DHT11 trên Wokwi | 4h |
| Tuần 2 | Kết nối Adafruit IO, gửi/nhận dữ liệu | 5h |
| Tuần 2 | Hoàn thiện, thử nghiệm và viết báo cáo | 5h |

## 5. Kết quả đạt được

- Hệ thống máy cho cá ăn tự động hoạt động đúng chức năng.

- Có thể điều khiển từ xa qua Adafruit IO và WiFi.

- Dữ liệu cảm biến hiển thị realtime trên dashboard.

- Mô hình được mô phỏng ổn định trên Wokwi.

## 6. Kết luận & hướng phát triển

- Kết luận: Đề tài chứng minh tính khả thi của việc ứng dụng IoT vào nuôi trồng thủy sản quy mô nhỏ, giúp tiết kiệm thời gian và công sức cho người nuôi.

- Hướng phát triển:

* Tích hợp camera ESP32-CAM để theo dõi hồ cá.
* Thêm chức năng tự động đo lượng thức ăn dựa theo số lượng cá.
* Dùng năng lượng mặt trời và pin dự phòng để hoạt động liên tục.

📘 Tóm tắt: Hệ thống máy cho cá ăn IoT sử dụng ESP32, cảm biến DHT11 và cảm biến pH, có khả năng điều khiển từ xa qua Adafruit IO, hiển thị dữ liệu môi trường hồ cá và trạng thái hoạt động, mang lại giải pháp tiện lợi cho người nuôi cá.

* Link demo: