

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

CỦ QUỐC CƯỜNG
HUỲNH CHÍ THÀNH

LUẬN VĂN TỐT NGHIỆP
HỆ THỐNG ĐỊNH VỊ TRONG NHÀ DỰA TRÊN CÔNG NGHỆ
BLUETOOTH LOW ENERGY SỬ DỤNG PHƯƠNG PHÁP GÓC
TỐI (AOA)

INDOOR POSITIONING SYSTEM BASED ON BLUETOOTH LOW
ENERGY TECHNOLOGY USING ANGLE-OF-ARRIVAL (AOA)
METHOD

CỦ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

TP. HỒ CHÍ MINH, 2025

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

CÙ QUỐC CƯỜNG – 2112956

HUỲNH CHÍ THÀNH – 2114774

LUẬN VĂN TỐT NGHIỆP

**HỆ THỐNG ĐỊNH VỊ TRONG NHÀ DỰA TRÊN CÔNG NGHỆ
BLUETOOTH LOW ENERGY SỬ DỤNG PHƯƠNG PHÁP GÓC
TỚI (AOA)**

**INDOOR POSITIONING SYSTEM BASED ON BLUETOOTH LOW
ENERGY TECHNOLOGY USING ANGLE-OF-ARRIVAL (AOA)
METHOD**

CỦ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN
THS. HỒ THANH PHƯƠNG

TP. HỒ CHÍ MINH, 2025

**KHÓA LUẬN TỐT NGHIỆP ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC BÁCH KHOA –ĐHQG -HCM**

Cán bộ hướng dẫn Khóa luận tốt nghiệp :
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 1 :
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 :
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Khóa luận tốt nghiệp được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG
Tp.HCM ngày tháng năm

Thành phần Hội đồng đánh giá khoá luận tốt nghiệp gồm:
(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ khóa luận tốt nghiệp)
1.
2.
3.
4.
5.

Xác nhận của Chủ tịch Hội đồng đánh giá khóa luận tốt nghiệp và Chủ nhiệm Bộ
môn sau khi luận văn đã được sửa chữa (nếu có).

CHỦ TỊCH HỘI ĐỒNG

CHỦ NHIỆM BỘ MÔN.....

TP. HCM, ngày.....tháng.....năm.....

NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN

Tên luận văn:

HỆ THỐNG ĐỊNH VỊ TRONG NHÀ DỰA TRÊN CÔNG NGHỆ
BLUETOOTH LOW ENERGY SỬ DỤNG PHƯƠNG PHÁP GÓC TỐI
(AOA)

Nhóm Sinh viên thực hiện:

Cù Quốc Cường

Huỳnh Chí Thành

Cán bộ hướng dẫn:

2112956 Hồ Thanh Phương

2114774 Hồ Thanh Phương

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại
Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Cù Quốc Cường /10

Huỳnh Chí Thành /10

Cán bộ hướng dẫn

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày.....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ PHẢN BIỆN**

Tên luận văn:

**HỆ THỐNG ĐỊNH VỊ TRONG NHÀ DỰA TRÊN CÔNG NGHỆ BLUETOOTH
LOW ENERGY SỬ DỤNG PHƯƠNG PHÁP GÓC TỐI (AOA)**

Nhóm Sinh viên thực hiện:

Cù Quốc Cường	2112956
Huỳnh Chí Thành	2114774

Cán bộ phản biện:

Đánh giá Luận văn

5. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

6. Về nội dung luận văn:

7. Về tính ứng dụng:

8. Về thái độ làm việc của sinh viên:

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Cù Quốc Cường /10

Huỳnh Chí Thành /10

Người nhận xét

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Quý Thầy cô khoa Điện – Điện tử nói chung và Bộ môn Điều khiển Tự động nói riêng trong suốt thời gian qua. Quãng thời gian học tập tại trường Đại học Bách khoa TP.HCM là một chặng đường gian nan và vất vả, tuy nhiên chúng em cảm thấy may mắn khi đã được các Quý Thầy cô nhiệt tình chỉ dẫn, truyền đạt những kiến thức, kinh nghiệm kỹ năng quý báu, nhờ vậy đã giúp ích rất nhiều trong quá trình học tập cũng như hoàn thành luận văn tốt nghiệp này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn và gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn trực tiếp ThS. Hồ Thanh Phương. Cô đã tận tình chỉ dẫn, giải đáp các khúc mắc cũng như tạo điều kiện thuận lợi giúp chúng em giải quyết được các vấn đề gặp phải trong quá trình nghiên cứu và hoàn thành đề tài một cách tốt nhất.

Chúng em cũng gửi lời cảm ơn đến các bạn bè cùng khóa đã giúp đỡ, hỗ trợ nhau hết mình trong quá trình học tập và hoàn thành luận văn này. Lời cảm ơn này cũng xin gửi đến ba mẹ và gia đình đã luôn ở cạnh bên động viên chúng em vượt qua những giai đoạn khó khăn trong quá trình học tập.

Do kiến thức của bản thân còn hạn chế và thiếu kinh nghiệm thực tiễn nên nội dung luận văn khó tránh khỏi những thiếu sót. Chúng em rất mong nhận sự góp ý, chỉ dạy thêm từ Quý Thầy cô.

Cuối cùng, chúng em xin chúc Quý Thầy Cô luôn có thật nhiều sức khỏe và đạt được nhiều thành công trong công việc.

Thành phố Hồ Chí Minh, tháng 5 năm 2025

Sinh viên thực hiện

Huỳnh Chí Thành

Cù Quốc Cường

TP. HCM, ngày.....tháng.....năm.....

ĐỀ CƯƠNG CHI TIẾT

TÊN LUẬN VĂN: HỆ THỐNG ĐỊNH VỊ TRONG NHÀ DỰA TRÊN CÔNG NGHỆ BLUETOOTH LOW ENERGY SỬ DỤNG PHƯƠNG PHÁP GÓC TÓI (AOA)

Cán bộ hướng dẫn: ThS. Hồ Thanh Phương

Thời gian thực hiện: Từ ngày 07/01/2025 đến ngày 07/05/2025

Sinh viên thực hiện:

Cù Quốc Cường – 2112956

Huỳnh Chí Thành – 2114774

Nội dung đề tài:

- + Nghiên cứu lý thuyết về công nghệ định vị trong nhà sử dụng BLE và phương pháp AoA, bao gồm nguyên lý hoạt động và các ưu, nhược điểm.
- + Thiết kế và tích hợp Tag C209 của u-blox phù hợp với yêu cầu của hệ thống định vị AoA, bao gồm thiết kế mạch, lập trình Firmware và tối ưu hóa tiêu thụ năng lượng.
- + Triển khai hệ thống sử dụng hai anchor ANT-B10 để xác định vị trí của Tag với độ chính xác cao trong môi trường trong nhà.
- + Phát triển thuật toán xử lý tín hiệu để tính toán góc đến (AoA) và vị trí của Tag dựa trên dữ liệu thu được từ các Anchor.

Kế hoạch thực hiện:

08/01/2025 – 22/01/2025: Tìm hiểu cơ sở lý thuyết về hệ thống định vị trong nhà, Module NINA-B406, Tag C209, Anchor ANT-B10 và các giao thức truyền thông sử dụng.

22/01/2025 – 14/02/2025: Thực hiện Hardware: thiết kế Tag C209 và gia công mạch PCB của Tag C209.

14/02/2025 – 20/3/2025: Thực hiện Firmware và Software: : vẽ mặt bằng 2D, hiển thị tọa độ các Tag và Anchor lên mặt bằng 2D.

20/03/2025 – 20/4/2025: Tiếp tục hoàn thành Firmware và Software.

20/04/2025 – 01/05/2025: Đo đạc thực nghiệm và hoàn thành báo cáo LVTN.

Xác nhận của Cán bộ hướng dẫn (Ký tên và ghi rõ họ tên)	TP. HCM, ngày tháng năm.....
Sinh viên (Ký tên và ghi rõ họ tên)	

DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số.....
ngày của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. Chủ tịch.
2. Thư ký.
3. Ủy viên.
4. Ủy viên.
5. Ủy viên.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	4
1. Tổng quan	4
2. Tình hình nghiên cứu trong và ngoài nước	4
2.1. Tình hình nghiên cứu trong nước.....	4
2.2. Tình hình nghiên cứu ngoài nước	5
3. Nhiệm vụ luận văn	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	7
1. Hệ thống định vị trong nhà Indoor Positioning System(IPS)	7
2. Phương pháp Angel of Arrival (AoA Method)	7
3. Module NINA-B4	9
3.1. Giới thiệu tổng quan về NINA B406	9
3.2. Đặc tính kỹ thuật chính	11
3.3. Lý do chọn NINA B406	20
4. C209 Tag	21
4.1. Tổng quan	21
4.2. Chi tiết.....	22
5. Anchor ANT-B10	25
5.1. Sơ đồ khối của ANT-B10	28
5.2. Các thành phần cơ bản của ANT-B10	28
5.3. Chức năng Angel of Arrival (AoA) của ANT-B10	30
5.4. Hướng của hệ trục tọa độ và định nghĩa các góc	32
6. Giao thức kết nối.....	33
6.1. Message Queuing Telemetry Transport (MQTT)	33
6.2. Universal Asynchronous Receiver/Transmitter (UART).....	35
7. Bộ lọc	37
7.1. Bộ lọc Kalman	37
7.2. Bộ lọc trung vị Median	40
CHƯƠNG 3: TỔNG QUAN HỆ THỐNG.....	43
1. Sơ đồ khối hệ thống	43
2. Phương pháp bố trí.....	44
2.1. Giới thiệu	44
2.2. Phương pháp bố trí Anchor	44

	2.3. Xác định vị trí Tag	46
L	CHƯƠNG 4: THỰC HIỆN ĐỀ TÀI	50
L	1. Thiết kế và thực hiện Hardware.....	50
	1.1. Giới thiệu phần mềm KICAD	50
	1.2. Thực hiện Hardware	51
K	2. Thiết kế và thực hiện Firmware	55
	2.1. Lưu đồ giải thuật.....	55
	2.2. Thực hiện Firmware.....	55
	2.3. Kết quả	61
K	3. Thiết kế và thực hiện Software	62
	3.1. Lưu đồ giải thuật.....	62
	3.2. Giải thích các hàm quan trọng	63
	3.3. Kết quả	78
	CHƯƠNG 5: KẾT QUẢ THỰC HIỆN.....	83
K	1. Không gian thực nghiệm.....	83
K	2. Phương pháp đo đạc.....	85
K	3. Kết quả	87
	3.1. Kết quả đo tọa độ X	87
	3.2 Kết quả đo tọa độ Y	88
K	4. Phân Tích Sai Số	88
	4.1. Công Thức Tính Sai Số.....	88
	4.2. Sai số tọa độ X	88
	4.3. Sai số tọa độ Y	89
	4.4. Tổng Hợp Sai Số Root Mean Square Error (RMSE).....	90
	5. Nhận Xét Về Bản Đồ Sai Số	91
	CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	93
K	1. Kết luận	93
	2. Hướng phát triển	94
	TÀI LIỆU THAM KHẢO	95
	PHỤ LỤC	96

Chu n b vài slide ph cō g i b h i th l yra(gi u bài nh MU gi u bài)

DANH MỤC HÌNH ẢNH

HINH 1. MÔ TẢ PHƯƠNG PHÁP AOA	7
HINH 2. TÍN HIỆU TỐI MẨNG ANTEN	8
HINH 3. SƠ ĐỒ KHỐI NINA B406	11
HINH 4. SƠ ĐỒ GÁN CHÂN NINA B406	13
HINH 5. KÍCH THƯỚC VẬT LÝ CỦA NINA B406.....	18
HINH 6. CÁC CỤM CHỨC NĂNG CỦA C209 TAG	21
HINH 7. SCHEMATIC C209 TAG	22
HINH 8. CỤM BOOTLOADER.....	23
HINH 9. CỤM CẢM BIẾN GIA TỐC	23
HINH 10. CỤM CẢM BIẾN ÁNH SÁNG.....	24
HINH 11. CỤM CẢM BIẾN ĐỘ ÂM, NHIỆT ĐỘ VÀ ÁP SUẤT	24
HINH 12. CỤM BỘ NHỚ FLASH.....	24
HINH 13. CỤM CHỌN NGUỒN CẤP	25
HINH 14. CỤM ĐÈN TRẠNG THÁI	25
HINH 15. CỤM SWITCH.....	25
HINH 16. ANT-B10 BOARD	26
HINH 17. MÔ TẢ CÁC GÓC TRONG KHÔNG GIAN 3 CHIỀU	27
HINH 18. SƠ ĐỒ KHỐI CỦA ANT-B10.....	28
HINH 19. MÔ TẢ CÁC MẨNG ĂN-TEN CỦA ANT-B10.....	29
HINH 20. MẶT SAU CỦA ANT-B10	29
HINH 21. MÔ TẢ CÁCH BỐ TRÍ CÁC ANCHOR	30
HINH 22. KẾT QUẢ NHẬN ĐƯỢC TỪ ANCHOR	31
HINH 23. MÔ TẢ HỆ TRỰC TỌA ĐỘ ANCHOR	32
HINH 24. GIAO THÚC MQTT	33
HINH 25. GIAO THÚC TRUYỀN THÔNG BÁT ĐỒNG BỘ UART	36
HINH 26. CẤU TRÚC KHUNG DỮ LIỆU UART	36
HINH 27. SƠ ĐỒ KHỐI BỘ LỌC KALMAN	38
HINH 28. SƠ ĐỒ KHỐI HỆ THỐNG.....	43
HINH 29. MÔ TẢ HOẠT ĐỘNG HỆ THỐNG	44
HINH 30. BỐ TRÍ ANCHOR NGẪU NHIÊN	45
HINH 31. BỐ TRÍ 2 ANCHOR TRÊN ĐƯỜNG THẲNG SONG SONG VỚI OX	45
HINH 32. BỐ TRÍ 2 ANCHOR NẰM TRÊN OX	46
HINH 33. TAG Ở VÙNG PHẦN BA THỨ NHẤT	47
HINH 34. TAG Ở VÙNG PHẦN BA THỨ HAI.....	48
HINH 35. TAG Ở VÙNG PHẦN BA THỨ BA	49
HINH 36. KHỐI XỬ LÝ TRUNG TÂM.....	51
HINH 37. KHỐI CẤP NGUỒN VÀ CHUYỂN ĐỔI UART	51
HINH 38. KHỐI CÁC CẢM BIẾN	52
HINH 39. KHỐI IC NHỚ (FLASH).....	52
HINH 40. KHỐI ĐÈN TRẠNG THÁI	52
HINH 41. KHỐI NÚT NHẤN RESET VÀ ĐIỀU KHIỂN	53
HINH 42. KHỐI CÁC CHÂN MỎ RỘNG	53

HINH 43. CẤU TRÚC MẶT TRƯỚC	53
HINH 44. CẤU TRÚC MẶT SAU.....	53
HINH 45. TỔNG HỢP CẤU TRÚC ĐA LỚP	54
HINH 46. BIỂU DIỄN KHÔNG GIAN MẶT TRƯỚC	54
HINH 47. BIỂU DIỄN KHÔNG GIAN MẶT SAU.....	54
HINH 48. SẢN PHẨM THỰC TẾ	54
HINH 49. LUU ĐỒ GIẢI THUẬT VI ĐIỀU KHIỂN.....	55
HINH 50. GITHUB CỦA NHÀ SẢN XUẤT.....	56
HINH 51. HƯỚNG DẪN NẠP FIRMWARE CHO TAG	57
HINH 52. KẾT QUẢ THỰC TẾ THU ĐƯỢC TỪ PHẦN MỀM U-LOCATED	61
HINH 53. LUU ĐỒ GIẢI THUẬT CỦA SOFTWARE	62
HINH 54. GIAO DIỆN KHỞI ĐỘNG ỨNG DỤNG	78
HINH 55. KẾT NỐI THÀNH CÔNG TỚI MQTT BROKER.....	79
HINH 56. THEO DÕI DỮ LIỆU Ở COMMAND PROMPT.	80
HINH 57. LUU TRỮ DỮ LIỆU	82
HINH 58. BẢN VẼ MẶT BẰNG KÍ TỨC XÁ	83
HINH 59. BẢN VẼ MẶT BẰNG PHÒNG TỰ HỌC	83
HINH 60. GÁN HỆ TRỰC CHO KHÔNG GIAN THỰC NGHIỆM	84
HINH 61. TỌA ĐỘ ĐO ĐƯỢC CỦA TAG	86
HINH 62. TỌA ĐỘ X THỰC TẾ CỦA TAG	86
HINH 63. TỌA ĐỘ Y THỰC TẾ CỦA TAG	87

DANH MỤC BẢNG

BẢNG 1. THÔNG TIN VỀ MODULE NINA B406	11
BẢNG 2. ĐIỆN ÁP ĐẦU VÀO.....	12
BẢNG 3. DÒNG TIÊU THỤ Ở MODULE VCC	12
BẢNG 4. ĐIỆN ÁP HOẠT ĐỘNG CÁC CHÂN I/O	15
BẢNG 5. GIÁ TRỊ ĐIỆN TRỞ KÉO LÊN Ở CÁC CHÂN SDA VÀ SCL.....	17
BẢNG 6. KÍCH THƯỚC VẬT LÝ CỦA NINA B406	20
BẢNG 7. MỘT SỐ GPIO QUAN TRỌNG CỦA NINA-B406	22
BẢNG 8. KẾT QUẢ ĐO TỌA ĐỘ X	87
BẢNG 9. KẾT QUẢ ĐO TỌA ĐỘ Y	88
BẢNG 10. SAI SỐ TỌA ĐỘ X.....	89
BẢNG 11. SAI SỐ TỌA ĐỘ Y	90
BẢNG 12. TỔNG HỢP SAI SỐ	91

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ thay thế
AoA	Angle of Arrival
BLE	Bluetooth Low Energy
IPS	Indoor Positioning System
MQTT	Message Queuing Telemetry Transport
UART	Universal Asynchronous Receiver/Transmitter

TÓM TẮT LUẬN VĂN BẰNG TIẾNG VIỆT

Luận văn này trình bày việc thiết kế, triển khai và đánh giá Hệ thống Định vị Trong nhà (IPS) dựa trên công nghệ Bluetooth Low Energy (BLE) sử dụng phương pháp Góc Tới (Angle-of-Arrival - AoA). Trong những môi trường mà tín hiệu GPS không đủ hoặc không khả dụng, như bên trong các tòa nhà, các giải pháp IPS cung cấp dịch vụ định vị quan trọng với độ chính xác cao. Nghiên cứu tập trung tận dụng thông tin hướng từ tín hiệu BLE để xác định vị trí của các Tag di động thông qua phương pháp AoA.

Hệ thống tích hợp Anchor u-blox ANT-B10 với Tag C209 dựa trên module NINA-B406, tạo nên cơ sở hạ tầng toàn diện cho việc định vị thời gian thực. Chúng em đã thiết kế và triển khai các thành phần phần cứng, phần mềm nhúng và phần mềm tùy chỉnh để thiết lập giải pháp hoàn chỉnh từ đầu đến cuối. Quá trình phát triển phần cứng bao gồm thiết kế PCB sử dụng KiCad, trong khi việc triển khai phần mềm nhúng tích hợp các thuật toán xử lý tín hiệu để nâng cao độ chính xác. Ứng dụng phần mềm đã phát triển cung cấp khả năng hiển thị thời gian thực, ghi nhật ký dữ liệu và các chức năng cảnh báo thông qua giao thức truyền thông MQTT.

Kết quả thực nghiệm được tiến hành trong một môi trường trong nhà có kiểm soát cho thấy độ chính xác định vị trung bình đạt mức sai số 10-15% trong các vùng tối ưu, với độ chính xác giảm dần ở các khu vực ngoại vi. Phân tích của chúng em xác định các yếu tố chính ảnh hưởng đến độ chính xác hệ thống, bao gồm cấu hình Anchor, độ nhạy góc và nhiều môi trường.

Nghiên cứu này đóng góp cho lĩnh vực bằng cách cung cấp những hiểu biết sâu sắc về các chiến lược bố trí Anchor tối ưu, kỹ thuật xử lý tín hiệu và phương pháp triển khai phần mềm cho các hệ thống định vị dựa trên BLE. Giải pháp đã phát triển tạo nền tảng cho nhiều ứng dụng khác nhau bao gồm theo dõi tài sản, phân tích bán lẻ, giám sát y tế và dẫn đường trong nhà. Các cải tiến trong tương lai có thể tích hợp thêm các Anchor để định vị ba chiều và các thuật toán lai kết hợp AoA với các phương pháp định vị khác để nâng cao độ chính xác hơn nữa.

ABSTRACT

This thesis presents the design, implementation, and evaluation of an Indoor Positioning System (IPS) based on Bluetooth Low Energy (BLE) technology utilizing the Angle-of-Arrival (AoA) method. In environments where GPS signals are inadequate or unavailable, such as inside buildings, IPS solutions provide essential positioning services with high precision. The research focuses on leveraging directional information from BLE signals to determine the location of mobile tags through the AoA approach.

The system integrates u-blox ANT-B10 anchors with C209 tags based on NINA-B406 modules, creating a comprehensive infrastructure for real-time positioning. We designed and implemented custom hardware, firmware, and software components to establish an end-to-end solution. The hardware development involved PCB design using KiCad, while the firmware implementation incorporated signal processing algorithms to enhance accuracy. The developed software application provides real-time visualization, data logging, and alert functionalities through the MQTT communication protocol.

Experimental results conducted in a controlled indoor environment demonstrated an average positioning accuracy with error rates of 10-15% in optimal zones, with accuracy gradually declining in peripheral areas. Our analysis identified key factors affecting system accuracy, including anchor configuration, angular sensitivity, and environmental interference.

This research contributes to the field by providing insights into optimal anchor placement strategies, signal processing techniques, and software implementation approaches for BLE-based positioning systems. The developed solution creates a foundation for various applications including asset tracking, retail analytics, healthcare monitoring, and indoor navigation. Future improvements could incorporate additional anchors for three-dimensional positioning and hybrid algorithms combining AoA with other positioning methods to further enhance accuracy.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1. Tổng quan

Trong bối cảnh các công nghệ thông minh ngày càng phát triển, định vị trong nhà (Indoor Positioning) đã trở thành một lĩnh vực nghiên cứu quan trọng, với nhiều ứng dụng thực tiễn trong các ngành như bán lẻ, quản lý tài sản, y tế, và hệ thống dẫn đường trong các tòa nhà lớn. Không giống như hệ thống GPS hoạt động hiệu quả ngoài trời, các môi trường trong nhà thường gặp phải những thách thức lớn như tín hiệu bị cản trở hoặc phản xạ bởi tường, trần, và các vật liệu xây dựng. Điều này đặt ra yêu cầu cần có các phương pháp định vị mới có độ chính xác cao, linh hoạt và phù hợp với đặc thù của môi trường trong nhà.

Phương pháp định vị dựa trên góc đến (Angle of Arrival - AoA) là một trong những giải pháp tiềm năng nhờ khả năng sử dụng thông tin góc để xác định vị trí của đối tượng trong không gian. Kỹ thuật này tận dụng các đặc điểm của sóng vô tuyến hoặc tín hiệu ánh sáng, kết hợp với hệ thống ăng-ten đa hướng hoặc cảm biến quang học để đo lường góc đến của tín hiệu phát ra từ thiết bị cần định vị. So với các phương pháp định vị khác như Time of Arrival (ToA) hoặc Received Signal Strength (RSS), AoA không chỉ đạt được độ chính xác cao hơn mà còn yêu cầu ít thiết bị hơn trong một số ứng dụng, qua đó giảm chi phí triển khai.

2. Tình hình nghiên cứu trong và ngoài nước

2.1. Tình hình nghiên cứu trong nước

Công nghệ định vị Angle of Arrival (AoA) của u-blox đã bắt đầu nhận được sự quan tâm nghiên cứu tại Việt Nam trong những năm gần đây, mặc dù vẫn còn khá mới so với các công nghệ định vị truyền thống. Dưới đây là tổng quan về tình hình nghiên cứu và ứng dụng công nghệ này tại Việt Nam:

Một trong những nghiên cứu tiêu biểu về định vị trong nhà tại Việt Nam là báo cáo tiểu luận "Định vị trong nhà sử dụng công nghệ Bluetooth Low Energy" ¹ do Võ Minh Phụng thực hiện dưới sự hướng dẫn của TS. Trịnh Xuân Dũng tại Đại học Bách Khoa TP.HCM. Nghiên cứu này đã tiến hành khảo sát toàn diện các phương pháp định vị trong nhà, trong đó có phân tích chi tiết về công nghệ Angle of Arrival (AoA). Tác giả nhấn mạnh rằng trong môi trường thực tế với nhiều vật cản, việc kết hợp nhiều phương pháp định vị có thể mang lại kết quả tốt hơn so với sử dụng một phương pháp đơn lẻ.

Tạp chí Khoa học và Công nghệ của Đại học Bách Khoa Hà Nội là một trong những kênh xuất bản uy tín về các nghiên cứu kỹ thuật tại Việt Nam. Theo thông tin từ website chính thức, tạp chí đã xuất bản hơn 2.000 bài viết trong hơn 20 năm hoạt động, bao gồm

¹ Võ Minh Phụng (21/03/2017), "Định vị trong nhà sử dụng công nghệ Bluetooth Low Energy". Truy cập từ: <https://tailieu.vn/doc/bao-cao-tieu-luan-dinh-vi-trong-nha-su-dung-cong-nghe-bluetooth-low-energy-1941844.html>

nhiều nghiên cứu về công nghệ định vị và IoT. Tạp chí đặc biệt chú trọng đến các nghiên cứu về "Kỹ thuật và Công nghệ cho phát triển bền vững" và "Hệ thống và Thiết bị Thông minh" - hai lĩnh vực có liên quan mật thiết đến công nghệ định vị AoA. Kể từ năm 2021, các bài báo được xuất bản trên tạp chí đều được đăng ký với CrossRef DOI, đảm bảo tính học thuật và khả năng truy xuất quốc tế¹.

Mang trong mình nhiều tiềm năng trong quản lý logistics, hệ thống định vị trong nhà,... nhưng việc triển khai và ứng dụng vẫn gặp nhiều thách thức:

- Chi phí đầu tư cho bộ kit và thiết bị thử nghiệm còn cao đối với nhiều đơn vị nghiên cứu.
- Nguồn nhân lực chuyên sâu về công nghệ định vị chính xác còn hạn chế.
- Kiến thức về hiệu chuẩn và tối ưu hóa hệ thống AoA trong điều kiện môi trường Việt Nam còn chưa đầy đủ.
- Sự phối hợp giữa nghiên cứu học thuật và ứng dụng thực tế trong doanh nghiệp còn chưa chặt chẽ.

2.2. Tình hình nghiên cứu ngoài nước

Công nghệ định vị AoA của u-blox đang được nghiên cứu và ứng dụng rộng rãi trên toàn cầu, với những tiến bộ đáng kể trong cả lĩnh vực học thuật và thương mại. Sự kết hợp của công nghệ này với các kỹ thuật học máy, IoT, và các công nghệ định vị khác đang mở ra nhiều triển vọng mới cho các ứng dụng định vị trong nhà chính xác cao. Một số ứng dụng nổi bật có thể kể đến như:

Nghiên cứu "Indoor Multipath Assisted Angle of Arrival Localization" đề xuất một phương pháp độc đáo khai thác hiện tượng đa đường truyền (multipath) - vốn thường được xem là trở ngại - để cải thiện độ chính xác định vị AoA. Thay vì cố gắng loại bỏ nhiều đa đường, các tác giả đã phát triển thuật toán sử dụng chính thông tin từ các phản xạ này để tăng cường khả năng định vị.²

Bài báo "Indoor localization algorithms based on Angle of Arrival with a benchmark comparison" cung cấp một đánh giá toàn diện về các thuật toán định vị AoA sử dụng công nghệ Bluetooth 5.1, đồng thời đề xuất các cải tiến dựa trên khái niệm "Vùng tin cậy" (Confidence Region).³

¹ Đại học Bách Khoa Hà Nội, "Tạp chí Khoa học và Công nghệ". Truy cập từ: <https://hust.edu.vn/vi/nghien-cuu/tap-chi-khoa-hoc-cong-nghe/jst-650113.html>

² Stijn Wielandt, Lieven De Strycker (02/11/2017), "Indoor Multipath Assisted Angle of Arrival Localization". Truy cập từ: *Indoor Multipath Assisted Angle of Arrival Localization*

³ José Luis Lázaro, Alfredo Gardel, Alvaro De-La-Llana-Calvo (18/09/2020), "Accuracy and Precision Assessment of AoA-Based Indoor Positioning Systems Using Infrastructure Lighting and a Position-Sensitive Detector". Truy cập từ: *Accuracy and Precision Assessment of AoA-Based Indoor Positioning Systems Using Infrastructure Lighting and a Position-Sensitive Detector - PMC*

3. Nhiệm vụ luận văn

Đề tài này nhằm mục đích nghiên cứu sâu về nguyên lý hoạt động của phương pháp AoA, bao gồm cơ sở lý thuyết, các thuật toán xử lý tín hiệu, và các yếu tố ảnh hưởng đến độ chính xác định vị. Ngoài ra, đề tài sẽ thực hiện xây dựng và kiểm nghiệm một mô hình thực nghiệm để đánh giá hiệu suất của phương pháp AoA trong các kịch bản thực tế như nhà ở, trung tâm thương mại hoặc nhà kho. Thông qua đó, nghiên cứu sẽ xác định các ưu điểm, hạn chế và đề xuất các giải pháp nhằm tối ưu hóa hiệu quả của phương pháp AoA. Nhiệm vụ đồ án bao gồm các mục tiêu như sau:

Nhiệm vụ 1: Cơ sở lý thuyết

Nhiệm vụ 2: Thiết kế và thực hiện Hardware với schematic tham khảo từ u-blox

Nhiệm vụ 3: Thực hiện Firmware

Nhiệm vụ 4: Thiết kế và thực hiện Software

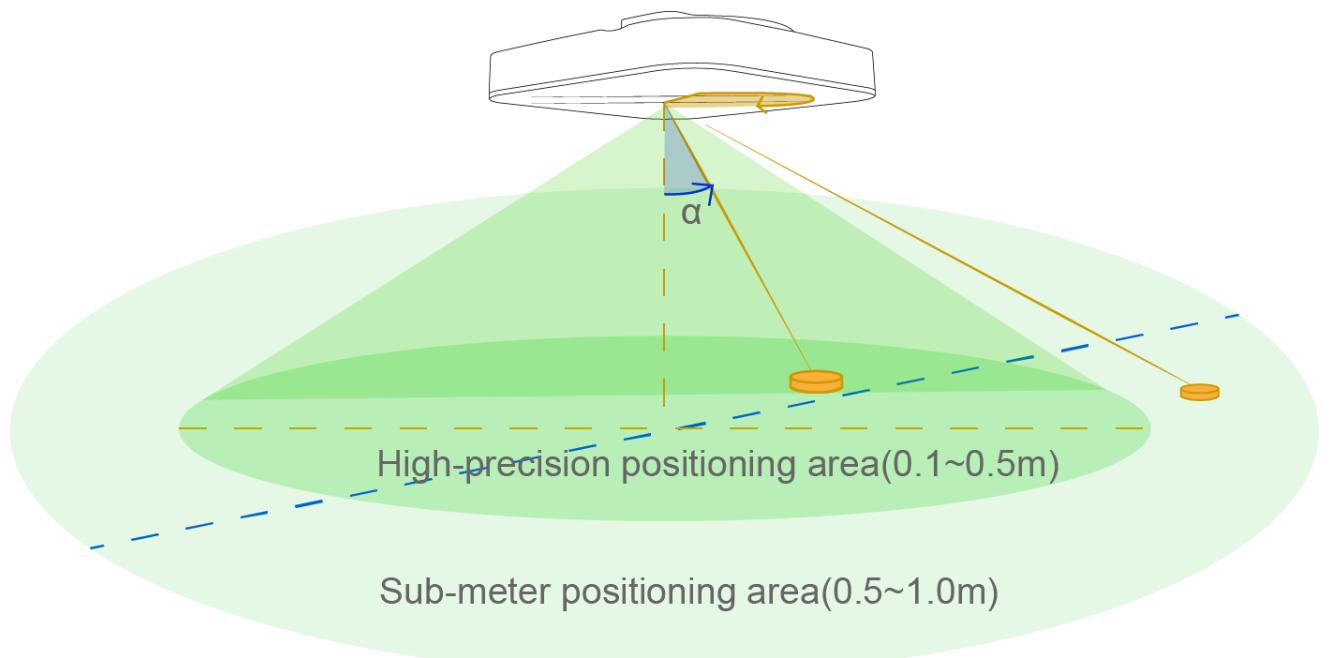
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. Hệ thống định vị trong nhà Indoor Positioning System (IPS)

Hệ thống định vị trong nhà là một mạng lưới các thiết bị và công nghệ được sử dụng để xác định vị trí người hoặc vật thể mà GPS hoạt động thiếu chính xác hoặc bị lỗi hoàn toàn như bên trong các tòa nhà nhiều tầng, sân bay, ngõ, nhà để xe và các vị trí dưới lòng đất. IPS chính là công nghệ định vị hỗ trợ cho GPS tại những khu vực nhỏ hơn, phức tạp hơn nhưng đòi hỏi độ chính xác cao hơn. Có rất nhiều kỹ thuật và thiết bị và công nghệ được sử dụng để cung cấp định vị trong nhà, từ các thiết bị phổ thông có sẵn như điện thoại thông minh với ăngten WiFi và Bluetooth, camera, máy ảnh kỹ thuật số dân dụng cho tới các hệ thống và công nghệ chuyên dụng hơn như cảm biến, la bàn, đèn, sóng vô tuyến, từ trường, tín hiệu âm thanh và phân tích hành vi đều được sử dụng trong mạng IPS.

2. Phương pháp Angel of Arrival (AoA Method)

Angle of Arrival (AoA) là một tính năng trong công nghệ Bluetooth cho phép xác định hướng tín hiệu từ nguồn phát. Tính năng này nâng cao khả năng dịch vụ dựa trên vị trí, cho phép các thiết bị xác định vị trí của chúng so với nguồn tín hiệu, điều này đặc biệt hữu ích trong các ứng dụng như điều hướng, định vị trong nhà.



Hình 1. Mô tả phương pháp AoA¹

Các khái niệm chính về AoA:

¹ Dusun (18/02/2025), “AoA Locator For Indoor RTLS Personnel Tracking and Asset Positioning”. Truy cập từ: [AoA Locator For Indoor RTLS Personnel Tracking and Asset Positioning](#)

- Nguyên lý cơ bản:

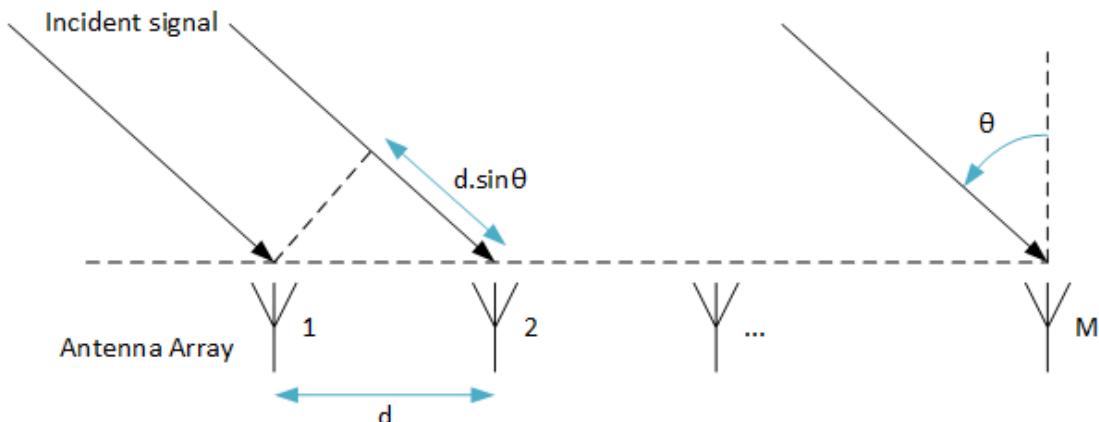
AoA dựa trên việc đo độ chênh lệch pha của tín hiệu nhận được từ nhiều ăng-ten được đặt theo một cách cụ thể. Bằng cách phân tích những chênh lệch này, hệ thống có thể tính toán hướng tín hiệu đến.

- Mảng ăng-ten (Antenna Array):

Thường sử dụng một mảng ăng-ten (thường là hai hoặc nhiều hơn) để tiếp nhận tín hiệu. Cách bố trí không gian của những ăng-ten này là rất quan trọng cho việc xác định độ chính xác.

- Đo lường độ chênh lệch pha:

Hệ thống đo thời gian trễ hoặc độ chênh lệch pha của các tín hiệu nhận được tại mỗi ăng-ten. Bằng cách xử lý thông tin này, hệ thống có thể xác định góc đến của tín hiệu.



Hình 2. Tín hiệu tới mảng Anten¹

- Trường hợp sử dụng:

AoA có thể được sử dụng trong nhiều ứng dụng, chẳng hạn như:

- Hệ thống định vị trong nhà kho để xác định vị trí của thiết bị hoặc tài sản.
- Ứng dụng thực tế ảo nơi cần biết hướng của thiết bị so với các đối tượng ảo.
- Thiết bị thông minh trong nhà có thể phản ứng theo vị trí của người dùng.

- Phiên bản Bluetooth:

Chức năng AoA là một phần của công nghệ Bluetooth 5.1, đã giới thiệu những cải tiến cho việc tìm hướng bên cạnh các tính năng tiêu chuẩn của Bluetooth Low Energy.

¹ Wikipedia, “Wi-Fi positioning system”. Truy cập từ: Wi-Fi positioning system - Wikipedia

- Lợi ích của AoA:

-Độ chính xác cao: Cung cấp dữ liệu vị trí chính xác hơn so với các phương pháp định vị khác.

-Tiêu thụ điện năng thấp: Tương thích với Bluetooth Low Energy, phù hợp cho các thiết bị chạy bằng pin.

-Khả năng mở rộng: Có thể được triển khai trong nhiều môi trường, từ các phòng nhỏ đến các khu vực lớn.

3. Module NINA-B4

3.1. Giới thiệu tổng quan về NINA B406

3.1.1. Đặc điểm chính và ứng dụng

Module **NINA B406** thuộc họ NINA B40, gồm module không dây Bluetooth 5.1 Low Energy nhỏ, độc lập, có công suất đầu ra +8 dBm, thích hợp cho môi trường hoạt động khắc nghiệt khi hoạt động tốt trong tầm nhiệt độ (-40°C;105°C).

NINA B406 nói chung và họ NINA hỗ trợ chức năng Bluetooth 5.1 Direction Finding thông qua 2 phương pháp Angle of Arrival (AoA) và Angle of Departure (AoD). Những tính năng này vô cùng hữu ích trong Indoor Positioning, wayfinding, và asset tracking.

NINA B406 chỉ cần nguồn cấp từ 1.7V đến 3.6V, có thể dùng mức điện áp này làm điện áp tham chiếu cho các chân Input/Output nên vi xử lý này có thể dễ dàng tích hợp vào các hệ thống chỉ hoạt động với một mức điện áp duy nhất, giúp đơn giản hóa thiết kế.

Ứng dụng của NINA B406:

- **Industrial Automation (Tự động hóa công nghiệp):** Sử dụng các hệ thống tự động để điều khiển và vận hành các quy trình sản xuất, giúp tăng hiệu suất và giảm chi phí lao động
- **Smart Buildings and Cities (Các tòa nhà và thành phố thông minh):** Sử dụng công nghệ IoT (Internet of Things) để thu thập và phân tích dữ liệu trong thời gian thực, nhằm cải thiện hiệu quả năng lượng, an ninh và chất lượng cuộc sống.
- **Low Power Sensors (Cảm biến tiêu thụ năng lượng thấp):** Các cảm biến tiêu thụ ít năng lượng, thường được sử dụng trong các thiết bị IoT để theo dõi môi trường xung quanh mà không cần thay pin thường xuyên.
- **Wireless-Connected and Configurable Equipment (Thiết bị kết nối không dây và có thể cấu hình):** Các thiết bị có khả năng kết nối mạng không dây và có thể được cấu hình từ xa, giúp dễ dàng quản lý và kiểm soát.

- **Point-of-Sales (POS - Điểm bán hàng):** Hệ thống POS được sử dụng trong các cửa hàng bán lẻ để xử lý các giao dịch, quản lý tồn kho và thu thập dữ liệu khách hàng.
- **Health Devices (Thiết bị y tế):** Các thiết bị y tế thông minh giúp theo dõi và phân tích các chỉ số sức khỏe của người dùng, hỗ trợ việc chăm sóc sức khỏe cá nhân.
- **Real-time Location Systems (RTLS - Hệ thống định vị thời gian thực):** Hệ thống theo dõi vị trí của các đối tượng trong thời gian thực, thường được sử dụng trong quản lý kho bãi và logistic.
- **Indoor Positioning (Định vị trong nhà):** Công nghệ định vị các đối tượng bên trong các tòa nhà, giúp cải thiện quản lý tài sản và cung cấp các dịch vụ dựa trên vị trí.
- **Asset Tracking (Theo dõi tài sản):** Theo dõi và quản lý tài sản của doanh nghiệp thông qua các công nghệ như RFID, GPS, và cảm biến, giúp cải thiện hiệu quả và giảm mất mát tài sản.

3.1.2. Vị trí của NINA B406 trong họ sản phẩm NINA

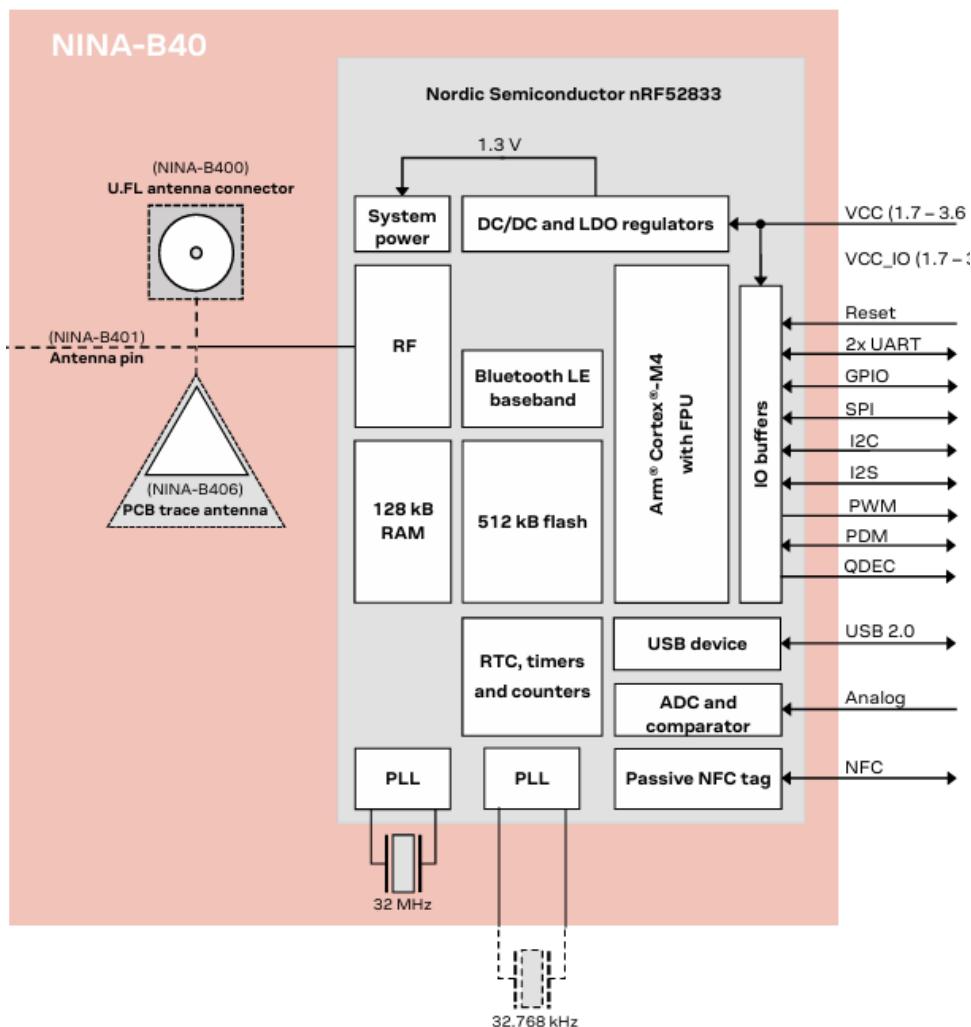
	NINA B400	NINA B401	NINA B406
Phiên bản Bluetooth	5.1		
Băng tần	2.4 GHz		
Công suất đầu ra	+8 dBm		
Độ nhạy RX (đo lường)	-95 dBm		
Độ nhạy RX ở chế độ tầm xa (đo lường)	-102 dBm		
Tốc độ truyền nhận dữ liệu Bluetooth Low Energy	1 Mbps 2 Mbps 500 kbps 125 kbps		
Kích thước Module	10.0 x 15.0 x 2.2 mm	10.0 x 11.6 x 2.2 mm	10.0 x 15.0 x 2.2 mm
Antenna	U.FL Connector	Antenna Pin (External)	Internal PCB Trace Antenna

Bảng 1. Thông tin về module NINA B406

NINA B406 không có chân RF để sử dụng với Antenna bên ngoài, thay vào đó lại tích hợp Antenna vào trong PCB.

3.2. Đặc tính kỹ thuật chính

3.2.1. Sơ đồ khói



Hình 3. Sơ đồ khói NINA B406¹

NINA B406 có Nordic Semiconductor nRF52833 với lõi xử lý Arm® Cortex-M4® với FPU, bộ nhớ flash 512 kB và RAM 128 kB.

3.2.1.1. Nguồn cấp

Module NINA B406 có tích hợp mạch hạ áp để chuyển đổi từ bên ngoài(1.7 - 3.6V) ở chân VCC thành điện áp ổn định cung cấp cho module (1.3V).

¹ Ublox (01/07/2024), “NINA-B40 series Stand-alone Bluetooth 5.1 low energy modules Data sheet”. Truy cập từ: [NINA-B40 series \(open CPU\) | u-blox](#)

Có 2 bộ chuyển đổi:

- Low dropout (LDO)
- DC/DC Buck

Module chuyển đổi linh hoạt giữa 2 bộ này để phù hợp với mức tiêu thụ điện. Khi Radio hoạt động, bộ DC/DC Buck sẽ hoạt động, trong khi bộ LDO hiệu quả hơn ở chế độ tiết kiệm năng lượng.

Tương tự với VCC, ở các chân I/O cũng hoạt động với mức điện áp $VCC_IO = VCC$

Kí hiệu	Thông số	Min	Typ	Max	Đơn vị
VCC	Điện áp đầu vào nguồn cấp	1.7	3.3	3.6	V
VCC_IO	Điện áp I/O tham chiếu		VCC		V

Bảng 2. Điện áp đầu vào

Điều kiện	Dòng điện hìn
Hệ thống TẮT, không duy trì RAM	600 nA
Hệ thống TẮT, duy trì toàn bộ 128 kB RAM	1.3 μA
Hệ thống BẬT, duy trì toàn bộ 128 kB RAM Hệ thống chạy trên xung nhịp 32.768 kHz từ bộ dao động nội	2.6 μA
CPU chạy kiểm tra CoreMark @ 64 MHz từ bộ nhớ flash, DC/DC	3.3 mA
Radio chỉ nhận @ 1 Mbps chế độ Bluetooth LE	6.0 mA
Radio chỉ phát, công suất đầu ra 0 dBm	6.0 mA
Radio chỉ phát, công suất đầu ra +8 dBm	15.5 mA

Bảng 3. Dòng tiêu thụ ở module VCC

3.2.1.2. RF Atenna

Khác với NINA B400 và NINA B401, NINA B40 có một antenna được tích hợp vào bên trong module PCB. Giải pháp này giúp tiết kiệm không gian và dễ dàng tích hợp vào các hệ thống nhúng.

Ứng dụng trong Direction Finding (AoA/AoD):

Sử dụng Bluetooth 5.1, NINA B406 hỗ trợ chức năng Bluetooth Direction Finding dựa trên 2 phương pháp Angle of Arrival (AoA) và Angle of Deviation (AoD), hiệu quả trong Indoor Positioning, Wayfinding và Tracking

Bộ thu AoA và bộ phát AoD sử dụng các Antenna đđược bật tuần tự, có thuật toán xử lý dựa trên tam giác để phát hiện hướng của tín hiệu Bluetooth với độ chính xác cao.

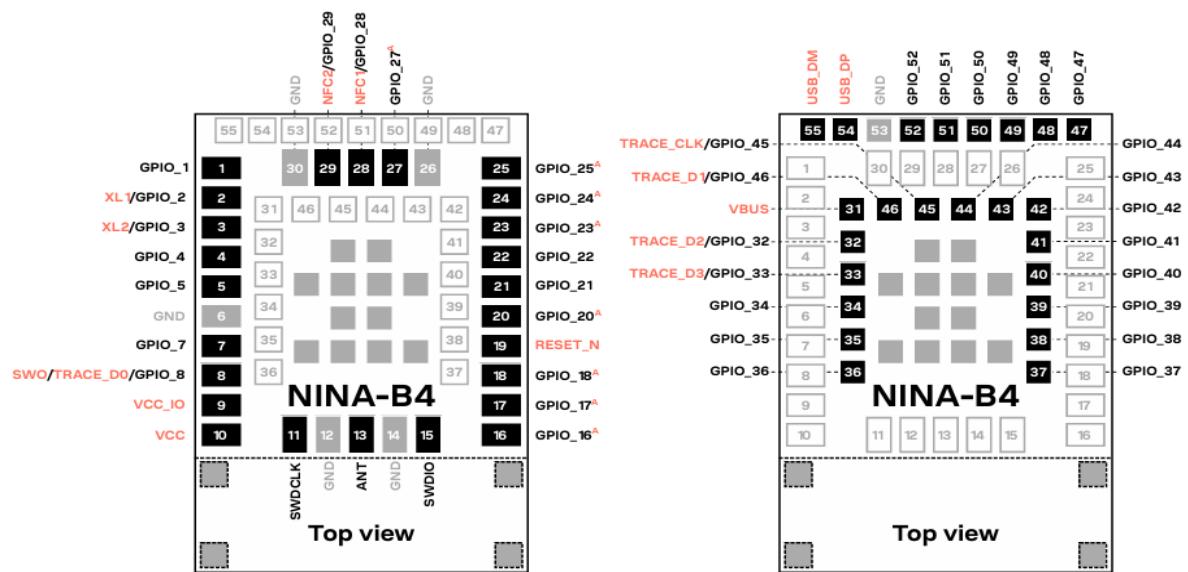
Chức năng khả dụng với Bluetooth Low Energy 1 Mbps và 2 Mbps.

3.2.1.3. Bộ dao động

NINA B406 có 2 bộ dao động: Bộ dao động tần số cao và bộ dao động tần số thấp. Bộ dao động tần số thấp được tích hợp trong module (bộ dao động RC), còn bộ dao động tần số cao sử dụng thạch anh 32.768 Hz.

Khuyến khích sử dụng thêm bộ dao động ngoại để hạn chế dòng điện tiêu thụ.

3.2.2. Các chân tín hiệu



1

Hình 4. Sơ đồ gán chân NINA B406

Lưu ý:

- Các chân màu xám là các chân GND.
- Vùng được khoanh nét đứt là vùng antenna, 4 chân ở 4 góc là các chân GND.
- Những chân được đánh màu đỏ là những chân Analog.

¹ Ublox (01/07/2024), “NINA-B40 series Stand-alone Bluetooth 5.1 low energy modules Data sheet”. Truy cập từ: [NINA-B40 series \(open CPU\) | u-blox](#)

Tên	Thông số	Nhỏ nhất	Điển hình	Lớn nhất	Đơn vị	Lưu ý
Tất cả chân Digital	Đầu vào mức thấp	0	0	0.3*VCC	V	
	Đầu vào mức cao	0.7*VCC	VCC	VCC	V	
	Đầu ra mức thấp	0		0.4	V	Độ mạnh truyền dẫn tiêu chuẩn và cao
	Đầu ra mức cao	VCC-0.4	VCC	VCC	V	Độ mạnh truyền dẫn tiêu chuẩn và cao
	Dòng điện Sink/Source	1	2	4	mA	Độ mạnh truyền dẫn tiêu chuẩn
		3			mA	Độ mạnh truyền dẫn cao, VCC<2.7V
		6	10	15	mA	Độ mạnh truyền dẫn cao, VCC>2.7V
		6	9	14	mA	Độ mạnh truyền dẫn cao, VCC>2.7V
	Thời gian cạnh lên/ cạnh xuống	9-25			ns	Độ mạnh truyền dẫn tiêu chuẩn, phụ thuộc điện dung tải
		4-8			ns	Độ mạnh truyền dẫn cao, phụ thuộc điện dung tải

	Điện trở kéo lên đầu vào	11	13	16	$k\Omega$	Có thể thêm vào bất kỳ chân GPIO nào được cấu hình làm đầu vào
	Điện trở kéo xuống đầu vào	11	13	16	$k\Omega$	Có thể thêm vào bất kỳ chân GPIO nào được cấu hình làm đầu vào
GPIO_28, GPIO_29	Dòng rò		1	10	uA	Không cấu hình NFC và có nhiều mức logic khác nhau
			1	15	uA	Không cấu hình NFC và có nhiều mức logic khác nhau, nhiệt độ > 85 °C

Bảng 4. Điện áp hoạt động các chân I/O

3.2.3. Các giao thức truyền thông được hỗ trợ

3.2.3.1. UART

Baudrate tối đa: 1 Mbps

Các chân giao tiếp:

- TXD (Transmit Data): Chân truyền dữ liệu từ NINA B406 đến thiết bị khác.
- RXD (Receive Data): Chân nhận dữ liệu từ thiết bị khác vào NINA B406.
- RTS (Request to Send): Chân yêu cầu gửi, được sử dụng để kiểm soát luồng dữ liệu.
- CTS (Clear to Send): Chân cho phép gửi, được sử dụng để kiểm soát luồng dữ liệu.

Khung truyền:

<Data>	<Parity>	<Stop>
8	1	1

Trong đó:

- Data Bit : 8 bit
- Parity Bit: 1 bit, sử dụng để kiểm tra lỗi trong quá trình truyền.
- Stop Bit: 1 bit, đánh dấu kết thúc của một khung dữ liệu.

Thứ tự bit: Bit LSB (Least Significant Bit) sẽ được truyền đi trước.

Với thiết lập này, NINA B406 có thể giao tiếp ổn định và hiệu quả trong các ứng dụng truyền dữ liệu, đảm bảo tính toàn vẹn và độ tin cậy của dữ liệu truyền và nhận. Các chế độ điều khiển luồng dữ liệu như RTS và CTS cũng giúp tránh các vấn đề về tắc nghẽn và mất dữ liệu trong quá trình truyền thông.

3.2.3.2. SPI

Cấu hình chân trong chế độ chính (Main node):

- SCLK (Serial Clock output): Đầu ra xung nhịp nối tiếp.
- MOSI (Main Output Sub Input): Dòng dữ liệu xuất chính, nhập phụ.
- MISO (Main Input Sub Output): Dòng dữ liệu nhập chính, xuất phụ.
- CS (Chip Select output): Đầu ra chọn chip, hoạt động ở mức thấp, chọn thiết bị phụ nào trên bus để giao tiếp. Mặc định chỉ có một đường chọn được kích hoạt, nhưng có thể thêm nhiều đường khác bằng cách chỉnh chân GPIO.
- DCX (Data/Command signal): Tín hiệu dữ liệu/lệnh, tín hiệu này là tùy chọn nhưng đôi khi được sử dụng bởi các thiết bị phụ SPI để phân biệt giữa lệnh SPI và dữ liệu.

Cấu hình chân trong chế độ phụ (Sub node):

- SCLK (Serial Clock input): Đầu vào xung nhịp nối tiếp.
- MOSI (Main Output Sub Input): Dòng dữ liệu xuất chính, nhập phụ.
- MISO (Main Input Sub Output): Dòng dữ liệu nhập chính, xuất phụ.
- CS (Chip Select input): Đầu vào chọn chip, hoạt động ở mức thấp, kết nối/ngắt kết nối giao diện phụ từ bus.

3.2.3.3. I2C

NINA B406 có thể làm Controller/Target với tốc độ truyền tiêu chuẩn 100 kbps hay tốc độ cao 400 kbps.

Cấu hình chân :

- SCL
- SDA

Điện trở kéo lên ở các chân SDA và SCL :

Kí hiệu	Tham số	Điện dung tải	Nhỏ nhất	Điện hình	Lớn nhất	Đơn vị
R_PUstandard	Ở chế độ tiêu chuẩn (100 kbps)	50 pF	1	-	23	kΩ
		200 pF	1	-	6	kΩ
		400 pF	1	-	5	kΩ
R_Pufast	Ở chế độ tốc độ cao (400 kbps)	50 pF	1	-	7	kΩ
		200 pF	1	-	1.75	kΩ
		400 pF	1	-	1	kΩ

Bảng 5. Giá trị điện trở kéo lên ở các chân SDA và SCL.

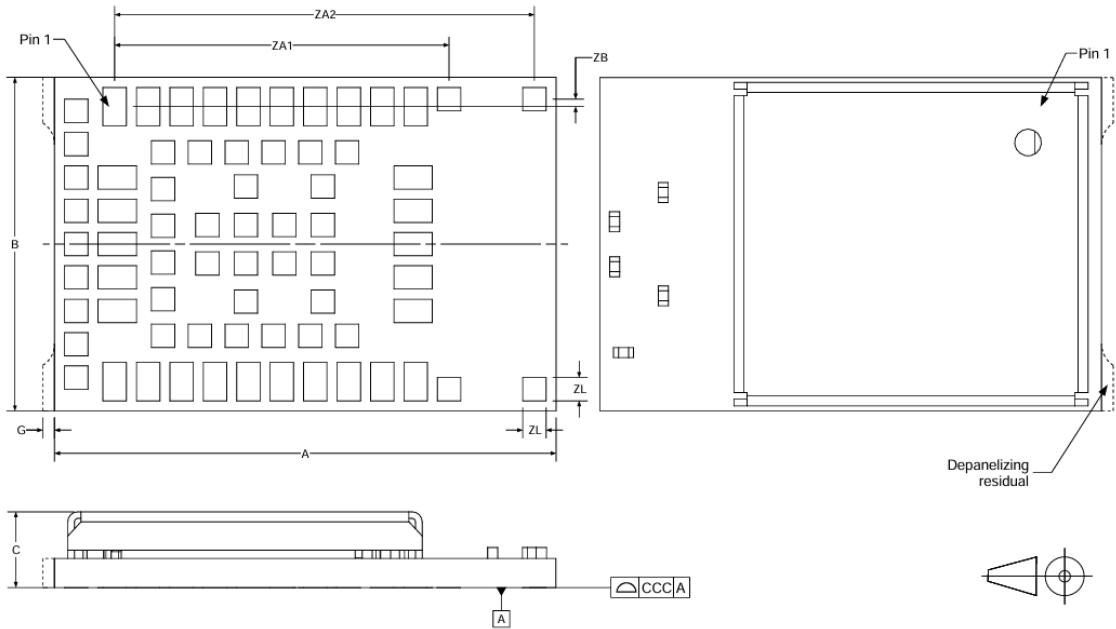
3.2.3.4. USB 2.0

Tốc độ truyền lên đến 12 Mbps

Cấu hình chân:

- VBUS, 5V nguồn cấp đầu vào.
- USB_DP, USB_DM.

3.2.4. Kích thước vật lý



Hình 5. Kích thước vật lý của NINA B406¹

Thông số	Mô tả	Điển hình [mm]	[mil]	Dung sai [mm]	[mil]
A	Chiều dài PCB Module	15.0	456.7	$\pm 0.20/-0.10$	+7.9/-3.9
B	Chiều rộng PCB Module	10.0	393.7	$\pm 0.20/-0.20$	± 7.9
C	Độ dày module	2.23	87.8	$\pm 0.40/-0.20$	+15.8/-7.9
ccc	Độ đồng phẳng mặt đặt	0.10	3.9	$\pm 0.02/-0.10$	+0.8/-3.9
D	Từ cạnh ngang đến tâm chân 1	1.80	70.9	± 0.10	± 3.9
E	Từ cạnh dọc đến tâm chân 1	0.875	34.5	± 0.10	± 3.9
F	Từ tâm chân 1 theo chiều dọc đến tâm chân bên	2.125	87.9	± 0.05	± 2.0
G	Phần dư sau khi tách bảng	0.10	3.9	$\pm 0.25/-0.1$	+9.8/-3.9

¹ Ublox (01/07/2024), ‘NINA-B40 series Stand-alone Bluetooth 5.1 low energy modules Data sheet’. Truy cập từ: [NINA-B40 series \(open CPU\) | u-blox](#)

H	Khoảng cách giữa các chân bên và chân ăng-ten	1.00	39.4	± 0.05	± 2.0
I	Chiều rộng chân bên, chân ăng-ten và chân ngoài	0.70	27.6	± 0.05	± 2.0
J	Chiều dài chân bên và chân ăng-ten	1.15	45.3	± 0.05	± 2.0
K	Từ tâm chân 1 theo chiều ngang đến tâm chân trung tâm	6.225	245.1	± 0.05	± 2.0
L	Từ tâm chân 1 theo chiều dọc đến tâm chân trung tâm	2.40	94.5	± 0.05	± 2.0
M	Từ tâm chân 1 theo chiều ngang đến tâm chân hàng trong	1.45	57.1	± 0.05	± 2.0
N	Từ tâm chân 1 theo chiều dọc đến tâm chân hàng trong	1.375	54.1	± 0.05	± 2.0
O	Chiều rộng và chiều dài chân trung tâm, hàng trong và hàng ngoài	0.70	27.6	± 0.05	± 2.0
P	Khoảng cách giữa các chân trung tâm	1.15	45.3	± 0.05	± 2.0
Q	Khoảng cách giữa các chân hàng trong	1.10	43.3	± 0.05	± 2.0
R	Từ tâm chân 1 theo chiều ngang đến tâm chân hàng ăng-ten	8.925	351.4	± 0.05	± 2.0
S	Khoảng cách giữa các chân hàng ngoài	1.00	39.4	± 0.05	± 2.0
T	Từ tâm chân 1 theo chiều dọc đến tâm chân hàng ngoài	0.125	4.9	± 0.05	± 2.0
U	Từ tâm chân 1 theo chiều ngang đến tâm chân hàng ngoài	1.15	45.3	± 0.05	± 2.0

Y	Từ tâm chân 1 theo chiều ngang đến tâm chân bên	0.075	3.0	± 0.05	± 2.0
ZA1	Từ tâm chân 1 theo chiều ngang đến tập chân GND ăng-ten thứ nhất	10.0	393.7	± 0.05	± 2.0
ZA2	Từ tâm chân 1 theo chiều ngang đến tập chân GND ăng-ten thứ hai	12.55	494.1	± 0.05	± 2.0
ZB	Từ tâm chân 1 theo chiều dọc đến tâm chân GND ăng-ten	0.225	8.9	± 0.05	± 2.0
ZL	Chiều rộng và chiều dài chân GND ăng-ten	0.70	27.6	± 0.05	± 2.0
	Khối lượng module [g]	<1.0			

Bảng 6. Kích thước vật lý của NINA B406

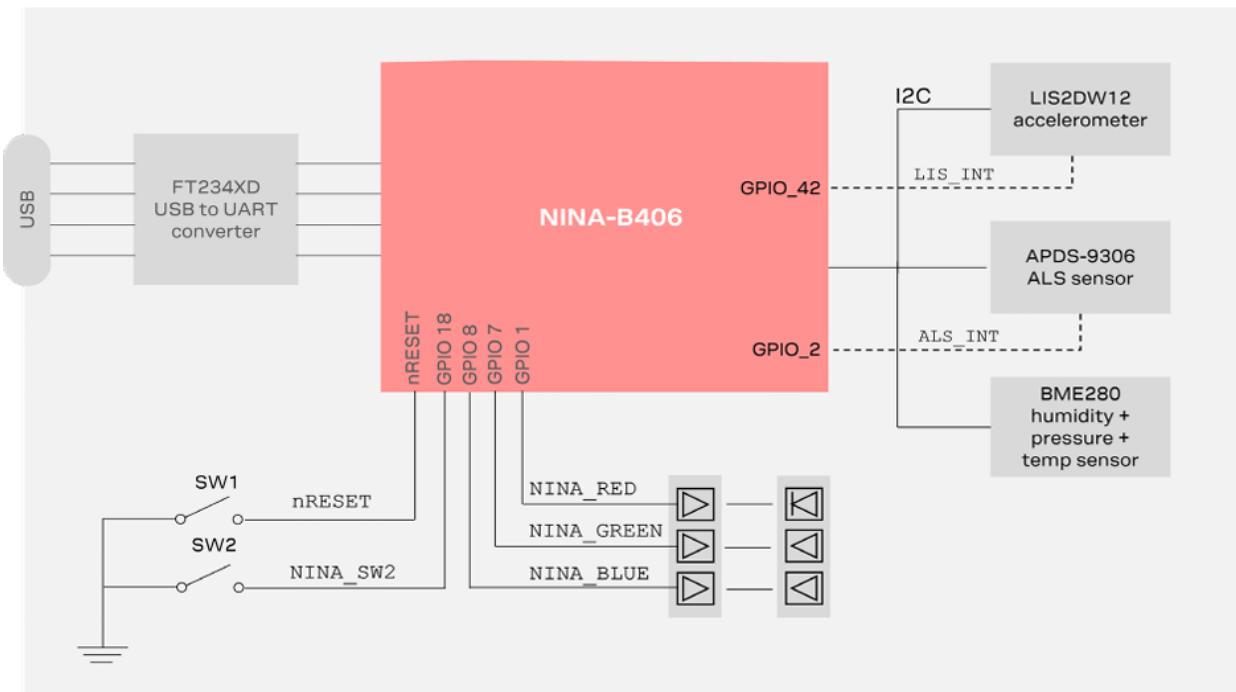
3.3. Lý do chọn NINA B406

NINA B406 không có chân RF để sử dụng với Antenna bên ngoài, thay vào đó lại tích hợp Antenna vào trong PCB. Một số ưu điểm được nhà sản xuất đưa ra:

- **Antenna nội bộ:** NINA-B406 được trang bị antenna PCB nội bộ, cung cấp giải pháp gọn nhẹ và hiệu suất cao. Điều này giúp giảm thiểu không gian và chi phí so với việc sử dụng antenna ngoài như NINA-B400 và NINA-B401.
- **Hiệu suất ổn định:** Antenna nội bộ của NINA-B406 mang lại hiệu suất ổn định và phạm vi rộng, phù hợp với các ứng dụng yêu cầu độ tin cậy cao.
- **Dễ dàng tích hợp:** Với antenna nội bộ, NINA-B406 dễ dàng tích hợp vào các thiết kế mạch mà không cần phải lo lắng về việc kết nối antenna ngoài

4. C209 Tag

4.1. Tổng quan



Hình 6. Các cụm chức năng của C209 Tag¹

C209 Tag là module mở rộng của CPU NINA-B406, được cấp nguồn pin CR2032 hoặc thông qua USB, C209 bao gồm các cụm cảm biến với các chức năng khác nhau:

- Cảm biến gia tốc LIS2DW12
- Cảm biến ánh sáng APDS-9306
- Cảm biến độ ẩm, áp suất và nhiệt độ BME280

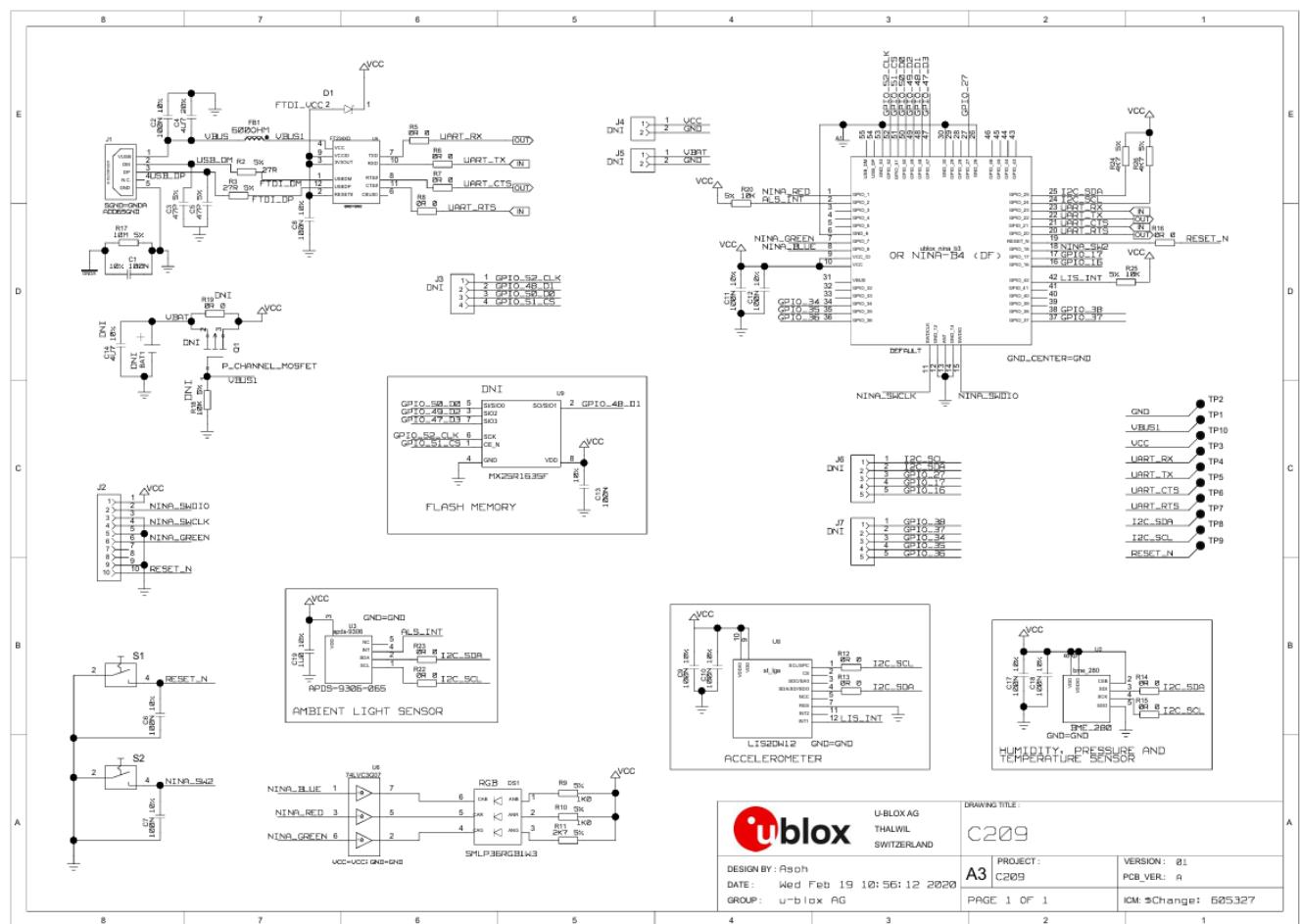
GPIO	Tên	I/O	Mô tả	Lưu ý
GPIO_1	RED	O	Đèn trạng thái	Tích cực thấp
GPIO_2	ALS_INT	I	Chân ngắt ngoài từ cảm biến ánh sáng	Có dòng rò
GPIO_7	GREEN	O	Đèn trạng thái	Tích cực thấp

¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”. Truy cập từ: [Product resources | u-blox](#)

GPIO_8	BLUE	O	Đèn trạng thái	Tích cực thấp
GPIO_18	SWITCH_2	I	Nút Switch_2	Tích cực thấp
GPIO_20	UART_RTS	O	Điều khiển gửi UART	
GPIO_21	UART_CTS	I	Xóa gửi UART	
GPIO_22	UART_TXD	O	Xuất dữ liệu UART	
GPIO_23	UART_RXD	I	Nhận dữ liệu UART	
GPIO_42	LIS_INT	I	Chân ngắt từ cảm biến gia tốc	

Bảng 7. Một số GPIO quan trọng của NINA-B406

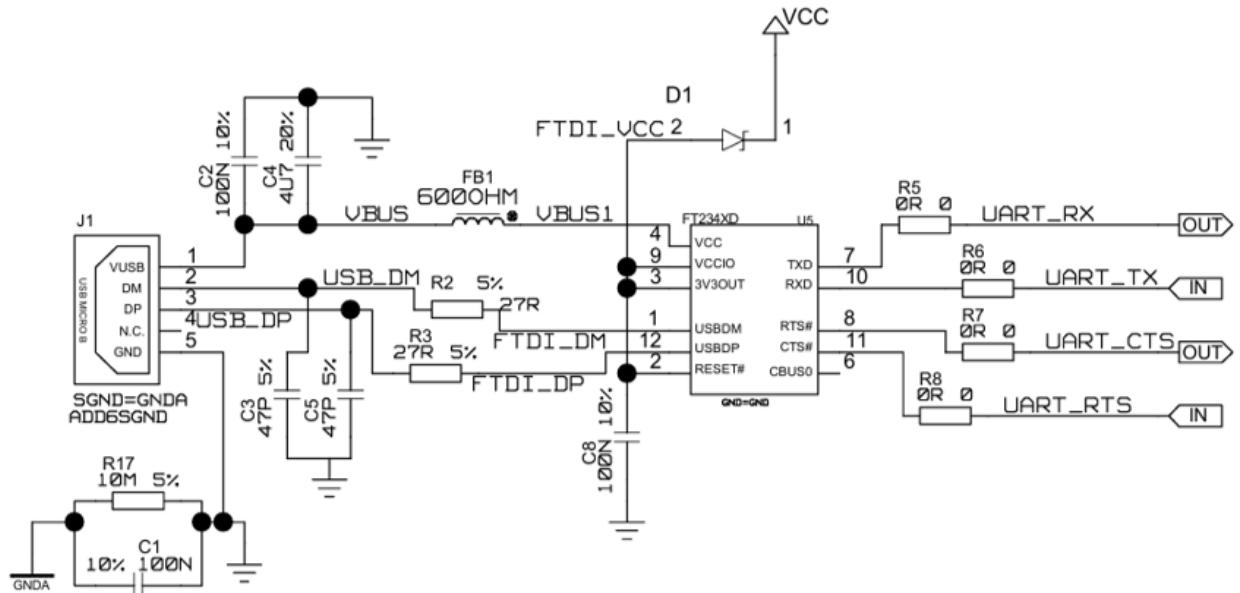
4.2. Chi tiết



Hình 7. Schematic C209 Tag

¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”. Truy cập từ: [Product resources | u-blox](#)

4.2.1. Cụm Bootloader

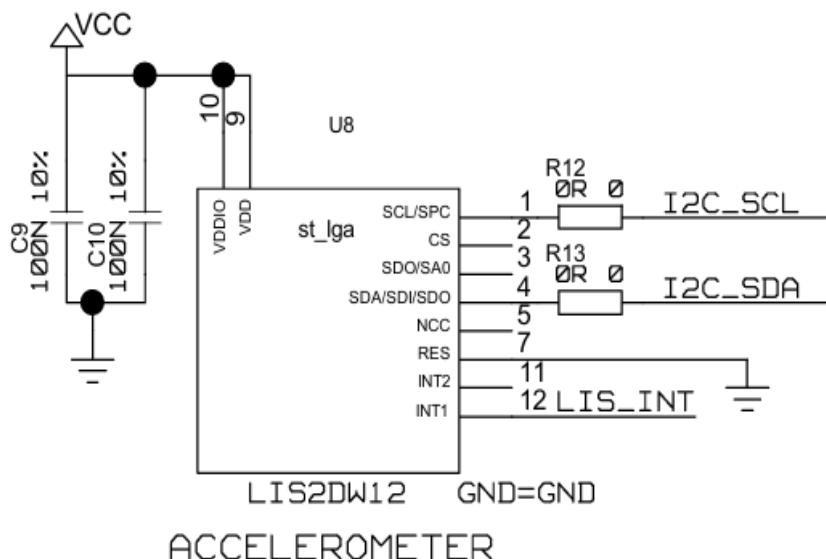


Hình 8. Cụm Bootloader

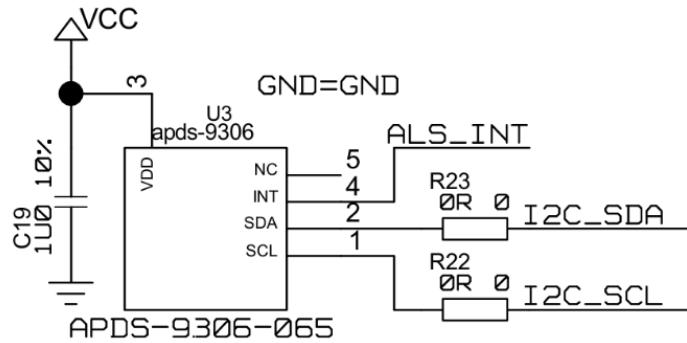
FT234XD sẽ chuyển giao thức USB 2.0 thành UART để giao tiếp với NINA-B406

VBUS 5V qua FB1 sẽ hạ áp xuống còn 3.3V để cấp cho VCC moudule

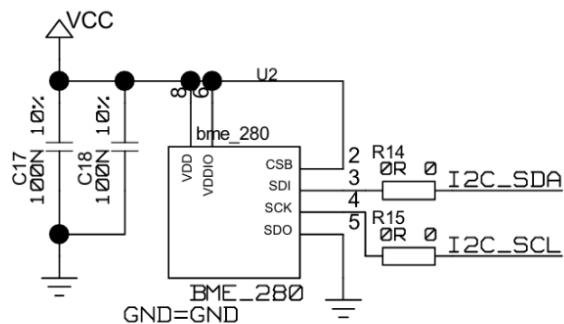
4.2.2. Các cụm cảm biến



Hình 9. Cụm cảm biến gia tốc



Hình 10. Cụm cảm biến ánh sáng



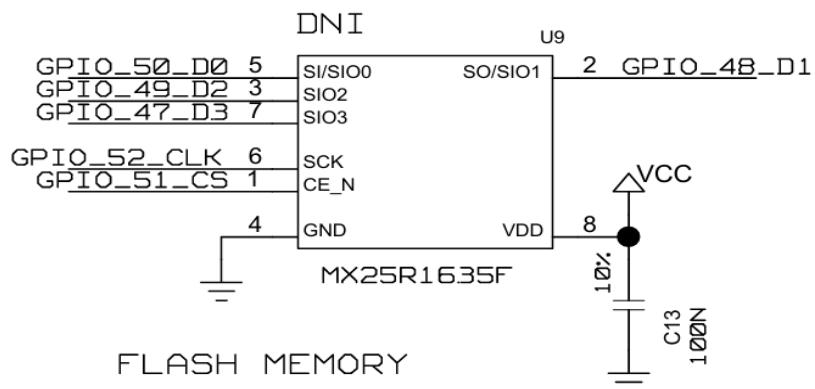
Hình 11. Cụm cảm biến độ ẩm, nhiệt độ và áp suất

Các cụm cảm biến giao tiếp với NINA-B406 qua giao thức I2C, có điện trở kéo lên ở các dây SDA và SCL.

Có ngắt ngoài cho các cụm cảm biến

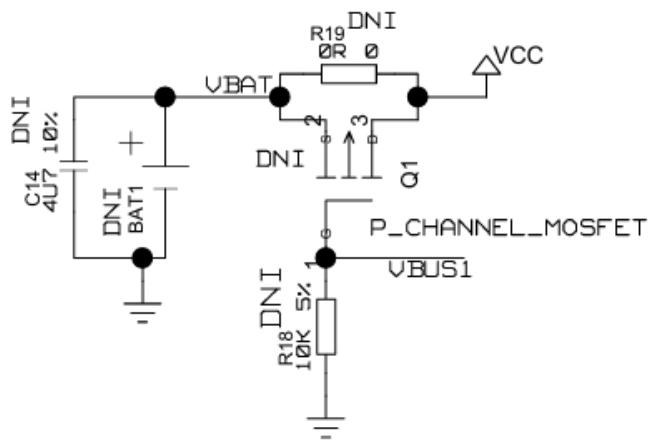
4.2.3. Một số cụm quan trọng khác

4.2.3.1. Cụm bộ nhớ FLASH



Hình 12. Cụm bộ nhớ FLASH

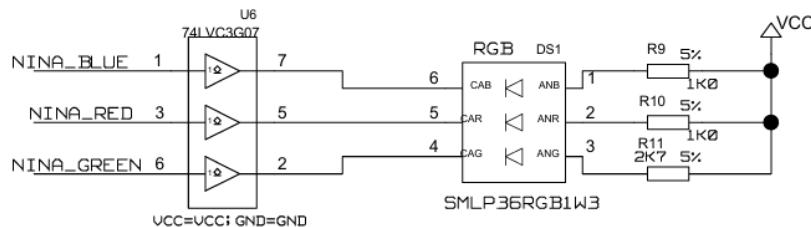
4.2.3.2. Cụm chọn nguồn cấp



Hình 13. Cụm chọn nguồn cấp

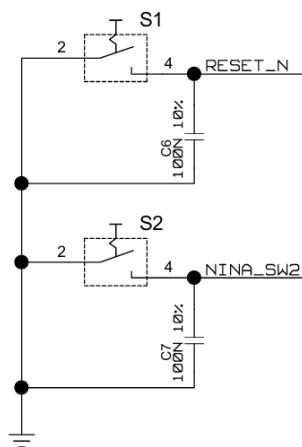
Có thể chuyển đổi qua lại giữa nguồn pin CR2032 và USB thông qua p-MOSFET.

4.2.3.3. Cụm đèn trạng thái



Hình 14. Cụm đèn trạng thái

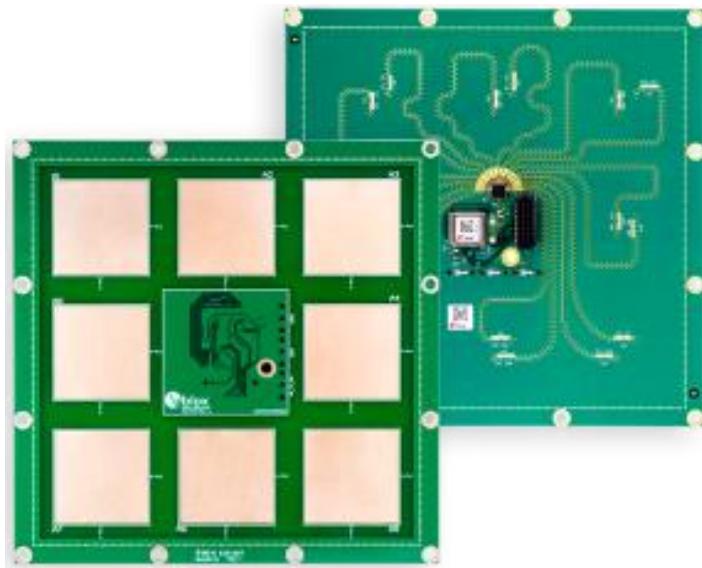
4.2.3.4. Cụm SWITCH



Hình 15. Cụm switch

5. Anchor ANT-B10

ANT-B10 là một bo mạch tìm hướng Bluetooth đa năng, được trang bị tám ăng-ten patch phân cực kép để tìm hướng.



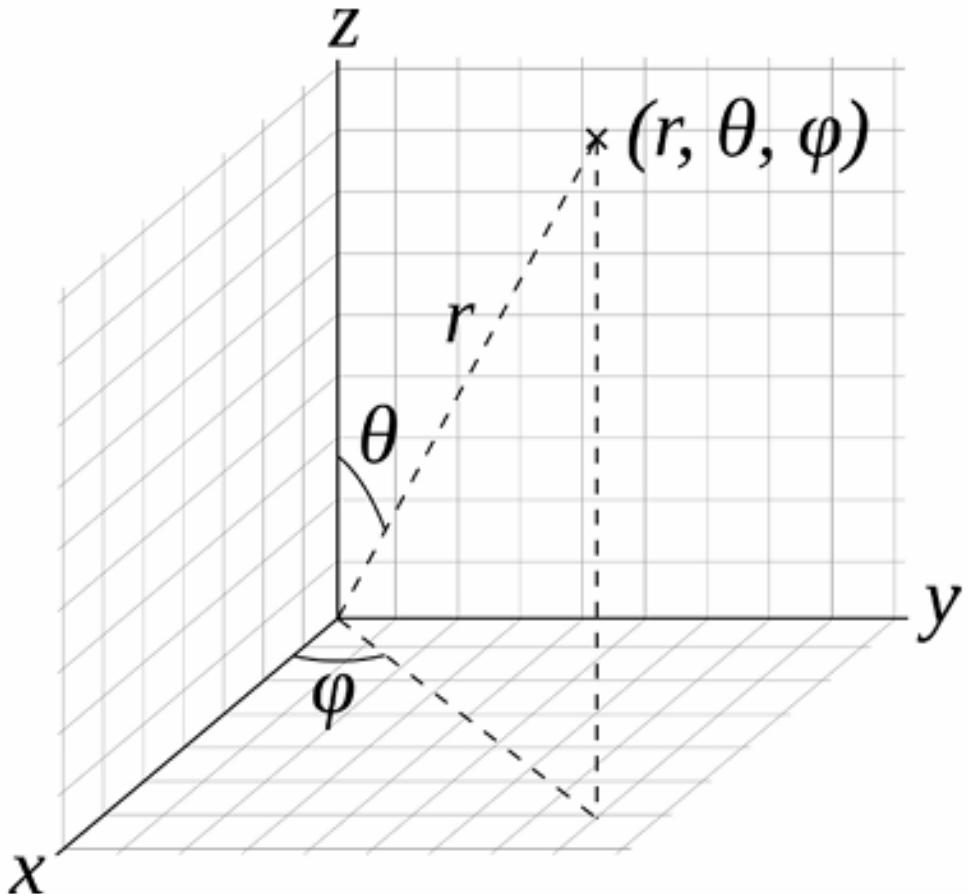
Hình 16. ANT-B10 Board¹

ANT-B10 là một bo mạch tìm hướng Bluetooth hiện đại, được thiết kế để hỗ trợ các ứng dụng định vị trong nhà (Indoor Positioning System - IPS) và IoT. Bo mạch nổi bật với 8 ăng-ten patch phân cực kép, mang lại hiệu suất cao trong việc xác định hướng tín hiệu. Đây là một giải pháp lý tưởng cho các hệ thống cần xác định chính xác góc tới (Angle of Arrival - AoA) của tín hiệu Bluetooth.

Bo mạch tích hợp mô-đun Bluetooth Low Energy (LE) NINA-B411, không chỉ điều khiển các ăng-ten mà còn thực hiện thuật toán tính toán góc tiên tiến thông qua phần mềm u-locateEmbed. Phần mềm này chạy trên vi điều khiển tích hợp (MCU) và cung cấp kết quả chính xác thông qua cổng UART, đảm bảo tính tương thích và dễ dàng tích hợp với các hệ thống khác.

Ngoài ra, ANT-B10 còn được thiết kế với khả năng mở rộng, hỗ trợ các giao diện USB và SPI trong các phiên bản phần mềm u-locateEmbed tương lai. Điều này giúp tăng tính linh hoạt và đáp ứng nhu cầu đa dạng của người dùng trong các ứng dụng như theo dõi tài sản, tự động hóa nhà thông minh, và định vị các thiết bị trong môi trường công nghiệp.

¹ Ublox, “ANT-B10 Bluetooth Low Energy 5.1 antenna board”. Truy cập từ: *ANT-B10 Product Summary*



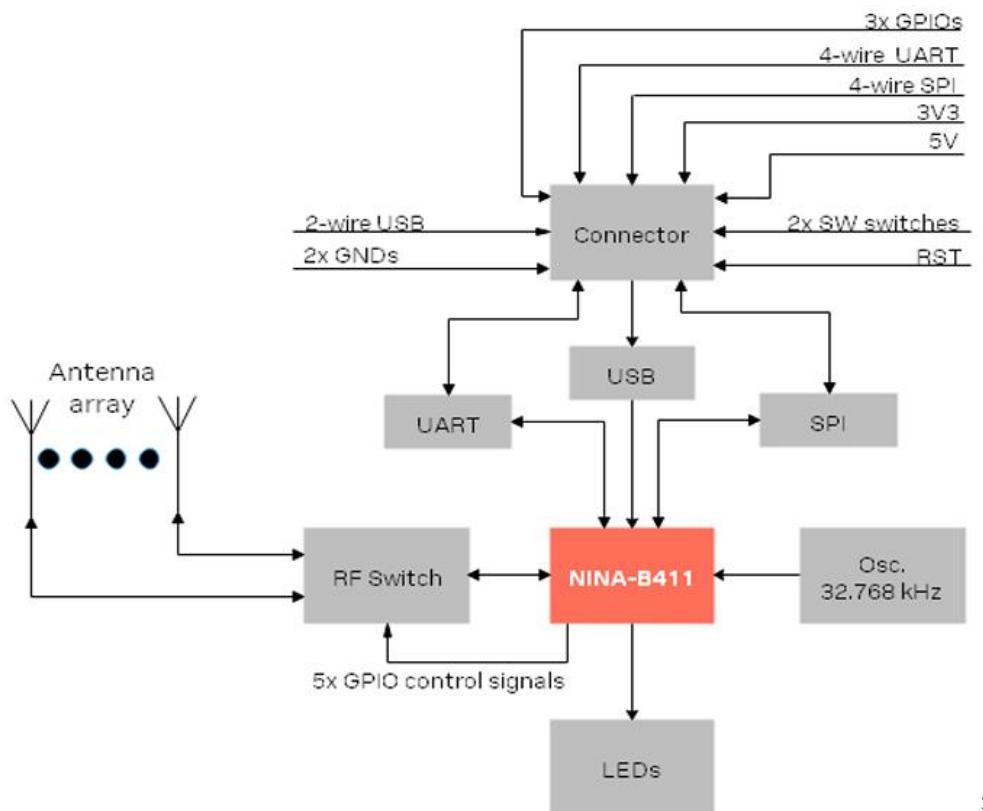
Hình 17. Mô tả các góc trong không gian 3 chiều¹

Các tính năng chính của ANT-B10:

- **Hệ thống ăng-ten vượt trội:** 8 ăng-ten patch phân cực kép cung cấp hiệu suất tối ưu trong tìm hướng.
- **Tích hợp NINA-B411:** Mô-đun Bluetooth LE mạnh mẽ, hỗ trợ điều khiển ăng-ten và tính toán góc chính xác.
- **Thuật toán AoA tiên tiến:** Xử lý tín hiệu và tính toán góc tới hiệu quả thông qua phần mềm u-locateEmbed.
- **Giao diện đa dạng:** Kết nối dễ dàng qua UART, với khả năng mở rộng sang USB và SPI trong tương lai.
- **Ứng dụng rộng rãi:** Phù hợp cho các giải pháp định vị trong nhà, theo dõi thiết bị và IoT.

¹ Byron V Galbraith, Frank Guenther, Massimiliano Versace (08/2015), “A neural network-based exploratory learning and motor planning system for co-robots”. Truy cập từ: (PDF) A neural network-based exploratory learning and motor planning system for co-robots

5.1. Sơ đồ khói của ANT-B10



1

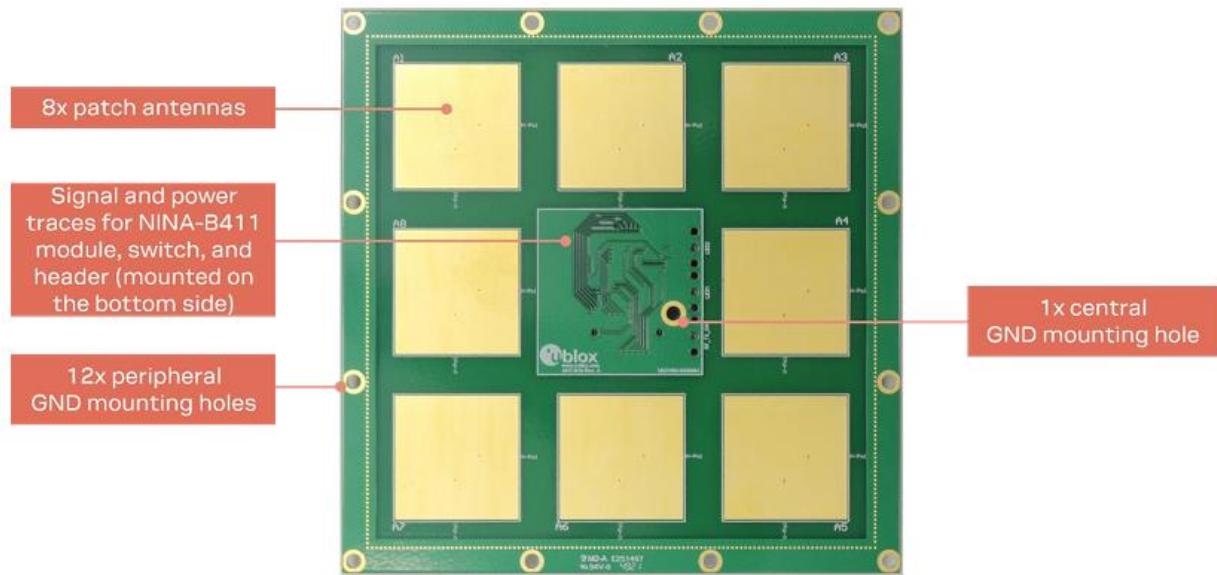
Hình 18. Sơ đồ khói của ANT-B10

5.2. Các thành phần cơ bản của ANT-B10

- **Antena Array:**

- Gồm **8 anten patch** dạng dual-polarized, được sắp xếp để tối ưu hóa hiệu suất trong môi trường có nhiều hiệu ứng đa đường (multipath effects).
- Mỗi anten có độ lợi khác nhau tùy theo tần số và cực tính (vertical/horizontal polarization).
- Anten được tối ưu hóa để hoạt động trong các môi trường phức tạp với hiệu ứng đa đường.

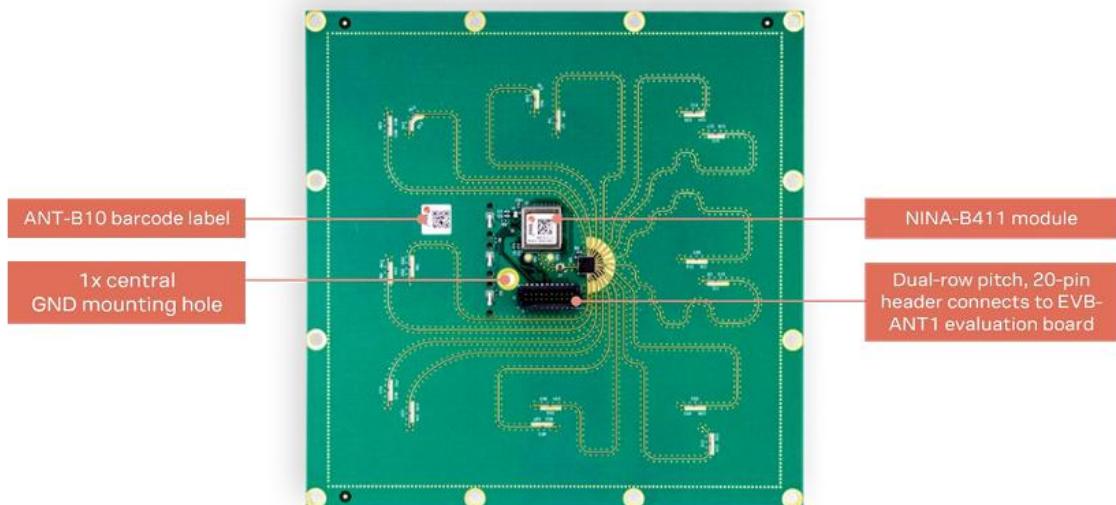
¹ Ublox, “ANT-B10 Bluetooth Low Energy 5.1 antenna board”. Truy cập từ: *ANT-B10 Product Summary*



Hình 19. Mô tả các mảng ăn-ten của ANT-B10¹

- **Module Bluetooth NINA-B411:**

- Đây là module Bluetooth Low Energy (BLE) 5.1, được tích hợp trên board ANT-B10.
- Module này xử lý tín hiệu từ các anten và chạy phần mềm **u-locateEmbed** để tính toán góc đến (AoA - Angle of Arrival).



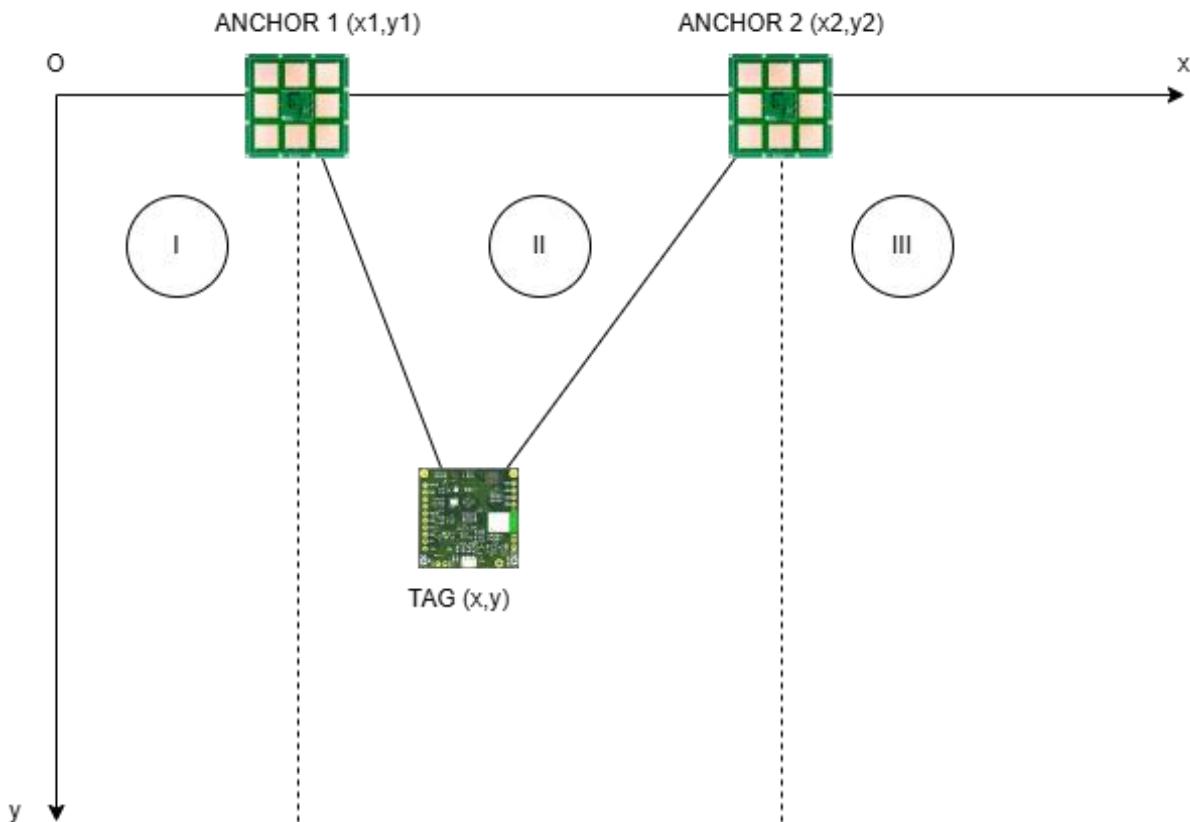
Hình 20. Mặt sau của ANT-B10²

¹ Ublox, “ANT-B10 Bluetooth Low Energy 5.1 antenna board”. Truy cập từ: *ANT-B10 Product Summary*

² Ublox, “ANT-B10 Bluetooth Low Energy 5.1 antenna board”. Truy cập từ: *ANT-B10 Product Summary*

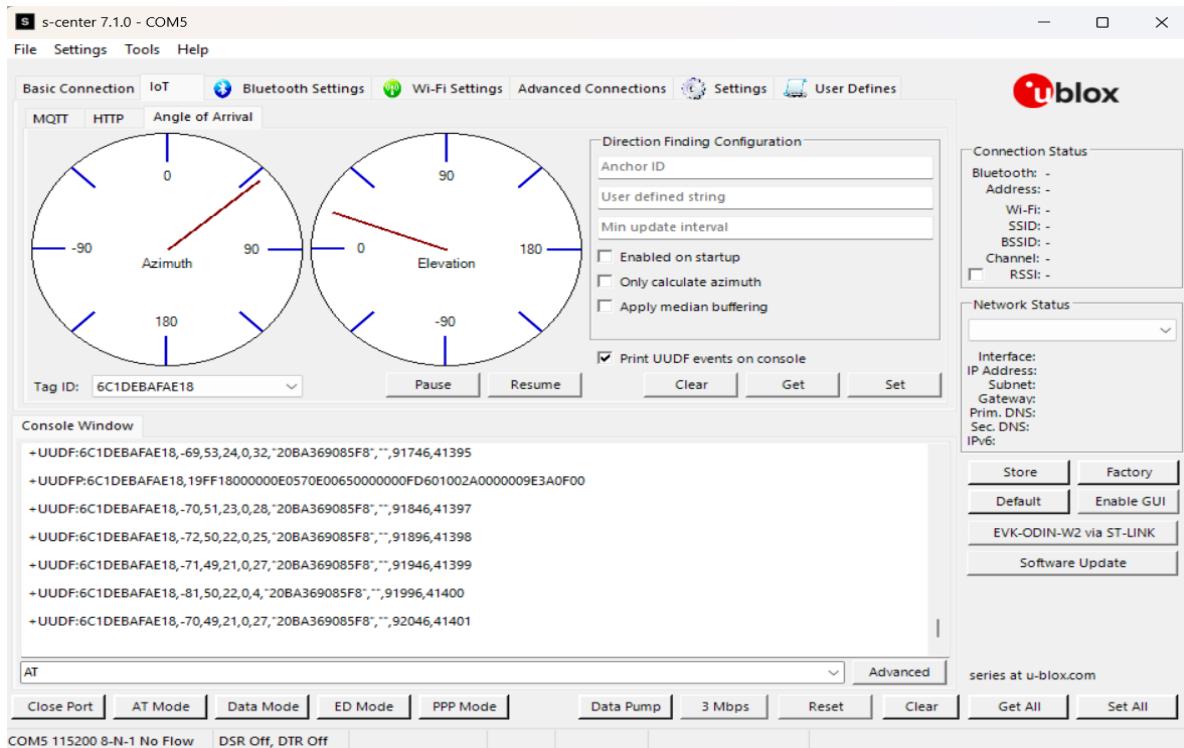
5.3. Chức năng Angel of Arrival (AoA) của ANT-B10

Chức năng AoA có thể được sử dụng với các bố trí các Anchor và Tag như hình sau



Hình 21. Mô tả cách bố trí các Anchor

Các Anchor được bố trí tại các điểm ngẫu nhiên trong khu vực với tọa độ xác định. Tag sẽ được cấu hình để liên tục phát tín hiệu BLE với công suất phát xác định tới các mảng anten trên kit. Lưu ý để đạt được kết quả tốt nhất khoảng cách giữa Anchor và Tag nên thuộc khoảng từ 2-15m Với kit đã được lập trình ta sẽ quan sát được kết quả như sau với phần mềm u-located(s-center):



Hình 22. Kết quả nhận được từ Anchor¹

Kết quả trả về như sau:

```
+UUDF:<ed_instance_id>,<rss>,<azimuth_angle>,<elevation_angle>,<not_used>,
<channel>,<anchor_id>,<user_defined_str>,<tstamp_ms>,<periodic_event_count>
```

Trong đó:

ed_instance_id: ID instance Eddystone 6 byte.

rss: Chỉ số cường độ tín hiệu nhận được (RSSI).

azimuth_angle/direct_angle: Góc phương vị hoặc góc trực tiếp (-90° đến 90°), tùy cấu hình của +UDFCFG param tag.

elevation_angle: Góc cao (-90° đến 90°); bo mạch chỉ báo cáo góc trực tiếp sẽ luôn hiển thị 0°.

<not used>: Thành phần dự trù.

channel: Kênh mà góc của gói tin được tính toán.

anchor_id: Giá trị được thiết lập bởi +UDFCFG param_tag.

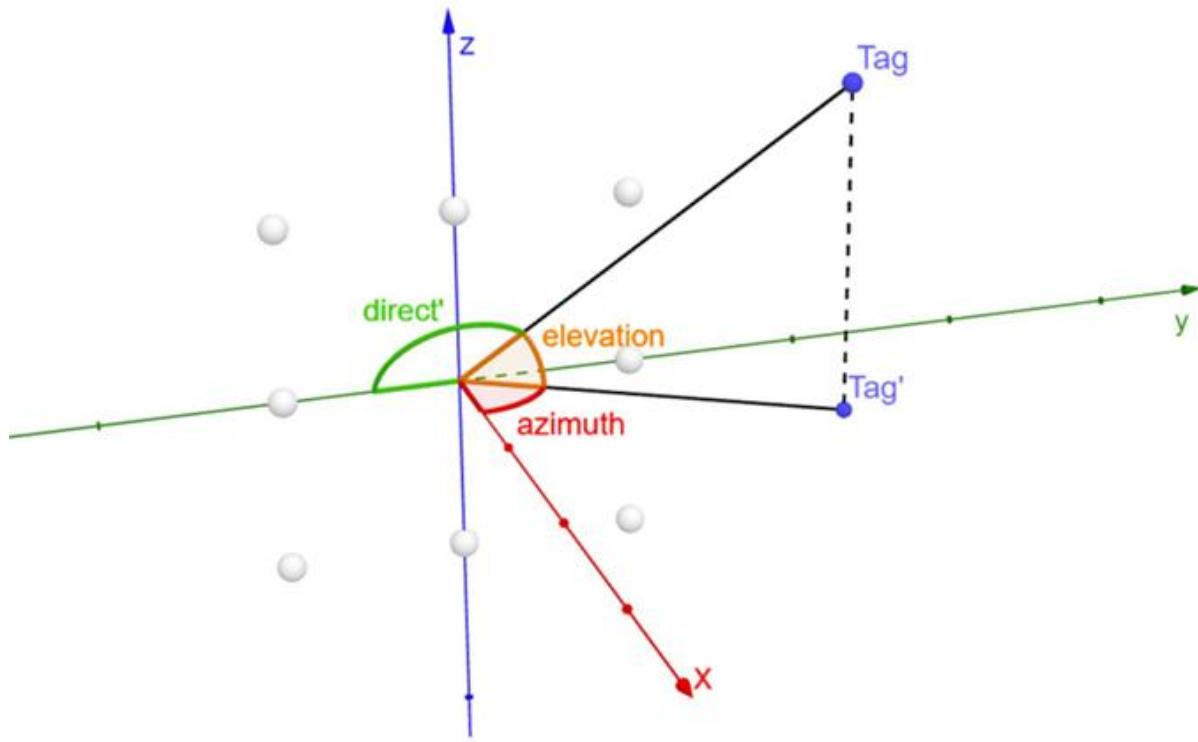
¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”. Truy cập từ: [Product resources | u-blox](#)

user_defined_str: Giá trị được thiết lập bởi +UDFCFG param_tag.

timestamp_ms: Thời gian kể từ khi khởi động (tính bằng mili-giây).

periodic_event_counter: Bộ đếm sự kiện định kỳ dùng để đồng bộ hóa giữa các neo trong thiết lập đa neo.

5.4. Hướng của hệ trục tọa độ và định nghĩa các góc



Hình 23. Mô tả hệ trục tọa độ Anchor¹

- Mô tả góc:

Elevation angel: Góc hợp bởi đường thẳng nối từ gốc hệ trục tọa độ đến Tag và trục Z.

Azimuth angel: Góc hợp bởi hình chiếu đường thẳng nối từ gốc tọa độ đến Tag và trục X.

- Tính toán giá trị tọa độ:

$$x = \text{distance} * \text{Sin(elevationRad)} * \text{Cos(azimuthRad)}$$

$$y = \text{distance} * \text{Sin(elevationRad)} * \text{Sin(azimuthRad)}$$

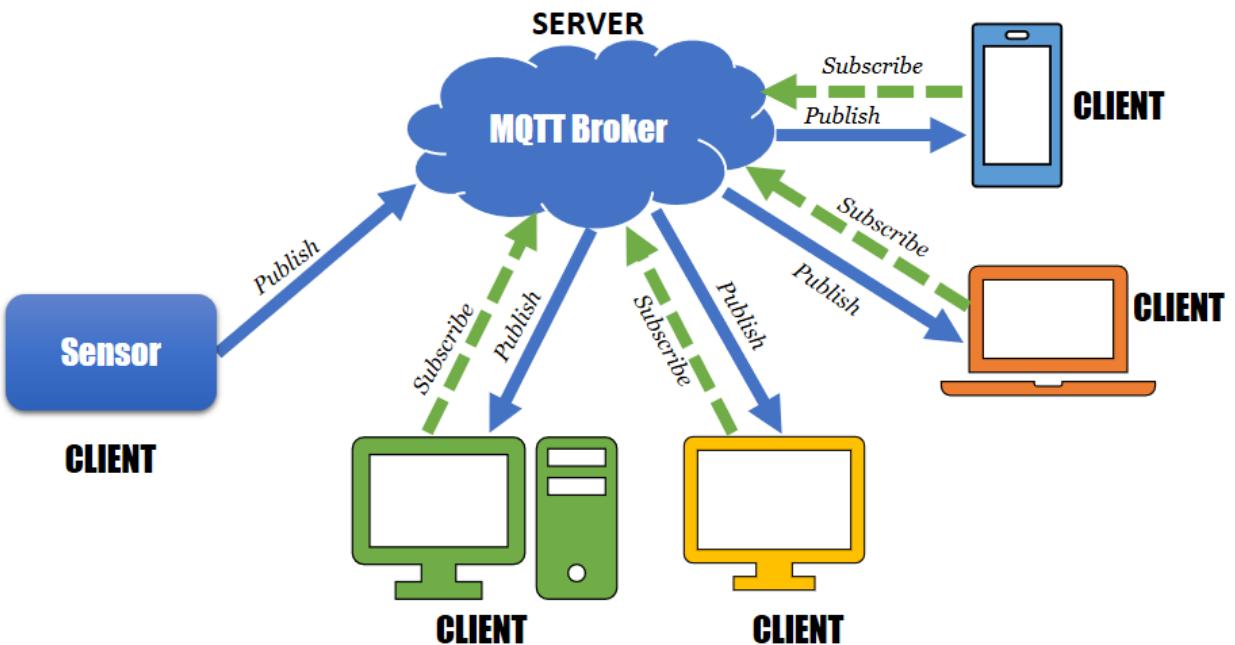
$$z = \text{distance} * \text{Cos(elevationRad)}$$

¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”. Truy cập từ: [Product resources | u-blox](#)

6. Giao thức kết nối

6.1. Message Queuing Telemetry Transport (MQTT)

MQTT (Message Queuing Telemetry Transport) là một giao thức nhắn tin tiêu chuẩn OASIS cho Internet of Things (IoT). Nó được thiết kế như một phương tiện truyền tải tin nhắn publish/subscribe (xuất bản/đăng ký) cực kỳ nhẹ, lý tưởng để kết nối các thiết bị từ xa với băng thông mạng thấp. MQTT ngày nay được sử dụng trong nhiều ngành công nghiệp, chẳng hạn như ô tô, sản xuất, viễn thông, dầu khí,...



Hình 24. Giao thức MQTT¹

6.1.1. Kết nối (Connection)

Quá trình kết nối trong MQTT bắt đầu khi Client tạo kết nối TCP/IP tới Broker thông qua một cổng. Cổng tiêu chuẩn mà MQTT sử dụng là **1883** cho giao tiếp không mã hóa và **8883** cho giao tiếp được mã hóa (SSL/TLS). Khi kết nối thành công, Client gửi một gói tin CONNECT tới Broker, kèm theo các thông tin cần thiết như:

- **Client ID:** Được sử dụng để xác định Client. Nếu một Client kết nối lại sau khi đã có kết nối trước đó, nó có thể tiếp tục một phiên cũ nếu sử dụng lại Client ID đó.
- **Clean Session:** Cờ này cho biết liệu Broker có cần lưu lại trạng thái của Client sau khi kết nối bị ngắt không. Nếu giá trị là false, trạng thái sẽ được lưu lại và khôi phục khi Client kết nối lại.

¹ Leonardo Cavagnis (04/12/2020), “Android and MQTT: A Simple Guide”. Truy cập từ: *Android and MQTT: A Simple Guide. How to develop an MQTT Client with... | by Leonardo Cavagnis | The Startup | Medium*

- **Keep Alive:** Thời gian mà Client yêu cầu giữ kết nối mở mà không có bất kỳ hoạt động nào. Nếu không nhận được dữ liệu trong thời gian này, kết nối sẽ bị đóng.

Khi Broker nhận được gói CONNECT, nó sẽ trả về một gói **CONNACK**, thông báo trạng thái kết nối, có thể là thành công hoặc thất bại. Một số nguyên nhân thất bại có thể bao gồm: Client ID không hợp lệ hoặc phiên kết nối bị lỗi.

6.1.2. Xác thực (Authentication)

Xác thực là một quá trình quan trọng trong giao tiếp MQTT, giúp đảm bảo rằng cả Client và Broker đều là những thực thể đáng tin cậy. Mặc dù không phải lúc nào cũng bắt buộc, nhưng xác thực có thể được thực hiện bằng nhiều phương thức khác nhau.

- **Xác thực qua tên người dùng và mật khẩu:** Trong trường hợp này, Client gửi thông tin xác thực (tên người dùng và mật khẩu) trong gói CONNECT đến Broker. Broker sẽ xác minh thông tin này và trả về một phản hồi phù hợp.
- **Xác thực SSL/TLS:** Nếu Broker yêu cầu một kết nối bảo mật, SSL/TLS sẽ được sử dụng để mã hóa dữ liệu truyền qua mạng và bảo vệ thông tin khỏi các cuộc tấn công bên ngoài. Trong trường hợp này, cả Client và Broker đều cần chứng thực lẫn nhau thông qua các chứng chỉ số. Điều này giúp bảo vệ giao tiếp khỏi các mối đe dọa như giả mạo và nghe lén.

Các Broker công cộng trên internet thường hỗ trợ xác thực qua SSL/TLS, và thậm chí có thể yêu cầu chứng thực từ phía Client, nơi Client cần cung cấp chứng chỉ số của mình để đảm bảo sự an toàn và bảo mật.

6.1.3. Giao tiếp (Communication)

Sau khi kết nối và xác thực thành công, Client và Broker có thể bắt đầu quá trình giao tiếp. MQTT sử dụng mô hình xuất bản/đăng ký (Publish/Subscribe), nơi Client có thể xuất bản (publish) thông tin lên các chủ đề (topics), và các Client khác có thể đăng ký (subscribe) để nhận các thông tin này. Các bước trong giai đoạn giao tiếp bao gồm:

- **Publish:** Client gửi một thông điệp lên một chủ đề mà nó đã đăng ký hoặc chọn.
- **Subscribe:** Client đăng ký để nhận thông điệp từ một chủ đề cụ thể. Broker sẽ chuyển tiếp các thông điệp đến tất cả các Client đã đăng ký vào chủ đề đó.
- **Quality of Service (QoS):** MQTT hỗ trợ ba mức độ QoS để đảm bảo độ tin cậy trong việc truyền tải thông tin:
 - **QoS 0:** Gửi một lần duy nhất, không đảm bảo thông điệp sẽ được nhận.
 - **QoS 1:** Gửi ít nhất một lần và đảm bảo thông điệp sẽ được nhận.

- **QoS 2:** Đảm bảo thông điệp sẽ được nhận đúng một lần.

Giao tiếp này được thực hiện qua các gói tin **PUBLISH**, **SUBSCRIBE**, và **UNSUBSCRIBE** giữa Client và Broker. Các thông điệp MQTT có thể có kích thước nhỏ, điều này giúp giảm thiểu băng thông và tài nguyên tính toán cần thiết cho các thiết bị IoT.

6.1.4. Kết thúc (Termination)

Khi Client không còn cần thiết duy trì kết nối hoặc kết thúc phiên làm việc, Client có thể gửi gói **DISCONNECT** đến Broker. Quá trình kết thúc này sẽ giúp giải phóng tài nguyên trên cả hai phía (Client và Broker). Sau khi nhận gói DISCONNECT, Broker sẽ ngừng giao tiếp với Client và giải phóng kết nối TCP/IP.

Đặc biệt, khi Client kết thúc kết nối, Broker sẽ lưu trữ trạng thái của Client (nếu có yêu cầu giữ lại) và có thể gửi lại bất kỳ thông tin chưa nhận được trong các lần kết nối sau.

6.2. Universal Asynchronous Receiver/Transmitter (UART)

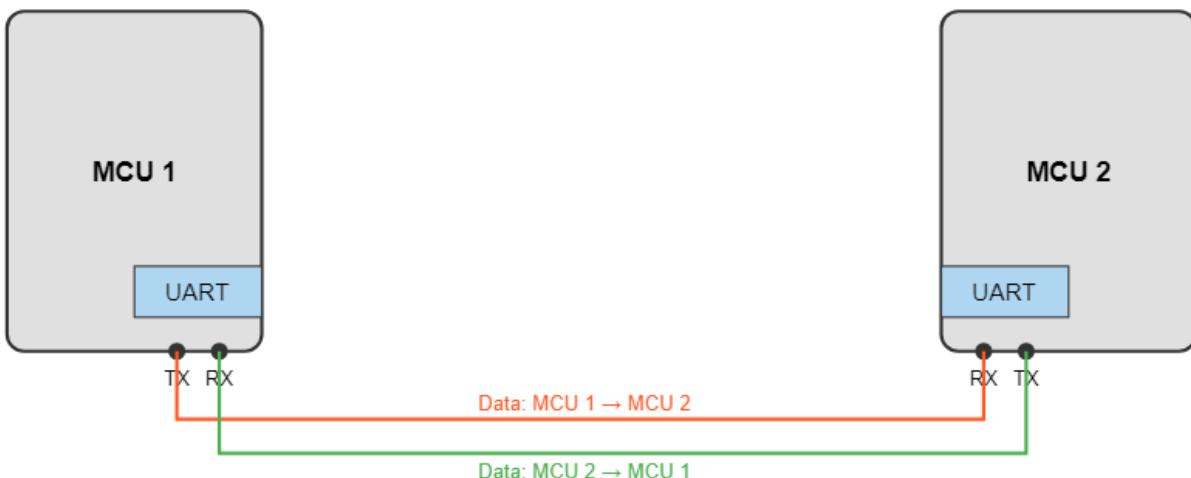
6.2.1. Tổng quan

UART (Universal Asynchronous Receiver/Transmitter) là một giao thức truyền thông nối tiếp được sử dụng rộng rãi trong các hệ thống nhúng và vi điều khiển. UART cho phép giao tiếp song công đầy đủ (full-duplex) giữa hai thiết bị thông qua chỉ hai đường dây tín hiệu, giúp thiết lập kết nối đơn giản và hiệu quả.

6.2.2. Nguyên lý hoạt động

Khác với các giao thức đồng bộ (như SPI, I2C), UART là giao thức bất đồng bộ, nghĩa là không có đường xung clock riêng biệt. Thay vào đó, cả bên phát và bên nhận phải thỏa thuận trước về tốc độ truyền (baud rate). Dữ liệu được truyền từng bit một với tốc độ đã định, trong đó:

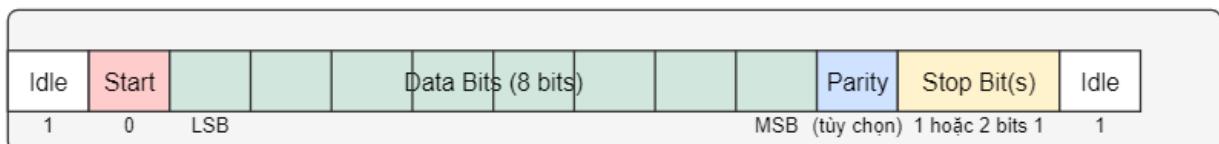
- Khi không truyền dữ liệu, đường tín hiệu ở mức cao (idle state)
- Quá trình truyền bắt đầu với một bit bắt đầu (start bit) ở mức thấp
- Tiếp theo là các bit dữ liệu (thường là 8 bit, nhưng có thể là 5, 6, 7 hoặc 9 bit)
- Có thể có một bit chẵn lẻ (parity bit) để kiểm tra lỗi
- Kết thúc bằng 1 hoặc 2 bit dừng (stop bits) ở mức cao



Hình 25. Giao thức truyền thông bắt đồng bộ UART

6.2.3. Cấu trúc khung dữ liệu

Một khung dữ liệu UART tiêu chuẩn bao gồm:



Hình 26. Cấu trúc khung dữ liệu UART

Trong đó:

Start bit: Luôn là bit 0 (mức thấp), báo hiệu bắt đầu truyền dữ liệu

Data bits: Thông thường là 8 bit, truyền từ LSB (Least Significant Bit) đến MSB (Most Significant Bit)

Parity bit: Tùy chọn, dùng để kiểm tra lỗi (chẵn, lẻ, hoặc không có)

Stop bit(s): 1 hoặc 2 bit ở mức cao (bit 1), đánh dấu kết thúc khung dữ liệu

6.2.4. Đặc điểm kỹ thuật

- Tốc độ truyền (Baud rate):** Thông thường từ 9600 đến 115200 bps, nhưng có thể từ 75 đến 3Mbps
- Không cần xung clock:** Cá hai bên phải cấu hình cùng tốc độ baud và tham số truyền
- Lấy mẫu:** Bên nhận thường lấy mẫu ở giữa mỗi bit để đảm bảo độ chính xác
- Khoảng cách truyền:** Thường giới hạn trong vài mét, nhưng có thể mở rộng với các chuẩn như RS-232, RS-485

6.2.5. Ứng dụng phổ biến

- Giao tiếp giữa vi điều khiển và các cảm biến
- Kết nối với máy tính thông qua cổng COM hoặc USB-to-UART
- Giao tiếp với module Bluetooth, WiFi, GPS
- Gỡ lỗi (debugging) và lập trình cho các hệ thống nhúng
- Giao tiếp giữa các vi điều khiển trong mạng đơn giản

6.2.6. Ưu điểm, hạn chế và so sánh

Ưu Điểm

- Đơn giản, chỉ cần 2 dây tín hiệu (TX và RX)
- Được hỗ trợ rộng rãi trên hầu hết các vi điều khiển
- Dễ dàng triển khai bằng phần mềm nếu không có phần cứng UART
- Tương thích với các chuẩn truyền thông như RS-232, RS-485, TTL

Hạn Chế

- Tốc độ chậm hơn so với giao thức đồng bộ như SPI
- Không hỗ trợ nhiều thiết bị trên cùng một bus (chỉ kết nối điểm-điểm)
- Yêu cầu đồng bộ về tốc độ baud giữa bên phát và bên nhận
- Khả năng phát hiện lỗi hạn chế (chỉ qua bit chẵn lẻ nếu được sử dụng)

So Sánh VỚI CÁC GIAO THỨC KHÁC

Đặc Điểm	UART	SPI	I ² C
Số dây	2 (TX/RX)	4 (MOSI, MISO, CLK, CS)	2 (SDA, SCL)
Tốc độ	Trung bình	Nhanh	Chậm-Trung bình
Khoảng cách	Ngắn-Trung bình	Ngắn	Ngắn
Thiết bị tối đa	2 (point-to-point)	Nhiều (với nhiều CS)	Nhiều (địa chỉ 7-bit)
Phức tạp	Thấp	Trung bình	Trung bình

Bảng 9. So sánh UART với các giao thức khác

7. Bộ lọc

7.1. Bộ lọc Kalman

7.1.1. Giới thiệu

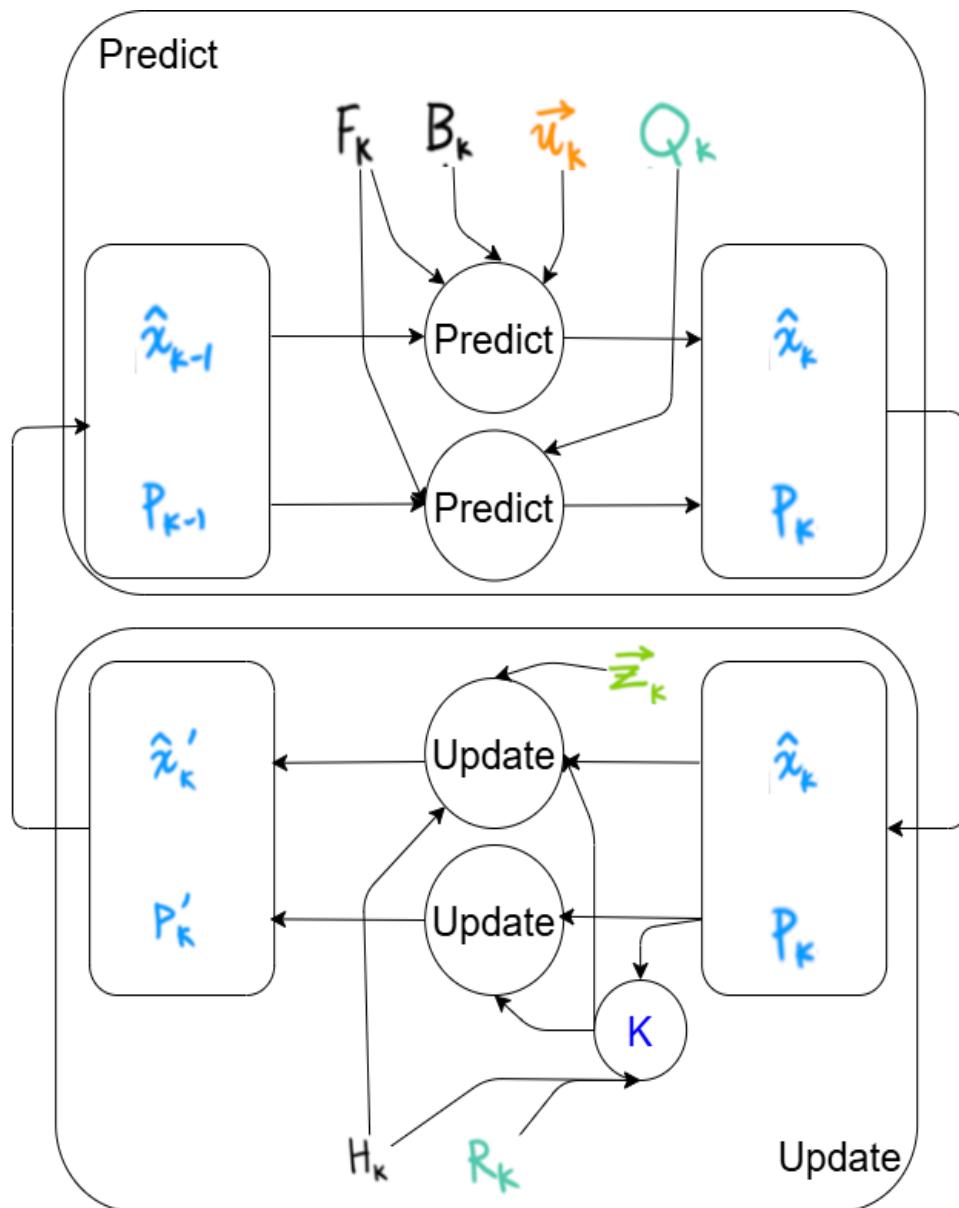
Bộ lọc Kalman là một thuật toán lọc đệ quy được phát triển bởi Rudolf E. Kálmán vào năm 1960. Thuật toán này đã trở thành một trong những công cụ quan trọng nhất trong lĩnh vực xử lý tín hiệu, điều khiển tự động, và các hệ thống điều hướng. Bộ lọc Kalman cho phép ước lượng trạng thái của hệ thống động dựa trên chuỗi các phép đo không hoàn hảo (chứa nhiễu).

Bộ lọc Kalman được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau do khả năng:

- Khử nhiễu hiệu quả từ dữ liệu đo lường
- Ước lượng các biến trạng thái không thể đo trực tiếp
- Kết hợp tối ưu thông tin từ nhiều nguồn khác nhau
- Thích ứng tốt với các hệ thống thời gian thực

7.1.2. Cơ sở lý thuyết

Bộ lọc Kalman hoạt động theo quy trình hai bước lặp lại: dự đoán (predict) và cập nhật (update). Thuật toán này dựa trên ý tưởng rằng, nếu chúng ta biết trạng thái hiện tại của hệ thống và mô hình động học của nó, chúng ta có thể dự đoán trạng thái tiếp theo. Sau đó, khi có phép đo mới, chúng ta kết hợp dự đoán này với phép đo để có được ước lượng tốt hơn.



Hình 27. Sơ đồ khái bô lọc Kalman

Trong đó :

- **X**: Biến trạng thái, biểu diễn giá trị hiện tại được ước lượng.
- **P**: Độ không chắc chắn của ước lượng (covariance).
- **Q**: Nhiễu quá trình (process noise), phản ánh độ không chắc chắn trong mô hình hệ thống.
- **R**: Nhiễu đo lường (measurement noise), phản ánh độ không chắc chắn trong các phép đo.
- **K**: Hệ số Kalman (Kalman gain), xác định mức độ tin cậy giữa dự đoán và phép đo.
- **F_k**: là ma trận chuyển trạng thái
- **B_k**: là ma trận điều khiển đầu vào
- **u̅_k**: là vector điều khiển (input)

7.1.3.Công thức toán học

- Bước Dự đoán (Prediction):

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k \vec{u}_k$$

$$P_k = F_k P_{k-1} F_k^T + Q_k$$

- Bước Cập nhật thông số (Update):

$$\hat{x}'_k = \hat{x}_k + K'(\vec{z}_k - H_k \hat{x}_k)$$

$$P'_k = P_k - K' H_k P_k$$

$$K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

7.1.4 Ứng dụng thực tế

Khi Tag đứng yên, chúng ta có thể áp dụng hai bộ lọc Kalman độc lập cho hai tọa độ (x,y) để khử nhiễu từ ESP32. Các thông số được đơn giản hóa như sau:

- $F_k = 1$ (ma trận đơn vị trong trường hợp vô hướng)
- $B_k * \vec{u}_k = 0$ (không có đầu vào điều khiển)
- $H_k = 1$ nghĩa là bạn đang đo lường trực tiếp trạng thái hệ thống. Trong trường hợp của bạn, nếu X là vị trí x hoặc y, và measurement cũng là vị trí x hoặc y, thì $H_k = 1$ là hợp lý.

Từ đó ta có bộ lọc Kalman đơn giản:

- Predict:

$$\hat{x}_k = \hat{x}_{k-1}$$

$$P_k = P_{k-1} + Q_k$$

- Update:

$$K' = P_k(P_k + R_k)^{-1}$$

$$\hat{x}'_k = \hat{x}_k + K'(\vec{z}_k - \hat{x}_k)$$

$$P'_k = P_k - K'P_k$$

7.2. Bộ lọc trung vị Median

7.2.1. Khái niệm cơ bản

Bộ lọc trung vị là một kỹ thuật xử lý tín hiệu phi tuyến tính, hoạt động bằng cách thay thế một giá trị trong chuỗi dữ liệu bằng **giá trị trung vị** của các giá trị trong một cửa sổ lân cận xung quanh điểm đó. Giá trị trung vị là giá trị nằm ở giữa khi các giá trị trong cửa sổ được sắp xếp theo thứ tự tăng dần.

- **Mục đích chính:**

- Loại bỏ các giá trị bất thường (outliers) trong chuỗi dữ liệu, ví dụ: các giá trị đột biến do lỗi cảm biến hoặc nhiễu ngẫu nhiên.
- Làm mịn tín hiệu mà vẫn bảo toàn các đặc trưng quan trọng, như các bước nhảy (step changes) hoặc xu hướng.
- Không làm mất thông tin cấu trúc của tín hiệu, khác với bộ lọc trung bình (mean filter) có thể làm mờ các thay đổi đột ngột.

- **So sánh với bộ lọc trung bình:**

- Bộ lọc trung bình tính trung bình cộng các giá trị trong cửa sổ, dễ bị ảnh hưởng bởi outliers.
- Bộ lọc trung vị chọn giá trị trung vị, giúp loại bỏ các giá trị cực đại/cực tiểu bất thường mà không làm thay đổi đáng kể cấu trúc tín hiệu.

7.2.2. Cách hoạt động của bộ lọc trung vị

Bộ lọc trung vị xử lý một chuỗi dữ liệu theo các bước sau:

1. **Xác định cửa sổ lân cận:**

- Chọn một cửa sổ có kích thước $2k+1$ (số lẻ để đảm bảo có giá trị trung vị duy nhất), trong đó k là số lượng điểm dữ liệu ở mỗi bên của điểm trung tâm.
- Ví dụ: Với cửa sổ kích thước 5 ($k=2$), cửa sổ tại điểm $x[n]$ bao gồm các giá trị $\{x[n-2], x[n-1], x[n], x[n+1], x[n+2]\}$.

2. Lấy giá trị trong cửa sổ:

- Thu thập tất cả các giá trị trong cửa sổ lân cận.

3. Sắp xếp giá trị:

- Sắp xếp các giá trị trong cửa sổ theo thứ tự tăng dần.

4. Chọn giá trị trung vị:

- Lấy giá trị ở vị trí giữa của danh sách đã sắp xếp. Với cửa sổ kích thước $2k+1$, giá trị trung vị là giá trị thứ $k+1$.

5. Thay thế giá trị gốc:

- Giá trị tại $x[n]$ được thay bằng giá trị trung vị vừa tính.

6. Di chuyển cửa sổ:

- Tiếp tục di chuyển cửa sổ qua từng điểm dữ liệu trong chuỗi và lặp lại quá trình.

7.2.3. Công thức toán học

Cho chuỗi dữ liệu $x[n] x[n] x[n]$ và cửa sổ kích thước $2k+1$, giá trị đầu ra của bộ lọc trung vị tại vị trí n được định nghĩa như sau:

$$y[n] = \text{median}\{x[n-k], x[n-k+1], \dots, x[n], \dots, x[n+k-1], x[n+k]\}$$

Trong đó:

- $y[n]$ là giá trị đầu ra của bộ lọc trung vị.
- median là hàm lấy giá trị trung vị, tức là giá trị ở giữa sau khi sắp xếp các giá trị trong cửa sổ.

7.2.4. Ưu điểm của bộ lọc trung vị

- **Loại bỏ outliers hiệu quả:** Các giá trị bất thường (ví dụ: do lỗi cảm biến hoặc nhiễu ngẫu nhiên) được loại bỏ vì chúng thường không phải là giá trị trung vị.
- **Bảo toàn các đặc trưng tín hiệu:** Giữ nguyên các thay đổi đột ngột (như bước nhảy) hoặc xu hướng, không làm mờ tín hiệu như bộ lọc trung bình.

- **Đơn giản và mạnh mẽ:** Không cần giả định về phân phối của dữ liệu (khác với các bộ lọc tuyến tính như bộ lọc Kalman).
- **Ứng dụng linh hoạt:** Phù hợp cho nhiều loại tín hiệu, từ dữ liệu thời gian thực đến dữ liệu cảm biến.

7.2.5. Nhược điểm của bộ lọc trung vị

- **Tốn tài nguyên tính toán:** Việc sắp xếp các giá trị trong cửa sổ đòi hỏi độ phức tạp tính toán cho mỗi điểm dữ liệu, đặc biệt với cửa sổ lớn.
- **Hiệu quả hạn chế với nhiễu Gaussian:** Bộ lọc trung vị không hiệu quả bằng các bộ lọc tuyến tính (như bộ lọc Gaussian) khi xử lý nhiễu có phân phối đều.
- **Có thể làm mất chi tiết nhỏ:** Nếu cửa sổ quá lớn, các biến đổi nhanh hoặc chi tiết nhỏ trong tín hiệu có thể bị làm mịn mất.
- **Không phù hợp với tín hiệu phức tạp:** Với các tín hiệu có nhiều tuần hoàn hoặc cấu trúc phức tạp, bộ lọc trung vị có thể không mang lại kết quả tối ưu.

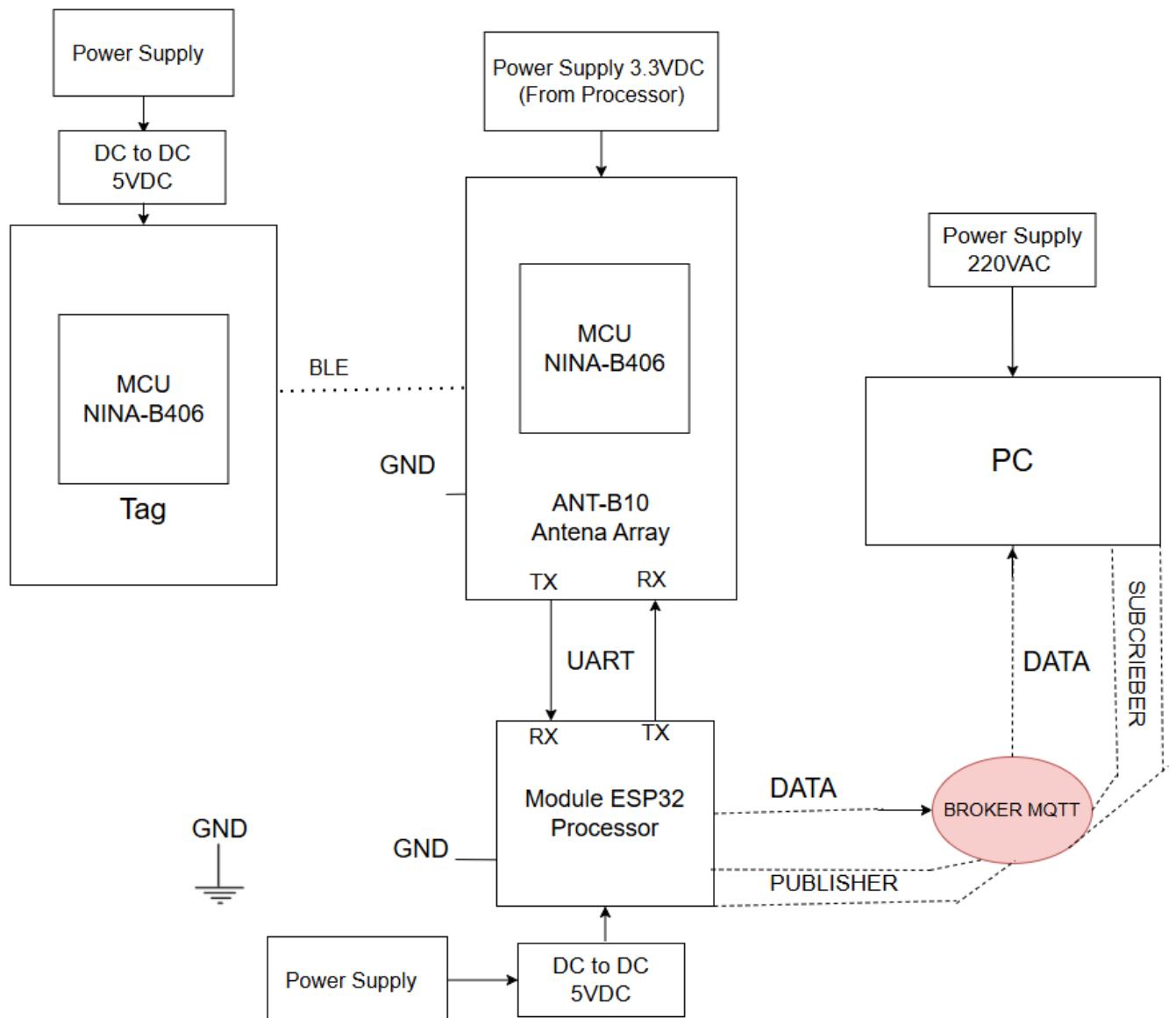
7.2.6. Ứng dụng thực tế

Bộ lọc trung vị được sử dụng rộng rãi trong xử lý tín hiệu và dữ liệu số, bao gồm:

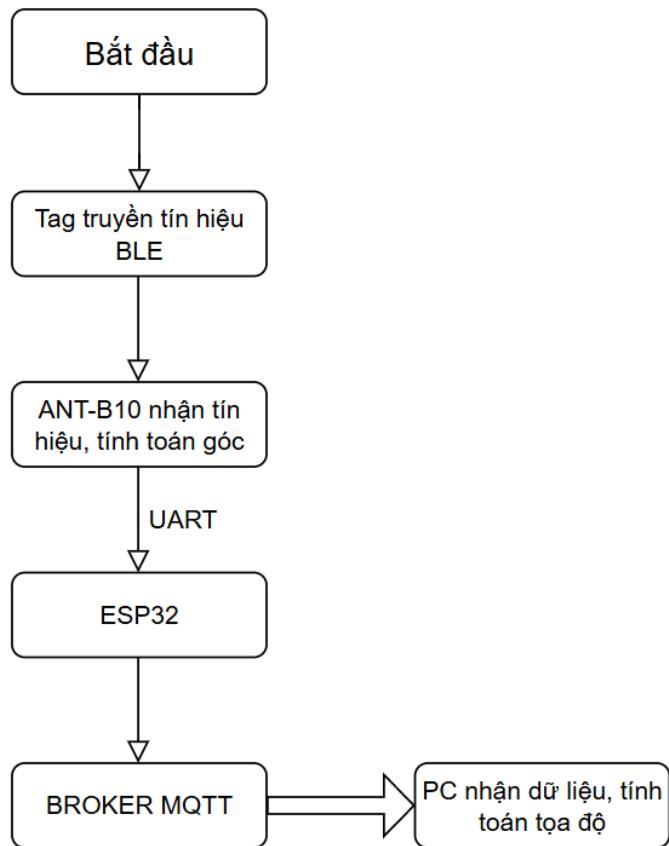
- **Xử lý tín hiệu thời gian thực:**
 - Làm mịn dữ liệu từ cảm biến (nhiệt độ, áp suất, gia tốc) để loại bỏ các giá trị bất thường.
 - Xử lý tín hiệu trong các hệ thống nhúng hoặc IoT.
- **Kỹ thuật điều khiển:**
 - Làm mịn tín hiệu đầu vào trong các hệ thống điều khiển để cải thiện độ chính xác của bộ điều khiển.

CHƯƠNG 3: TỔNG QUAN HỆ THỐNG

1. Sơ đồ khái niệm hệ thống



Hình 28. Sơ đồ khái niệm hệ thống



Hình 29. Mô tả hoạt động hệ thống

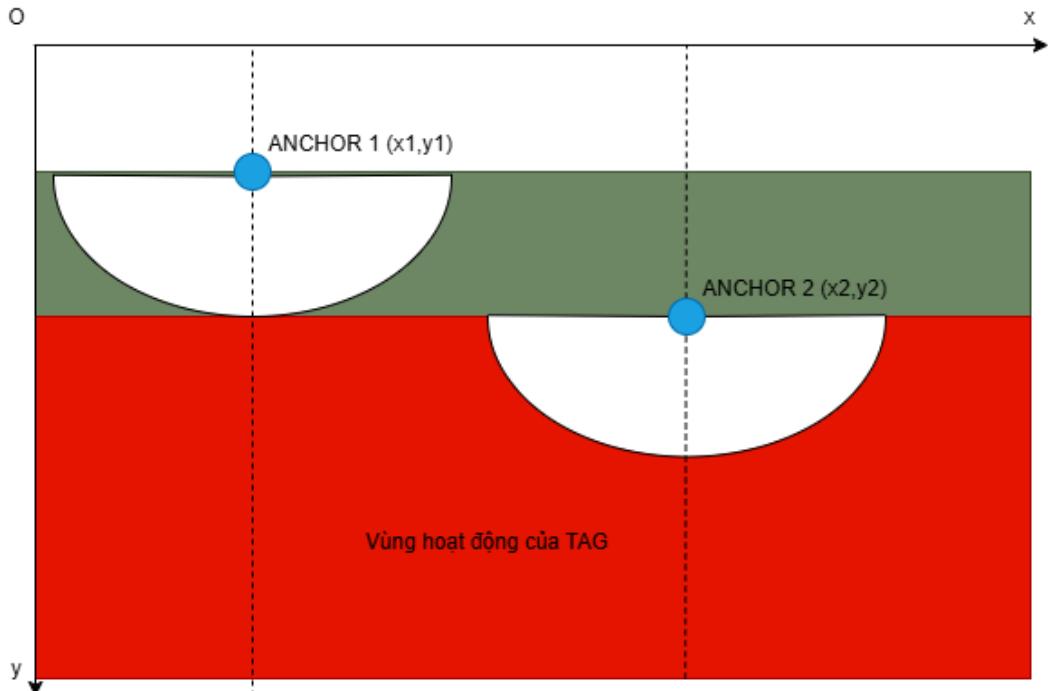
2. Phương pháp bố trí

2.1. Giới thiệu

Nghiên cứu này tập trung phân tích các phương án bố trí Anchor để tối đa hóa vùng phủ sóng và độ chính xác của hệ thống định vị. Xét môi trường là một căn phòng hình chữ nhật được gắn hệ trực tọa độ Oxy, với hai Anchor cố định và một Tag di động cần xác định vị trí. Nghiên cứu này phân tích ba phương án bố trí Anchor khác nhau để xác định cấu hình tối ưu.

2.2. Phương pháp bố trí Anchor

2.2.1. Bố trí ngẫu nhiên

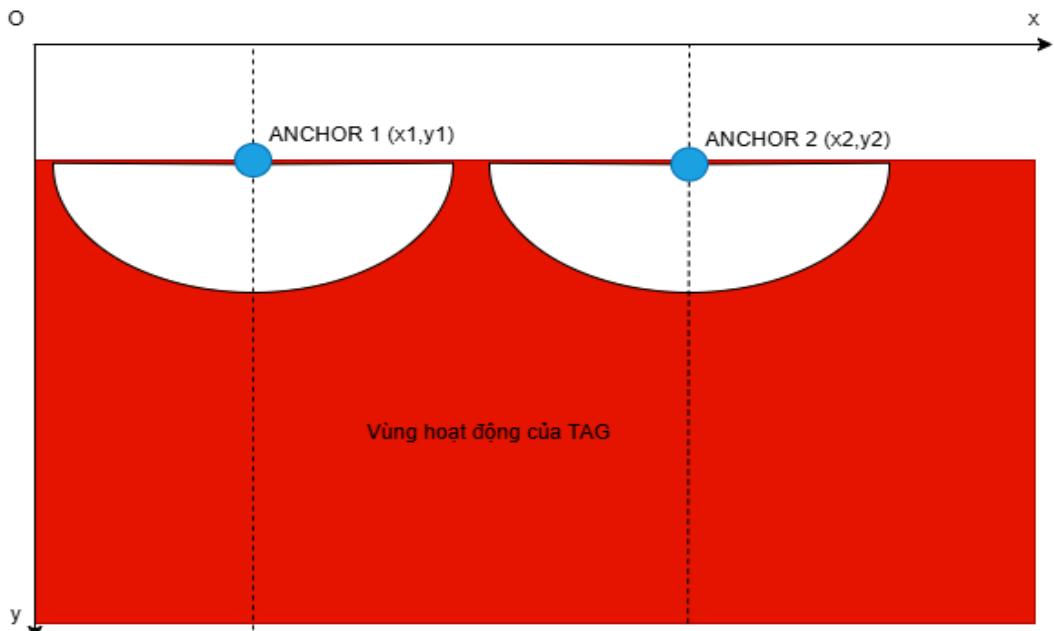


Hình 30. Bố trí Anchor ngẫu nhiên

Ta thấy vùng hoạt động của Tag sẽ không được tối ưu:

- Ở vùng xanh lá, góc Azimuth của Anchor 2 không hoạt động nên Tag sẽ không thể được định vị ở vùng này.
- Tương tự với vùng màu trắng, Tag cũng không thể được định vị ở vùng này .

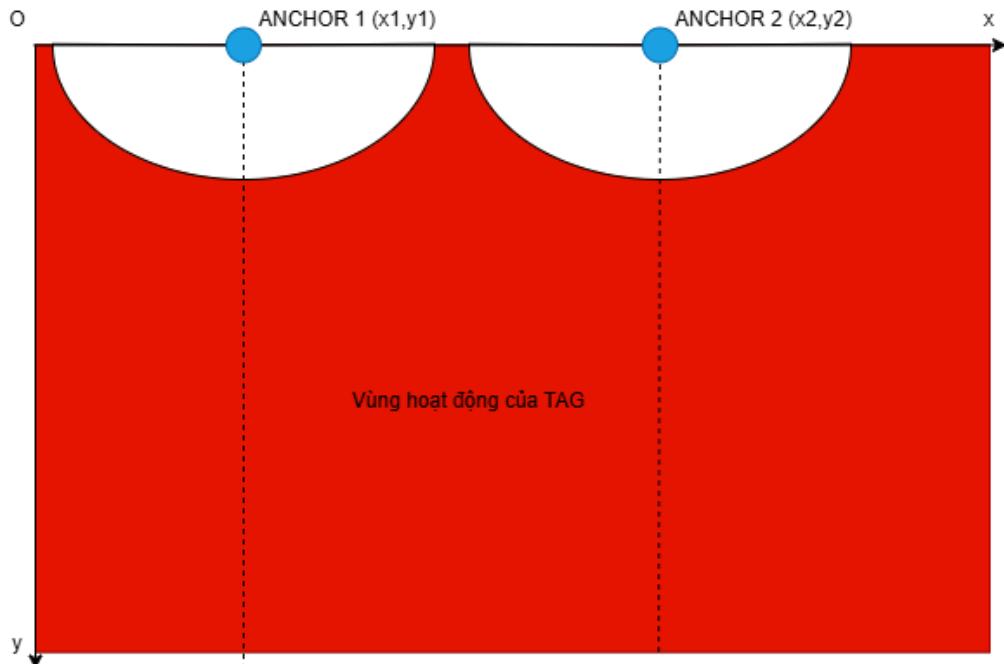
2.2.2. 2 Anchor nằm trên cùng đường thẳng song song với Ox



Hình 31. Bố trí 2 Anchor trên đường thẳng song song với Ox

Ta thấy vùng phủ sóng đã được mở rộng nhưng ở vùng màu trắng, Tag vẫn không thể được định vị vì quá phạm vi của 2 góc Azimuth.

2.2.3. Đặt 2 Anchor nằm trên trục Ox



Hình 32. Bố trí 2 Anchor nằm trên Ox

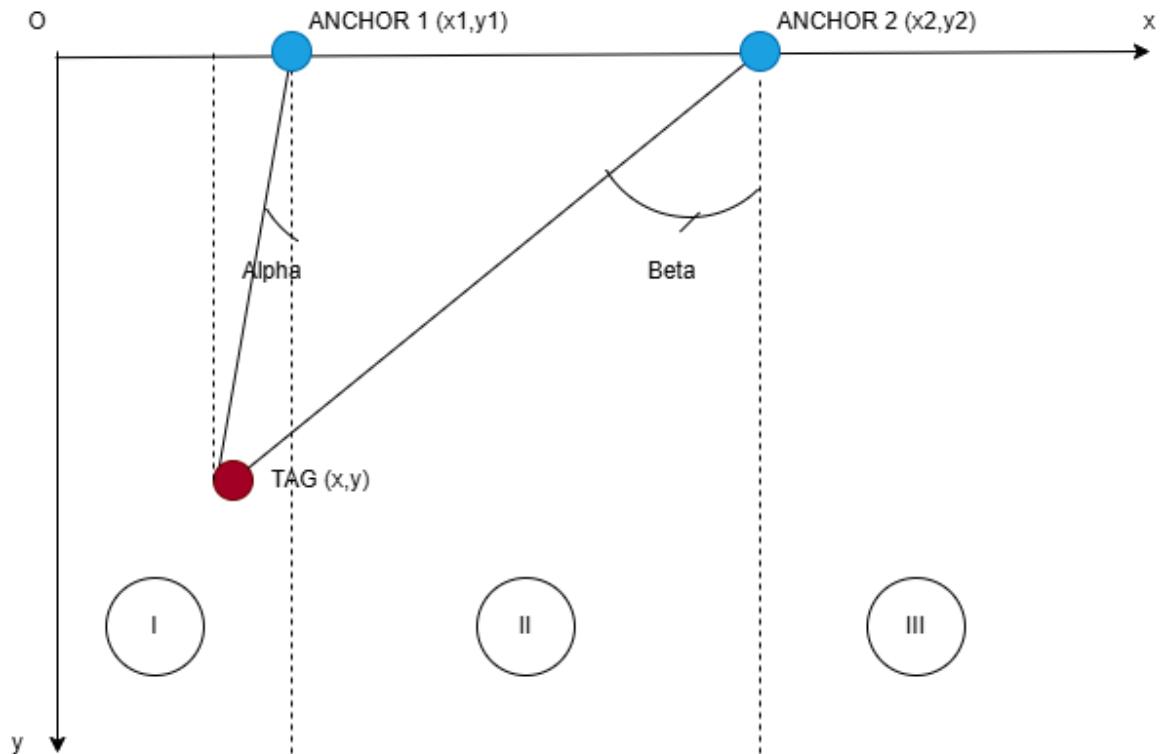
Khi đặt 2 Anchor ở Ox (sát tường), ta thấy vùng phủ sóng của Tag đã được mở rộng ra toàn bộ căn phòng.

➔ Cách bố trí 2 Anchor tối ưu nhất.

2.3. Xác định vị trí Tag

Sau khi đã xác định được các bố trí 2 Anchor, ta tiến hành tính toán vị trí Tag dựa trên dữ liệu đã có từ cách bố trí 2 Anchor: $y_1 = 0$; $y_2 = 0$

2.3.1. Khi Tag ở vùng phần ba thứ nhất



Hình 33. Tag ở vùng phần ba thứ nhất

Từ hình, ta có phương trình (1) và (2):

$$\frac{x_2 - x}{y} = \tan(\beta) \quad (1)$$

$$\frac{x_1 - x}{y} = \tan(a) \quad (2)$$

Ta có hệ phương trình:

$$x + y * \tan(a) = x_1$$

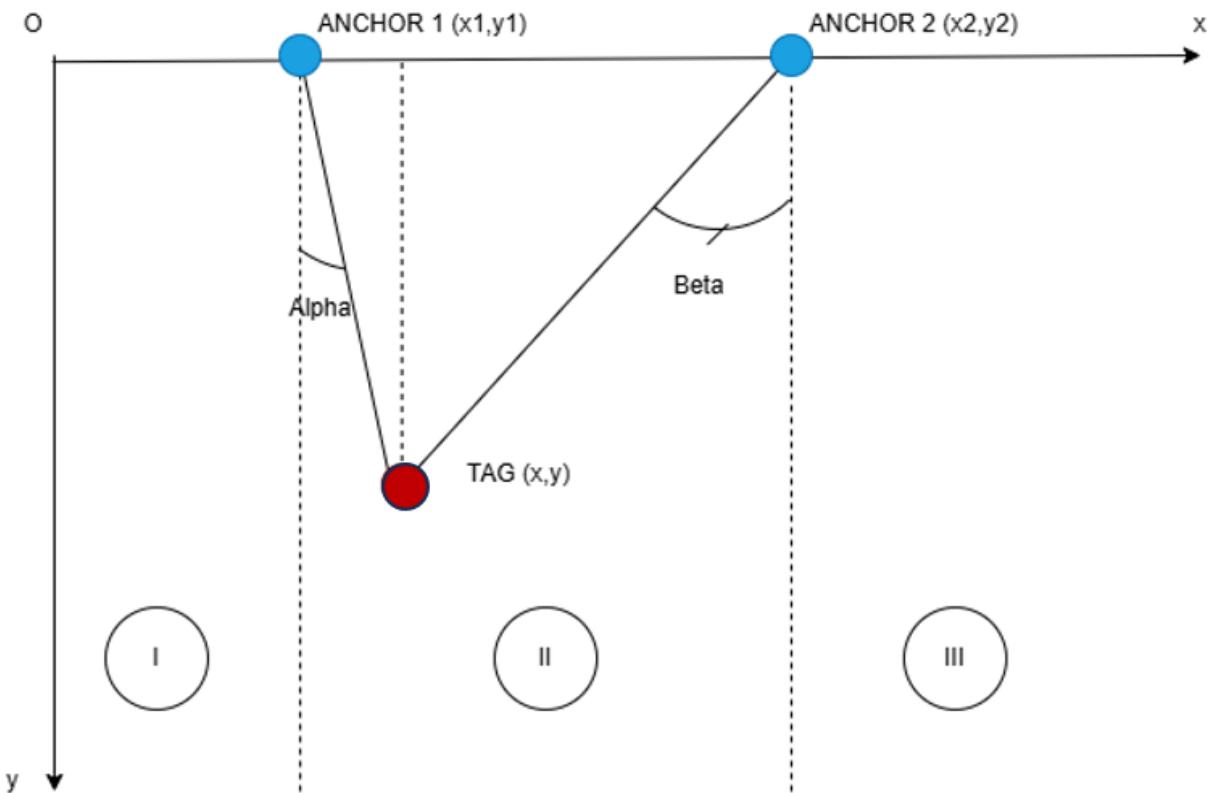
$$x + y * \tan(\beta) = x_2$$

Giải hệ phương trình, ta được

$$x = \frac{x_1 * \tan(a) - x_2 * \tan(\beta)}{\tan(\beta) - \tan(a)}$$

$$y = \frac{x_2 - x_1}{\tan(\beta) - \tan(a)}$$

2.3.2. Khi Tag ở vùng phần ba thứ hai



Hình 34. Tag ở vùng phần ba thứ hai

Từ hình, ta có:

$$\frac{x - x_1}{y} = \tan(-\alpha) = -\tan(\alpha)$$

$$\frac{x_2 - x}{y} = \tan(\beta)$$

Ta có hệ phương trình:

$$x + y * \tan(\alpha) = x_1$$

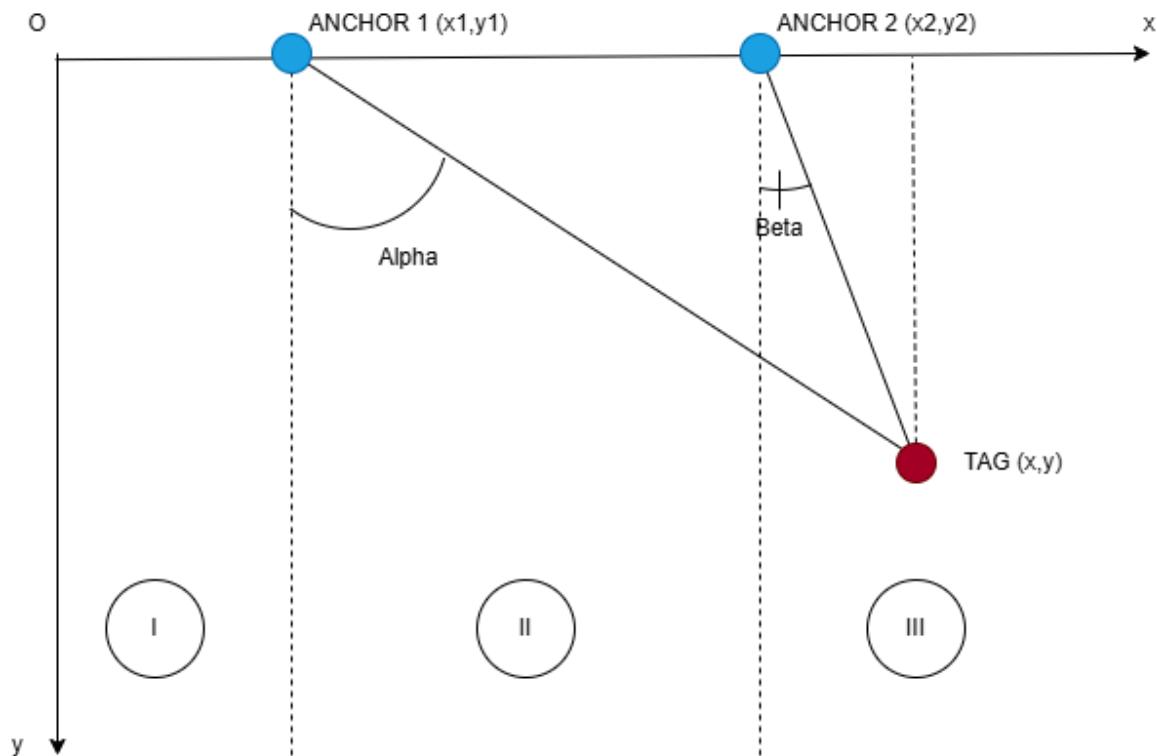
$$x + y * \tan(\beta) = x_2$$

Giải hệ phương trình, ta được

$$x = \frac{x_1 * \tan(\alpha) - x_2 * \tan(\beta)}{\tan(\beta) - \tan(\alpha)}$$

$$y = \frac{x_2 - x_1}{\tan(\beta) - \tan(\alpha)}$$

2.3.3. Khi Tag ở vùng phần ba thứ ba



Hình 35. Tag ở vùng phần ba thứ ba

Từ hình, ta có:

$$\frac{x_1 - x}{y} = \tan(-\alpha) = -\tan(\alpha)$$

$$\frac{x_2 - x}{y} = \tan(-\beta) = -\tan(\beta)$$

Ta có hệ phương trình:

$$x + y * \tan(\alpha) = x_1$$

$$x + y * \tan(\beta) = x_2$$

Giải hệ phương trình, ta được

$$x = \frac{x_1 * \tan(\alpha) - x_2 * \tan(\beta)}{\tan(\beta) - \tan(\alpha)}$$

$$y = \frac{x_2 - x_1}{\tan(\beta) - \tan(\alpha)}$$

➔ Vậy ở cả ba vùng, ta đều có:

$$x = \frac{x_1 * \tan(\alpha) - x_2 * \tan(\beta)}{\tan(\beta) - \tan(\alpha)}; y = \frac{x_2 - x_1}{\tan(\beta) - \tan(\alpha)}$$

CHƯƠNG 4: THỰC HIỆN ĐỀ TÀI

1. Thiết kế và thực hiện Hardware

1.1. Giới thiệu phần mềm KICAD

KiCad là một bộ phần mềm thiết kế điện tử, bao gồm các công cụ để thiết kế sơ đồ nguyên lý, bố trí board mạch in, và tạo file Gerber để sản xuất PCB. Phần mềm này được phát triển bởi một cộng đồng mã nguồn mở, với sự hỗ trợ từ CERN – tổ chức nghiên cứu khoa học nổi tiếng thế giới.

Tính năng của KiCad:

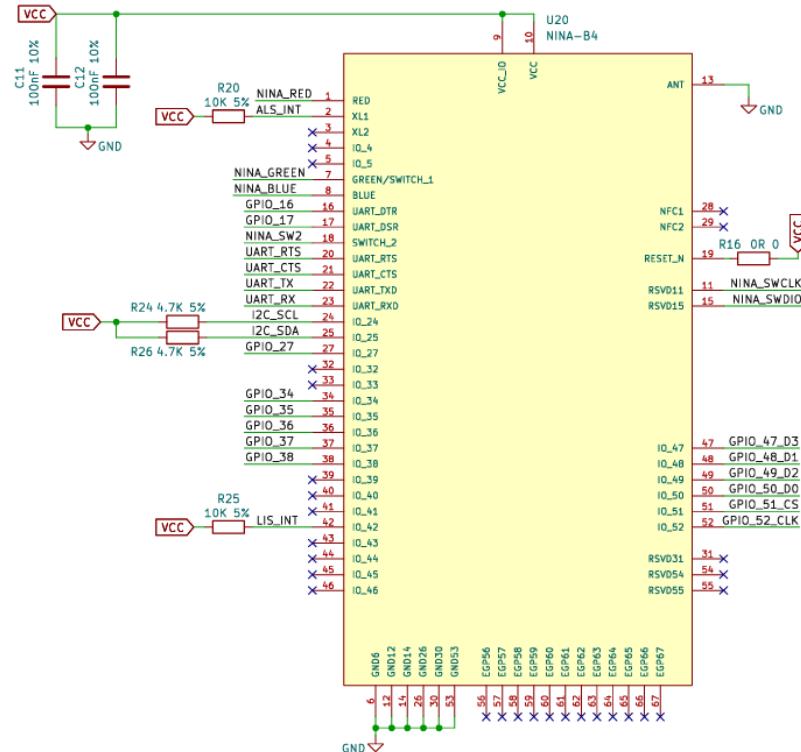
- Thiết kế sơ đồ nguyên lý (Schematic Capture): KiCad cung cấp một môi trường thiết kế sơ đồ nguyên lý trực quan, hỗ trợ hàng ngàn linh kiện điện tử trong thư viện. Người dùng cũng có thể tự tạo linh kiện tùy chỉnh.
- Hỗ trợ sản xuất: KiCad dễ dàng xuất các file Gerber, Drill, và BOM (Bill of Materials) để gửi đến nhà sản xuất PCB.
- Thư viện phong phú và tùy chỉnh: KiCad đi kèm thư viện khổng lồ gồm các linh kiện phổ biến. Người dùng cũng có thể tùy chỉnh và tạo thư viện riêng để đáp ứng nhu cầu cá nhân.

Lý do chọn KiCad cho đề tài:

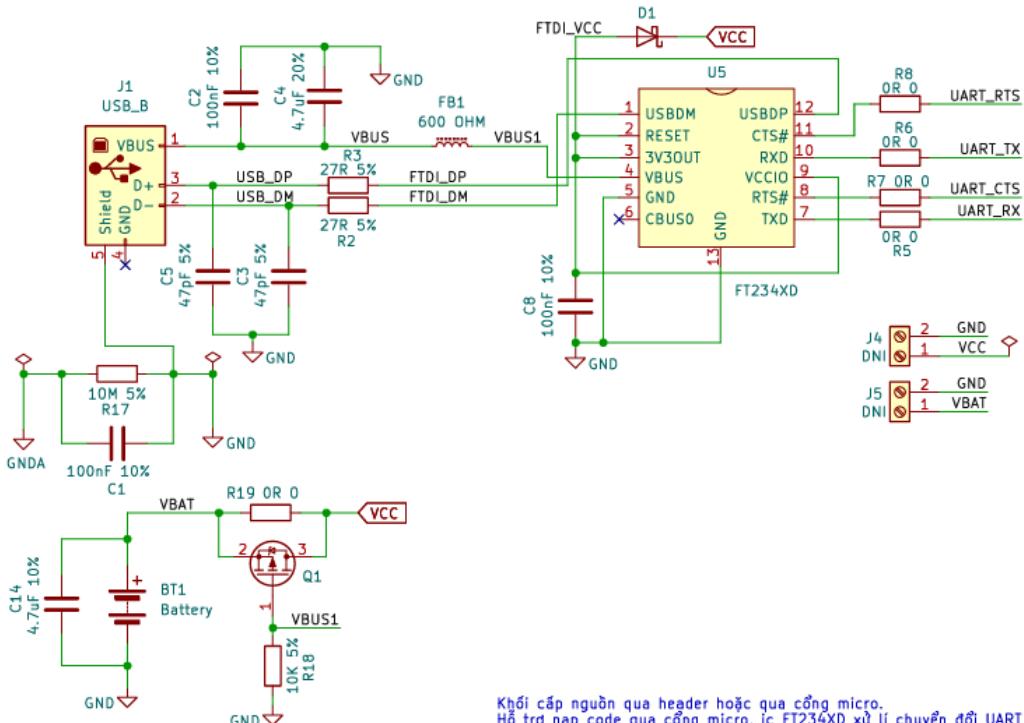
- Miễn phí hoàn toàn: Không yêu cầu chi phí bản quyền, rất phù hợp cho sinh viên, người mới học và cả doanh nghiệp nhỏ.
- Mã nguồn mở: Người dùng có thể tùy chỉnh phần mềm theo nhu cầu hoặc tham gia đóng góp cải thiện phần mềm.
- Hỗ trợ cộng đồng: KiCad có một cộng đồng người dùng lớn, sẵn sàng chia sẻ kiến thức, tài nguyên và giải đáp thắc mắc.

1.2. Thực hiện Hardware

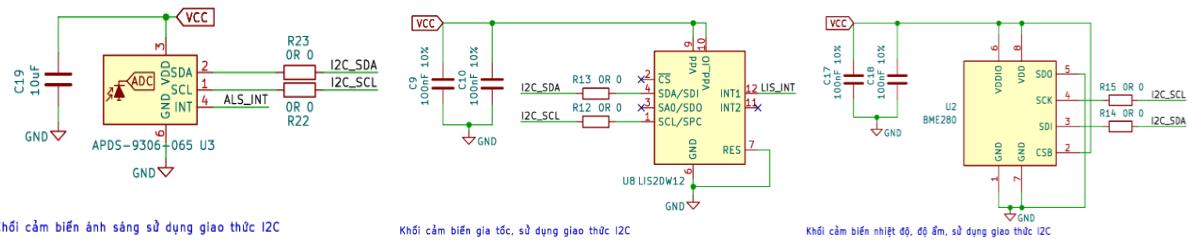
1.2.1. Schematic



Hình 36. Khối xử lý trung tâm IC NINA-B406

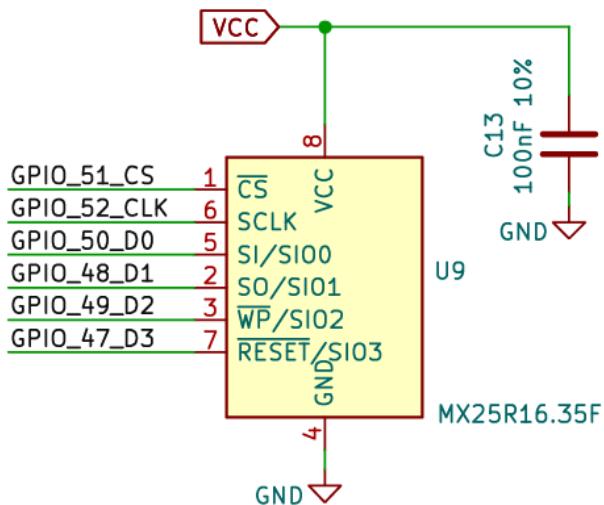


Hình 37. Khối cấp nguồn và chuyển đổi UART



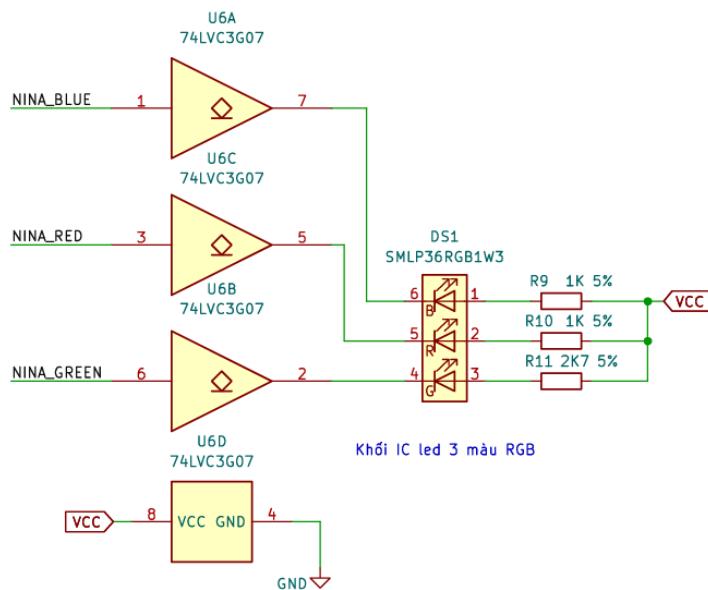
Hình 38. Khối các cảm biến

Ngoài ra còn có một số khối khác:



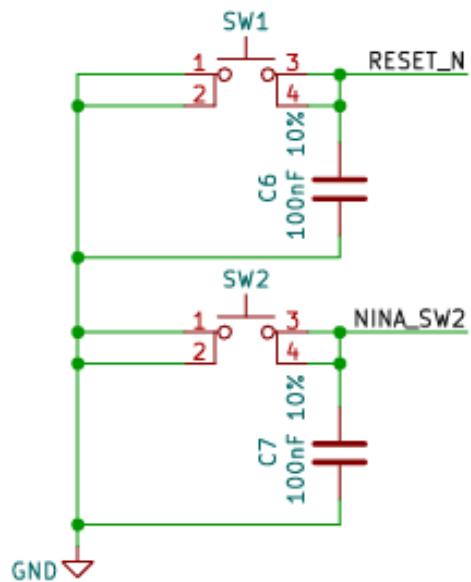
Khối IC nhớ (flash), dùng để lưu trữ các thông số cấu hình. Sử dụng giao thức SPI.

Hình 39. Khối IC nhớ (FLASH)



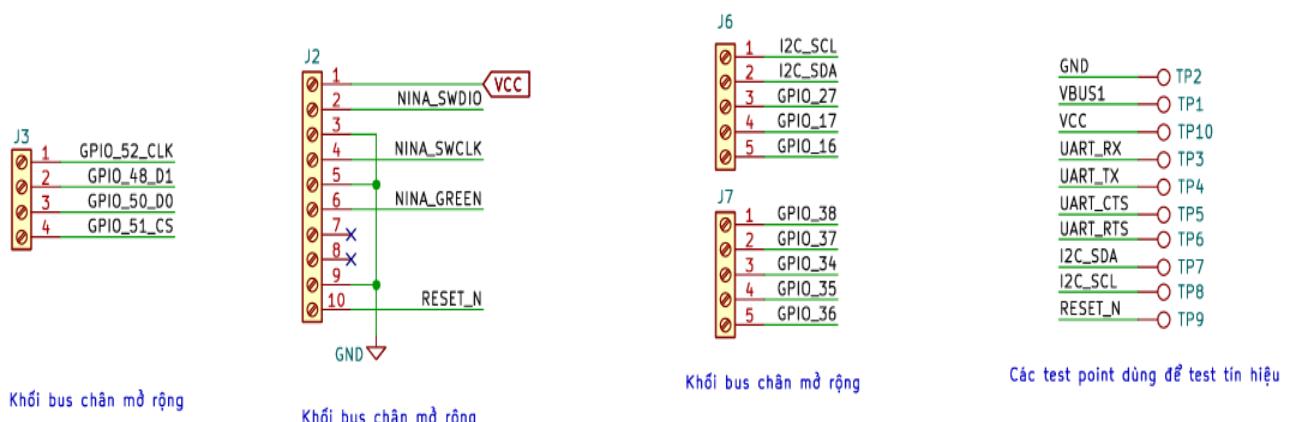
Khối IC đèn dòng để điều khiển tải lớn.

Hình 40. Khối đèn trạng thái



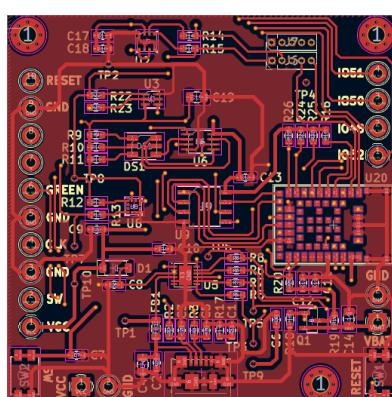
Nút nhấn reset và điều khiển

Hình 41. Khối nút nhấn RESET và điều khiển

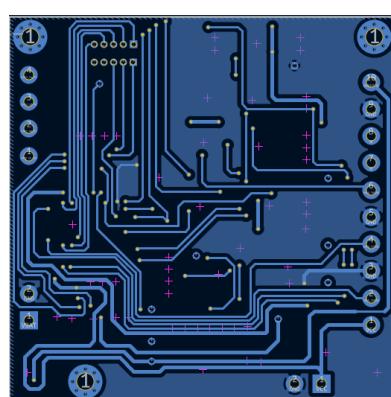


Hình 42. Khối các chân mở rộng

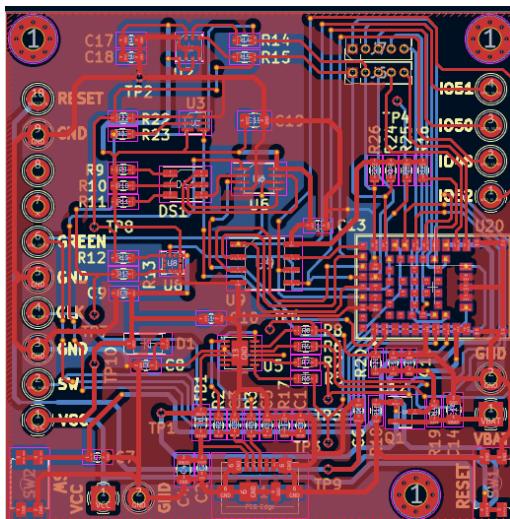
1.2.2. Thiết kế mạch in



Hình 43. Mặt trên PCB (TOP Layer)

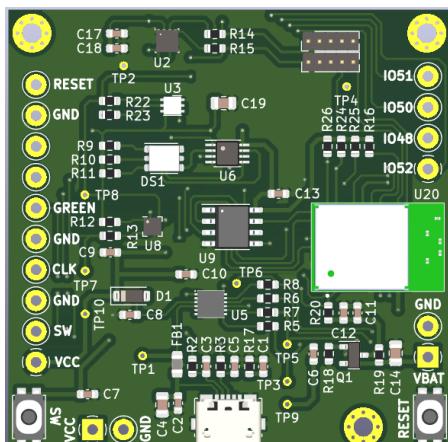


Hình 44. Mặt dưới PCB (BOT Layer)

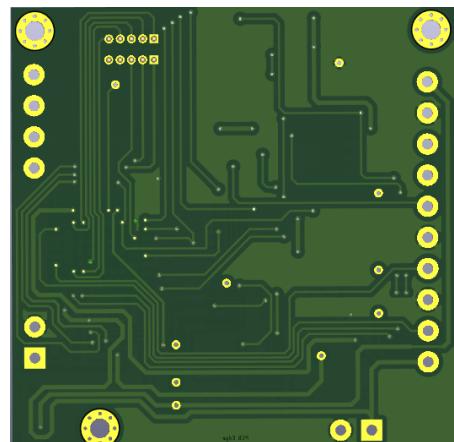


Hình 45. Tổng hợp cấu trúc đa lớp

1.2.3. Bố trí 3D PCB của module C209 Tag

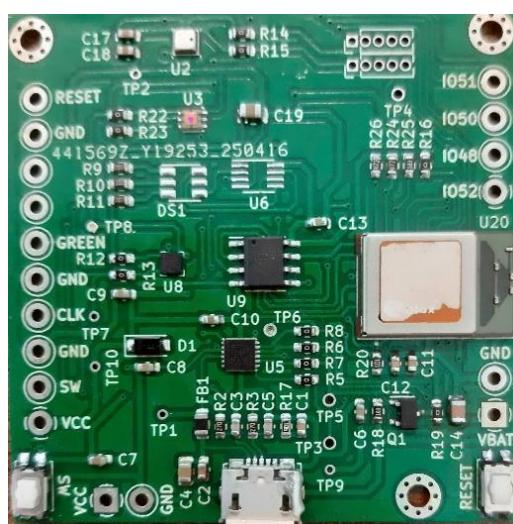


Hình 46. Biểu diễn không gian mặt trước



Hình 47. Biểu diễn không gian mặt sau

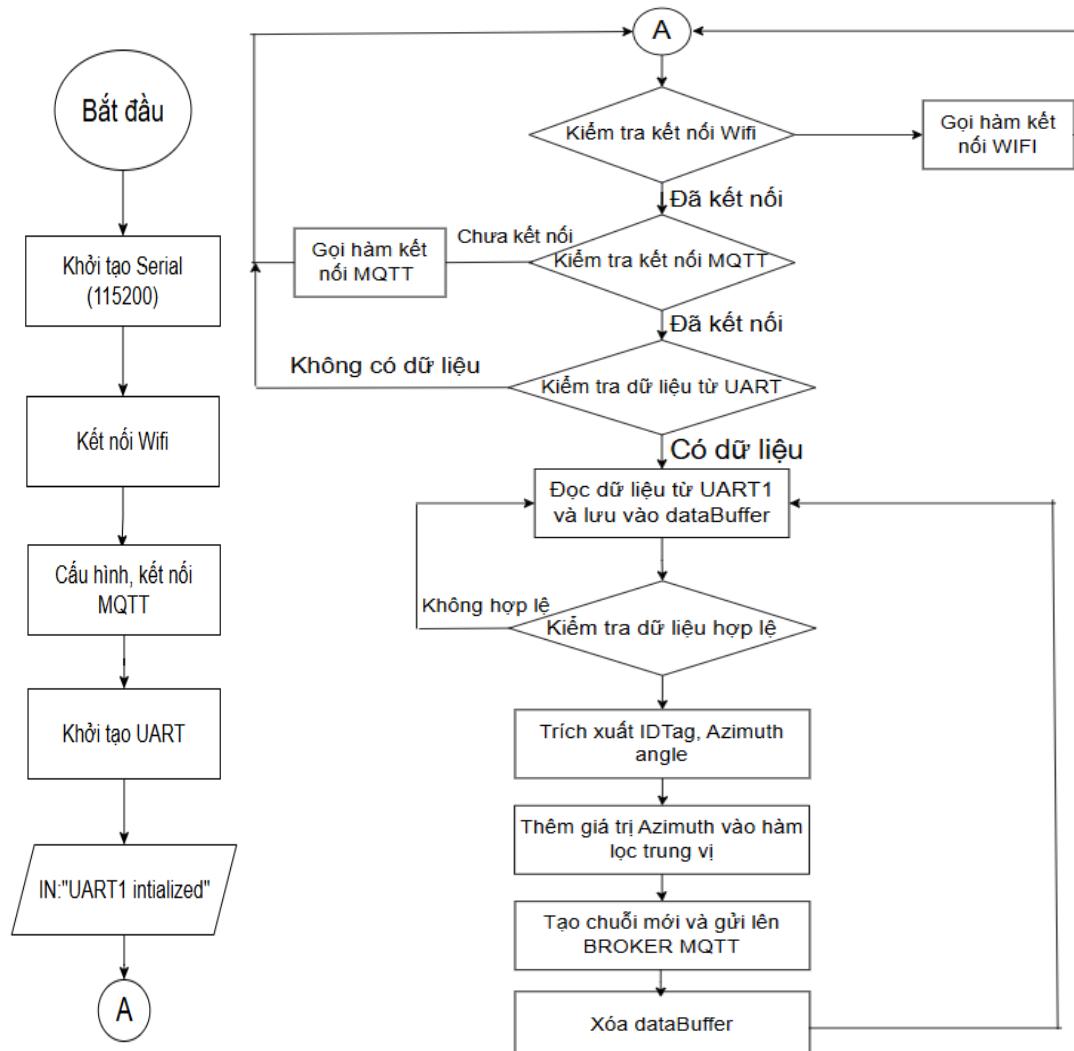
1.2.4. Kết quả



Hình 48. Sản phẩm thực tế

2. Thiết kế và thực hiện Firmware

2.1. Lưu đồ giải thuật



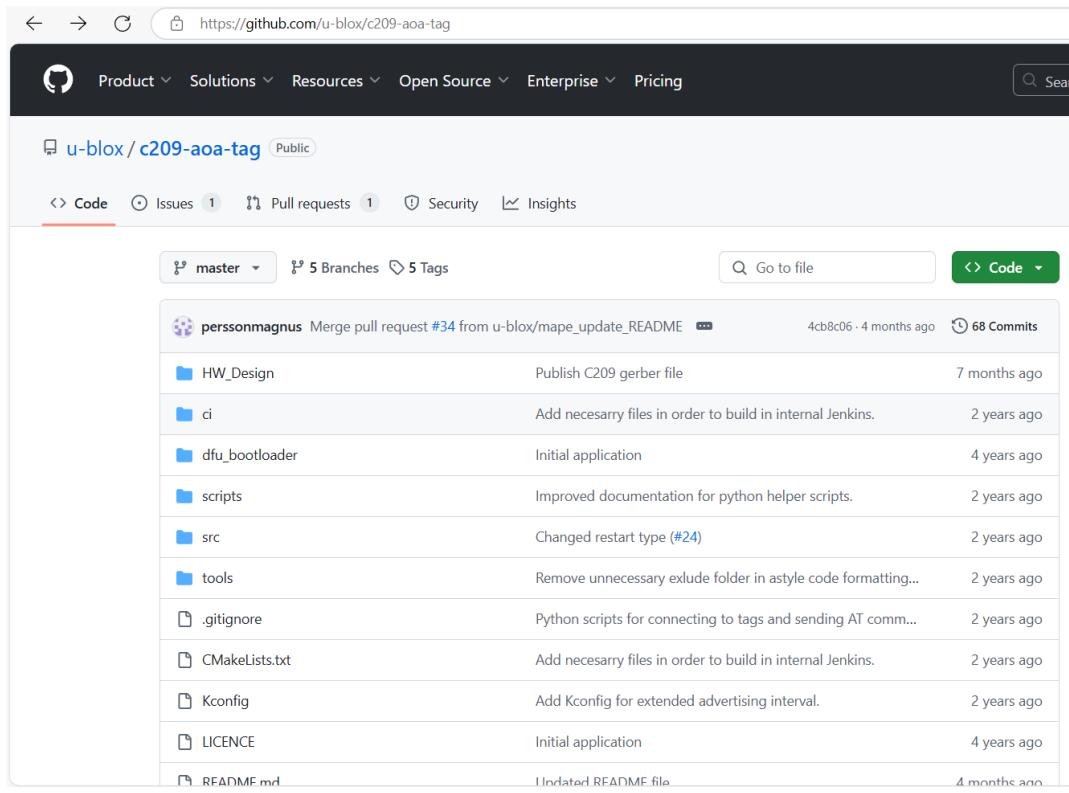
Hình 49. Lưu đồ giải thuật vi điều khiển

2.2. Thực hiện Firmware

2.2.1. Nạp firmware cho Tag

Để tiến hành nạp firmware cho Tag, ta sử dụng source (mã nguồn) do nhà sản xuất cung cấp và đã được biên dịch thành mã nhị phân.

Trước hết, truy cập vào Github của nhà sản xuất để lấy firmware cho Tag theo đường dẫn sau: *GitHub - u-blox/c209-aoa-tag: Bluetooth Direction Finding Tag sample application. Works with u-connectLocate.*



Hình 50. Github của nhà sản xuất.

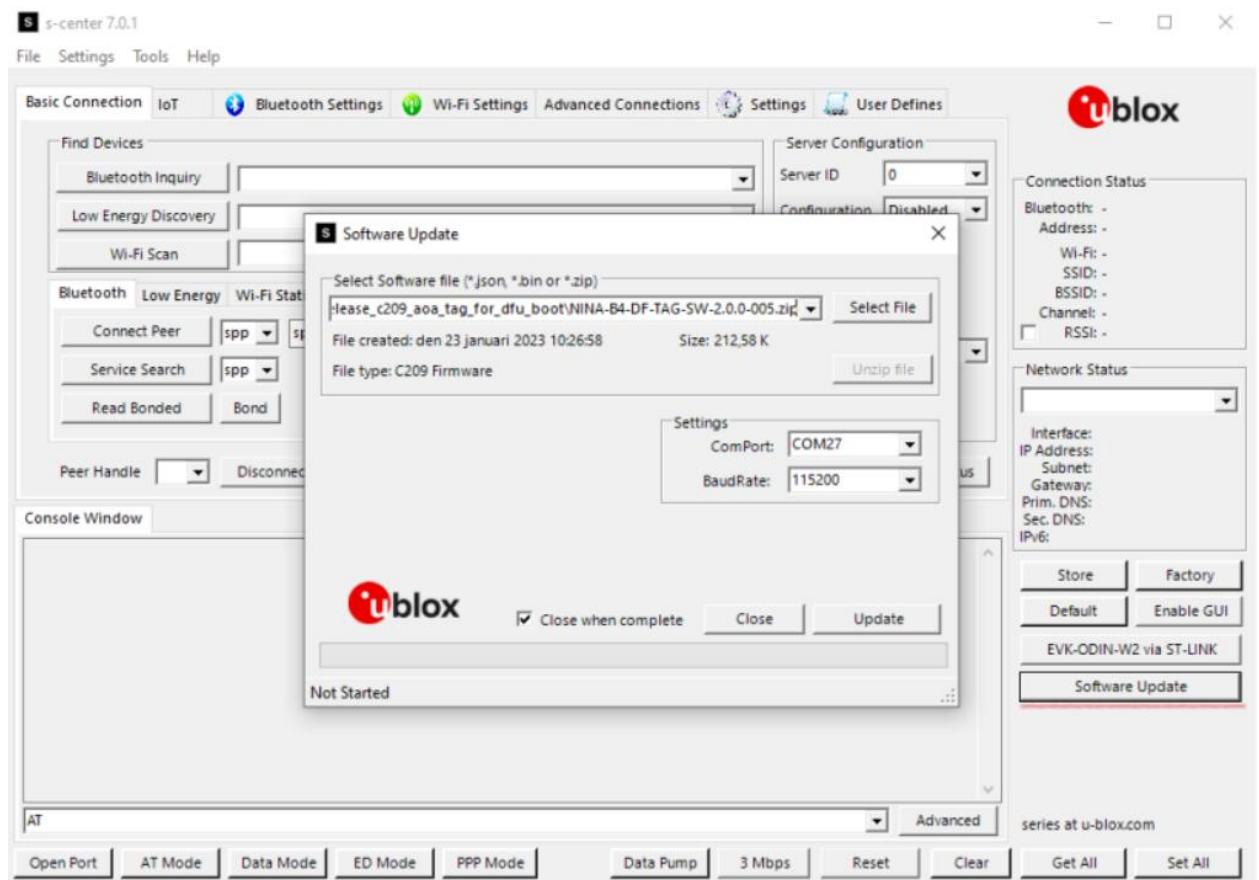
Nạp firmware theo hướng dẫn trong tài liệu của nhà sản xuất.

The C209 tags can be flashed over the UART using s-center [7].

Flashing requires s-center 6.1 or later.

Follow the flashing procedure outlined below when using s-center:

1. Download and unpack the `c209_aoa_tag_<version>.zip` file from the u-blox C209 software repository using this [direct link](#), or go to the "Releases" section in the main GitHub repository page [12]. The unpacked file includes the `NINA-B4-DF-TAG-SW-<version>.zip` file used for the installation.
2. Press and hold the SW2 button on the C209 while resetting the board (by inserting the USB cable or clicking the RESET button) to set the bootloader in "download" mode. See also [Figure 24](#).
3. Open s-center on the correct COM port once the module is in the software download mode.
4. Select **Software Update**.
5. In the "Software Update" dialog, select the `NINA-B4-DF-TAG-SW-<version>.zip` file from the software container, as shown in [Figure 25](#). Make sure the flashing speed is set to 115200, and that you select the zip from a folder with "for_dfu_boot" in the name.
6. Press **Update**.



Hình 51. Hướng dẫn nạp firmware cho Tag¹

Các bước nạp firmware:

1. Tải phần mềm S-Center.
2. Tải tệp c209_aoa_tag_2_0_1-001 ở Github của nhà sản xuất.
3. Nhấn và giữ nút SW2 để đưa cụm bootloader của Tag vào chế độ “Download”.
4. Mở phần mềm s-center. Chọn mục “Software Update”, chọn file “NINA-B4-DF-TAG-SW-2.0.1-001.zip” trong tệp firmware đã tải, lưu ý phải chọn tệp nén trong tệp “for_dfu_boot”.
5. Chọn tốc độ baudrate là 115200, cổng COM phù hợp và nhấn “Update”
6. Nhấn nút Reset của Tag để tiến hành nạp chương trình.

2.2.2. Firmware ESP32

- Khai báo thư viện

```
#include <WiFi.h>
#include <PubSubClient.h>
```

¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”. Truy cập từ: [Product resources | u-blox](#)

Sử dụng 2 thư viện chính WiFi.h để kết nối wifi và PubSubClient.h để kết nối MQTT.

- Kết nối wifi

```
void connectWiFi() {
    Serial.print("Connecting to WiFi...");
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(100);
        if (millis() - previousWiFiCheckTime > 10000) {
            Serial.println("\nWiFi failed, retrying...");
            return;
        }
    }
    Serial.println("\nWiFi connected. IP: " + WiFi.localIP().toString());
}
```

- Khởi tạo Broker và các hàm kết nối MQTT

```
const char* MQTT_SERVER = "192.168.0.102";
const int MQTT_PORT = 1883;
const char* MQTT_TOPIC_PUB = "cuong151";
const char* MQTT_TOPIC_SUB = "receive";
```

```
void setup() {
    Serial.begin(115200);
    connectWiFi();
    mqttClient.setServer(MQTT_SERVER, MQTT_PORT);
    mqttClient.setCallback(callback);
    mqttClient.setKeepAlive(15);
```

- Hàm kết nối lại MQTT nếu kết nối thất bại

```
void reconnectMQTT() {
    while (!mqttClient.connected()) {
        Serial.print("Attempting MQTT connection...");
        String clientId = "ESP32Client-" + String(random(0xffff), HEX);
        if (mqttClient.connect(clientId.c_str())) {
            Serial.println("Connected to MQTT");
            mqttClient.subscribe(MQTT_TOPIC_SUB);
        } else {
            Serial.print("Failed, rc=");
            Serial.print(mqttClient.state());
            Serial.println(" Retry in 2 seconds...");
            delay(2000);
        }
    }
}
```

- Khởi tạo và hàm nhận dữ liệu từ UART

```
#define RX_PIN 16
#define TX_PIN 17
#define BAUD_RATE 115200
String dataBuffer = "";
HardwareSerial MySerial(1);

bool isValidData(const String& data) {
    if (!data.startsWith("+UUDF:")) return false;

    int commaCount = 0;
    for (char c : data) {
        if (c == ',') commaCount++;
    }
    return commaCount >= 9;
}

void loop() {
    if (WiFi.status() != WL_CONNECTED) {
        connectWiFi();
    }

    if (!mqttClient.connected()) {
        reconnectMQTT();
    }
    mqttClient.loop();
    while (MySerial.available()) {
        char c = MySerial.read();
        dataBuffer += c;

        if (c == '\n') {
            if (isValidData(dataBuffer)) {
                processData(dataBuffer);
            }
            dataBuffer = "";
        }
    }
}
```

- Hàm xử lý dữ liệu và gửi lên Broker

```
void processData(String& dataBuffer) {
    int startIndex = dataBuffer.indexOf(":") + 1;
    int firstCommaIndex = dataBuffer.indexOf(",", startIndex);
    String deviceID = dataBuffer.substring(startIndex, firstCommaIndex);

    String values = dataBuffer.substring(startIndex);

    int commaIndex1 = values.indexOf(',');
    int commaIndex2 = values.indexOf(',', commaIndex1 + 1);
    int commaIndex3 = values.indexOf(',', commaIndex2 + 1);
    int commaIndex4 = values.indexOf(',', commaIndex3 + 1);

    a = values.substring(commaIndex2 + 1, commaIndex3).toInt();
    e = values.substring(commaIndex3 + 1, commaIndex4).toInt();

    addToBuffer(aBuffer, a, aBufferIndex);
    addToBuffer(eBuffer, e, eBufferIndex);
```

```

int medianA = getMedian(aBuffer);
int medianE = getMedian(eBuffer);

String filteredData = "AC2," + deviceID + "," + String(medianA) + "," +
String(medianE);
mqttClient.publish(MQTT_TOPIC_PUB, filteredData.c_str());
Serial.println("Sent valid data via MQTT: " + filteredData); }

```

- Bộ lọc Median giảm nhiễu dữ liệu gửi đi
 - Tạo một mảng tạm sorted có kích thước WINDOW_SIZE.
 - Sao chép dữ liệu từ buffer vào sorted bằng memcpy.
 - Sắp xếp mảng sorted theo thứ tự tăng dần bằng std::sort.
 - Trả về giá trị ở giữa của mảng đã sắp xếp.

```

void addToBuffer(int* buffer, int value, int& bufferIndex) {
    buffer[bufferIndex] = value;
    bufferIndex = (bufferIndex + 1) % WINDOW_SIZE;
}

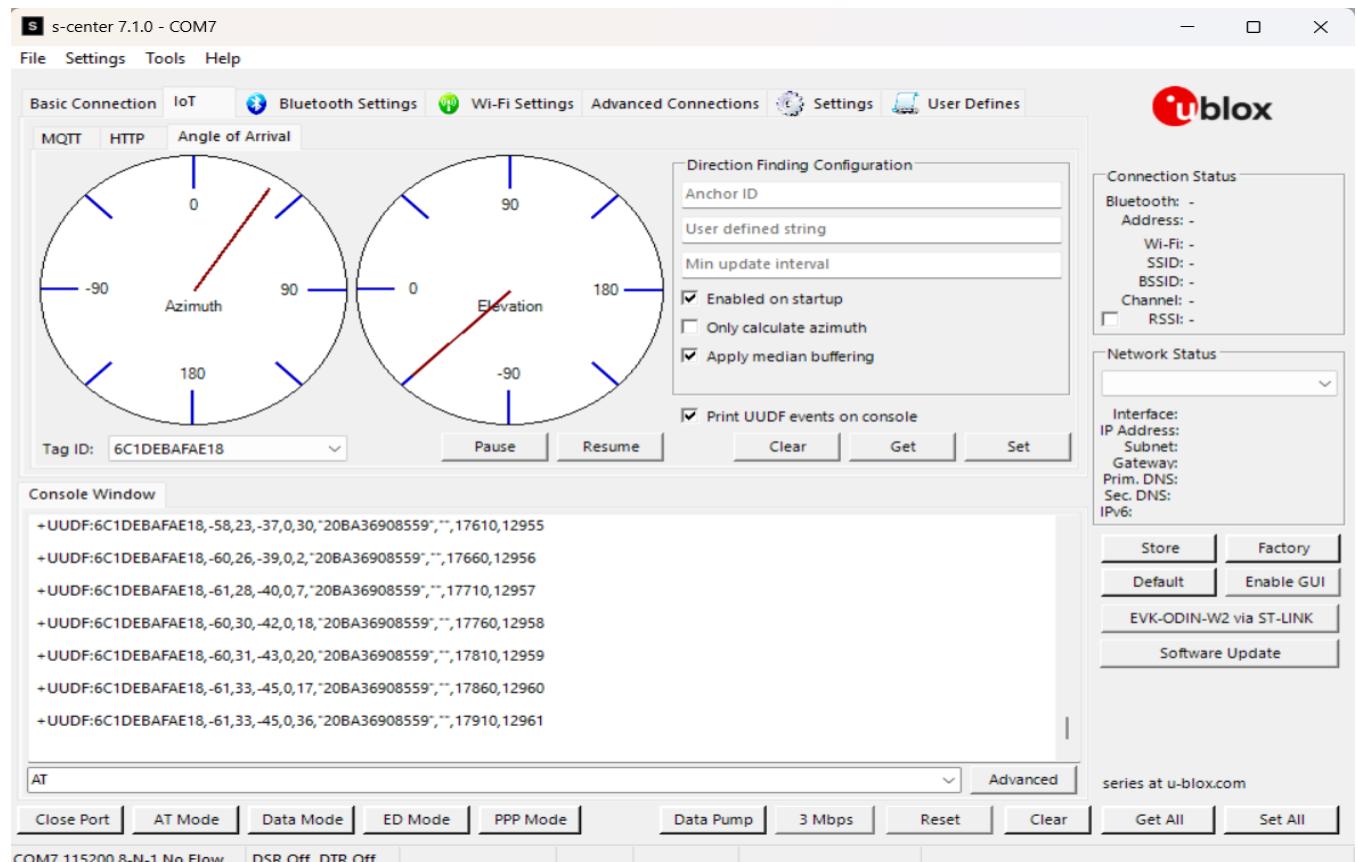
int getMedian(int* buffer) {
    int sorted[WINDOW_SIZE];
    memcpy(sorted, buffer, WINDOW_SIZE * sizeof(int));
    std::sort(sorted, sorted + WINDOW_SIZE);
    return sorted[WINDOW_SIZE / 2];
}

```

2.3. Kết quả

2.3.1. Kết quả nạp chương trình cho Tag

Sau khi cài đặt firmware cho Kit và Tag tiến hành thu dữ liệu ta có kết quả sau:



Hình 52. Kết quả thực tế thu được từ phần mềm u-located¹

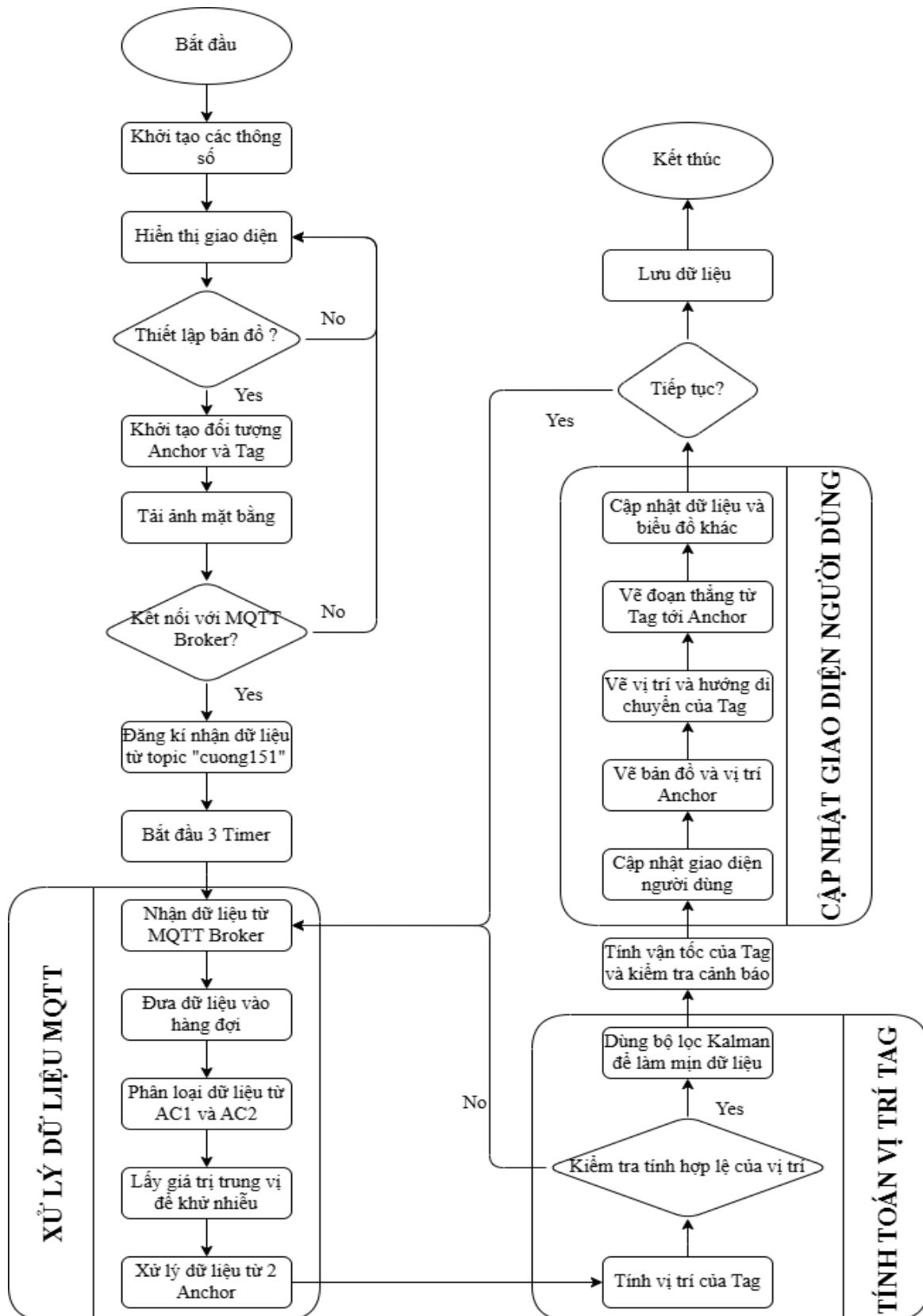
Trong đó ta lưu ý đến các giá trị quan trọng mà kit thu thập được đó là:

- RSSI: -61 (dB).
- Azimuth angel: 33 độ.
- Elevation angel: -45 độ.

¹ Ublox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits Bluetooth direction finding User guide”.
Truy cập từ: [Product resources | u-blox](#)

3. Thiết kế và thực hiện Software

3.1. Lưu đồ giải thuật



Hình 53. Lưu đồ giải thuật của Software

3.2. Giải thích các hàm quan trọng

3.2.1. Hàm thiết lập và kết nối

- **ParametersSetup()**

```
private void ParametersSetup()
{
    try
    {
        Tags.Clear();
        TagCount = Convert.ToInt16(tbTagcount.Text);
        for (int i = 0; i < TagCount; i++) { Tag tag = new Tag(); Tags.Add(tag);
    }
    label18.Text = "Anchors: 2 Online";
    NameTagID();
    Anchor_1.x = (float)Convert.ToDouble(Anchor1x.Text);
    Anchor_1.y = (float)Convert.ToDouble(Anchor1y.Text);
    Anchor_2.x = (float)Convert.ToDouble(Anchor2x.Text);
    Anchor_2.y = (float)Convert.ToDouble(Anchor2y.Text);
    W = (float)Convert.ToDouble(Width.Text);
    H = (float)Convert.ToDouble(Height.Text);
    scalex = pnlMap.Width / W;
    scaley = pnlMap.Height / H;
    // Update DataGridView when tag count changes
    InitializeTagRows();
}
catch (Exception ex)
{
    MessageBox.Show("Please set up the map before starting.", "Notification",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
```

Đây là hàm khởi tạo các tham số cơ bản của hệ thống. Nó đọc thông tin về số lượng Tag, vị trí của các Anchor, và kích thước của bản đồ từ giao diện người dùng. Hàm này thiết lập tỷ lệ tỷ lệ giữa kích thước thực và kích thước hiển thị trên giao diện.

- **ConnectToMQTT()**

```
private void ConnectToMQTT()
{
    try
    {
        string brokerAddress = tbIP.Text.Trim().ToString();//"192.168.0.100"; // IP address of MQTT broker
        MqttClient.BrokerAddress = brokerAddress;
        MqttClient.Connect("PC");
        MqttClient.DataReceived += OnMqttDataReceived;
        MqttClient.Subscribe("cuong151");
        mqttConnected = true;
        MessageBox.Show("Successfully connected to MQTT broker!", "Notification",
    }
```

```
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    Console.WriteLine($"CONNECTED {brokerAddress}");
}
catch (Exception ex)
{
    MessageBox.Show($"MQTT connection error: {ex.Message}", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    mqttConnected = false;
}
```

Thiết lập kết nối với MQTT broker (máy chủ MQTT) thông qua địa chỉ IP được chỉ định. Sau khi kết nối thành công, nó đăng ký (subscribe) đến chủ đề (topic) “cuong151” cần thiết để nhận dữ liệu về vị trí từ 2 Anchor.

- **OnMqttDataReceived()**

```
private void OnMqttDataReceived(object sender, MqttEventArgs e)
{
    string message = e.Message;
    Console.WriteLine($"Received MQTT: {message}"); // Add log for debug

    if (message.StartsWith("AC1"))
    {
        lock (queueLock)
        {
            pendingAC1Messages.Add(message);
            if (pendingAC1Messages.Count > 2) pendingAC1Messages.RemoveAt(0);
            dataAnchor1 = message;
        }
    }
    else if (message.StartsWith("AC2"))
    {
        lock (queueLock)
        {
            pendingAC2Messages.Add(message);
            if (pendingAC2Messages.Count > 2) pendingAC2Messages.RemoveAt(0);
            dataAnchor2 = message;
        }
    }
}
```

Hàm callback được gọi khi có dữ liệu mới từ MQTT broker. Nó phân loại dữ liệu thành dữ liệu từ Anchor 1 hoặc Anchor 2 và lưu trữ trong các danh sách tương ứng để xử lý sau đó.

3.2.2. Các hàm xử lý dữ liệu vị trí

- **ProcessMessagePair()**

```

private void ProcessMessagePair(string messageAC1, string messageAC2)
{
    try
    {
        // Parse messages
        string[] partsAC1 = messageAC1.Split(',');
        string[] partsAC2 = messageAC2.Split(',');

        // Check message format
        if (partsAC1.Length < 3 || partsAC2.Length < 3)
            return;

        // Check if both messages have the same tagID
        if (partsAC1[1] != partsAC2[1])
            return;

        string tagId = partsAC1[1];

        // Extract azimuth angles
        float azimuth1 = float.Parse(partsAC1[2], CultureInfo.InvariantCulture);
        float azimuth2 = float.Parse(partsAC2[2], CultureInfo.InvariantCulture);

        // Find corresponding tag
        foreach (Tag tag in Tags)
        {
            if (tag.tagID == tagId)
            {
                // Calculate position from azimuth angles
                CalculatePositionImproved(tag, azimuth1, azimuth2);

                // Mark as having new data to update UI
                partsAC1 = null;
                partsAC2 = null;
                hasNewData = true;
                // Update tag connection status
                UpdateTagStatus(tagId);

                break;
            }
        }
    }
    catch (Exception ex)
    {
        // Handle exception safely
        Console.WriteLine($"Error processing message pair: {ex.Message}");
    }
}

```

Xử lý cặp tin nhắn từ cả hai Anchor và tính toán vị trí của Tag. Nó kiểm tra xem tin nhắn có cùng tagID không, trích xuất góc azimuth và gọi hàm tính toán vị trí.

- **CalculatePositionImproved()**

```

private void CalculatePositionImproved(Tag tag, float Azimuth1, float Azimuth2)
{
    float x1 = Anchor_1.x;
    float y1 = Anchor_1.y;
    float x2 = Anchor_2.x;
    float y2 = Anchor_2.y;

    // Convert degrees to radians

    float Azimuth1_Radian = Azimuth1 * ((float)Math.PI / 180);
    float Azimuth2_Radian = Azimuth2 * ((float)Math.PI / 180);
    float Tan_Azimuth1 = (float)Math.Tan(Azimuth1_Radian);
    float Tan_Azimuth2 = (float)Math.Tan(Azimuth2_Radian);

    // Check division by zero
    if (Math.Abs(Tan_Azimuth2 - Tan_Azimuth1) < 0.00001f)
        return;

    // Correct equation for intersection point
    float newX = (x1 * Tan_Azimuth2 - x2 * Tan_Azimuth1) / (Tan_Azimuth2 -
    Tan_Azimuth1);
    float newY = (x2 - x1) / (Tan_Azimuth2 - Tan_Azimuth1);
    if (Math.Abs(newX - tag.x) < 0.1 && Math.Abs(newY - tag.y) < 0.1)
    {
        // Ignore changes that are too small
        newX = tag.x;
        newY = tag.y;
    }
    // Calculate slope of line from each Anchor

    // Check division by zero

    // Calculate intersection coordinates (correct formula)

    // Update time for velocity calculation
    DateTime currentTime = DateTime.Now;
    double timeDelta = (currentTime - tag.LastUpdateTime).TotalSeconds;

    // Update velocity if not first time
    if (tag.LastUpdateTime != DateTime.MinValue && timeDelta > 0)
    {
        UpdateVelocity(tag, newX, newY, timeDelta);
    }

    // Apply Kalman filter or adaptive filter
    ApplyPositionFilter(tag, newX, newY);

    // Update time
    tag.LastUpdateTime = currentTime;
}

```

```

// Update distances
distToAnchor1 = CalculateDistance(tag.x, tag.y, x1, y1);
distToAnchor2 = CalculateDistance(tag.x, tag.y, x2, y2); }
```

Hàm quan trọng nhất để tính toán vị trí của Tag dựa trên các góc azimuth nhận được từ cả hai Anchor. Nó sử dụng kỹ thuật tam giác đo (triangulation) để xác định giao điểm của hai đường thẳng từ các Anchor, từ đó xác định vị trí của Tag.

Công thức tính toán chính:

```

float newX = (x1 * Tan_Azimuth2 - x2 * Tan_Azimuth1) / (Tan_Azimuth2 - Tan_Azimuth1);
float newY = (x2 - x1) / (Tan_Azimuth2 - Tan_Azimuth1);
```

- **ApplyPositionFilter()**

```

private void ApplyPositionFilter(Tag tag, float newX, float newY)
{
    try
    {
        // Initialize filter if first time
        if (tag.lastFilteredX == 0 && tag.lastFilteredY == 0)
        {
            tag.filterX.SetInitialValue(newX);
            tag.filterY.SetInitialValue(newY);
            tag.lastFilteredX = newX;
            tag.lastFilteredY = newY;

            // Initialize list with first value
            tag.recentX = new List<float> { newX, newX, newX };
            tag.recentY = new List<float> { newY, newY, newY };

            tag.targetX = newX;
            tag.targetY = newY;
            return;
        }

        // Apply Kalman filter
        float filteredX = tag.filterX.Update(newX);
        float filteredY = tag.filterY.Update(newY);

        // Save filtered value
        tag.lastFilteredX = filteredX;
        tag.lastFilteredY = filteredY;

        // Check for outliers
        if (tag.recentX.Count >= 3)
        {
            if (IsOutlier(filteredX, filteredY, tag))
            {
                // Skip outlier point
                return;
            }
        }
    }
}
```

```

        // Add to recent values list
        tag.recentX.Add(filteredX);
        tag.recentY.Add(filteredY);
        if (tag.recentX.Count > 5) tag.recentX.RemoveAt(0);
        if (tag.recentY.Count > 5) tag.recentY.RemoveAt(0);

        // Median filter if enough data
        if (tag.recentX.Count >= 3)
        {
            tag.targetX = tag.recentX.OrderBy(x =>
x).ElementAt(tag.recentX.Count / 2);
            tag.targetY = tag.recentY.OrderBy(y =>
y).ElementAt(tag.recentY.Count / 2);
        }
        else
        {
            // Not enough data, use filtered value directly
            tag.targetX = filteredX;
            tag.targetY = filteredY;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"ApplyPositionFilter Error: {ex.Message}");
    }
}

```

Áp dụng bộ lọc Kalman và bộ lọc trung vị để loại bỏ nhiễu và làm mịn dữ liệu vị trí. Điều này giúp giảm thiểu sự dao động và tăng độ chính xác của vị trí hiển thị.

- **UpdateVelocity()**

```

private void UpdateVelocity(Tag tag, float newX, float newY, double timeDelta)
{
    // Calculate velocity for each axis
    tag.velocityX = (newX - tag.lastFilteredX) / (float)timeDelta;
    tag.velocityY = (newY - tag.lastFilteredY) / (float)timeDelta;

    // Calculate total velocity (magnitude)
    tag.velocityMagnitude = (float)Math.Sqrt(
        Math.Pow(tag.velocityX, 2) + Math.Pow(tag.velocityY, 2));
    // Add log for debug

    // Save velocity to history for analysis
    if (tag.velocityHistory == null)
        tag.velocityHistory = new RollingPointPairList(1000); // Save 100 most
recent points

    tag.velocityHistory.Add(DateTime.Now.ToOADate(), tag.velocityMagnitude);

    // Limit velocity to avoid abnormal values
}

```

```

        float maxVelocity = 50.0f; // Units per second
        tag.velocityX = Math.Max(-maxVelocity, Math.Min(maxVelocity, tag.velocityX));
        tag.velocityY = Math.Max(-maxVelocity, Math.Min(maxVelocity, tag.velocityY));
        tag.velocityMagnitude = Math.Min(maxVelocity, tag.velocityMagnitude);
    }

```

Tính toán vận tốc của Tag dựa trên sự thay đổi vị trí theo thời gian. Nó tính cả vận tốc theo trục X, trục Y và vận tốc tổng (magnitude) để sử dụng cho việc cảnh báo và hiển thị.

- **UpdateSmoothPositions()**

```

private void UpdateSmoothPositions()
{
    foreach (Tag tag in Tags)
    {
        try
        {
            // If tag not initialized, skip
            if (tag.x == 0 && tag.y == 0 && tag.targetX == 0 && tag.targetY == 0)
                continue;

            // Initialize previousX/Y if first time
            if (tag.previousX == 0 && tag.previousY == 0)
            {
                tag.previousX = tag.x;
                tag.previousY = tag.y;
            }

            // Check if tag is stationary
            float movement = (float)Math.Sqrt(
                Math.Pow(tag.x - tag.previousX, 2) +
                Math.Pow(tag.y - tag.previousY, 2));

            if (movement < 0.1f) // If movement < 2mm
            {
                tag.stabilityCounter++;
                if (tag.stabilityCounter > 2) // After 10 cycles
                {
                    tag.isStable = true;
                }
            }
            else
            {
                tag.stabilityCounter = 0;
                tag.isStable = false;
            }

            // Calculate distance to target position
            float dx = tag.targetX - tag.x;
            float dy = tag.targetY - tag.y;
        }
    }
}

```

```

    // Prevent tag from standing still at start
    if (Math.Abs(dx) < 0.001 && Math.Abs(dy) < 0.001 && tag.x == tag.targetX
&& tag.targetX == 0)
    {
        // Tag has no position yet, don't update
        continue;
    }

    // If tag is stationary and movement is small, maintain position
    if (tag.isStationary && Math.Abs(dx) < 0.5f && Math.Abs(dy) < 0.5f)
    {
        // Don't update position or update very slightly
        tag.x += dx * 0.01f;
        tag.y += dy * 0.01f;
        continue;
    }

    // Limit movement
    float maxMovement = 1.0f; // 5mm per update
    if (Math.Abs(dx) > maxMovement) dx = Math.Sign(dx) * maxMovement;
    if (Math.Abs(dy) > maxMovement) dy = Math.Sign(dy) * maxMovement;

    // Update position with smoothing factor
    tag.x += dx * tag.smoothingFactor;
    tag.y += dy * tag.smoothingFactor;

    // Save current position to check stability
    tag.previousX = tag.x;
    tag.previousY = tag.y;
}

catch (Exception ex)
{
    Console.WriteLine($"UpdateSmoothPositions Error: {ex.Message}");
}
}
}
}

```

Cập nhật vị trí mềm của các Tag bằng cách áp dụng các thuật toán làm mịn. Nó giúp chuyển động của Tag trên giao diện trở nên mượt mà hơn, thay vì nhảy đột ngột giữa các vị trí.

3.2.3. Các hàm hiển thị và cảnh báo

- **pnlMap_Paint()**

```

private void pnlMap_Paint(object sender, PaintEventArgs e)
{
    if (isSetup)
    {
        Graphics g = e.Graphics;
    }
}

```

```

    // Draw the bitmap if it exists
    if (mapBitmap != null)
    {
        g.DrawImage(mapBitmap, 0, 0, pnlMap.Width, pnlMap.Height);
    }
    else
    {
        using (Pen pen = new Pen(Color.Blue, 2))
        {
            // Draw a rectangle with top-left corner at origin (0,0)
            g.DrawRectangle(pen, 0, 0, W * scalex, H * scaley);

            // Optional: Fill the rectangle with color
            using (SolidBrush brush = new SolidBrush(Color.FromArgb(100, 0, 0, 255)))
            {
                g.FillRectangle(brush, 0, 0, W * scalex, H * scaley);
            }
        }

        DrawPoint(g, $"Anchor1({Anchor_1.x},{Anchor_1.y})", Anchor_1.x,
        Anchor_1.y, Color.Red);
        DrawPoint(g, $"Anchor2({Anchor_2.x},{Anchor_2.y})", Anchor_2.x,
        Anchor_2.y, Color.Red);
        if (isConnect)
        {
            foreach (Tag tag in Tags)
            {
                TagFunction(g, tag);
            }
        }
    }
}

```

Hàm vẽ chính của ứng dụng, được gọi mỗi khi cần cập nhật giao diện. Nó vẽ bản đồ nền, vị trí của các Anchor, vị trí và đường đi của các Tag, cũng như các đường kết nối giữa Anchor và Tag.

- **DrawTrail()**

```

private void DrawTrail(Graphics g, Tag tag, Color color, float thickness =
3)
{
    if (tag.TrailPoints.Count < 2) return;

    // Only draw n newest points, e.g. 10-20 points
    int pointsToShow = Math.Min(1000, tag.TrailPoints.Count);
    int startIndex = tag.TrailPoints.Count - pointsToShow;

    PointF[] screenPoints = new PointF[pointsToShow];

```

```

for (int i = 0; i < pointsToShow; i++)
{
    screenPoints[i] = new PointF(
        tag.TrailPoints[startIndex + i].X * scalex,
        tag.TrailPoints[startIndex + i].Y * scaley
    );
}

using (Pen pen = new Pen(Color.FromArgb(150, color), thickness))
{
    g.DrawLines(pen, screenPoints);
}

```

Vẽ đường đi (trail) của các Tag trên bản đồ, giúp hiển thị quỹ đạo chuyển động của Tag theo thời gian.

- **CheckForAlarms()**

```

private void CheckForAlarms()
{
    foreach (Tag tag in Tags)
    {
        bool hadAlarmBefore = tag.HasAlarm;
        AlarmType previousAlarm = tag.CurrentAlarm;

        // Reset alarm status
        tag.HasAlarm = false;
        tag.AlarmMessage = "";
        tag.CurrentAlarm = AlarmType.None;

        // Check velocity alarm (if tag moves too fast)
        if (tag.velocityMagnitude > tag.velocityThreshold)
        {
            tag.HasAlarm = true;
            tag.AlarmMessage = $"High velocity: {tag.velocityMagnitude:F2}"
> {tag.velocityThreshold:F2}";
            tag.CurrentAlarm = AlarmType.VelocityTooHigh;
            tag.AlarmTime = DateTime.Now;

            // If it's a new alarm or different from previous, add to
            datagridview
            if (!hadAlarmBefore || previousAlarm !=
AlarmType.VelocityTooHigh)
            {
                AddAlarmToDataGridView(tag,
tag.velocityMagnitude.ToString("F2"));
            }
        }
        // Check connection alarm
        else if (!tag.IsConnected)
        {

```

```

        tag.HasAlarm = true;
        tag.AlarmMessage = "Connection lost";
        tag.CurrentAlarm = AlarmType.ConnectionLost;
        tag.AlarmTime = DateTime.Now;

        // If it's a new alarm or different from previous, add to
        datagridview
        if (!hadAlarmBefore || previousAlarm != 
        AlarmType.ConnectionLost)
        {
            AddAlarmToDataGridView(tag, "N/A");
        }
    }
    // Check out of bounds alarm
    else if (tag.x < 0 || tag.x > W || tag.y < 0 || tag.y > H)
    {
        tag.HasAlarm = true;
        tag.AlarmMessage = $"Out of bounds: ({tag.x:F2}, {tag.y:F2})";
        tag.CurrentAlarm = AlarmType.OutOfBounds;
        tag.AlarmTime = DateTime.Now;

        // If it's a new alarm or different from previous, add to
        datagridview
        if (!hadAlarmBefore || previousAlarm != AlarmType.OutOfBounds)
        {
            AddAlarmToDataGridView(tag, $"({tag.x:F2}, {tag.y:F2})");
        }
    }
}

```

Kiểm tra các điều kiện có thể dẫn đến cảnh báo, bao gồm: vận tốc cao bất thường, mất kết nối, và Tag di chuyển ra ngoài phạm vi cho phép.

- **SaveTrailData()** và **SaveAlarmsToCSV()**

```

private void SaveTrailData(Tag tag)
{
    try
    {
        using (SaveFileDialog saveFileDialog = new SaveFileDialog())
        {
            saveFileDialog.Filter = "CSV Files (*.csv)|*.csv|All Files (*.*)|*.*";
            saveFileDialog.Title = "Save Trail Data";
            saveFileDialog.DefaultExt = "csv";
            saveFileDialog.FileName =
$"TagTrail_{tag.tagID}_{DateTime.Now:yyyyMMdd_HHmmss}.csv";

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            Using (StreamWriter writer = new
StreamWriter(saveFileDialog.FileName, false, Encoding.UTF8))

```

```

    {
        // Header
        writer.WriteLine("ID,Seq,X,Y,Time,Distance to Anchor1,Distance
to Anchor2");

        // Data
        for (int i = 0; i < tag.TrailPoints.Count; i++)
        {
            PointF point = tag.TrailPoints[i];

            // Calculate distance from point to anchors
            distToAnchor1 = CalculateDistance(point.X, point.Y,
Anchor_1.x, Anchor_1.y);
            distToAnchor2 = CalculateDistance(point.X, point.Y,
Anchor_2.x, Anchor_2.y);

            // Write data for each point
            writer.WriteLine($"{tag.ToString()},{i +
1},{point.X},{point.Y},{" +
 $"{DateTime.Now.AddSeconds(-tag.TrailPoints.Count +
i).ToString("yyyy-MM-dd HH:mm:ss.fff")}, {" +
 $"{distToAnchor1:F2},{distToAnchor2:F2}");
        }
    }

    MessageBox.Show("Trail data has been saved successfully.", "Notification",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}

}

catch (Exception ex)
{
    MessageBox.Show($"Error saving trail data: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

```

private void SaveAlarmsToCSV()
{
    try
    {
        using (SaveFileDialog saveFileDialog = new SaveFileDialog())
        {
            saveFileDialog.Filter = "CSV Files (*.csv)|*.csv|All Files (*.*)|*.*";
            saveFileDialog.Title = "Save Alarm Log";
            saveFileDialog.DefaultExt = "csv";
            saveFileDialog.FileName =
$"Alarm_Log_{DateTime.Now:yyyyMMdd_HHmmss}.csv";
    }
}

```

```

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            using (StreamWriter writer = new
StreamWriter(saveFileDialog.FileName, false, Encoding.UTF8))
            {
                // Write header
                writer.WriteLine("Time,Tag ID,Alarm Type,Message,Value");

                // Write data from DataGridView
                foreach (DataGridViewRow row in dgvAlarms.Rows)
                {
                    string time = row.Cells["Time"].Value?.ToString() ?? "";
                    string tagID = row.Cells["TagID"].Value?.ToString() ?? "";
                    string alarmType = row.Cells["AlarmType"].Value?.ToString() ?? "";
                    string message = row.Cells["Message"].Value?.ToString() ?? "";
                    string value = row.Cells["Value"].Value?.ToString() ?? "";

                    // Ensure valid CSV data
                    message = message.Replace(", ", ";");

                    writer.WriteLine($"{time},{tagID},{alarmType},{message},{value}");
                }
            }
            MessageBox.Show("Alarm log saved successfully!", "Notification",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error saving alarm log: {ex.Message}", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Lưu trữ dữ liệu đường đi và cảnh báo ra file CSV để phân tích sau này.

3.2.4. Timer

- Monitor Timer

```

private void MonitorTimer_Tick(object sender, EventArgs e)
{
    if (isSetup && isConnected)
    {
        UpdateSmoothPositions();
        pnlMap.Invalidate();
        Anchor1.Text = $"Anchor1: ({Anchor_1.x},{Anchor_1.y})";
    }
}

```

```

Anchor2.Text = $"Anchor2: ({Anchor_2.x},{Anchor_2.y})";
// number of tags changed

label16.Text = "System Status: ONLINE";
label19.Text = "Last Update: " + DateTime.Now.ToString("HH:mm:ss
dd/MM/yyyy");

// Add tag status check
CheckTagStatus();
// Check for alarms
CheckForAlarms();
// Update DataGridView
UpdateTagsDataGridView();
// Update velocity graph
UpdateVelocityGraph();
UpdateIconPositions();

}

else
{
    label16.Text = "System Status: OFFLINE";
}
}

```

MonitorTimer đóng vai trò như "nhiệt tim" của ứng dụng. Nó chịu trách nhiệm:

- Cập nhật vị trí mượt mà của các Tag
- Làm mới giao diện người dùng
- Kiểm tra trạng thái kết nối của các Tag
- Phát hiện cảnh báo (vận tốc cao, mất kết nối, vượt ra khỏi vùng)
- Cập nhật bảng dữ liệu và biểu đồ

Timer này đảm bảo người dùng luôn nhìn thấy thông tin mới nhất và các cảnh báo được phát hiện kịp thời.

- **Simulation Timer**

```

private void SimulationTimer_Tick(object sender, EventArgs e)
{
    if (isSetup && isConnected)
    {
        foreach (Tag tag in Tags)
        {
            UpdateTrail(tag);
        }
    }
}

```

ProcessingTimer hoạt động như một "bộ xử lý dữ liệu" trong nền:

- Kiểm tra hàng đợi tin nhắn từ các Anchor
- Sử dụng thuật toán lọc trung vị để loại bỏ dữ liệu nhiễu
- Xử lý cặp tin nhắn từ cả hai Anchor để tính toán vị trí
- Đảm bảo quá trình xử lý dữ liệu không bị treo nếu xảy ra lỗi

Timer này rất quan trọng vì nó tách biệt quá trình xử lý dữ liệu từ giao diện người dùng, giúp ứng dụng chạy mượt mà hơn.

- **Processing Timer**

```
private void ProcessingTimer_Tick(object sender, EventArgs e)
{
    try
    {
        if (pendingAC1Messages.Count > 0 && pendingAC2Messages.Count > 0)
        {
            // Get median value from each Anchor
            string medianAC1 = GetMedianMessage(pendingAC1Messages);
            string medianAC2 = GetMedianMessage(pendingAC2Messages);

            if (medianAC1 != null && medianAC2 != null)
            {
                // Process this pair
                ProcessMessagePair(medianAC1, medianAC2);

                // Clear all processed messages
                lock (queueLock)
                {
                    pendingAC1Messages.Clear();
                    pendingAC2Messages.Clear();
                }
            }
            else
            {
                // Fallback method: Process data as before
                ProcessMessageBatch();
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"ProcessingTimer Error: {ex.Message}");
        }
    }
}
```

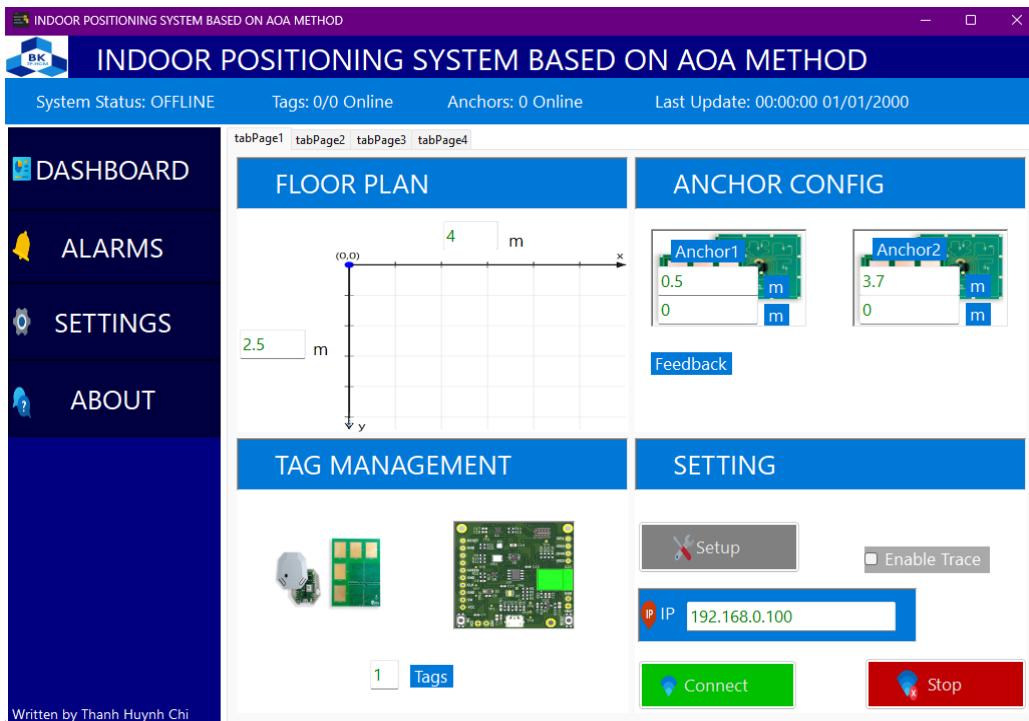
ProcessingTimer hoạt động như một "bộ xử lý dữ liệu" trong nền. Nó:

- Kiểm tra hàng đợi tin nhắn từ các Anchor
- Sử dụng thuật toán lọc trung vị để loại bỏ dữ liệu nhiễu
- Xử lý cặp tin nhắn từ cả hai Anchor để tính toán vị trí

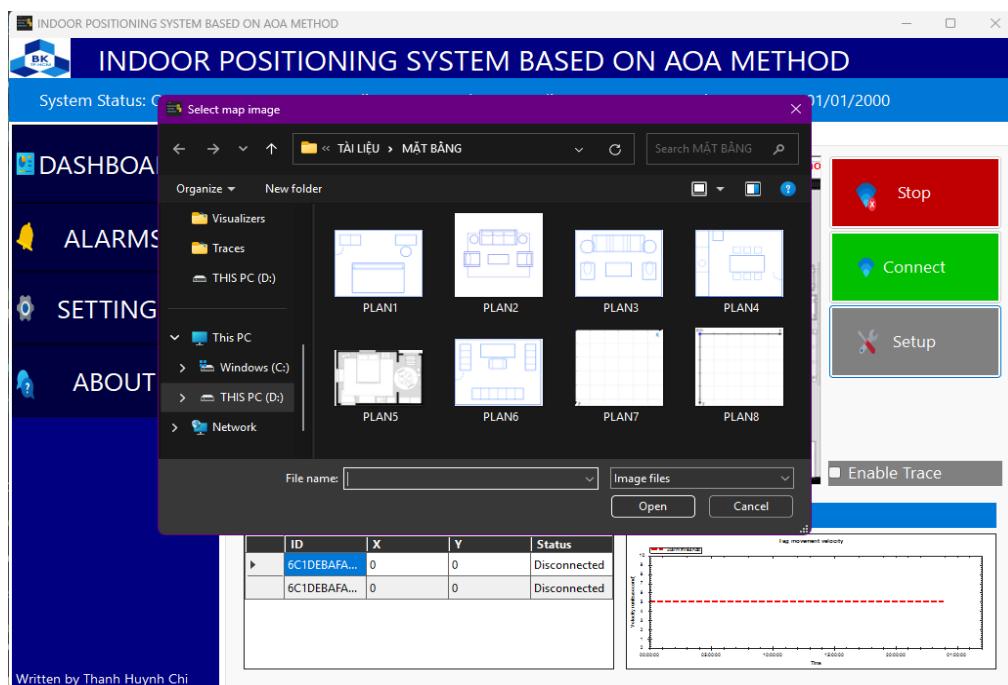
- Đảm bảo quá trình xử lý dữ liệu không bị treo nếu xảy ra lỗi

Timer này rất quan trọng vì nó tách biệt quá trình xử lý dữ liệu từ giao diện người dùng, giúp ứng dụng chạy mượt mà hơn.

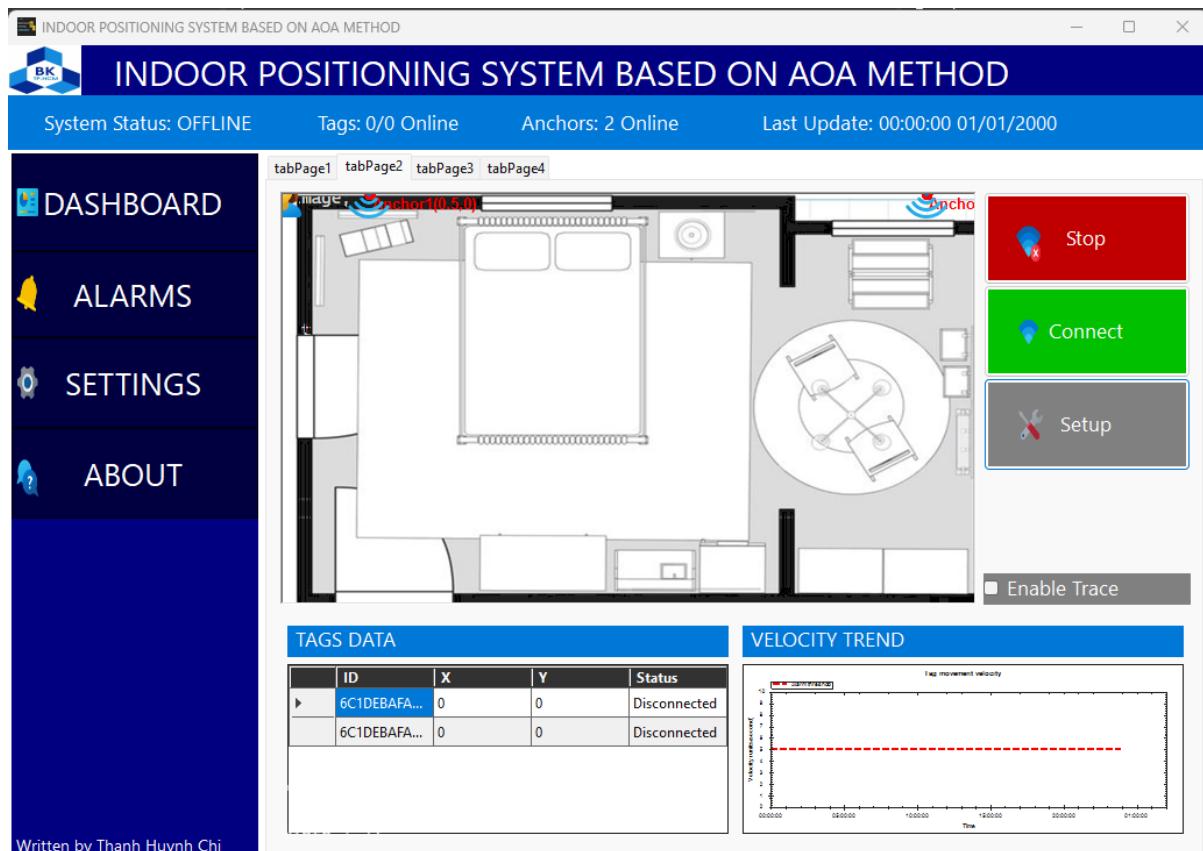
3.3. Kết quả



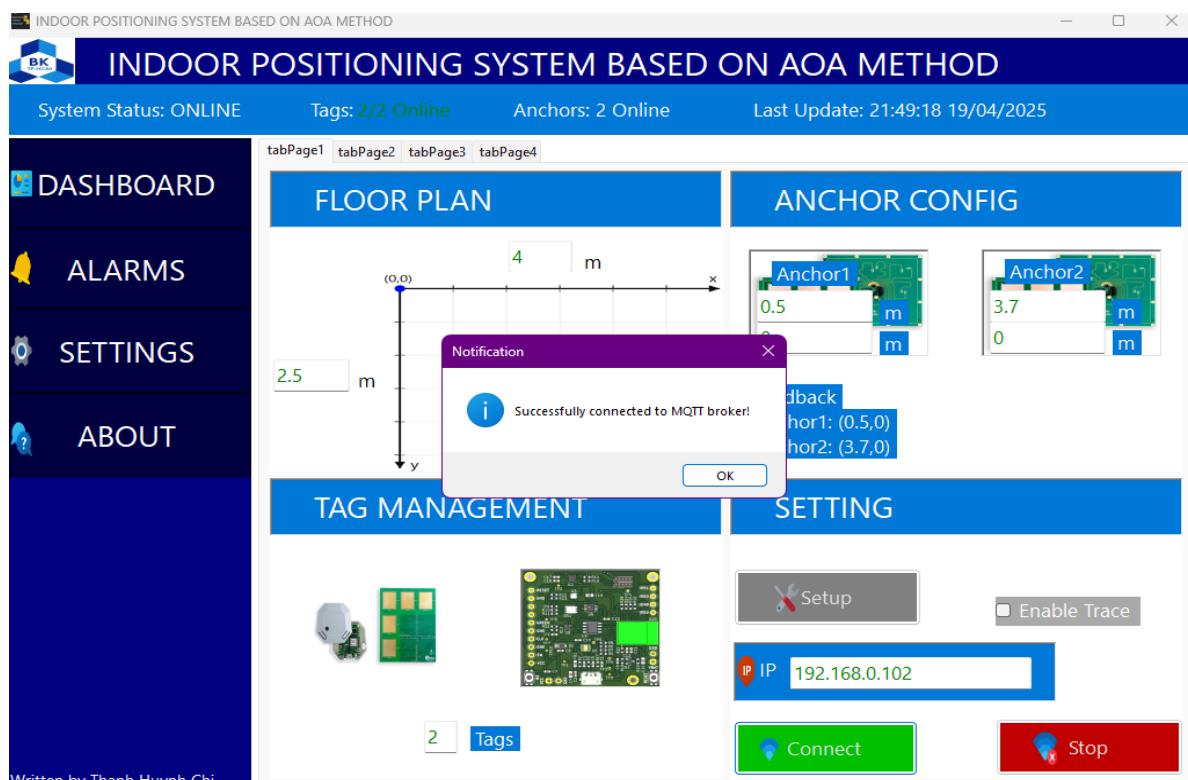
Hình 54. Giao diện khởi động ứng dụng



Thiết lập mặt bằng (Kích thước, mặt cắt 2D) và các thông số cần thiết (IP, số lượng Anchor, vị trí Anchor, số lượng Tag).



Sau khi thiết lập xong, giao diện Dashboard theo dõi đã sẵn sàng để kết nối MQTT Broker.



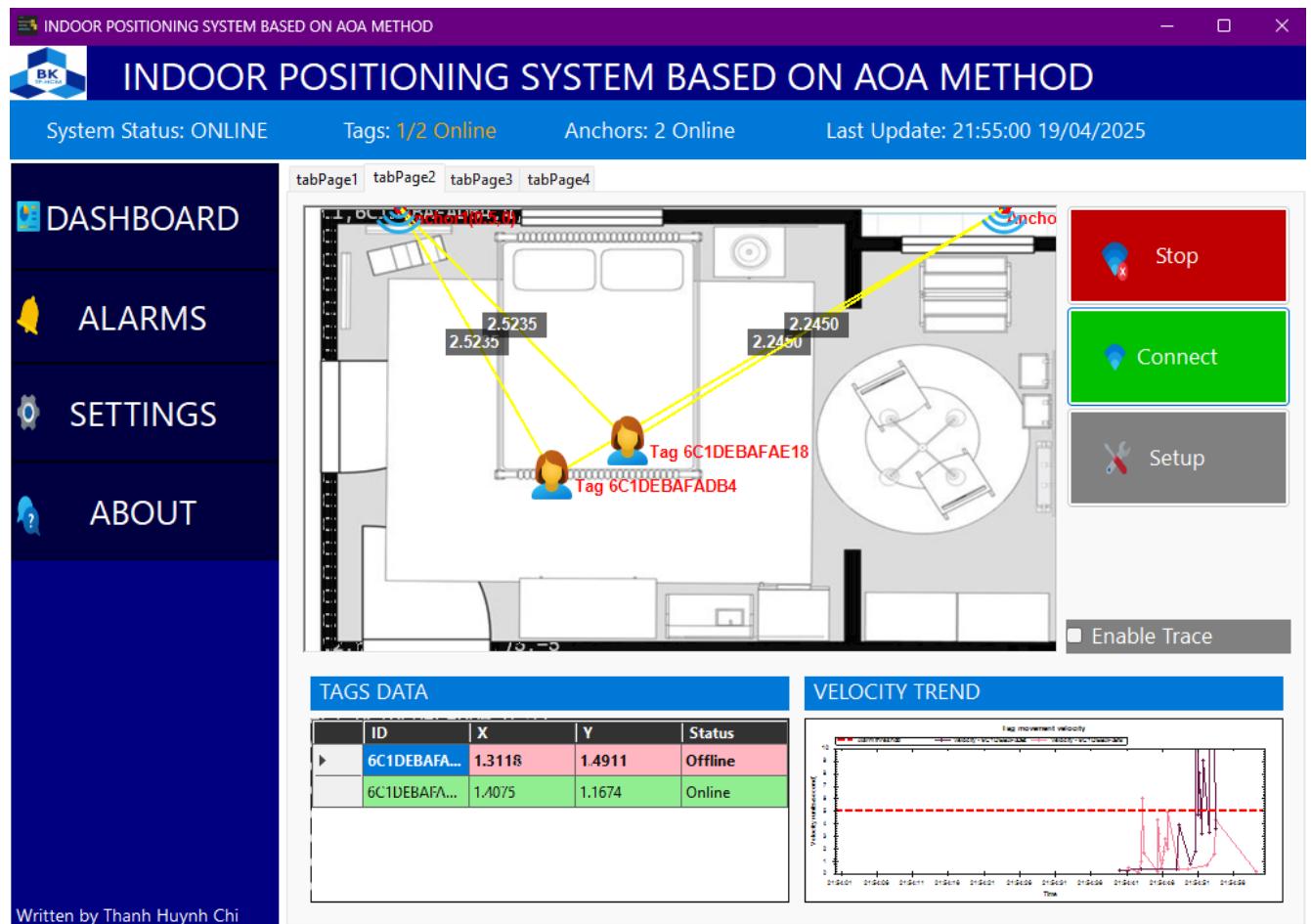
Hình 55. Kết nối thành công tới MQTT Broker

```

Received MQTT: AC1,6C1DEBAFAE18,-45,25 Trail tracking disabled for 6C1DEBAFADB4
Received MQTT: AC1,6C1DEBAFADB4,75,-39 Trail tracking disabled for 6C1DEBAFAE18
Received MQTT: AC1,6C1DEBAFAE18,-44,26 Trail tracking disabled for 6C1DEBAFADB4
Received MQTT: AC1,6C1DEBAFADB4,75,-39 Trail tracking disabled for 6C1DEBAFAE18
Received MQTT: AC1,6C1DEBAFAE18,-39,26 Trail tracking disabled for 6C1DEBAFADB4
Received MQTT: AC1,6C1DEBAFADB4,75,-39 Trail tracking disabled for 6C1DEBAFAE18
Received MQTT: AC1,6C1DEBAFAE18,-39,26 Trail tracking disabled for 6C1DEBAFADB4
Received MQTT: AC1,6C1DEBAFAE18,-39,25 Received MQTT: AC2,6C1DEBAFADB4,59,-13
Received MQTT: AC1,6C1DEBAFADB4,75,-39 Received MQTT: AC2,6C1DEBAFAE18,75,35
Received MQTT: AC1,6C1DEBAFAE18,-39,25 Received MQTT: AC2,6C1DEBAFADB4,59,-13
Received MQTT: AC1,6C1DEBAFADB4,75,-39 Received MQTT: AC2,6C1DEBAFAE18,75,35
Trail tracking disabled for 6C1DEBAFADB4
Trail tracking disabled for 6C1DEBAFAE18
Trail tracking disabled for 6C1DEBAFADB4

```

Hình 56. Theo dõi dữ liệu ở Command Prompt.



Giao diện theo dõi Tag, phân biệt bằng TagID (được khai báo trong class Tag).

INDOOR POSITIONING SYSTEM BASED ON AOA METHOD

System Status: ONLINE Tags: 1/2 Online Anchors: 2 Online Last Update: 21:55:32 19/04/2025

DASHBOARD

ALARMS

SETTINGS

ABOUT

Written by Thanh Huynh Chi

Time	Tag ID	Alarm Type	Message	Value
21:54:43 19/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 6.04 > 5.00	6.04
21:54:48 19/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
21:54:50 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 22.31 > 5.00	22.31
21:54:51 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 8.00 > 5.00	8.00
21:54:51 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 9.07 > 5.00	9.07
21:54:52 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 40.99 > 5.00	40.99
21:54:59 19/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
21:55:04 19/04/2025	6C1DEBAFAE18	Connection Lost	Connection lost	N/A
21:55:10 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 7.33 > 5.00	7.33
21:55:11 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 6.83 > 5.00	6.83
21:55:11 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 12.39 > 5.00	12.39
21:55:11 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 13.07 > 5.00	13.07
21:55:13 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 12.16 > 5.00	12.16
21:55:23 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 14.09 > 5.00	14.09
21:55:23 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 30.06 > 5.00	30.06
21:55:24 19/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 5.54 > 5.00	5.54
21:55:25 19/04/2025	6C1DEBAFAE18	Connection Lost	Connection lost	N/A
21:55:30 19/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A

Cảnh báo nếu Tag di chuyển quá nhanh hoặc mất kết nối đột ngột.

INDOOR POSITIONING SYSTEM BASED ON AOA METHOD

Tag 6C1DEBAFADB4 Details

Tag 6C1DEBAFADB4

Position: (2.80, 0.70)

Velocity: 6.17 units/sec

Status: Connected

Distance to Anchor 1: 2.41 units

Distance to Anchor 2: 1.14 units

Last Update: 21:56:26.648

Alarm Status: ALARM ACTIVE

Alarm Message: High velocity: 6.17 > 5.00

Velocity Over Time

Velocity (units/second)

21:55:30 21:55:40 21:55:50 21:56:00 21:56:10 21:56:20

6C1DEBAFAE18 1.4181 0.6114

Online Last Update: 21:56:29 19/04/2025

BASED ON AOA METHOD

Online Last Update: 21:56:29 19/04/2025

Anchor 1.1422

DEBAFAE18

Tag 6C1DEBAFADB4

Stop

Connect

Setup

Enable Trace

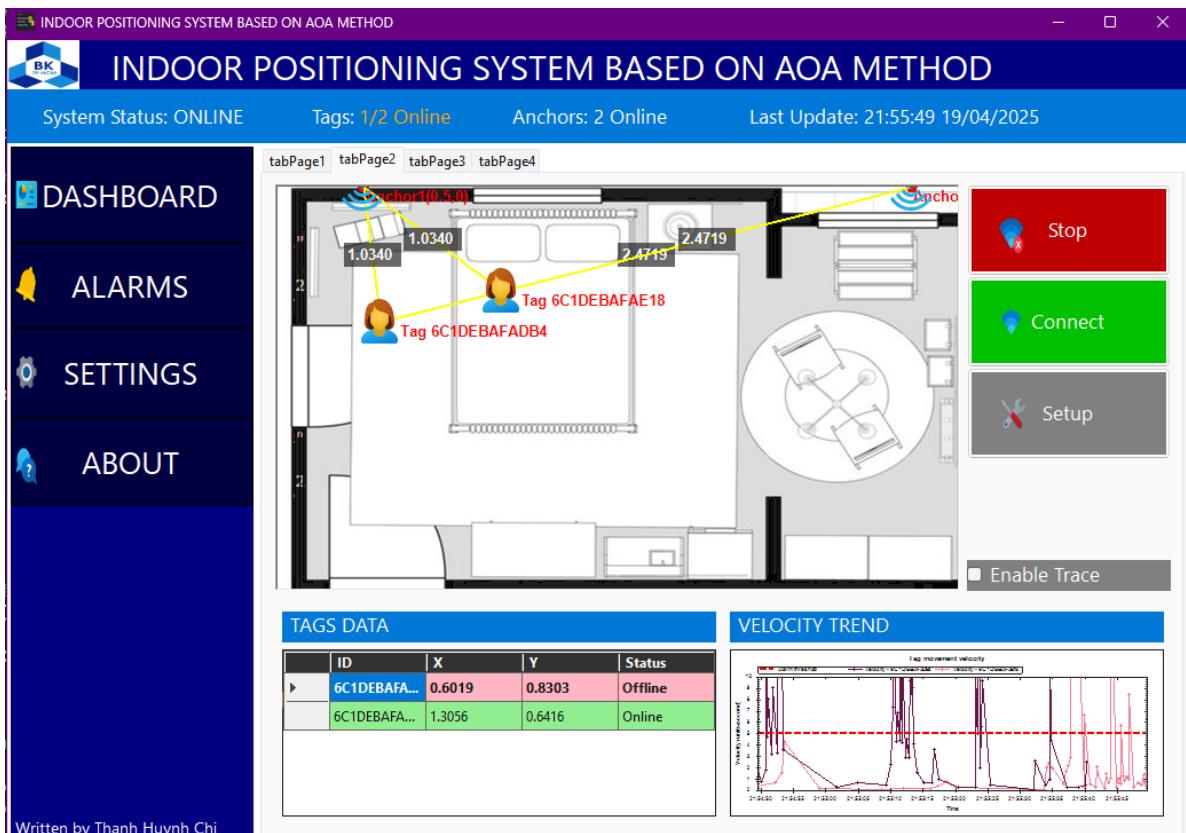
VELOCITY TREND

Tag movement velocity

Time

Written by Thanh Huynh Chi

Khi ấn vào biểu tượng (Icon) của Tag, một Tag Faceplate sẽ hiện lên để theo dõi các thông số của Tag.



Khi có Tag đột ngột mất kết nối, Trạng thái (Status) của Tag đó sẽ chuyển sang màu đỏ.

	New	Open	Save	Print	Copy	Cut	Delete	Sort	View	...
	Home	Name	Date modified	Type	Size					
		Alarm_Log_20250418_004443	4/18/2025 12:44 AM	Microsoft Excel C...	3 KB					

	Alarm_Log_				
	File	Home	Insert	Page Layout	Formulas
1	A	B	C	D	E
2	Time	Tag ID	Alarm Type	Message	Value
3	00:40:22 18/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
4	00:40:34 18/04/2025	6C1DEBAFAE18	Connection Lost	Connection lost	N/A
5	00:40:48 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 11.95 > 5.00	11.95
6	00:40:48 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 12.35 > 5.00	12.35
7	00:40:53 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 50.00 > 5.00	50
8	00:40:54 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 6.50 > 5.00	6.5
9	00:41:04 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 5.44 > 5.00	5.44
10	00:41:10 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 7.76 > 5.00	7.76
11	00:41:12 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 8.76 > 5.00	8.76
12	00:41:13 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 50.00 > 5.00	50
13	00:41:14 18/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
14	00:41:15 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 6.80 > 5.00	6.8
15	00:41:20 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 8.65 > 5.00	8.65
16	00:41:22 18/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
17	00:41:25 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 49.19 > 5.00	49.19
18	00:41:25 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 10.22 > 5.00	10.22
19	00:41:26 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 9.03 > 5.00	9.03
20	00:41:32 18/04/2025	6C1DEBAFAE18	High Velocity	High velocity: 5.47 > 5.00	5.47
21	00:41:47 18/04/2025	6C1DEBAFAE18	Connection Lost	Connection lost	N/A
22	00:41:49 18/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
23	00:41:52 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 13.17 > 5.00	13.17
24	00:41:58 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 23.38 > 5.00	23.38
25	00:42:06 18/04/2025	6C1DEBAFADB4	Connection Lost	Connection lost	N/A
26	00:42:19 18/04/2025	6C1DEBAFADB4	High Velocity	High velocity: 5.97 > 5.00	5.97

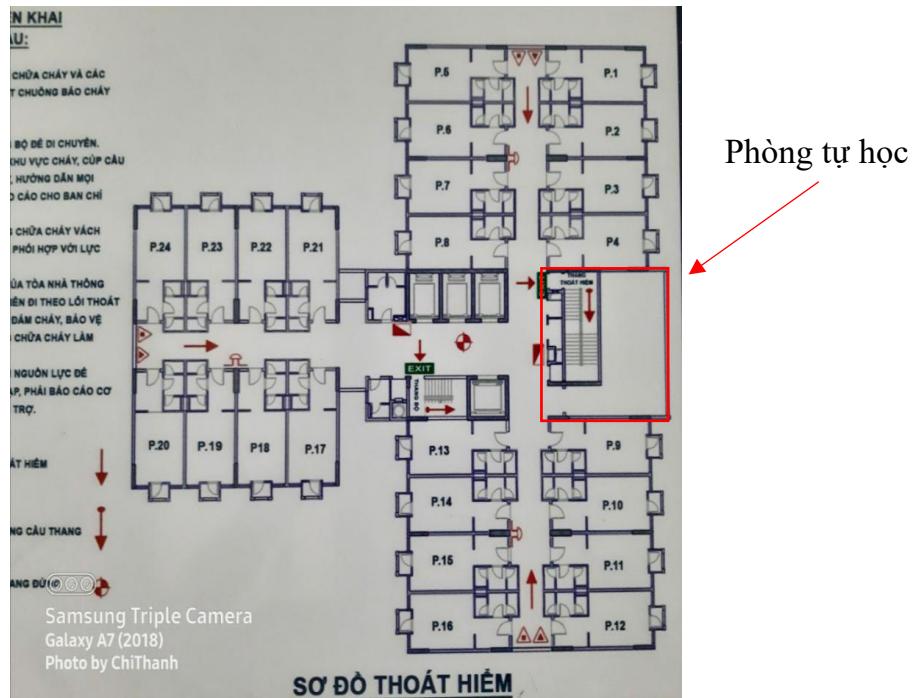
Hình 57. Lưu trữ dữ liệu

Khi tắt ứng dụng, dữ liệu sẽ được lưu vào file .CSV.

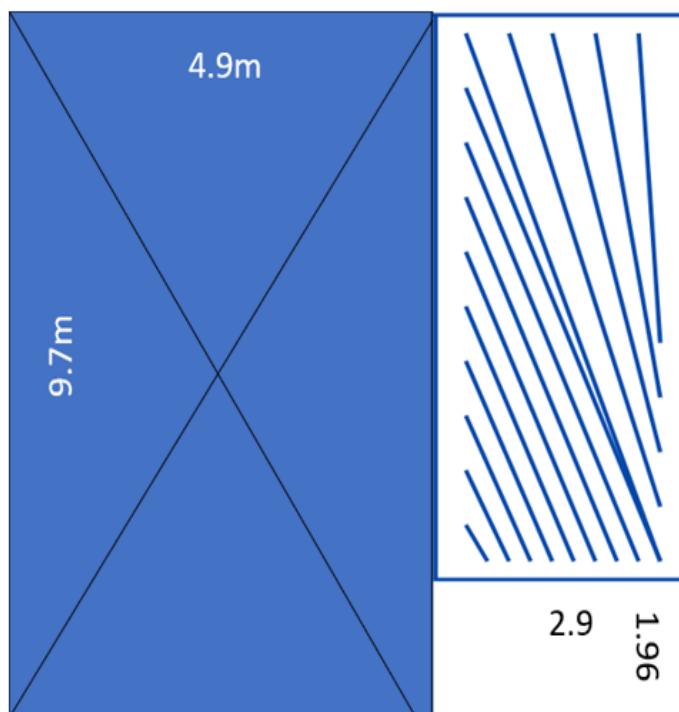
CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM

1. Không gian thực nghiệm

Thực nghiệm được tiến hành trong phòng tự học của Kí túc xá khu B Đại học Quốc gia TP.HCM. Không gian đo đạc được lựa chọn là vùng trống có diện tích phù hợp để đảm bảo tính chính xác của kết quả.



Hình 58. Khu vực thực nghiệm trên mặt bằng kí túc xá



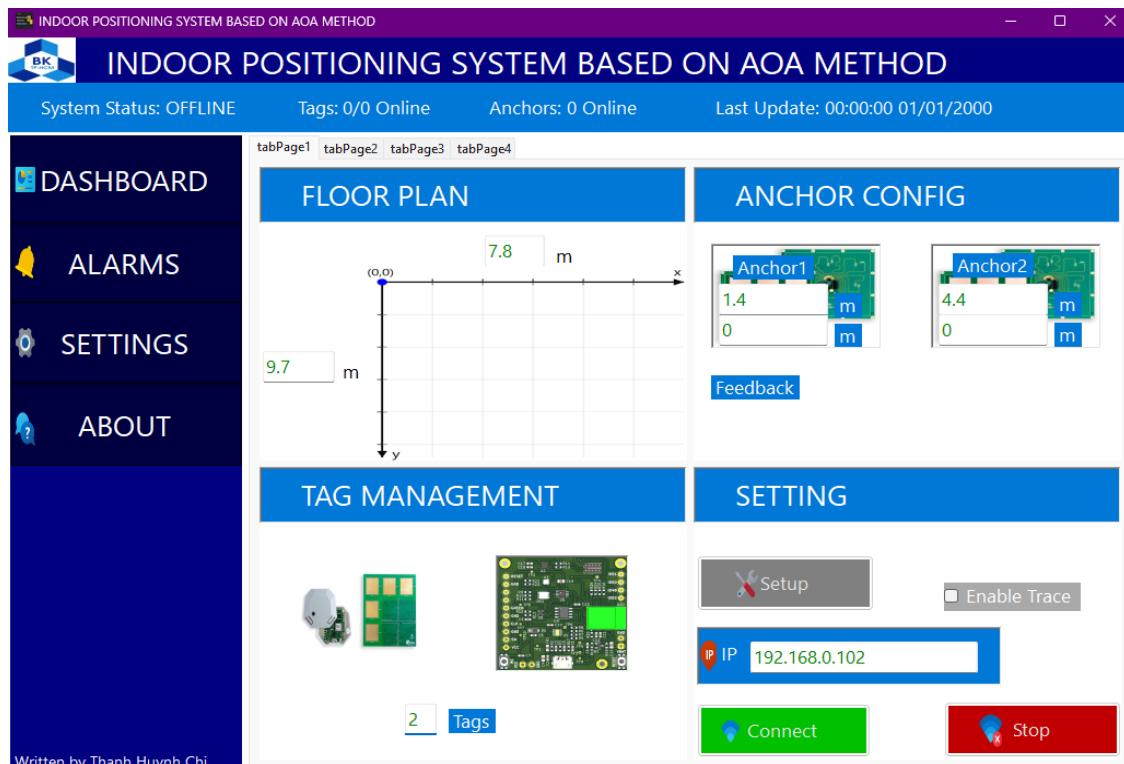
Hình 59. Bản vẽ mặt bằng phòng tự học

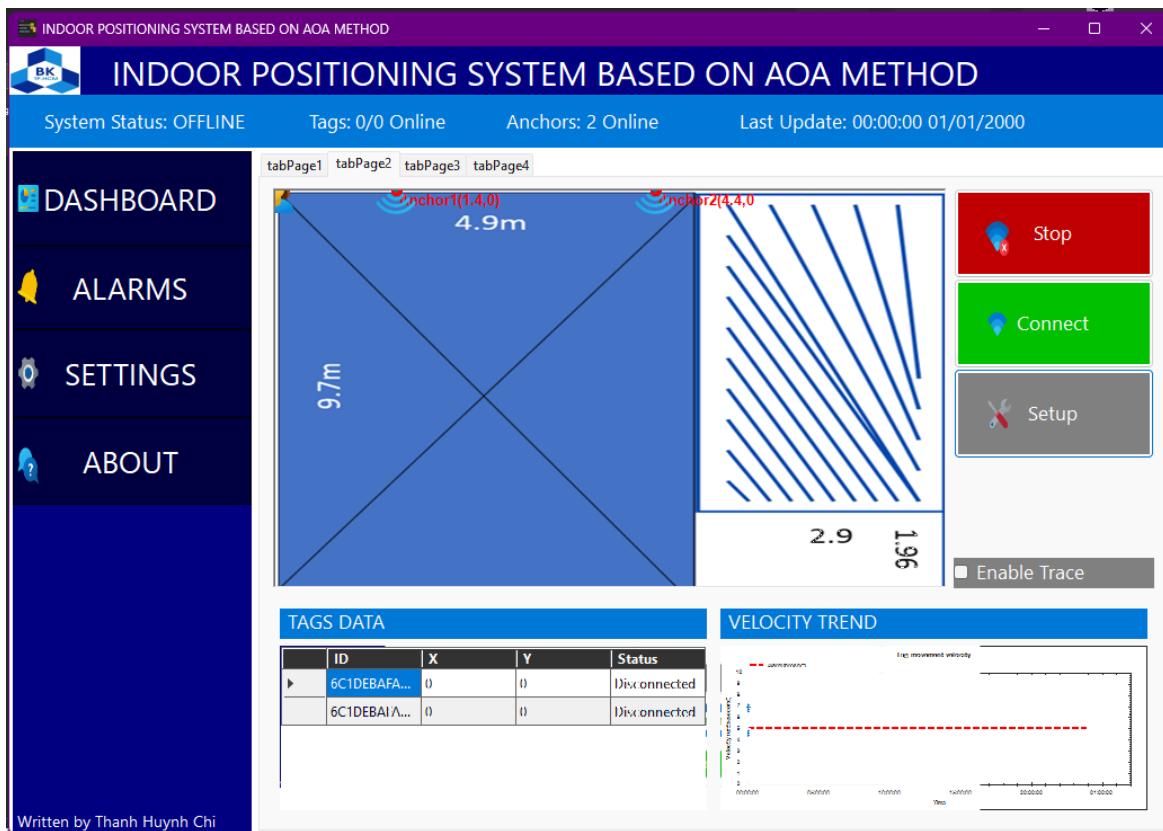
Vị trí đo đạc được xác định tại vùng màu xanh trong phòng, nơi cung cấp điều kiện tối ưu cho việc thu thập dữ liệu. Các thiết bị Anchor được bố trí theo sơ đồ đã thiết kế trước, với tọa độ được xác định chính xác làm cơ sở tham chiếu cho toàn bộ hệ thống.



Hình 60. Gán hệ trục cho không gian thực nghiệm

Hai thiết bị Anchor được cố định trên tường ở độ cao 2m so với mặt sàn.



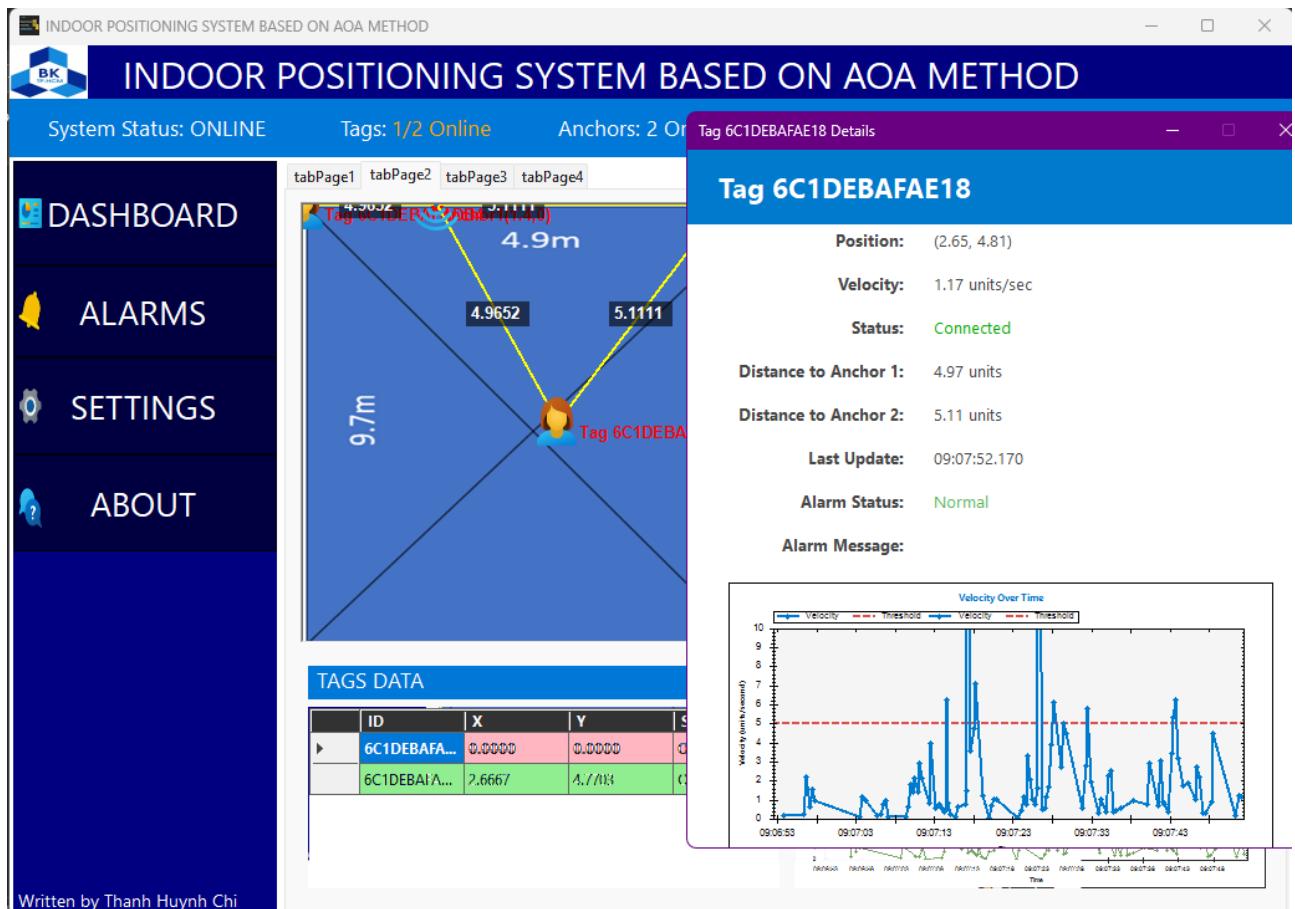


Trên mặt phẳng Oxy, tọa độ của Anchor 1 được xác định tại điểm (1.4,0), trong khi Anchor 2 được đặt tại tọa độ (4.4,0). Sau khi hoàn tất việc lắp đặt thiết bị, hệ thống được cấu hình trên Visual Studio để tiến hành theo dõi và thu thập dữ liệu.

2. Phương pháp đo đạc

Để đánh giá hiệu suất hệ thống đo lường một cách toàn diện, phương pháp đo đạc được thực hiện như sau:

- Chia chiều dài căn phòng thành 10 phần đều nhau (từ 0.5m đến 4.9m theo trục X)
- Chia chiều rộng căn phòng thành 10 phần đều nhau (từ 1m đến 9.7m theo trục Y)
- Tiến hành đo tại 100 vị trí giao nhau của lưới tọa độ
- Ghi nhận giá trị đo được và so sánh với giá trị thực tế
- Tính toán sai số phần trăm cho từng tọa độ



Hình 61. Tọa độ đo được của Tag



Hình 62. Tọa độ x thực tế của Tag



Hình 63. Tọa độ y thực tế của Tag

3. Kết quả

3.1. Kết quả đo tọa độ X

X Y \	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
1m	0.30	1.05	1.66	2.23	2.42	3.08	3.52	4.06	4.57	4.83
2m	0.50	1.01	1.64	2.14	2.50	3.18	3.50	4.00	4.53	4.85
3m	0.61	1.13	1.70	2.18	2.55	2.98	3.40	4.02	4.83	4.74
4m	0.60	1.20	1.52	2.17	2.50	3.17	3.53	4.20	4.41	4.75
5m	0.44	1.03	1.63	2.02	2.51	3.02	3.50	4.02	4.52	4.75
6m	0.50	1.14	1.37	1.75	2.54	3.00	3.75	4.06	4.77	4.60
7m	0.83	1.35	1.49	2.30	2.32	2.78	3.85	3.41	4.08	4.07
8m	1.13	1.71	2.22	2.50	2.22	3.00	3.86	4.88	3.72	4.03
9m	1.57	2.01	2.35	1.97	3.93	4.02	4.02	3.42	3.78	4.01
9.7m	1.33	1.99	1.89	2.72	2.75	2.22	3.01	3.22	3.73	4.12

Bảng 8. Kết quả đo tọa độ x

3.2 Kết quả đo tọa độ Y

X Y \ X	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
1m	1.26	1.02	1.17	1.20	1.20	1.07	1.00	1.09	1.09	1.15
2m	2.20	2.21	2.17	2.09	2.13	2.16	1.90	2.02	1.97	2.20
3m	3.40	3.08	3.14	3.07	3.10	2.88	3.00	3.05	3.14	3.35
4m	3.75	4.02	4.20	4.18	4.06	3.97	4.08	4.30	4.11	3.86
5m	5.07	4.79	5.14	5.12	4.88	5.02	5.01	5.13	5.01	5.30
6m	6.05	6.05	6.02	5.93	5.93	6.20	6.17	5.6	5.55	5.83
7m	7.35	7.80	7.40	7.64	7.64	7.32	7.40	8.74	8.25	7.94
8m	6.87	7.84	5.90	8.90	6.70	9.43	8.43	6.97	7.83	8.94
9m	9.54	9.47	08.56	8.29	7.97	8.16	9.62	6.74	7.78	9.62
9.7m	8.52	8.67	7.99	8.99	8.51	8.75	8.77	8.03	6.83	7.86

Bảng 9. Kết quả đo tọa độ y

4. Phân Tích Sai Số

4.1. Công Thức Tính Sai Số

Phần trăm sai số được tính theo công thức:

$$\text{Phần trăm sai số (\%)} = \frac{|\text{Giá trị đo được} - \text{Giá trị thực}|}{|\text{Giá trị thực}|} \times 100\%$$

4.2. Sai số tọa độ X

X Y \ X	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
1m	40.00	5.00	10.67	11.50	3.20	2.67	0.57	1.50	1.56	1.43
2m	0.00	1.00	9.33	7.00	0.00	6.00	0.00	0.00	0.67	1.02
3m	22.00	13.00	13.33	9.00	2.00	0.67	2.86	0.50	7.33	3.27
4m	20.00	20.00	1.33	8.50	0.00	5.67	0.86	5.00	2.00	3.06
5m	12.00	3.00	8.67	1.00	0.40	0.67	0.00	0.50	0.44	3.06

X Y \	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
6m	0.00	14.00	8.67	12.50	1.60	0.00	7.14	1.50	6.00	6.12
7m	66.00	35.00	0.67	15.00	7.20	7.33	10.00	14.75	9.33	16.94
8m	126.00	71.00	48.00	25.00	11.20	0.00	10.29	22.00	17.33	17.76
9m	214.00	101.00	56.67	1.50	57.20	34.00	14.86	14.50	16.00	18.16
9.7m	166.00	99.00	26.00	36.00	10.00	26.00	14.00	19.50	17.11	15.92

Bảng 10. Sai số tọa độ x

4.3. Sai số tọa độ Y

X Y \	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
1m	26.00	2.00	17.00	20.00	20.00	7.00	0.00	9.00	9.00	15.00
2m	10.00	10.50	2.17	2.09	2.13	2.16	1.90	2.02	1.97	2.20
3m	13.33	2.67	4.67	2.33	3.33	4.00	0.00	1.67	4.67	11.67
4m	6.25	4.02	4.20	4.18	4.06	3.97	4.08	4.30	4.11	3.86
5m	1.40	4.20	2.80	2.40	2.40	0.40	0.20	2.60	0.20	6.00
6m	0.83	0.83	0.33	1.17	1.17	3.33	2.83	6.67	7.50	2.83

X Y	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
7m	5.00	11.43	5.71	9.14	9.14	4.57	5.71	24.86	17.86	13.43
8m	14.13	2.00	26.25	11.25	16.25	17.88	5.38	12.88	2.13	11.75
9m	6.00	5.22	4.89	7.89	11.44	9.33	6.89	25.11	13.56	6.89
9.7m	12.16	10.62	17.63	7.32	12.27	9.79	9.59	17.22	29.59	18.97

Bảng 11. Sai số tọa độ y

4.4. Tổng Hợp Sai Số Root Mean Square Error (RMSE)

Để đánh giá tổng thể hiệu suất hệ thống tại mỗi điểm đo, ta tính toán tổng sai số bằng phương pháp RMSE (Root Mean Square Error):

$$\text{Tổng Sai Số} = \sqrt{(\text{Sai số X}^2 + \text{Sai số Y}^2)}$$

Kết quả tổng sai số được phân loại theo các mức độ chính xác:

- Màu xanh lá cây: Độ chính xác rất tốt (0% - 5%)
- Màu vàng: Độ chính xác tốt (5% - 20%)
- Màu cam: Độ chính xác trung bình (20% - 50%)
- Màu đỏ: Độ chính xác kém (> 50%)

X Y	0.5m	1m	1.5m	2m	2.5m	3m	3.5m	4m	4.5m	4.9m
1m	47.71	5.39	20.07	23.07	20.25	7.49	0.57	9.12	9.13	15.07
2m	10.00	10.55	9.58	7.31	2.13	6.38	1.90	2.02	2.08	2.43
3m	25.73	13.27	14.13	9.30	3.89	4.06	2.86	1.74	8.69	12.12
4m	20.95	20.40	4.41	9.47	4.06	6.92	4.17	6.59	4.57	4.93
5m	12.08	5.16	9.11	2.60	2.43	0.78	0.20	2.65	0.49	6.74
6m	0.83	14.02	8.67	12.55	1.98	3.33	7.68	6.83	9.60	6.75
7m	66.19	36.82	5.75	17.57	11.64	8.64	11.52	28.90	20.15	21.62
8m	126.79	71.03	54.71	27.41	19.74	17.88	11.61	25.49	17.46	21.29
9m	214.08	101.13	56.88	8.03	58.33	35.26	16.38	29.00	20.97	19.43
9.7m	166.45	99.57	31.41	36.74	15.83	27.78	16.97	26.01	34.18	24.76

Bảng 12. Tổng hợp sai số

5. Nhận Xét Về Bản Đồ Sai Số

Vùng độ chính xác rất cao (màu xanh lá - dưới 5%)

- Khu vực có độ chính xác tốt nhất tập trung xung quanh điểm (X=3, Y=5) với giá trị RMSE chỉ 0.20%
- Điểm (X=3.5, Y=5) cũng đạt độ chính xác cao với sai số 0.20%

- Điểm ($X=0.5$, $Y=6$) có độ chính xác tốt bất ngờ (0.83%) trong khi các điểm lân cận có sai số cao hơn nhiều
- Khu vực $Y=5$ và $Y=6$ với X từ 2.5-4 tạo thành một vùng có độ chính xác ổn định

Vùng độ chính xác tốt (màu vàng - 5% đến 20%)

- Chiếm phần lớn diện tích của không gian đo lường
- Bao gồm phần lớn khu vực trung tâm và bên phải của bản đồ
- Đặc biệt ổn định tại hàng $Y=2$, với hầu hết các giá trị X đều cho sai số ở mức chấp nhận được
- Nhiều điểm tại $Y=4$ cũng có độ chính xác khá tốt

Vùng độ chính xác trung bình và kém (màu cam và đỏ - trên 20%)

- Chiếm phần lớn ở vùng ngoài cùng, từ vùng có tọa độ $Y=7$ trở đi
- Ở vùng ngoài cùng bên trái, độ chính xác rất kém với những ô màu đỏ
- Ở vùng ngoài cùng bên phải, độ chính xác trung bình với những ô màu cam

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Qua quá trình nghiên cứu và thực nghiệm, đề tài đã xác định được các nguồn gốc chính gây ra sai số trong hệ thống định vị sử dụng phương pháp Góc Đến (Angle of Arrival - AoA):

- **Hạn chế của cấu hình hai Anchor:** Khi đối tượng cần định vị (Tag) nằm trên hoặc gần đường thẳng nối hai Anchor, các đường định hướng từ các Anchor trở nên gần như song song. Hiện tượng này tạo ra tình trạng bất định trong tính toán giao điểm, dẫn đến sai số lớn và kết quả định vị không ổn định.
- **Độ nhạy cao của phương pháp AoA:** Sự thay đổi nhỏ trong phép đo góc đến có thể biến đổi thành sai số lớn trong kết quả định vị không gian. Đặc biệt, khi khoảng cách giữa Tag và Anchor tăng, mức độ ảnh hưởng của sai số góc càng trở nên nghiêm trọng theo tỷ lệ bình phương của khoảng cách.
- **Giới hạn về định vị không gian:** Cấu hình hai Anchor chỉ cung cấp khả năng xác định vị trí trong không gian hai chiều (2D), không đáp ứng được nhu cầu định vị trong không gian ba chiều (3D) của nhiều ứng dụng thực tế.

Khi so sánh với nghiên cứu "Indoor Multipath Assisted Angle of Arrival Localization", ta thấy Stijn Wielandt và Lieven De Strycker đã giải quyết vấn đề bố trí Anchor: Thay vì sử dụng nhiều Anchor, tác giả dùng chỉ một mảng ăng-ten duy nhất nhưng khai thác hiện tượng đa đường truyền (multipath). Điều này loại bỏ hoàn toàn vấn đề "đường thẳng nối hai Anchor" mà nhóm đề cập.

Ở nghiên cứu "Định vị trong nhà sử dụng công nghệ Bluetooth Low Energy" của Võ Minh Phụng, tác giả đã chỉ ra rằng nên kết hợp nhiều phương pháp định vị (AoA + RSSI + ToA) thay vì chỉ dựa vào một phương pháp như hệ thống mà luận án đề cập.

Tóm lại, các nghiên cứu được đề cập đã thành công trong việc khắc phục những hạn chế cơ bản của phương pháp AoA truyền thống, đặc biệt là vấn đề cấu hình hai Anchor mà hệ thống mà nhóm gặp phải. Họ đạt được điều này thông qua các cách tiếp cận sáng tạo như khai thác đa đường truyền, sử dụng công nghệ ánh sáng, hoặc kết hợp nhiều phương pháp định vị. Điều này làm cho các giải pháp của họ không chỉ chính xác hơn mà còn thực tế và kinh tế hơn khi triển khai trong môi trường thực tế.

2. Hướng phát triển

Dựa trên những phân tích về hạn chế hiện tại, nhóm nghiên cứu đề xuất các hướng phát triển sau đây nhằm nâng cao hiệu quả của hệ thống:

Tối ưu hóa số lượng và vị trí Anchor:

- Triển khai hệ thống với ba hoặc bốn Anchor được bố trí không thẳng hàng sẽ giúp xác định vị trí duy nhất trong hầu hết các trường hợp, loại bỏ vấn đề về vị trí đối xứng.
- Cấu hình nhiều Anchor góp phần cải thiện đáng kể chỉ số "độ pha loãng hình học của độ chính xác" (Geometric Dilution of Precision - GDOP), từ đó thu hẹp vùng không chắc chắn của vị trí ước tính.

Cải tiến thuật toán xử lý:

- Phát triển và ứng dụng các bộ lọc hiệu quả hơn như bộ lọc Kalman mở rộng (Extended Kalman Filter) hoặc bộ lọc phần tử (Particle Filter) để xử lý tốt hơn các nhiễu trong dữ liệu góc đèn.
- Kết hợp với các phương pháp ước lượng vị trí khác như RSSI (Received Signal Strength Indicator) để tạo ra hệ thống lai có độ chính xác cao hơn.

Mở rộng sang định vị ba chiều:

- Bổ sung thêm Anchor và tối ưu vị trí không gian để cho phép xác định tọa độ trong không gian ba chiều, đáp ứng nhu cầu của các ứng dụng thực tế phức tạp hơn.

TÀI LIỆU THAM KHẢO

- [1] u-blox AG (05/2023), "XPLR-AOA-3: Evaluation kit for ANT-B10 antenna boards - User Guide, Truy cập từ: <https://www.u-blox.com/en/product/xplr-aoa-3-kit>
- [2] u-blox AG (09/2024), "u-locateEmbed: Short range standalone modules AT commands manual". Truy cập từ: <https://www.u-blox.com/en/product/u-locateEmbed>
- [3] u-blox AG (01/2024), "ANT-B10: Direction Finding antenna board - System integration manual". Truy cập từ: <https://www.u-blox.com/en/product/u-locateEmbed>
- [4] ANT-B10 data sheet, UBX-22008373.
- [5] u-blox, *u-blox tag software example*. Truy cập từ: <https://github.com/u-blox/c209-aoa-tag>
- [6] Karanam, S., Korany, B., & Mostofi, Y. (2018). *Magnitude-Based Angle-of-Arrival Estimation, Localization, and Target Tracking*. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 251–262).
- [7] Zhu, Z., Chen, C., & Yang, B. (2024). *Angle of Arrival Estimation with Transformer: A Sparse and Gridless Method with Zero-Shot Capability*. arXiv preprint arXiv:2408.09362.
- [8] Wei, Y.-L., & Choudhury, R. R. (2021). *Estimating Angle of Arrival (AoA) of Multiple Echoes in a Steering Vector Space*. arXiv preprint arXiv:2109.13072
- [9] Eman Saad, Mostafa Elhosseini (12/2018), “Recent achievements in sensor localization algorithms”. Truy cập từ: *Recent achievements in sensor localization algorithms - ScienceDirect*
- [10] u-blox (02/02/2024), “XPLR-AOA-1 and XPLR AOA-2 explorer kits”. Truy cập từ: *Product resources | u-blox*
- [11] jornbh (17/01/2023). Truy cập từ: *Releases · NordicSemiconductor/pc-nrfutil*
- [12] u-blox (01/07/2024), “NINA-B40 series”. Truy cập từ: *Product resources | u-blox*
- [13] u-blox (31/03/2023), “s-center Evaluation software for short range modules”. Truy cập từ: *Product resources | u-blox*

PHỤ LỤC