

Use Case UNO R4 WiFi - Hệ Thống Quản Lý Bãi Đỗ Xe Thông Minh

Mô Tả Tổng Quát

UNO R4 WiFi là bộ điều khiển cổng vào/ra, quản lý 2 barrier (cơ khí chắn đường), 2 RFID reader, 2 ultrasonic sensor. UNO R4 cũng phát WiFi AP để kết nối với ESP32 và Backend.

Use Case 1: Khởi Tạo WiFi AP - UNO R4 Phát Sóng WiFi

Trường	Nội Dung
Tên Use Case	Khởi Tạo WiFi AP - UNO R4 Tạo Mạng Riêng
Mô Tả	UNO R4 phát WiFi AP (SSID: UNO-R4-AP, IP: 192.168.4.2), cho phép ESP32 và Backend kết nối.
Tác Nhân	UNO R4 (Hardware), ESP32, Backend Server
Mô Tả Chi Tiết	UNO R4 WiFiS3 AP mode. SSID="UNO-R4-AP", Password="12345678", IP=192.168.4.2. Khởi động WebServer port 80 để nhận HTTP request.
Điều Kiện Tiên Quyết	UNO R4 WiFi module hoạt động; WiFiS3 library install; RFID/Servo/Ultrasonic init
Luồng Chính	1. setup() khởi động 2. Init Serial, SPI, Servo, Ultrasonic pins 3. WiFi.beginAP("UNO-R4-AP", "12345678") 4. Cấu hình IP tĩnh: 192.168.4.2 5. WebServer begin() port 80 6. Log "WiFi AP started" 7. Init 2 barriers (IN/OUT) 8. sẵn sàng nhận request
Luồng Thay Thế	A1: WiFi Local Mode - Kết nối router thay vì AP
Luồng Ngoài Lề	E1: WiFi init fail → log error, retry E2: WebServer fail → log nhưng tiếp tục
Điều Kiện Sau	WiFi AP phát sóng; WebServer sẵn sàng

Use Case 2: Quét RFID Thẻ Vào (IN Reader)

Trường	Nội Dung
--------	----------

Trường	Nội Dung
Tên Use Case	Quét RFID Thẻ Vào - IN Reader Phát Hiện Xe Vào
Mô Tả	Khi xe vào, RFID reader IN quét thẻ. UNO R4 đọc UID, gửi Backend (/api/cards/scan), nhận lệnh mở barrier IN.
Tác Nhân	Người lái xe, RFID Reader IN (RC522, SS=10), UNO R4, Backend
Mô Tả Chi Tiết	RFID reader IN (SS=10, RST=9). readRFID() đọc UID (e.g., "A1B2C3D4"). Cooldown 200ms. Gửi Backend HTTP POST /api/cards/scan. Backend xác thực, trả response. Nếu success, mở barrier IN. Nếu fail, log unknown card.
Điều Kiện Tiên Quyết	RFID reader IN kết nối; Card database ở Backend; WiFi AP hoạt động
Luồng Chính	<div>1. Loop() liên tục</div> <div>2. readRFID(rfidIn, "IN")</div> <div>3. Nếu có card:</div> <div>- Đọc UID hex</div> <div>- Kiểm tra cooldown 200ms</div> <div>4. sendRFIDToServer(uid, "IN")</div> <div>- POST /api/cards/scan</div> <div>- JSON: <code>{card_id: uid, direction: "IN"}</code></div> <div>5. Backend kiểm tra card</div> <div>6. Nếu success:</div> <div>- Trả action="entry"</div> <div>- UNO R4 gọi openBarrier(barrierIn)</div> <div>7. Nếu fail:</div> <div>- Backend add unknown_cards</div> <div>- Barrier không mở</div>
Luồng Thay Thế	A1: Manual control - Admin bấm nút, không RFID A2: License plate OCR - Dùng camera + OCR A3: QR code - Dùng QR code điện thoại
Luồng Ngoài Lề	E1: RFID reader lỗi → log error E2: Card không valid → barrier không mở E3: Backend offline → timeout, barrier không mở E4: Duplicate read → cooldown skip E5: Anti-passback → Backend từ chối
Điều Kiện Sau	RFID UID gửi Backend; Card status update; Barrier sẵn sàng mở

Use Case 3: State Machine Mở Barrier Vào (Opening Sequence)

Trường	Nội Dung
Tên Use Case	Mở Barrier Vào - Non-Blocking State Machine
Mô Tả	Sau RFID scan success, barrier IN thực hiện: IDLE → OPENING (servo 0→90) → WAITING_VEHICLE (detect xe) → VEHICLE_PRESENT (xe qua) → CLOSING (servo 90→0) → IDLE.
Tác Nhân	UNO R4, Servo Motor IN, Ultrasonic IN
Mô Tả Chi Tiết	Struct BarrierControl chứa state, servo, pins, timers. updateBarrier() transition state dựa thời gian & sensor. Non-blocking: không delay(), tất cả state update trong loop iteration.
Điều Kiện Tiên Quyết	RFID scan success; openBarrier(barrierIn) gọi; Servo IN config
Luồng Chính	1. openBarrier(barrierIn) gọi 2. servo.write(90°), state=OPENING 3. updateBarrier() mỗi iteration 4. elapsed = now - stateStartTime 5. Nếu elapsed >= 500ms (servo fully open): - state = WAITING_VEHICLE 6. Đọc ultrasonic, nếu detect 3x: - state = VEHICLE_PRESENT 7. Nếu ultrasonic không detect 3x: - state = CLOSING - servo.write(0°) 8. Nếu elapsed >= 500ms (servo closed): - state = IDLE 9. Timeout 30s → auto-close (safety)
Luồng Thay Thế	A1: Sync blocking - dùng delay() (không recommend) A2: Manual close - admin bấm nút
Luồng Ngoài Lề	E1: Servo fail → timeout auto-close E2: Ultrasonic fail → assume xe qua E3: Double scan → bỏ qua E4: Obstacle → servo cố force, safe
Điều Kiện Sau	Barrier IN mở; Xe vào; Barrier auto-close

Use Case 4: Phát Hiện Xe Vào (Ultrasonic Detection)

Trường	Nội Dung
--------	----------

Trường	Nội Dung
Tên Use Case	Phát Hiện Xe Vào - Ultrasonic Sensor HC-SR04
Mô Tả	Sau khi barrier mở, ultrasonic IN detect xe đi qua. Nếu distance < 10cm 3 lần liên tiếp, confirm xe. Timeout 30s auto-close.
Tác Nhân	UNO R4, Ultrasonic HC-SR04 (TRIG=3, ECHO=4)
Mô Tả Chi Tiết	readDistanceCM() phát TRIG pulse, đợi ECHO, tính distance. Read mỗi ~100ms. presentCount để chống nhiễu (phải 3x liên tiếp mới confirm).
Điều Kiện Tiên Quyết	Barrier state=WAITING_VEHICLE; Ultrasonic IN kết nối
Luồng Chính	<ol style="list-style-type: none"> 1. State = WAITING_VEHICLE 2. updateBarrier() gọi readDistanceCM() 3. Nếu distance < 10cm: <ul style="list-style-type: none"> - presentCount++ 4. Nếu distance >= 10cm: <ul style="list-style-type: none"> - absentCount++ 5. Nếu presentCount >= 3: <ul style="list-style-type: none"> - state = VEHICLE_PRESENT - Log "Vehicle confirmed" 6. Timeout 30s → TIMEOUT_CLOSING
Luồng Thay Thế	A1: Không ultrasonic - chỉ timeout auto-close A2: Laser sensor - tốt hơn ultrasonic A3: Pressure mat
Luồng Ngoài Lề	E1: Ultrasonic timeout → ghi 0 E2: Nhiễu → 3-count stability E3: Temperature drift → timeout vẫn work
Điều Kiện Sau	Xe confirm vào; State=VEHICLE_PRESENT; Chuẩn bị close

Use Case 5: Phát Hiện Xe Ra & Quét RFID OUT

Trường	Nội Dung
Tên Use Case	Quét RFID Thẻ Ra & Mở Barrier OUT
Mô Tả	Khi xe ra, RFID reader OUT quét thẻ. UNO R4 gửi Backend (/api/cards/scan, direction="OUT"). Backend verify, trả action="exit". Barrier OUT mở, xe qua, barrier auto-close.
Tác Nhân	Người lái xe, RFID Reader OUT (SS=7), UNO R4, Ultrasonic OUT, Backend

Trường	Nội Dung
Mô Tả Chi Tiết	Tương tự UC-2/3 nhưng OUT direction. RFID OUT (SS=7, RST=8). Post /api/cards/scan với direction="OUT". Backend check card status=1 (inside). Nếu ok, return action="exit". UNO R4 mở barrier OUT, detect xe, auto-close. Backend log exit, tính parking_duration.
Điều Kiện Tiên Quyết	Xe đã vào (status=1); RFID OUT hoạt động
Luồng Chính	<ol style="list-style-type: none"> 1. readRFID(rfidOut, "OUT") 2. Nếu card: <ul style="list-style-type: none"> - Post /api/cards/scan direction="OUT" 3. Backend check status=1 4. Nếu ok, trả action="exit" 5. openBarrier(barrierOut) 6. State machine: OPENING → WAITING_VEHICLE → VEHICLE_PRESENT → CLOSING → IDLE 7. Ultrasonic phát hiện xe qua 8. Backend log exit, status=0
Luồng Thay Thế	A1: Exit không RFID - ultrasonic auto-open A2: Exit LPR/OCR
Luồng Ngoài Lề	E1: Card inside không match → block exit E2: Anti-passback → Backend từ chối E3: Overstay → cảnh báo nhưng cho exit
Điều Kiện Sau	Xe ra; Barrier close; Backend update status=0

Use Case 6: Song Hành State Machine 2 Barrier (Parallel Processing)

Trường	Nội Dung
Tên Use Case	Quản Lý 2 Barrier Đồng Thời - Non-Blocking Parallel
Mô Tả	UNO R4 xử lý barrier IN & OUT song song. Khi IN OPENING, OUT có thể độc lập xử lý RFID OUT. 2 state machine chạy async, không block lẫn nhau.
Tác Nhân	UNO R4, 2 Servo, 2 Ultrasonic
Mô Tả Chi Tiết	Struct BarrierControl cho mỗi barrier. updateBarrier(barrier) cập nhật 1 barrier. Trong loop(), gọi updateBarrier(IN) rồi updateBarrier(OUT). Không delay(), single-threaded nhưng fast enough.

Trường	Nội Dung
Điều Kiện Tiên Quyết	2 barrier init; 2 servo attach; 2 ultrasonic config
Luồng Chính	<ol style="list-style-type: none"> 1. setup() init barrierIn & barrierOut 2. loop() chạy: <ul style="list-style-type: none"> - updateBarrier(barrierIn) - updateBarrier(barrierOut) 3. Nếu RFID IN scan: openBarrier(IN) state=OPENING 4. Cùng lúc, RFID OUT scan: openBarrier(OUT) state=OPENING 5. Cả 2 barrier run state machine song song 6. updateBarrier() tuần tự nhưng rất nhanh (<1ms mỗi cái) 7. Không need mutex (single-threaded, fast)
Luồng Thay Thế	A1: Sequential - xử lý 1 barrier tại 1 thời (chậm) A2: Threading - RTOS, mỗi barrier thread
Luồng Ngoài Lề	E1: Servo conflict - PCA9685 lib handle E2: Pin conflict → unlikely (design ok) E3: Stack overflow → unlikely
Điều Kiện Sau	2 barrier hoạt động song song; Vào/ra đồng thời

Use Case 7: WebServer Simple Health Check (Default Endpoint)

Trường	Nội Dung
Tên Use Case	WebServer Health Check - Default Response
Mô Tả	Khi client kết nối UNO R4 WebServer port 80 (default path "/"), UNO R4 trả simple JSON: <code>{status: ok, device: UNO R4 WiFi, ip: 192.168.4.2}</code> . Backend có thể dùng để kiểm tra UNO R4 online.
Tác Nhân	Backend, UNO R4 WebServer
Mô Tả Chi Tiết	UNO R4 WebServer handler (line 288-307) nhận HTTP request (bất kỳ path nào), trả JSON response. Dùng để quick health check mà không cần formal endpoint.
Điều Kiện Tiên Quyết	UNO R4 WebServer hoạt động

Trường	Nội Dung
Luồng Chính	<div>1. Backend (hoặc monitoring script) gọi GET http://192.168.4.2:80/<any_path></div> <div>2. UNO R4 WebServer handler trigger</div> <div>3. Đọc HTTP request (simple read)</div> <div>4. Trả response:</div> <div>
 HTTP/1.1 200 OK
 Content-Type: application/json
 {
 "status": "ok",
 "device": "UNO R4 WiFi",
 "ip": "192.168.4.2"
 }
</div> <div>5. client.stop()</div> <div>6. Quay lại loop</div>
Luồng Thay Thế	<div>A1: Formal endpoint - UNO R4 support /health path</div> <div>A2: Heartbeat ping - return simple "OK" text</div> <div>A3: WebSocket - push health status liên tục</div>
Luồng Ngoài Lề	<div>E1: WebServer hang → request timeout</div> <div>E2: Too many requests → WebServer slow</div>
Điều Kiện Sau	<div>Backend nhận health status; Biết UNO R4 online</div>

Kết Luận

UNO R4 WiFi quản lý cổng vào/ra với 2 RFID readers, 2 servo barriers, 2 ultrasonic sensors. State machine non-blocking xử lý 2 barrier song song. Giao tiếp Backend qua HTTP POST /api/cards/scan để xác thực card. WebServer simple health check để Backend monitor.