

BÁO CÁO DỰ ÁN CUỐI KỲ  
CƠ SỞ ĐO LƯỜNG VÀ ĐIỀU KHIỂN SỐ

Hệ thống theo dõi nông  
nghiệp thông minh

# Thành viên nhóm 20

Họ và tên	Mã sinh viên
Nguyễn Trọng Nam	22022161
Cao Song Toàn	22022188
Doãn Đức Minh	22022135
Lê Thế Minh	22022215

# HỆ THỐNG THEO DÕI NÔNG NGHIỆP THÔNG MINH

I. Giới Thiệu Chung

II. Thiết kế hệ thống

III. Thí nghiệm và đánh  
giá

VI. Kết luận.

# I. GIỚI THIỆU CHUNG

## MỤC TIÊU ĐỀ TÀI

DỰ ÁN TẬP TRUNG PHÁT TRIỂN MỘT HỆ THỐNG NÔNG NGHIỆP THÔNG MINH, NHẪM GIẢI QUYẾT HẠN CHẾ CỦA MÔ HÌNH CANH TÁC TRUYỀN THỐNG VỐN PHỤ THUỘC NHIỀU VÀO KINH NGHIỆM VÀ LAO ĐỘNG THỦ CÔNG.

## PHẠM VI ĐỀ TÀI

VỚI SỰ HỖ TRỢ CỦA CÔNG NGHỆ IOT, HỆ THỐNG CÓ THỂ GIÁM SÁT LIÊN TỤC CÁC THÔNG SỐ MÔI TRƯỜNG NHƯ ÁNH SÁNG, MƯA, NHIỆT ĐỘ, ĐỘ ẨM KHÔNG KHÍ VÀ ĐỘ ẨM DỮ LIỆU SAU KHI ĐƯỢC XỬ LÝ SẼ ĐƯỢC HIỂN THỊ THÔNG TIN THỜI GIAN THỰC TRÊN WEB VÀ ĐỒNG BỘ DỮ LIỆU LÊN NỀN TẢNG ĐÁM MÂY PHỤC VỤ PHÂN TÍCH, LƯU TRỮ. NGOÀI RA, NGƯỜI DÙNG CÓ THỂ LỰA CHỌN CHẾ ĐỘ ĐIỀU KHIỂN MÁY BƠM NƯỚC THỦ CÔNG HOẶC TỰ ĐỘNG THÔNG QUA GIAO DIỆN WEB. PHẠM VI DỰ ÁN HIỆN DỪNG Ở MỨC GIÁM SÁT VÀ ĐIỀU KHIỂN CƠ BẢN THÔNG QUA WEBSERVER KẾT HỢP ĐIỆN TOÁN ĐÁM MÂY.

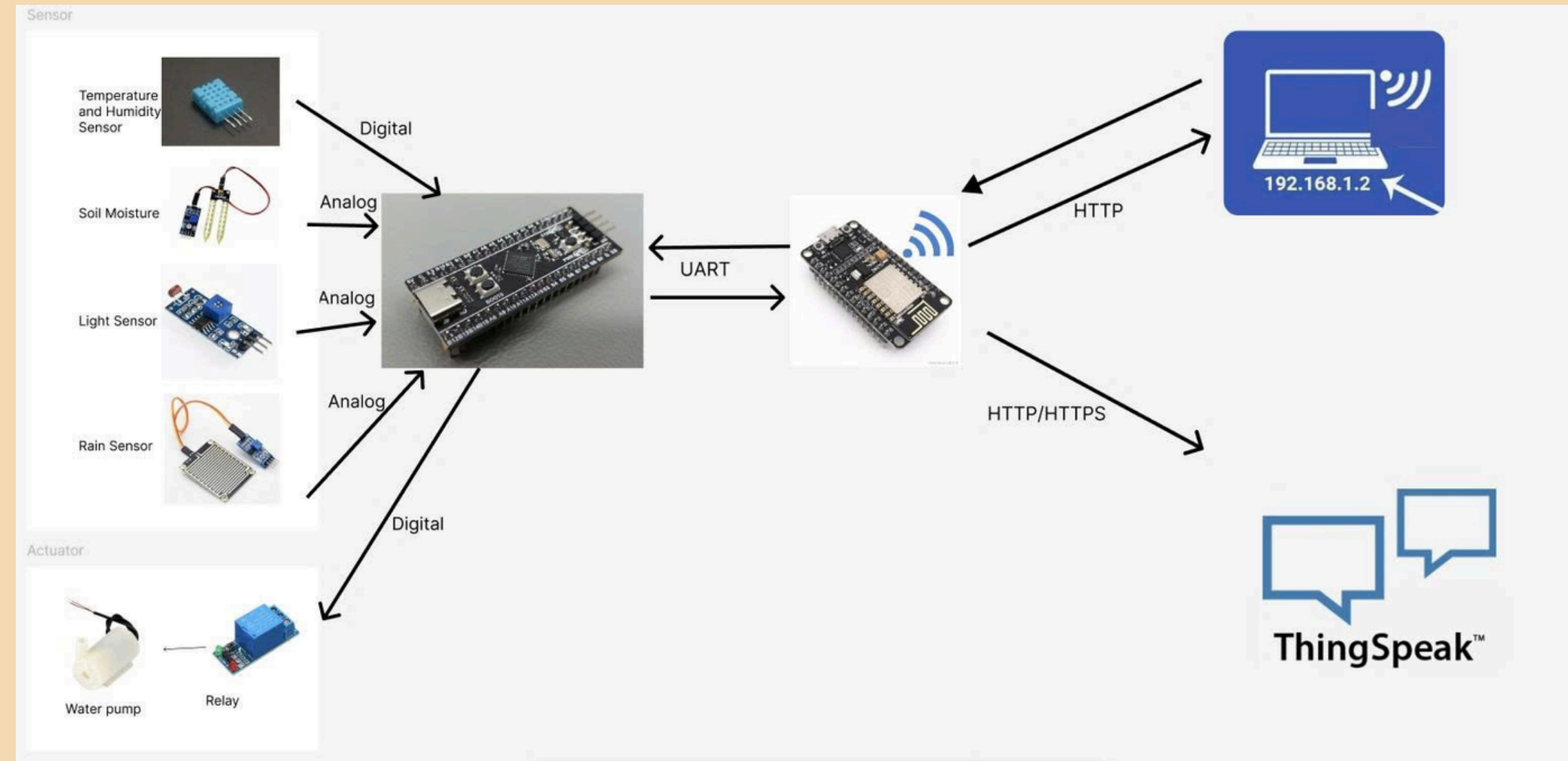
**CÔNG NGHỆ SỬ DỤNG:** STM32, UART, ESP32, THINGSPEAK, WEBSERVER



# II. THIẾT KẾ HỆ THỐNG

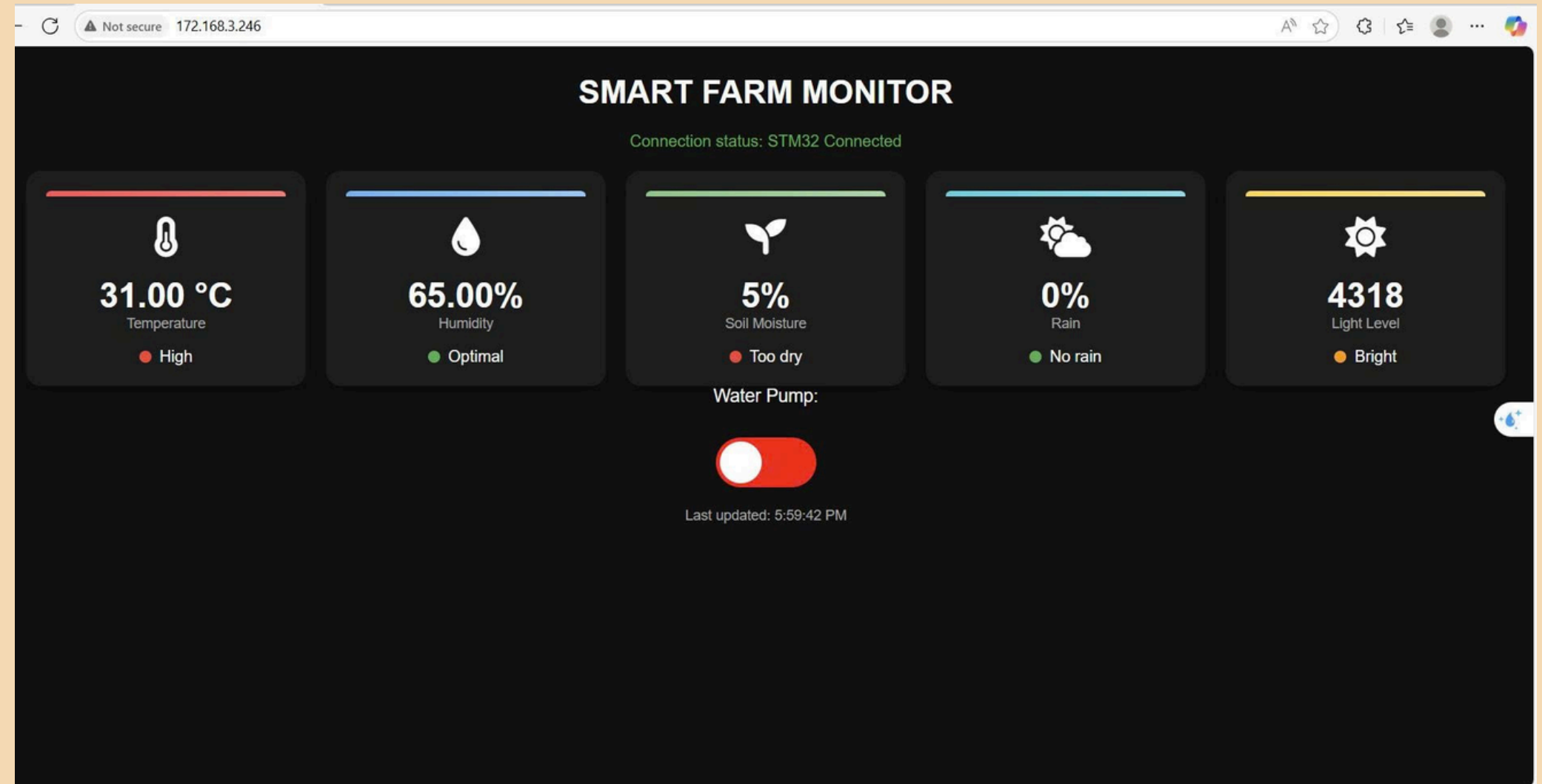
**Hệ thống vận hành theo các bước như sau:**

1. Thu thập dữ liệu: Các cảm biến môi trường bao gồm:
  - DHT11 (đo nhiệt độ & độ ẩm không khí) xuất tín hiệu số (Digital)
  - Cảm biến độ ẩm đất, ánh sáng (LDR) và mưa xuất tín hiệu tương tự (Analog) → Tất cả tín hiệu được đưa về STM32 để xử lý.
2. Xử lý tại STM32: STM32 đọc tín hiệu digital qua GPIO và tín hiệu analog qua ADC. Sau đó tiến hành lọc nhiễu, hiệu chuẩn, đóng gói dữ liệu thành chuỗi ký tự định dạng cách nhau bởi dấu phẩy và kết thúc bằng dấu xuống dòng.



3. Truyền dữ liệu qua UART: Chuỗi dữ liệu ví dụ "45,26,59,48,0 " được gửi từ STM32 sang ESP32 thông qua giao tiếp UART hai chiều.
4. ESP32 hiển thị và lưu trữ: ESP32 nhận và phân tích dữ liệu, hiển thị real-time lên giao diện web thông qua HTTP (truy cập nội mạng với IP như 192.168.1.2), đồng thời định kỳ gửi dữ liệu lên nền tảng ThingSpeak qua HTTP/HTTPS để lưu trữ và phân tích.
5. Điều khiển bơm: Khi người dùng điều khiển từ web (bật/tắt), ESP32 gửi lệnh "1 " hoặc "0 " qua UART về STM32. STM32 sẽ xuất tín hiệu digital qua GPIO để kích hoạt hoặc ngắt relay, điều khiển bơm nước.

# II. THIẾT KẾ HỆ THỐNG



*GIAO DIỆN WEB*

THINKSPEAK LƯU TRỮ VÀ  
HIỂN THỊ DỮ LIỆU

# III. THÍ NGHIỆM VÀ ĐÁNH GIÁ

## Môi trường và cấu hình thí nghiệm

Môi trường	<p>Địa điểm: Văn phòng FSOF (mô phỏng trong nhà).</p> <ul style="list-style-type: none"><li>- Điều kiện độ ẩm đất: Sử dụng giấy ăn bọc ngoài đầu cảm biến ( sử dụng giấy ăn là để dễ kiểm soát được độ ẩm). Bản chất của tỉ lệ độ ẩm này là lượng nước / lượng nước tối đa có thể chứa, tính lượng nước tối đa có thể chứa của 1 nắm giấy ăn bằng cách nhúng nó vào nước sau đó vắt khô vào 1 cốc thì lượng nước có trong cốc là lượng chứa tối đa từ đó ước lượng được chính xác các khoảng cần thí nghiệm.</li><li>- Ánh sáng: Đặt trong phòng dưới ánh sáng phòng, cường độ ~500 lux, được test các thay đổi bằng cách lấy nguồn sáng từ đèn flash trên điện thoại. ( lấy số liệu lux từ nhà sản xuất)</li><li>- Nhiệt độ phòng: <math>25 \pm 2</math> °C.</li><li>- Mô phỏng mưa: Sử dụng bình phun tia nước định lượng. ( lượng mưa được xác định bằng độ phủ trên bề mặt cảm biến)</li></ul>
Cấu hình	<ul style="list-style-type: none"><li>- Giao tiếp UART (STM32-ESP32): Tốc độ 9600 bps, 8 bit dữ liệu, không parity, 1 stop bit (8N1).</li><li>- Tần suất cập nhật Webserver: 1 lần/giây.</li><li>- Tần suất gửi dữ liệu lên ThingSpeak: 1 lần/15 giây (theo giới hạn của gói miễn phí).</li><li>- Nguồn cấp: 5V DC.</li></ul>

# III. THÍ NGHIỆM VÀ ĐÁNH GIÁ

Bảng tổng hợp kết quả

	Chỉ số	Kết quả
Độ trễ hiển thị	Trung bình từ STM32 → Webserver	~ 180 ms
	Trung bình từ STM32 → ThingSpeak	~ 1.2 s
Điều khiển bơm	Thời gian phản hồi trung bình	~ 1s
	Tỷ lệ thực hiện lệnh thành công	95% (95/100 lần)
Độ ổn định	Thời gian chạy liên tục	1 tiếng
	Dòng tiêu thụ trung bình	130 mA @ 5V
	Sự cố phát sinh	Không tràn bộ nhớ, không treo hệ thống
Giá trị cảm biến	Khi thay đổi điều kiện môi trường	Đáp ứng chính xác



# Các vấn đề gặp phải

Trong quá trình thí nghiệm, hệ thống đã bộc lộ hai vấn đề cần được cải thiện:

1. Hiện tượng trễ (lag) của Webserver sau thời gian dài hoạt động:

- Mô tả: Sau khoảng 30 phút hoạt động liên tục, các tác vụ trên giao diện web (tải lại, nhấn nút) có hiện tượng phản hồi chậm hơn so với ban đầu.
- Phân tích nguyên nhân: Nguyên nhân có thể do việc quản lý bộ nhớ trên ESP32 chưa được tối ưu hoàn toàn, dẫn đến rò rỉ bộ nhớ (memory leak) sau nhiều chu kỳ cập nhật dữ liệu. Ngoài ra, việc thiếu cơ chế tản nhiệt cho ESP32 cũng có thể khiến vi điều khiển bị quá nhiệt, làm giảm hiệu suất xử lý.

2. Nhiễu tín hiệu cảm biến khi điều khiển bơm:

- Mô tả: Tại thời điểm relay đóng/ngắt để bật/tắt máy bơm, giá trị đọc từ các cảm biến analog (ánh sáng, độ ẩm đất) bị dao động đột ngột trong khoảnh khắc.
- Phân tích nguyên nhân: Vấn đề này có thể xuất phát từ hai nguyên nhân chính:
  - Sụt áp nguồn: Máy bơm khi khởi động tiêu thụ một dòng điện lớn tức thời, gây ra sụt áp tạm thời trên toàn hệ thống. Sự sụt áp này ảnh hưởng đến điện áp tham chiếu ( $V_{ref}$ ) của bộ ADC trên STM32, dẫn đến sai lệch kết quả đo.
  - Nhiễu điện từ (EMI): Hoạt động đóng/ngắt của relay tạo ra tia lửa điện và nhiễu điện từ, có thể lan truyền qua đường dây và ảnh hưởng đến các tín hiệu analog nhạy cảm.

# IV. KẾT LUẬN

Thông qua PROJECT Nhóm đã làm được

- Xây dựng được hệ thống hiển thị dữ liệu môi trường (độ ẩm đất, nhiệt độ, độ ẩm không khí, ánh sáng, lượng mưa) với vi điều khiển STM32, đảm bảo đọc và xử lý tín hiệu chính xác thông qua lọc nhiễu và hiệu chuẩn
- Thiết lập thành công kênh truyền UART giữa STM32 và ESP32, với định dạng dữ liệu rõ ràng, đồng bộ baudrate ổn định, giúp truyền – nhận dữ liệu nhịp nhàng và tin cậy
- Phát triển giao diện webserver trên ESP32, cập nhật thông số theo thời gian thực và tích hợp chức năng điều khiển bơm từ xa, đồng thời gửi dữ liệu lên ThingSpeak để lưu trữ và phân tích dài hạn