

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS (CO3037)

**ĐỀ TÀI: ỨNG DỤNG IOT TRONG VIỆC QUẢN LÝ VÀ CHĂM SÓC BỂ CÁ
THỦY SINH**

Lớp L02 - Nhóm 9 - HK221

Giảng viên hướng dẫn: Huỳnh Hoàng Kha

STT	Sinh viên thực hiện	Mã số sinh viên	Đánh giá
1	Phạm Duy Quang	2011899	100%
2	Viên Minh Phúc	2011865	100%
3	Ứng Văn Hưng	2013401	100%

Thành phố Hồ Chí Minh, tháng 01 năm 2023

PHÂN CHIA CÔNG VIỆC

STT	Sinh viên thực hiện	Khối lượng công việc
1	Phạm Duy Quang	Hỗ trợ xử lý code hiện thực ra thiết bị phần cứng và IOT Server; viết, hoàn thiện và tổng hợp báo cáo
2	Viên Minh Phúc	Tìm hiểu, xử lý code hiện thực ra thiết bị phần cứng và IOT Server; viết và chỉnh sửa báo cáo
3	Ứng Văn Hưng	Tìm hiểu và tìm mua các thiết bị phần cứng phù hợp cho việc demo; viết, chỉnh sửa và hoàn thiện báo cáo

MỤC LỤC

1. Tổng quan về đề tài thực hiện	3
1.1. Giới thiệu về đề tài	3
1.2 Tính thực tế của đề tài	4
1.3. Giải pháp đặt ra	5
1.4. Các chức năng của sản phẩm	5
2. Quá trình thiết kế và hiện thực	7
2.1. Quá trình thiết kế	7
2.1.1. Giai đoạn lên ý tưởng	7
2.1.2. Lựa chọn thiết bị	9
2.1.2.1. Lựa chọn cảm biến	9
a) Cảm biến đo độ pH và nhiệt độ SEN0106	9
b) Cảm biến đo độ Oxy hòa tan SEN0237	10
2.1.2.2. Lựa chọn Gateway - Vi điều khiển ESP32	11
2.1.2.3. Lựa chọn Actuators	12
a) Bộ cho cá ăn có tích hợp wifi	12
b) Máy sục khí Oxy cho bể cá SOBO SB-4000	13
c) Thiết bị sưởi bể cá DTC-2201	15
2.1.3. Xây dựng web server	16
2.1.4. Các vấn đề phát sinh	16
2.2. Quá trình hiện thực	17
2.2.1. Hiện thực phần cứng	17
2.2.2. Hiện thực chương trình (có đính kèm link source code)	18
3. Quá trình mô phỏng (có đính kèm link video demo)	23
4. Tổng quan và đánh giá	29
4.1. Tổng kết về sản phẩm	29

4.2. Những khó khăn gặp phải và hướng giải quyết	29
<i>4.2.1. Khó khăn gặp phải</i>	29
<i>4.2.2. Hướng giải quyết</i>	30
4.3. Hướng phát triển trong tương lai	30
5. Tài liệu tham khảo	31

1. Tổng quan về đề tài thực hiện

Với tiêu chí dự án IoT phải đem lại sự tiện lợi, cải tiến từ những điều kiện đã có sẵn, nhóm hướng đến những dự án quy mô vừa và nhỏ, có tính thực tế cao và giải quyết được vấn đề liên quan tới sự tiện lợi, giải quyết được vấn đề đó từ xa (Do đa số người ở các thành phố lớn đều bận rộn với công việc). Và khi tìm hiểu, nhóm nhận thấy việc chăm sóc thú cưng là một nhu cầu có tính thiết thực, do mọi người không thể ở bên cạnh chúng 24/7, do vậy họ rất cần một công cụ để theo dõi, giám sát chúng từ xa. Nhưng với các loài thú cưng có thể chạy, nhảy (như chó hay mèo) sẽ rất khó khăn khi xây dựng một hệ thống như trên do phạm vi bao quát của hệ thống phải rộng, cũng như không phải ai cũng chịu nhốt thú cưng của mình trong chuồng suốt ngày.

Từ đó, nhóm tìm đến những loài có thể quản lý được trong một không gian nhỏ và dễ chăm sóc, phổ biến → Cá cảnh là một lựa chọn tốt, do đó, nhóm triển khai dự án theo hướng chăm sóc và giám sát cho loài này.

1.1. Giới thiệu về đề tài



- Theo nhịp sống ngày càng phát triển, chất lượng và điều kiện sống của con người cũng theo đó mà tăng lên, nuôi thú cưng cũng dần trở thành nhu cầu tất yếu.

- Việc sở hữu một chiếc bể cá nhỏ xinh, ngắm nhìn những chú cá bơi lội tung tăng là một cách giải tỏa căng thẳng vô cùng hiệu quả sau những giờ làm việc mệt mỏi, giúp cho không gian nhà và nơi làm việc đẹp hơn, góp phần tạo cuộc sống vui vẻ hơn.

- Vậy chúng ta đã biết cách nào để quan tâm đến chúng từ xa hay là những giải pháp hỗ trợ cho những người mới tập nuôi hay không?

→ Nhóm 9 chúng em xin đề xuất giải pháp: Sử dụng ứng dụng IoT vào mô hình nuôi cá trong bể để tạo thành bể cá thông minh.

1.2 Tính thực tế của đề tài

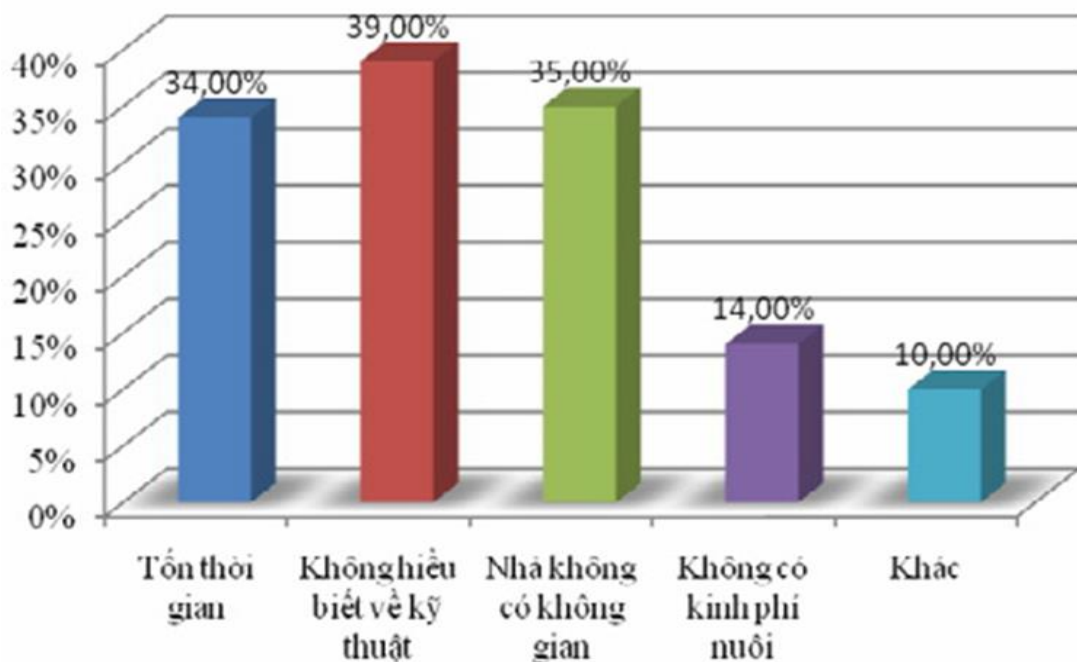
- Con người ngày càng có xu hướng chuyển từ sinh sống ở nhà mặt đất dần qua chung cư, căn hộ cao cấp, smart-home,...

- Việc nuôi cá cảnh lại dễ dàng đáp ứng những điều kiện: nhỏ gọn, dễ di chuyển cũng như phù hợp với nhu cầu nhà ở.



- Hiện nay trên địa bàn thành phố Hồ Chí Minh có khoảng 286 cơ sở hoạt động về nuôi và sản xuất cá cảnh, khu vực sản xuất tập trung nhiều nhất ở Quận 8, Quận 12 và rải rác một số quận ven khác như Quận 9, Gò Vấp, Bình Thạnh và huyện Bình Chánh. Số lượng sản xuất và tiêu thụ cá cảnh tại thành phố đạt khoảng 15 - 17 triệu con.

- Việc nuôi cá cảnh giải trí làm mất thời gian cũng như đòi hỏi kỹ thuật chăm sóc nhiều hơn là nuôi chó, mèo hay động vật cảnh khác. Theo kết quả phỏng vấn trực tiếp những người dân TP.HCM về nguyên nhân chưa nuôi cá cảnh, có 39% do không hiểu biết kỹ thuật nuôi, 35% cho rằng diện tích nhà không đủ rộng để nuôi cá, 34% cho là mất nhiều thời gian, 14% không có kinh phí để nuôi, và có 10% là do các nguyên nhân khác như họ đã thích loài thú nuôi khác (chó, mèo, chim, thỏ...), họ không muốn nuôi vì sợ làm cá chết.



- Bên cạnh đó, kết quả phỏng vấn cũng cho thấy rằng khi có điều kiện nuôi thú cưng trong nhà thì chó mèo là vật nuôi ưu tiên chiếm 43%, kế đến là cá cảnh 42%, chim cảnh 11% và động vật khác như thỏ, bọ, rùa... chiếm 3% số người được hỏi.

- Qua đó cho thấy thú vui nuôi cá cảnh được mọi người rất quan tâm, khi có một điều kiện nào thuận lợi sẽ tạo kích thích sự đam mê làm cho họ muốn và thích nuôi cá cảnh.

→ Đề tài ứng dụng IoT trong việc quản lý và giám sát bể cá thủy sinh của nhóm chúng em là đề tài có tính khả thi cao do giải quyết được nhiều vấn đề khiến cho người dân ngần ngại chưa muốn nuôi bể cá cảnh.

Nguồn số liệu tham khảo:

https://fof.hcmuaf.edu.vn/data/file/51_%20DTQ%20Ngan%20%20NM%20Duc-DHNL-Hien%20trang%20nuoi%20ca%20canh.pdf.

1.3. Giải pháp đặt ra

Từ những nhu cầu thực tế đó chúng em đã đưa ra biện pháp sử dụng IoT trong việc quản lý và chăm sóc bể cá cảnh bằng cách sử dụng dữ liệu nhận được từ các cảm biến, sau đó dùng vi điều khiển làm gateway để đẩy dữ liệu lên Server, từ những số liệu nhận được đó Server sẽ xử lý và gửi lại yêu cầu thực hiện các chức năng đến Gateway và các Actuators sẽ thực hiện đầu ra tương ứng.

1.4. Các chức năng của sản phẩm

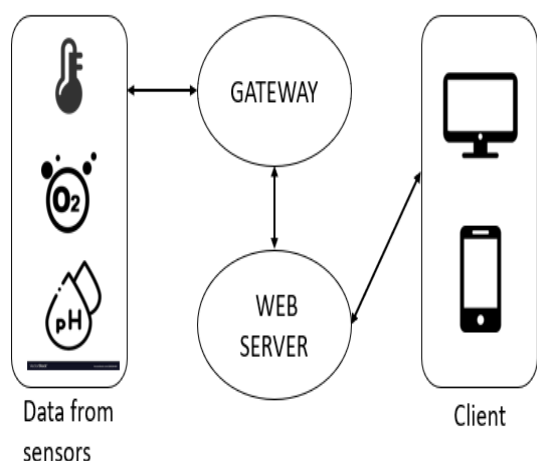
- Giảm bớt đáng kể thời gian quan tâm cho bể cá cảnh, giải quyết vấn đề chăm lo cho cá từ xa (cho những người bận bịu với công việc...).

- Hỗ trợ những người mới trong việc nuôi và chăm sóc một bể cá cảnh.

Đầu vào: Dữ liệu được đo từ các cảm biến sau đó được đẩy lên server:

+ Dữ liệu nhận được từ cảm biến đo oxy hòa tan.

+ Dữ liệu nhận được từ cảm biến đo độ PH.



+ Dữ liệu nhận được từ cảm biến đo nhiệt độ.

Đầu ra: Server dựa trên dữ liệu được đưa lên sẽ ra thông báo đề xuất đến người dùng:

+ Thông báo cho người dùng đã đến khoảng thời gian cho cá ăn.

+ Thông báo cho người dùng nếu nước trong bể có nồng độ Oxy thấp hơn mức sống của cá, hoặc độ pH quá cao/quá thấp.

+ Thông báo nếu nhiệt độ nước quá cao/quá thấp.

→ Nếu sau 1 phút người dùng không truy cập server thực hiện chức năng thì hệ thống sẽ tự động cho cá ăn, thay nước, sục khí oxy, bật đèn... cho bể cá.

2. Quá trình thiết kế và hiện thực

2.1. Quá trình thiết kế

2.1.1. Giai đoạn lên ý tưởng

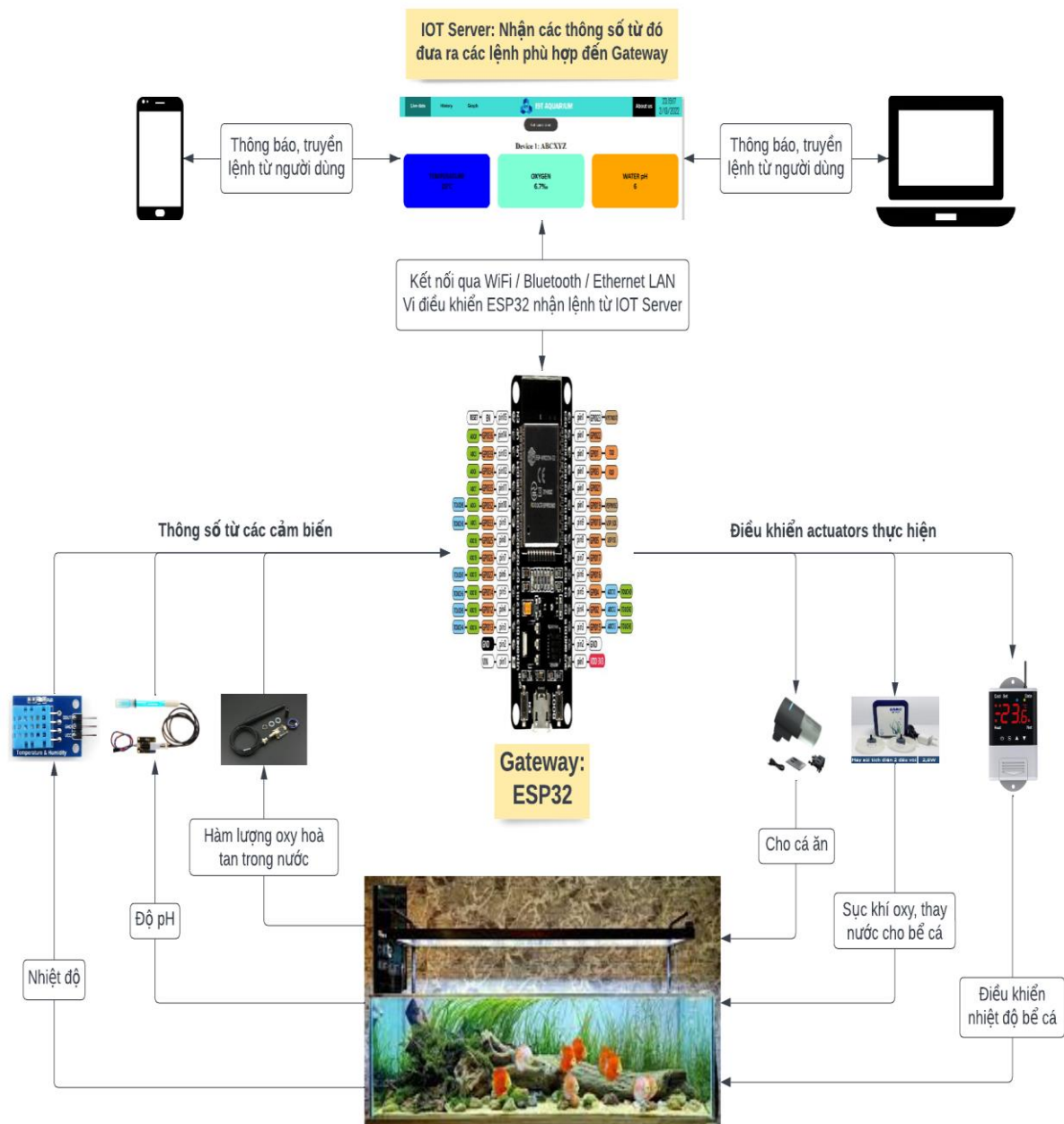
- Các thông số phù hợp với mức sống của cá: Đối với 1 bể cá gia đình, nhóm quan tâm đến 3 yếu tố: Nhiệt độ nước, Lượng oxy hòa tan, pH của nước. => Tìm kiếm các cảm biến đo được những thông số trên và phù hợp với môi trường nước, cũng như quy mô của bể cá trong các hộ gia đình hoặc cửa hàng cá cảnh.

- Về phần cứng: Nhóm lựa chọn vi điều khiển ESP32 do giá thành rẻ, có nhiều thư viện hỗ trợ cũng như có đầy đủ các chức năng cần thiết cho việc hiện thực hệ thống. Các actuator cũng được lựa chọn dựa trên việc tương thích với cảm biến, vi điều khiển cũng như là phù hợp để hoạt động khi các thông số cảm biến vượt/thấp hơn mức cho phép.

- Về việc lập trình: Ở mảng lập trình vi điều khiển, Arduino IDE là lựa chọn hàng đầu do đã có sẵn rất nhiều công cụ hỗ trợ cho việc lập trình ESP32, ngoài ra độ phổ biến của IDE này cũng giúp nhóm dễ tìm ra giải pháp khắc phục nhiều khó khăn gặp phải với hệ thống trong tương lai (nếu có phát sinh) từ documentation và các forum hỗ trợ trên Internet.

- Sau khi thảo luận, nhóm trước tiên đi đến làm Web server đồng thời cũng là giao diện cho người dùng theo dõi và điều khiển bể cá từ xa.

- Ý tưởng kết nối được nhóm thể hiện qua sơ đồ như sau:



- Sau buổi thuyết trình ý tưởng đề tài Bài tập lớn của nhóm cũng như sự tư vấn của giảng viên hướng dẫn, nhóm quyết định hiện thực thêm gateway điều khiển các Actuators bên cạnh phần server.

2.1.2. Lựa chọn thiết bị

2.1.2.1. Lựa chọn cảm biến

a) Cảm biến đo độ pH và nhiệt độ SEN0106

Bảng đầu nối tín hiệu có đèn LED hoạt động như đèn báo nguồn. Bo mạch cũng có đầu nối loại BNC và giao diện cảm biến pH. Để sử dụng, chỉ cần kết nối cảm biến pH với đầu nối BNC, sau đó cắm giao diện pH vào cổng đầu vào tương tự của bất kỳ bộ vi điều khiển nào. Nếu được lập trình, ta sẽ nhận được giá trị pH dễ dàng. Ta cũng có thể sử dụng đồng hồ vạn năng để đo điện áp đầu ra.



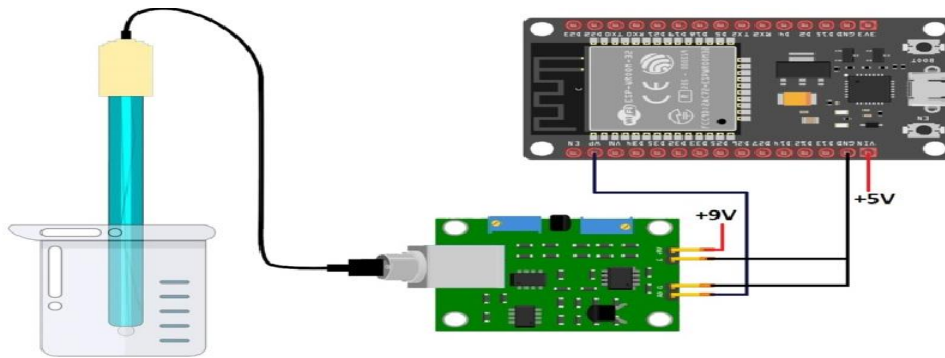
Các thông số kỹ thuật chính:

- Nguồn: 5.00V
- Kích thước: 43mm×32mm
- Khoảng đo độ pH: 0 - 14 pH
- Khoảng đo nhiệt độ: 0 - 60°C
- Độ chính xác: $\pm 0.1\text{pH}$ (25°C)
- Thời gian phản hồi: $\leq 1\text{min}$
- Cảm biến pH với đầu nối BNC
- Đầu nối tín hiệu: Giao diện gravity analog (PH 2.0 - 3P)
- Chiều dài cáp từ cảm biến đến đầu nối BNC: 660 mm

Lý do lựa chọn cảm biến: Với nguồn điện đầu vào nhận là 5.00V bằng với nguồn điện GateWay sử dụng giúp ta không phải sử dụng máy biến áp trung gian, với khoảng đo PH rộng từ 0-14 và độ chính xác cao chỉ $\pm 0.1\text{pH}$ giúp dữ liệu nhận được có độ chính xác cao.

Ngoài ra với thời gian phản hồi ngắn (< 1 phút) ta có thể nhận được dữ liệu một cách liên tục và đầu tín hiệu gravity analog 2.0 ta có thể sử dụng cảm biến một cách dễ dàng chỉ bằng cách cắm vào dung dịch hoặc nước.

Minh họa cho sự kết nối giữa sensor và Gateway ESP32:



b) Cảm biến đo độ Oxy hòa tan SEN0237

Module cảm biến oxy hòa tan đầu ra Analog SEN0237 này được sử dụng để đo mức oxy hòa tan trong nước và phản ánh chất lượng nước. Cảm biến này giúp chúng ta nhanh chóng phát hiện ra lượng oxy hòa tan trong nước thông qua đầu dò. Đầu dò mạ điện, không cần thời gian phân cực và luôn có sẵn bất cứ lúc nào.

Các thông số kỹ thuật chính:

- Đầu dò Oxy hòa tan với loại đầu dò Galvanic

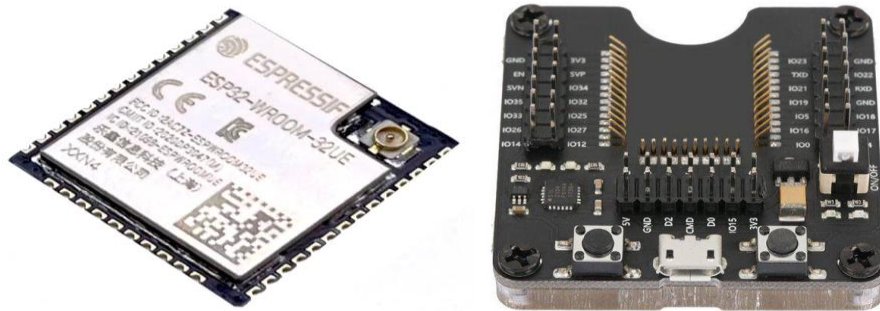
- Phạm vi phát hiện: 0 ~ 20 mg/L
- Thời gian phản hồi: Phản hồi đầy đủ lên tới 98%, trong vòng 90 giây (250 C)
- Phạm vi áp suất: 0 ~ 50 PSI
- Đầu nối đầu dò – Đầu nối cáp: BNC
- Điện áp hoạt động: 3.3 ~ 5.5V
- Tín hiệu đầu ra: 0 ~ 3.0V
- Đầu nối tín hiệu: Giao diện gravity analog (PH 2.0 - 3P)



Lý do lựa chọn cảm biến: Với tỉ lệ phản hồi lên tới 98% trong 90 giây giúp chúng ta có thể nhận được dữ liệu một cách liên tục, phạm vi đo áp suất cao lên tới tối đa 50 PSI giúp linh động trong việc đo đạc. Với điện áp hoạt động từ 3.3 - 5.5V cũng rất phù hợp với điện áp của Gateway, ngoài ra với giao diện tín hiệu gravity analog giúp chúng ta đo đạc một cách dễ dàng chỉ bằng cách cắm trực tiếp cảm biến vào nước hoặc dung dịch.

2.1.2.2. Lựa chọn Gateway - Vi điều khiển ESP32

ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth (tạm dịch: Bluetooth chế độ kép). Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.



- Model: Wifi BLE SoC ESP32 ESP-WROOM-32UE chính hãng Espressif (tương thích hoàn toàn với phiên bản cũ ESP-WROOM-32U hiện đã ngưng sản xuất).

- ESP32-D0WD-V3 embedded, Xtensa® dual-core 32-bit LX6 microprocessor, up to 240 Mhz.

- Điện áp sử dụng: 3~3.6VDC
- Dòng điện sử dụng: ~90mA
- 448 KB ROM for booting and core functions
- 520 KB SRAM for data and instructions
- 16 KB SRAM in RTC
- 4MB (XXN4) / 16MB (XXN16) SPI Flash
- Kiểu Anten: Ipex socket
- Wi-Fi:
 - + 802.11b/g/n
 - + Bit rate: 802.11n up to 150 Mbps
 - + A-MPDU and A-MSDU aggregation
 - + 0.4 μ s guard interval support

- + Center frequency range of operating channel: 2412 ~ 2484 MHz
- Bluetooth®
- Bluetooth V4.2 BR/EDR and Bluetooth LE specification
- Class-1, class-2 and class-3 transmitter
- AFH
- CVSD and SBC
- Hardware
- + Interfaces: SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®, compatible with ISO11898-1)
- + 40 MHz crystal oscillator
- Kích thước: 18 x 19.2 x 3.2mm

Lý do lựa chọn: Là một trong những vi điều khiển phổ biến nhất hiện nay với giá thành hợp lý nhưng tích hợp đầy đủ Wifi tốc độ cao và Bluetooth v4.2 và BLE giúp ta đa dạng trong cách kết nối, 34 cổng GPIO nên việc sử dụng một số lượng khá khá sensor một cách dễ dàng.

2.1.2.3. Lựa chọn Actuators

a) Bộ cho cá ăn có tích hợp wifi

- Thức ăn và thời gian cho ăn có thể điều chỉnh bằng thanh trượt (Tối đa, Tối thiểu), có thể tăng hoặc giảm lượng thức ăn được phân phát mỗi lần quay.
- Điều chỉnh cho ăn đơn giản lên đến 4 lần một ngày. Có thể vận hành cả tự động và thủ công.
- Thiết kế tinh giản với chất liệu nhựa ABS chịu lực hỗ trợ người dùng dễ dàng kiểm soát lượng thức ăn cho cá trong bể thủy sinh.



Các thông số kỹ thuật chính:

- Thương hiệu: iLONDA.
- Chất liệu: Nhựa ABS.
- Kích thước: Xấp xỉ 60 x 115 x 140mm.
- Kẹp cố định có thể kẹp độ dày: $\geq 3,5$ cm.
- Chế độ cung cấp điện: Cung cấp điện microUSB.
- Công suất tiêu thụ ở chế độ chờ: 135 mW.
- Điện áp: 5V / 2V.
- Dung tích lưu trữ: 165ml - 175ml, khoảng 140g - 160g.

Lý do lựa chọn thiết bị: Dung tích lưu trữ ở mức ổn có thể cung cấp đủ số lượng thức ăn cho cá ăn trong 1 ngày, công suất ở mức chờ nhỏ nên gần như rất ít hao phí điện áp khi không sử dụng. Với cả chế độ thủ công và tự động là một lợi thế rất lớn của thiết bị có thể giúp chủ bể cá dù không có ở nhà vẫn có thể cho cá ăn được.

b) Máy sục khí Oxy cho bể cá SOBO SB-4000

Đây là loại máy bơm thổi khí thường được sử dụng trong bể cá cảnh nói chung hoặc bể thủy sinh nói riêng, với bể cá loại nhỏ (bình, chậu) có thể không cần sử dụng. Máy sủi có tác dụng tăng cường khí oxy trong nước hoặc trộn đều nước bằng cách thổi không khí vào trong môi trường nước và tạo các dòng chảy nhân tạo đảo nước, khi đó không khí chứa oxy sẽ được hòa tan vào nước giúp động vật thủy sinh trong bể hô hấp khỏe mạnh. Ngoài ra, nó còn có tác dụng đảo đều nước, đặc biệt là dung dịch thủy canh.



Các thông số kỹ thuật chính:

- Công suất: 2.8 W
- Lưu lượng khí: 2 x 3 lít / phút
- Số lượng đầu vòi: 2
- Chế độ điều chỉnh lưu lượng khí: không có.
- Loại Máy: Máy sủi tích điện
- Nguồn điện: 220V – 240V / 50Hz
- Mức độ gây ồn: Trung bình

Đặc điểm kỹ thuật:

- Chất liệu: ABS + 5730 hạt đèn LED
- Phích cắm: EU/US

Lý do lựa chọn thiết bị: Với lưu lượng sục khí khá phù hợp so với những chiếc bể cá nhỏ mà đề tài hướng đến, cùng với công suất chỉ 2.8W ăn khá là ít điện so với các máy cùng loại là yếu tố đầu tiên mà nhóm lựa chọn thiết bị này. Bên cạnh đó việc sử dụng nguồn điện gia đình 220V giúp ta không phải tốn tiền để mua các máy biến áp mà có thể dễ dàng, độ ồn chỉ ở mức trung bình cũng là một điểm đáng chú ý vì dù sử dụng ở không gian nhỏ như chung cư thì cũng không quá ảnh hưởng đến cuộc sống thường nhật.

c) Thiết bị sưởi bể cá DTC-2201



Các thông số kỹ thuật chính:

- Nguồn điện cung cấp: AC110-230V
- Phạm vi đo: -40 – 99 °C
- Độ chính xác đo: ± 1 °C (-40 °C ~ + 70 °C)
- Công suất đầu ra: 10A ~ 250VAC
- Kích thước máy đo: 93x56x34mm
- Trọng lượng máy đo: 91g
- Chiều dài dây: 100cm
- Cảm biến NTC với độ chính xác ± 0.1 °C
- Hỗ trợ kết nối WiFi

Lý do lựa chọn: Ngoài tác dụng sưởi ấm, máy còn có khả năng đo nhiệt độ với khoảng đo nhiệt độ rất rộng không chỉ phù hợp với nhu cầu trong nước mà còn với quốc tế. độ chính xác cao chỉ ± 1 °C giúp dữ liệu nhận được một cách chính xác. Kích thước nhỏ gọn kèm kết nối Wifi là một lợi thế rất lớn ở thời điểm hiện tại so với các thiết bị chỉ tương tác thủ công đang có trên thị trường.

Ngoài ra với nguồn điện sử dụng là 220V là nguồn điện gia đình nên ta không phải tốn tiền mua các loại máy biến áp bên cạnh đó cách sử dụng, đơn giản chỉ cần cắm vào bể là có thể sưởi và điều khiển dễ dàng.

2.1.3. Xây dựng web server

Do hạn chế về quy mô và thời gian, nhóm quyết định thiết kế 1 web server với các thành phần cơ bản như:

- Dashboard: Theo dõi các thông số về nhiệt độ, lượng oxy hòa tan và độ pH từ các cảm biến.

- Hoạt động của các actuator: Theo dõi xem các actuator đang ở trạng thái nào. Có thể chủ động bật/tắt actuator bằng nút nhấn ảo trên server.

- Gửi thời gian thực về vi điều khiển.

Sau khi nghiên cứu các documentation và các tutorial, nhóm quyết định sử dụng các thư viện: WiFi, AsyncTCP và ESPAsyncWebServer để hiện thực 1 web server chạy trên localhost với các ngôn ngữ HTML, CSS và JavaScript cho phần front-end.

- Thư viện Wifi (#include “WiFi.h”): Giúp vi điều khiển ESP32 kết nối được với WiFi.

- Thư viện AsyncTCP và ESPAsyncWebServer (#include “ESPAsyncWebServer.h”): Giúp tạo ra 1 web server trên localhost. Ngoài ra nó còn giúp gửi data từ cảm biến lên server và gửi phản hồi từ server về vi điều khiển bằng các phương thức HTTP GET và HTTP POST.

2.1.4. Các vấn đề phát sinh

- Vấn đề chi phí: Để hiện thực được hệ thống, chi phí cho các cảm biến và actuator mà nhóm bỏ ra là không hề nhỏ (Lên đến cả triệu đồng). Do ở quy mô của một bài tập lớn, nhóm quyết định sẽ kết nối và điều khiển các LED tượng trưng cho các actuator và “giả” dữ liệu được sinh ra từ cảm biến một cách ngẫu nhiên, nhưng tuân thủ theo 3 mức: Bình thường, thấp và cao. Khi mô phỏng hoạt động, ta có thể điều chỉnh các mức này bằng biến trở núm vặn, sử dụng nút nhấn hoặc thay đổi bằng cửa sổ Serial monitor trong Arduino IDE (Nhóm lựa chọn cách này).

- Vấn đề khi hiện thực Web server: Do các thành viên trong nhóm chưa có nhiều kinh nghiệm và kiến thức cần có để thực hiện một dự án IoT (có thành viên chưa học Mạng máy tính, lập trình Web,...) nên nhóm đã gặp nhiều khó khăn trong việc thực hiện Web server cũng như làm sao cho dự án phù hợp với thực tế. Do đó, giao diện Web server chưa được bắt mắt.

- Vấn đề kết nối Wifi: Kết nối Wifi không thực sự ổn định ở khoảng cách xa (từ 10 - 15m trở lên). Do đó, trong lúc hiện thực nhóm thỉnh thoảng gặp trục trặc vì vi điều khiển mất kết nối với Web server.

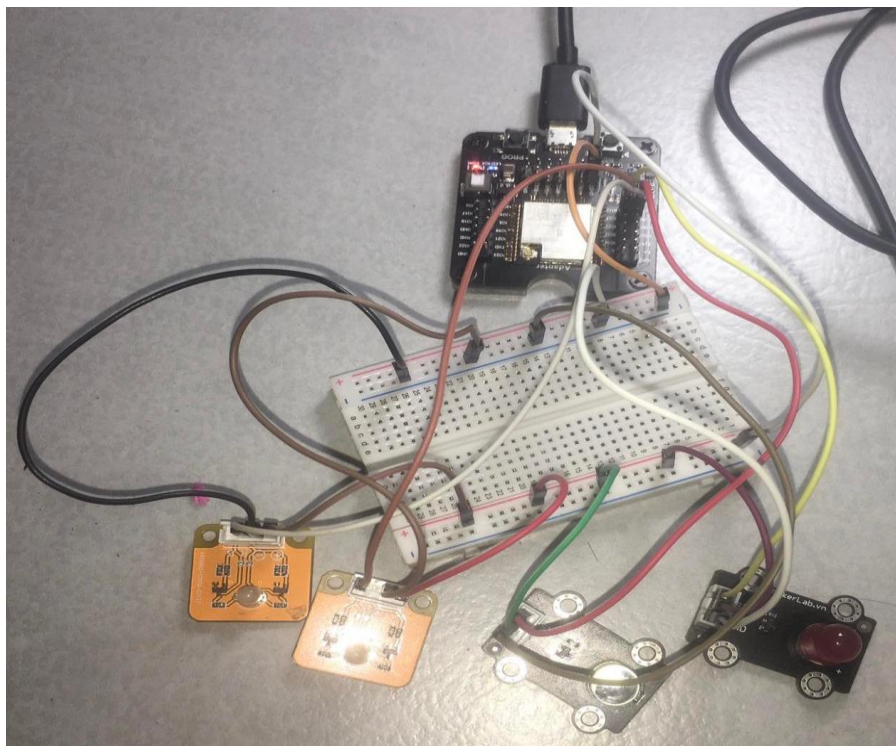
2.2. Quá trình hiện thực

2.2.1. Hiện thực phần cứng

- Bố trí cảm biến: Do các cảm biến đều có thể hoạt động trong môi trường nước, cũng như dễ dàng kết nối với vi điều khiển bằng dây cắm hoặc WiFi, cộng thêm việc kích thước của vi điều khiển là khá nhỏ. Do vậy ta có thể kết nối tất cả sensor để tạo thành 1 sensor node (Dự tính 1 sensor node cho 1 bể), sau đó kết nối chúng với gateway là ESP32.

- Tuy nhiên do bài toán chi phí đã đề cập ở trên và với quy mô của một bài tập lớn, để mô phỏng nhóm quyết định sử dụng các đèn LED để tượng trưng cho các Actuators và sử dụng các hàm sinh ra số ngẫu nhiên trong ngôn ngữ C để “giả” thông số cho các cảm biến.

- Nhóm sử dụng breadboard để mở rộng kết nối cho các Actuators tượng trưng bằng các đèn LED. Sau đó, nhóm kết nối vi điều khiển ESP-WROOM-32 qua cổng USB đến máy tính để lập trình trên Arduino IDE. Do bản thân các actuator đều đã có công tắc hoặc dây điện để bật/tắt, do đó nút nhấn vật lý không thật sự cần thiết trong trường hợp này (Trong video demo có dùng 1 nút nhấn chỉ để thử kết nối cho các module LED).



Ở hình trên, từ trái sang phải: Hệ thống sưởi (IO26), máy bơm nước (IO27), máy sục khí (IO14), bộ cho cá ăn (IO12).

2.2.2. Hiện thực chương trình (có đính kèm link source code)

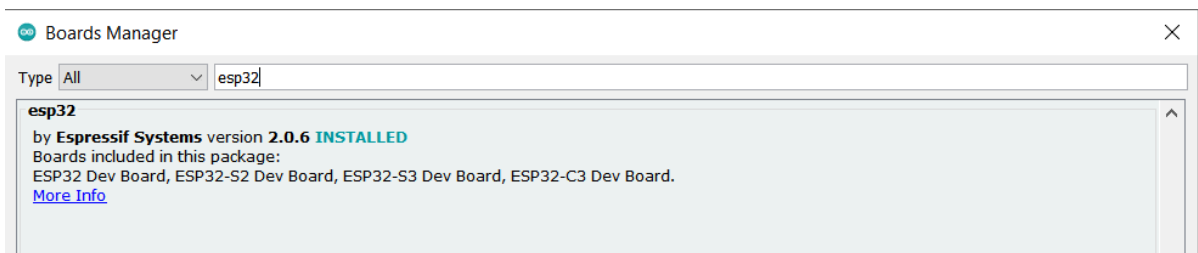
Cài đặt board ESP32: Chọn **Files >> Preferences**. Sau đó copy đường link sau và dán vào mục **Additional Boards Manager URLs**:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Additional Boards Manager URLs:

Nhấn OK để hoàn thành. Sau đó ta có thể vào mục **Tools >> Board >> Boards Manager >> Gõ “esp32”** trên thanh tìm kiếm và tiến hành cài đặt.

Sau khi cài đặt ta sẽ nhận được kết quả:



Để phục vụ cho việc lập trình, ta cần cài đặt thêm các thư viện cần thiết từ bên ngoài (Do không có sẵn trong IDE):

- ESPAsyncWebServer: <https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip>.

- AsyncTCP: <https://github.com/me-no-dev/AsyncTCP/archive/master.zip>.

Lưu ý: Sau khi tải về, người dùng cần đổi tên 2 file .zip trên bằng cách xóa đi phần **-master**. Sau đó ta có thể cài đặt bằng cách: Chọn **Sketch >> Include library >> Add .ZIP library**. Chọn 2 file .zip vừa đổi tên là ta đã cài đặt xong.

Một số phần code của server:

- Include các thư viện:

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
```

```
#include "time.h"
#include "stdlib.h"

//Tên và mật khẩu của mạng WiFi mà người dùng muốn kết nối
const char* ssid = "sample_name";
const char* password = "sample_password";

// Tạo server ở port 80 trên localhost
AsyncWebServer server(80);
```

- Khởi tạo server (Code đặt trong hàm void setup()):

```
//Kết nối đến Serial Monitor để thuận tiện cho việc debug
Serial.begin(115200); //Baud rate = 115200
//Kết nối vào WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
}
//Khi đã kết nối thành công, hiển thị địa chỉ IP để người dùng
truy cập
Serial.println(WiFi.localIP());
```

- Gửi dữ liệu cảm biến lên server (Code đặt trong hàm void setup()): Sử dụng phương thức HTTP GET để gửi request và nhận response.

```
//Gửi đoạn code HTML lên để hiển thị giao diện cho server
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
});
//Gửi các request để hiển thị 3 thông số cần thiết
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest
*request){
    request->send_P(200, "text/plain", readTemperature().c_str());
```

```

});
server.on("/oxygen", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readOxygen().c_str());
});
server.on("/ph", HTTP_GET, [] (AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readPH().c_str());
});
//Các request được gửi lên server để hiển thị thời gian hoạt động của
bộ cho cá ăn
    server.on("/sethour", HTTP_GET, [] (AsyncWebServerRequest *request){
        request->send_P(200, "text/plain", String(set_hour).c_str());
    });
    server.on("/setminute", HTTP_GET, [] (AsyncWebServerRequest
*request){
        request->send_P(200, "text/plain", String(set_minute).c_str());
    });
    server.on("/setduration", HTTP_GET, [] (AsyncWebServerRequest
*request){
        request->send_P(200, "text/plain", String(set_duration).c_str());
    });
//Yêu cầu 4 response từ server về trạng thái hoạt động của các
actuator
server.on("/control", HTTP_GET, [] (AsyncWebServerRequest *request){
    if (request->hasParam("hs")) {
        manual[0] = (request->getParam("hs")->value()).toInt();
    }
    else if (request->hasParam("wp")) {
        manual[1] = (request->getParam("wp")->value()).toInt();
    }
    else if (request->hasParam("ap")) {
        manual[2] = (request->getParam("ap")->value()).toInt();
    }
    else if (request->hasParam("ff")) {
        manual[3] = (request->getParam("ff")->value()).toInt();
    }
}

```

```

    }
    request->send_P(200, "text/plain", "OK");
});

//Bắt đầu chạy server
server.begin();

```

Ở request root ("/"), index_html là 1 string được tạo ra bằng cú pháp nối chuỗi:

```

const char index_html[] PROGMEM = R"rawliteral(
<Phần code web của người dùng ở đây>
)rawliteral";

```

Khoảng giá trị của các cảm biến: Sau khi tham khảo từ nhiều trang khoa học trên Internet để khoảng giá trị phải có tính thực tế và phù hợp với đa số giống cá cảnh phổ biến, nhóm quyết định cài đặt khoảng giá trị cho các thông số theo 3 mức: thấp, bình thường, cao, chi tiết như sau:

Thông số	Mức thấp	Mức bình thường	Mức cao
Nhiệt độ	20 - 23 °C	23 - 27 °C	27 - 30 °C
Oxy hòa tan	4 - 6 ‰	6 - 8 ‰	8 - 10 ‰
Độ pH	5 - 6.8	6.8 - 7.8	7.8 - 9

Nhóm sử dụng hàm rand() trong ngôn ngữ lập trình C để hiện thực các thông số này.

Ngoài ra, còn có 3 biến cờ (flag) được điều chỉnh thông qua cửa sổ Serial Monitor nhằm thay đổi khoảng giá trị cho các cảm biến để mô phỏng hoạt động của hệ thống. Dưới đây là danh sách các lệnh điều khiển thông qua Serial monitor (theo định dạng !COMMAND!):

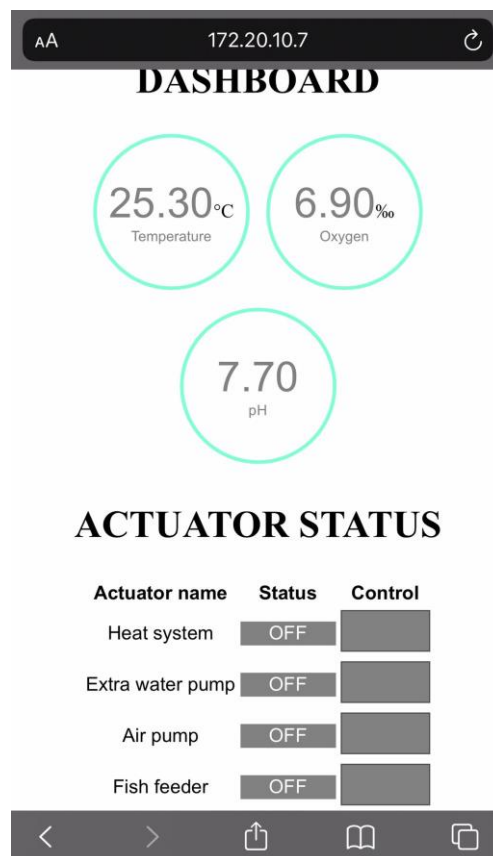
Tên lệnh	Chức năng
temp-low/temp-normal/temp-high	Điều chỉnh dữ liệu cảm biến nhiệt độ “giả” ở các mức tương ứng.
oxygen-low/oxygen-normal/oxygen-high	Điều chỉnh dữ liệu cảm biến oxy hòa tan “giả” ở các mức tương ứng.
ph-low/ph-normal/ph-high	Điều chỉnh dữ liệu cảm biến độ pH “giả” ở các mức tương ứng.
eat-hour-minute-duration	<p>Đặt lịch hẹn cho cá ăn ở thời điểm hour:minute (Định dạng 24 giờ và phải nhập đủ 2 chữ số), bộ cho cá ăn sẽ bật trong [duration] giây.</p> <p>Ví dụ:</p> <p>!eat-16-25-45!: Cho cá ăn vào lúc 16:25 trong vòng 45 giây.</p> <p>!eat 08-03-06!: Cho cá ăn vào lúc 08:03 trong vòng 6 giây.</p> <p>!eat 7-9-20!: Câu lệnh lỗi</p>
time	Xuất ra thời gian thực hiện tại theo định dạng hour:minute:second (Định dạng 24 giờ).
reset	Đặt lại thông số cho tất cả cảm biến “giả” ở mức bình thường.

Link mã nguồn: <https://github.com/phucvien2511/iot-assignment>

Sau khi chạy code (Sử dụng nút **Upload**), người dùng vào cửa sổ Serial Monitor để lấy địa chỉ IP của server để có thể truy cập sau khi vi điều khiển kết nối thành công vào WiFi. Lưu ý điều chỉnh Baud rate là 115200.

```
COM11
11:55:29.581 -> Connecting to WiFi..
11:55:30.407 -> Connecting to WiFi..
11:55:31.386 -> Connecting to WiFi..
11:55:32.413 -> Connecting to WiFi..
11:55:33.387 -> Connecting to WiFi..
11:55:34.414 -> Connecting to WiFi..
11:55:35.394 -> Connecting to WiFi..
11:55:36.416 -> Connecting to WiFi..
11:55:37.397 -> Connecting to WiFi..
11:55:37.397 -> 172.20.10.7
```

Giao diện của server (Địa chỉ IP là 172.20.10.7):

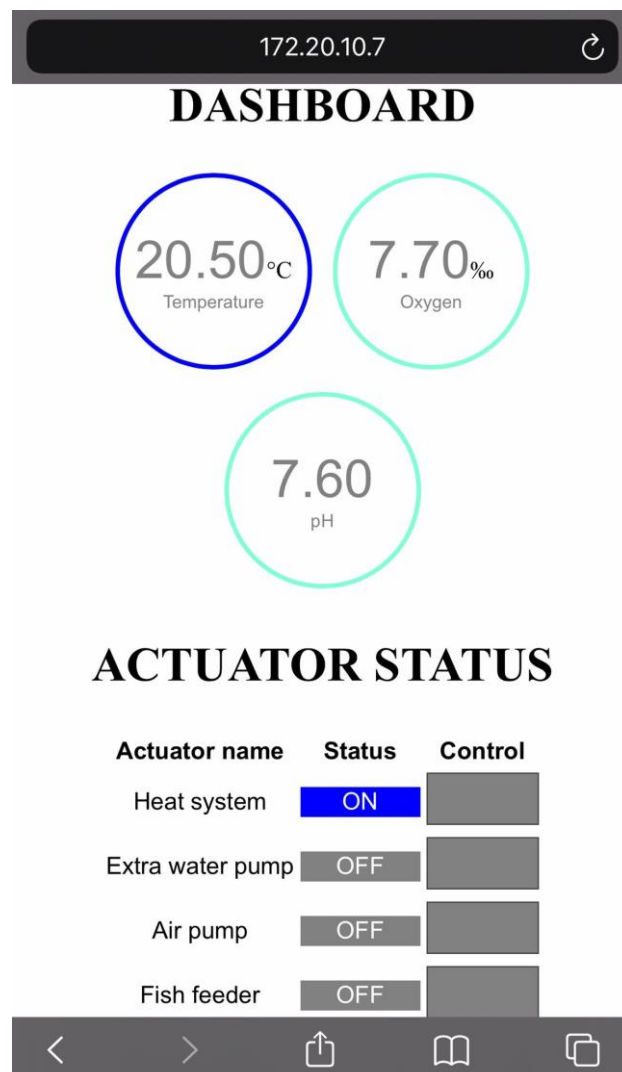


3. Quá trình mô phỏng (có đính kèm link video demo)

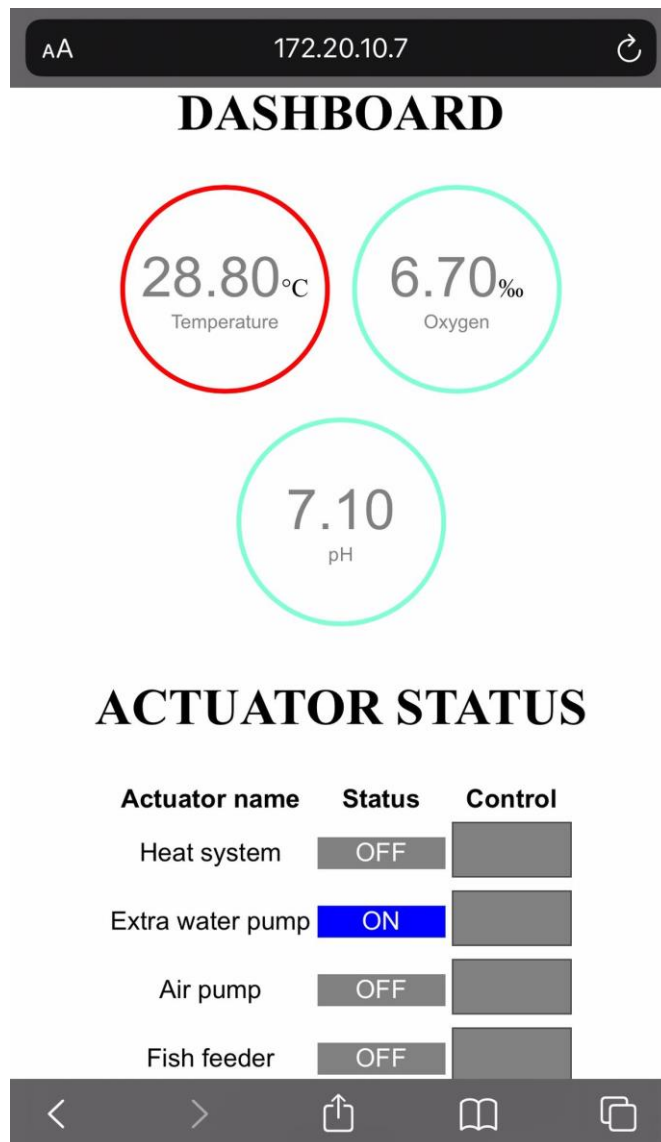
- Điều chỉnh giá trị cảm biến “giả” thông qua Serial Monitor (Ví dụ !temp-low! => Giảm nhiệt độ xuống mức thấp):

```
COM11
11:55:29.381 -> Connecting to WiFi..
11:55:30.407 -> Connecting to WiFi..
11:55:31.386 -> Connecting to WiFi..
11:55:32.413 -> Connecting to WiFi..
11:55:33.387 -> Connecting to WiFi..
11:55:34.414 -> Connecting to WiFi..
11:55:35.394 -> Connecting to WiFi..
11:55:36.416 -> Connecting to WiFi..
11:55:37.397 -> Connecting to WiFi..
11:55:37.397 -> 172.20.10.7
11:55:37.397 -> >> Command: sensor-range, eat-hh-mm-duration, time, reset
11:55:37.397 -> >> Sensors: temp, oxygen, ph
11:55:37.397 -> >> Ranges: low, normal, high
11:55:55.807 -> >> temp-low
11:55:55.807 -> >> Successfully changed temperature to low range
```

Lúc này server phản hồi bằng cách bật hệ thống sưởi, nút Control vẫn màu xám, chứng tỏ đây là hành vi do server tự quyết định:

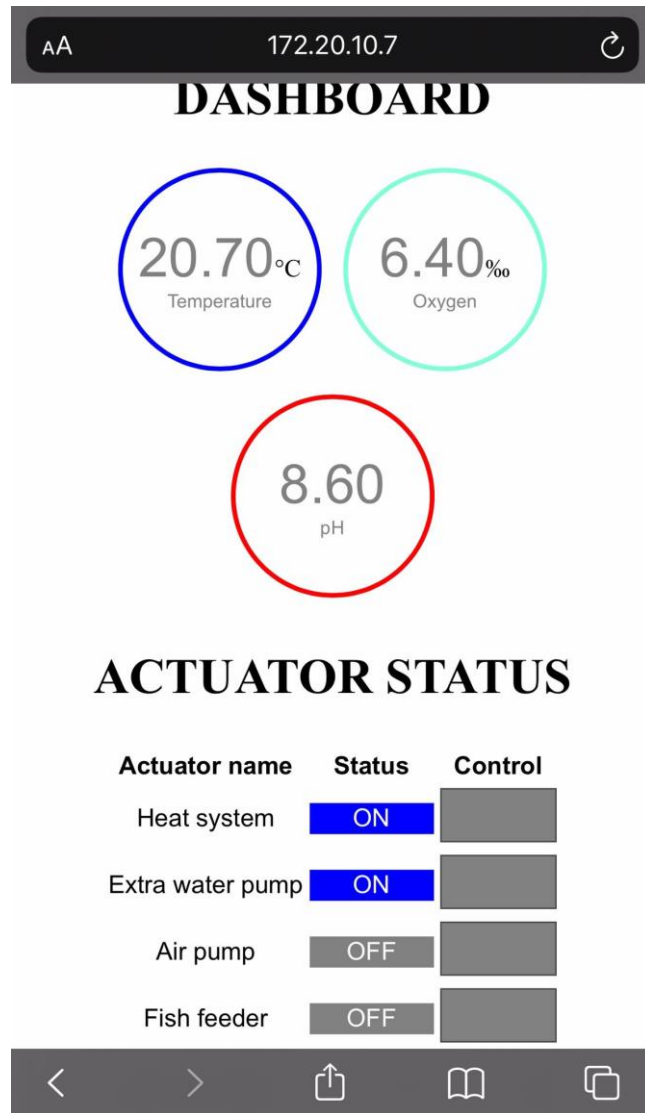


- Trường hợp khác, khi nhiệt độ ở mức cao, máy bơm tăng cường được bật, hệ thống sưởi bị tắt đi để làm mát cho bể:

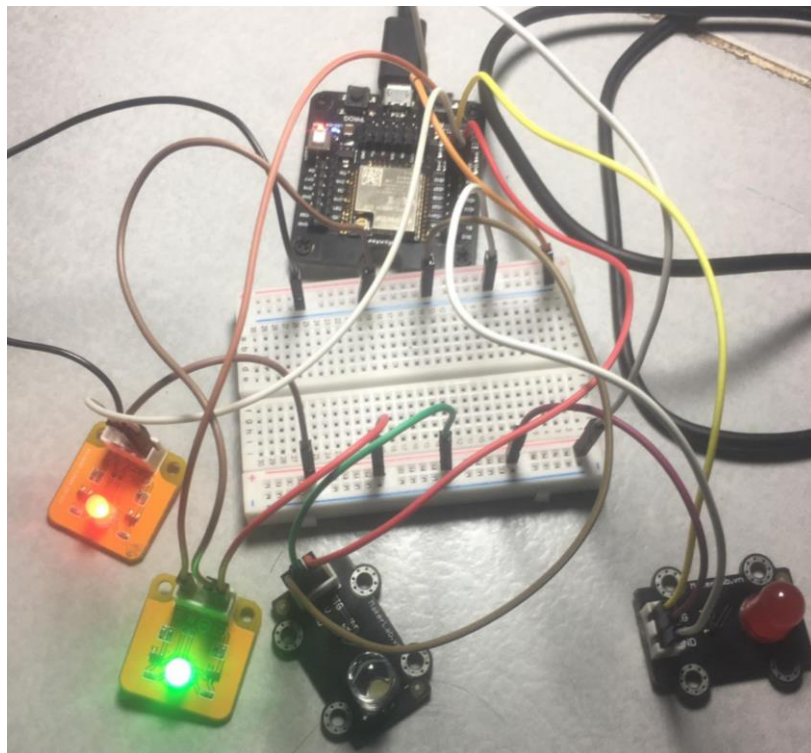


- Trường hợp khác, khi nhiệt độ xuống thấp, đồng thời độ pH ở mức cao, server sẽ bật hệ thống sưởi và máy bơm tăng cường:

```
21:27:26.792 -> >> ph-high
21:27:26.792 -> >> Successfully changed pH to high range
21:27:35.077 -> >> temp-low
21:27:35.077 -> >> Successfully changed temperature to low range
```



Đèn tượng trưng cho hệ thống sưởi và máy bơm nước được bật:



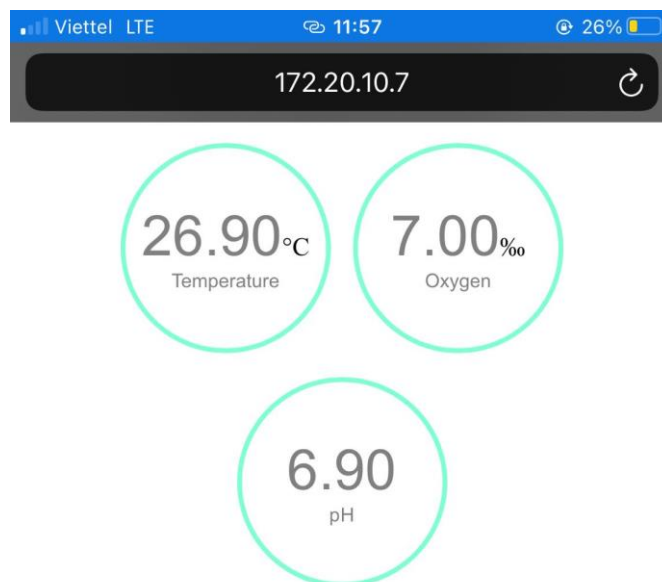
- Mô phỏng việc cho cá ăn: !eat-11-57-30! - cho cá ăn vào lúc 11 giờ 57 phút, trong khoảng thời gian 30 giây.

```
>> eat-11-57-30
>> Successfully setup automatic feeder at 11:57 with duration is 30 seconds
```

Sau đó, vào lúc 11 giờ 57 phút, hệ thống gửi thông báo bộ cho cá ăn đã được kích hoạt:

```
11:56:39.681 -> >> eat-11-57-30
11:56:39.681 -> >> Successfully setup automatic feeder at 11:57 with duration
11:56:59.377 -> >> Activated auto feeder at 11:57
```

Ở phía server, status của bộ cho cá ăn được cập nhật là ON, cũng như thông báo về khoảng thời gian cho ăn đã được hiển thị:



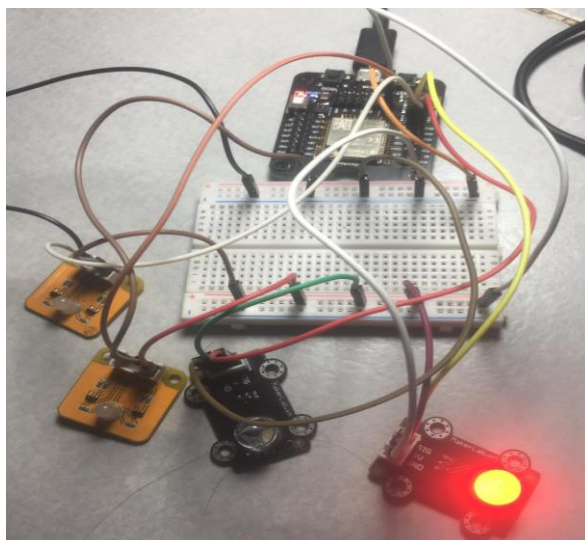
ACTUATOR STATUS

Actuator name	Status	Control
Heat system	OFF	
Extra water pump	OFF	
Air pump	OFF	
Fish feeder	ON	

Fish feeder activated at 11:57 - Duration: 30 second(s).



Đèn báo bộ cho cá ăn đã được bật:



Sau khi khoảng thời gian cho ăn kết thúc, status của fish feeder được cập nhật lại là OFF, cũng như trên Serial Monitor gửi thông báo tắt:

```
11:56:39.681 -> >> eat-11-57-30
11:56:39.681 -> >> Successfully setup automatic feeder at 11:57 with durati
11:56:59.377 -> >> Activated auto feeder at 11:57
11:57:32.761 -> >> Deactivated auto feeder at 11:57
```

Quá trình demo chi tiết hơn cho hệ thống, thầy có thể xem video demo:

<https://drive.google.com/file/d/1EEfSRKpGd-kssohGthLBjVDVT98-ZA70/view?usp=sharing>

4. Tổng quan và đánh giá

4.1. Tổng kết về sản phẩm

- Nhóm đã hoàn thành tương đối một số chức năng cơ bản về dự án bể cá IoT mà nhóm đề ra: hiển thị thông số nhiệt độ, oxy hoà tan, pH thu được từ cảm biến lên Web server, điều khiển các Actuators một cách tự động từ các thông số, thông tin hành vi các Actuators đang hoạt động.

- Giao diện Web Server đơn giản và dễ sử dụng cho người dùng dù chưa được đẹp và bắt mắt. Các thiết bị dự án IoT của nhóm chọn cũng nhỏ gọn, dễ dàng di chuyển và có tính di động cao. Nó cũng dễ dàng cho việc thêm các thiết bị, sensors, actuators,... cũng như mở rộng quy mô dự án (điều khiển nhiều bể cá, thêm nhiều sensor node, các actuators cũng như thêm các thông số cần thiết,...).

- Tuy nhiên, để có thể đưa dự án ra thực tế, nhóm cần cải thiện nhiều mặt như:

+ Giao diện Web server cần được bắt mắt và thân thiện với người dùng hơn.

+ Với việc demo dự án, nhóm chỉ thực hiện các hành vi bật/tắt các actuators tượng trưng bằng các đèn LED, vì vậy nhóm cần trực tiếp thực hiện trên các thiết bị thực tế và cải thiện source code để có thể xử lý các actuators thật có các hành vi hoạt động phức tạp hơn.

4.2. Những khó khăn gặp phải và hướng giải quyết

4.2.1. Khó khăn gặp phải

- Chi phí thực hiện dự án IoT chính là một trong những vấn đề lớn nhất mà nhóm gặp phải. Do thiếu kinh phí và nguồn lực cũng như quy mô của một Bài tập lớn, nhóm chọn những thiết bị rẻ để có thể demo mô phỏng dự án IoT của nhóm mà chưa thực hiện trên các thiết bị mà nhóm đã chọn (với chi phí lên tới hàng triệu đồng).

- Do các thành viên trong nhóm chưa có nhiều kinh nghiệm và kiến thức cần có để thực hiện một dự án IoT (có thành viên chưa học Mạng máy tính, lập trình Web,...) nên nhóm đã gặp nhiều khó khăn trong việc thực hiện dự án làm sao cho phù hợp với thực tế.

- Trong quá trình hiện thực server, nhóm gặp nhiều khó khăn khi tiếp cận với những thư viện mới, với một số bug trong việc layout trang web. Việc kết nối Wifi giữa Web server và vi điều khiển không thực sự ổn định nên trong lúc hiện thực nhóm thỉnh thoảng

gặp trục trặc vì vi điều khiển mất kết nối với Web server. Ngoài ra, việc bố trí ngoại vi và dây cắm chưa thực sự gọn gàng, đẹp mắt.

- Một vấn đề với nhóm khi quá trình thực hiện dự án IoT Bài tập lớn môn học vướng phải thời gian bận và khoảng cách địa lý của các thành viên trong nhóm (bận lịch học, lịch thi, deadline các môn học khác, các thành viên ở cách xa nhau,...) khiến cho việc tập hợp các thành viên để thực hiện dự án IoT gặp đôi chút khó khăn.

4.2.2. Hướng giải quyết

- Về bài toán chi phí, nhóm có thể lập quỹ chung để chia sẻ, mua sắm một số thiết bị thật để hoàn thiện dự án tốt hơn.

- Tự mày mò, học hỏi, tìm hiểu và bổ sung thêm các kiến thức cần thiết để có thể hoàn thiện và mở rộng hơn nữa quy mô dự án. Kết hợp với việc tìm hiểu các documentation cần thiết để việc lập trình trở nên tốt hơn.

- Các thành viên trong nhóm sẽ thường xuyên thảo luận, bàn bạc, chủ động thời gian nhóm tập trung một nơi để làm việc với nhau nhiều nhất có thể. Nếu trong trường hợp bất khả kháng, nhóm tổ chức họp và làm việc online. Các thành viên trong nhóm cũng cần có trách nhiệm và sự chủ động, cố gắng hoàn thành sớm phần công việc được phân công.

4.3. Hướng phát triển trong tương lai

- Nhóm sẽ mở rộng hơn quy mô của dự án: Thêm các sensors cộng với thêm các thông số cần thiết khác, thêm các actuators để việc quản lý cá trong bể được kỹ lưỡng, chính chu và chi tiết hơn. Mở rộng các gateway để có thể quản lý nhiều bể cá hơn, tạo app giao diện người dùng thân thiện để có thể quản lý một hay nhiều bể cá trong nhà, thêm database vào backend để người dùng có thể theo dõi lịch sử đo của cảm biến, thêm chức năng login để cho phép nhiều người dùng sử dụng.

- Ứng dụng thêm thị giác máy tính và AI để theo dõi sức khỏe của cá (cá bệnh, cá chết...). Lập trình cho gateway có thể tự động nhận biết để gửi tín hiệu đến actuators thực hiện các chức năng phù hợp trong trường hợp mất kết nối với server.

5. Tài liệu tham khảo

1. Diệp Thị Quế Ngân, Nguyễn Minh Đức. *Hiện trạng nuôi cá cảnh giải trí của người dân thành phố Hồ Chí Minh*. Truy cập từ:
https://fof.hcmuaf.edu.vn/data/file/51_%20DTQ%20Ngan%20_%20NM%20Duc-DHNL-Hien%20trang%20nuoi%20ca%20canh.pdf.
2. *ESP32 Async Web Server – Control Outputs with Arduino IDE (ESPAsyncWebServer library)*. Truy cập từ:
<https://randomnerdtutorials.com/esp32-async-web-server-espasyncwebserver-library>.
3. Water & Aquarium Temperature. Truy cập từ: <https://fishlab.com/aquarium-temperature>.
4. What is pH and Why Is It Important?. Truy cập từ:
<https://www.aqueon.com/articles/ph-importance#:~:text=Most%20freshwater%20tropical%20fish%20do,or%20lower%20than%20these%20values>.
5. W3Schools Online Web Tutorials. Truy cập từ: <https://www.w3schools.com>.