**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

---------------o0o---------------



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**XÂY DỰNG GIẢI PHÁP TÍNH TIỀN NHANH TRONG SIÊU THỊ DỰA TRÊN MẠNG ZIGBEE**

|  |  |  |
| --- | --- | --- |
| **HỘI ĐỒNG:** | **KỸ THUẬT MÁY TÍNH** | |
| **GVHD:** | Vũ Tuấn Thanh | |
| **GVPB:** |  | |
| **SVTH:** |  |  |
|  | 1. Nguyễn Quang Vinh | 50702974 |
|  | 1. Nguyễn Văn Hải | 50700680 |
|  | 1. Phạm Hoàng Phúc | 50701839 |

TP. HỒ CHÍ MINH, Tháng 12/2011

LỜI CẢM ƠN

🙠🙢🕮🙠🙢

Sau 4 năm học tập chuyên ngành Khoa Học và Kỹ Thuật Máy Tính tại trường Đại Học Bách Khoa thành phố Hồ Chí Minh, được sự cho phép của nhà trường, nhóm em thực hiện báo cáo luận văn hoàn thành khóa học.

Nhóm em xin chân thành cảm ơn các thầy cô trong bộ môn Khoa Học và Kỹ Thuật Máy Tính, các anh chị hướng dẫn đã tận tình giúp đỡ và truyền đạt kiến thức để nhóm em có thể hoàn thành luận văn này.

Cảm ơn tất cả các bạn cùng khóa đã nhiệt tình chia sẻ kinh nghiệm và những kiến thức quý báo giúp nhóm em hoàn thành tốt luận văn tốt nghiệp của mình.

Mặc dù nhóm em đã cố gắng hoàn thành luận văn với tất cả sự nỗ lực của nhóm, nhưng luận văn chắc chắn không tránh khỏi những thiếu sót. Nhóm em kính mong quý thầy cô tận tình chỉ bảo.

Cuối cùng nhóm em xin gửi đến quý thầy cô lời chúc sức khỏe và lời cảm ơn chân thành nhất!

TP.HCM, tháng 12 năm 2011

Nhóm sinh viên thực hiện đề tài

LỜI CAM ĐOAN

Mục lục

[LỜI CẢM ƠN i](#_Toc311971698)

[LỜI CAM ĐOAN ii](#_Toc311971699)

[Mục lục iii](#_Toc311971700)

[Danh sách hình vẽ v](#_Toc311971701)

[Danh sách bảng biểu vi](#_Toc311971702)

[Danh sách các từ viết tắt vii](#_Toc311971703)

[1 GIỚI THIỆU ĐỀ TÀI 1](#_Toc311971704)

[1.1 Giới thiệu đề tài 1](#_Toc311971705)

[1.2 Mô hình Queue-busting 2](#_Toc311971706)

[1.2.1 Mô hình queue busting 2](#_Toc311971707)

[1.2.2 Nguyên tắc hoạt động 3](#_Toc311971708)

[1.2.3 Khả năng ứng dụng 3](#_Toc311971709)

[1.3 Kế hoạch 3](#_Toc311971710)

[Tìm hiểu mạng Zigbee và hardware sẽ hiện thực hệ thống : 3](#_Toc311971711)

[Làm bộ chuyển đổi điện áp RS232 3](#_Toc311971712)

[1.4 Công cụ hỗ trợ 4](#_Toc311971713)

[2 KIẾN THỨC NỀN TẢNG 5](#_Toc311971714)

[2.1 Tổng quan về mạng ZigBee 5](#_Toc311971715)

[2.1.1 Thị trường mà ZigBee nhắm tới. 5](#_Toc311971716)

[2.1.2 ZigBee là gì? 6](#_Toc311971717)

[2.1.3 Kiến trúc mạng cơ bản của ZigBee 10](#_Toc311971718)

[2.2 Application Layer 12](#_Toc311971719)

[2.3 Application Support Layer 12](#_Toc311971720)

[2.4 Network Layer 13](#_Toc311971721)

[2.4.1 ZigBee và IEEE 802.15.4 13](#_Toc311971722)

[2.4.2 Thiết lập, kết nối và kết nối lại các mạng ZigBee. 14](#_Toc311971723)

[2.4.3 Tái tham gia mạng 18](#_Toc311971724)

[2.4.4 Gán địa chỉ ZigBee 19](#_Toc311971725)

[2.4.5 Tìm đường cho packet trong ZigBee 22](#_Toc311971726)

[2.5 MAC và PHY Layer 23](#_Toc311971727)

[2.5.1 PHY Layer 23](#_Toc311971728)

[2.5.2 MAC Layer 23](#_Toc311971729)

[2.6 Giới thiệu Z-stack của Texas Instrument 23](#_Toc311971730)

[3 HIỆN THỰC 25](#_Toc311971731)

[3.1 Kiến trúc hệ thống 25](#_Toc311971732)

[3.1.1 Bộ KIT phát triển CC2530 của TI 27](#_Toc311971733)

[3.1.2 Datalogic Handheld corded scanner 32](#_Toc311971734)

[3.1.3 Bộ chuyển đổi điện áp RS232 32](#_Toc311971735)

[3.2 Thiết bị cashier 32](#_Toc311971736)

[3.2.1 Nguyên tắc hoạt động 32](#_Toc311971737)

[3.2.2 Hardware 33](#_Toc311971738)

[3.2.3 Software 33](#_Toc311971739)

[3.3 Thiết bị handheld 37](#_Toc311971740)

[3.3.1 Ý tưởng 37](#_Toc311971741)

[3.3.2 Mô hình 38](#_Toc311971742)

[3.3.3 Hiện thực 40](#_Toc311971743)

[3.4 Ứng dụng trên PC (personal computer) 41](#_Toc311971744)

[3.4.1 Mục đích 41](#_Toc311971745)

[3.4.2 Hiện thực 42](#_Toc311971746)

[3.4.3 Hoạt động của ứng dụng PC 42](#_Toc311971747)

[4 KẾT LUẬN 49](#_Toc311971748)

Danh sách hình vẽ

[Hình 1.2‑1: Khái niệm hệ thống Queue Busting 2](#_Toc311971749)

[Hình 2.1‑1: So sánh các kĩ thuật không giây 6](#_Toc311971750)

[Hình 2.1‑2 Kết nối mạng mesh trong ZigBee 7](#_Toc311971751)

[Hình 2.1‑3: Việc tìm lại đường đi trong mạng mesh ZigBee 8](#_Toc311971752)

[Hình 2.1‑4: Thời gian sử dụng Pin của ZigBee 9](#_Toc311971753)

[Hình 2.1‑5: Các mảng thị trường của ZigBee 10](#_Toc311971754)

[Hình 2.1‑6: Kiến trúc của ZigBee 11](#_Toc311971755)

[Hình 2.4‑1: Quá trình ZigBee tạo mạng 16](#_Toc311971756)

[Hình 2.4‑2: Quá trình ZigBee tham gia mạng 18](#_Toc311971757)

[Hình 2.4‑3: Gán địa chỉ Cskip trong cây đối xứng 21](#_Toc311971758)

[Hình 3.1‑1: Sơ đồ kết nối thiết bị của hệ thống 25](#_Toc311971759)

[Hình 3.1‑2: Sơ đồ phân lớp kiến trúc phần mềm 26](#_Toc311971760)

[Hình 3.1‑3: SmartRF05 Evaluation Board 27](#_Toc311971761)

[Hình 3.1‑4: SmartRF05 Battery Board 28](#_Toc311971762)

[Hình 3.1‑5: CC2530 Evaluation Modules 29](#_Toc311971763)

[Hình 3.1‑6: CC2531 USB Dongle 29](#_Toc311971764)

[Hình 3.1‑7: CC2530 Antenna 30](#_Toc311971765)

[Hình 3.1‑8: Kiến trúc CC2530 31](#_Toc311971766)

[Hình 3.22‑3.2‑1: Sơ đồ khối các chức năng 33](#_Toc311971767)

[Hình 3.2‑2: ??? 34](#_Toc311971768)

[Hình 3.2‑3: ??? 34](#_Toc311971769)

[Hình 3.2‑4: ??? 35](#_Toc311971770)

[Hình 3.2‑5: ??? 36](#_Toc311971771)

[Hình 3.2‑6: ??? 37](#_Toc311971772)

[Hình 3.3‑1: Các phần cứng cho hiện thực 38](#_Toc311971773)

[Hình 3.3‑2: Sơ đồ mô tả tổng quát task ứng dụng trong Handheld 39](#_Toc311971774)

[Hình 3.3‑3: Sơ đồ xử lý dữ liệu từ Scanner trong Handheld 39](#_Toc311971775)

[Hình 3.3‑4: Sơ đồ xử lý dữ liệu từ mạng ZigBee trong Handheld 40](#_Toc311971776)

[Hình 3.4‑1: Sơ đồ hiện thực ứng dụng trên PC 42](#_Toc311971777)

Danh sách bảng biểu

[Bảng 1.3‑1: Bảng kế hoạch công việc 4](#_Toc311971778)

[Bảng 2.1‑1: So sánh xu hướng ứng dụng giữa các giao thức wireless khác và ZigBee. 5](#_Toc311971779)

[Bảng 2.4‑1: Cskip được tính toán cho stack profile 0x01 20](#_Toc311971780)

[Bảng 2.4‑2: So sánh các phương tìm đường trong ZigBee 22](#_Toc311971781)

Danh sách các từ viết tắt

|  |  |
| --- | --- |
| ZC | ZigBee Coordinator |
| ZR | ZiBee Router |
| ZED | ZigBee End Device |
| EB | Evaluation Board |
| BB | Battery Board |
| PC | Personal Computer |
| CSMA-CA | Carrier Sense Multiple Access Collision Avoidance |
| O-QPSK | Offset-Quadrature Phase-Shift Keying |
| DSSS | Direct Sequence Spread Spectrum |
|  |  |

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu đề tài

Đề tài : Xây dựng giải pháp tính tiền nhanh trong siêu thị Queue Busting dựa trên mạng Zigbee.

Sự cần thiết của đề tài :

Ngày nay, khi nền kinh tế phát triển, đời sống con người ngày càng được nâng cao thì nhu cầu mua sắm của con người cũng tăng lên đáng kể. Các siêu thị mọc lên ngày càng nhiều nhưng vẫn không đủ để phục vụ. Bằng chứng là số lượng người chờ tính tiền tại các siêu thị luôn luôn đông đúc, con số này tăng lên rất nhanh vào những ngày lễ, ngày cuối tuần.

Bên cạnh đó, hệ thống tính tiền truyền thống tại các siêu thị, được cho là “nỗi ám ảnh” đối với những khách hàng đang đợi trong một hàng dài, không thể đáp ứng kịp, không tận dụng được thời gian đợi của khách hàng. Điều này làm cho khách hàng tốn rất nhiều thời gian để mua sắm, đồng thời làm giảm lợi nhuận của siêu thị, tăng áp lực cho các nhân viên tính tiền, đặc biệt là vào dịp cuối tuần, lễ, tết.

Do đó, giải pháp Queue Busting được đưa ra nhằm tận dụng thời gian chờ của khách hàng để giảm thời gian checkout tại quầy tính tiền. Để giải pháp có thể trở nên cơ động hơn, việc đưa mạng không dây vào mô hình là yêu cầu thiết thực.

Mục tiêu của đề tài :

* Hiểu được hoạt động của mô hình Queue Busting.
* Hiểu được các kiến thức về chuẩn 802.15.4.
* Xây dựng mô hình Queue Busting thử nghiệm dựa trên mạng Zigbee.

Phân tích đề tài:

Đề tài của chúng tôi là xây dựng giải pháp Queue Busting trên mạng không dây Zigbee nên việc tìm hiểu về mạng không dây Zigbee là ưu tiên hàng đầu. Tuy nhiên, mô hình Queue Busting đã được áp dụng vào thực tế nên việc tìm hiểu nguyên tắc hoạt động của mô hình này cũng rất cần thiết. Chúng ta có thể tóm tắt lại những vấn đề chúng ta cần giải quyết như sau:

* Tìm hiểu mô hình Queue Busting thực tế.
* Nắm vững các kiến thức về mạng Zigbee
* Hiện thực giải pháp trên theo chuẩn mạng Zigbee
* Kiểm tra hoạt động và khắc phục các lỗi có thể xãy ra trong quá trình hoạt động.
* Hiện thực thêm các phương thức bảo mật cho hệ thống.

## Mô hình Queue-busting

### Mô hình queue busting

Ngày nay, việc xếp hàng để tính tiền rất bất tiện và mất thời gian đối với khách hàng cũng như đối với siêu thị hoặc cửa hàng. Việc này làm cho các cửa hàng, siêu thị mất một lượng khách hàng và thu nhập khá lớn, cũng như làm cho khách hàng tốn nhiều thời gian không cần thiết.

Do đó, chúng ta xây dựng một hệ thống giúp quét hàng hóa của khách hàng nhanh hơn để việc thanh toán diễn ra mau chóng, giảm thời gian xếp hàng. Giúp khách hàng tiết kiệm thời gian khi mua sắm. Giúp cửa hàng, siêu thị tăng số lượng khách hàng mua sắm trong ngày → tăng doanh thu.

Hệ thống như thế, được gọi là Queue Busting :

+ Queue Busting giải quyết vấn đề xếp hàng dựa trên nguyên tắc quét hàng hóa (mã vạch của hàng hóa) của khách hàng trước khi họ đến quầy tính tiền. Do đó, giảm thời gian quét hàng hóa đối với nhân viên thu ngân, giúp cho việc thanh toán được diễn ra nhanh hơn.

+ Để việc quét hàng hóa trở nên nhanh hơn, hệ thống queue busting cần phải làm việc được với các thiết bị quét mã vạch cầm tay và cố định đặt ở quầy thu ngân. Các thiết bị này, giao tiếp với nhau thông qua mạng không dây, để thiết bị cầm tay có thể hoạt động ở phạm vi rộng khi mà hàng đã quá dài. Hệ thống queue busting được mô tả như hình sau :



Hình 1.2‑1: Khái niệm hệ thống Queue Busting

### Nguyên tắc hoạt động

Khi có rất nhiều khách hàng đứng xếp hàng, một vào nhân viên sẽ cầm máy quét mã vạch cầm tay, quét hàng hóa của từng khách hàng. Sau khi quét cho một khách hàng xong, nhân viên sẽ quét tiếp một mã số định danh dành riêng cho khách hàng đó. Tất cả thông tin về hàng hóa sẽ được lưu trữ trong máy quét mã vạch cầm tay này.

Khi đến quầy tính tiền, khách hàng chỉ phải đưa cho nhân viên thu ngân mã số định danh trên. Tất cả thông tin về hàng hóa của khách hàng sẽ được truyền về cho máy tính của nhân viên thu ngân. Việc thanh toán sẽ được diễn ra rất nhanh chóng.

### Khả năng ứng dụng

Hiện tại, hệ thống queue busting đã được triển khai trên một số của hàng, siêu thị trên thế giới. Tuy nhiên, hệ thống chỉ dừng lại ở mức 1-1, tức là một thiết bị quét mã vạch cầm tay kết nối với một thiết bị quét mã vạch cố định ở quầy thu ngân. Do nhu cầu mở rộng, tất cả các thiết bị quét mã vạch cầm tay (handheld scanner) có thể kết nối với tất cả thiết bị quét mã vạch cố định ở quầy thu ngân (point of sale – POS), và tiết kiệm chi phí, nên giải pháp queue busting dựa trên mạng không dây Zigbee được đưa ra.

## Kế hoạch

|  |  |  |  |
| --- | --- | --- | --- |
| STT | DETAILS | DEADLINE | ASSIGN |
| 1 | Tìm hiểu mạng Zigbee và hardware sẽ hiện thực hệ thống :  Application layer và Network layer của mạng Zigbee  Datasheet và các cấu hình cho 2 board EB và BB | 01/11 | Team |
| 1 | Làm bộ chuyển đổi điện áp RS232 | 01/11 | Hải |
| 2 | Xây dựng trên Evaluation Board (EB):  1 module để nhận data từ barcode scanner.  1 module truyền nhận data với PC.  1 databases mã khách hàng | 11/11 | Vinh |
| 3 | Xây dựng trên Battery Board (BB):  1 module đọc, ghi, xóa datacho Flash.  1 module nhận data từ barcode scanner | 11/11 | Hải |
| 4 | Xây dựng trên PC :  1 chương trình thanh toán hóa đơn.  1 databases mã hàng hóa và đơn giá. | 11/11 | Phúc |
| 5 | Alpha release :  Kết hợp các module trên EB và BB lại để tạo thành 1 bộ code hoàn chỉnh.  Hoàn thành chức năng cơ bản nhất của hệ thống | 18/11 | Team |
| 6 | Alpha test :  Kiểm tra chức năng cơ bản của toàn bộ hệ thống từ quét hàng hóa đến lúc thanh toán  Sửa chữa các lỗi xảy ra nếu có | 25/11 | Team |
| 7 | Beta release :  Thêm các tính năng cần thiết khác : báo lỗi, xử lý lỗi có thể xảy ra trong quá trình hoạt động …  Thêm tính năng bảo mật cho hệ thống | 05/12 | Team |
| 8 | Beta test :  Kiểm tra tất cả cách tính năng của hệ thống  Sửa chữa các lỗi xảy ra nếu có | 10/12 | Team |

Bảng 1.3‑1: Bảng kế hoạch công việc

## Công cụ hỗ trợ

* IAR Embedded Workbech for 8051 8.10 : công cụ lập trình và kiểm tra lỗi của toàn bộ luận văn.
* Barcode studio : hỗ trợ việc tạo ra các barcode cho việc kiểm tra hoạt động của hệ thống
* ComTestSerial 3.0.0.103 : hỗ trợ việc kiểm tra hoạt động truyền nhận dữ liệu giữa các thiết bị thông qua cổng COM
* Eclipse (Java) : hiện thực chương trình thanh toán trên PC

# KIẾN THỨC NỀN TẢNG

## Tổng quan về mạng ZigBee

### Thị trường mà ZigBee nhắm tới.

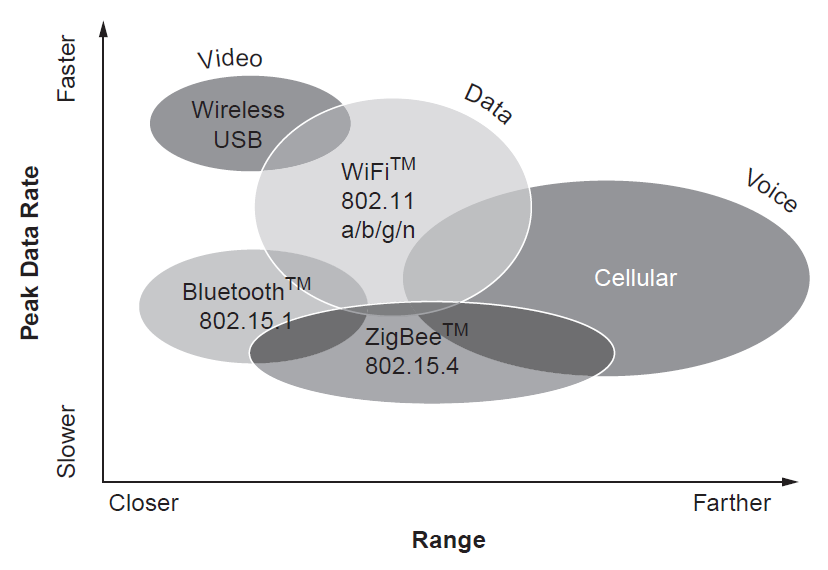
Chuẩn mạng ZigBee wireless phù hợp với nhu cầu thị trường mà các kĩ thuật wireless khác không thể đáp ứng (xem ).

|  |  |  |
| --- | --- | --- |
| Xu hướng | Các giao thức wireless khác | ZigBee |
| Tốc độ | Nhanh hơn | Truyền dữ liệu thấp |
| Tinh năng | Nhiều tính năng | Nhắm tới một stack nhỏ |
| Nhu cầu | Media độ nét cáo | Điều khiển thiết bị |
| Thời gian sử dụng Pin | Vài giờ, vài ngày | Hàng năm |
| Vòng đời sản phẩm | 1 đến 2 năm | Hàng thập kĩ |

Bảng 2.1‑1: So sánh xu hướng ứng dụng giữa các giao thức wireless khác và ZigBee.

Trong khi hầu hết các chuẩn wireless hướng tới tốc độ nhanh hơn thì ZigBee nhắm tới tốc độ truyền data thấp. Trong khi các giao thức wireless khác thêm càng nhiều tính năng thì ZigBee nhắm tới một tiny stack mà phù hợp với các vi điều khiển 8-bit. Trong khi các kĩ thuật wireless khác hướng tới cung cấp truyền data tới Internet hay phấn phối dòng media độ nét cao (high-definition) thì ZigBee hướng tới điều khiển đèn hoặc gửi dữ liệu nhiệt độ từ các cảm biến. Trong khi các kĩ thuật wireless khác được thiết kế để chạy trong vài giờ hoặc có thể vài ngày bằng pin thì ZigBee chạy tới hang năm. Trong khi các kĩ thuật wireless khác cung cấp 12 đến 24 tháng vòng đời cho một sản phẩm thì các sản phẩm ZigBee có thể dùng trong hàng thập kĩ hoặc hơn trong các ứng dụng đặc trưng.

Các dịch vụ mà ZigBee hướng tới là việc kết nối cảm biến không dây và điều khiển hay đơn giản là điều khiển không dây. Thật ra, slogan cho ZigBeee là “Wireless Control That Simply Works”.



Hình 2.1‑1: So sánh các kĩ thuật không giây

Thị trường điều khiển không dây các nhiều yêu cầu mà chỉ có ZigBee mới phù hợp:

* Tin cậy cao
* Chi phí thấp
* Năng lượng cần rất thấp
* Bảo mật cao
* Một chuẩn mở

Để có được năng lượng tiêu thụ thấp và chi phí thấp, ZigBee đã thêm một ràng buộc kĩ thuật là tốc độ truyền thấp.

### ZigBee là gì?

ZigBee có độ tin cậy cao.

Sự truyền thông không dây là không tin cậy. Chứng minh điều này bằng việc đi long vòng với điện thoại di động, sau đó bước vào thang mày. Bất cứ ai sử dụng điện thoại đều gặp sự cố cuộc gọi bị ngắt hoặc đường truyền yếu. Tất cả bởi vì song radio cũng chỉ là các song. Chúng chạy qua các vật cản, có thể bị chặn bởi kim loại, nước hoặc khối bê tong và phụ thuộc vào nhiều yếu tố phức tạp gồm thiết kế ăng-ten, sự khoách đại năng lượng, và thậm chí các điều kiện thời tiết.

Tuy nhiên, điều khiển không dây thường không có cùng vấn đềnhư trong một cuộc điện thoại , việc di chuyển để tìm điểm nhận song tốt hơn hay việc đợi để cố quay lại sau. Hiệp hội ZigBee hiểu điều này và vậy sự đặc tả ZigBee thể hiện điều này. ZigBee giành khả năng tin cậy cao trong nhiều cách:

* IEEE 802.15.4 với O-QPSK và DSSS
* CSMA-CA
* 16-bit CRCs
* Acknowledgment tại mỗi hop (chặng)
* Việc nối mạng lưới (mesh) để tìm ra đường đi tin cậy
* End-to-end acknowledgments để kiểm tra dữ liệu đến đích

Điều đầu tiên là dựa vào một kĩ thuật wireless rất tin cậy, khoảng cách thấp, sự đặc tả IEEE 802.15.4. Đặc tả này là một kĩ thuật radio mạnh, rất hiện đại được xây dựng trên 40 năm kinh nghiệm của IEEE. Nó dùng những gì được gọi là Offset-Quadrature Phase-Shift Keying (O-QPSK) và Direct Sequence Spread Spectrum (DSSS), một sự kết hợp của các kĩ thuật mà cung cấp hiệu suất tuyệt vời trong các môi trường tỉ lệ signal-to-noise (tín hiệu trên nhiễu) thấp.

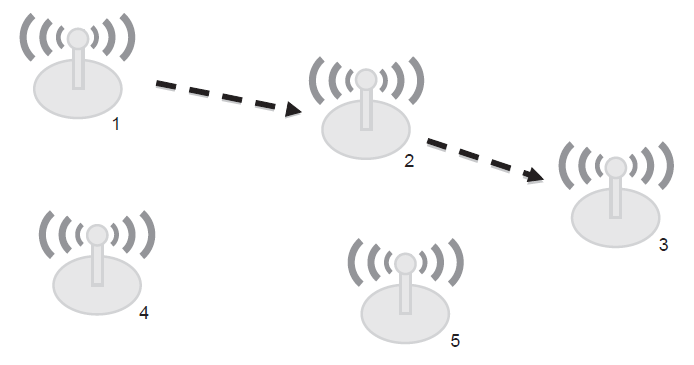
ZigBee dùng cái gọi là “Carrier Sense Multiple Access Collision Avoidance” (CSMA-CA) để tang khả năng tin cậy. Trước khi truyền, ZigBee lắng nghe kênh truyền. Khi kênh truyền trống, ZigBee bắt đầu truyền. Điều này ngăn các radio khỏi việc gây ra xung đột dữ liệu. CSMA-CA tương tự những gì con người làm trong các hội thoại. Chúng ta chờ người khác nói xong mới nói.

ZigBee dùng 16-bit CRC trên mỗi gói (packet), được gọi là một Fame Checksum (FCS). Điều này đảm bảo các bit dữ liệu chính xác.

Mỗi packet được thử lại 3 lần (trong toàn bộ 4 lần truyền). Nếu packet không thể truyền qua sau lần truyền thứ tư, thì ZigBee thông báo node gửi một vài điều có thể bị về việc truyền này.

Một cách khác mà ZigBee có được khả năng tin cậy là kết nối mạng lưới (mesh). Mạng lưới một cách cơ bản cung cấp 3 khả năng tang cường cho một mạng wireless: mở rộng khoảng cách bằng multi-hop, việc tạo mạng ad-hoc, và quan trọng hầu hết là tìm đường đi tự động và tự phục hồi.

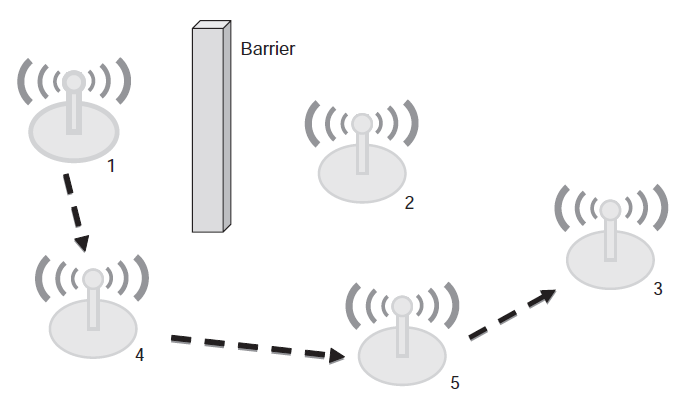
Với mạng lưới, dữ liệu từ node đầu tiên có thể đến bất cứ node nào khác trong mạng ZigBee, đánh giá khoảng cách bằng các radio để gửi message (xem ).



Hình 2.1‑2 Kết nối mạng mesh trong ZigBee

Node 1 muốn giao tiếp với node 3, nhưng nó ra khỏi vùng phủ song của node 3. ZigBee tự động tìm ra đường tốt nhất và node 1 sẽ gửi thong tin cho node 2, rồi truyền tiếp đến node 3.

Bây giờ giả sử rằng, có vài thứ xảy ra đến đường đi này. Có thể node 2 hoàn toàn bị loại bỏ hoặc chết hay vài vật cản như một bức tường bê-tông hoặc một thùng nước lớn. Điều này không hề gì với ZigBee. ZigBee sẽ tự động phát hiện sự thất bại của đường đi và đi vòng (xem Hình 2.1‑3).



Hình 2.1‑3: Việc tìm lại đường đi trong mạng mesh ZigBee

Thêm vào mạng lưới, ZigBee cung cấp việc broadcasting tin cậy, một kĩ thuật cho việc phân phối một message đến nhiều node trong mạng. ZigBee cũng cung cấp multicasting có thể gửi một message đến bất kì group các node. Và như một kĩ thuật tìm đường back-up, ZigBee cung cấp tìm đường cây (tree routing) để augment mạng lưới ZigBee trong các hệ thống giới hạn RAM.

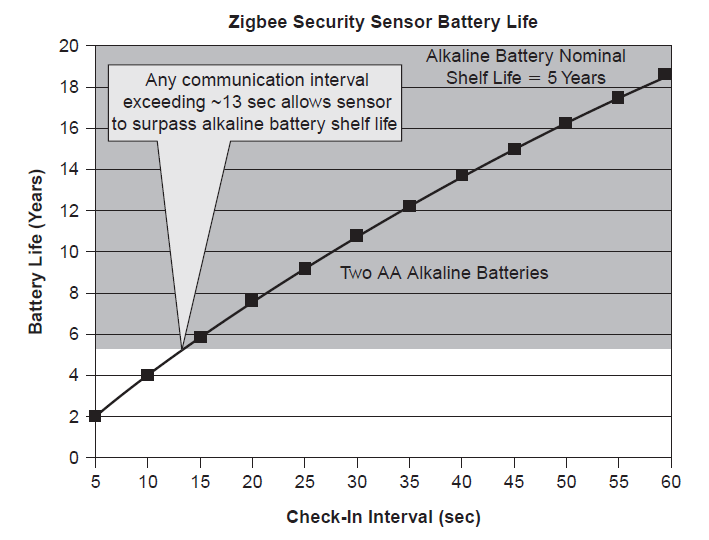
ZigBee cũng cung cấp tự động end-to-end acknowledgments. Ứng dụng có thể biết một gói cụ thể được nhận bởi node khác hay không. Với tất cả các retries, ZigBee lọc ra các gói trùng, nên ứng dụng không cần bận tâm.

*ZigBee có chi phí thấp*

Nhiều nhà cung cấp stack và silicon, các module ZigBee và nhiều tài nguyên phân phối với chi phí phát triển thấp cho các thiết bị ZigBee.

*ZigBee sử dụng năng lượng thấp*

Các thiết bị trong một mạng ZigBee có thể chạy trong nhiều năm chỉ 1 cặp pin AA phụ thuộc ứng dụng.



Hình 2.1‑4: Thời gian sử dụng Pin của ZigBee

ZigBee bảo mật cao

Cho việc bảo mật mạng, ZigBee dùng National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES). Chuẩn này, AES-128, là một mã hóa khối (block cipher) mà mã hóa và giải mã các packets trong một phương thức khó để bẽ khóa. Đây là một trong những chuẩn nổi tiếng. Nguyên nhân mà nó được dùng bởi ZigBee là:

* Chuẩn được xác thực quốc tế.
* Miến phí …
* Có thể hiện thực trên một vi điều khiển 8-bit

*ZigBee là một chuẩn mở toàn bộ*

Nhiều nhà cung cấp ZigBee stack, silicon và các giải pháp ứng dụng.

Đặc tả ZigBee có thể được tải miễn phí từ <http://www.zigbee.org>

*ZigBee có tốc độ dữ liệu thấp*

Để có được chi phí thấp và năng lượng tiêu hao thấp và việc xem xét không gian và thị trường ứng dụng mà ZigBee nhắm tới, hiệp hội ZigBee đã quyết định giữ giao thức trong một môi trường tốc độ truyền dữ liệu thấp.

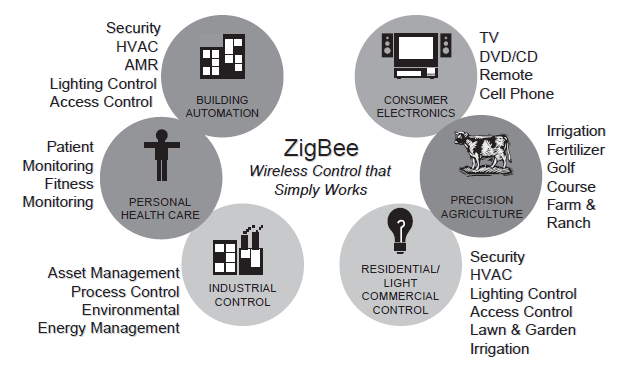
ZigBee nằm trên các IEEE 802.15.4 transceivers, trong không gian 2.4GHz truyền thong tại 250kbps, nhưng do số lần retries, sự mã hóa và giải mã, và giao thức lưới đầy đủ được dùng nên through-put thực sự khoảng 25kbps.

Transceivers là half-duplex, đây cũng là một yếu tố giảm through-put từ 250 tới 25kbps.

*Các ứng dụng sử dụng ZigBee*

* Home Automation
* Commercial Building Automation
* Industrial Plant Monitoring
* Telecommunication Applications
* Automatic Metering Initiative
* Personal Home and Health Care

ZigBee xuất hiện ở nhiều thị trường gồm nhà, thương mại, công nghiệp tự động, y tế và các dịch vụ local-based.



Hình 2.1‑5: Các mảng thị trường của ZigBee

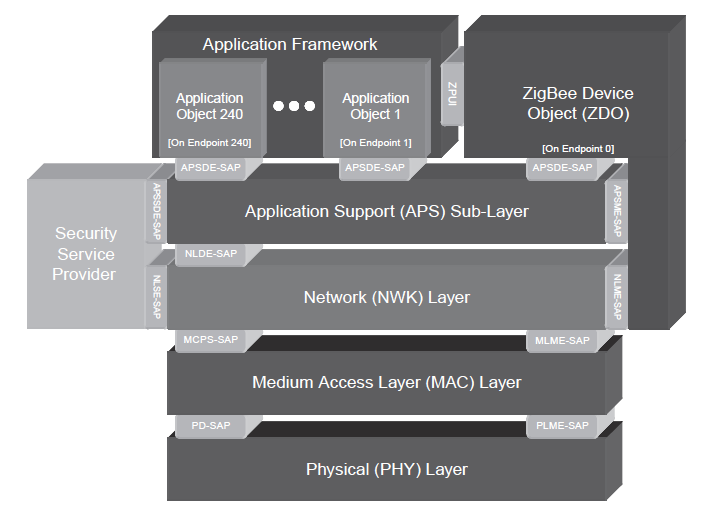
### Kiến trúc mạng cơ bản của ZigBee

Kiến trúc mạng ZigBee chia làm 3 phần chính:

* Ứng dụng
* ZigBee stack
* Radio

Và được chia thành các lớp, mỗi lớp không biết gì về lớp trên nó. Lớp trên được xem như một “master” mà gửi yêu cầu cho “slave” bên dưới làm.

ZigBee không giống mô hình mạng OSI 7-layer, nhưng nó có vài thành phần giống gồm PHY (physical), MAC (link layer), NWK (network). Các lớp 4-7 (transport, session , presentation và application) được chuyển qua các lớp APS (APplication support) và ZDO (ZigBee Device Object) trong mô hình ZigBee



Hình 2.1‑6: Kiến trúc của ZigBee

Giữa các lớp là “Service Access Points”(SAPs). SAPs cung cấp API tách biệt bên trong lớp khỏi các lớp trên và bên dưới. Giống như đặc tả IEEE 802.15.4, ZigBee dùng 2 SAPs cho mỗi lớp, một cho dữ liệu và một cho sự quản lý.Ví dụ, tất cả các sự truyền thong dữ liệu đến và từ lớp network đi qua “Network Layer Data Entity Service Access Point” (NLDE-SAP). Các yêu cầu trong đặc tả ZigBee giống như APSDE-DATA.request. Một yêu cầu gửi dữ liệu ra radio nhưng chỉ được khởi tạo ở lớp APS.

Hai lớp thấp nhất, MAC và PHY được định nghĩa bởi đặc tả IEEE 802.15.4. Lớp PHY đơn giản dịch các packet thành các over-the-air bits và ngược lại. Lớp MAC cung cấp khái niệm của một network, gồm một PAN ID, và kết nối thông qua các beacon requests và reponses. Nó cũng cung cấp các per-hop acknowledgment và một vài lệnh cho việc tham gia và tạo một mạng. Lớp MAC không có multi-hop hay mesh.

Lớp NWK có trách nhiệm cho hình thành mạng mesh, gồm broadcasting các packets qua mạng, xác định các đường đi cho các unicasting packets, và đảm bảo các packets được gửi một cách tin cậy từ một node đến node khác. Lớp network cũng có một tập các lệnh cho mục đích bảo mật, gồm bảo mật tham gia và tái tham gia mạng. Tất cả các mạng ZigBee được bảo mật ở lớp NWK, và toàn bộ payload của NWK frame được mã hóa.

Lớp APS có trách nhiệm cho ứng dụng. Nó hoạt động như một bộ lọc cho ứng dụng chạy phía trên nó các endpoints đơn giản là logic trong các ứng dụng này. Nó hiểu những gì các clusters và endpoints đưa ra, và kiểm tra xem endpoint là một thành viên của Application Profile và group trước khi gửi message lên trên. Lớp APS cũng lọc các message trùng mà hoàn toàn được gửi lên bởi lớp NWK. Lớp APS giữ một bảng local binding, một bảng chỉ các nodes hoặc các groups trong network mà node muốn giao tiếp đến.

Lớp ZDO (bao gồm ZigBee Device Profile, ZDP) có trách nhiệm cho quản lý cục bộ và over-the-air của network. Nó cung cấp các dịch vụ để khám phá các nodes khác và các dịch vụ trong network, và có trách nhiệm trực tiếp cho trạng thái hiện tại của node trên network.

Application Framework chứa ZigBee Cluster Library và cung cấp một framework mà các ứng dụng chạy bên trong. Các endpoints là cơ chế được sử dụng tách biệt một ứng dụng khỏi các ứng dụng khác.

Các dịch vụ bảo mật được dùng bởi nhiều lớp, và có thể được dùng bởi các lớp ZDO, APS, or NWK, do đó nó nằm ở cạnh(xem ).

Tất cả các layers có cái được gọi là một thông tin cơ bản. Tại lớp MAC, được gọi là một PAN information Base (PIB). Tại lớp network được gọi là Network Information Base (NIB), và tất nhiên AIB cho lớp APS. Tất cả “information base” nghĩa là các cài đặt của lớp đó. Bao nhiêu retries được yêu cầu? PAN ID hay địa chỉ network hiện giờ của một node cụ thể là gì? Các trường này trong “information base” nhìn chung được cài đặt bởi các lớp cao hơn hoặc thông qua việc dùng các câu lệnh quản lý thông qua các management SAPs.

Chú ý không có gì ở đây về sự tương tác với bất kì phần cứng trong một thiết bị ZigBee hơn là radio. Không có gì nói về giao tiếp LEDs, LCD, speaker, GPIO ports, bộ nhớ non-volatile hoặc flash. ZigBee chỉ quan tâm tới giao thức mạng và hành vi over-the-air. ZigBee kiểm tra phù hợp yếu tố này. Khi tất cả các message over-the-air có thể được hiểu một cách chính xác bởi bất kì node ZigBee khác, cho phép các nhà cung cấp cải tiến trong khi vẫn cung cấp khả năng tương thích hoàn toàn giữa các nhà cung cấp.

## Application Layer

## Application Support Layer

## Network Layer

### ZigBee và IEEE 802.15.4

Tên “ZigBee” và 802.15.4 thường được dùng qua lại lẫn nhau nhưng nó không giống nhau.

Đặc tả 802.15.4 được tạo và hỗ trợ bởi IEEE. Đặc tả này định nghĩ các lớp vật lý và MAC cho một mạng không dây, cá nhân, low-power.

IEEE 802.15.4 định nghĩa:

* Các cơ chế cho việc tìm mạng
* Các cơ chế cho việc thiết lập và kết nối vào mạng
* Các các cơ chế cho thay đổi kênh truyền
* Các cơ chế việc phát hiện vật cản và nhiễu trên một kênh
* Phương pháp phấn phối data-packet, single-hop, ACK, việc dung CSMA-CA để tránh đụng độ
* Phương pháp data-broadcast, single-hop, không có ACK

IEEE 802.15.4 không định nghĩa về truyền thông multi-hop, gán địa chỉ, hay khả năng tương tác với mức ứng dụng.

Nếu mạng xây dựng chỉ với một hop, thì 802.15.4 MAC/PHY có thể làm tất cả những gì cần. Các nhà cung cấp hỗ trợ một môi trường ứng dụng 802.15.4 mà không yêu cầu ZigBee.

Ngoài ZigBee, các nhiều protocol network hoàn toàn được xây dựng trên tiêu chuẩn 802.15.4, một số là dạng mesh hay multi-hop, một số là single-hop hay mạng star.

Nhưng ZigBee là protocol chính mà được xây dựng trên tiêu chuẩn 802.15.4, thêm vào một lớp network có khả năng của mạng mesh, peer-to-peer, multi-hop; một lớp bảo mật có khả năng xử lý các trường hợp bảo mật phức tạp, và một lớp ứng dụng cho các profile ứng dụng có thể tương thích với nhau.

Trong sơ đồ kiến trúc ZigBee chuẩn, các lớp MAC và PHY thể hiện dưới IEEE, trong khi phần còn lại thể hiện dưới ZigBee.

Công việc của lớp MAC là chuyển các packet từ chuỗi byte thành phổ RF và ngược lại. Lớp MAC cho phép một mạng được thiết lập, các kênh được chia sẽ, và dữ liệu được truyền (single-hop) trong một cách thức tinh cậy, hợp lý.

ZigBee đặc tả tất cả các lớp trên MAC và PHY, gồm NWK(network), APS, ZDO và các lớp bảo mật. ZigBee cung cấp mạng mesh và các khả năng multi-hop, tang cường khả năng tin cậy của phân phối packet dữ liệu, và đặc tả tương thích giữa các ứng dụng.

ZigBee không dung tất cả sự đặc tả 802.15.4 MAC/PHY; chỉ một phần nhỏ. Điều này cho phép các nhà cung cấp stack có các giải pháp nhỏ hơn (dung ít RAM và flash) bằng cách cung cấp một lớp MAC giới hạn cho ZigBee stack của họ. Ví dụ, ZigBee không dùng các phương pháp 802.15.4 beaconing, hay khe thời gian được đảm bảo. ZigBee là bất đồng bộ. Bất kỳ node nào có thể truyền tại bất cứ lúc nào. Chỉ CSMA-CA (carrier-sense multiple-access with collision avoidance), một cơ chế mức MAC, để ngăn các node khỏi truyền đè lên nhau.

ZigBee cũng có một số điều chỉnh trong 802.15.4 chuẩn. Một trong số đó là mô hình bảo mật. MAC định nghĩa một thứ gọi mà CCM, được viết tắt từ “counter-mode cipher-block chaining-message authentication code”. CCM yêu cầu bảo mật khác nhau cho mối lớp. Do rạng buộc hiệu suất trên các vi xử lý nhỏ, ZigBee không làm vậy. Mô hình bảo mật ZigBee cũng được gọi là CCM ( một điều chỉnh nhỏ của bảo mật MAC CCM).

Một trong các phần thú vị hơn mà ở đó ZigBee khác với sự đặc tả 802.15.4 là time-out cho các beacon response. Với cách này, các beacon request và các beacon response không làm gì khi network là một beaconing network hay không.

Một beacon đơn giản là một packet chứa thông tin về node và network. Được dùng trong ZigBee để tìm các network. Trong các mạng với hơn 30 nodes trong cùng vùng nghe, các timeout 802.15.4 mặc định cho các phản hồi các beacon request không cho phép đủ thời gian cho tất cả các node phản hồi. Đặc tả 802.15.4 đã không được xây dựng với các mạng lớn, nhưng ZigBee thì có thể.

Đặc tả 802.15.4 MAC hoàn toàn ổn định từ tháng 11 năm 2003. Một số nhà cung cấp chip thậm chí cung cấp 802.15.4 MAC trong ROM. Nhưng IEEE vẫn chưa dừng tại đó. Năm 2006, IEEE ra một đặc tả 802.15.4 khác được gọi là 802.15.4-2006.

Thay đổi lớn nhất trong IEEE 802.15.4-2006 là một PHY tốt hơn cho các radio dưới 1GHz. Trong đặc tả 802.15.4-2003, 868MHz và 900 MHz bị giới hạn 20kbps và 40kbps. Tốc độ truyền dữ liệu tại tần số dưới 1GHz quá chậm cho ZigBee; radio cho ZigBee là 2.4GHz, hoạt động với 250kbps. IEEE 802.15.4-2006 đã thay đổi tất cả. Đặc tả này đã thêm 2 PHY tùy chọn mới cho tần số dưới 1GHz cho phép truyền lên tới 250kbps.

### Thiết lập, kết nối và kết nối lại các mạng ZigBee.

Trước khi bất cứ các node ZigBee có thể giao tiếp trên một mạng, nó phải thiết lập một mạng mới hay kết nối vào một mạng đang tồn tại. Chỉ ZigBee Coordinator có thể thiết lập một mạng. Chỉ ZigBee Routers và ZigBee End-Devices có thể kết nối vào mạng. Nhiều nhà cung cấp stack cung cấp khả năng để một node được chỉ định như một ZC, ZRm ZED tại lúc biên dịch (tiết kiệm code và RAM) hay tại lúc chạy (giảm các phần OEM-manufactured)

Mỗi node bắt đầu với một địa chỉ 64-bit IEEE (MAC), được gán bởi OEM trong sản xuất. Trong quá trình kết nối mạng, mỗi node được gán một địa chỉ 16-bit short duy nhất (NwkAddr) để dùng khi giao tiếp với các node khác qua mạng. Địa chỉ 16-bit thường dùng cho các giao tiếp gần, để giảm overhead trong over-the-air protocol.

**Thiết lập mạng**

ZigBee Coordinator thiết lập mạng. Quá trình thiết lập một mạng là xác định một định danh duy nhất cho mạng, được gọi là PAN ID, và chọn một trong 16 kênh 802.15.4 (11-16) để điều hành mạng.

Trong suốt quá trình thiết lập mạng, một gói đơn được gửi over-the-air trên mỗi kênh: một MAC active scan (hay beacon request).

Một ZigBee Coordinator có nhiệm vụ sau:

* Thiết lập mạng
* Thiết lập kênh 802.15.4 trên mạng sẽ hoạt động
* Thiết lập extended và short PAN ID cho mạng
* Quyết định stack profile để dùng
* Hoạt động nhứ Trust Center cho các ứng dụng bảo mật và mạng
* Hoạt động như người đứng giữa cho việc End-Device-Bind
* Hoạt động như một router trong mesh routing
* Hoạt động như là gốc của tree, nếu tree routing được sử dụng

ZigBee Coordinator thực sự chỉ là một router nếu network không được ủy thác. Và có nhiều cách để thay thế ZigBee Coordinator sau khi một network chạy nếu thiết bị ZC trục trặc do một số nguyên nhân.

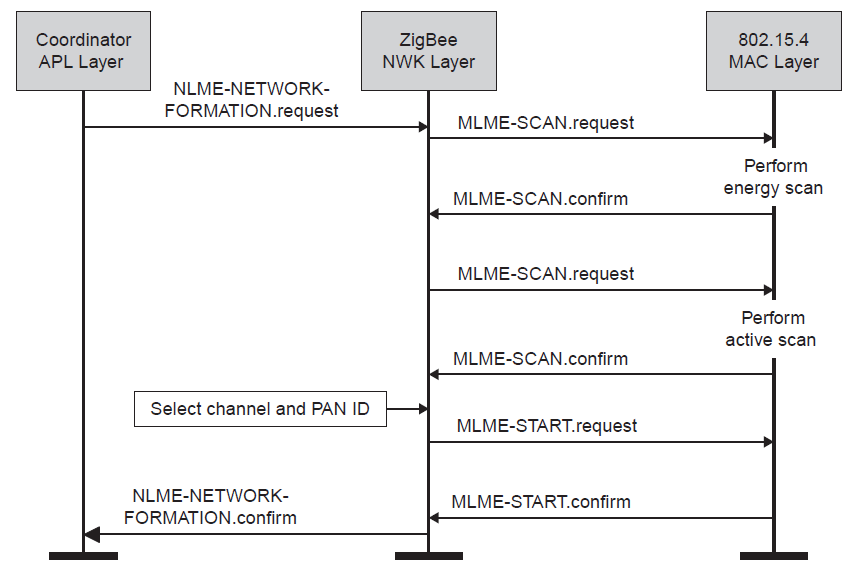
Ứng dụng chạy trên một node ZigBee Coordinator thực sự quyết định khi nào để thiết lập một network, từ một tập các kênh và từ tập các PAN ID. Ứng dụng trên ZC có thể là bất cứ thứ gì: một gateway được kết nối vói Internet, một hộp điều khiển, một bộ ổn nhiệt, đèn, hay công tơ điện. Các khả năng là vô hạn. Khi năng lượng được cấp đến thiết bị mà chứa ZigBee Coordinator, nó có thể tức khắc hình thành một network, hay có thể đợi một vài sự kiện trước khi tạo mạng. Nó thậm chí kiểm tra để xem những mạng nào sẵn sang ở đó, và quyết định trở thành ZigBee Router hơn là một Coordinator, nếu một node khác đã hoàn toàn tạo mạng mong muốn. Tuy nhiên ứng dụng đã được lập trình, một ZigBee Coordinator sẽ tạo một mạng.

NLME-NETWORK-FORMATION.request được khởi tạo bởi ZDO để tạo một mạng bởi ứng dụng. Trong Z-stack:

NLME\_NetworkFormationRequest();

Tiếp theo ZigBee gọi lớp MAC để thể hiện 2 scan: energy scan và active scan. Energy scan được dùng để xác định kênh nào là kênh yên tĩnh nhất từ tập các kênh được cụ thể trong biến thông tin cơ bản APS, apsChannelMask. Energy scan mất 0.5 giây cho mỗi kênh và chỉ là một kiểm tra “moment-in-time”. Kênh có thể thực sự nhiễu trong 1 giờ trước đó và quá trình này không phát hiện ra điều này. Việc quét tất cả 16 kênh mất khoảng 8 giây.

Tiếp theo là active scan, đơn giản một MAC beacon request được trả về không hoặc hơn các beacon response, được dùng để tìm mạng khác trong vùng. Active scan đảm bảo ZigBee không tạo một mạng có cùng PAN ID. Active scan có thể mất thời gian.



Hình 2.4‑1: Quá trình ZigBee tạo mạng

**Tham gia mạng**

ZigBee Router và ZigBee End-Device tham gia mạng. ZigBee Router thường được cấp nguồn chính, luôn bật, lắng nghe các packet để tìm đường. ZED thường dùng nguồn Pin và sleeping, chỉ waking up để giao tiếp một cách ngắn trước khi trở về sleep.

ZigBee Router có nhiệm vụ:

* Tìm và tham gia mạng
* Duy trì các broadcast thông qua mạng
* Tham gia việc tìm đường, gồm khám phá và duy trì đường đi
* Cho phép các thiết bị khác tham gia mạng
* Lưu các packet thay cho các children đang sleep

ZigBee End-Device có nhiệm vụ:

* Tìm và tham gia mạng
* Polling parent của nó để xem có bất kỳ các messages đã được gửi tới chúng khi chúng sleep hay không
* Tìm một parent mới nếu kết nối tới parent cũ bị mất (NWK rejoin)
* Sleep hầu hết thời gian để tiết kiệm Pin

Việc tham gia mạng là một quá trình của tìm mạng và node nào trong vùng và sau đó chọn một trong chúng để tham gia. Sự kết hợp được cung cấp có thể chấp nhận bởi network, việc tham gia hoàn thành và node tham gia sẽ có 1 địa chỉ trên mạng.

Quá trình tham gia sử dụng beacon request. Bất kì ZCs và ZRs trong vùng trả lời bằng beacon response. Các ZCs và ZRs có cùng kênh với node muốn tham gia, và không phân biệt PAN ID.

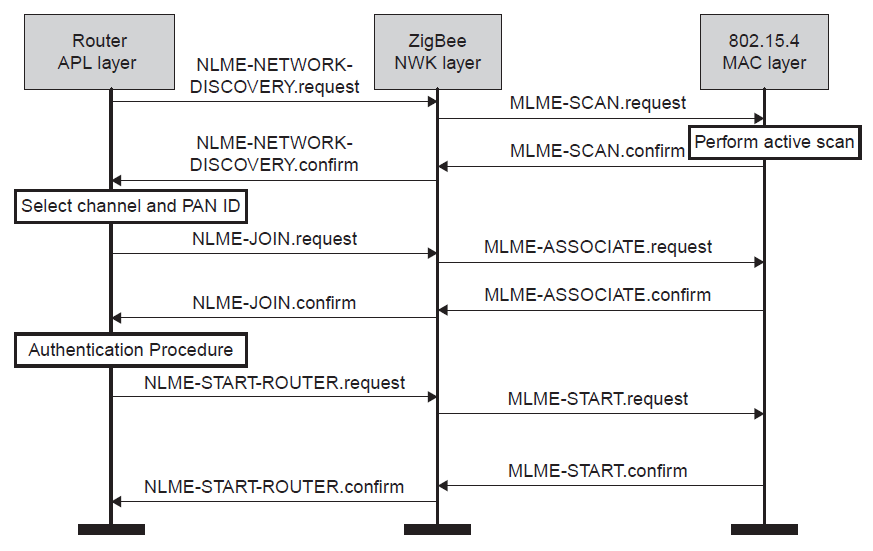
Các beacon response chứa khá nhiều thông tin về mạng ZigBee, gồm PAN ID, extended PAN ID, sự cho phép tham gia và node có đủ khả năng cho router hay end-device để tham gia hay không. Những gì beacon thiếu là thông tin về mức ứng dụng (application). Cho điều này, một node đang tìm kiếm đầu tiên phải tham gia mạng, tìm kiếm ứng dụng phù hợp, nếu không thấy thì rời mạng và thử các mạng khác.

ZR và ZED tham gia một node cụ thể, không phải một mạng, dùng 64-bit MAC cho địa chỉ đích và nguồn của MAC association request. Node thực hiện việc tham gia được gọi là child. Node nhận association request được gọi là parent.

ZR và ZC có thể là parent của node khác nhưng ZED luôn là child. Mối quan hệ parent/child không là gì trong mạng kiểu mesh. Bất kì các routers nào cũng có thể tìm đường qua bất kì router khác trong vùng nghe trên cùng mạng. Tất cả các routers là ngang hang. Nếu một parent hay child của router ra khỏi vùng nghe (thậm chí rời mạng) không tác động tới các đường đi, đường đi khác sẽ tới các thiết bị cụ thể này. ZigBee Router không tìm đường thông qua các mạng khác, chỉ trong cùng PAN ID và kênh.

Tuy nhiên, đối với ZEDs, mối quan hệ parent/child rất đặc biệt. ZEDs trong khi chúng có thể giao tiếp với các node khác trong mạng, chúng chỉ giao tiếp trực tiếp với parent. Hop kế tiếp của ZED luôn là parent của nó. Nếu một ZED mất kết nối với parent, nó phải tìm parent khác để giữ liên lạc với mạng, được gọi là tham gia lai (rejoin).

Không thường xuyên ZED mất liên lạc với parent. Sự giao tiếp 2.4GHz bị ảnh hưởng bởi nước và lien kết bị mất. Trong vài giây, ZED sẽ tìm parent mới, thông báo đến mạng rằng nó hoàn toàn di chuyển và các giao tiếp tiếp tục.



Hình 2.4‑2: Quá trình ZigBee tham gia mạng

Quá trình tham gia của ZR và ZED được mô tả trong đặc tả ZigBee. Đầu tiên một active scan (beacon request) gửi ra trên mỗi kênh. ZC hoặc ZED chờ một thời gian cho beacon response. Thời gian được cài bởi ứng dụng, nhưng mặc định là 0.5 giây trên kênh. Khi các beacon được thu thập, chúng được phân tích kênh và PAN ID. Ngoài ra còn có permit-join để ngăn các node tham gia vào và cũng có thể được dùng để bắt buộc các node có một parent cụ thể.

Sau khi active scan hoàn thành và một parent phù hợp được chọn, quá trình xác thực bắt đầu. Chú ý rằng node có một địa chỉ trên mạng trước thời gian xác thực bắt đầu. Sự xác thực chỉ có ở mạng bảo mật, và đưa trust center quyền từ chối/cho phép node tham gia. Một node giả mạo mà chỉ giả mạo một địa chỉ trên PAN sẽ không nhận key mạng và nên không thể giao tiếp với các node khác. Nếu xác thực không hoàn thành thành công, parent sẽ thông báo với child chưa được xác thực rời và đánh dấu địa chỉ đó có thể dùng bởi node khác muốn tham gia.

Khi một node hoàn toàn tham gia một mạng, nó có thể giao tiếp với bất kì node khác trong toàn bộ mạng. Không có yêu cầu cho việc binding hoặc các cơ chế khác. Đơn giản gửi dữ liệu từ node này đến node khác, miễn là biết địa chỉ short của node đó. Tất nhiên, ứng dụng muốn thấy packet thì Application Profile phải giống nhau ở hai phía và endpoint nguồn trên node gửi và endpoint đích trên node nhận phải được đăng kí vơi ZDO.

### Tái tham gia mạng

Việc tái tham gia giả sử node sẵn sang tham gia mạng, có một PAN ID, extended PAN ID, security key, và short address. Có nhiều nguyên nhân một node cần phải tái tham gia mạng:

* Một ZED mất lien lạc với parent của nó
* Năng lượng bị thay thế và nhiều hay tất cả các node trong mạng tái tham gia  « silently«
* Tham gia một mạng bảo mật nếu permit-joining tắt.

ZED luôn giao tiếp trực tiếp với parent của nó. Nếu parent không trả lời, child phải tìm parent mới để giữ giao tiếp trên mạng. Child tự quyết khi nào nó mất parent. ZigBee không cụ thể số lần poll hay message trước khi ZED cho rằng nó không thể giao tiếp với parent; ứng dụng quyết định.

Quá trình tái tham gia bắt đầu với một beacon request để tìm parent phù hợp. Nó không lien quan tới node parent tiềm năng có thể có permit-joining hay không. Chỉ một thứ là chúng có khả năng chứa thiết bị hay không. Sauk hi beacon request, ZED lấy một node (cùng PAN) làm parent, thực hiện tái tham gia, nhận một short address mới (chỉ trong stack profile 0x01), và cuối cùng phát một device-annouce để nói cho mạng rằng node hoàn toàn di chuyển. Bước cuối rất quan trọng để bảo vệ binding trong mạng.

Một loại khác của việc tái tham gia là “silent rejoin”. Silent rejoin không được đặc tả trong ZigBee, nhưng tất cả các nhà cung cấp stack có vì nó cần thiết trong một mạng hiện thực với bất kì kích thước nào. Ví dụ, tưởng tượng rằng năng lượng bị tái cung cấp đến tất cả các router trong một mạng 1000 node. Khi năng lượng có trở lại, nếu tất cả các node đều cố gằng tham gia (tái tham gia) mạng cùng một lúc, mạng sẽ thất bại : đơn giản quá nhiều lưu lượng trong mạng. Nhưng khi các routers biết thông tin mạng của nó (PAN ID, extended PAN ID, NwkAddr, security key), chúng đơn giản bắt đầu một cách thầm lặng, không nói bất cứ gì. Các node ZigBee không cần nói chuyện để duy trì trạng thái mạng. Khi mất điện có thể xem như mạng không nói chuyện trong một lúc. Khi điện có trở lại, mỗi router nhận chế độ trên PAN ID, extended PAN ID, NwkAddr, security key chính xác như thể mạng không bao giờ tắt. Điều này gọi là « silent rejion«

Silent Rejion cũng được dùng khi mạng chuyển đến kênh mới, một đặc tính mới trong ZigBee 2007 và Pro. Silent rejoin chỉ làm việc nếu các node có một vài loại lưu trữ vĩnh viễn (non-volatile memory).

Một cách dùng khác của tái tham gia là dùng NWK-Rejoin để tham gia một mạng mà có permit-joining tắt. Điều này thình thoảng được dùng nếu quá trình ủy nhiệm có network key, PAN ID,… sẵn sàng lập trình trong node. Quá trình NWK-Rejoin sẽ đưa thiết bị đó và địa chỉ trên mạng, và ZDP : DeviceAnnce sẽ cho phép tất cả các node trong mạng biết nó hoàn toàn tham gia.

### Gán địa **chỉ** ZigBee

Định địa chỉ là quan trọng trong một mạng. Địa chỉ mỗi node phải duy nhất trong ZigBee. ZigBee dùng hai địa chỉ duy nhất trên một node : long address (IEEE hay MAC address) và short address (NwkAddr).

Long address (64-bit), cũng được gọi là IEEE hay MAC address, được gán bởi nhà sản xuất thiết bị dùng 802.15.4 radio (không phải nhà sản xuất chip), và không thay đổi trong đời thiết bị. Long address định nghĩa một cách duy nhất thiết bị với tất cả các thiết bị khác trên thế giới.

Short address (16-bit) được gán tới một node tại thời điểm node tham gia mạng.

Chú ý rằng cả MAC layer header và NWK layer header có cả một địa chỉ nguồn và một địa chỉ đích. Nếu địa chỉ MAC 8-byte được dùng, thì sẽ dùng tới 32 byte của 127byte over-the-air packet. Do đó, ZigBee dùng một địa chỉ mạng 2-byte, giảm các trường này đến 8 bytes, cho phép hơn 24 bytes cho ứng dụng dùng.

Tại sao phải dùng các trường địa chỉ này trong cả MAC header và NWK header? Nếu việc gửi packet từ node “A” đến node “Z”, hop đầu tiên từ “A” tới “B”, hop tiếp theo từ “B” tới “C”, … cho đến hop cuối cùng từ “Y” tới “Z”. NwkSrc và NwkDst luôn chỉ từ “A” tới “Z”, trong khi MacSrc và MacDst là địa chỉ tại mỗi hop.

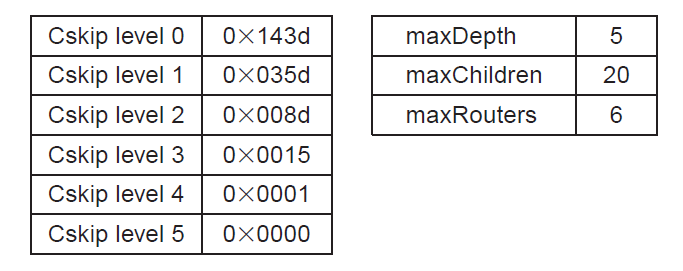
Khi nói về việc gán địa chỉ ZigBee, thì là liên quan tới địa chỉ short address. Short address được mạng ZigBee gán tại thời điểm một node tham gia mạng hoặc thiết lập mạng, và không liên quan tới IEEE address. ZigBee dùng hai mô hình để gán short address:

* Cskip
* Stochastic (random)

Gán địa chỉ Stochastic, một node tham gia vào một mạng chọn riêng cho mình một địa chỉ. Sau đó nó gửi một thông báo broadcast đến mạng để xem có node nào khác có địa chỉ đó hay không. Nếu có, node chọn một địa chỉ khác. Nếu không node giữ địa chỉ đó. Việc định địa chỉ theo tochastic có trong stack profile 0x02 (còn được gọi là ZigBee Pro).

Trong stack profile 0x01, các địa chỉ được gán với một mối quan hệ parent-child tạo một tree đối xứng. Mô hình gán địa chỉ trong stack profile 0x01 dùng một số được tính toán cho mỗi “depth” (số hop từ ZC), được gọi là Cskip (Child skip).

Trong Cskip, ZC là node 0 (0x0000). Node tiếp theo tham gia mạng sẽ nhận một địa chỉ từ node parent. Địa chỉ mà node này được gán phụ thuộc vào child là một router, mà có thể có các children của riêng nó hay một thiết bị, mà không có child.



Bảng 2.4‑1: Cskip được tính toán cho stack profile 0x01

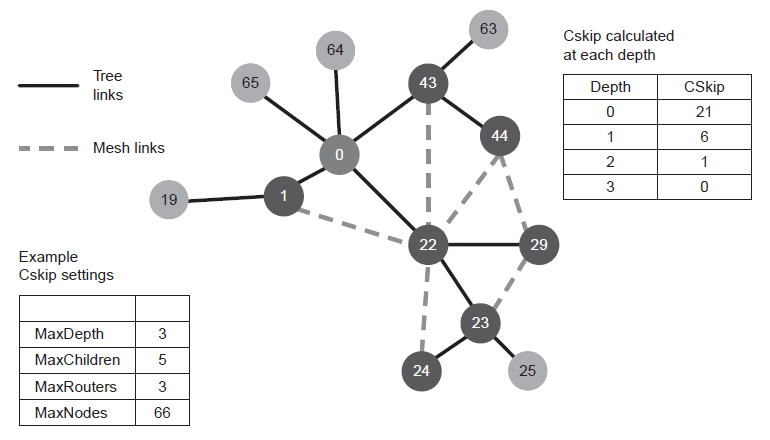
Cskip dùng ba thông số để xác định việc địa chỉ : maxDepth, maxChildren và maxRouter. Việc dùng các thông số này, Cskip có thể xác định một cách toán học những địa chỉ child mới tham gia, và như thế nào để tìm đường một packet theo tree đối xứng. Stack profile 0x01 dùng giá trị maxDepth(5), maxChildren(20), và maxRouter(6), mà giới hạn tổng số node trong mạng tới 31,101 nodes.

Khái niệm cơ bản là : Tree chia thành « levels«  level 0 là ZC, level 1 là children của nó, level 2 là children của children của nó … Router đầu tiên tham gia mạng ZC nhận địa chỉ 0x0001. Router tiếp theo tham gia ZC nhận địa chỉ (0x0001 + Cskip tại level đó), như vậy là 0x143e, bởi vì Cskip tại level 0 là 0x143d. Số này, 0x143d, đủ lớn cho router này và tất cả các children và grandchildren của nó phủ hết tree đối xứng.

ZED đầu tiên tham gia tại level 0 nhận một địa chỉ sau tất cả các router, đó là 0x796f. Công thức:

1[ZC nhận 1 địa chỉ]+6[maxRouter]x0x143d[Cskip tại level 0]=0x796f

Khái niệm của Cskip được giải thích dễ hơn với một mạng dùng các thông số Cskip nhỏ, mặc dù nó không tương thích với stack profile 0x01. Xem Hình 7.11. Các thông số là maxDepth(3), maxChildren(5) và maxRouter(3). Điều này chỉ cho phép 66 node trong mạng. Các thông số này chỉ để giải thích tree đối xứng cho dễ hiểu mà thôi.



Hình 2.4‑3: Gán địa chỉ Cskip trong cây đối xứng

Để hiểu tree định địa chỉ, dễ nhất để bắt đầu tại đáy của tree (maxDepth). Xét node 24. Bởi vì nó nằm ở maxDepth, có ba hop từ ZC, nó không thể bất kì children, nên Cskip tại mức này là 0. Parent của nó là node 23. Node này tại mức 3, có hai hop từ ZC, nên nó có Cskip là 1. Mỗi children của nó sẽ dùng một địa chỉ. Node 23 có năm children, nên địa chỉ của nó (23) cộng với năm children của nó (24-28) dùng hết sáu địa chỉ. Do đó, Cskip của parent (node 22) của nó là 6. Node 22 và tất cả các children và grandchildren dùng tổng cộng 21 địa chỉ: 1+3x6+2=21. Ở đây 2 là số ZED có thể tham gia vào node:

2 = maxChildren(5)-maxRouter(3)= maxChildren(2)

Do đó, Cskip tại mức 0, cho ZC trong mạng này là 21. Mỗi router child tại mức 0 dùng 21 địa chỉ cho chính nó và tất cả các nhánh của nó. Các ZED không bao giờ có children, nên các ZED dùng hai địa chỉ. Công thức cho tổng các node cho phép trong mạng với các thông số ở trên là:

1 + 3x21 + 2 = 66

Bằng cách giả sử một tree đối xứng, ZigBee có thể biết, dùng các công thức toán học đơn giản, địa chỉ node là một child (bao gồm grandchildren) hay không. Nếu nó là một child, packet được gửi đến hop tiếp theo bên dưới, đến một router hoặc địa chỉ chính nó. Nếu địa chỉ không phải là một child, packet được gửi tới parent. Node 22 biết rằng bất kì địa chỉ từ 23 đến 42 là child. Còn lại là không phải nên nó gửi tới parent.

Vấn đề của Cskip, và nguyên nhân nó không được dùng trong ZigBee Pro (stack profile 0x02) là nó không thể co giản ngoài maxDepth(5), cho phép lớn nhất 10 hops trong mạng (2x maxDepth). Nếu muốn nhiều hop hownm chỉ thay đổi maxDepth đến 6 và maxChildren(20) và maxRouter(6) cho phép 186,621 node, một số không thể chứa trong 16-bit short address.

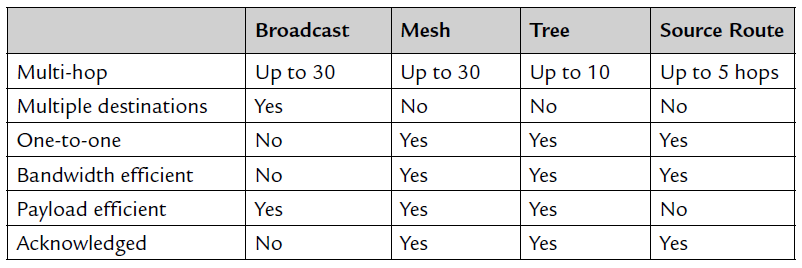
Trong hình Hình 2.4‑3, bao gồm cả kết nối tree (đường đen) và mesh (đường đứt). ZigBee luôn luôn là một mạng mesh. Tree có thể được dùng như một mô hình tìm đường backup nếu mesh quá tải, nhưng không thay thế mesh.

### Tìm đường cho packet trong ZigBee

ZigBee dùng nhiều phương pháp cho việc tìm đường cho các packet từ một node đến node khác:

* Broadcast (từ một node đến nhiều node)
* Mesh routing (unicast từ một node đến node khác)
* Tree routing (unicast từ một node đến node khác, chỉ trong stack profile 0x01)
* Source routing (unicast từ một node đến node khác, chỉ trong stack profile 0x02)

Mỗi phương pháp có ưu điểm và nhược điểm



Bảng 2.4‑2: So sánh các phương tìm đường trong ZigBee

Broadcast cho phép một node vươn tới nhiều node khác với chỉ một request. Phương pháp này không cần ACK và cần nhiều tài nguyên.

Mesh routing là bảng điều khiển và rất hữu dụng (time, bandwidth, tài nguyên bộ nhớ), khi đường đi được thiết lập. Packet được gửi theo mesh có ACK, nên node gửi biết packet đã nhận hay chưa. Các đường đi mesh được phân phối, mà điều này giảm overhead trong over-the-air packet. ZigBee mesh có thể phân phối các packet đến 30 hop.

Tree routing, cũng có ACK, chỉ có trong stack profile 0x01. Nó được diễn tả ở Cskip. Tree có hiệu quả băng thông như mesh và hiệu quả hơn về bộ nhớ. Nhưng khi liên kết giữa parent và child đứt thì nó không thể phục hồi. Vậy nên ZigBee dùng mesh như mặc định.

Source routing, như mesh và tree, có ACK, chỉ có trong stack profile 0x02. Được sử dụng chính khi một data concentrator (hoặc gateway) cần giao tiếp với nhiều node. Với mesh routing, mỗi đường đi cần một table entry, và các node ZigBee không đủ RAM cho một ngàn đường đi. Trong source routing, một node đơn (đắt hơn) có nhiều RAM để lưu tất cả các đường đi. Đường đi cho bất kì giao tiếp cụ thể nào được gửi như một over-the-air packet. Hạn chế lớn nhất là nó giới hạn lớn nhất 5 hop.

## MAC và PHY Layer

Lớp MAC và PHY được đặc tả trong IEEE 802.15.4, nhưng ZigBee không hiện thực đầy đủ các đặc điểm mà IEEE đưa ra, bởi ZigBee muốn tối thiểu kích thước bộ nhớ để phù hợp với các ứng dụng trong một vi điều khiển 8-bit và giảm tiêu hao năng lượng không cần thiết.

### PHY Layer

Trong mạng ZigBee, lớp giao thức thấp nhất là lớp vật lý IEEE 802.15.4, hay PHY. Lớp này là gần nhất với phần cứng và điều khiển và giao tiếp trực tiếp với radio transceiver. Lớp PHY chịu trách nhiệm cho việc kích hoạt radio truyền và nhận packet. PHY cũng lựa chọn tần số kênh và đảm bảo kênh hiện không được dùng bởi các thiết bị khác trên mạng khác.

### MAC Layer

Medium Access Control (MAC) layer cung cấp giao diện giữa PHY layer và NWK layer. MAC có trách nhiệm sinh ra các beacon và đồng bộ thiết bị với các beacon (trong mạng beacon-enabled). MAC layer cũng cung cấp dịch vụ kết hợp và không kết hợp.

## Giới thiệu Z-stack của Texas Instrument

Z-Stack™ là một stack giao thức phù hợp với ZigBee của TI cho các sản phẩm và platform IEEE 802.15.4. Z-Stack™phù hợp với đặc tả ZigBee® 2007 (ZigBee và ZigBee PRO), hỗ trợ tính chất cả ZigBee và ZigBee PRO trên CC2530 System-on-chip, MSP430+CC2520 và Stellaris LM3S9B96+CC2520. Z-Stack™ hỗ trợ Smart Energy và Home Automation profiles.

Toàn bộ Z-Stack™ download tại trang <http://www.ti.com/tool/z-stack> . Z-Stack™ đóng một phần source-code nên được biên dịch sẵn cho một số dòng chip mà Z-Stack™ hỗ trợ. Tài liệu liên quan đến Z-Stack™ cũng được hỗ trợ đầy đủ kèm theo gói stack được tải về.

# HIỆN THỰC

## Kiến trúc hệ thống

* Toàn bộ hệ thống bao gồm :

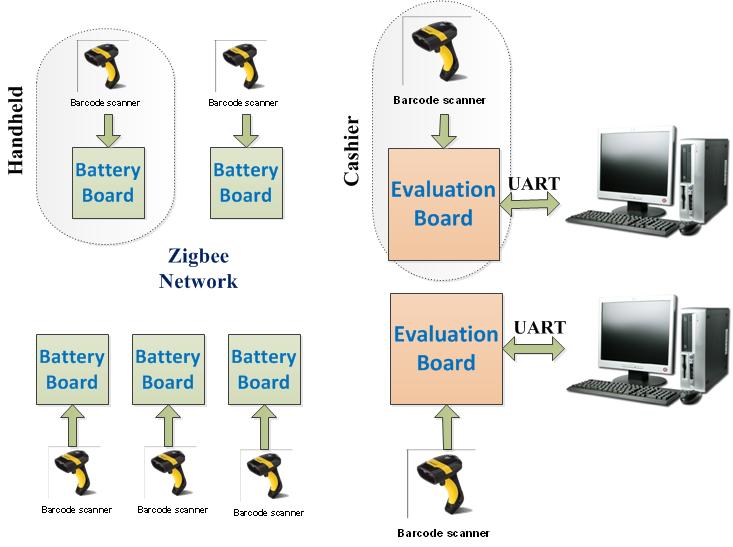
+ 2 Cashier : bao gồm 1 Evolution Board và 1 barcode scanner, có chức năng request thông tin giỏ hàng của khách hàng rồi gửi lên cho PC để tính toán giá cả cho giỏ hàng.

+ 5 Handheld : bao gồm 1 Battery Board và 1 barcode scanner, có chức năng scan từng món hàng của khách hàng rồi chứa vào trong bộ nhớ

+ Handheld và Cashier giao tiếp với nhau thông qua mạng Zigbee

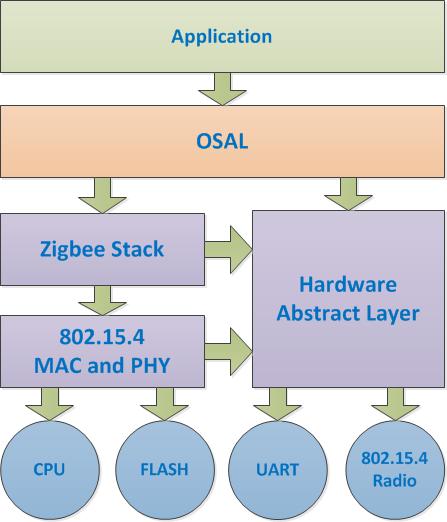
+ Các giao tiếp khác (giữa barcode scanner với board, giữa board với PC) : truyền nhận dữ liệu thông qua cổng COM

* Sơ đồ kết nối thiết bị được mô tả như hình dưới đây :

****

Hình 3.1‑1: Sơ đồ kết nối thiết bị của hệ thống

* Kiến trúc software của hệ thống :



Hình 3.1‑2: Sơ đồ phân lớp kiến trúc phần mềm

Hệ thống queue busting mà nhóm thực hiện :

* Giao tiếp không dây : bộ phần cứng CC2530 ZDK của Texas Instrument.
* Máy quét mã vạch : máy quét cầm tay có dây của Datalogic (Datalogic Handheld corded scanner)
* Giao tiếp giữa Scanner-PC, giữa Scanner-CC2530 : thông qua cổng COM (tự xây dựng)

### Bộ KIT phát triển CC2530 của TI

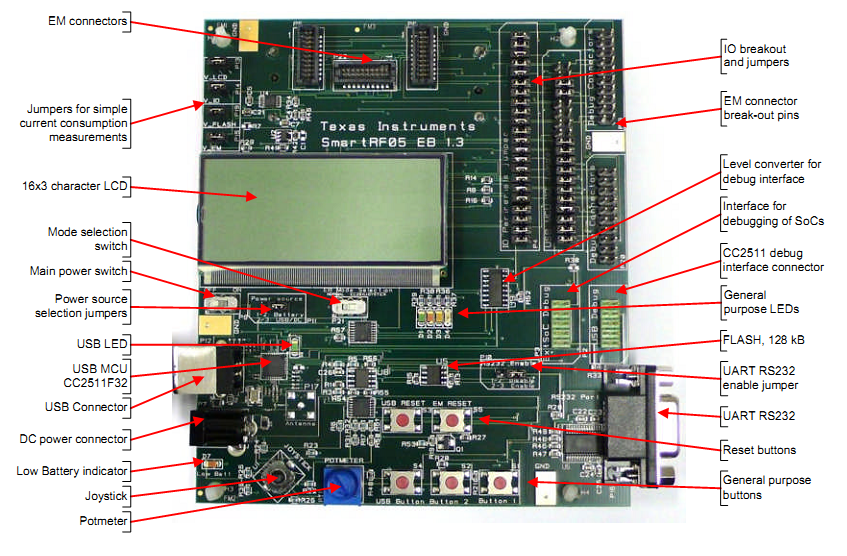
Bộ Kit phát triển CC2530DK hỗ trợ CC2530 System-on-chip phù hợp chuẩn 2.4GHz IEE 802.15.4 thế hệ thứ hai của TI và chứa tất cả phần cứng, phần mềm và các công cụ cần thiết để xây dựng sản phẩm phù hợp với 802.15.4.Có thể hoạt động ở 125°C và điện áp thấp, tiết kiệm năng lượng.

CC2530DK gồm:

* 7 x CC2530EM Evaluation Modules
* 7 x 2.4 GHz Antennas
* 2 x SmartRF05EB Evaluation Boards
* 2 x SmartRF05BB Battery Boards
* 1 x CC2531 USB Dongle
* USB cables và tài liệu

#### SmartRF05 Evaluation Boards

* Dùng để nạp code, kiểm tra lỗi cho CC2530 Evaluation Modules.
* Cung cấp nguồn cho CC2530 từ 2 pin AA, DC-in hoặc từ USB.
* Gắn các thiết bị ngoại vi kết nối với CC2530 Evaluation Modules như : LCD, Flash (128kB), 2 UART, Joystick, Switch…
* Trong hệ thống, board này đóng vai trò là cashier. Thực hiện việc thanh toán (checkout) các mã giỏ hàng (basket id) của khách hàng, kết nối với PC qua cổng COM để truy cập cơ sở dữ liệu và in hóa đơn thanh toán.



Hình 3.1‑3:SmartRF05 Evaluation Board

#### SmartRF05 Battery Boards

* Nhỏ hơn và đơn giản hơn SmartRF05 Evaluation Boards.
* Chỉ có các ngoại vi cần thiết.
* Trong hệ thống, đóng vai trò là handheld. Thực hiện nhiệm vụ kết nối với máy quét mã vạch (scanner) và lưu trữ vào flash nội dung giỏ hàng của khách hàng. Khi cashier có yêu cầu, handheld sẽ gửi trả nội dung giỏ hàng thông qua mạng zigbee.



Hình 3.1‑4:SmartRF05 Battery Board

#### CC2530 Evaluation Modules

* Module chính thực hiệc các công việc liên quan đến mạng không dây.
* Cần kết nối với board SmartRF05 và antenna để hoạt động.

****

Hình 3.1‑5:CC2530 Evaluation Modules

#### Một số các thiết bị khác



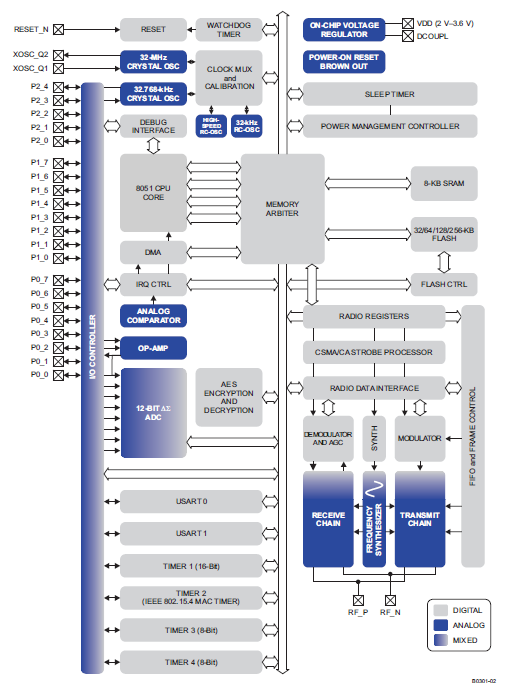
Hình 3.1‑6: CC2531 USB Dongle

****

Hình 3.1‑7: CC2530 Antenna

#### Giới thiệu chip CC2530

* RF/Layout
  + Bộ truyền nhận RF phù hợp 2.4GHz IEEE 802.15.4
  + Độ nhạy nhận cao
  + Năng lượng đầu ra có thể lập trình tới 4.5 dBm
  + Cần rất it các thành phần bên ngoài
  + Chỉ một thạch anh đơn giản cho đồng bộ các mạng
* Năng lượng thấp
  + Active-Mode RX (CPU Idle): 24mA
  + Active-Mode TX tại 1dBm (CPU Idle): 29mA
  + Dãy điện áp cung cấp rộng (2V – 3.6V)
* Vi điều khiển
  + Nhân vi điều khiển 8051 năng lượng thấp và hiệu suất cao với Code Prefetch
  + Bộ nhớ Flash In-System-Programmable 32-, 64-, 128-, 2560Kbytes.
  + 8Kbytes RAM được duy trì trong tất cả các chế độ năng lượng
  + Hỗ trợ debug phần cứng
* Ngoại vi
  + Năm kênh DMA và bộ mã hóa, giải mã AES
  + Bộ so sánh năng lượng siêu thấp
  + IEEE 802.15.4 MAC timer và các timer mục đích chung (một 16-bit, hai 8-bit)
  + Hỗ trợ phần cứng CSMA/CD
  + Cảm biến nhiệt độ và theo dõi Pin
  + 12-bit ADC với 8 kênh và độ phân giải có thể cấu hình
  + Hai USART với hỗ trợ các giao thức nối tiếp
  + 21 I/O chân mục đích chung
  + Watchdog Timer
* Ứng dụng
  + Các hệ thống 2.4GHz IEEE 802.15.4
  + Các hệ thống điều khiển từ xa RF4CE
  + Hệ thống ZigBee
  + Home/Building Automation
  + Hệ thống ánh sáng
  + Giám sát và điều khiển công nghiệp
  + Mạng cảm biến không dây năng lượng thấp
  + Điện tử dân dụng và chăm sóc sức khỏe



Hình 3.1‑8: Kiến trúc CC2530

### Datalogic Handheld corded scanner

Datalogic scanner là thiết bị đọc mã vạch do Datalogic sản xuất, với giao tiếp chuẩn RS232 được cấu hình với9600 baud rate, none parity bit, stop bit là 1

### Bộ chuyển đổi điện áp RS232

Chuyển đổi điện áp 3.3V dùng cho board sang điện áp của chuẩn RS232 để giao tiếp với máy tính và máy đọc mã vạch dùng MAX3232

## Thiết bị cashier

### Nguyên tắc hoạt động

Sử dụng 3 loại frame chính để giao tiếp với các thiết bị handhled :

* Basket frame : các lệnh liên quan đến giỏ hàng.

+ Basket request frame : '%'+[Basket ID:8bytes].

+ Basket respond frame : 'H'+[Short Addr:2bytes]+[Basket ID len:1byte]+[Product ID len:1byte]+[Basket].

* Status frame :các lệnh liên quan đến trạng thái hiện tại của mạng.

+ Status request frame : 'S'.

+ Status respond frame : 'S'+[MAC addr:6bytes]+[Short Addr:2bytes]+[Parent Addr: 2bytes].

* Delete frame : các lệnh mà cashier hay PC cần handheld xử lý.

+ Delete all basket frame : '^0' hoặc '@FlashReset'.

+ Delete 1 basket frame : '^'+[num:1byte]+[Basket IDs].

Cashier giao tiếp với 3 device chính :

* Giao tiếp với barcode scanner :

+ Nếu dữ liệu nhận từ scanner là *'+'* hay *'-'*, thì chuyển sang mode “prod”. Trong mode này, nếu dữ liệu nhận từ scanner là mã sản phẩm (products ID) thì hàng hóa đó sẽ được thêm/xóa khỏi giỏ hàng. Việc thêm/xóa hàng hóa này sẽ được gửi lên PC để tính tiền, và không ảnh hưởng đến handheld đang chứa giỏ hàng đó.

+ Nếu dữ liệu nhận từ scanner là *'%'*, thì chuyển sang mode “basket”. Trong mode này, nếu dữ liệu nhận từ scanner là basket ID thì cashier sẽ broadcast basket request frame cho tất cả các handheld có trong mạng.

+ Nếu dữ liệu nhận từ scanner là *'@FlashReset'*, cashier sẽ broadcast cho tất cả các thiết bị handheld để xóa toàn bộ dữ liệu đang lưu trong bộ nhớ flash. Chỉ dùng cho khi siêu thị đã ngừng hoạt động, cần reset lại các thiết bị.

+ Nếu dữ liệu nhận từ scanner là *'S'* , cashier sẽ broadcast cho tất cả các thiết bị handheld để yêu cầu các handheld báo cáo trạng thái của mình. Chỉ dùng khi cần kiểm tra thiết bị có hoạt động bình thường hay không.

* Giao tiếp với PC :

+ Nếu dữ liệu nhận được từ PC có dạng *'^'+[num:1byte]+[Basket IDs]*, cashier sẽ tìm trong dữ liệu của mình để gửi tới 1 handheld đang giữ giỏ hàng có Basket ID tương ứng. Khi handheld nhận được frame này, handheld sẽ xóa basket có Basket ID tương ứng trong flash của mình.

+ Nếu dữ liệu nhận được từ PC có dạng *'S',*cashier sẽ broadcast cho tất cả các thiết bị handheld để yêu cầu các handheld báo cáo trạng thái của mình.

* Giao tiếp với mạng Zigbee :

+ Nếu không nhận được gói ACK của handheld thì sẽ gửi mã lỗi tương ứng cho PC.

+ Nếu nhận được basket respond frame, cashier đọc qua 4 byte đầu để lưu lại địa chỉ và basket ID tương ứng của handheld. Sau đó sẽ chuyển toàn bộ frame đó cho PC.

+ Nếu nhận được status respond frame, cashier sẽ chuyển toàn bộ frame đó cho PC.

### Hardware

Phần cứng (hardware) của cashier bao gồm :

* 1 module scanner: giao tiếp với MCU thong qua UART
* 1 module radio: được tích hợp sẵn trên CC2530F256, 8k Bytes RAM, 256k Bytes Flash
* 1 module quản lý nguồn: sử dụng 2 Pin AA ~ 3.3V cho MCU và một mạch boost từ 3.3V lên 5V cho Scanner. Tuy nhiên, có thể sử dụng nguồi DC-in hay nguồn từ USB.
* 2 LED trạng thái



Hình 3.22‑3.2‑1: Sơ đồ khối các chức năng

### Software

Chương trình (software) trên cashier :

* Chương trình dùng một hệ điều hành thời gian thực của Z-Stack (OSAL) để quản lý các thiết bị được tối ưu. OSAL quản lý các thiết bị thông qua các sự kiện (event), có 4 event chính như sau :

+ Timer : gây ra do timer đã đến hết khoảng thời gian quy định trước, chủ yếu dùng để báo lỗi.

+ Scanner : gây ra do barcode scanner truyền dữ liệu thông qua UART 2.

+ PC : gây ra do PC dư liệu thông qua UART 1.

+ Radio Frequency : gây ra do có sự thay đổi trong mạng.



Hình 3.2‑2: ???

* Timer Event : khi một quá trình xử lý cần tính toán thời gian để xuất ra các thông báo có liên quan, Timer sẽ được kích hoạt để bắt đầu đếm thởi gian. Khi Timer đã đếm đến thời điểm xác định, Timer sẽ gây ra một event trong OSAL.



Hình 3.2‑3: ???

* Scanner Event : khi barcode scanner đọc 1 mã vạch, dữ liệu sẽ được gửi đến thông qua UART (điều khiển bằng DMA). Khi dữ liệu đã được truyền hết thì UART sẽ gây ra một event trong OSAL.



Hình 3.2‑4: ???

* PC Event : khi PC truyền dữ liệu xuống thông qua UART (quản lý theo cơ chế interrupt) sẽ gây ra một event trong OSAL.



Hình 3.2‑5: ???

* Radio Frequency Event : gây ra khi có bất cứ sự thay đổi của mạng, có rất nhiều sư kiện dạng này tuy nhiên, chúng tôi chỉ tập trung vào 2 sự kiện chính là có frame dữ liệu và frame ack.



Hình 3.2‑6: ???

## Thiết bị handheld

### Ý tưởng

Ứng dụng có 3 quá trình xử lý dữ liệu:

Quá trình kết nối mạng

* Khi ứng dụng khởi động sẽ tự động kết nối với mạng ZigBee với cấu hình sẵn và LED trạng thái sẽ nhấp nháy khi kết nối
* Kết nối thành công LED trạng thái sẽ tắt

Quá trình nhận dữ liệu từ người dùng

* Người dùng quét một Basket ID từ Scanner, LED1 sáng. Handheld sẵn sàng cho người dùng quét Product IDs.
* Khi quét Product IDs, LED2 sẽ nhấp nháy báo đã nhận dữ liệu
* Khi quét xong các Product IDs, người dùng quét lại Basket ID để lưu vào bộ nhớ flash và LED1 tắt.

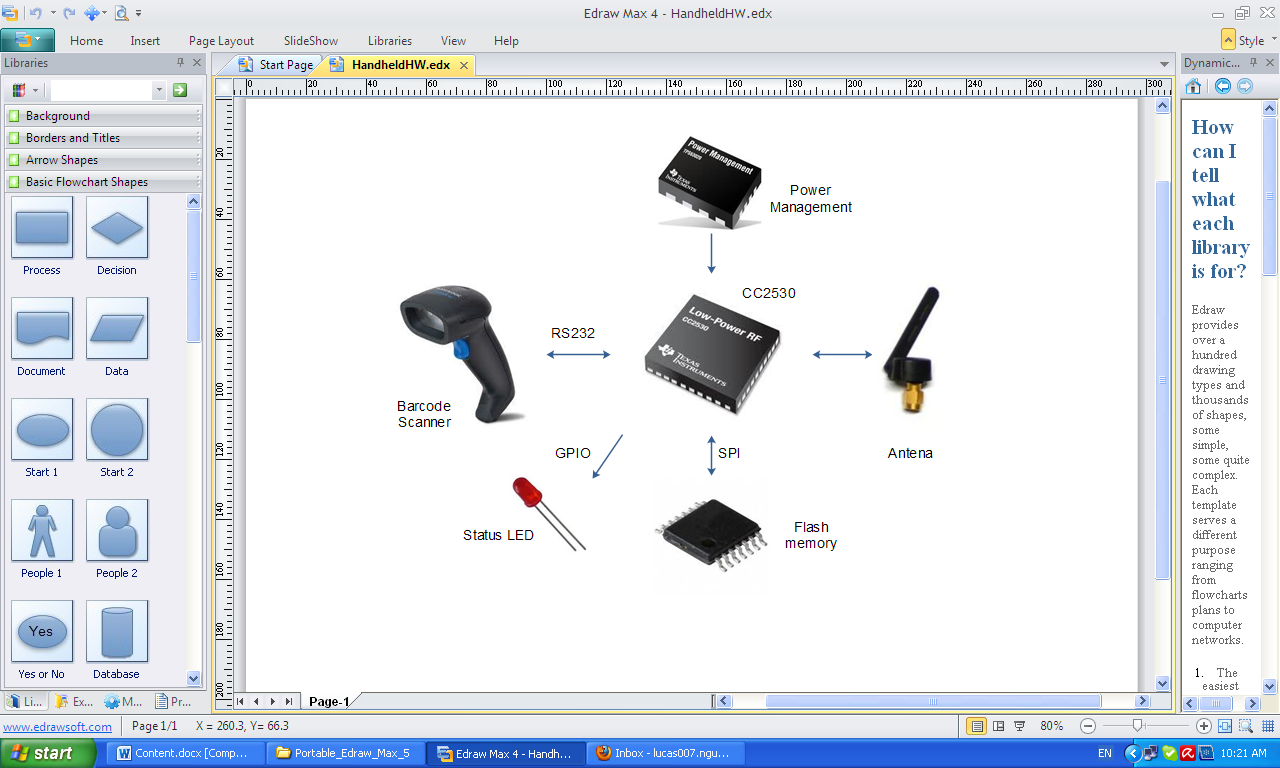
Quá trình nhận dữ liệu và trả lời cho hệ thống

* Yêu cầu dữ liệu: Đọc bộ nhớ flash và tìm dữ liệu, nếu có thì gửi dữ liệu theo địa chỉ của thiết bị yêu cầu.
* Xóa dữ liệu: Yêu cầu xóa một Basket hoặc toàn bộ trong flash.

### Mô hình

Hardware:

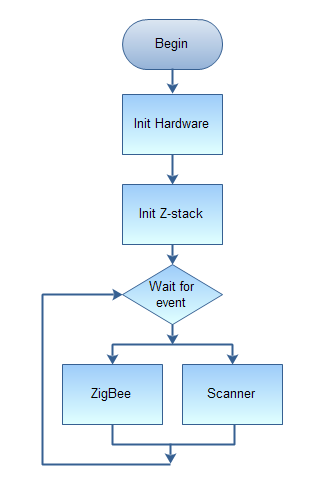
* 1 module scanner: giao tiếp với MCU thong qua UART
* 1 module radio: được tích hợp sẵn trên CC2530F256, 8k Bytes RAM, 256k Bytes Flash
* 1 module quản lý nguồn: sử dụng 2 Pin AA ~ 3.3V cho MCU và một mạch boost từ 3.3V lên 5V cho Scanner.
* 1 Flash 256kBytes: cho việc lưu trữ dữ liệu
* 2 LED trạng thái



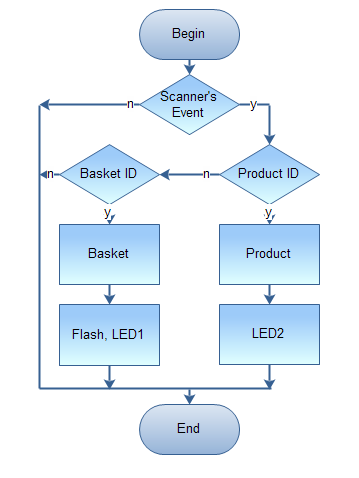
Hình 3.3‑1: Các phần cứng cho hiện thực

Sofware cho ứng dụng bao gồm:

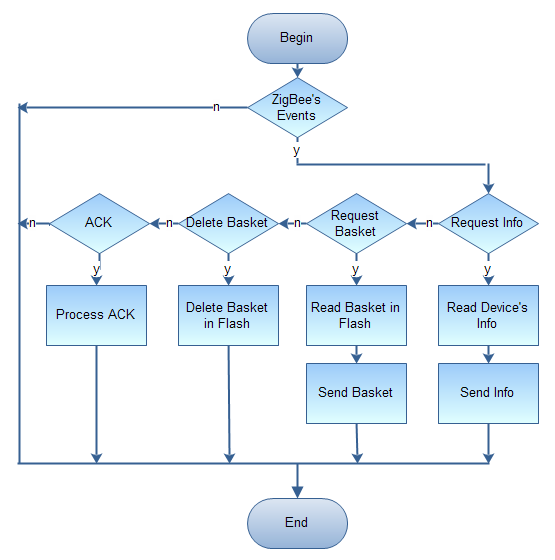
* Giao tiếp với Scanner thông qua UART: 9600 baud rate, none parity bit, stop bit là 1
* Giao tiếp với bộ nhớ Flash M25PE20: xây dựng bộ thư viện thao tác như cấu hình, đọc, ghi, xóa …
* Giao tiếp với mạng ZigBee: sử dụng Z-stack của TI



Hình 3.3‑2: Sơ đồ mô tả tổng quát task ứng dụng trong Handheld



Hình 3.3‑3: Sơ đồ xử lý dữ liệu từ Scanner trong Handheld



Hình 3.3‑4: Sơ đồ xử lý dữ liệu từ mạng ZigBee trong Handheld

### Hiện thực

Layout của bộ nhớ Flash

* Header
  + Số lượng Basket: 1byte, có tối đa 255 Basket trong Flash
  + Số thứ tự Basket cuối cùng: 1byte
  + Chiều dài tối đa của một Basket: 2byte
* Data
  + Một cờ: 1 byte, cho biết vùng nhớ có basket hay không
  + Data của Basket: có chiều dài được định nghĩa header của Flash

Mỗi Basket sẽ được lưu trong với một kích thước cố định được quy định bởi chiều dài tối đa và có cấu trúc dữ liệu như sau:

typedef struct {

char id[BASKET\_ID\_LEN];

uint8 len;

Product prods[MAX\_PRODS];

}Basket;

Trong mỗi Basket chứa một số sản phẩm nhất định (do giới hạn bộ nhớ của thiết bị) và cấu trúc dữ liệu của sản phẩm như sau:

typedef struct Product{

uint8 id[PRODS\_ID\_LEN];

uint8 num;

}Product;

Toàn bộ ứng dụng trên Handheld được hiện thực trong một task của OSAL. Sử dụng cơ chế tạo event trong OSAL của Z-stack để gửi sự kiện đến task này cho việc xử lý. Có 2 nhánh sự kiện chính cho 2 luồng dữ liệu:

* Sự kiện có dữ liệu từ Scanner
* Sự kiện từ mạng ZigBee, Z-stack

## Ứng dụng trên PC (personal computer)

### Mục đích

a. Yêu cầu chung :

- Ngôn ngữ lập trình : Java. Thiết kế giao diện sử dụng thư viện SWING, AWT của Java.

- Chương trình có chức năng nhận các gói dữ liệu từ Cashier thông qua giao tiếp COM, tiến hành xử lý và xuất kết quả về thông tin sản phẩm, giá tiền, … cho khách hàng.

- Chương trình sử dụng file excel chứa thông tin của hàng hóa, và thông tin gói sản phẩm thành tiền.

b. Mục đích của ứng dụng:

- Người sử dụng ( nhân viên tính tiền) chỉ cần chọn cổng COM kết nối với board, và sau đó quét các mã vạch chứa ID của người mua, tiến hành tính tiền.

- Chương trình dùng :

+ 1 file Database.xls giả lập 1 cơ sở dữ liệu (CSDL) các sản phẩm có trong siêu thị.

+ 1 file MacTables.xls để định danh các máy handheld trong siêu thị.

+ Sau mỗi lượt tính tiền, thông tin gói hàng sẽ được lưu xuống 1 file excel có tên là [thời gian tính tiền]+ ID gói hàng.

+ Nếu hệ thống có phát sinh lỗi, những lỗi đó sẽ được lưu lại trong file log.txt.

- Chương trình được hiện thực nhằm mục đích demo hệ thống, Giả lập 1 chương trình tính tiền của siêu thị. Nếu hệ thống được ứng dụng trong thực tế phải xây dựng lại CSDL, và in ra phiếu tính tiền thật…

c. Các gói dữ liệu :

\*) Dữ liệu vào:

- Gói dữ liệu thông thường (Packet A): là gói dữ liệu chứa các thông số của giỏ hàng bao gồm :

+ Ký tự nhận dạng.

+ Địa chỉ MAC của handle gửi đến.

+ Định dạng chiều dài của mã gói dữ liệu và mã sản phẩm.

+ Mã gói dữ liệu, số lượng loại sản phẩm.

+ [Mã sản phẩm] , [số lượng sản phẩm], ……….

- Gói dữ liệu thêm vào hoặc bớt ra các sản phẩm trong giỏ hàng (Packet B): bao gồm:

+ Ký tự nhận dạng.

+ Mã sản phẩm.

- Gói dữ liệu chứa thông tin cập nhật trạng thái mạng hiện tại (Packet C): bao gồm:   
 + Ký tự nhận dạng.

+ Địa chỉ MAC của handle

+ Địa chỉ SHORT của handle.

+ Địa chỉ PARENT của handle.

- Gói dữ liệu báo lỗi từ hệ thống (Packet D):

+ Ký tự nhận dạng.

+ Mã lỗi.

\*) Dữ liệu ra:

- Gói dữ liệu yêu cầu cập nhật trạng thái mạng hiện tại (Packet E):

+ Ký tự nhận dạng.

- Gói dữ liệu yêu cầu xóa gói dữ liệu đã xử lý xong ở handle (Packet F):

+ Ký tự nhận dạng.

+ Mã gói dữ liệu đã xử lý xong.

### Hiện thực

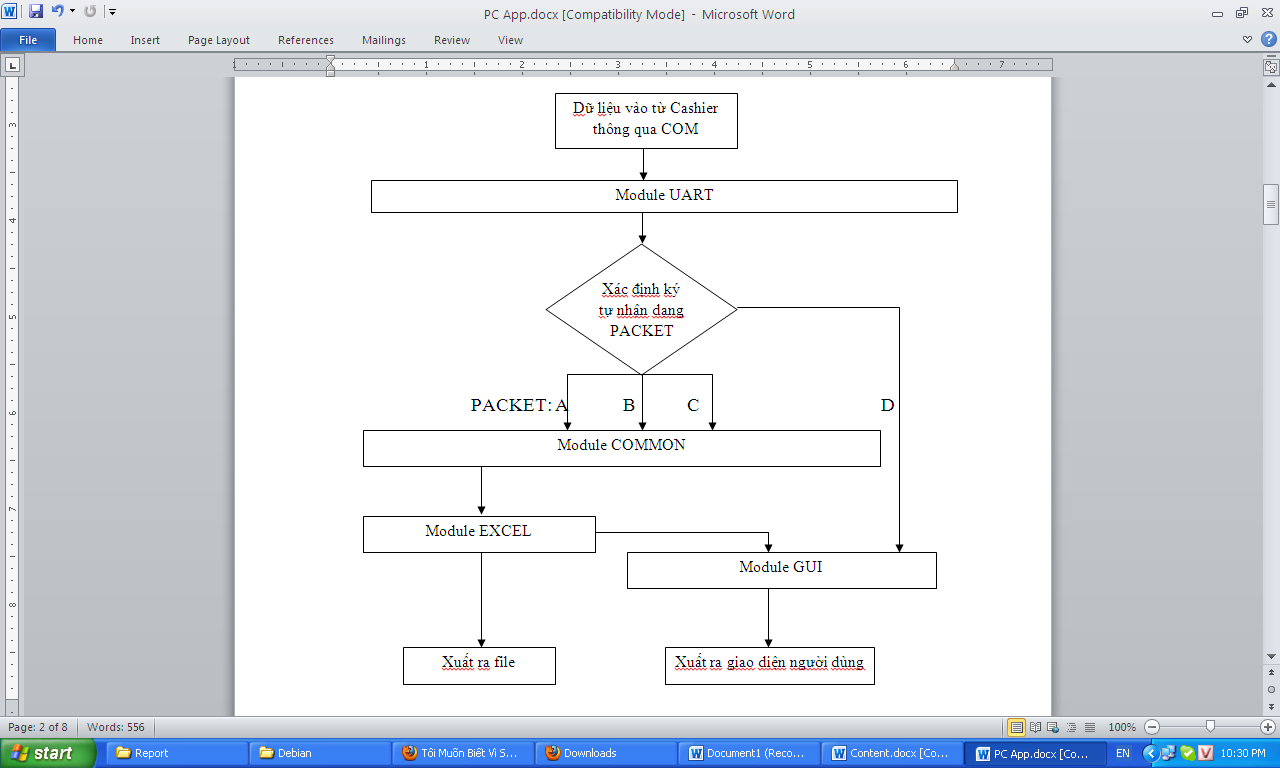
Cấu trúc chương trình gồm các khối sau:

+ UART: các hàm thực hiện việc kết nối, đọc, ghi dữ liệu từ COM.

+ EXCEL: các hàm thực hiện việc đọc, ghi, tạo file .xls.

+ COMMON: các hàm xử lý dữ liệu vào và trả về dữ liệu cho người dùng.

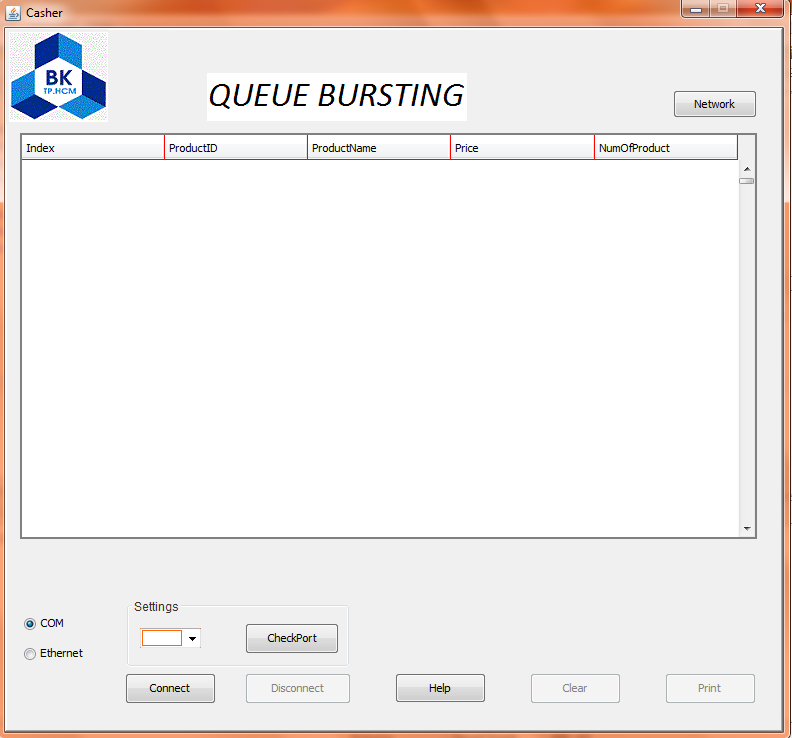
+ GUI: Xử lý giao diện để xuất ra cho người dùng xem.



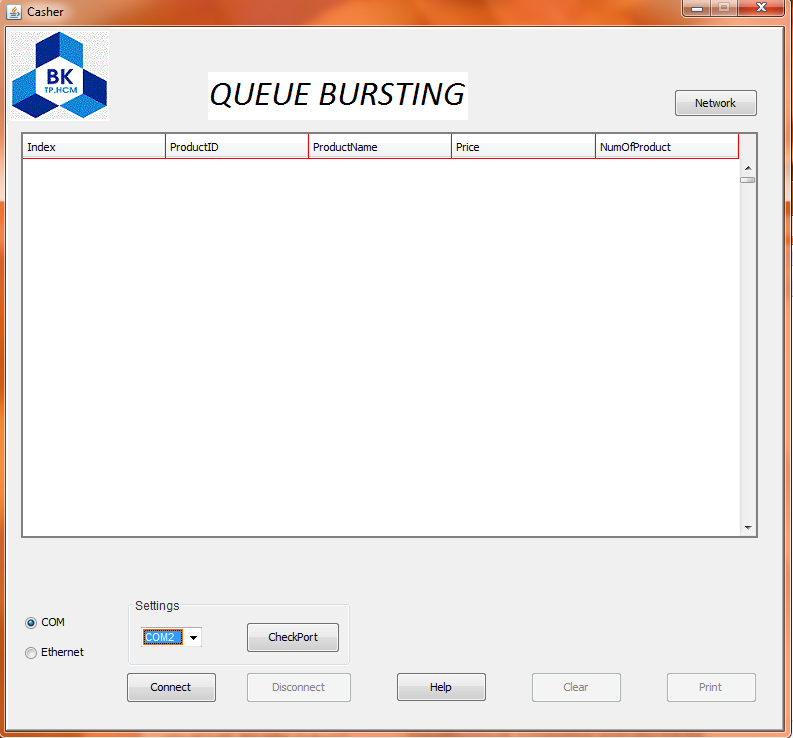
Hình 3.4‑1: Sơ đồ hiện thực ứng dụng trên PC

### Hoạt động của ứng dụng PC

Giao diện chính:

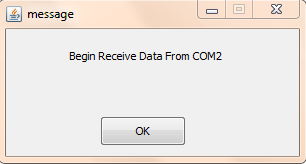


Chọn Button CheckPort:

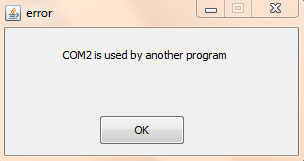


Chọn Port muốn kêt nối và nhấn Connect.

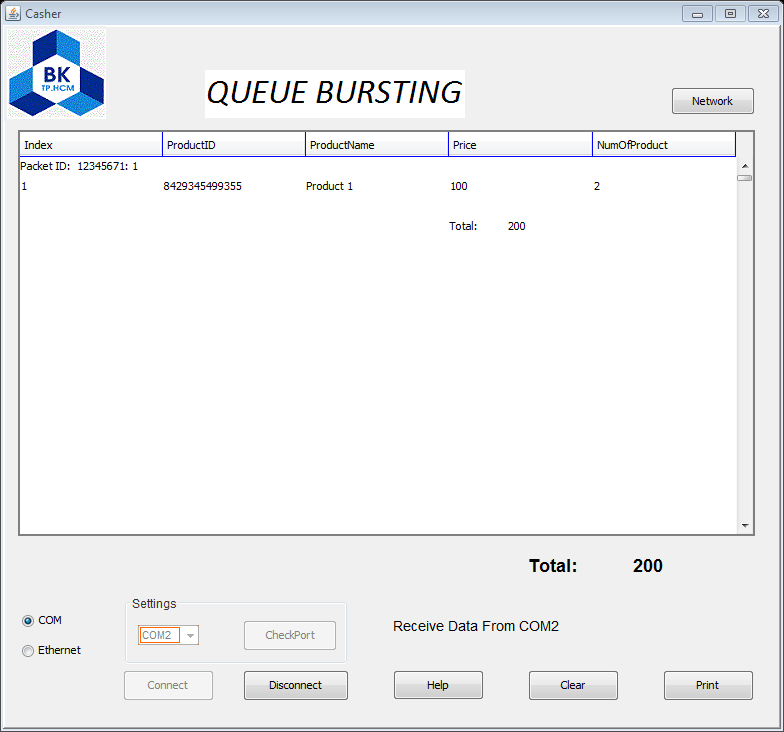
Nếu kết nối thành công nhận được thông báo :



Nếu Port không sẵn sàng kêt nối sẽ nhận được thông báo :

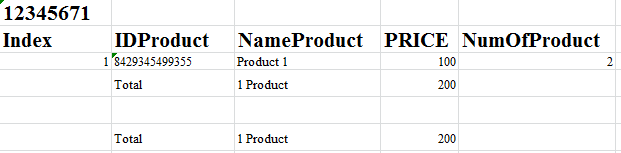


Khi kết nối thành công có thể tiến hành quét gói bình thường :



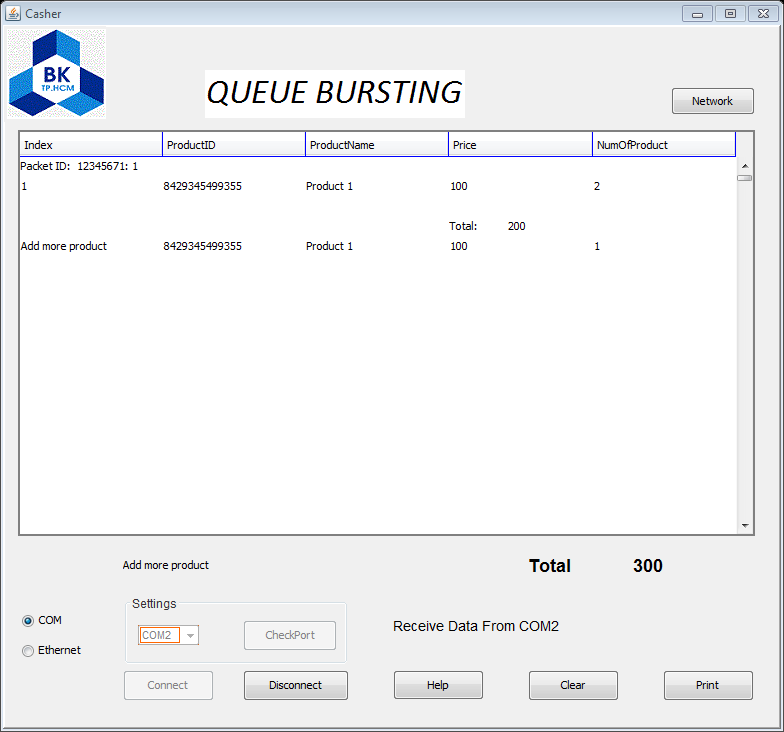
Bấm button Print để ghi xuống file và kêt thúc 1 khách hàng :

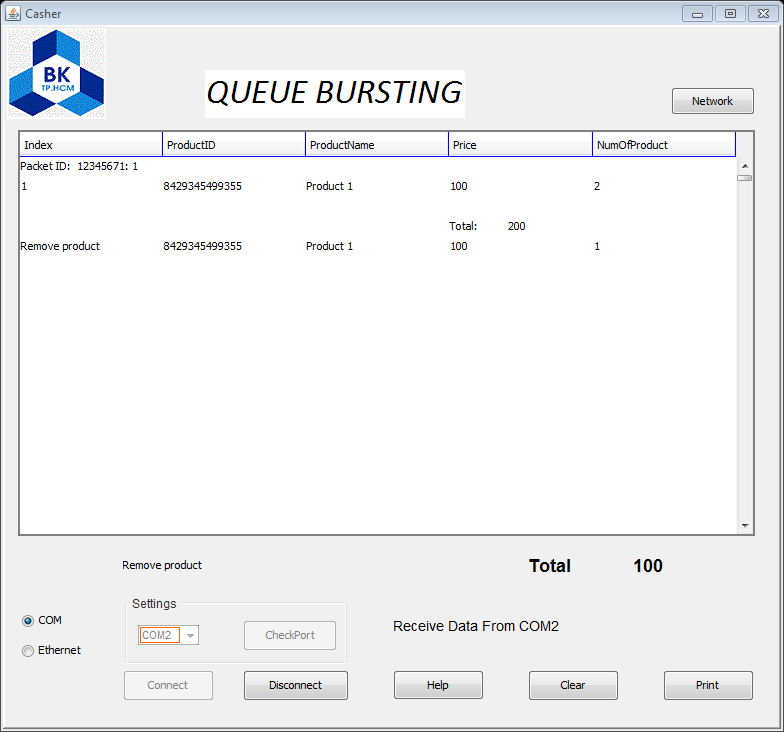
Thông tin gói hàng được lưu xuống file xls



Các tính năng khác :

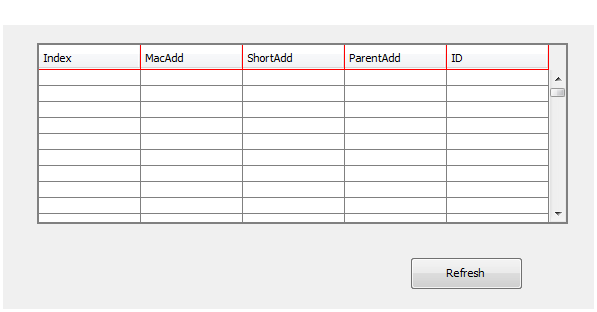
1. Thêm hoặc bớt 1 sản phẩm trong gói hàng:



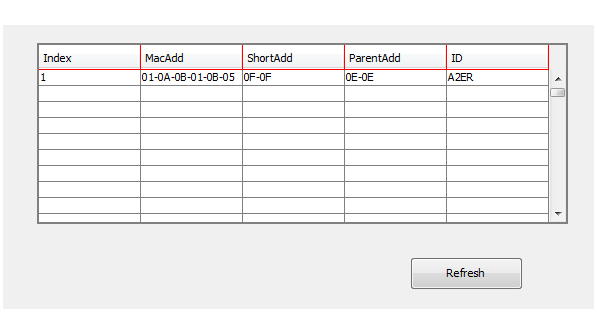


2. Kiểm tra các network hiện tại của hệ thống:

nhấn button Network :



Khi có 1 thiết bị gửi gói tin xác nhận tồn tại thì sẽ được cập nhật trên table. Các thông báo lỗi hệ thống:



# KẾT LUẬN

* *A*
* *B*
* *C*