

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO ĐỒ ÁN THIẾT KẾ MẠCH
Đề tài: NGHIÊN CỨU VÀ THỬ NGHIỆM
CÔNG NGHỆ ZIGBEE CHO SMART
HOME

Giảng viên hướng dẫn: **KS. Phan Đình Duy**

Sinh viên thực hiện:

Cao Văn Nhân - **10520355**

Nguyễn Khánh Phong - **10520477**

Tp.HCM, tháng 12 năm 2013

MỤC LỤC

| | |
|---|-----------|
| 1. Tổng quan | 1 |
| 2. Phân tích..... | 1 |
| 2.1 Giới thiệu về NUC140..... | 1 |
| 2.2 Giới thiệu sơ lược về mạng Zigbee: | 3 |
| 2.1.1 Giới thiệu:..... | 3 |
| 2.1.2 Các tính năng | 3 |
| 2.1.3 Ứng dụng | 5 |
| 2.1.4 Kiến trúc mạng của Zigbee..... | 5 |
| 2.1.5 Thành phần mạng | 7 |
| 2.1.6 Mô hình mạng..... | 8 |
| 2.3 Giới thiệu XBEE..... | 9 |
| 2.2.1 XBEE là gì? | 9 |
| 2.2.2 Cấu hình XBee..... | 11 |
| 2.4 Giới thiệu cặp IC PT2262 – PT2272M4..... | 14 |
| 2.5 Sơ đồ khối thuật toán xử lý trên NUC140 | 15 |
| 3. Hiện thực | 19 |
| 3.1 Giao diện điều khiển trên PC..... | 19 |
| 3.2 Giao diện LCD trên NUC140 | 22 |
| 3.3 Bảng các biến hàm xây dựng trên NUC140 | 23 |
| 3.4 Cấu hình XBEE model XB24-Z7WIT-004 | 26 |
| 3.5 Thiết kế schematic mạch xử lý thu phát RF | 29 |
| 4. Kết luận và hướng phát triển | 32 |
| 4.1 Kết quả đạt được | 32 |
| 4.2 Hướng phát triển | 32 |
| TÀI LIỆU THAM KHẢO..... | 33 |

DANH MỤC HÌNH

| | |
|--|----|
| Hình 1.1: Sơ đồ tổng quan đề tài..... | 1 |
| Hình 2.1 Kit Nuvoton NUC140..... | 3 |
| Hình 2.2: So sánh các kỹ thuật không dây | 4 |
| Hình 2.3: So sánh phạm vi hoạt động của Zigbee | 5 |
| Hình 2.4: Kiến trúc của ZigBee..... | 6 |
| Hình 2.5: Các loại mô hình mạng..... | 8 |
| Hình 2.6: Mô hình mạng Zigbee..... | 9 |
| Hình 2.7: XBee | 10 |
| Hình 2.8: Giao diện khởi động của X-CTU..... | 11 |
| Hình 2.9: Giao diện khởi động của X-CTU..... | 12 |
| Hình 2.10: Giao diện khởi động của X-CTU..... | 12 |
| Hình 2.11: Giao diện khởi động của X-CTU..... | 13 |
| Hình 2.12: Giao diện khởi động của X-CTU..... | 14 |
| Hình 2.13: Cặp IC PT2262/PT2272L4..... | 14 |
| Hình 2.14: Sơ đồ thực hiện trong hàm main..... | 18 |
| Hình 2.15: Sơ đồ thực hiện trong hàm xử lý ngắt UART | 19 |
| Hình 3.1: Giao diện tổng quát trên PC..... | 20 |
| Hình 3.2: Giao diện kết nối thành công trên PC | 20 |
| Hình 3.3: Chức năng setup trạng thái thiết bị | 21 |
| Hình 3.4: Chức năng làm mới lại danh sách thiết bị..... | 21 |
| Hình 3.5: Chức năng đổi mật khẩu..... | 22 |
| Hình 3.6: Giao diện LCD theo trạng thái tương ứng..... | 23 |
| Hình 3.7: Cấu hình cho COORDINATOR | 26 |
| Hình 3.8: Cấu hình cho ROUTER | 27 |
| Hình 3.9: Gửi và nhận trên Router..... | 28 |
| Hình 3.10: Gửi và nhận trên Coordinator | 29 |
| Hình 3.11: Schematic mạch phát RF | 30 |
| Hình 3.12: Schematic mạch thu RF | 31 |

DANH MỤC BẢNG

| | |
|--|-----------|
| Bảng 2.1: Bảng lệnh nhận trên NUC | 16 |
| Bảng 2.2: Bảng lệnh gửi trên NUC | 16 |
| Bảng 2.3: Bảng cấu trúc dữ liệu lưu trong flash..... | 16 |
| Bảng 2.4: Bảng dữ liệu cấu hình mặc định lưu trên flash | 17 |
| Bảng 3.1: Bảng các biến sử dụng trong chương trình..... | 24 |
| Bảng 3.2: Bảng các hàm sử dụng trong chương trình | 26 |

1. Tổng quan

Đề tài thực hiện việc giao tiếp giữa PC và NUC140, sử dụng port GPE0-15 để điều khiển.



Hình 1.1: Sơ đồ tổng quan đề tài

2. Phân tích

2.1 Giới thiệu về NUC140

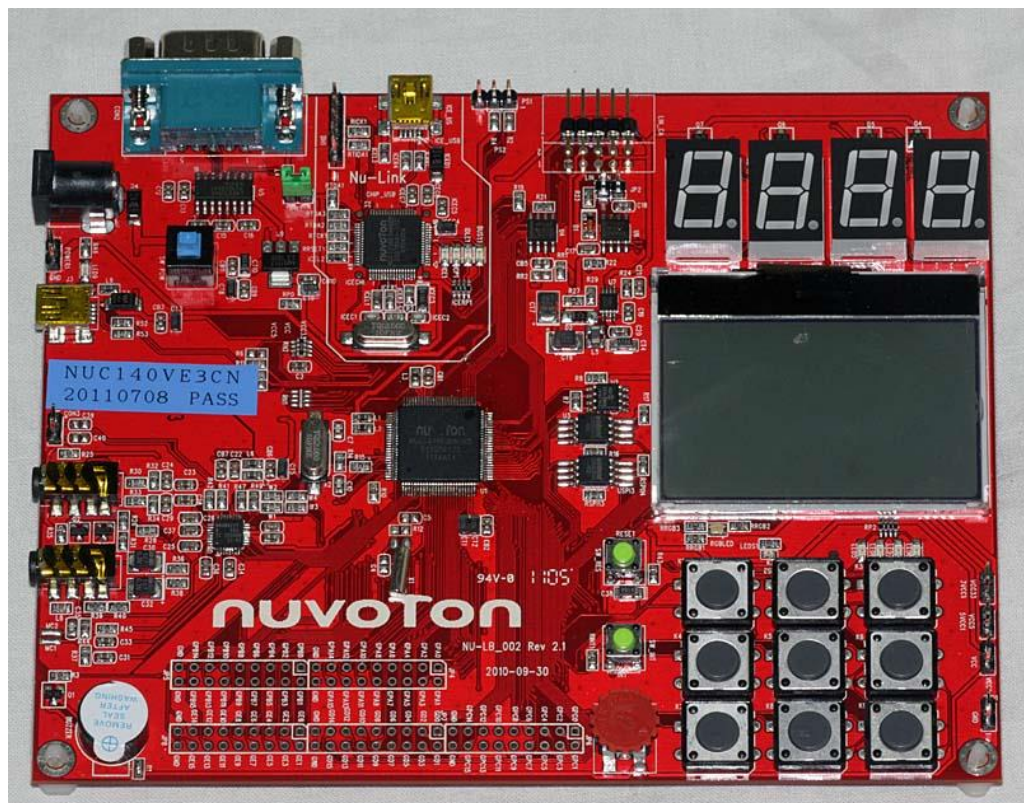
NUC140 là sản phẩm của hãng Nuvoton là công ty con được tách ra từ Tập đoàn Điện tử Winbond – một hãng điện tử bán dẫn đứng hàng đầu Đài Loan. Hãng này có 3 dòng chip vi điều khiển (MCU) 4-bit, 8-bit và 32-bit (ARM Cortex).

Dòng vi điều khiển ARM Cortex-M được thiết kế nhúng tối ưu hóa cho các ứng dụng vi xử lý MCU. Dòng ARM Cortex-M0 là dòng vi điều khiển lõi ARM có kích thước nhỏ nhất, tiêu thụ điện năng thấp nhất và có kiến trúc được sắp xếp hợp lý tương thích với việc sử dụng tools nạp của các hãng khác để phát triển các ứng dụng.

Đặc trưng của NUC140:

- NUC130/140 là vi điều khiển 32-bit lõi ARM Cortex-M0, trình đơn phần cứng 32 bit, chạy lên tới 50MHz.
- Có 4 mức ưu tiên ngắt đầu vào, có 128 KB flash ROM cho bộ nhớ chương trình.
- 16KB SRAM, 4KB bộ nhớ flash cho nạp chương trình trong hệ thống.
- Giao tiếp thiết bị ngoại vi: 8 kênh 12bit ADC, UART nối tiếp tốc độ cao, SPI lên đến 32MHz, I2C lên đến 1MHz; kết nối thiết bị ngoại vi USB 2.0, CAN, LIN...
- Thiết bị ngoại vi có tính năng phong phú: PWM, RTC, bộ ngắt nhận dạng Brownout, GPIO, PDMA và 4 bộ Timer 32 bit.
- Dải điện áp hoạt động rộng từ 2,5V~5,5V, chống nhiễu tiếng ồn tốt, tích hợp dữ liệu flash, dao động thạch anh nội chính xác $\pm 1\%$ với nhiệt độ phòng, có khả năng bảo mật trên chip, điện áp reset lại mạch thấp.

Ứng dụng tiềm năng: điều khiển mạng lưới, chẩn đoán điện tử, ứng dụng mạng nhúng, hệ thống điều khiển mạng, điều khiển công nghiệp và tự động điều khiển.



Hình 2.1 Kit Nuvoton NUC140.

2.2 Giới thiệu sơ lược về mạng Zigbee:

2.1.1 Giới thiệu:

ZigBee là một giao thức truyền thông bậc cao được phát triển dựa trên chuẩn truyền thông không dây IEEE 802.15.4, sử dụng tín hiệu radio cho các mạng cá nhân PAN (personal area network).

ZigBee thích hợp với những ứng dụng không đòi hỏi tốc độ truyền dữ liệu quá cao nhưng cần có mức độ bảo mật lớn và thời gian hoạt động dài. Các mạng ad-hoc sử dụng sóng radio tương tự ZigBee đã được thai nghén từ những năm 1998-1999 khi giới khoa học bắt đầu nhận thấy Wifi và Bluetooth không phù hợp cho nhiều ứng dụng công nghiệp. Tuy nhiên chỉ đến năm 2004, bộ tiêu chuẩn ZigBee mới chính thức được tạo dựng và thông qua bởi tổ chức ZigBee Alliance.

Tên gọi ZigBee lấy cảm hứng từ điệu nhảy theo đường zig-zag của ong mật (honey bee), điệu nhảy này được loài ong sử dụng để trao đổi thông tin với nhau về vị trí của hoa và nguồn nước.

2.1.2 Các tính năng

- **Sử dụng đơn giản, giá thành hợp lý, tiết kiệm năng lượng**

Thiết bị được sản xuất phù hợp cho cả người sử dụng tự lắp đặt hay các nhà tích hợp hệ thống chuyên nghiệp. Tối ưu hóa năng lượng, giảm hao phí điện năng khi sử dụng.

Thiết bị sử dụng tiêu chuẩn mở phù hợp với mọi thiết bị điện - điện tử trên thị trường.

- **Dễ dàng điều khiển**

Công nghệ không dây làm giảm chi phí và những rắc rối của mạng có dây truyền thống. Sử dụng tần số quốc tế 2.4 Ghz dễ dàng lắp đặt và sử dụng.

Tính năng điều khiển tự động hoặc bán tự động: Giải thoát sức lao động của con người. Kết nối Internet cho phép điều khiển từ xa. Tự lắp đặt, tự cài đặt dễ dàng.

- **An toàn**

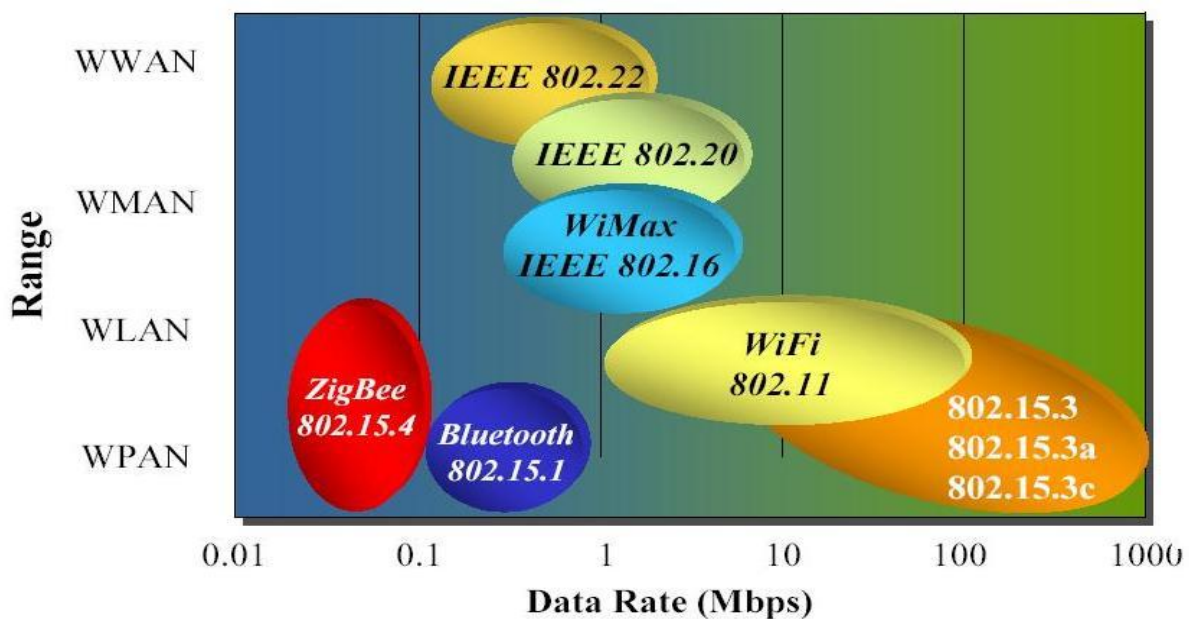
Dễ dàng lắp đặt cảm biến không dây để giám sát an ninh ngôi nhà. Nhận thông báo tức thì khi có sự kiện bất thường xảy ra.

AES hệ thống không dây được mã hóa đặc biệt, đảm bảo chỉ duy nhất chủ nhà có khả năng điều khiển hệ thống

- **Liên kết hoạt động**

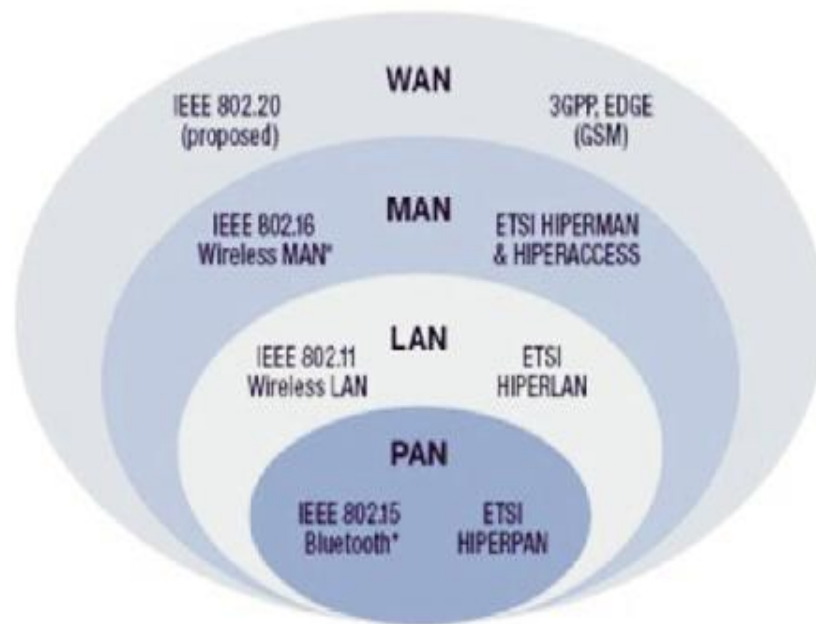
Tích hợp điều khiển và giám sát các phân hệ điện của ngôi nhà cũng như các hệ an ninh, kiểm soát truy nhập...

Vì các ứng dụng đều được xây dựng ở dạng module do đó người sử dụng chỉ phải mua những thiết bị mà mình cần. Có thể kết hợp sử dụng nhiều dòng sản phẩm mà không cần quan tâm tới nhà sản xuất có thể làm việc với mạng ZigBee khác



Hình 2.2: So sánh các kỹ thuật không dây

Chuẩn truyền thông không dây



Hình 2.3: So sánh phạm vi hoạt động của Zigbee

2.1.3 Ứng dụng

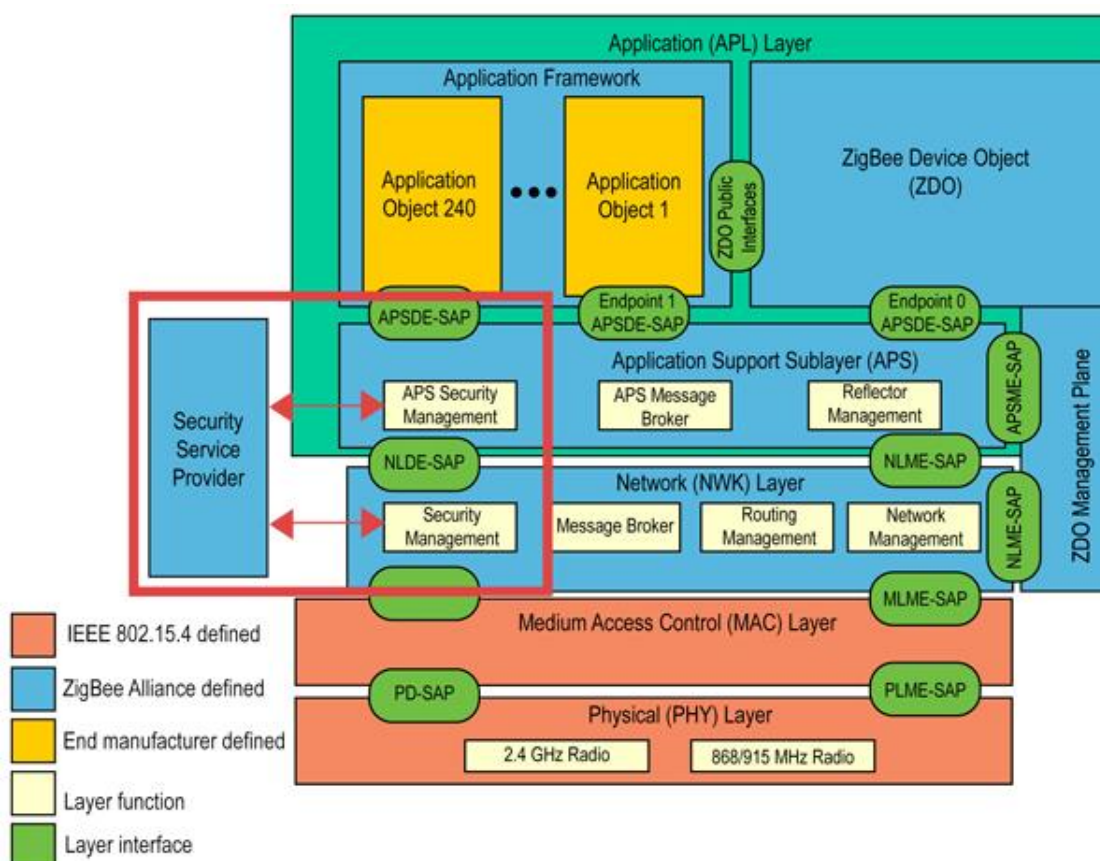
- Tần số chung toàn cầu 2.4 GHz
- Pin sử dụng có tuổi thọ lớn
- Khoảng cách không dây 70m trong nhà, 400m ngoài nhà
- Việc kết nối mạng linh hoạt đảm bảo đáp ứng mọi quy mô của ngôi nhà
- Khả năng mở rộng tới hàng ngàn thiết bị
- Tích hợp khả năng điều khiển và giám sát trạng thái hoạt động của các hệ chiếu sáng, an ninh, rèm cửa, bơm nước, bình nóng lạnh, điều hòa....

2.1.4 Kiến trúc mạng 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.



Hình 2.4: 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 thông 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 responses. 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.

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.

2.1.5 Thành phần mạng

Một mạng kiểu ZigBee gồm có 3 loại thiết bị:

- * ZC (Zigbee Coordinator): đây là thiết bị gốc có khả năng quyết định kết cấu mạng, quy định cách đánh địa chỉ và lưu giữ bảng địa chỉ. Mỗi mạng chỉ có duy nhất một Coordinator và nó cũng là thành phần duy nhất có thể truyền thông với các mạng khác.

- * ZR (Zigbee Router): có các chức năng định tuyến trung gian truyền dữ liệu, phát hiện và lập bản đồ các nút xung quanh, theo dõi, điều khiển, thu thập dữ liệu như nút bình thường. Các router thường ở trạng thái hoạt động (active mode) để truyền thông với các thành phần khác của mạng.

- * ZED (Zigbee End Device): các nút này chỉ truyền thông với Coordinator hoặc Router ở gần nó, chúng được coi như điểm cuối của mạng và chỉ có nhiệm vụ hoạt động/đọc thông tin từ các thành phần vật lý. ZED có kết cấu đơn giản và thường ở

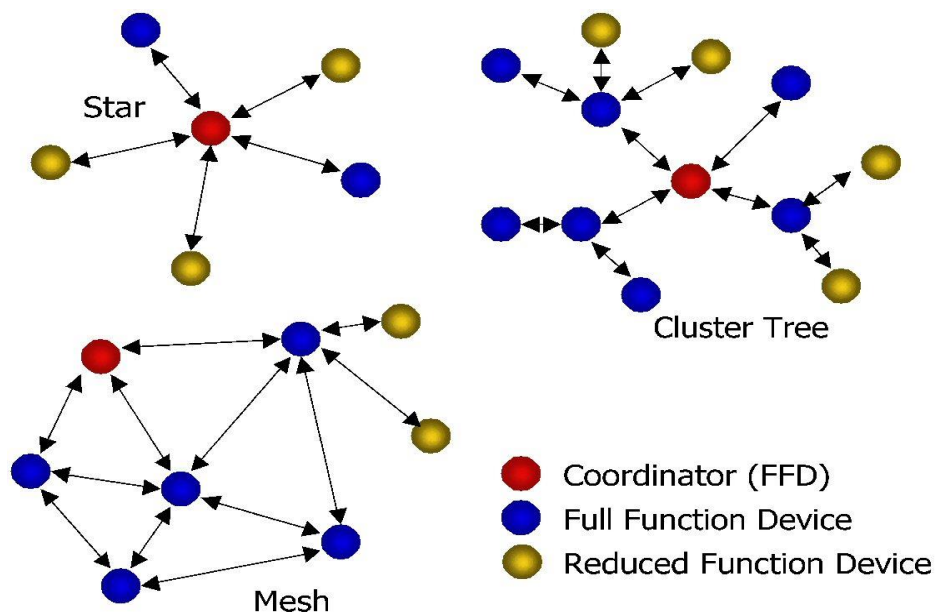
trạng thái nghỉ (sleep mode) để tiết kiệm năng lượng. Chúng chỉ được "đánh thức" khi cần nhận hoặc gửi một thông điệp nào đó.

Các thiết bị này thường được chia làm 2 loại là FFD (Full Function Device) và RFD (Reduced Function Device). Trong đó FFD có thể hoạt động như một Coordinator, Router hoặc End Device, còn RFD chỉ có thể đóng vai trò End Device trong một mạng ZigBee.

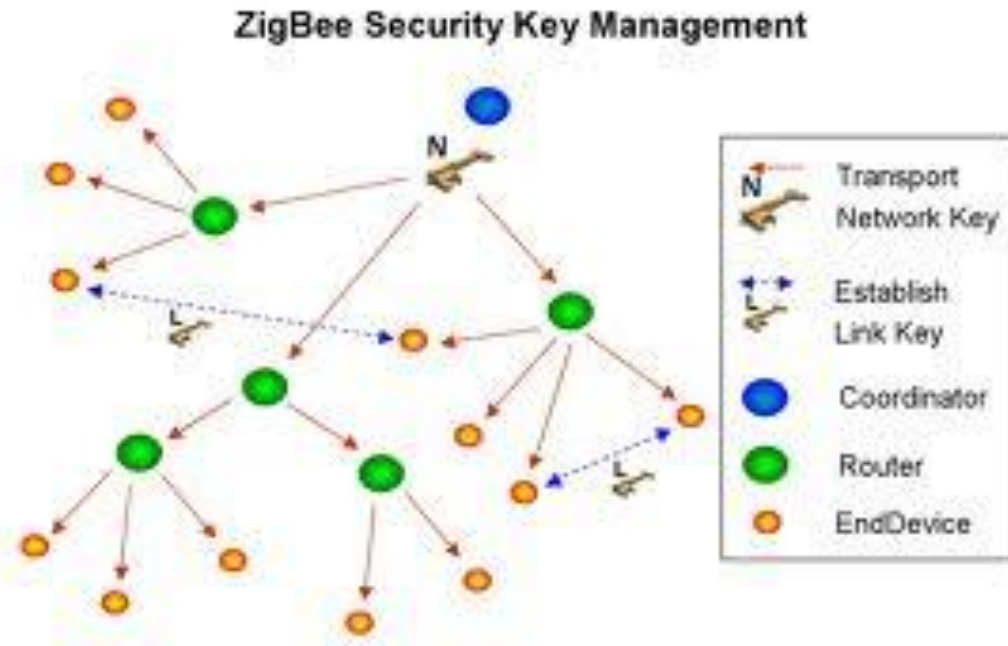
2.1.6 Mô hình mạng

Có 3 cấu hình mạng cơ bản, tùy vào từng ứng dụng cụ thể mà người ta thiết lập mạng theo các cấu hình khác nhau:

- + Mạng hình sao (star network)
- + Mạng hình lưới (mesh network)
- + Mạng hình cây (cluster tree topology)



Hình 2.5: Các loại mô hình mạng.



Hình 2.6: Mô hình mạng Zigbee

2.3 Giới thiệu XBEE

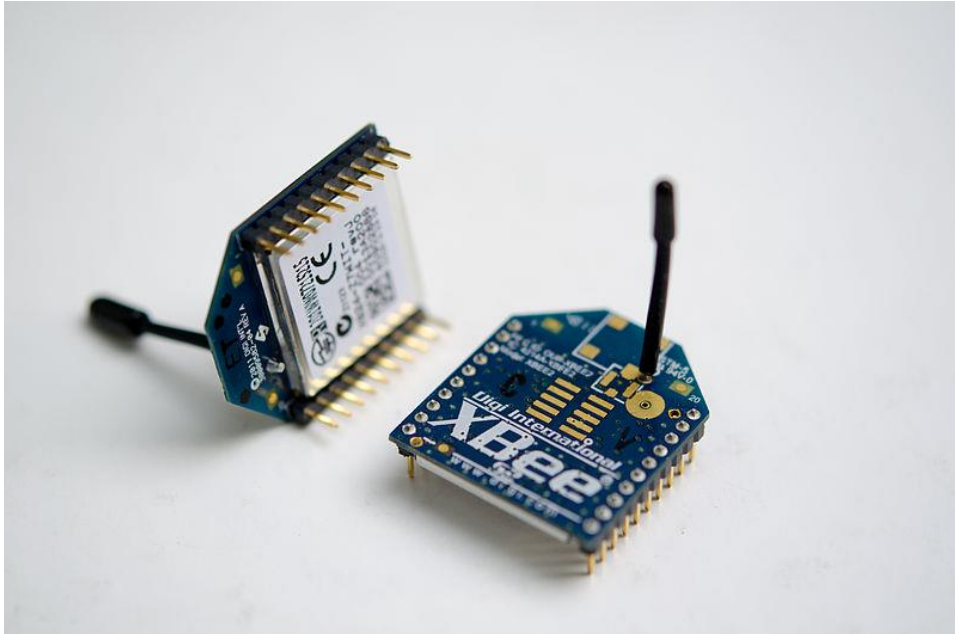
2.2.1 XBEE là gì?

XBee là một thương hiệu từ Digi International. Radio XBee đầu tiên đã được giới thiệu vào năm 2005 và được dựa trên các tiêu chuẩn thiết kế 802.15.4-2003 point-to-point và over-the-air tốc độ truyền 250 kbit / s.

Hai kiểu ban đầu được giới thiệu-với chi phí thấp 1 mW XBee và công suất cao hơn 100 mW XBee-PRO. Kể từ khi lần giới thiệu đầu, một số radio XBee mới đã được giới thiệu và tất cả XBees đang bán trên thị trường và được bán dưới thương hiệu Digi.

Radio XBee tất cả đều có được sử dụng với số lượng tối thiểu là 4 của các kết nối - điện (3,3 V), đất, dữ liệu vào và dữ liệu ra (UART), với đường dây khác được đề nghị là Reset và Sleep. Ngoài ra, hầu hết các gia đình XBee có những kiểm soát lưu lượng khác, I / O, A / D và các dòng chỉ được xây dựng nhập.

Một phiên bản của XBees gọi là XBee có thể lập trình có một bộ xử lý khác tích hợp cho code của người dùng. XBee có thể lập trình và một bề mặt mới lắp (SMT) phiên bản của các radio XBee cả hai đã được giới thiệu trong năm 2010.



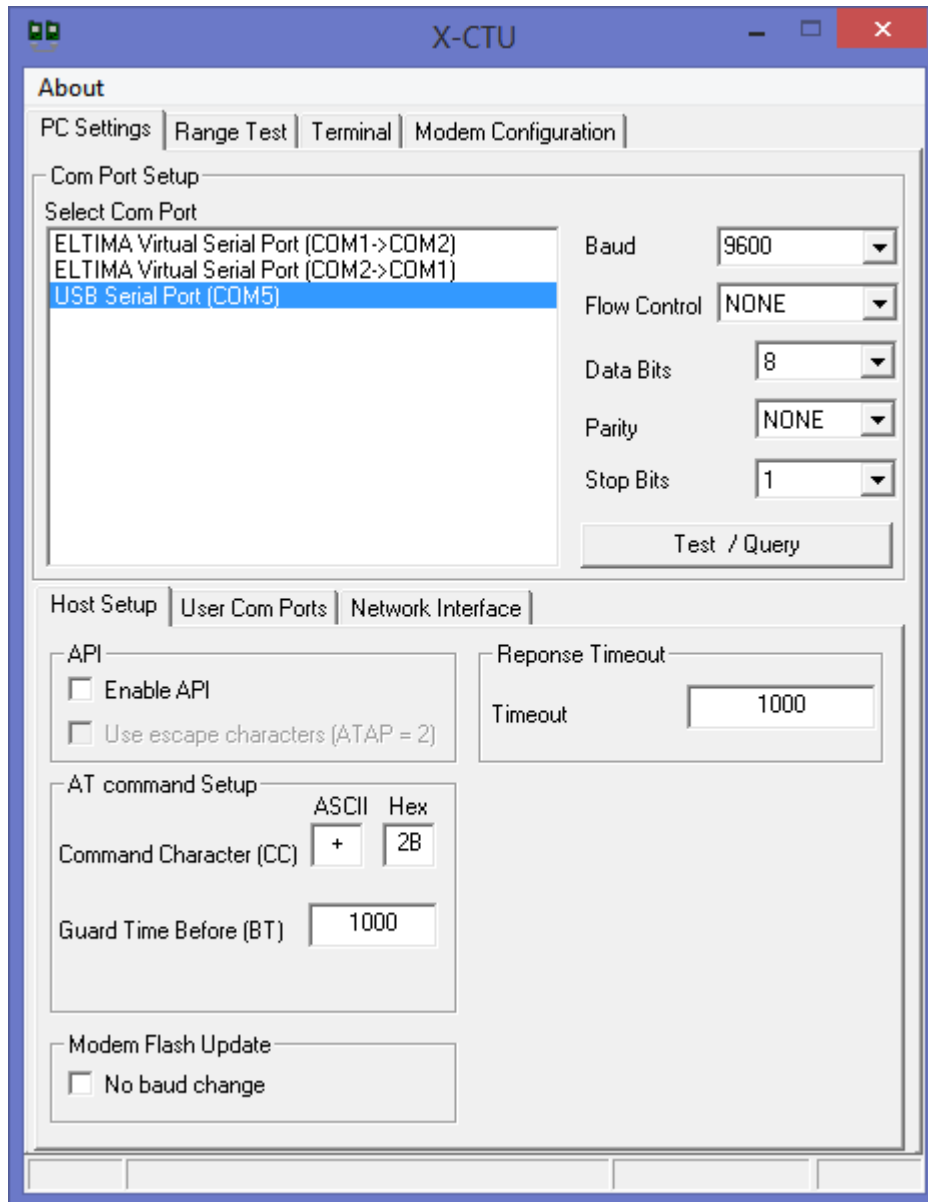
Hình 2.7: XBee

Các loại XBee:

- XBee 802.15.4
- XBee-PRO 802.15.4
- XBee DigiMesh 2.4
- XBee-PRO DigiMesh 2.4
- XBee ZB
- XBee-PRO ZB
- XBee ZB SMT
- XBee-PRO ZB SMT
- XBee SE
- XBee-PRO SE
- XBee-PRO
- XBee-PRO 900 (Legacy)
- XBee-PRO XSC (S3B)

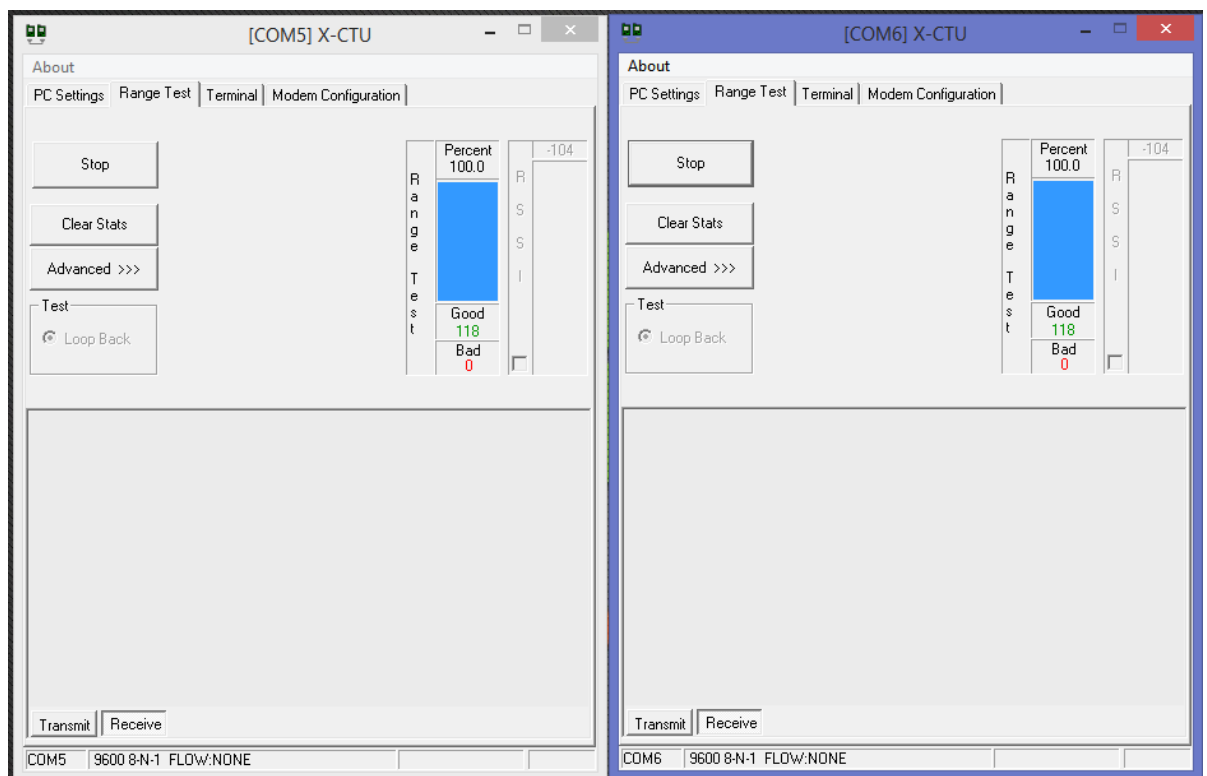
2.2.2 Cấu hình Xbee

Chính vì XBee mang thương hiệu Digi International nên để cấu hình XBee cũng sử dụng chương trình mang tên X-CTU được cung cấp từ hãng.



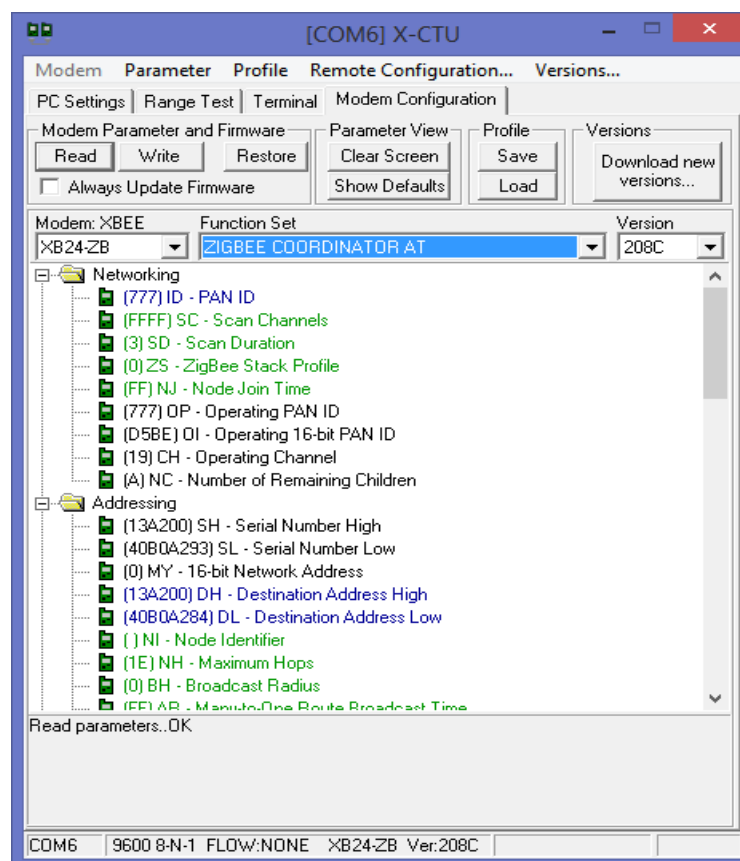
Hình 2.8: Giao diện khởi động của X-CTU

Khi kết nối một thiết bị XBee vào computer thì chương trình sẽ nhận ra thiết bị ở dòng “USB Serial Port (COM5)” trong hình. Bên cạnh đó chương trình còn thể hiện các thông số đặc trưng cho chế độ truyền dữ liệu nối tiếp như baud, data bits, stop bits, ... Sau đó là các tab Range Test để kiểm tra khoảng cách kết nối giữa các thiết bị.



Hình 2.9: Giao diện khởi động của X-CTU

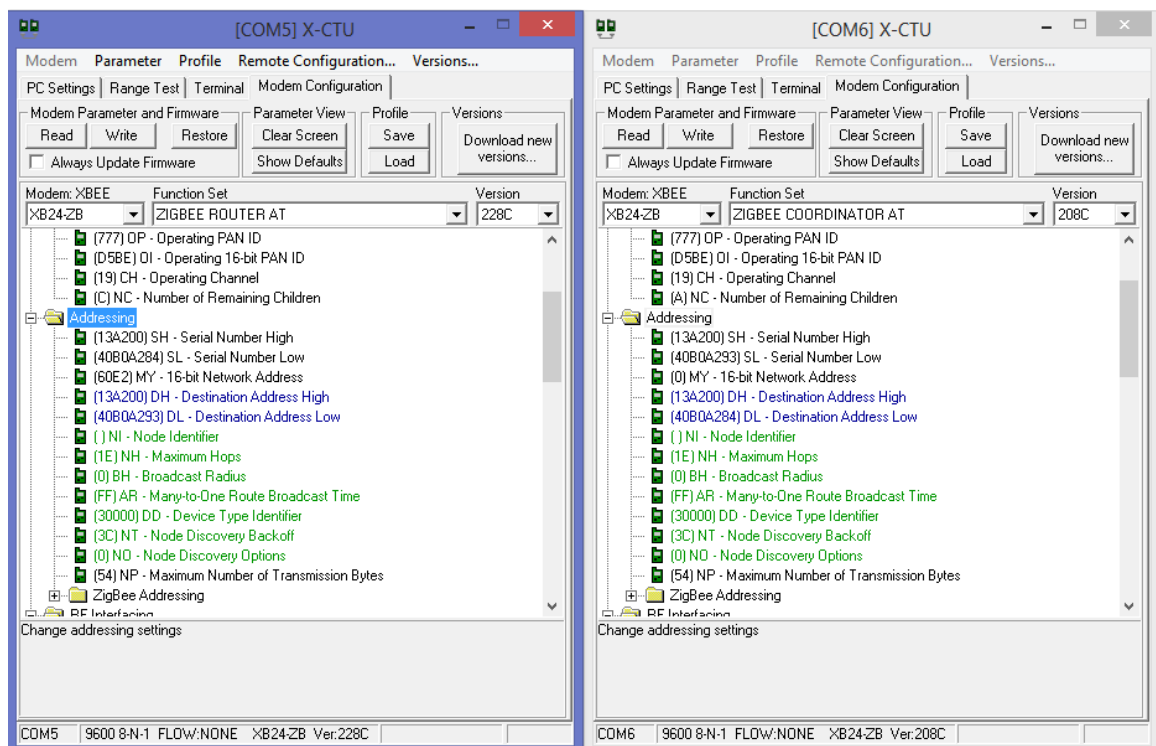
Tab Modem Configuration để cấu hình cho XBee. Sau khi bấm “Read” thì các thông số về XBee đó sẽ được đọc và thể hiện ra.



Hình 2.10: Giao diện khởi động của X-CTU

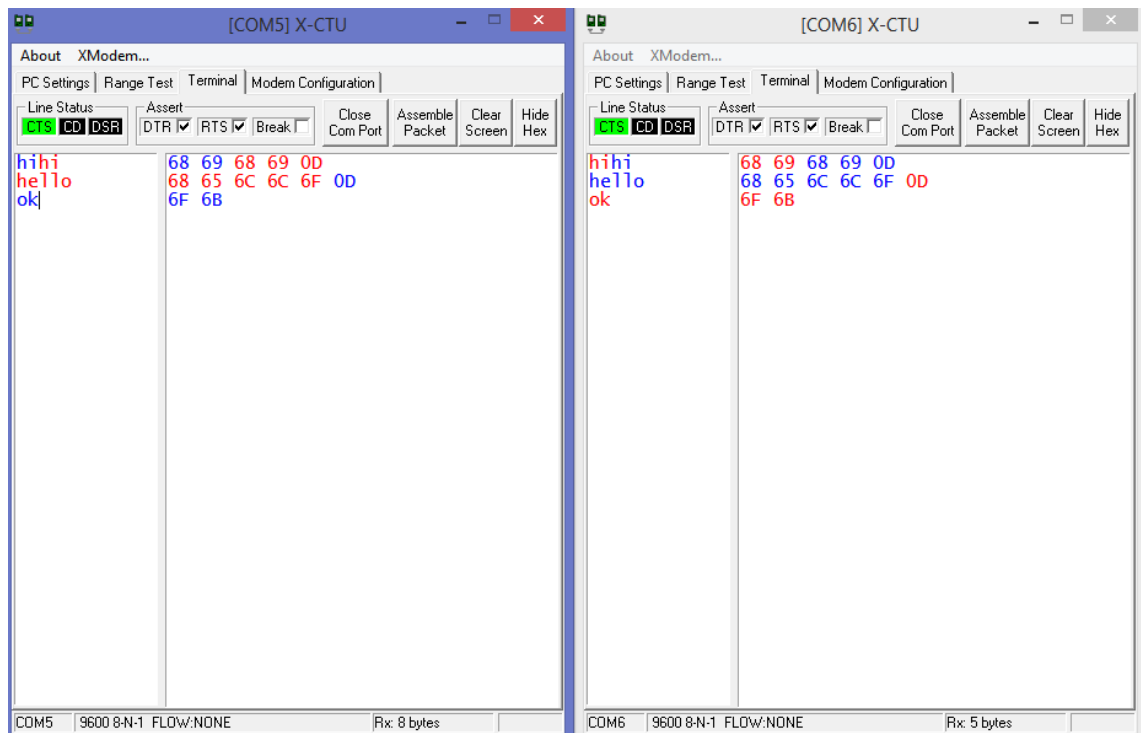
Mỗi loại XBee thuộc các modem khác nhau và các version khác nhau. Nếu chương trình không nhận ra được XBee đó thì phải cập nhật các phiên bản mới nhất. Trong phần Function Set thì xác lập vai trò và kiểu giao tiếp của XBee. XBee có thể đóng vai trò là coordinator hay router hay end device và các kiểu giao tiếp là at, api, analog io hoặc digital io tùy thuộc vào vai trò của XBee.

Cấu hình một thiết bị lò coordinator at và một thiết bị là router at. Tạo ID mạng là 777 (có thể chọn số khác trong vòng 16bit) và các thiết bị nào muốn tham gia mạng này phải đặt ID giống với coordinator. Ở phần addressing phần địa chỉ đích sẽ là tương ứng giữa 2 thiết bị. Sau đó chọn “Write” để ghi cấu hình xuống thiết bị.



Hình 2.11: Giao diện khởi động của X-CTU

Sau khi cấu hình xong có thể chuyển sang tab Terminal để kiểm tra thiết lập.



Hình 2.12: Giao diện khởi động của X-CTU

2.4 Giới thiệu cặp IC PT2262 – PT2272M4



Hình 2.13: Cặp IC PT2262/PT2272L4

PT2262 và PT2272 là sản phẩm của Princeton Technology được phát triển và ra đời sau dòng mã hóa 12E/D của hãng Holtek. PT2262 có 2 loại chính : loại có 8 địa chỉ mã hóa , 4 địa chỉ dữ liệu và loại có 6 địa chỉ mã hóa và 6 địa chỉ dữ liệu. Mã hóa 12 bit 1 khung A0 đến A7, D0 đến D3.

Các linh kiện PT2262 đưa vào việt Nam chỉ có loại PT2262 với 8 địa chỉ mã hóa và 4 địa chỉ dữ liệu. Tương tự với PT2262 có 2 kiểu thì PT2272 cũng có 2 kiểu:

PT2272 có 8 địa chỉ giải mã và 4 dữ liệu đầu ra thường được kí hiệu: T2272 - L4

PT2272 có 6 địa chỉ giải mã và 6 giữ liệu ra: kí hiệu PT2272 - L6 .Loại L4 là thông dụng ở việt nam và ít có loại L6.

PT2262 có 3^{12} mã hóa tức là có thể mã hóa 531441 mã mới có thể trùng lặp lại. So với HT12E chỉ có 2^{12} mã hóa thì trội hơn hẳn về cái kiểu mật mã này. (HT12E chỉ có 2 mũ 12 mã hóa).

Cách mã hóa PT2262 có thể làm được bằng cách nối ngắn mạch các chân mã hóa địa chỉ lên dương nguồn (mã hóa +) và xuống âm nguồn (mã hóa -) hoặc có thể bỏ trống.

Dữ liệu và mã hóa được truyền trên một khung 12 bit gồm 8 bit đầu là mã hóa A0 đến A7 và 4 bit dữ liệu . Bởi vậy có thể truyền được song song 4 bit dữ liệu 0 hoặc 1. Nếu để truyền dữ liệu thì nên để mặc định cho 4 chân dữ liệu này là 0 hoặc là 1 bằng cách nối thêm điện trở kéo lên hoặc đưa xuống GND để tránh nhiễu.

PT2262 dùng dao động ngoài, đơn giản là chỉ cần lắp thêm 1 điện trở dao động vào chân 15 và chân 16 của PT2262. Tín hiệu encoder được đưa ra ở chân 17 của PT2262, chân này thường ở mức 1 khi tín hiệu nghỉ và mức 0 khi tín hiệu hoạt động. Tín hiệu đưa ra gồm : sóng mang dao động $< 700\text{KHz}$ + địa chỉ mã hóa + dữ liệu. Tần số sóng mang dao động được quyết định bởi R chân 15 và 16 và được tính bằng : $f = R/12$. Ví dụ : mắc điện trở 470k vào chân 15 và 16 đầu ra chân 17 sẽ có $470/12 =$ khoảng 39Khz. PT2262 có điện áp rộng, có thể làm việc được từ 2,5V đến 15 V .

PT2272 là con giải mã của PT2262 nó cũng có 8 địa chỉ giải mã tương ứng + 4 bit dữ liệu ra + 1 chân báo hiệu mã đúng(chân 17). Cách giải mã như sau: Chân 15 và 16 cũng cần một điện trở để làm dao động giải mã . Trong dải hồng ngoại hoặc dưới 100 KHz có thể dùng R rất lớn hoặc không cần. Nhưng từ khoảng 100 KHz dao động trở lên thì bắt buộc phải dùng R để tạo dao động cho PT2272.

Giá trị R của PT2272 sẽ bằng khoảng : (Giá trị R của PT2262) chia cho 10. Ví dụ: PT2262 mắc điện trở 4,7 megaom thì PT2272 sẽ mắc 470k.

2.5 Sơ đồ khối thuật toán xử lý trên NUC140

Để giao tiếp giữa 2 thiết bị là PC và NUC140, chúng em xây dựng phương thức giao tiếp qua lại như sau:

| Bảng lệnh nhận trên NUC | | | | |
|-------------------------|-----------|-----------|-----|------------------|
| Mã lệnh | Tham số 1 | Tham số 2 | End | Chú thích |
| 0 | Tên PC | Pass | # | Khởi tạo kết nối |
| 1 | Pass mới | | # | Đổi pass |
| 2 | pin | status | # | Set pin |
| 3 | | | # | Hủy kết nối |

| | | | | |
|---|--|--|---|--------------|
| 4 | | | # | Lưu cấu hình |
| 5 | | | # | Send List |

Bảng 2.1: Bảng lệnh nhận trên NUC

| Bảng lệnh gửi trên NUC | | | | |
|-------------------------------|-----------|-----------|-----|-----------------|
| Mã lệnh | Tham số 1 | Tham số 2 | End | Chú thích |
| 0 | | | # | Kết nối ok |
| 1 | pin | status | # | Gửi list device |
| 2 | | | # | Cập nhật ok |
| 3 | | | # | Hủy kết nối |

Bảng 2.2: Bảng lệnh gửi trên NUC

Trong hàm main, có 5 stage:

- ✚ Stage 0: Trạng thái ban đầu, chờ nhấn phím để thực hiện các lệnh khác
- ✚ Stage 1: Cài đặt lại cấu hình mặc định và ghi vào flash
- ✚ Stage 2: Kiểm tra trạng thái các thiết bị có cập nhật đúng không
- ✚ Stage 3: Lưu cấu hình hiện tại vào flash
- ✚ Stage 4: Giai đoạn để trở về giao diện ban đầu là stage 0
- ✚ Stage 5: Giai đoạn xử lý sau khi kiểm tra kết nối thành công, nhằm giảm thiểu thời gian thực hiện trong hàm xử lý ngắt

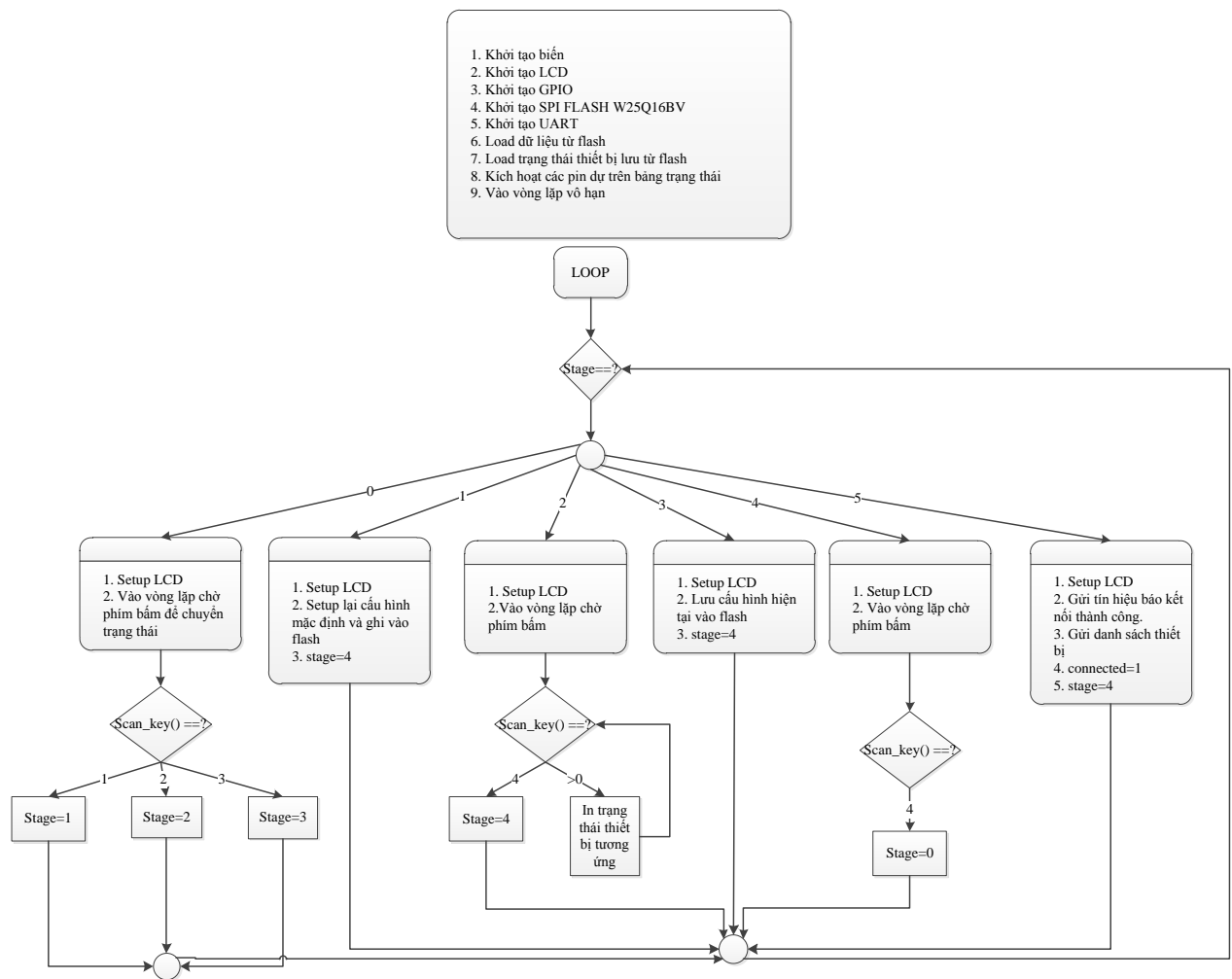
| Cấu trúc của 1 page info of device | | |
|---|---------------------|-----------------------|
| status#####.....## | | |
| | | |
| Bảng dữ liệu bộ nhớ: | | |
| Page | Content | Address = Page*256 |
| 0 | Password | |
| 1 | Number of device=16 | |
| 2 | Info device 1 | |
| 3 | Info device 2 | |
| ... | ... | |
| 17 | Info device 16 | |

Bảng 2.3: Bảng cấu trúc dữ liệu lưu trong flash

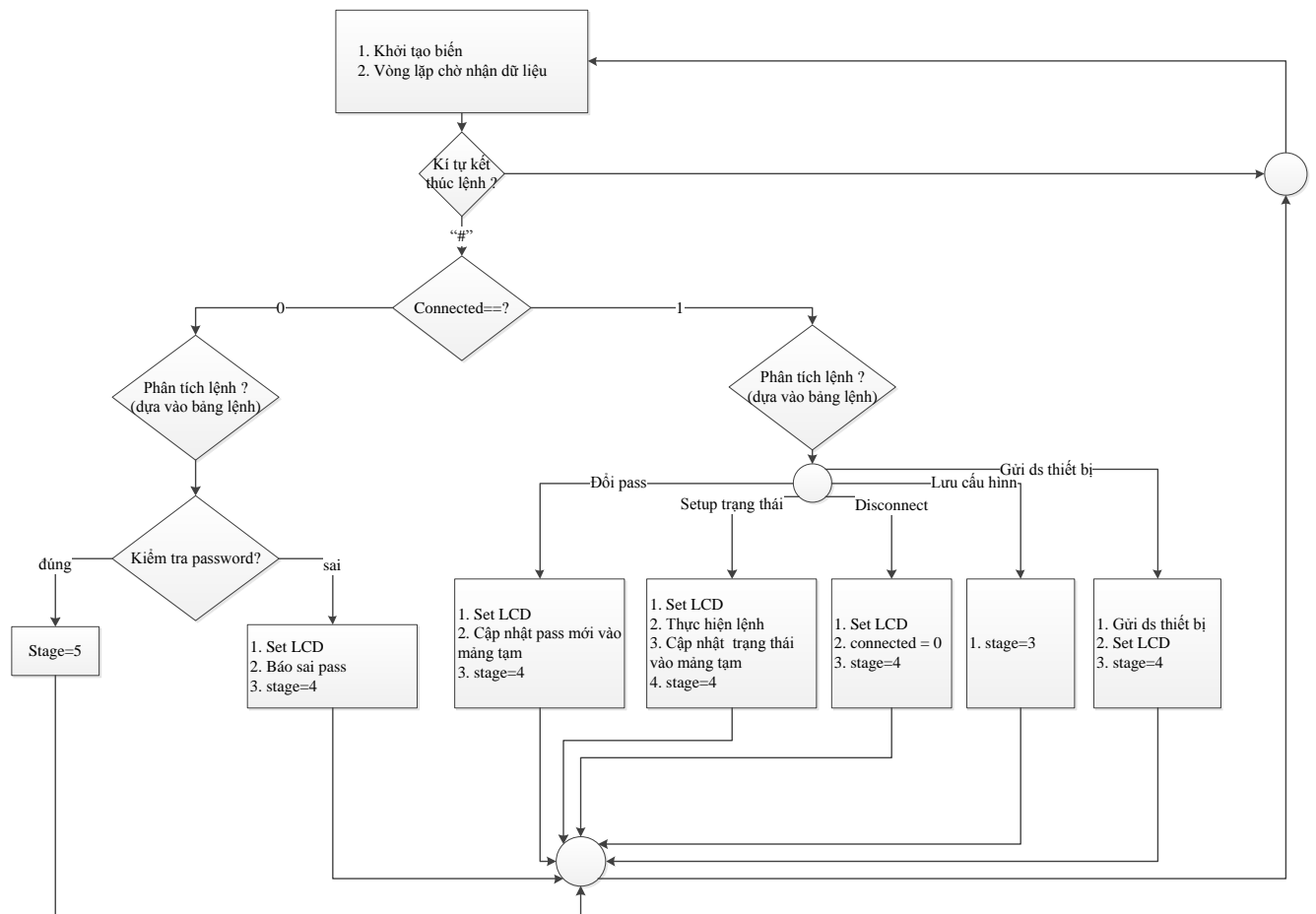
| Mặc định dữ liệu chứa trong flash |
|--|
|--|

| Page | Nội dung | Chú thích |
|------|----------|------------------------|
| 0 | 123 | password |
| 1 | 16 | numberdevice |
| 2 | 0# | device1(status###...) |
| 3 | 0# | device2(status###...) |
| 4 | 0# | device3(status###...) |
| 5 | 0# | device4(status###...) |
| 6 | 0# | device5(status###...) |
| 7 | 0# | device6(status###...) |
| 8 | 0# | device7(status###...) |
| 9 | 0# | device8(status###...) |
| 10 | 0# | device9(status###...) |
| 11 | 0# | device10(status###...) |
| 12 | 0# | device11(status###...) |
| 13 | 0# | device12(status###...) |
| 14 | 0# | device13(status###...) |
| 15 | 0# | device14(status###...) |
| 16 | 0# | device15(status###...) |
| 17 | 0# | device16(status###...) |

Bảng 2.4: Bảng dữ liệu cấu hình mặc định lưu trên flash



Hình 2.14: Sơ đồ thực hiện trong hàm main



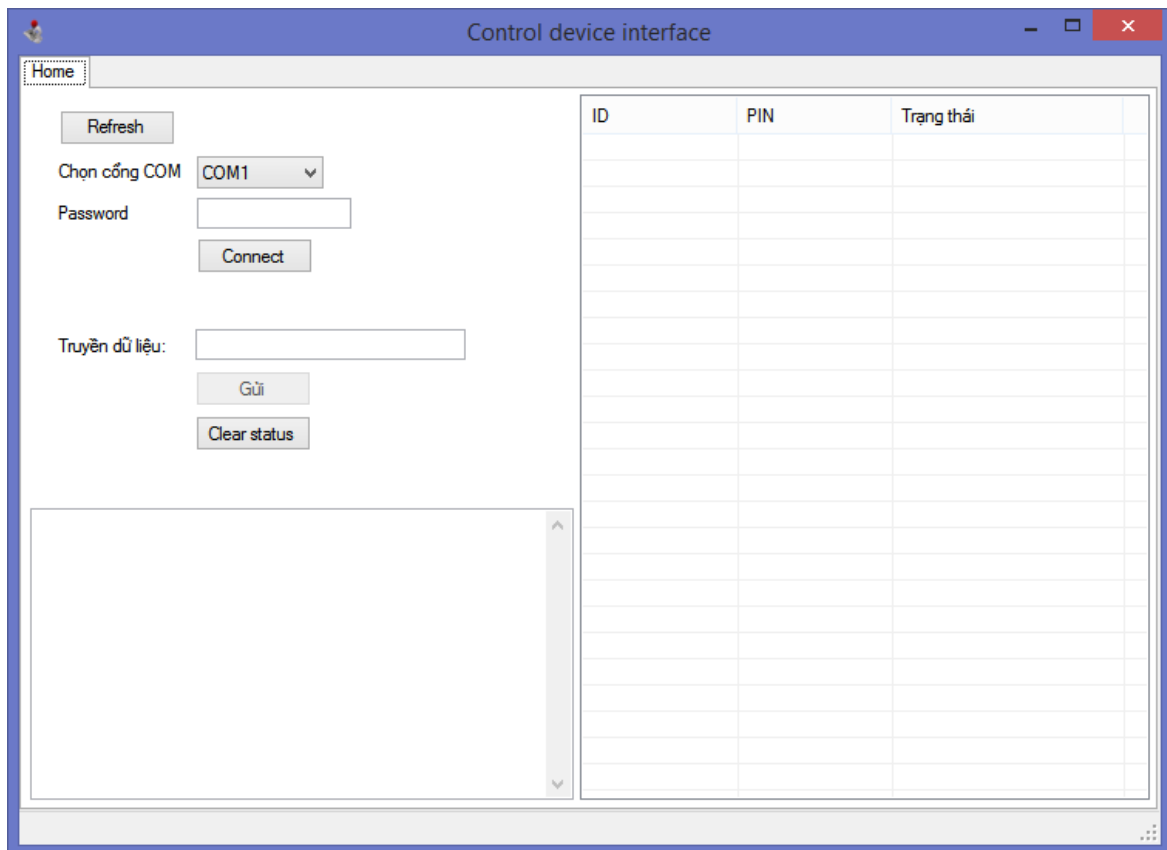
Hình 2.15: Sơ đồ thực hiện trong hàm xử lý ngắt UART

3. Hiện thực

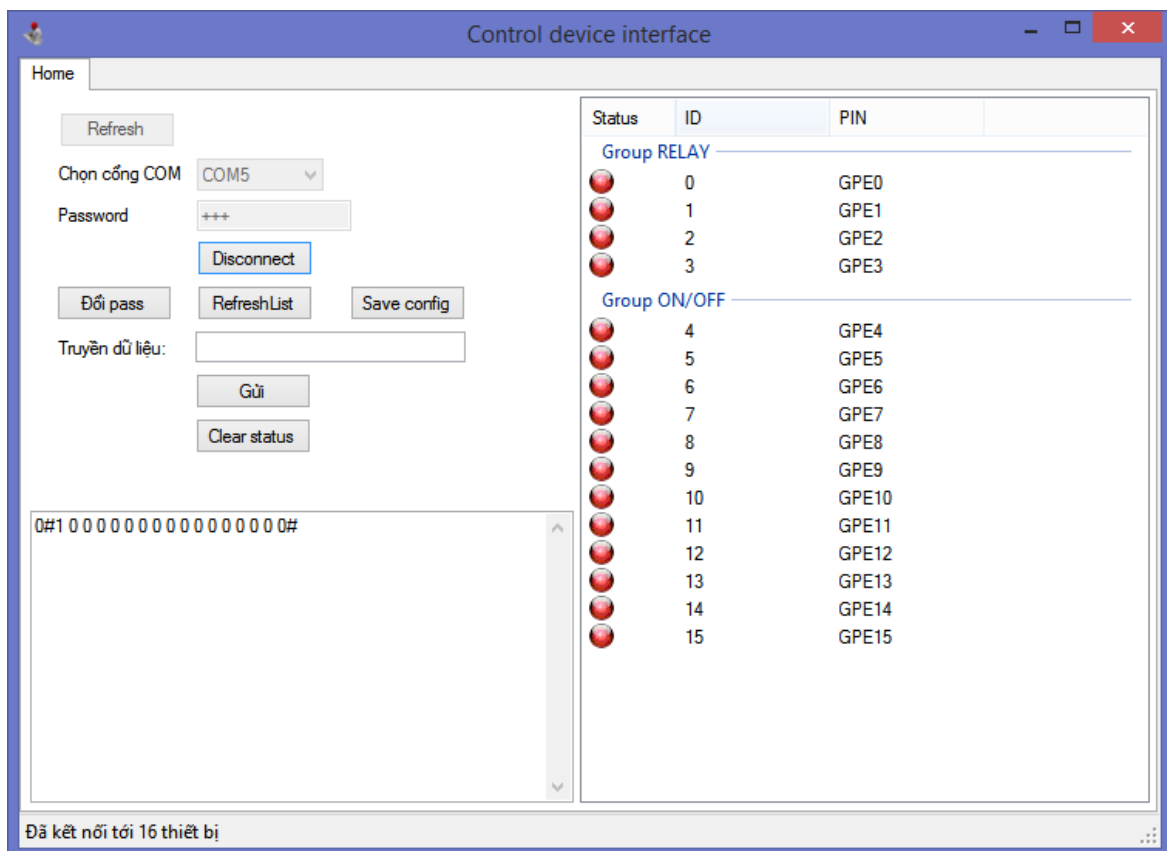
3.1 Giao diện điều khiển trên PC

Khi đăng nhập vào hệ thống thành công thì trên phần mềm sẽ được sử dụng các chức năng sau:

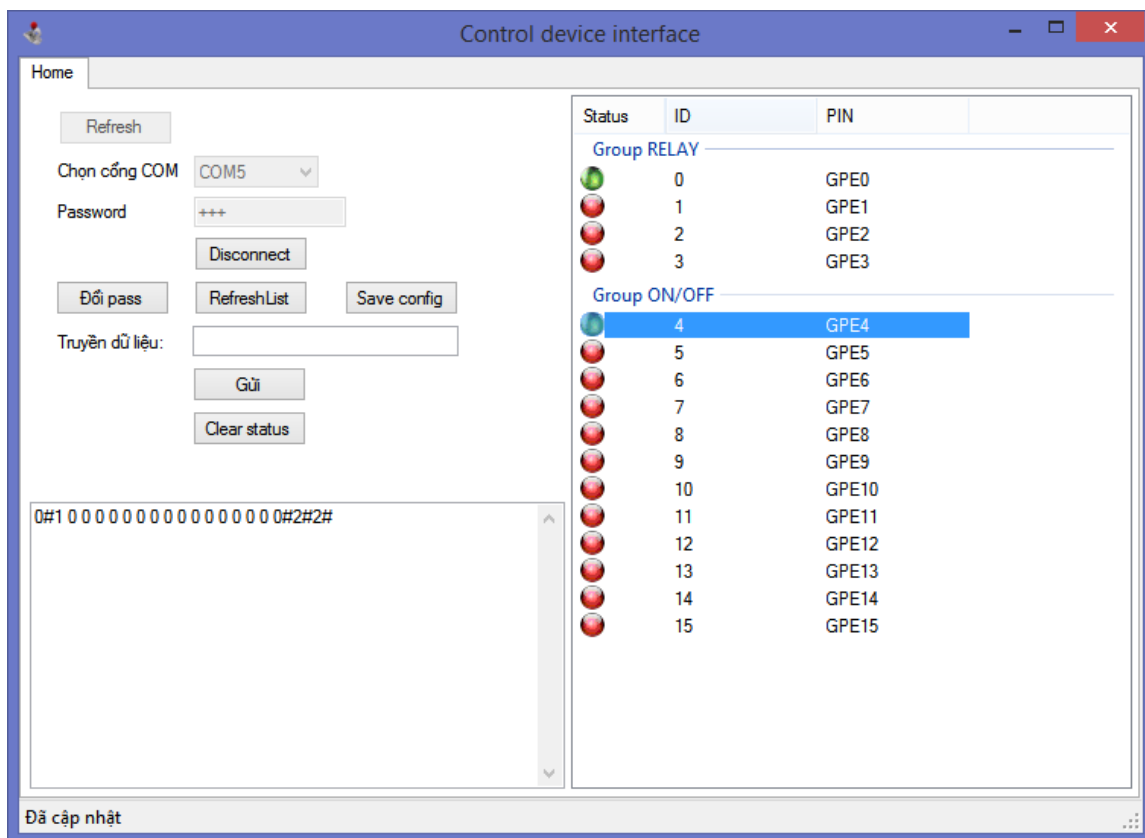
- Đổi mật khẩu
- Lưu cấu hình
- Làm mới danh sách thiết bị
- Điều chỉnh trạng thái thiết bị



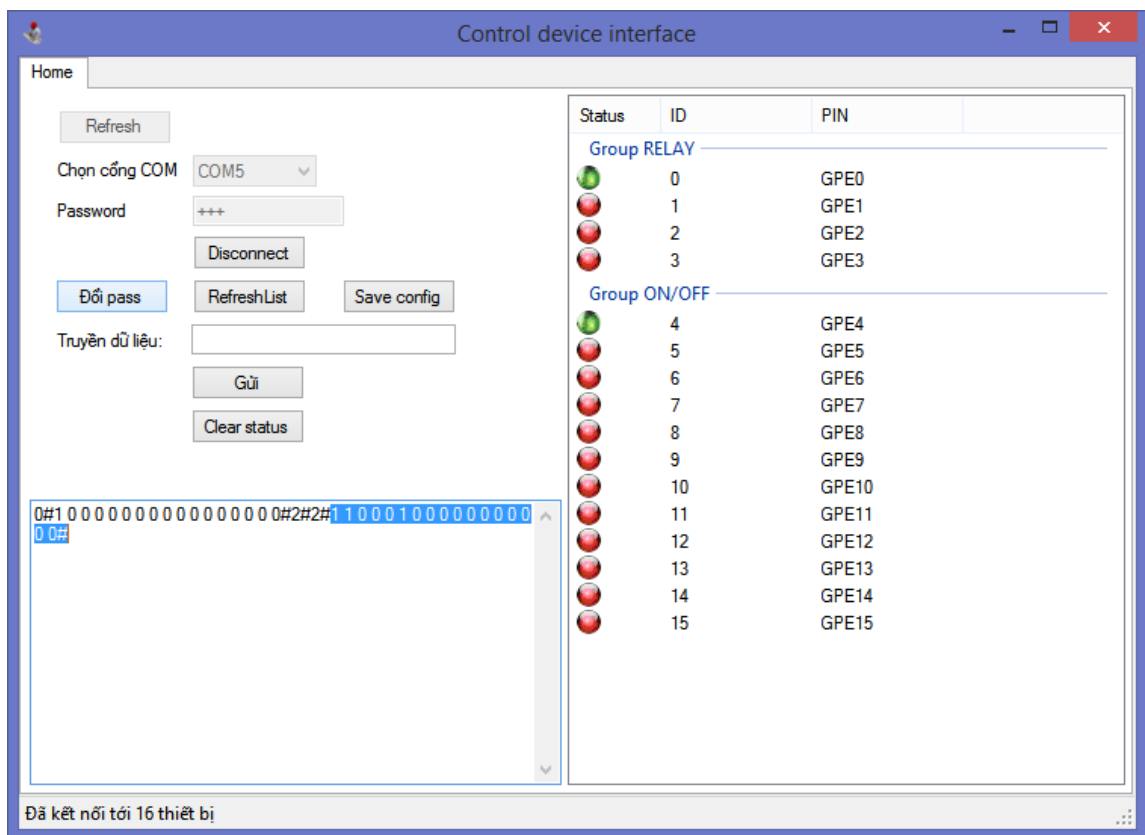
Hình 3.1: Giao diện tổng quát trên PC



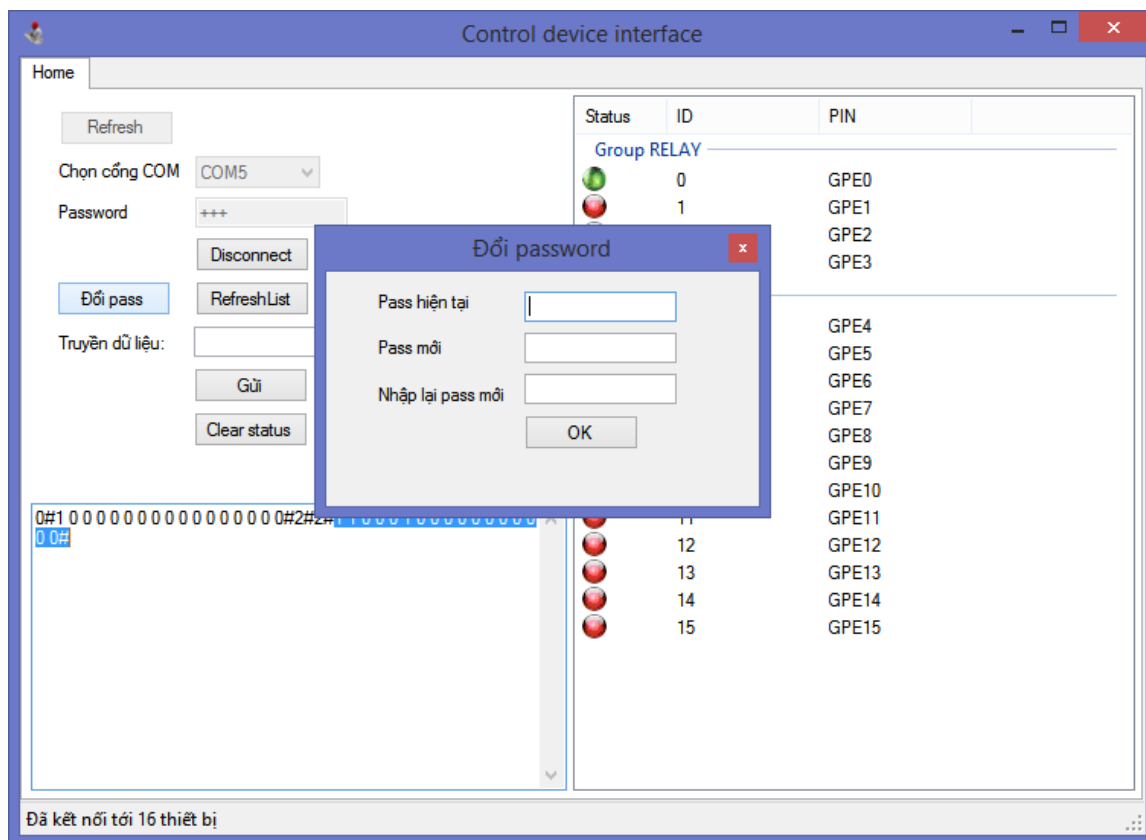
Hình 3.2: Giao diện kết nối thành công trên PC



Hình 3.3: Chức năng setup trạng thái thiết bị

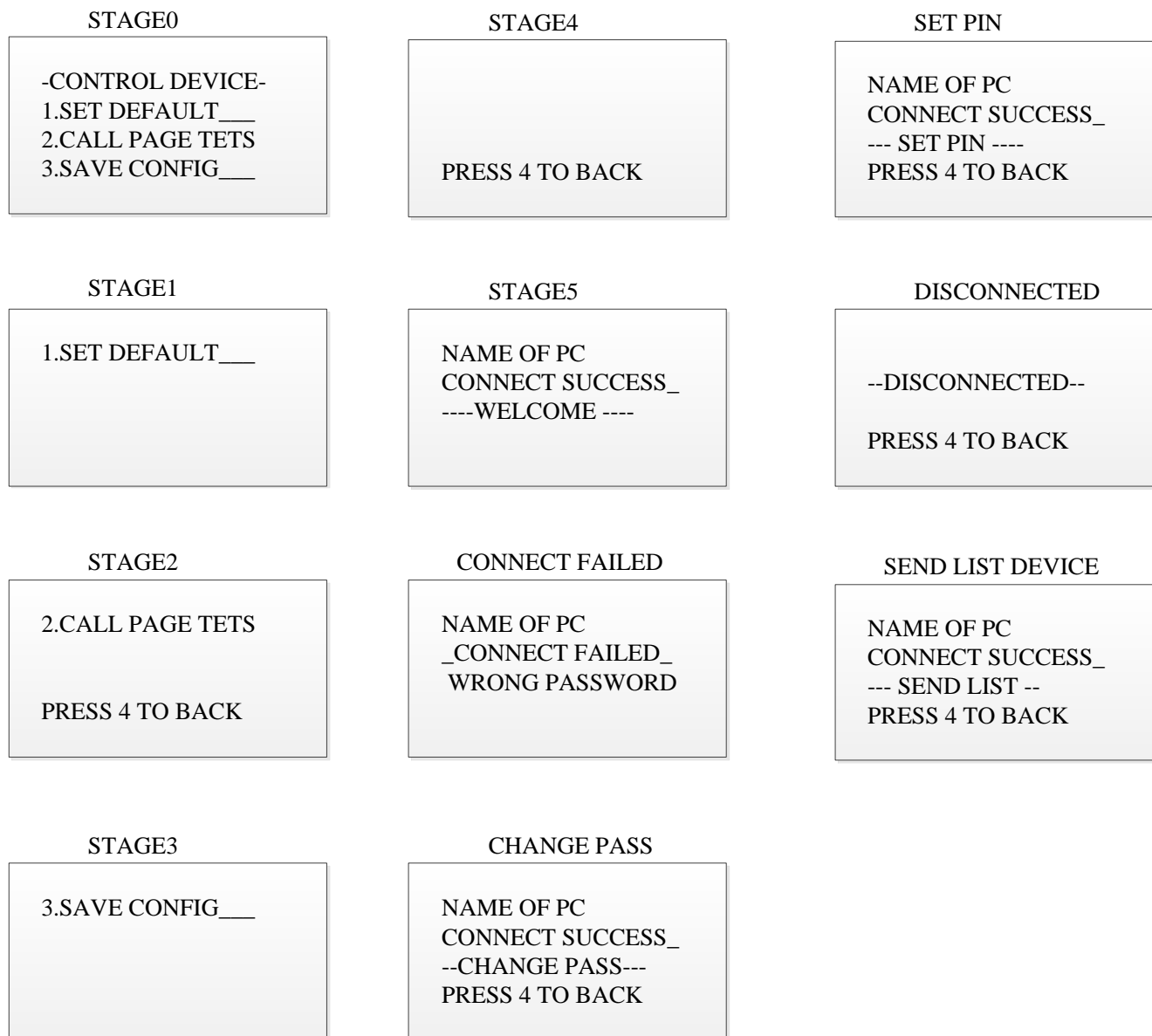


Hình 3.4: Chức năng làm mới lại danh sách thiết bị



Hình 3.5: Chức năng đổi mật khẩu

3.2 Giao diện LCD trên NUC140



Hình 3.6: Giao diện LCD theo trạng thái tương ứng

3.3 Bảng các biến hàm xây dựng trên NUC140

| Các biến sử dụng |
|-----------------------------------|
| #define TEST_LENGTH 256 |
| STR_PDMA_T sPDMA; |
| STR_UART_T sParam; |
| volatile uint32_t PDMA0_INT_Flag; |
| volatile uint32_t PDMA1_INT_Flag; |
| uint8_t SrcArray[TEST_LENGTH]; |

| |
|---------------------------------|
| uint8_t DestArray[TEST_LENGTH]; |
| uint8_t password[TEST_LENGTH]; |
| uint8_t* comStatus[16]; |
| uint8_t comRbuf[2000]; |
| uint8_t comTbuf[2000]; |
| uint8_t binChar[1]={0xFF}; |
| uint8_t connected=0; |
| uint8_t stage=0; |
| uint8_t mode=0; |
| uint16_t comRbytes=0; |
| uint16_t comTbytes=0; |
| uint16_t numDevice; |
| uint32_t SPIPort; |
| uint32_t u32ByteCount; |
| uint32_t u32FlashAddress; |
| uint32_t u32PageNumber; |
| uint32_t pin, status; |
| char* tenthietbi; |
| char* chpin; |
| char* chstatus; |
| char* chnumdevice; |
| char chnamepc[TEST_LENGTH]; |

Bảng 3.1: Bảng các biến sử dụng trong chương trình

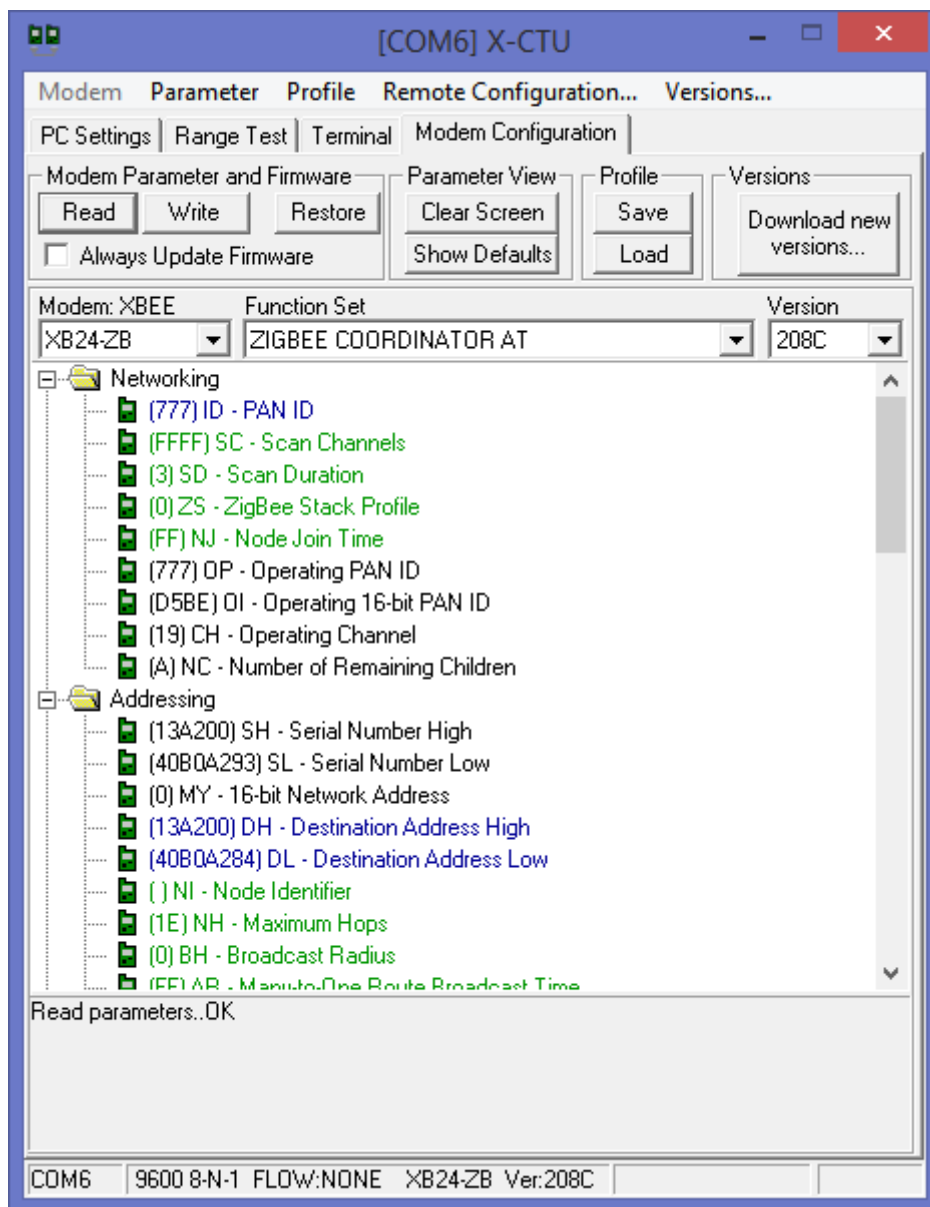
| Các hàm xây dựng | Chức năng |
|---|---------------------------------------|
| uint8_t scan_key(void); | Quét phím |
| uint8_t matchArray(char *sour, uint8_t sizeDes, uint8_t des[MAX_SIZE]); | Hàm so sánh sizeDes kí tự của 2 chuỗi |
| uint32_t SpiFlash_ReadStatusReg1(void); | Hàm đọc thanh ghi trạng thái SpiFlash |

| | |
|---|--|
| uint32_t SpiFlash_ReadStatusReg2(void); | Hàm đọc thanh ghi trạng thái SpiFlash |
| | |
| void PDMA0_Callback(void); | Hàm xử lý ngắt PDMA0 |
| void PDMA1_Callback(void); | Hàm xử lý ngắt PDMA1 |
| | |
| void SpiFlash_ReadMidDid(void); | Hàm đọc thông tin thiết bị Flash (For W25Q16BV, ID: 0xEF; Device ID: 0x14) |
| void SpiFlash_ChipErase(void); | Hàm xóa Flash |
| void SpiFlash_WaitReady(void); | Hàm chờ trong khi xử lý trên Flash |
| void SpiFlash_PageProgram(uint32_t, uint32_t); | Hàm viết dữ liệu vào page trên Flash |
| void SpiFlash_ReadData(uint32_t, uint32_t); | Hàm đọc dữ liệu từ Flash |
| | |
| void configSPI(void); | Hàm cấu hình SPI PDMA |
| void EraseSPI(void); | Hàm xóa Flash (gọi lại hàm SpiFlash_ChipErase(void);) |
| void CloseSPI_PDMA(void); | Hàm đóng kết nối SPI PDMA |
| void SysTimerDelay(uint32_t); | Hàm delay hệ thống |
| | |
| void ClrSrcArray(void); | Làm rỗng mảng SrcArray |
| void ClrDesArray(void); | Làm rỗng mảng DesArray |
| void SetSrcArray(char* s); | Sao chép chuỗi s vào mảng SrcArray |
| void SetDefaultConf(void); | Đặt cấu hình mặc định |
| void ReadPage(uint32_t page); | Đọc một page dữ liệu từ flash |
| void WritePage(uint32_t page); | Viết dữ liệu vào 1 page trên flash |
| void WriteDevice(uint32_t sttdevice, char* status); | Viết trạng thái thiết bị vào flash |
| void Initial_main(void); | Hàm khởi tạo ban đầu trong hàm main |
| void Initial_annel(void); | Hàm khởi tạo giao tiếp LCD |
| void clr_all_annel(void); | Hàm reset LCD |
| void print_lcd(unsigned char, char *); | Hàm xuất dữ liệu ra LCD theo từng dòng |
| void setInterfaceLCD(char* line0, char* line1, char* line2, char* line3); | Hàm xuất dữ liệu ra LCD theo 4 dòng |
| void delayT(void); | Hàm delay |
| void delay(void); | Hàm delay |
| void InitGPIO(void); | Hàm khởi tạo các cổng GPIO |
| void setArray(char* ch); | Hàm sao chép chuỗi ch vào mảng comTbuf |
| void sendString(char* s); | Hàm gửi dữ liệu qua cổng UART0 |
| void sendListDevice(void); | Hàm gửi list trạng thái thiết bị qua cổng UART0 |
| void getPassword(void); | Hàm lấy password từ flash |
| void getListDevice(void); | Hàm lấy list trạng thái thiết bị từ flash |
| void getNumberDevice(void); | Hàm lấy số lượng thiết bị lưu trên flash |
| void saveconfig(void); | Hàm lưu cấu hình xuống flash |

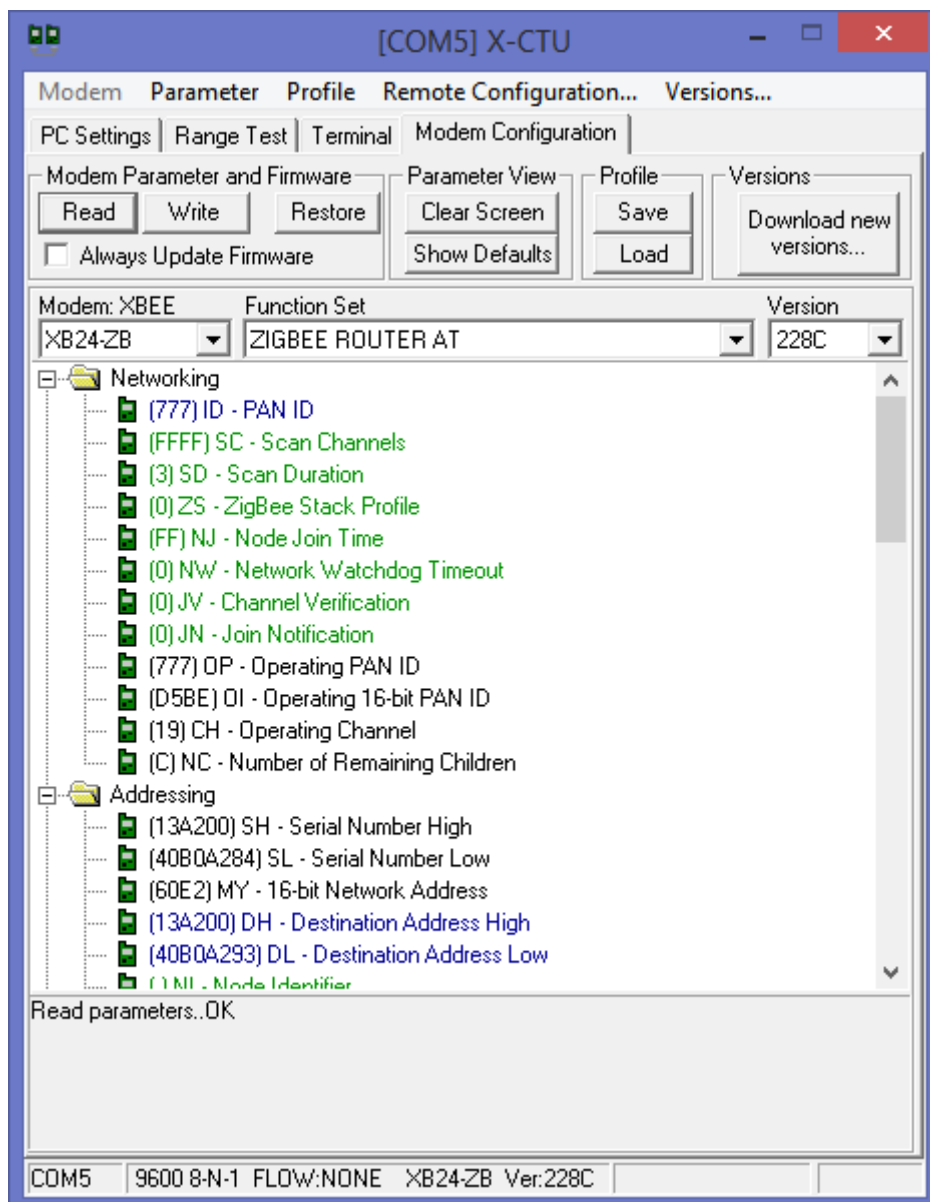
| | |
|-----------------------|--|
| void xulymode1(void); | Hàm xử lý ngắt nhận dữ liệu cho UART0 mode 1 |
| void xulymode2(void); | Hàm xử lý ngắt nhận dữ liệu cho UART0 mode 2 |

Bảng 3.2: Bảng các hàm sử dụng trong chương trình

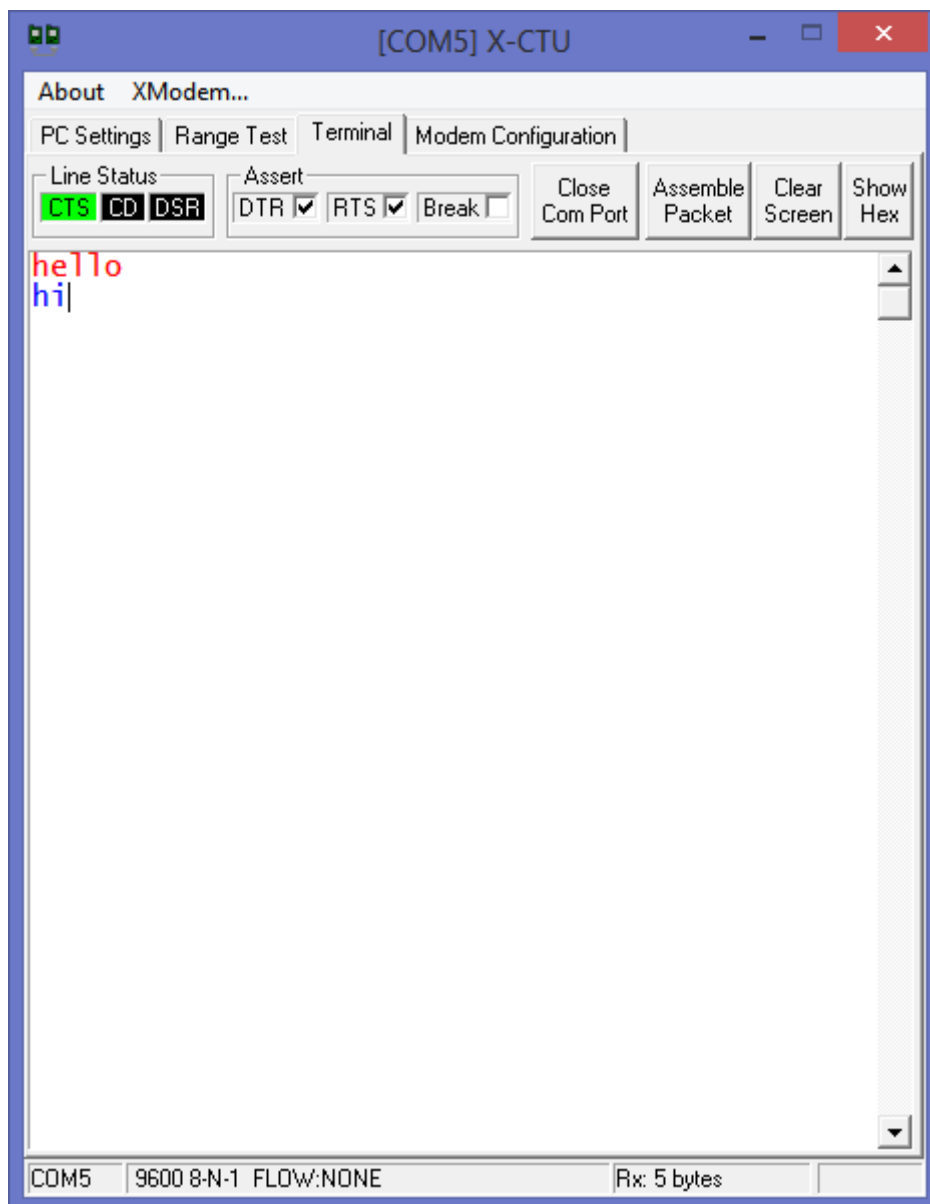
3.4 Cấu hình XBEE model XB24-Z7WIT-004



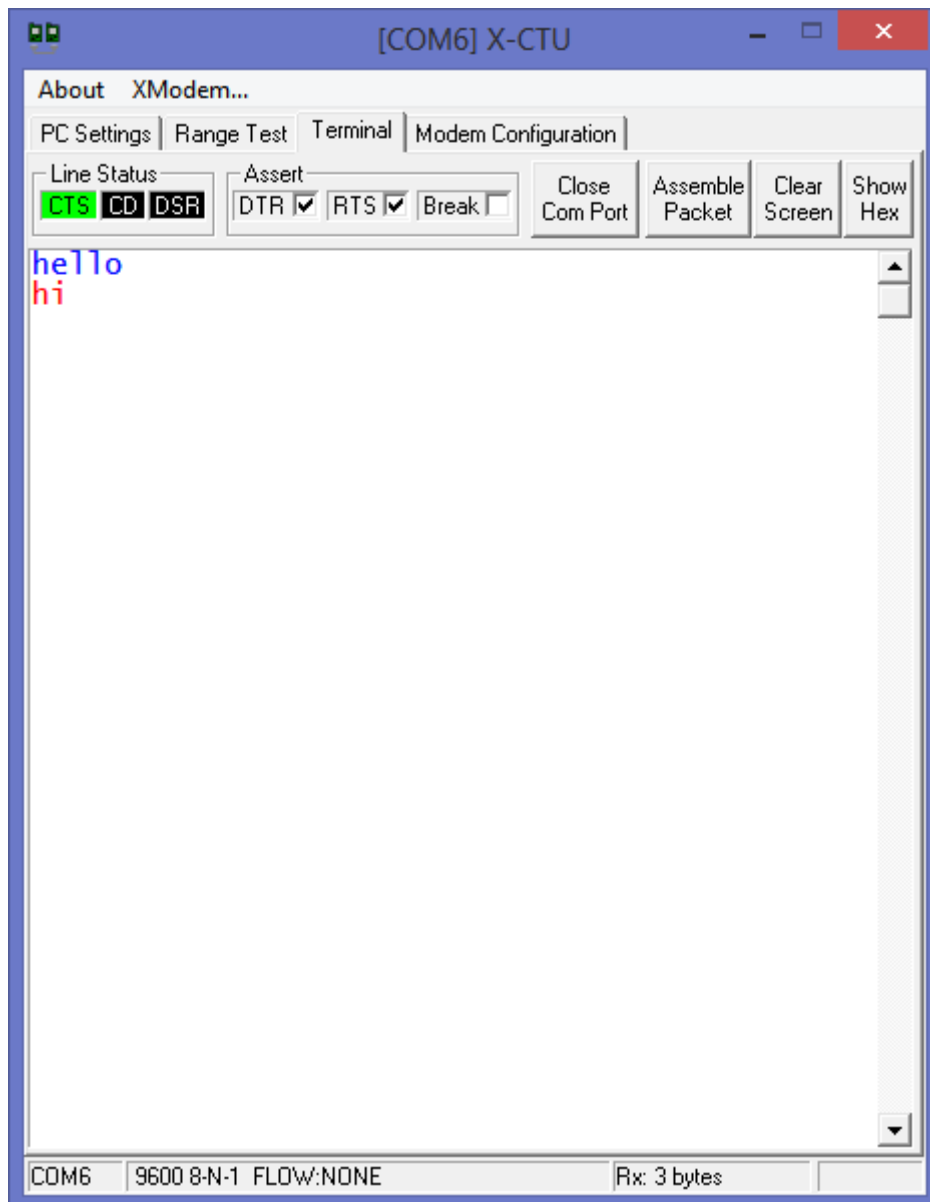
Hình 3.7: Cấu hình cho COORDINATOR



Hình 3.8: Cấu hình cho ROUTER

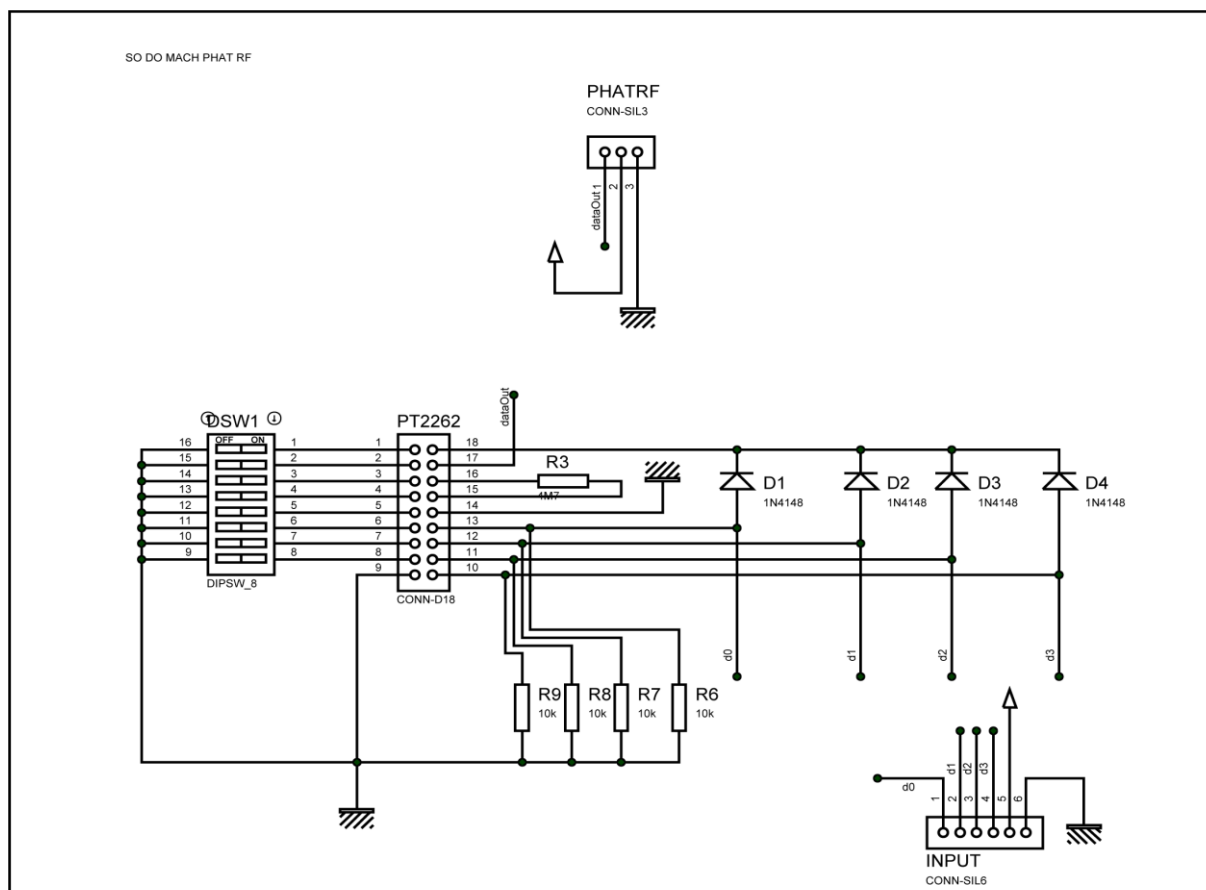


Hình 3.9: Gửi và nhận trên Router

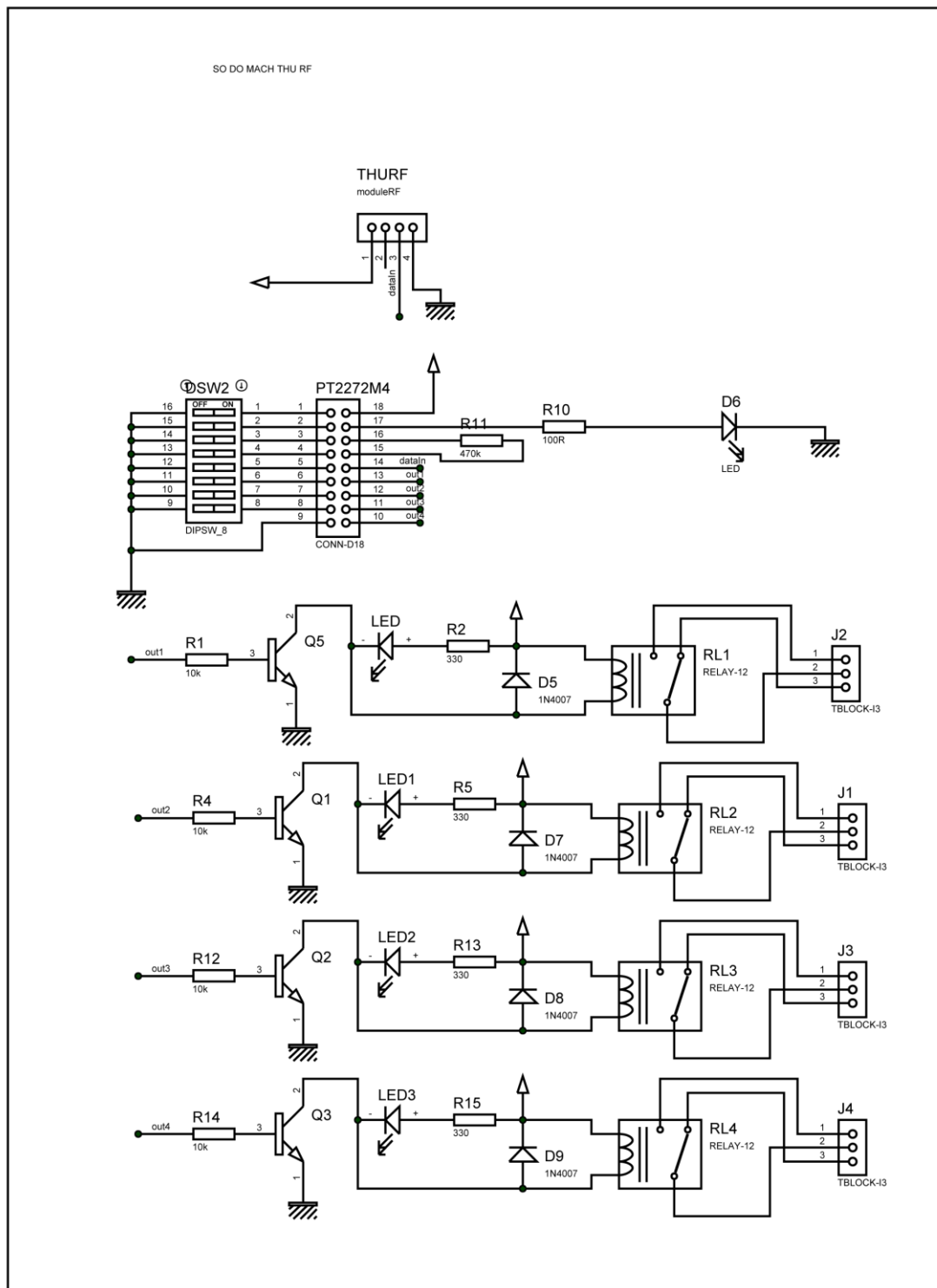


Hình 3.10: Gửi và nhận trên Coordinator

3.5 Thiết kế schematic mạch xử lý thu phát RF



Hình 3.11: Schematic mạch phát RF



Hình 3.12: Schematic mạch thu RF

4. Kết luận và hướng phát triển

4.1 Kết quả đạt được

- Hiểu thêm về công nghệ mạng không dây zigbee
- Ứng dụng được điều khiển từ xa qua sóng RF
- Hiện thực được hệ thống giao tiếp PC với NUC140 bằng các giao thức tự định nghĩa.
- Hệ thống kết nối và hoạt động ổn định.

4.2 Hướng phát triển

- Mở rộng hệ thống và thiết lập server lên vi xử lý ARM để có thể điều khiển thông qua mạng internet và trên nhiều nền tảng cũng như những thiết bị di động.
- Cài đặt xử lý giọng nói độc lập ở các thiết bị để xử lý khi cần thiết mà không cần phải truy cập vào hệ thống phức tạp.

TÀI LIỆU THAM KHẢO

1. Nuvoton, 2013, Datasheet NUC140
2. Nuvoton, 2013, Tutorial NUC 140
3. Nuvoton, 2013, Sample code NUC140
4. Các nguồn khác:
 - ^[1] <http://stackoverflow.com/>
 - ^[2] <http://www.cplusplus.com/>
 - ^[3] <http://www.ZigBee.org>