**BÁO CÁO TIẾN ĐỘ**

**PROJECT: LOREN**

**Thành viên:**

Trần Hoàng Lộc

MSSV: 13520462

Nguyễn Phú Cường

MSSV: 13520106

**Giáo viên hướng dẫn:** Trịnh Lê Huy

**Mục Lục**

[**I.** **Tìm hiểu về công nghệ LoRa** 6](#_Toc485597687)

[**1.** **Lợi ích khi dùng LoRa:** 8](#_Toc485597688)

[**2.** **Phân loại các lớp End-device:** 8](#_Toc485597689)

[**II.** **Tìm hiểu Module iM880A** 11](#_Toc485597690)

[**1.Đặc tính:** 12](#_Toc485597691)

[**2. LoRa Mote** 14](#_Toc485597692)

[**3. Module iM880A Modified** 18](#_Toc485597693)

[**III.** **Giải thuật định vị** 19](#_Toc485597694)

[**1.** **Tìm hiểu các hệ thống định vị** 19](#_Toc485597695)

[**IV.** **Lập trình truyền nhận dữ liệu giữa nhiều board Node iM880A** 29](#_Toc485597696)

[**V.** **Giao tiếp module Bluetooth và ứng dụng Android** 35](#_Toc485597697)

[**1.** **Giao tiếp với module Bluetooth HC-05** 35](#_Toc485597698)

[**2.** **Ứng dụng Android LoRen Bluetooth Chat** 36](#_Toc485597699)

[**a.** **Bluetooth chat** 36](#_Toc485597700)

[**b.** **Hiển thị khoảng cách theo thời gian thực** 39](#_Toc485597701)

[**3.** **Mô phỏng định vị trên phần mềm Matlab** 40](#_Toc485597702)

[**VI.** **Giao tiếp với SD Card và lưu trữ dữ liệu** 43](#_Toc485597703)

[**1.** **FATFS – Generic FAT Filesystem Module** 43](#_Toc485597704)

[**a.** **FatFs là gì** 43](#_Toc485597705)

[**b.** **Các đặc tính của FatFS** 44](#_Toc485597706)

[**c.** **Application interface** 44](#_Toc485597707)

[**d.** **Media Access Interface** 45](#_Toc485597708)

[**2.** **Giao tiếp và lưu trữ dữ liệu** 46](#_Toc485597709)

[**VII.** **Kết quả quá trình test thử và định vị** 50](#_Toc485597710)

[**1. Dữ liệu RSSI khoảng cách 1 m:** 50](#_Toc485597711)

[**2.** **Test khoảng cách 12 tầng** 51](#_Toc485597712)

[**3.** **Test định vị thực tế:** 52](#_Toc485597713)

[**a.** **Sơ đồ hệ thống** 52](#_Toc485597714)

[**b.** **Tính toán hệ số môi trường và giá trị RSSI trên 1m** 53](#_Toc485597715)

[**c.** **Tính giá trị khoảng cách từ các điểm nhận** 56](#_Toc485597716)

[**VIII.** **Tài liệu tham khảo** 57](#_Toc485597717)

**Mục lục hình ảnh**

[**Figure 1. Đặc tính của công nghệ LoRa với các công nghệ truyền thông không dây khác 6**](#_Toc485597853)

[**Figure 2. Phân loại các lớp End-device 7**](#_Toc485597854)

[**Figure 3. Đặc điểm của class A 8**](#_Toc485597855)

[**Figure 4. Đặc điểm của class B 9**](#_Toc485597856)

[**Figure 5. Module iM880A 11**](#_Toc485597857)

[**Figure 6. Sơ đồ khối của Module iM880A 12**](#_Toc485597858)

[**Figure 7. Bảng thể hiện hiệu suất của module iM880A 12**](#_Toc485597859)

[**Figure 8. giá trị Spreading Factor của module Sx1272 13**](#_Toc485597860)

[**Figure 9. Coding Rate của module iM880A 13**](#_Toc485597861)

[**Figure 10. Schematic của board LoRa Mote Semtech 14**](#_Toc485597862)

[**Figure 11. Schematic của các module thành phần trên board LoRa Mote 14**](#_Toc485597863)

[**Figure 12. Schematic của các module thành phần trên board LoRa Mote 15**](#_Toc485597864)

[**Figure 13. Schematic của các module thành phần trên board LoRa Mote 15**](#_Toc485597865)

[**Figure 14. Schematic của các module thành phần trên board LoRa Mote 16**](#_Toc485597866)

[**Figure 15. Hình ảnh thực tế của board LoRa Mote 16**](#_Toc485597867)

[**Figure 16. Schematic của board iM880A Modified 17**](#_Toc485597868)

[**Figure 17. Layout của board iM880A Modified 17**](#_Toc485597869)

[**Figure 18. HÌnh ảnh mạch thực tế của board iM880A Modified 18**](#_Toc485597870)

[**Figure 19. Phân loại các hệ thống định vị trong mạng WSNs 19**](#_Toc485597871)

[**Figure 20. Mô hình kiến trúc của phương pháp định vị RSSI 20**](#_Toc485597872)

[**Figure 21. Làm mịn giá trị RSS thu được 21**](#_Toc485597873)

[**Figure 22. Một ví dụ về mô hình ROCRSSI 23**](#_Toc485597874)

[**Figure 23. Giải thuật định vị ROCRSSI 23**](#_Toc485597875)

[**Figure 24. Bảng so sánh độ chính xác của các phương pháp định vị 24**](#_Toc485597876)

[**Figure 25. Độ trễ của tín hiệu nhận được trong giải thuật SLA 24**](file:///E:\HOC%20KI%20VIII\KHOA%20LUAN%20TOT%20NGHIEP\BÁO-CÁO-TIẾN-ĐỘ-14-5-2017.docx#_Toc485597877)

[**Figure 26. Giải thuật SLA 25**](#_Toc485597878)

[**Figure 27. Độ trễ của các tín hiệu nhận được bù đắp 25**](#_Toc485597879)

[**Figure 28. Giải thuật DLA 26**](#_Toc485597880)

[**Figure 29. So sánh độ chính xác của 2 giải thuật SLA và DLA 26**](#_Toc485597881)

[**Figure 30. Định tuyến sử dụng giải thuật APIT 27**](#_Toc485597882)

[**Figure 31. So sánh giữa 2 phương pháp định vị Indoor và Outdoor 28**](#_Toc485597883)

[**Figure 32. Giao diện bluetooth chat tích hợp với thiết đặt thông số công thức tính RSSI 36**](#_Toc485597884)

[**Figure 33. Giao diện danh sách thiết bị bluetooth kết nối 36**](#_Toc485597885)

[**Figure 34. Hiển thị thông tin gói tin nhận được 37**](#_Toc485597886)

[**Figure 35. Bấm nút SOS giao diện sẽ chuyển sang hiển thị Google map và khoảng cách dựa vào chỉ số RSSI 38**](#_Toc485597887)

[**Figure 36. Sơ đồ hoạt động của hệ thống định vị 39**](file:///E:\HOC%20KI%20VIII\KHOA%20LUAN%20TOT%20NGHIEP\BÁO-CÁO-TIẾN-ĐỘ-14-5-2017.docx#_Toc485597888)

[**Figure 37. Sơ đồ quá trình thu nhận RSSI của 3 Anchor Node 40**](file:///E:\HOC%20KI%20VIII\KHOA%20LUAN%20TOT%20NGHIEP\BÁO-CÁO-TIẾN-ĐỘ-14-5-2017.docx#_Toc485597889)

[**Figure 38. Giao diện công cụ nhận dữ liệu RSSI thông qua giao tiếp UART 41**](#_Toc485597890)

[**Figure 39. Các gói tin nhận được 41**](#_Toc485597891)

[**Figure 40. Mô phỏng định vị trên phần mềm Matlab 42**](#_Toc485597892)

[**Figure 41. Module FatFS 43**](#_Toc485597893)

[**Figure 42. Các FatFs API là cầu nối giữa tầng ứng dụng và FatFs Module 44**](#_Toc485597894)

[**Figure 43. Module FatFs giao tiếp với các thiết bị lưu trữ thông qua Device I/F 45**](#_Toc485597895)

[**Figure 44. Thẻ nhớ Micro SD 45**](#_Toc485597896)

[**Figure 45. Sơ đồ nối chân của thẻ SD và Micro SD 46**](#_Toc485597897)

[**Figure 46. Tên các chân Pin của SD Card và MicroSD 46**](#_Toc485597898)

[**Figure 47. Sơ đồ chân ra của module iM880A 47**](#_Toc485597899)

[**Figure 48. Bảng ra chân PIN của iM880A 47**](#_Toc485597900)

[**Figure 49. Sơ đồ liên kết các file trong project 48**](#_Toc485597901)

[**Figure 50. Dữ liệu gửi được lưu vào file "write.txt" 48**](#_Toc485597902)

[**Figure 51. Chỉ số RSSI 49**](#_Toc485597903)

[**Figure 52. Biểu diễn mật độ của các giá trị RSSI 49**](#_Toc485597904)

[**Figure 53. Giá trị RSSI 50**](#_Toc485597905)

[**Figure 54. Biểu diễn mật độ 50**](#_Toc485597906)

[**Figure 55. Giá trị RSSI 51**](#_Toc485597907)

[**Figure 56. Biểu diễn mật độ 51**](#_Toc485597908)

[**Figure 57. Sơ đồ tọa độ demo 52**](#_Toc485597909)

[**Figure 58. Biểu đồ thể hiện phân phối chuẩn giá trị RSSI trên khoảng cách 1m 53**](#_Toc485597910)

[**Figure 59. Biểu đồ thể hiện phân phối chuẩn giá trị RSSI trên khoảng cách 3m 54**](#_Toc485597911)

[**Figure 60. Dữ liệu sau khi được lọc Gauss, những biểu đồ bên trái là dữ liệu thô, bên phải là dữ liệu đã qua bộ lọc 55**](#_Toc485597912)

[**Figure 61. Áp dụng phân phối chuẩn đối với dữ liệu vừa được lọc 55**](#_Toc485597913)

[**Figure 62. Biểu diễn vị trí các node nhận và vị trí node gửi tính toán được so với thực tế 56**](#_Toc485597914)

1. **Tìm hiểu về công nghệ LoRa**

**Các khái niệm cơ bản:**

**Spreading Factor (SF)**

Chỉ số SF càng lớn, lượng thông tin truyền đi càng nhiều. Vì vậy, để tang độ lợi của quá trình truyền thì phải tăng độ nhạy nhận được

Các giá trị SF có thể lập trình được: 7,8,9,10,11,12

**Bandwidth (BW)**

Với mỗi chỉ số SF có được, Banwidth càng hẹp thì càng tăng độ nhạy nhận được, tuy nhiên sẽ tăng thời gian truyền trong không khí.

Các giá trị Banwidth lập trình được: 125 kHz, 250 kHz, 500 kHz

**Forward Error Correction Code Rate (CR):**

Cung cấp những khả năng dự phòng để xác định lỗi và sửa chúng

**SNR** (viết tắt của **Signal-to-Noise Ratio**)

Là tỉ số giữa năng lượng của tín hiệu trên nhiễu tại một thời điểm nhất định. SNR thường được biểu diễn dưới dạng [dB](https://tudien.vntelecom.org/w/index.php?title=DB&action=edit&redlink=1) ([Decibel](https://tudien.vntelecom.org/w/index.php?title=DB&action=edit&redlink=1)).

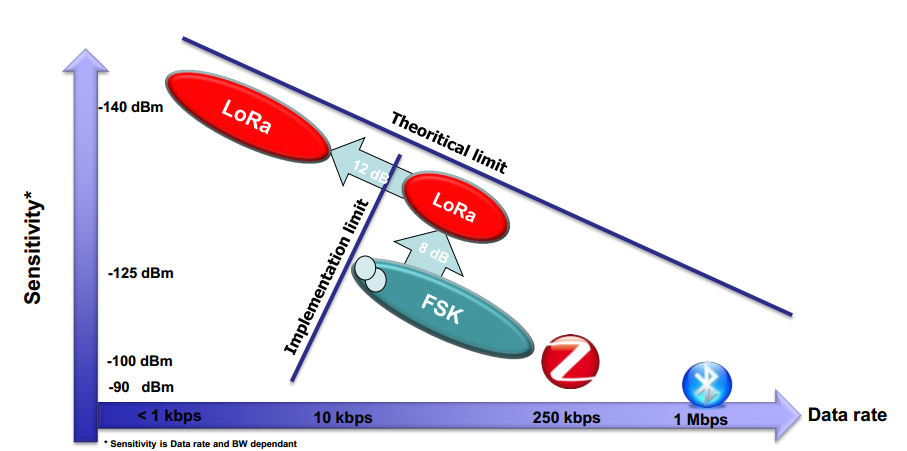


Figure 1. Đặc tính của công nghệ LoRa với các công nghệ truyền thông không dây khác

**ASK (Amplitude Shift Keying)**

Là điều chế số theo [biên độ tín hiệu](https://vi.wikipedia.org/w/index.php?title=Bi%C3%AAn_%C4%91%E1%BB%99_t%C3%ADn_hi%E1%BB%87u&action=edit&redlink=1). Tín hiệu ASK có dạng sóng dao động có tần số *f*, mỗi bit đặc trưng bởi biên độ khác nhau của tín hiệu.

Ví dụ: tín hiệu ASK có tần số 100 KHz, biên độ tín hiệu = 1 cho [bit](https://vi.wikipedia.org/wiki/Bit) 0 và biên độ tín hiệu = -1 cho bit 1.

**FSK (Frequency Shift Keying)**

Là điều chế số theo tần số tín hiệu. Tín hiệu FSK có dạng sóng dao động có tần số khác nhau, mỗi bit đặc trưng bởi tần số khác nhau này của tín hiệu. ưu điểm của điều chế này là dễ chế tạo nhưng lại hay mắc lỗi khi truyến

Ví dụ: f = 100Khz cho bit 0 và f' = 200Khz cho bit 1

**PSK (Phase Shift Keying)**

Là điều chế số theo pha tín hiệu. Tín hiệu PSK có dạng sóng dao động có tần số *f*, mỗi bit đặc trưng bởi [góc pha](https://vi.wikipedia.org/w/index.php?title=G%C3%B3c_pha&action=edit&redlink=1) khác nhau của tín hiệu.

Ví dụ: pha= 90° cho bit 0 và pha = -90° cho bit 1.

**QPSK (Quadrature Phase Shift Keying)**

Là điều chế pha cầu phương (điều chế pha vuông góc). QPSK là 1 kỹ thuật điều chế tín hiệu số, mã hóa 2 bit thành 1 *symbol*.

1. **Lợi ích khi dùng LoRa:**

Tối ưu hóa kĩ thuật Điều chế trải phổ

Mang lợi thế của điều chế trải phổ định hướng chuỗi

Không phải nhận hệ quả

Tiếp cận 1 cách bình thường (đặt 1 giới hạn lớn với sự ổn định LO)

Tiếp cận phương pháp trải phổ

1. **Phân loại các lớp End-device:**

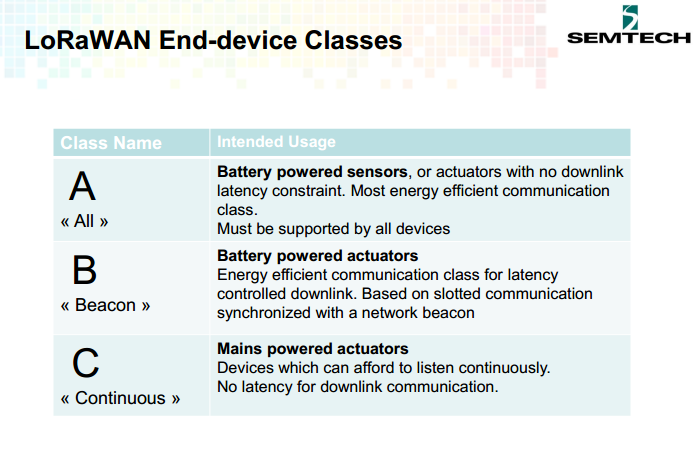


Figure 2. Phân loại các lớp End-device

**Class A: Battery Powered end-device**

* + Giao tiếp song song
  + Message đơn hướng
  + End-device khởi đầu giao tiếp (uplink)
  + Server giao tiếp với end-device (downlink) suốt quá trình trả lời
  + Ưu điểm: Lowest power consumption = longest battery life
  + Nhược điểm: Đường truyền trễ do uplink

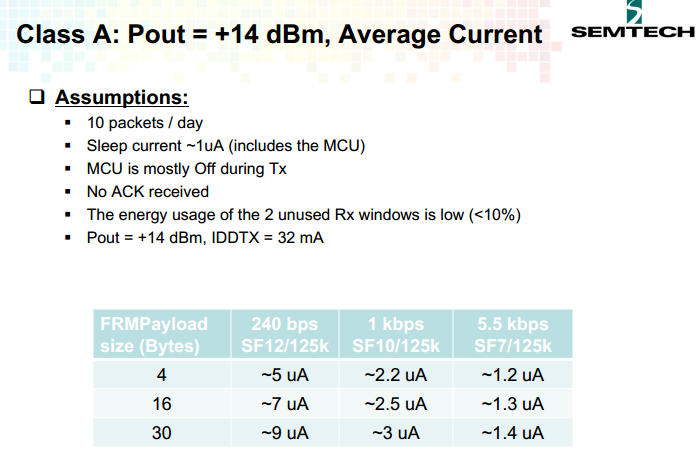


Figure 3. Đặc điểm của class A

**Class B: Low lownllink latency end-device**

* + Truyền song công với Slots lịch trình nhận
  + Message đơn hướng và đa hướng
  + Mốc định kì từ Gateway
  + Thêm cửa sổ nhận
  + Server có thể khởi tạo truyền với chu kì cố định
  + Ưu điểm: Xác định đc độ trễ tín hiệu Downlink
  + Nhược điểm: Tiêu thụ điện năng cao hơn

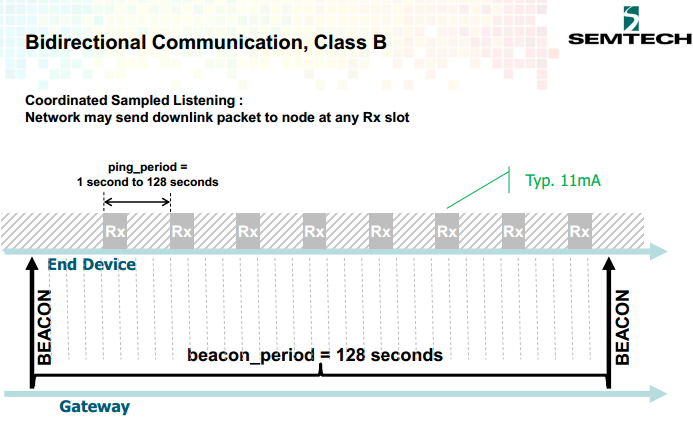


Figure 4. Đặc điểm của class B

**Class C: Không có độ trễ với End-device**

* Giao tiếp song song
* Unicast và Multi cast message
* Server có thể khởi tạo truyền bất cứ lúc nào
* End-device nhận liên tục
* Ưu điểm: Độ trễ thấp nhất
* Khuyết điểm: Tốn năng lượng nhất

**Modulation setting for Europe:**

**Giao tiếp khoảng cách xa nhất:**

Data Rate: 0

Spreading Factor: SF12

Bandwidth: 125kHz

Coding Rate: 4/5

Low Datarate Optimize: Enable

**Dữ liệu truyền nhiều nhất:**

Data Rate: 0

Spreading Factor: SF7

Bandwidth: 250kHz

Coding Rate: 4/5

Low Datarate Optimize: Disable

1. **Tìm hiểu Module iM880A**

Module iM880A là 1 module nhỏ gọn, tiêu thụ ít năng lượng, truyền dữ liệu song song với tần số 868 MHz sử dụng công nghệ Semtech’s LoRa. Module cung cấp giao tiếp Ultra-long range spread spectrum và miễn nhiễu cao với dòng tiêu thụ thấp.

iM880A là 1 module tiết kiệm với những ứng dụng song RF có chi phí đắt và lâu dài.

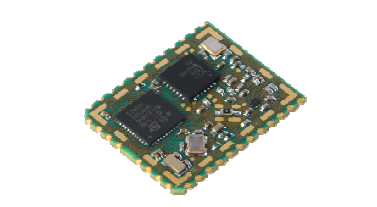


Figure 5. Module iM880A

## **1.Đặc tính:**

* Compact module 20.0 x 25 x 2 mm
* LoRa modulation technology
* Sensitivity down to -137 dBm
* UART, SPI and I2C interface
* Digital inputs and outputs
* Analog inputs
* Supply voltage range from 2.4 to 3.6V
* RF interface optimized to 50 ohm
* Output power level up to +19dBm
* High link budget up to 156 dB
* Range up to 15000m (Line of Sight)
* STM32L151Cx
* Pre-Certified according to EN 300 220

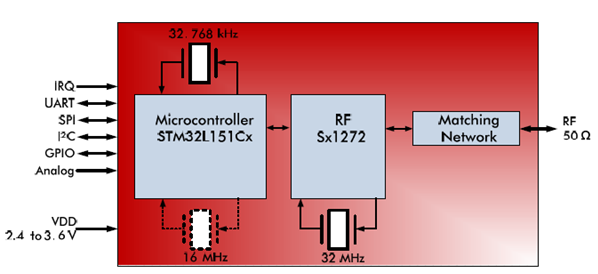


Figure 6. Sơ đồ khối của Module iM880A

Module iM880A sử dụng 1 kĩ thuật điều chế song LoRa đã được cấp bằng sang chế bởi Semtech với sự kết hợp giữa điều chế phổ (spread spectrum modulation) và kĩ thuật sửa lỗi trước nhằm tăng khoảng cách truyền và độ bền vững của các liên kết truyền thông vô tuyến so với các phương pháp điều chế truyền thống như FSK hay OOK.

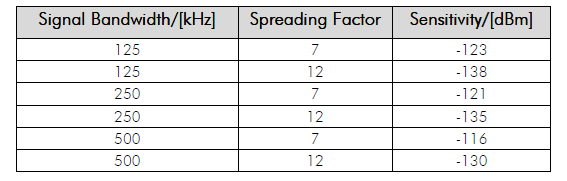


Figure 7. Bảng thể hiện hiệu suất của module iM880A

iM880A sử dụng công nghệ điều chế phổ độc quyền LoRa của Semtech. Với kĩ thuật điều chế này, ngược lại với các kĩ thuật điều chế thông thường, cho phép tăng mức năng lượng sẵn có (link budget) và tăng độ miễn nhiễu trong dãi băng tần. (Đạt độ nhạy cao hơn 8 dB so với phương pháp điều chế bằng FSK)

LoRa còn mang lại những lợi ích đáng kể trong việc ngăn chặn và chọn lọc, giải quyết vấn đề giữa việc làm hài hòa giữa khoảng cách truyền , miễn nhiễu và năng lượng tiêu thụ.

Trong Module LoRa, ta được cung cấp 3 lựa chọn băng thông 125kHz, 250 kHz, 500 kHz, với giá trị Spreading factors ranging từ 7 tới 12.

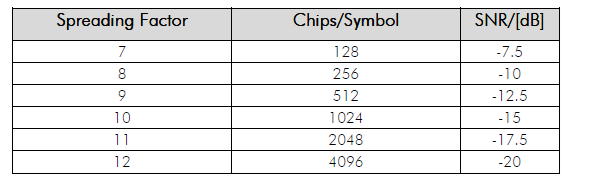


Figure 8. giá trị Spreading Factor của module Sx1272

Để tăng độ bền của liên kết, module iM880A cung cấp mã hóa lỗi tuần tự (cyclic error coding) với các tỉ lệ Coding rate khác nhau. Với việc sử dụng tỉ lệ này, ta có thể phát hiện và hiệu chỉnh lỗi.

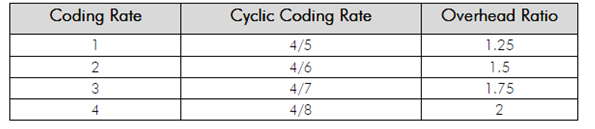


Figure 9. Coding Rate của module iM880A

## **2. LoRa Mote**

Board được sản xuất bởi Semtech dùng để nghiên cứu và lập trình các ứng dụng cơ bản cho công nghệ LoRa với các Module được tích hợp sẵn trên Board

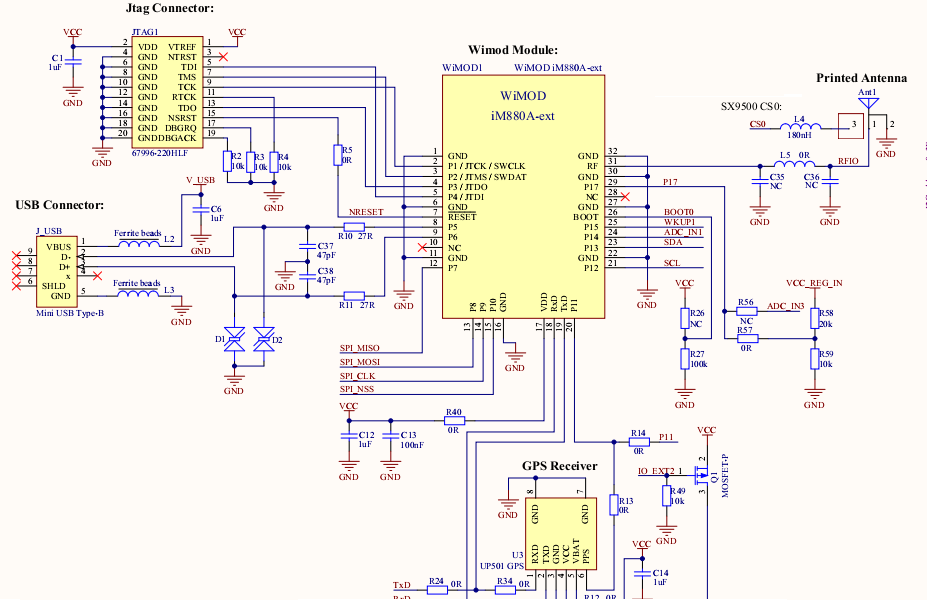
****

Figure 10. Schematic của board LoRa Mote Semtech

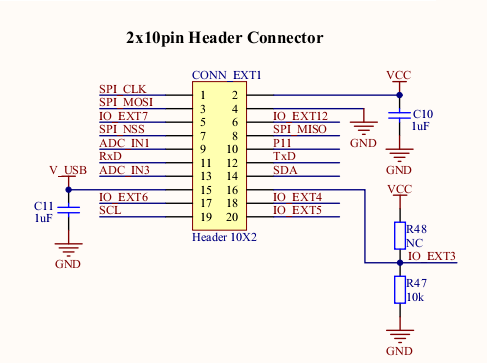
****

Figure 11. Schematic của các module thành phần trên board LoRa Mote

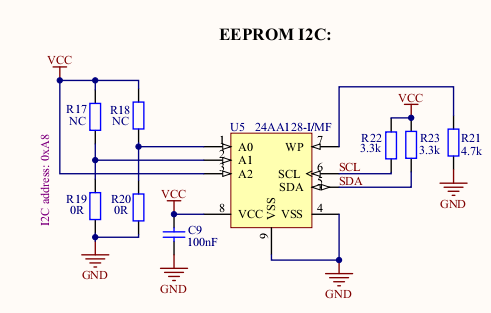


Figure 12. Schematic của các module thành phần trên board LoRa Mote

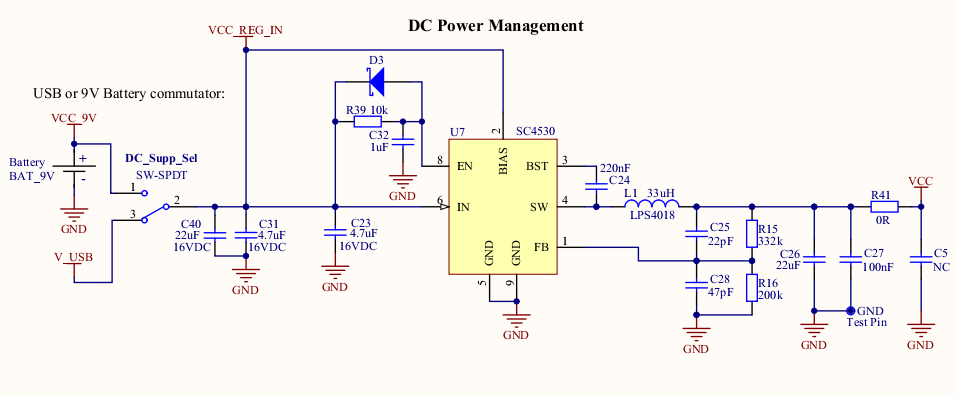
****

Figure 13. Schematic của các module thành phần trên board LoRa Mote

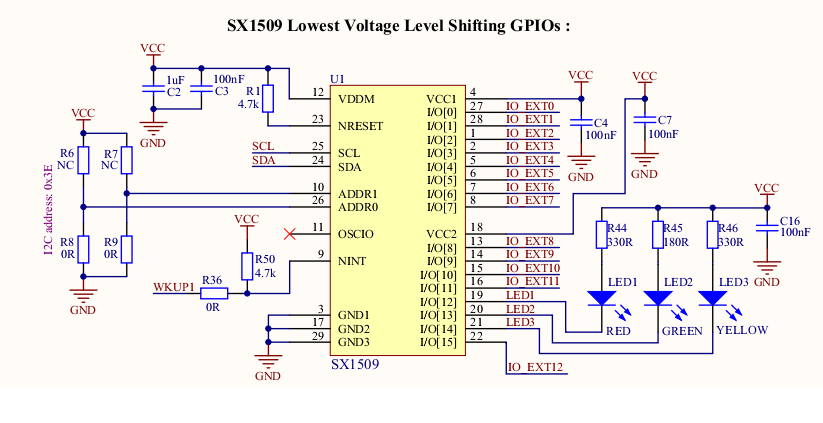
****

Figure 14. Schematic của các module thành phần trên board LoRa Mote

**Mạch thực tế:**



Figure 15. Hình ảnh thực tế của board LoRa Mote

## **3. Module iM880A Modified**

Module iM880A Modified được chỉnh lại, phù hợp hơn với nhu cầu sử dụng

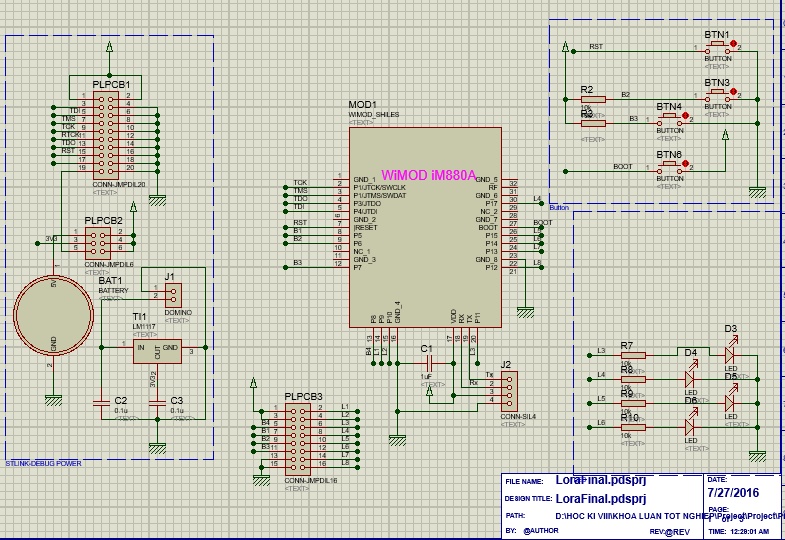


Figure 16. Schematic của board iM880A Modified

**Layout PCB**

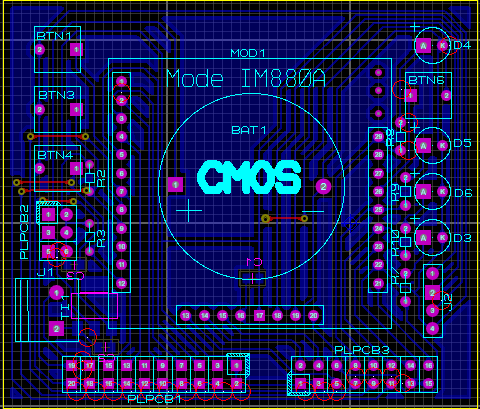


Figure 17. Layout của board iM880A Modified

**Mạch thực tế:**



Figure 18. HÌnh ảnh mạch thực tế của board iM880A Modified

1. **Giải thuật định vị**
2. **Tìm hiểu các hệ thống định vị**

**Các phương pháp định vị trong môi trường WSNs**

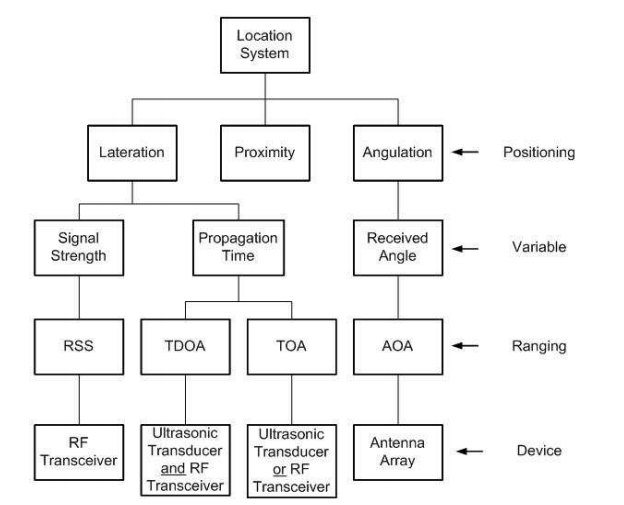


Figure 19. Phân loại các hệ thống định vị trong mạng WSNs

Có nhiều thuật toán định vị sử dụng Wireless Sensor Networks (WSNs), mỗi thuật toán có những yêu cầu về tài nguyên phần cứng, phi phí và độ phức tạp khác nhau. Tuy nhiên được chia thành 2 thể loại chính:

Ranged-based methods: Dựa trên sự giải định khoảng cách tuyệt đối giữa 1 Sender và 1 Receiver bởi Received Signal Strength (RSS), hoặc Time-Of-Flight (TOF) giữa 2 thiết bị. Độ chính xác của việc ước tính này phụ thuộc vào môi trường truyền dẫn, môi trường xung quanh và thường có phần cứng phức tạp

**1.1 RSSI:**

Đây là kĩ thuật định vị dựa trên việc tính toán khoảng cách giữa 1 Blind node cần xác định với các Reference Nodes xung quanh dựa vào giá trị RSS thu được của 2 node. Giá trị này sau đó thông qua giải thuật Trilateration để định vị vị trí của thiết bị.

Quá trình này chia làm các giai đoạn:

**Deterministic Phase:** Hiệu chỉnh giá trị RSSI cho mỗi Reference Node. Vì các yếu tố môi trường khác nhau tại mỗi node ảnh hưởng tới giá trị đo RSSI. Ta cần thống kê các giá trị RSSI thô từ Blind Node tới Reference Node, giá trị này được xử lí để tìm ra hằng số truyền phù hợp với mỗi Node.

Công thức tính giá trị RSSI:

Trong đó:

n là thông số môi trường

d là khoảng cách giữa 2 Node gửi và nhận

A là giá trị RSS giữa 2 thiết bị ở khoảng cách 1 mét

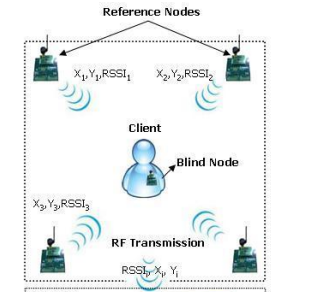


Figure 20. Mô hình kiến trúc của phương pháp định vị RSSI

**Probabilistic Phase:** Từ giá trị n và A tìm được ở giai đoạn trên, tiến hành đo đạc thực tế để tìm ra vị trí của thiết bị qua 2 công đoạn nhỏ: Ước lượng khoảng cách (Distance Estimation) và ước lượng vị trí (Position Estimation)

Ước lượng khoảng cách giữa 2 node: do chỉ số RSS giữa 2 node có 1 độ nhạy nhất định, nên sau khi thu được giá trị RSS và ước lượng được khoảng cách, cần dung 1 giải thuật làm mịn giá trị đo được để đạt độ chính xác cao trong việc xác định khoảng cách.

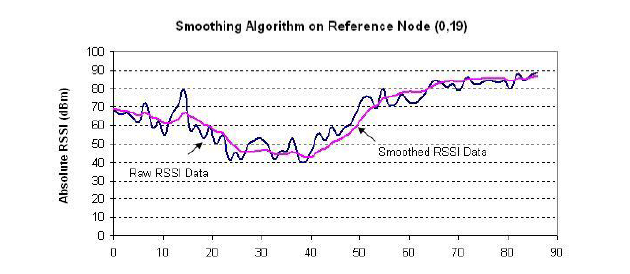


Figure 21. Làm mịn giá trị RSS thu được

Ước lượng vị trí : Sử dụng ít nhất 3 Reference Node để định vị vị trí của Blind Node thông qua giải thuật Trilateration.

Công thức định vị:

Đặc điểm của giải thuật sử dụng RSSI: độ chính xác và độ phân giải của vị trí phụ thuộc vào số lượng Reference Node, điều kiện môi trường. Do vậy, trong điều kiện thực tế, môi trường thay đổi làm giá trị RSSI và giải thuật tính toán bị giảm độ chính xác, đặc biệt với những đối tượng di chuyển.

**1.2 Angle of arrival (AOA):**

Tính toán và định vị dựa trên góc tới của tín hiệu. Bằng việc so sánh tín hiệu đến với 1 hướng tham chiếu, góc đến sẽ được tính toán. Phía nhận cũng có thể có định hướng riêng của mình để việc đo góc tốt hơn. Tuy nhiên giải thuật này cần hệ thống trang bị antenna định hướng.

* 1. **Time of arrival (TOA):**

Phương pháp này tính toán thời gian giữa 2 lần truyền nhận của Transmitter và Receiver. Phương pháp này có 2 hướng tiếp cận:

* Sử dụng 1 Transmitter gửi tín hiệu đến tất cả Receiver, sau đó các Receivers gửi lại thông tin thời gian cho hệ thống trung tâm tính toán
* Sử dụng nhiều Transmitter gửi tín hiệu đến 1 Receiver, Receiver sau đó tính toán thời gian đến của tất cả các tín hiệu sau đó được gửi lên hệ thống trung tâm. Hướng tiếp cận này có vấn đề là phải đồng bộ tất cả để thông tin thời gian được chính xác, mặt khác, tín hiệu có thể mất do nhiều tín hiệu được nhận cùng lúc.
  1. **Time difference of arrival (TDOA):**

Phương pháp cải tiến của TOA để giải quyết vấn đề đồng bộ thời gian và mất dữ liệu. Phương pháp này đòi hỏi ở Transmitter gửi 2 gói tin cho Receiver bằng 2 tốc độ truyền khác nhau. Ở Receiver có thể đo được sự khác biệt thời gian giữa 2 tín hiệu. Tận dụng sự khác nhau của thời gian đến, thời gian truyền của tín hiệu sẽ được tìm ra và nó đúng bằng thời gian truyền của tín hiệu

Range-free localization: không ước lượng khoảng cách tuyệt đối giữa 2 thiết bị dựa trên RSS. Vì vậy thiết kế phần cứng được đơn giản hóa đi nhiều, thích hợp cho môi trường WSNs.

**1.5 ROCRSSI**

Một trong những giải thuật định vị theo Range-free localization có thể kể đến như: Ring Overlapping based on Comparison of Received Signal Strength Indicator (ROCRSSI). Giải thuật này sử dụng đường vành khan để ước lượng khoảng cách tương đối của thiết bị phát và thu.

Ưu điểm của giải thuật ROCRSSI:

Không cần đối tượng Node gửi Control messages, chi phí giao tiếp phụ thuộc vào thiết bị nhận tín hiệu.

RSSI được so sánh với giải thuật Triangle-Overlapping và đã được chứng minh là hoạt động tốt hơn trong môi trường WSNs

Ring-Overlapping là 1 giải thuật mạnh mẽ trong môi trường truyền vô tuyến

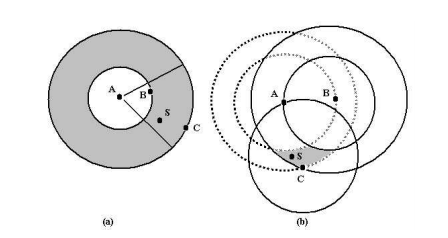


Figure 22. Một ví dụ về mô hình ROCRSSI

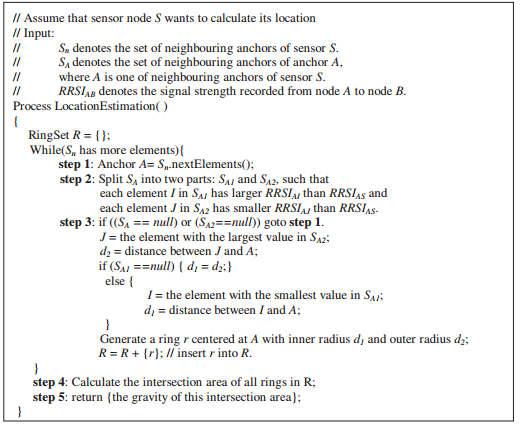


Figure 23. Giải thuật định vị ROCRSSI

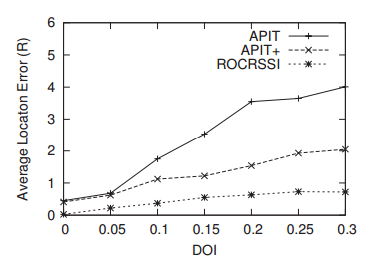


Figure 24. Bảng so sánh độ chính xác của các phương pháp định vị

**1.6 Định vị động:**

Trong thực tế, việc định vị các vị trí di chuyển trở nên rất khó khan bởi các lí do: một mặt, là do lỗi đo đạc và sự thay đổi điều kiện giữa các lần đo. Mặt khác, vì thiết bị cần định vị di chuyển nên giải thuật Triangulation dựa trên nhiều lần đo trở nên thiếu chính xác, vì chúng không cùng nhận được giá trị đo tại cùng 1 thời điểm. Vì vậy sinh ra phương pháp dự đoán trước vị trí của điểm chuyển động.

Định vị động được phát triễn thành 2 giải thuật:

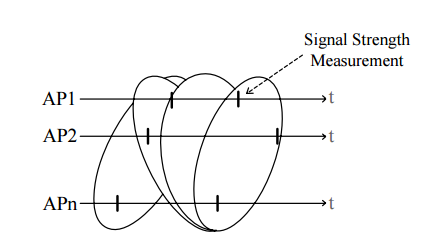
**Static Localisation Algorithm (SLA):** coi như các tín hiệu RSSI thu được là xấp xỉ cùng thời điểm với nhau. Tuy nhiên vì thiết bị cần định vị là di động nên độ chính xác của giải thuật này không cao vì RSSI thu về là không cùng lúc.

Figure 25. Độ trễ của tín hiệu nhận được trong giải thuật SLA

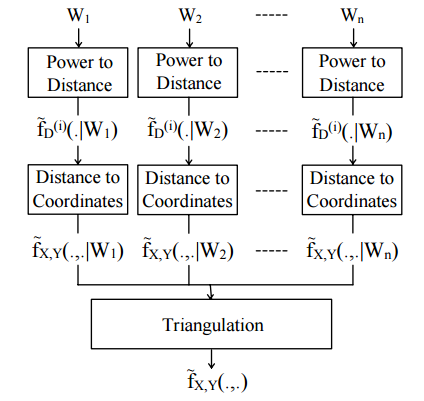


Figure 26. Giải thuật SLA

**Dynamic Localisation Algorithm (DLA):** để bù đắp cho thời gian delay, giải thuật DLA sử dụng mô hình chuyển động để mô phỏng

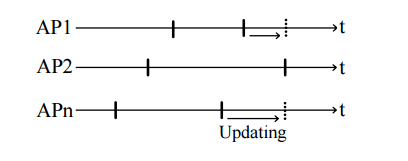


Figure 27. Độ trễ của các tín hiệu nhận được bù đắp

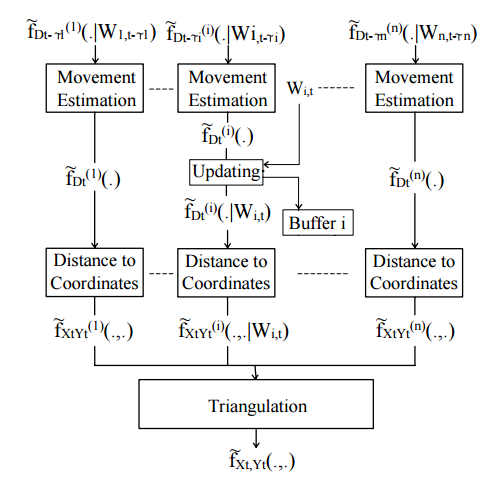


Figure 28. Giải thuật DLA

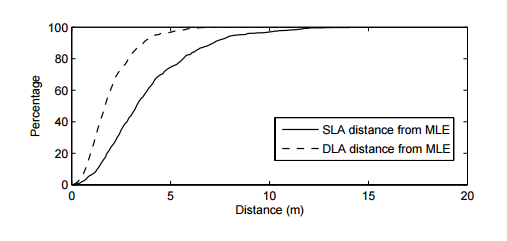


Figure 29. So sánh độ chính xác của 2 giải thuật SLA và DLA

**1.7 Approximate Point-Intriangulation Test (APIT)**

Phương pháp ước lượng điểm xấp xỉ xác định vị trí của nó bằng cách tham chiếu đến các vị trí xung quanh nó. Cho bởi công thức:

Với N là tổng các Node lân cận Node đối tượng trong vùng ước lượng

Tuy nhiên, giải thuật này có độ chính xác không cao do sự đơn giản và không đầy đủ. Mặt khác, giải thuật cũng cần số lượng Node đủ lớn. Về mặc định, nó là tất cả số Node xung quanh Node đối tượng mà nó có thể phát hiện và giao tiếp được. Tuy nhiên, kết quả ước lượng có thể không được chấp nhận nếu Node đối tượng ở quá gần biên của mạng lưới

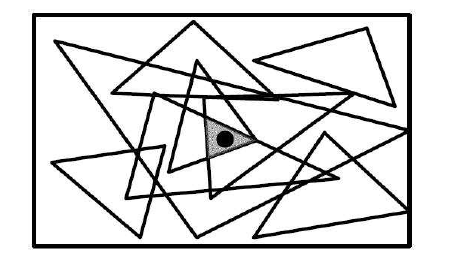


Figure 30. Định tuyến sử dụng giải thuật APIT

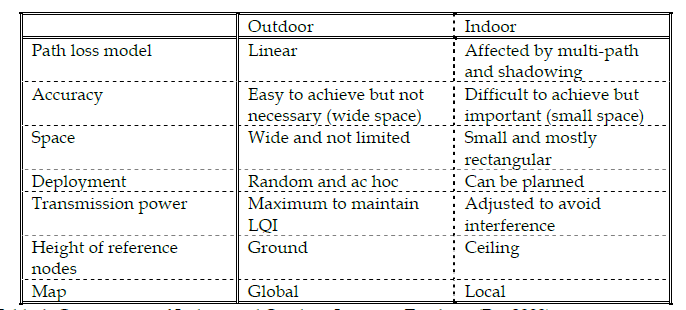


Figure 31. So sánh giữa 2 phương pháp định vị Indoor và Outdoor

Giải thuật định vị có sự khác nhau giữa trong nhà (Indoor) và ngoài trời (Outdoor), đặc biết với phương pháp RSSI, việc đo đạc thông số môi trường n và giá trị hằng số A là khác nhau với mỗi Reference Node. Ngoài ra số lượng Node trong phương pháp này cũng ảnh hưởng đến độ chính xác của chúng

Yêu cầu về phần cứng và chi phí thực thi với mỗi phương pháp là khác nhau. Từ việc xem xét các nghiên cứu của các khóa luận trước đây và khả năng tài chính hiện tại , nhóm nhận thấy, việc cần làm là xây dựng 1 hệ thống định vị Tĩnh ở Trong nhà (Indoor) để có thể so sánh và đối chiếu kết quả với thực tế (Vì có thể xác định được độ chính xác ) hơn là thực hiện đo đạc các giá trị ở bên ngoài (Outdoor)

1. **Lập trình truyền nhận dữ liệu giữa nhiều board Node iM880A**

**Cấu hình Board truyền nhận dữ liệu:**

* Frequency: 868MHz
* Output power: 20dBm
* Bandwith: 500kHz
* Spreading factor: 11 (11 chips for 1 symbol
* Coding rate: 4/5 (1 redundancy bit)
* Preamble length: 8
* Symbol time out: 5 symbols
* RX Timeout: 1000000
* Buffer size: 64
* Chế độ khởi động ban đầu: LOW\_POWER

**Ý nghĩa các hàm**

* OnTxDone: Hàm được thực thi khi sự kiện truyền tín hiệu xảy ra
  + Gọi hàm Radio.Sleep()
  + Đặt trạng thái là: TX
* OnRxDone: Hàm được thực thi khi sự kiện nhận tín hiệu xảy ra
  + Các tham số:
    - Con trỏ payload: trỏ tới địa chỉ chứa buffer
    - Size: độ lớn của buffer
    - RSSI: giá trị RSSI tính được khi nhận thông tin (dBm)
    - SNR: giá trị SNR thô được cho bởi radio hardware (FSK không có giá trị SNR, đơn vị SNR của LoRa được tính bằng dB)
  + Gán giá trị của biến toàn cục BufferSize bởi tham số size
  + Copy memory block được trỏ tới từ con trỏ payload vào buffer
  + Tiếp nhận các giá trị RSSI và SNR
  + Cài đặt lại trạng thái là RX
  + In giá trị từ payload ra LCD\
  + Gán data = payload
* OnTxTimeOut: Hàm được gọi khi hết thời gian chờ truyền tín hiệu
  + Gọi hàm Radio.Sleep()
  + Cài đặt trạng thái là TX\_TIMEOUT
* OnRxTimeOut: Hàm được gọi khi hết thời gian chờ nhận tín hiệu
  + Gọi hàm Radio.Sleep()
  + Cài đặt trạng thái là RX\_TIMEOUT
* OnRxError: Hàm được gọi khi xảy ra lỗi trong quá trình nhận tín hiệu
  + Gọi hàm Radio.Sleep()
  + Cài đặt trạng thái là RX\_ERROR

Hàm main():

* Lược qua các phần khởi tạo liên quan đến LCD và UART
* Radio initialization
  + Gọi các hàm tương ứng với các thuộc tính của đối tượng biến toàn cục RadioEvents (thuộc kiểu RadioEvents\_t đã được khởi tạo trước đó): TxDone, RxDone, OnTxTimeout, OnRxTimeout, Rx.Error. Đây được gọi là các driver callback
  + Khởi tạo Radio thông hàm Radio.Init
  + Cài đặt channel tương ứng với tần số đã cài đặt bên trên
* Hàm Radio.SetTxConfig: cài đặt các biến khi truyền tín hiệu
  + Danh sách tham số:
    - modem: radio modem sử dụng 🡪 1
      * 0: chọn FSK
      * 1: chọn LoRa
    - power: cài đặt output power 🡪 20dBm (đã đặt bên trên)
    - fdev: cài đặt frequency deviation (mô tả được chênh lệch lớn nhất giữa tần số điều biến và tần số sóng mang định danh) – chỉ dành cho modem FSK, nếu sử dụng LoRa thì đặt là 0
    - bandwith: cài đặt băng thông – không dùng cho FSK (đặt là 0), nếu là LoRa thì có 4 chế độ: 🡪 500 kHz
      * 0: 125kHz
      * 1: 250kHz
      * 2: 500kHz
      * 3: Reserved
    - datarate: cài đặt data rate 🡪 10
      * FSK: 600..300000 bit/s
      * LoRa: 6:64, 7:128, 8:256, 9:512, 10:1024, 11:2048, 12:4096 chips
    - coderate: cài đặt code rate – không dùng cho FSK (đặt là 0) 🡪 4/5
      * 1: 4/5
      * 2: 4/6
      * 3: 4/7
      * 4: 4/8
    - preambleLen: cài đặt độ dài của preamble 🡪 8
      * FSK: số bytes
      * LoRa: độ dài trong ký tự (phần cứng bổ sung thêm 4 ký tự)
    - fixLen: đóng gói với độ dài cố định 🡪 false
      * 0: không cố định
      * 1: cố định
    - crcOn: disable hoặc enable CRC 🡪 true
      * 0: disable
      * 1: enable
    - FreqHopOn: disable hoặc enable intra-packet frequency hopping 🡪 0
      * FSK: N/A (đặt là 0)
      * LoRa (0: OFF, 1: ON)
    - HopPeriod: số ký tự giữa mỗi hop 🡪 0
      * FSK: N/A (đặt là 0)
      * LoRa: số ký tự
    - iqInverted: đảo tín hiệu IQ 🡪 false
      * FSK: N/A (đặt là 0)
      * LoRa: (0: not inverted, 1: inverted)
    - Timeout: giới hạn thời gian truyền 🡪 3000000
* Hàm Radio.SetRxConfig: cài đặt các biến khi nhận tín hiệu
  + Danh sách tham số:
    - modem: radio modem sử dụng 🡪 1
      * 0: chọn FSK
      * 1: chọn LoRa
    - bandwith: cài đặt băng thông 🡪 500 kHz
      * FSK: >= 2600 và <= 250000Hz
      * LoRa:
        + 0: 125kHz
        + 1: 250kHz
        + 2: 500kHz
        + 3: Reserved
    - datarate: cài đặt data rate 🡪 10
      * FSK: 600..300000 bit/s
      * LoRa: 6:64, 7:128, 8:256, 9:512, 10:1024, 11:2048, 12:4096 chips
    - coderate: cài đặt code rate – không dùng cho FSK (đặt là 0) 🡪 4/5
      * 1: 4/5
      * 2: 4/6
      * 3: 4/7
      * 4: 4/8
    - bandwidthAfc: Cài đặt AFC Bandwidth (chỉ dùng cho FSK) 🡪 0
      * FSK : >= 2600 and <= 250000 Hz
      * LoRa: N/A (đặt là 0)
    - preambleLen: cài đặt độ dài của preamble 🡪 8
      * FSK: số bytes
      * LoRa: độ dài trong ký tự (phần cứng bổ sung thêm 4 ký tự)
    - symbTimeout: cài đặt giá trị RxSingle timeout (chỉ dùng cho LoRa) 🡪 5
      * FSK: N/A (đặt là 0)
      * LoRa: timeout tính bằng ký tự (symbols)
    - fixLen: đóng gói với độ dài cố định 🡪 0
      * 0: không cố định
      * 1: cố định
    - payloadLen: đặt độ dài cố định cho payload (khi fixLen được kích hoạt)
    - crcOn: disable hoặc enable CRC 🡪 true
      * 0: disable
      * 1: enable
    - FreqHopOn: disable hoặc enable intra-packet frequency hopping 🡪 0
      * FSK: N/A (đặt là 0)
      * LoRa (0: OFF, 1: ON)
    - HopPeriod: số ký tự giữa mỗi hop 🡪 0
      * FSK: N/A (đặt là 0)
      * LoRa: số ký tự
    - iqInverted: đảo tín hiệu IQ 🡪 false
      * FSK: N/A (đặt là 0)
      * LoRa: (0: not inverted, 1: inverted)
    - rxContinuous: đặt mode nhận dự liệu là Continuous 🡪 true
      * false: Single mode
      * true: Continuous mode
* Hàm Radio.Rx: Đặt radio vào trạng thái nhận tín hiệu vào tham số đầu vào là thời gian timeout (nếu để ở chế độ Continuous thì đặt là 0)

1. **Giao tiếp module Bluetooth và ứng dụng Android**
2. **Giao tiếp với module Bluetooth HC-05**

HC-05 là module Bluetooth thông dụng được sử dụng khá phổ biến hiện nay. HC-05 có 2 chế độ làm việc:

* **Order-response:** người dùng có thể sử dụng chế độ này để truyền lệnh AT và điều chỉnh các thông số (tên, mật khẩu kết nối, baudrate,...) trên HC-05
* **Automatic connection:** module dựa vào các thiết lập đã được cài đặt trước đó và thực hiện kết nối, truyền, nhận dữ liệu

2 chế độ làm việc trên được sử dụng bằng cách điều khiển chân PIO11 trên HC-05:

* **Order-response:** tích cực mực cao tại chân PIO11 (PIN 34) rồi sau đó cấp nguồn cho HC-05. Một cách khác để truy cập vào chế độ này là bấm giữ button trên module HC-05 và cấp nguồn cho module.
* **Automatic connection:** cấp nguồn bình thường cho module

HC-05 phân biệt các chế độ làm việc thông qua việc hiển thị chớp tắt của LED được tích hợp trên module. Nếu LED nháy liên tục thì đó là chế độ Automatic connection, LED nháy với chu kỳ 2 giây là chế độ module đã được kết nối, nháy với chu kỳ 1 giây chế độ order-response.

Khi kết nối với bluetooth HC-05 cần quan tâm đến baudrate của module để có thể giao tiếp được với module. Với cách kết nối trên, baudrate mặc định của module sẽ là 38400, do đó khi thiết lập kết nối UART để truyền lệnh xuống HC-05 cũng cần cài đặt thông số tương ứng. Khi đã vào chế độ order-response, ta có thể cài đặt lại các thông số này thông qua lệnh AT (danh sách các lệnh AT hỗ trợ được viết trong tài liệu tham khảo).

Khi kết nối bluetooth với module HC-05, khi cần truyền dữ liệu ta chỉ cần sử dụng giao tiếp UART thông thường là có thể truyền dữ liệu đến các thiết bị khác thông qua kết nối bluetooth này.

1. **Ứng dụng Android LoRen Bluetooth Chat**

Các chức năng chính của ứng dụng:

* Thiết lập kết nối bluetooth với module HC-05 và nhận dữ liệu RSSI từ module LoRa
* Tính toán và chuyển giá trị RSSI thành giá trị khoảng cách
* Hiển thị khoảng cách lên bản đồ realtime
  1. **Bluetooth chat**

Sử dụng Bluetooth API được Google hỗ trợ để thực hiện chức năng kết nối và giao tiếp với module HC-05.

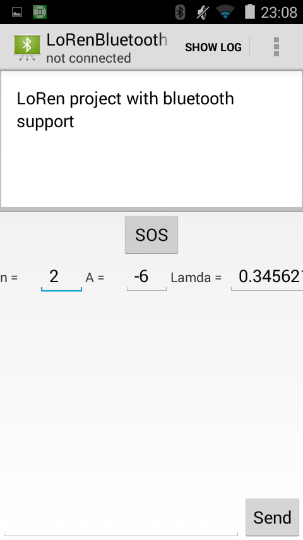


Figure 32. Giao diện bluetooth chat tích hợp với thiết đặt thông số công thức tính RSSI

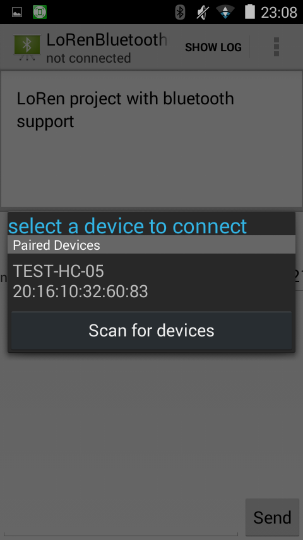


Figure 33. Giao diện danh sách thiết bị bluetooth kết nối

Module bluetooth nên được kết nối thông qua trình kết nối mặc định của điện thoại trước đó.

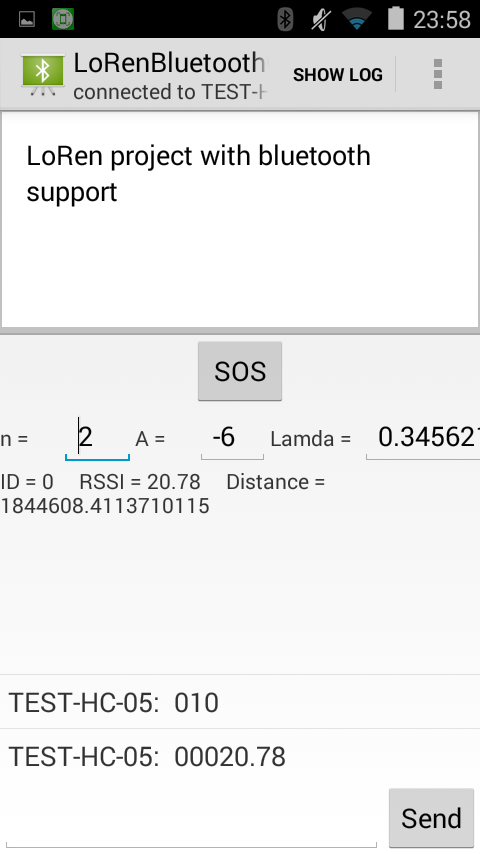


Figure 34. Hiển thị thông tin gói tin nhận được

Gói tin truyền đi từ module LoRa đến điện thoại được định dạng như sau:

***[0][ID][Giá trị RSSI]***

Trong đó:

* [0]: Ký tự mở đầu cho gói tin
* [ID]: 3 chữ số
* [Giá trị RSSI]: 4 chữ số ở phần nguyên (không quan tâm đến số chữ số phần thập phân)

Định dạng gói tin truyền đi là một chuỗi ký tự gồm toàn số nên sẽ dễ dàng hơn trong việc chuyển chuỗi này thành số float và xử lý để tách thông tin gói tin. Cụ thể, ta chia giá trị nhận được cho 10.000 để lấy được giá trị ID và trừ cho ID\*10000 để lấy được giá trị RSSI.

Ngoài ra, giao diện cũng hỗ trợ người dụng thiết đặt các thông số cho công thức tính khoảng cách từ RSSI, các tham số này đã được gán giá trị mặc định ban đầu như hình trên.

* 1. **Hiển thị khoảng cách theo thời gian thực**

Trong quá trình kết nối điện thoại với module LoRa để nhận dữ liệu, các gói tin sẽ được truyền về liên tục, do đó để có thể theo dõi thông tin truyền về, ứng dụng sẽ quét 5 giây một lần các giá trị gói tin truyền về và cập nhật lại khoảng cách tính toán dựa trên giá trị RSSI tại thời điểm đó.

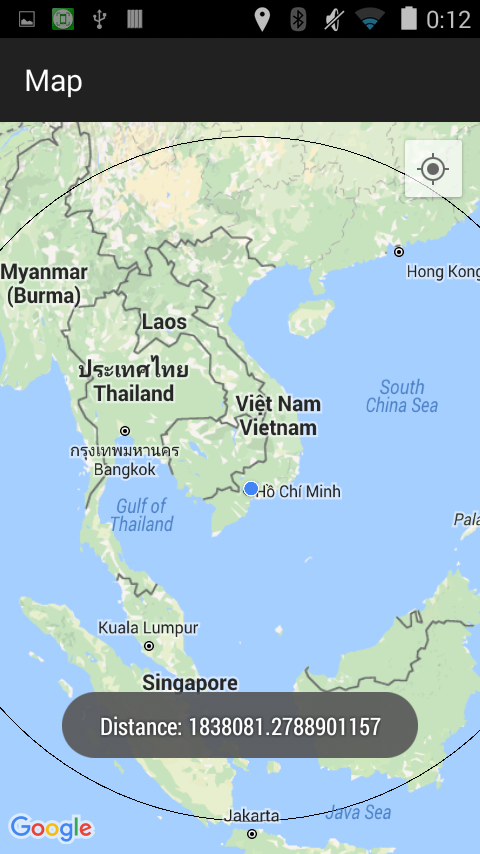


Figure 35. Bấm nút SOS giao diện sẽ chuyển sang hiển thị Google map và khoảng cách dựa vào chỉ số RSSI

Dựa trên khoảng cách tính toán được, ứng dụng sẽ hiển thị khoảng cách đó thông qua 1 vòng tròn trên bản đồ. Trong hình trên, do công thức tính toán bị sai sót nên đường tròn thể hiện khá lớn.

Nếu như trong lúc hiển thị, các gói tin truyền về chứa giá trị RSSI được cập nhật thì đường tròn và toast hiển thị cũng sẽ được cập nhật với chu kỳ 5 giây 1 lần.

1. **Mô phỏng định vị trên phần mềm Matlab**

Sơ đồ hoạt động hệ thống định vị:

Figure 36. Sơ đồ hoạt động của hệ thống định vị

Anchor 2

Anchor 1

Anchor 3

Blind Node

**RSSI 1 (d1)**

**)**

**RSSI 2 (d2)**

**RSSI 3 (d3)**

Để định vị được vị trí của 1 Blind Node, ta cần ít nhất 3 Node Anchor. Trong trường hợp này là Anchor 1, Anchor 2, Anchor 3.

Mỗi Anchor Node cần cung cấp cho hệ thống tính toán khoảng cách của chúng đối với Blind Node thông qua giá trị RSSI.

Máy tính được kết nối với các Node Anchor để nhận dữ liệu và tính toán, mô phỏng kết quả.

* RSSI 1 lưu chỉ số RSSI giữa Blind Node và Anchor 1
* RSSI 2 lưu chỉ số RSSI giữa Blind Node và Anchor 2
* RSSI 3 lưu chỉ số RSSI giữa Blind Node và Anchor 3

Máy tính nhận 3 chỉ số RSSI, phân biệt chúng thông qua chỉ số ID của các gói tin nhận được. Lưu thông tin lại thành Text file

Sơ đồ quá trình thu thập RSSI của 3 Node:

Figure 37. Sơ đồ quá trình thu nhận RSSI của 3 Anchor Node

Lưu RSSI của d1

Lưu RSSI của d2

Lưu RSSI của d3

Lưu RSSI thành file text lưu trên máy tính

Bắt đầu

Blind Node gửi tin tới các Node Anchor 1, Anchor 2, Anchor 3

1

2

ID Message=?

3

Thiết kế Tool nhận Uart và chuyển thông tin gói tin thành các Text file:

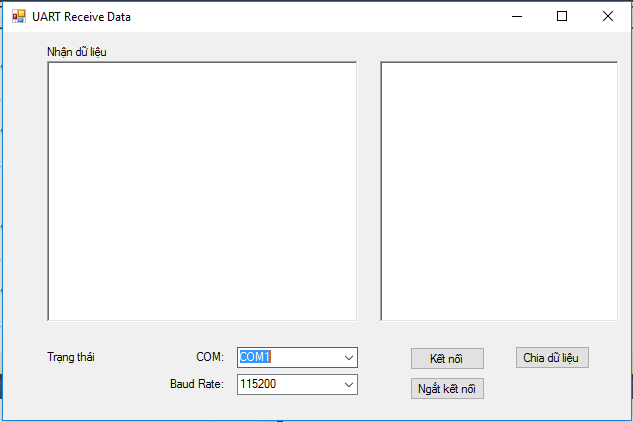


Figure 38. Giao diện công cụ nhận dữ liệu RSSI thông qua giao tiếp UART

Kết quả của quá trình thu thập thông tin:

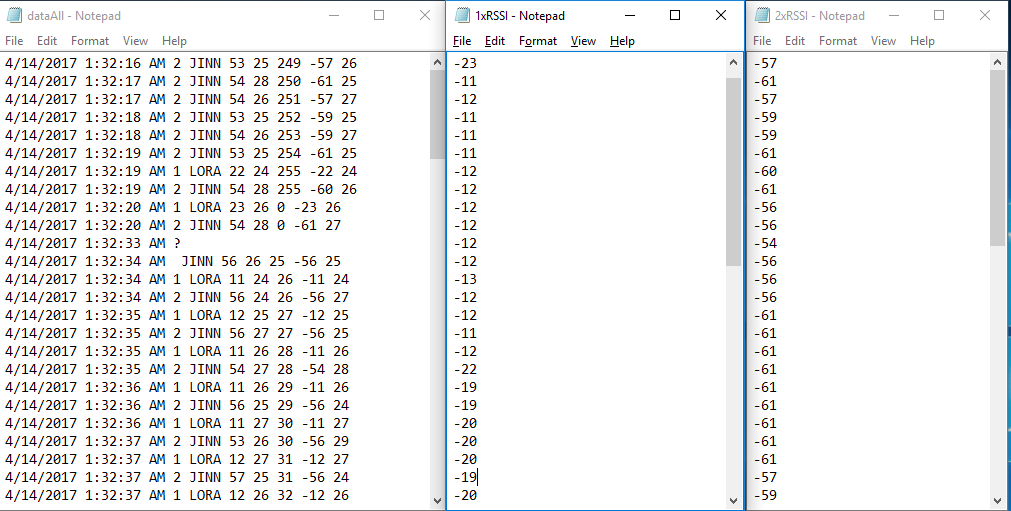


Figure 39. Các gói tin nhận được

Các giá trị RSSI lưu thành từng file, sau đó được mô phỏng tọa độ trên phần mềm Matlab

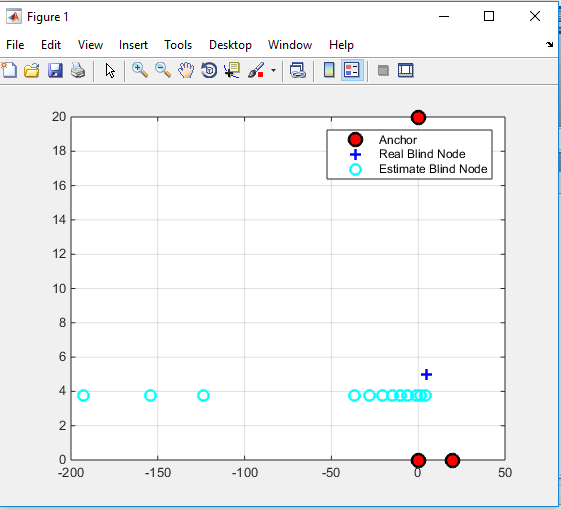


Figure 40. Mô phỏng định vị trên phần mềm Matlab

# **Giao tiếp với SD Card và lưu trữ dữ liệu**

1. **FATFS – Generic FAT Filesystem Module**
   1. **FatFs là gì**

FatFs là một module filesystem FAT/exFAT chung dành cho các hệ thống nhúng nhỏ. Module FatFs được viết theo quy chuẩn ANSI C (C89) và hoàn toàn tách biệt khỏi lớp disk I/O. Vì thế FatFs được xem là một nền tảng độc lập. Nó có thể tích hợp vào các bộ vi xử lý nhỏ với tài nguyên hạn chế như 8051, PIC,AVR, ARM,v.v...

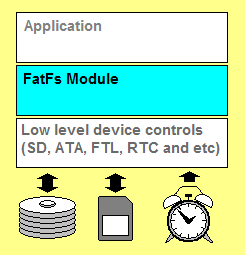


Figure 41. Module FatFS

* 1. **Các đặc tính của FatFS**

-Filesystem FAT/exFAT tương thích với DOS/Windows

-Nền tảng độc lập, dễ dàng sử dụng trên nhiều hệ thống

-Chiếm dung lượng nhỏ

-Hỗ trợ nhiều lựa chọn tùy chỉnh:

* Hỗ trợ nhiều volumes (ổ đĩa vật lý hoặc partitions)
* Hỗ trợ nhiều code pages bao gồm DBCS
* Hỗ trợ file với tên dài bằng OEM hay Unicode
* exFat filesystem
* Đảm bảo tiến trình an toàn
* Kích thước vector cố định hoặc không cố định
* Hỗ trợ read-only, các API tùy chọn, I/O buffer,...
  1. **Application interface**

FatFs hỗ trợ rất nhiều hàm filesystem dành cho các ứng dụng, chia thành các loại sau:

* File Access: chứa các hàm hỗ trợ việc tạo, mở, đóng, ghi, xóa, truy cập vào file
* Directory Access: chứa các hàm hỗ trợ việc đóng, mở, kiệt kê, tìm và mở các file trong một thư mục
* File and Directory Management: quản lý thông tin file hoặc thư mục, tạo, xóa thư mục
* Volume Management and System Configuration: quản lý các volume và tùy chỉnh hệ thống

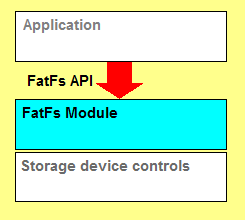


Figure 42. Các FatFs API là cầu nối giữa tầng ứng dụng và FatFs Module

* 1. **Media Access Interface**

Bởi vì module FatFs là lớp filesystem độc lập khỏi nền tảng cũng như các phương tiện lưu trữ, nó hoàn toàn tách biệt khỏi các thiết bị phần cứng như thẻ nhớ, hard disk hay bất cứ thiết bị lưu trữ nào khác. Các module điều khiển thiết bị cấp thấp không phải là một phần của module FatFs và nó cần được cung cấp bởi bộ thực thi. FatFs truy cập vào các thiết bị lưu trữ thông qua một interface truy cập media đơn giản dưới đây:

* disk\_status: Đọc tình trạng của ổ đĩa
* disk\_ initialize: Khởi tạo thiết bị
* disk\_read: đọc các sector
* disk\_read: ghi vào các sector
* disk\_ioctl: điều khiển các hàm phụ thuộc thiết bị
* get\_fattime: đọc thời gian hiện tại

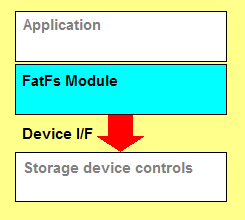


Figure 43. Module FatFs giao tiếp với các thiết bị lưu trữ thông qua Device I/F

1. **Giao tiếp và lưu trữ dữ liệu**

Thẻ nhớ là thiết bị lưu trữ dữ liệu, sử dụng công nghệ flash để ghi xóa bộ nhớ. Trong đó thẻ uSD, SD được sử dụng rộng rãi và được nhiều người biết đến nhất, ứng dụng trong các thiết bị cầm tay, máy ảnh kỹ thuật số, smartphone,…



Figure 44. Thẻ nhớ Micro SD

Có 3 cách giao tiếp thẻ nhớ:

* Giao tiếp thông qua SPI
* Giao tiếp bằng SDIO 1bit
* Giao tiếp bằng SDIO 4bit

Trong đồ án này, ta chọn cách giao tiếp thứ nhất

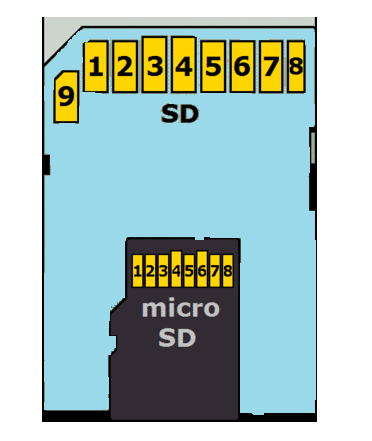


Figure 45. Sơ đồ nối chân của thẻ SD và Micro SD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SPI BUS MODE | | | | | |
| SD Pin | microSD Pin | Name | I/O | Logic | Description |
| 1 | 2 | nCS | I | PP | SPI Card Select [**CS**] (Negative Logic) |
| 2 | 3 | DI | I | PP | SPI Serial Data In [**MOSI**] |
| 3 |  | VSS | S | S | Ground |
| 4 | 4 | VDD | S | S | Power |
| 5 | 5 | CLK | I | PP | SPI Serial Clock [S**CLK**] |
| 6 | 6 | VSS | S | S | Ground |
| 7 | 7 | DO | O | PP | SPI Serial Data Out [**MISO**] |

Figure 46. Tên các chân Pin của SD Card và MicroSD

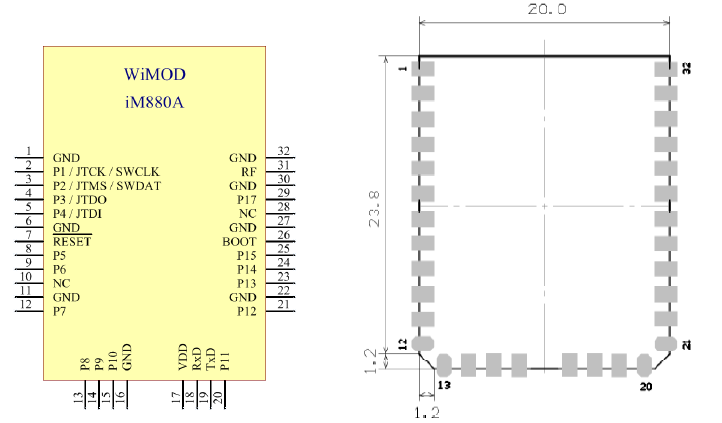


Figure 47. Sơ đồ chân ra của module iM880A

|  |  |  |  |
| --- | --- | --- | --- |
| **PIN Name** | **PIN Type** | **MCU Pin** | **Description** |
| P8 | D IN/OUT | PB\_15(P28) | Digital IO/ SPI2\_MOSI |
| P7 | D IN/OUT | PB\_14(P27) | Digital IO/ SPI2\_MISO |
| P9 | D IN/OUT | PB\_13(P26) | Digital IO/ SPI2\_CLK |
| P10 | D IN/OUT | PB\_12(P25) | Digital IO/ SPI2\_NSS |

Figure 48. Bảng ra chân PIN của iM880A

**Lập trình:**

Ta sử dụng thêm thư viện của FatFS để lập trình đọc ghi file vào thẻ SD

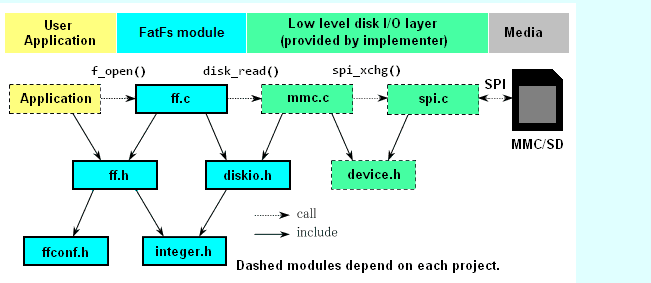


Figure 49. Sơ đồ liên kết các file trong project

**Kết quả:**

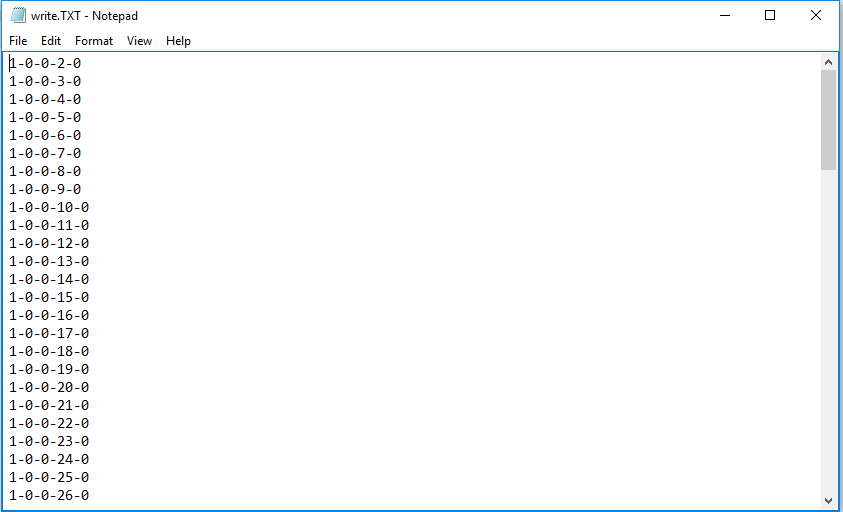


Figure 50. Dữ liệu gửi được lưu vào file "write.txt"

# **Kết quả quá trình test thử và định vị**

## **1. Dữ liệu RSSI khoảng cách 1 m:**

Figure 51. Chỉ số RSSI

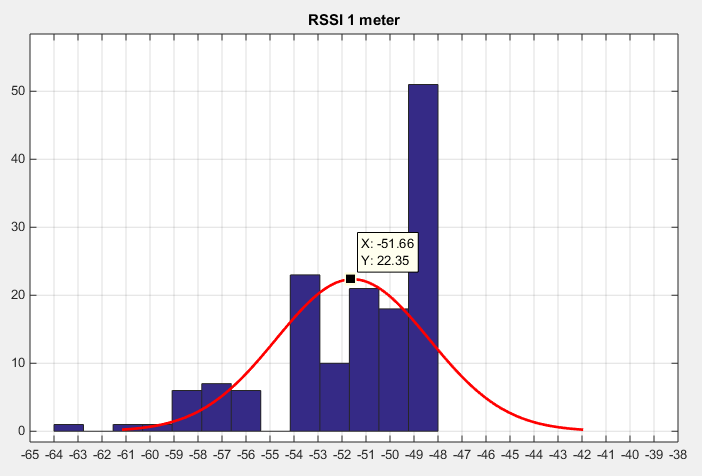


Figure 52. Biểu diễn mật độ của các giá trị RSSI

RSSI của 1m là khoảng -51

RSSI khoảng cách m:

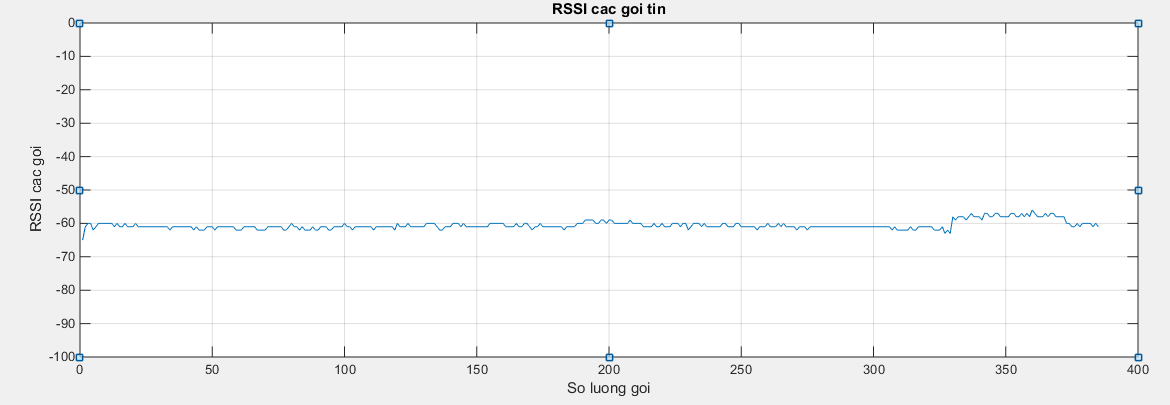


Figure 53. Giá trị RSSI

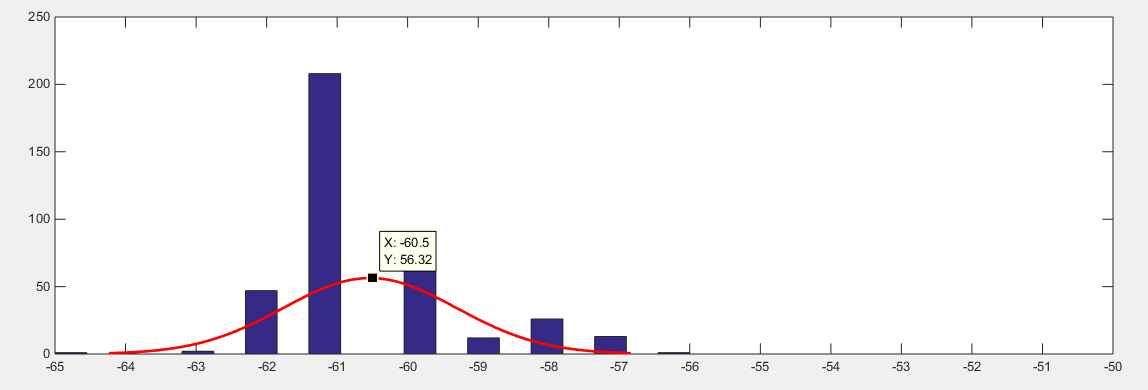


Figure 54. Biểu diễn mật độ

Ở khoảng cách m (~ 7,07m) RSSI là -61

Theo công thức:

Với A=-51, RSSI=-61, d=7.07 => path loss n=1.17

1. **Test khoảng cách 12 tầng**

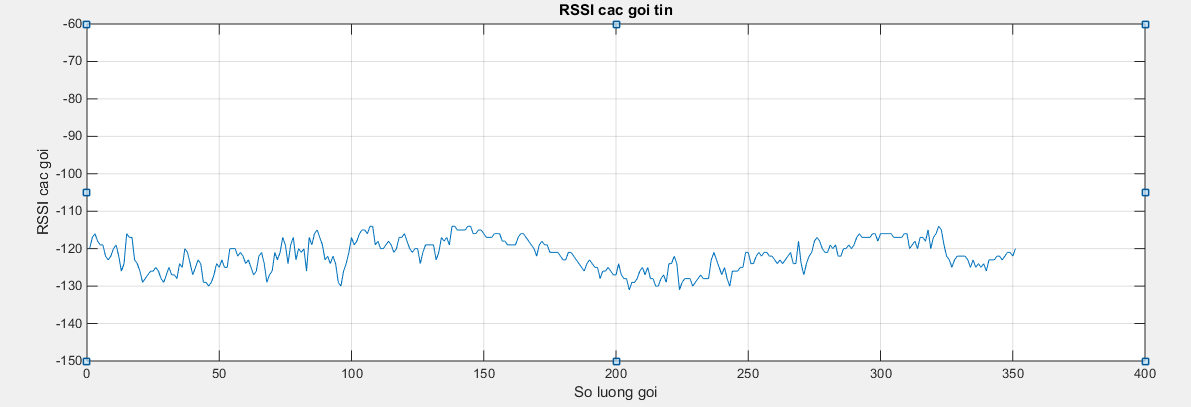


Figure 55. Giá trị RSSI

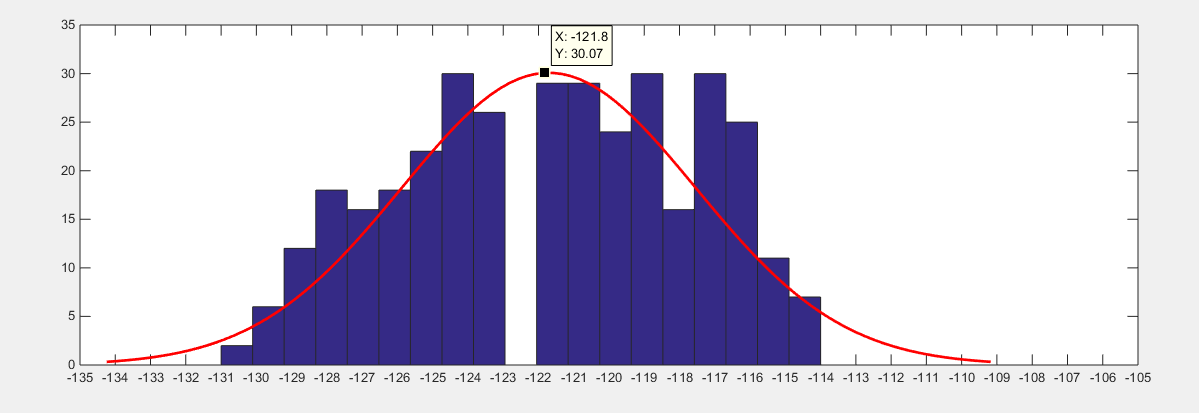


Figure 56. Biểu diễn mật độ

1. **Test định vị thực tế:**
2. **Sơ đồ hệ thống**

Sơ đồ tọa đồ demo hệ thống được thể hiện như sau:

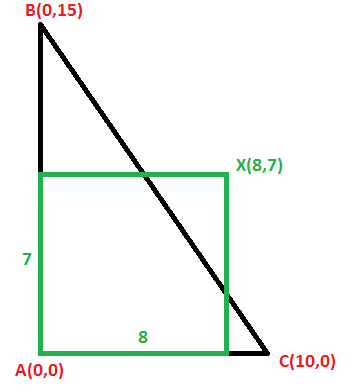


Figure 57. Sơ đồ tọa độ demo

**Quy trình demo:**

* Bước 1: Đặt node phát tín hiệu tại điểm X(8,7)
* Bước 2: Dùng 1 node nhận nhận tín hiệu lần lượt tại các vị trí A(0, 0), B(0, 15), C(10, 0). Tại mỗi điểm thực hiện nhận ít nhất 300 gói tin
* Bước 3: Đo tín hiệu RSSI trên khoảng cách 1m
* Bước 4: Đo tín hiệu RSSI trên khoảng cách 3m

1. **Tính toán hệ số môi trường và giá trị RSSI trên 1m**

Sử dụng bộ lọc Gauss để lọc dữ liệu và áp dụng phân phối chuẩn trên các mẫu thử lấy từ Bước 3 và Bước 4 phía trên để lấy ra giá trị RSSI chuẩn nhất trên khoảng cách 1m và 3m.

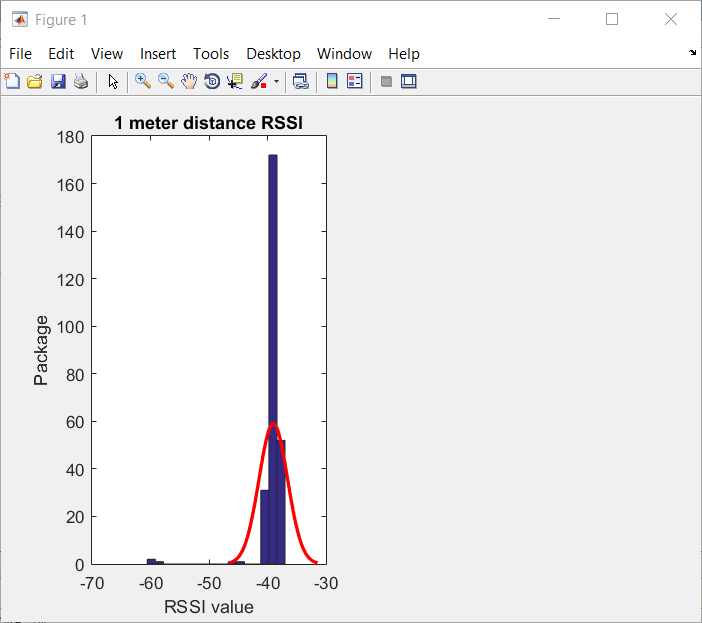


Figure 58. Biểu đồ thể hiện phân phối chuẩn giá trị RSSI trên khoảng cách 1m

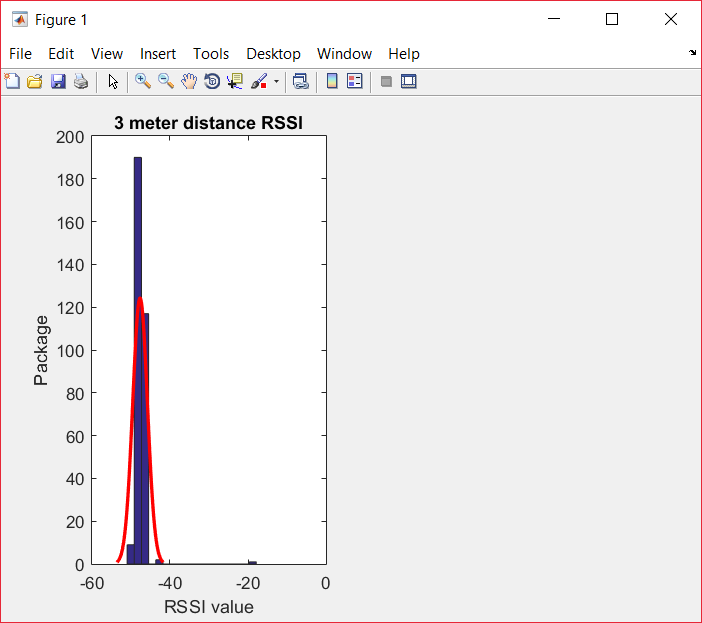


Figure 59. Biểu đồ thể hiện phân phối chuẩn giá trị RSSI trên khoảng cách 3m

Sau khi áp dụng các công thức tính toán trên Matlab, ta rút ra được các giá trị:

*RSSI\_1m = -39.0077*

*RSSI\_3m = -47.6740*

Áp dụng công thức tíng giá trị RSSI sau:

(1)

Trong đó: A là giá trị RSSI trên 1m, n là hệ số môi trường, d là khoảng cách

Từ 2 giá trị RSSI 1m và 3m phía trên, ta có được hệ phương trình với 2 ẩn là A và n, giải hệ ta có kết quả:

*A = -39.0077*

*N = 1.8164*

1. **Tính giá trị khoảng cách từ các điểm nhận**

Trong số 300 mẫu tin nhận được, ta chọn ra 200 mẫu có giá trị ổn định lấy từ những mẫu dưới cùng ngược lên. Dựa trên mẫu thu được, áp dụng bộ lọc gauss và phân phối chuẩn để chuẩn hóa dữ liệu thu được. Từ những dữ liệu đã qua xử lý đó, áp dụng công thức (1) với A và n vừa tìm được, ta thu được kết quả sau:

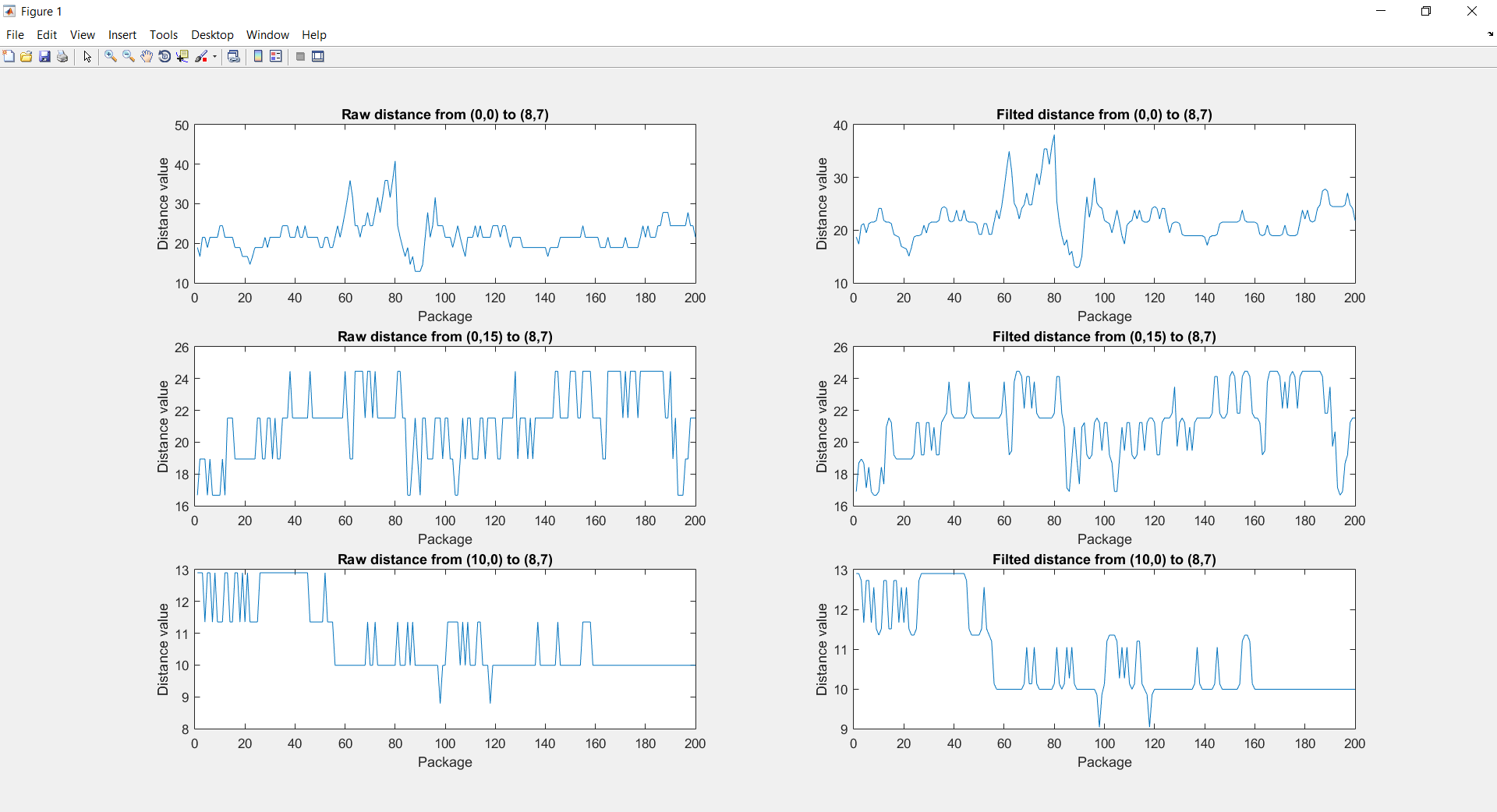


Figure 60. Dữ liệu sau khi được lọc Gauss, những biểu đồ bên trái là dữ liệu thô, bên phải là dữ liệu đã qua bộ lọc

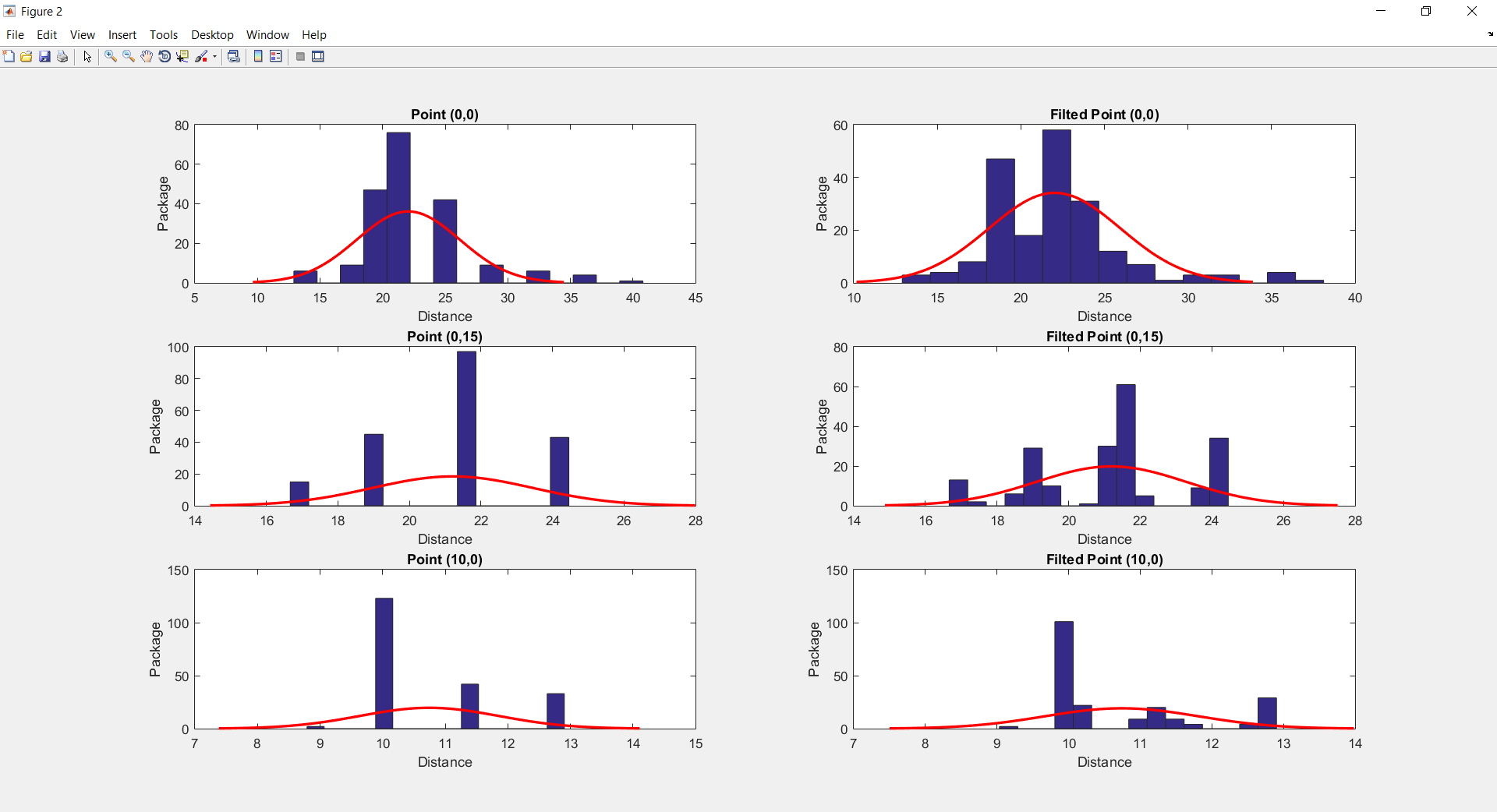


Figure 61. Áp dụng phân phối chuẩn đối với dữ liệu vừa được lọc

Ta có kết quả khoảng cách như sau:

* Khoảng cách từ A đến X: dA = 22.0235m
* Khoảng cách từ B đến X: dB = 21.1926m
* Khoảng cách từ C đến X: dC = 10.7413m

Hiển thị trên biểu đồ như sau:

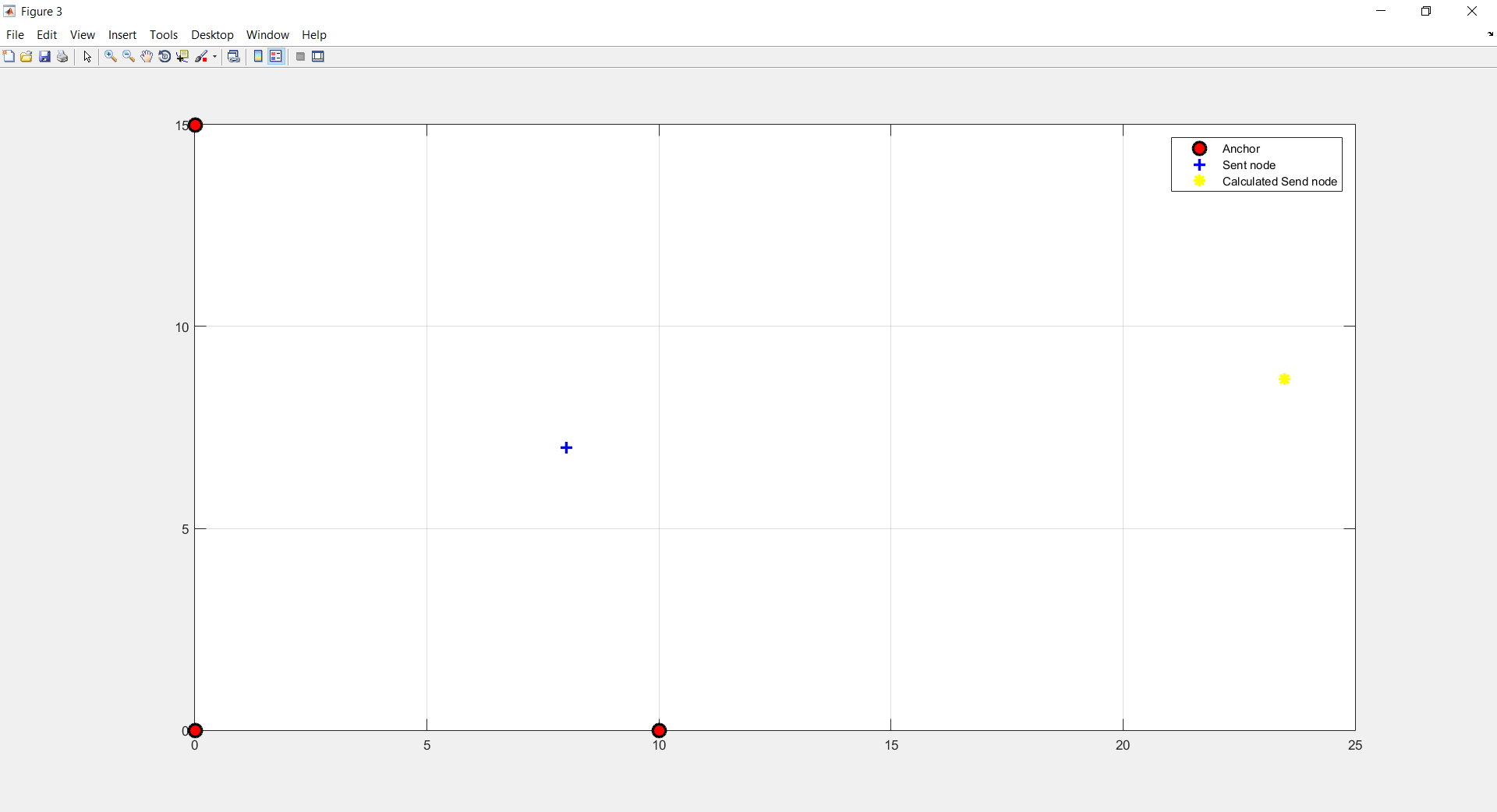


Figure 62. Biểu diễn vị trí các node nhận và vị trí node gửi tính toán được so với thực tế

# **Tài liệu tham khảo**

* + - 1. Vũ Đức Lung, Phan Đình Duy, Phạm Quốc Cường- *“Giải thuật định vị vị trí trong không gian 3D cho thẻ RFID dựa vào cường độ tính hiệu RSS”-* Tạp chí khoa học Trường Đại học Cần Thơ.
      2. Erin-Ee-Lin Lau, Boon-Giin Lee, Seung-Chul Lee, Wan-Young Chung- *“Enhanced RSSI- Based high accuracy Real-Time user location tracking system for indoor and outdoor environments”-* International Journal on smart sensing and Intelligent system, vol. 1, no. 2, June 2008.
      3. Santiago Ezpeleta, Jose M. Claver, Juan J.Perez- Solano, V. Marti- *“RF- Based Location Using Interpolation Functions to Reduce Fingerprint Mapping”-* Open access Sensor ISSN 1424-8220 (mdpi.com/journal/sensors).
      4. Oguejiofor O.S, Aniedu A.N, Ejiofor H.C, Okolibe A.U - *“Trilaterration Based localization Algorithm for Wireless Sensor Network”-* International Journal of Science and Modern Engineering (IJISME).
      5. Guillaume Monghal, Yann Malidor, Jo˜ao Figueiras, Hans-Peter Schwefel, Istv´an Z. Kov´acs- *“Enhanced triangulation method for positioning of moving devices”*- Department of Communication Technology Aaborg University.
      6. Shelei, Xueyong Ding, Tingting Yang- *“Analysis of Five Typical Localization Algorithm for Wireless Sensor Networks”-* Polytechnical School, Sanya College, Sanya, China.