**Báo cáo chi tiết tích hợp Keypad trong hệ thống quản lý công việc**

**Giới thiệu**

Báo cáo này trình bày chi tiết việc tích hợp bàn phím 4x4 vào hệ thống quản lý công việc trên nền tảng ESP32, kết hợp với màn hình TFT ILI9341 và các thành phần khác. Bàn phím được sử dụng để công nhân tương tác với hệ thống, chọn công việc (lắp ráp, sửa chữa, đóng gói), ghi nhận kết quả công việc (hoàn thành hoặc lỗi), và điều hướng qua các menu. Báo cáo tập trung vào tích hợp bàn phím, cơ chế hoạt động, các bước triển khai, ưu điểm, hạn chế và đề xuất cải tiến.

**Mục tiêu**

Việc tích hợp bàn phím 4x4 nhằm đạt các mục tiêu sau:

* Cung cấp giao diện nhập liệu trực quan cho công nhân để chọn công việc và ghi nhận kết quả.
* Hỗ trợ điều hướng qua các menu (menu chính, menu công việc, menu làm việc) một cách dễ dàng.
* Đảm bảo tính chính xác và phản hồi nhanh chóng khi công nhân nhấn phím.
* Hiển thị trạng thái và hướng dẫn trên màn hình TFT để hỗ trợ tương tác.

**Tích hợp Keypad**

Bàn phím 4x4 được sử dụng để nhập liệu, với 16 phím bao gồm số (0-9), ký tự đặc biệt (\*, #) và chữ cái (A-D). Thư viện Keypad được tích hợp để quản lý bàn phím.

**Cấu hình Keypad**

* **Chân kết nối**:
  + Hàng: GPIO 13, 14, 12, 27
  + Cột: GPIO 26, 25, 33, 32
* **Bố cục phím**:
* {'1', '2', '3', 'A'},
* {'4', '5', '6', 'B'},
* {'7', '8', '9', 'C'},
* {'\*', '0', '#', 'D'}
* **Thư viện**: Keypad để khởi tạo và đọc trạng thái phím.

**Các hàm liên quan đến Keypad**

* **Khởi tạo Keypad**:
  + Trong hàm setup(), bàn phím được khởi tạo với bố cục phím và các chân GPIO bằng Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);.
* **Đọc phím**:
  + Hàm keypad.getKey() được gọi trong loop() để lấy ký tự phím được nhấn, trả về 0 nếu không có phím nào được nhấn.
* **Xử lý phím**:
  + Các phím được xử lý trong một switch-case dựa trên trạng thái hệ thống (CHECK\_IN, MAIN\_MENU, WORK\_MENU, TASK\_MENU, WORKING, CHECK\_OUT).
  + Ví dụ:
    - Phím '1' trong CHECK\_IN kích hoạt quét RFID.
    - Phím '1', '2', '3' trong WORK\_MENU chọn công việc tương ứng (lắp ráp, sửa chữa, đóng gói).
    - Phím '\*' dùng để quay lại menu trước đó.

**Cơ chế hoạt động**

* **Điều hướng menu**:
  + Bàn phím cho phép công nhân chọn các tùy chọn trong menu chính (MAIN\_MENU), menu công việc (WORK\_MENU), và menu làm việc (TASK\_MENU, WORKING).
  + Phím '\*' được sử dụng để quay lại menu trước đó, đảm bảo điều hướng linh hoạt.
* **Ghi nhận công việc**:
  + Trong trạng thái WORKING, phím '1' ghi nhận một công việc hoàn thành, phím '2' ghi nhận một lỗi.
  + Kết quả được hiển thị trên TFT và lưu vào biến cục bộ trước khi gửi lên cơ sở dữ liệu.
* **Check-in và Check-out**:
  + Phím '1' kích hoạt quét RFID trong các trạng thái CHECK\_IN và CHECK\_OUT.
  + Phím '0' trong CHECK\_OUT (sau khi hiển thị tóm tắt) đưa hệ thống về trạng thái ban đầu.
* **Xử lý lỗi**:
  + Nếu nhấn phím không hợp lệ, hệ thống hiển thị thông báo lỗi trên TFT và yêu cầu nhập lại.

**Các bước triển khai**

1. **Cấu hình phần cứng**:
   * Kết nối bàn phím 4x4 với ESP32 qua các chân GPIO (hàng: 13, 14, 12, 27; cột: 26, 25, 33, 32).
   * Đảm bảo kết nối chắc chắn và không bị chập mạch.
2. **Cài đặt thư viện**:
   * Cài đặt thư viện Keypad trong Arduino IDE.
   * Định nghĩa bố cục phím và chân GPIO trong mã nguồn.
3. **Triển khai đọc phím**:
   * Sử dụng keypad.getKey() để phát hiện phím nhấn trong hàm loop().
   * Chỉ xử lý phím trong các trạng thái yêu cầu nhập liệu để tối ưu hiệu suất.
4. **Tích hợp giao diện người dùng**:
   * Hiển thị menu và hướng dẫn trên màn hình TFT tương ứng với từng trạng thái.
   * Cập nhật nội dung TFT khi nhấn phím để phản hồi hành động của công nhân.
5. **Xử lý trạng thái**:
   * Sử dụng enum State để quản lý các trạng thái hệ thống.
   * Xây dựng logic switch-case để xử lý phím dựa trên trạng thái hiện tại.
6. **Thử nghiệm và gỡ lỗi**:
   * Kiểm tra phản hồi của từng phím trong các trạng thái khác nhau.
   * Đảm bảo thông báo trên TFT và Serial Monitor hiển thị đúng khi nhấn phím hợp lệ hoặc không hợp lệ.

**Ưu điểm**

* **Dễ sử dụng**: Bàn phím 4x4 cung cấp giao diện nhập liệu đơn giản, phù hợp với công nhân không cần đào tạo phức tạp.
* **Linh hoạt**: Hỗ trợ nhiều chức năng (chọn công việc, ghi nhận kết quả, điều hướng) với số lượng phím hạn chế.
* **Phản hồi nhanh**: Thời gian xử lý phím nhanh, đảm bảo trải nghiệm mượt mà.
* **Tích hợp tốt với TFT**: Menu và thông báo được cập nhật đồng bộ với thao tác phím.

**Hạn chế**

* **Số phím giới hạn**: Chỉ có 16 phím, hạn chế khả năng mở rộng cho các chức năng phức tạp hơn.
* **Không có xác nhận nhập liệu**: Không có cơ chế xác nhận trước khi thực hiện hành động quan trọng (như check-out).
* **Phụ thuộc vào phần cứng**: Nếu bàn phím bị lỗi hoặc tiếp xúc kém, hệ thống sẽ không nhận được đầu vào.
* **Xử lý lỗi cơ bản**: Chỉ hiển thị thông báo "Phím không hợp lệ" mà không cung cấp gợi ý chi tiết.

**Đề xuất cải tiến**

* Thêm cơ chế xác nhận (ví dụ: nhấn phím '#' để xác nhận hành động quan trọng).
* Sử dụng bàn phím lớn hơn (như 4x5) để hỗ trợ thêm chức năng.
* Tích hợp âm thanh phản hồi (như buzzer) khi nhấn phím để tăng trải nghiệm người dùng.
* Cải thiện xử lý lỗi bằng cách hiển thị gợi ý phím hợp lệ cụ thể cho từng trạng thái.

**Kết luận**

Tích hợp bàn phím 4x4 vào hệ thống quản lý công việc cung cấp một giao diện nhập liệu hiệu quả và dễ sử dụng. Bàn phím cho phép công nhân điều hướng menu, chọn công việc và ghi nhận kết quả một cách thuận tiện. Tuy nhiên, cần cải thiện số lượng phím, xử lý lỗi và thêm cơ chế xác nhận để tăng độ tin cậy. Trong tương lai, việc kết hợp bàn phím với các phương pháp nhập liệu khác (như cảm ứng) có thể nâng cao tính linh hoạt của hệ thống.