* Tiếp sau phần giới thiệu các thành phần, phần cứng sử dụng cho đề tài, em xin phép được đi chi tiết vào việc hiện thực từng thiết bị. Thiết bị đầu tiên là thiết bị đặt ở từng lot trong bãi xe, Lot Device, hay gọi là Lot node.
* Thành phần đầu tiên của lot node là module RF nRF24L01+. Module này được kết nối với arduino nano qua cái pin D7, D8, D11, D12 và D13, giao tiếp bằng giao thức SPI. Để sử dụng module này nhóm em sử dụng thư viện RF24 đã có sẵn, đồng thời nhóm tụi em tự hiện thực thêm 2 module là CRC24 để hiện thực cơ chế phát hiện lỗi trên đường truyền và RFUtil để cung cấp những functions và metadata cần thiết cho quá trình nhận và gửi payload.
* Thành phần kế tiếp của lot node chính là module từ kế HMC5883L. Module này được kết nối với Arduino nano bằng giao thức I2C, tức là thông qua pin A4 và A5. Nhóm chúng em tạo ra module code cùng tên HMC5883L để cung cấp cho nano function duy nhất là kiểm tra xem có xe ở gần hay không. Ở trong module code, tụi em sử dụng thư viện Wire được hỗ trợ sẵn để đơn giản hóa việc giao tiếp thông qua giao thức I2C.
* Thành phần tiếp theo chính là LED RGB. Ở đây tụi em dừng lại ở mô hình nên chỉ sử dụng 1 đèn LED duy nhất nhưng việc điều khiển LED dây cũng tương tự. Module tụi em tạo ra ở đây tên là RGBLED cung cấp những function hỗ trợ việc phối màu. LED được kết nối với nano thông qua 3 pin PWM là 3, 5, 6. Tuy nhiên, vì mục tiêu phát triển sau này có thể cần sử dụng LED có công suất lớn hơn nên tụi em ko nối trực tiếp led với 3 pin của nano mà thông qua module TIP122.
* Thành phần cuối cùng của lot node là Servo SG90. Pin D4 của nano kết nối trực tiếp pin data của servo, và được điều khiển thông qua thư viện Servo được hỗ trợ sẵn.
* Về phần luồng chạy của hệ thống, chúng em xin phép được tóm tắt ngắn gọn luồng xử lý dữ liệu như sau:
* Sau khi nano được cấp nguồn, sẽ thực hiện quá trình thiết lập các pin để điều khiển cho từng module tương ứng, và những module đặc thù như nRF24L01+ và HMC5883L có thêm 1 bước thiết lập để tiến vào hoạt động.
* Sau hi thực hiện xong bước thiết lập, tiến vào vòng lặp không giới han của quá trình hoạt động của thiết bị.
* Ở đầu mỗi vòng lặp, thiết bị sẽ thực hiện quá trình kiểm tra xem có xe đang đỗ tại lot đó hay không, và thay đổi những biến trạng thái nếu cần.
* Kế tiếp, sẽ kiểm tra xem vùng nhớ của module RF có một payload nào mới được gửi tới từ Hub hay không, nếu có thì sẽ thực hiện công việc như thông báo trạng thái, thay đổi trạng thái của led và servo… tùy thuộc vào payload nhận được.
* Và kết thúc 1 vòng lặp.
* Kế tiếp em xin được giới thiệu việc hiện thực thiết bị thứ 2, information device được đặt làm các bảng thông báo ở ngoài cổng bãi xe hoặc ở những ngã rẽ, còn được gọi là Sign node.
* Tương tự như lot node, ở sign node tụi em cũng hiện thực việc kết nối và sử dụng module nRF24L01+ thông qua giao thức SPI.
* Thành phần còn lại Sign node là các bảng đèn led 7 đoạn. Nhóm tụi em điều khiển những led 7 đoạn này thông qua module TPIC6B595 để giảm thiểu tối đa số pin cần thiết, và nhóm tụi em tạo module code tên là SEVEN\_SEGMENT để cung cấp các function cần thiết.
* Luồng chạy của sign node cũng tương tự với lot node nhưng tương đối đơn giản hơn
* Sau khi thiết bị được cấp nguồn, cũng tiến hành các bước thiết lập pin và module RF.
* Khi thực hiện xong bước cài đặt, tiến vào vòng lặp của thiết thì ở mỗi vòng lặp chỉ thực hiện việc kiểm sự tồn tại của payload và thực hiện việc cập nhật giá trị hiện ra trên led 7 đoạn.
* Thành phần kế tiếp trong hệ thống phần cứng chính là Raspberry Pi 3, đóng vai trò là Hub node của cả hệ thống.
* Tương tự với 2 node kia, ở trên Hub node tụi em cũng thực hiện việc kết nối raspberry pi 3 với module RF thông qua giao thức SPI. Ở phần hiện thực code thì cũng tương tự, sử dụng thư viện RF24 có sẵn với sự hỗ trợ của 2 module crc24 và rf util do tụi em viết. Đồng thời còn có thêm 2 class là Area và ParkingLot đóng vai trò là object lần lượt cho Sign node và Lot node
* Thành phần kế tiếp chính là các kết nối với server PubNub. Ở đây tụi em hiện thực code thông qua thư viện PubNub được server cung cấp, kết hợp với module pubnub meta do nhóm em viết để chứa những thông tin và function cần thiết cho việc giao tiếp với server và class UserPackage là object đại diện cho các thông tin gửi đến và đi Pi từ server.
* Và cuối cùng là module api service chứa những function để giao tiếp với Api server của nhóm tụi em.
* Về phần luồng chạy của chương trình trên hub, em xin phép được tóm tắt như sau
* Sau thiết bị được khởi động và chạy script, sẽ tiến hành thiết lập pin giao tiếp module RF, và những thiết lập cần thiết khác để kết nối server Pubnub.
* Sau bước thiết lập, chương trình chạy được tách ra thành 2 thread. 1 thread để xử lý việc poll dữ liệu từ các lot node, và thread còn lại để xử lý những request nhận được từ server Pubnub.
* Ở poll thread là vòng lặp vô hạn, mà ở mỗi bước lặp thì sẽ kiểm tra xem có request nào ở trên request thread hay không, nếu không có thì sẽ set flag để ra hiệu rằng đang thực hiện việc poll và tiến hành các bước cần thiết để lấy thông tin từ lot node cũng như cập nhật sign node nếu cần thiết.
* Ở request thread cũng là 1 vòng lặp vô hạn mà ở mỗi bước lặp sẽ thực hiện việc kiểm tra xem có nhận được request mới từ Pubnub hay không, nếu có thì sẽ set flag request để ra hiệu là đang có 1 request cần xử lý và chờ cho tới khi xong lượt poll đang chạy để thực hiện và xử lý request đó.
* Thành phần kế tiếp ở trong cả hệ thống của đề tài này là mobile application. Ứng dụng trên mobile này đóng vai trò như là UI để cả hệ thống giao tiếp được với người dùng.
* Tụi em tạo ứng dụng trên android, và sử dụng mô hình thuần túy nhất của một app android là MVC.
* Ở việc hiện thực ứng dụng android này, tụi em chia các java class vào từng package riêng lẻ đóng vai trò hỗ trợ lẫn nhau, gồm: activities, adapter, helper, interface, network và model.
* Những thành phần chức năng chính của ứng dụng, theo như hình ảnh trên slide lần lượt từ trái qua gồm: giao diện bản đồ cho người dùng tìm kiếm bãi xe, giao diện thông tin chi tiết bãi xe và giao diện quản lý bãi xe dành riêng cho manager.
* Sau khi trình bày khái quát về cách nhóm em hiện thực hệ thống phần cứng và ứng dụng, em xin được đi vào phần kết luận
* Về những mặt tốt thì trên phương diện phần cứng nhóm tụi em đã tạo nên thiết bị không dây dễ dàng lắp đặt, tương đối nhỏ gọn và sử dụng những module cũng khá thông dụng nên dễ bảo trì, thay thế và giá thành cũng tương đối thấp so với lợi nhuận bãi xe. Trên phương diện phần mềm thì giao diện mobile tụi em hiện thực tương đối thân thiện và trực quan với người dùng. Còn cả 2 server tụi em sử dụng và xây dựng đều dựa trên cơ sở dễ dàng mở rộng và phát triển ra khi lưu lượng người dùng gia tăng.
* Về những điểm còn yếu kém của đề tài thì trên phương diện phần cứng, vì tụi em chỉ dừng lại trên sản phẩm là mô hình nên lượng test mô phỏng của nhóm em còn khá kém, chưa thể tái hiện được giống hoặc gần giống như một bãi xe thật sự. Và ở trên điều kiện test của tụi em, do module RF sử dụng ở tần số 2.4GHz nên vẫn còn 1 vài trường hợp gây nhiễu cao, đặc biệt là ở những khu vực có số lượng người tạo nhiều hotspot wifi. Trên phương diện phần mềm thì các giải thuật xử lý và luồng dữ liệu tụi em xây dựng còn khá đơn giản, có thể đáp ứng được cho các bãi xe nhỏ và vừa, nhưng với các bãi xe lớn và chuyên nghiệp thì có thể sẽ sinh ra sự sai lệch về thông tin do độ chậm trễ của quá trình quét dữ liệu. Ứng dụng của tụi em hiện tại chỉ ở trên android, và những chức năng cung cấp cho manager còn tương đối đơn giản.
* Kế hoạch phát triển sắp tới của tụi em là sẽ khắc phục những điểm còn yếu kém, bao gồm: tìm cách cải thiện việc test, nếu có thể thì sẽ tìm cách xây dựng môi trường test giống thật hết mức trên các bãi xe thực tế. Cải thiện giải thuật truyền gửi dữ liệu thông qua RF để cải thiện phạm vi truyền và tốc độ truyền. Và cuối cùng là xây dựng ứng dụng đa nền tảng, cả mobile lẫn web để đem lại tiện lợi nhất cho người dùng.