# Hệ thống tưới cây tự động thông minh (Smart Plant Watering System - SPWS)

Thiết kế và lập trình một hệ thống nhúng quản lý việc tưới nước cho cây trồng. **Hệ thống phải có khả năng đọc dữ liệu cảm biến, đưa ra quyết định tưới nước dựa trên các tham số cấu hình, và thông báo trạng thái hoạt động**. Đề bài tập trung vào việc hiểu và hiện thực hóa một hệ thống tự động hóa thực tế với các yêu cầu về quản lý trạng thái, dữ liệu và tương tác người dùng.

**Giả định phần cứng và tài nguyên**:

1. Vi điều khiển: Một vi điều khiển phổ biến với các tài nguyên cơ bản (GPIO, Timer, ADC, một lượng RAM/Flash đủ dùng cho ứng dụng nhỏ).
2. Cảm biến:
   1. Cảm biến độ ẩm đất: Đo phần trăm độ ẩm của đất (ví dụ: 0-100%).
   2. Cảm biến nhiệt độ môi trường: Đo nhiệt độ không khí xung quanh.
3. Thiết bị:
   1. Bơm nước mini: Được điều khiển để BẬT/TẮT việc tưới nước.
   2. LED trạng thái: Một LED RGB hoặc ba LED đơn sắc để hiển thị trạng thái hoạt động (ví dụ: Xanh - bình thường, Vàng - đang tưới, Đỏ - lỗi/cảnh báo).
   3. Nút nhấn: Hai nút nhấn vật lý:
      1. Nút 1: BẬT/TẮT chế độ tự động.
      2. Nút 2: Kích hoạt tưới nước thủ công (khi hệ thống ở chế độ chờ)/
4. Giao tiếp (mô phỏng): Hàm gửi thông báo trạng thái hoặc dữ liệu quan trọng đến một giao diện người dùng (ví dụ: in ra console, hoặc gửi qua UART/mô-đun không dây).

**Yêu cầu phần mềm**:

1. **Định nghĩa các trạng thái hoạt động**:
   1. Chế độ hệ thống:
      1. MODE\_AUTO: Hệ thống tự động quản lý việc tưới nước dựa trên cảm biến.
      2. MODE\_MANUAL: Hệ thống chờ lệnh thủ công hoặc bị khóa tự động.
   2. Trạng thái tưới nước của bơm:
      1. PUMP\_OFF: Bơm không hoạt động.
      2. PUMP\_ON: Bơm đang hoạt động.
   3. Trạng thái LED:
      1. LED\_NORMAL: Hệ thống hoạt động bình thường, không tưới.
      2. LED\_WATERING: Hệ thống đang tưới nước.
      3. LED\_LOW\_MOISTURE\_ALERT: Độ ẩm đất quá thấp (chưa tưới hoặc tưới không đủ).
      4. LED\_ERROR: Lỗi hệ thống (ví dụ: cảm biến hỏng, bơm kẹt).
2. **Cấu trúc dữ liệu chính**:
   1. Dữ liệu cảm biến: Cần có một cấu trúc để lưu trữ giá trị đọc được từ cảm biến độ ẩm đất và nhiệt độ môi trường.
   2. Cài đặt hệ thống: Cần một cấu trúc để lưu trữ các tham số cấu hình như:
      1. Ngưỡng độ ẩm tối thiểu để bắt đầu tưới.
      2. Ngưỡng độ ẩm tối đa để dừng tưới.
      3. Thời gian tưới tối đa cho mỗi lần.
      4. Chu kỳ kiểm tra cảm biến (ví dụ: mỗi 5 phút).
      5. Trạng thái chế độ hiện tại (Tự động/Thủ công).
3. **Yêu cầu chức năng**:
   1. Khởi tạo hệ thống:
      1. Thiết lập các chân GPIO cho cảm biến, bơm, LED, nút nhấn.
      2. Đặt các giá trị mặc định ban đầu cho các cài đặt hệ thống (ngưỡng độ ẩm, thời gian tưới, v.v.).
      3. Đưa hệ thống vào chế độ MODE\_AUTO mặc định ban đầu.
   2. Đọc và xử lý dữ liệu cảm biến:
      1. Định kỳ đọc giá trị từ cảm biến độ ẩm đất và nhiệt độ.
      2. Lưu trữ các giá trị này vào cấu trúc dữ liệu cảm biến.
      3. Có thể thực hiện làm mịn dữ liệu (averaging) nếu cần.
   3. Logic điều khiển tưới nước tự động:
      1. Nếu hệ thống đang ở MODE\_AUTO:
         * So sánh độ ẩm đất hiện tại với ngưỡng tưới tối thiểu.
         * Nếu độ ẩm quá thấp và đã hết thời gian chờ giữa các lần tưới, kích hoạt bơm.
         * Khi bơm hoạt động, tiếp tục theo dõi độ ẩm và dừng bơm khi đạt ngưỡng tối đa hoặc hết thời gian tưới tối đa.
      2. Cần có một bộ đếm thời gian (timer) để quản lý chu kỳ kiểm tra cảm biến và thời gian tưới nước.
   4. Điều khiển bơm:
      1. Bật/Tắt bơm nước dựa trên quyết định của logic điều khiển.
      2. Cần có hàm điều khiển trạng thái bơm (turn\_pump\_on(), turn\_pump\_off()).
   5. Điều khiển LED trạng thái:
      1. Luôn cập nhật màu LED dựa trên trạng thái hoạt động của hệ thống (bình thường, đang tưới, độ ẩm thấp, lỗi).
   6. Xử lý nút nhấn:
      1. Nút BẬT/TẮT chế độ tự động: Khi nhấn, chuyển đổi giữa MODE\_AUTO và MODE\_MANUAL. Khi chuyển sang MODE\_MANUAL, đảm bảo bơm nước đang TẮT.
      2. Nút tưới thủ công: Chỉ có tác dụng khi hệ thống ở MODE\_MANUAL. Khi nhấn, kích hoạt bơm nước trong một khoảng thời gian cố định (ví dụ: 10 giây), sau đó tự động tắt.
   7. Thông báo trạng thái:
      1. Định kỳ hoặc khi có thay đổi trạng thái quan trọng, gửi thông báo về tình hình hệ thống (ví dụ: "Độ ẩm đất: 35%", "Đang tưới nước...", "Chế độ: Thủ công").
      2. Tối ưu bộ nhớ: Tránh sử dụng quá nhiều chuỗi ký tự cố định (const char\*) nếu không gian Flash hạn chế. Có thể sử dụng các mã số hoặc enum để biểu thị thông báo.
4. **Luồng hoạt động chính**:
   1. Sau khi khởi tạo, hệ thống sẽ chạy trong một vòng lặp vô hạn.
   2. Trong mỗi vòng lặp, hệ thống sẽ:
      1. Kiểm tra và xử lý các sự kiện nút nhấn.
      2. Nếu ở chế độ tự động, kiểm tra và xử lý logic tưới nước.
      3. Cập nhật trạng thái LED.
      4. Có thể gửi báo cáo trạng thái định kỳ.
      5. Có một độ trễ nhỏ để tiết kiệm năng lượng và phù hợp với chu kỳ hoạt động.

**Yêu cầu triển khai**:

1. Triển khai toàn bộ chức năng của hệ thống SPWS bằng ngôn ngữ C.
2. Tổ chức mã nguồn một cách rõ ràng, dễ đọc, với các file header và source file riêng biệt cho các module (ví dụ: main.c, sensors.c, watering\_logic.c, actuators.c, buttons.c, config.h).
3. Sử dụng enum và struct phù hợp để quản lý trạng thái và dữ liệu.
4. Quản lý các biến static và extern một cách hợp lý để điều khiển phạm vi và chia sẻ dữ liệu giữa các module.
5. Mô phỏng hoạt động của cảm biến, bơm nước, LED và nút nhấn bằng cách in ra console các thông báo và thay đổi trạng thái tương ứng.
6. Đảm bảo hệ thống có thể chuyển đổi mượt mà giữa các chế độ và xử lý các tình huống như độ ẩm quá thấp, quá cao, hoặc người dùng can thiệp.