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

.....o0o.....

ĐỒ ÁN 2: ĐIỀU KHIỂN TỰ ĐỘNG

Đề tài: Indoor Positioning via Bluetooth

(Angel of Arrival method)

GVHD: Th.S Hồ Thanh Phương

Sinh viên thực hiện: Cù Quốc Cường

MSSV: 2112956

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2024

MỤC LỤC

NỘI DUNG

LỜI CẢM ƠN

Kính gửi ThS. Hồ Thanh Phương

Em xin chân thành cảm ơn Cô đã tận tình hướng dẫn, hỗ trợ và đồng hành cùng em trong suốt quá trình thực hiện đồ án. Sự tận tâm của Cô trong việc chia sẻ kiến thức, kinh nghiệm quý báu và những góp ý chân thành đã giúp em không chỉ hoàn thiện đồ án mà còn trau dồi thêm nhiều kỹ năng quan trọng. Những bài học mà em nhận được sẽ là hành trang quý giá cho con đường học tập và sự nghiệp tương lai của em.

Một lần nữa, em xin kính chúc Cô luôn mạnh khỏe, hạnh phúc và gặt hái thêm nhiều thành công trong sự nghiệp giảng dạy và nghiên cứu.

Trân trọng,

CÙ QUỐC CƯỜNG.

GIỚI THIỆU ĐỀ TÀI

Indoor Positioning via Bluetooth

(Angel of Arrival method)

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.

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 như độ phân giải của ăng-ten, nhiễu tín hiệu, hoặc sự phức tạp của môi trường. 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 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.

DANH MỤC HÌNH ẢNH

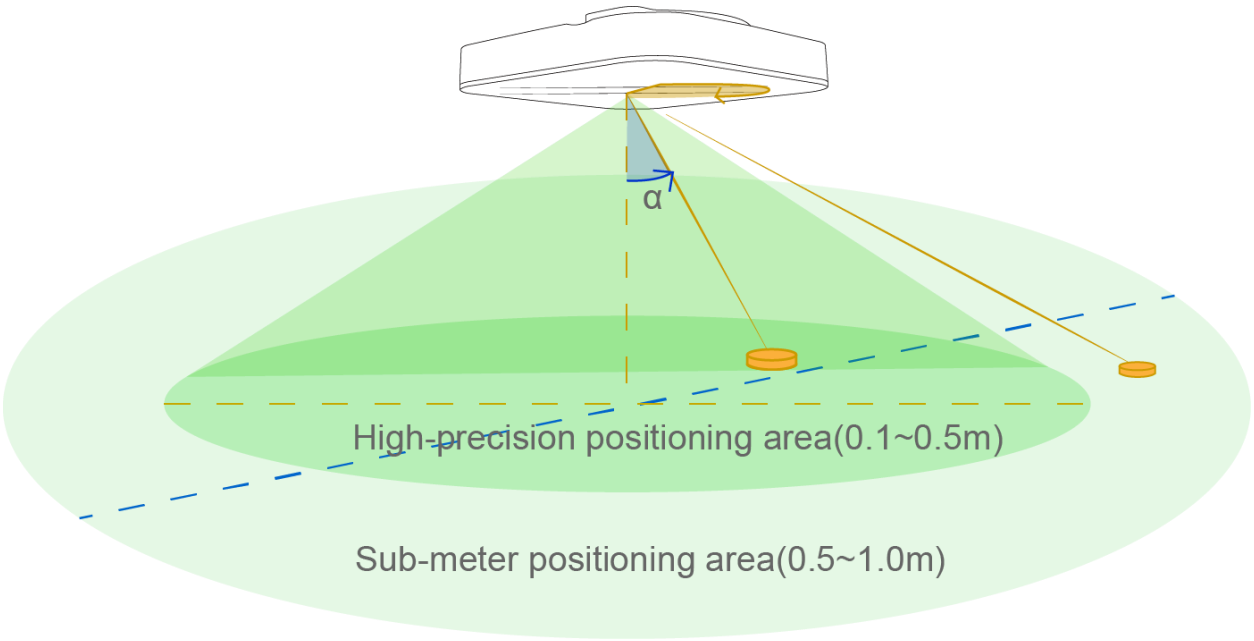
I. Tổng quan về Indoor Positioning System(IPS)

1. Indoor Positioning System

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 gnhe65 chuyên dụng nhơn 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 AoA*

Các khái niệm chính về AoA:

- Nguyên lý cơ bản:

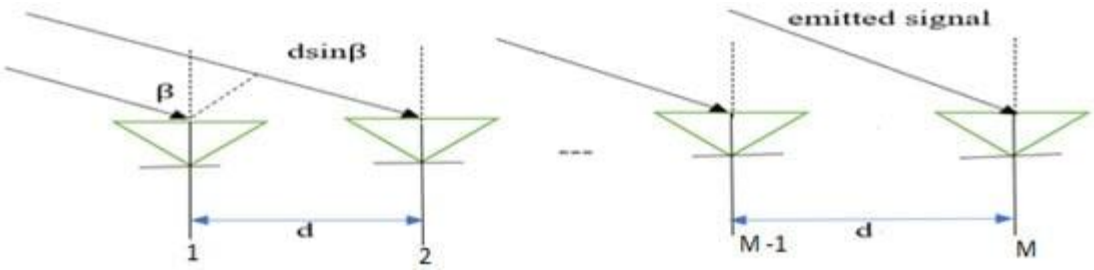
AoA dựa trên việc đo độ chênh lệch pha của tín hiệu nhận được từ nhiều ăng-ten được đặt theo một cách cụ thể. Bằng cách phân tích những chênh lệch này, hệ thống có thể tính toán hướng tín hiệu đến.

- Mảng ăng-ten:

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 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 Anten*

- 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 thông số kỹ thuật Bluetooth 5.1, đã giới thiệu những cải tiến cho việc tìm hướng bên cạnh các tính năng tiêu chuẩn của Bluetooth Low Energy.

- Lợi ích của AoA:

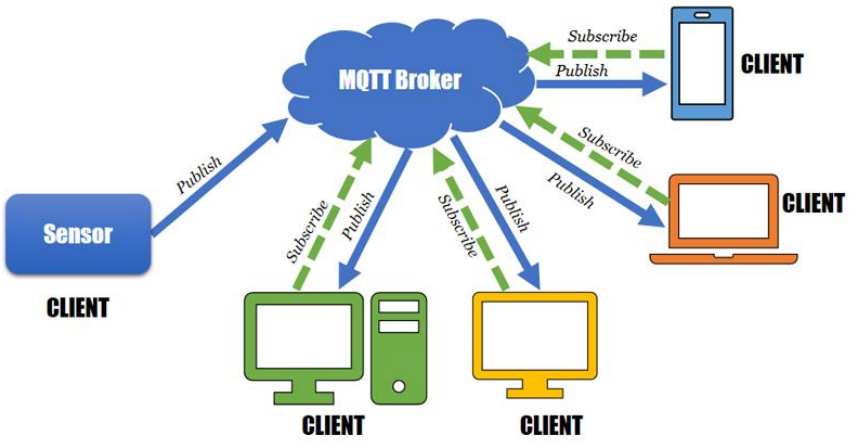
-Độ chính xác cao: Cung cấp dữ liệu vị trí chính xác hơn so với các phương pháp định vị khác.

-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.

II. Giới thiệu giao thức truyền thông 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 3. *Giao thức MQTT*

Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc. Client bắt đầu bằng cách tạo kết nối Transmission Control Protocol/Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker. Khi tạo kết nối, điều quan trọng là phải nhận ra rằng máy chủ có thể tiếp tục một phiên cũ nếu nó được cung cấp ID Client mà được sử dụng lại.

Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) / Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, Client cần kiểm chứng và xác thực máy chủ. Client cũng có thể cung cấp tính xác thực cho broker trong quá trình giao tiếp. Broker có thể sử dụng điều này để xác thực Client. Mặc dù không phải là một phần cụ thể của đặc trưng MQTT, nhưng các broker đã trở thành thông lệ để hỗ trợ xác thực máy khách bằng SSL/TLS phía máy khách.

Trong đó:

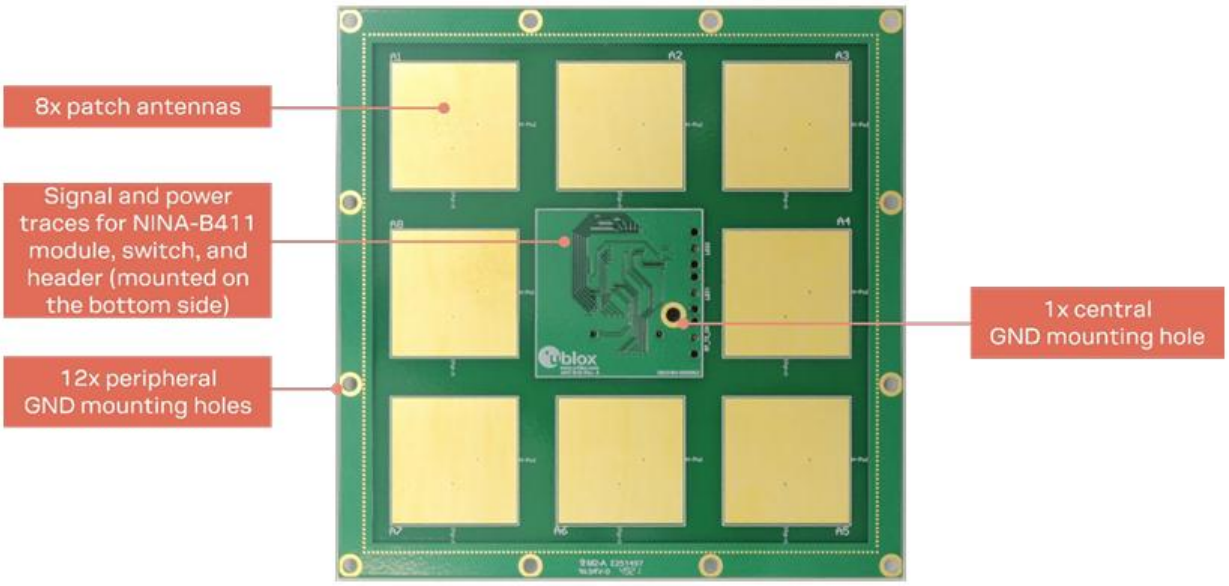
-Broker: là một thành phần trung gian (môi giới)

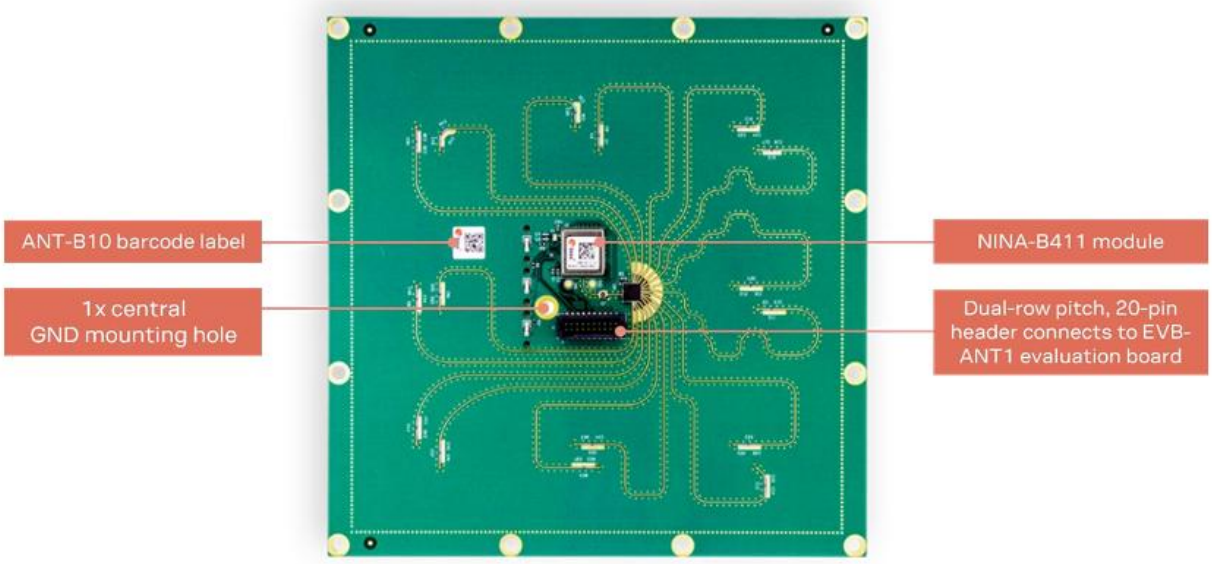
-SSL/TLS (Secure Sockets Layer/Transport Layer Security) là kỹ thuật mã hóa truyền tin trên internet.

Giao thức MQTT được thiết kế để phục vụ các thiết bị IoT và giảm thiểu tài nguyên tiêu thụ. Trong nhiều trường hợp, SSL/TLS không phải là yêu cầu bắt buộc và có thể không cần thiết. Thay vào đó, việc xác thực thường được thực hiện thông qua tên người dùng và mật khẩu, được máy khách gửi đến máy chủ như một phần của gói CONNECT/CONNACK. Một số broker, đặc biệt là các broker công khai trên internet, cho phép sử dụng tài khoản máy khách ẩn danh. Trong các trường hợp này, tên người dùng và mật khẩu được để trống, không cần thiết lập tài khoản.

III. Giới thiệu về kit 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.



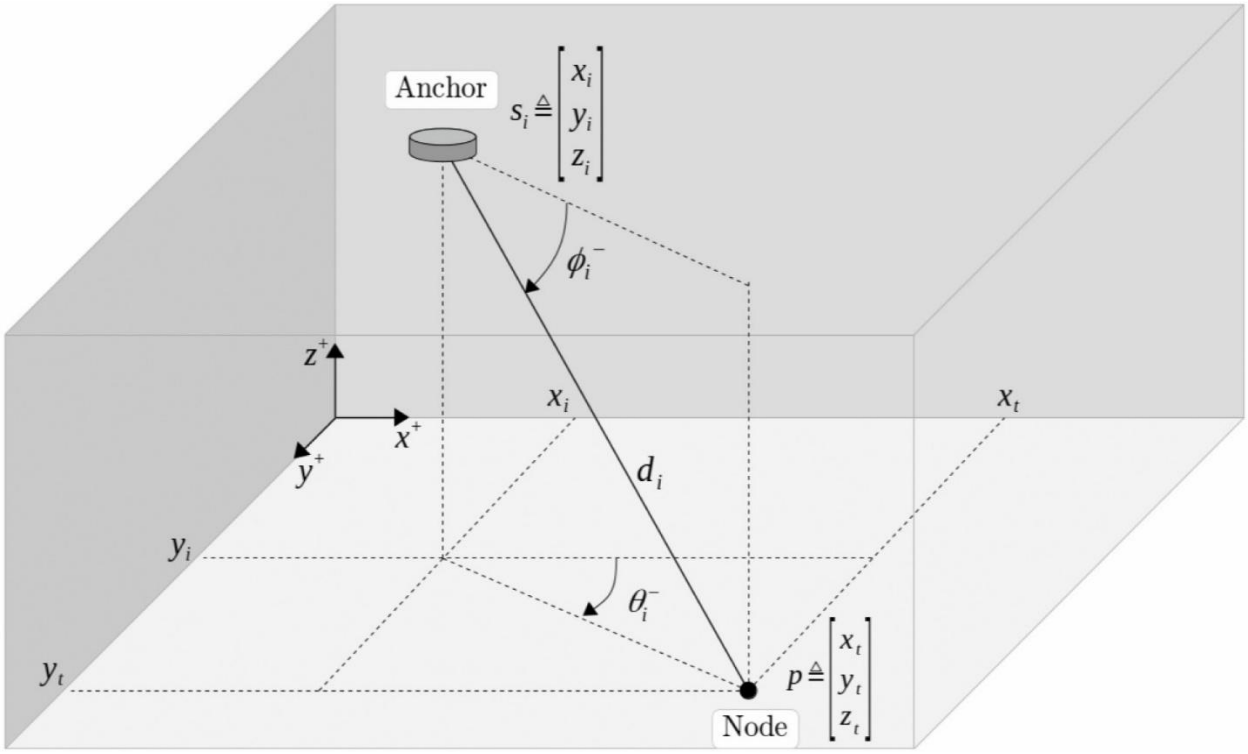


***Hình 4. ANT-B10** Board*

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, bo 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.



Hình 5. *Mô tả các góc trong không gian 3 chiều*

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. Các thành phần cơ bản của ANT-B10

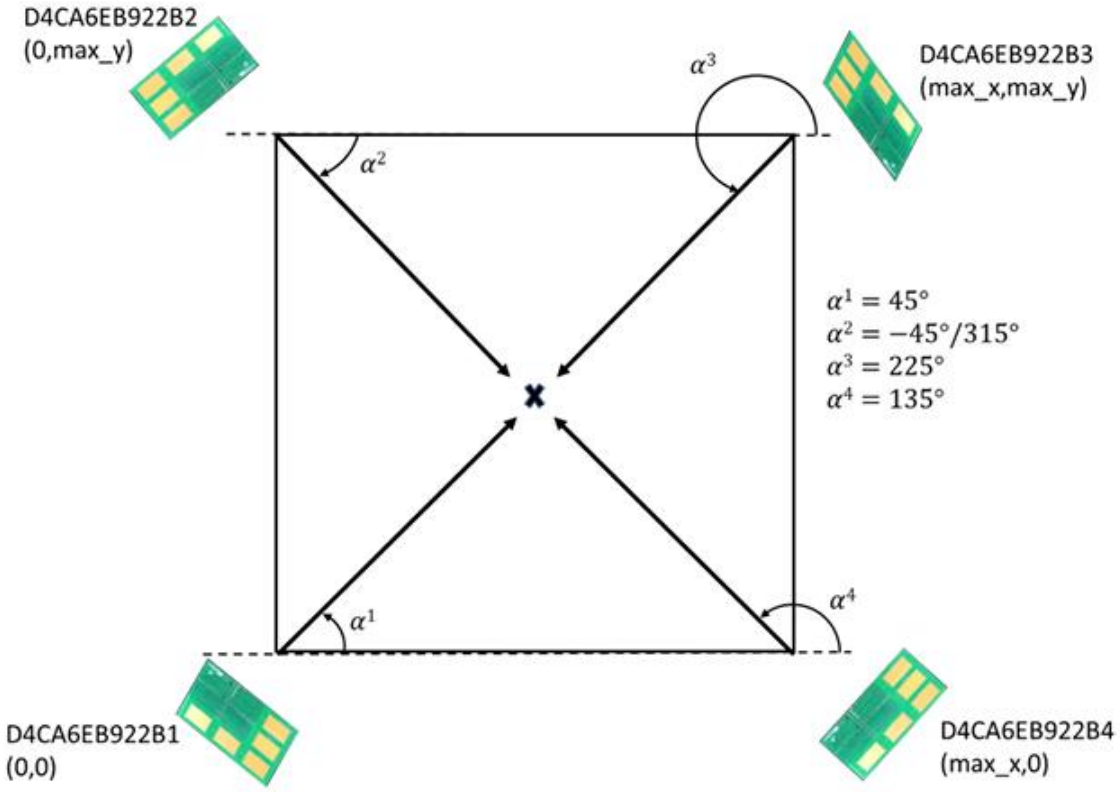
- **Antena Array**:

- 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.

- **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).

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

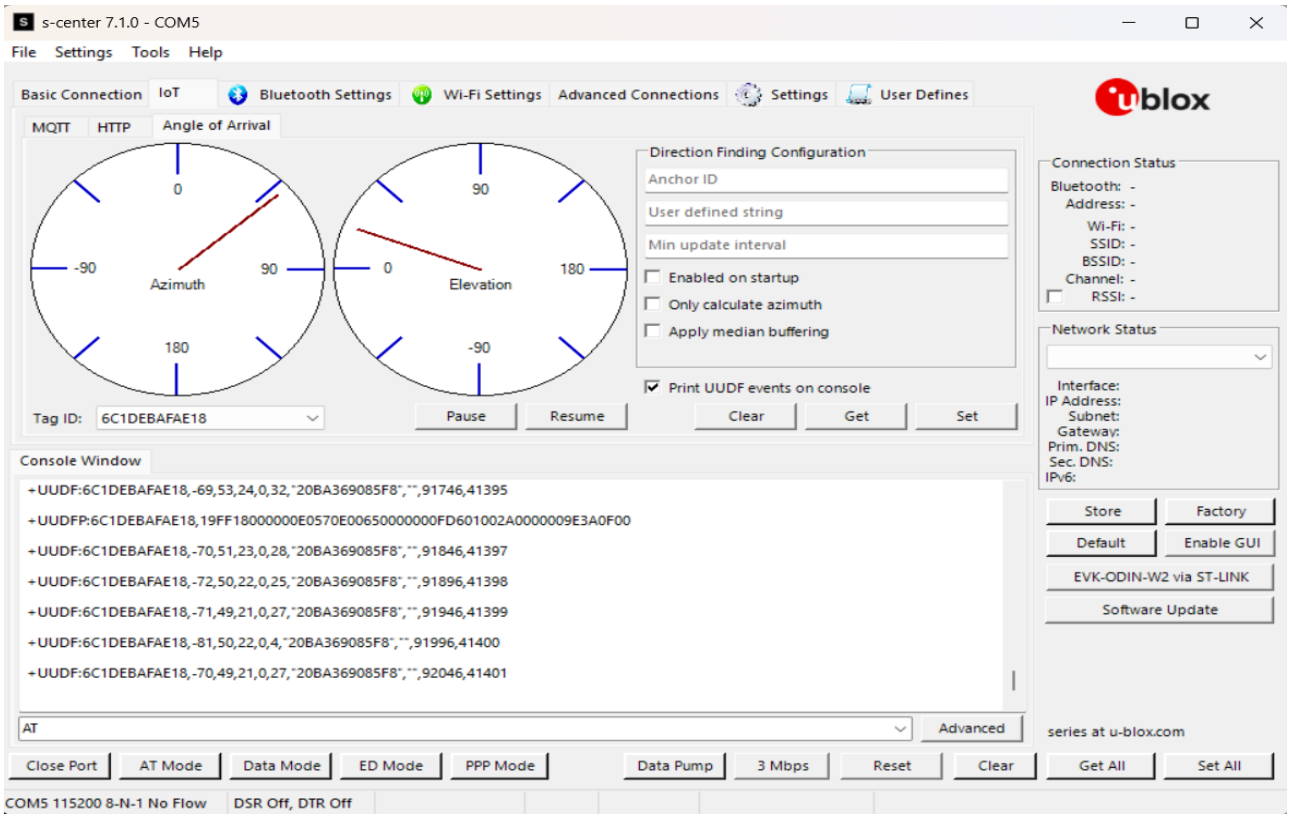
Chức năng AoA có thể được sử dụng với các bố trị các Anchor và Tag như hình sau



Hình 6. *Bố trí các Anchor*

Các Anchor được bố trí tại các điểm ngẫu nhiên trong khi 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 7. *Kết quả nhận được từ Anchor*

Kết quả trả về như sau

```
+UUDF:<ed_instance_id>,<rssi>,<azimuth_angle>,<elevation_angle>,<not_used>,<channel>,
<anchor_id>,<user_defined_str>,<timestamp_ms>,<periodic_event_counter>
```

Trong đó:

ed_instance_id: ID instance Eddystone 6 byte.

rssi: 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 trữ.

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

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

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

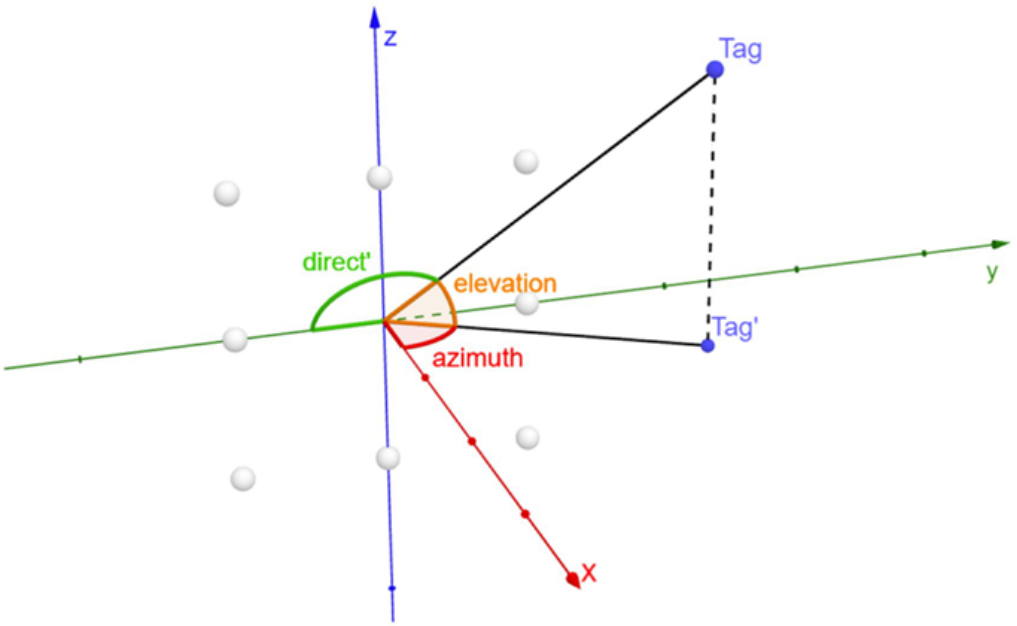
timestamp_ms: Thời gian kể từ khi khởi động (tính bằng mili-giây).

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

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

- azimuth – azimuth_angle
- elevation – elevation_angle
- direct' - the complementary angle of direct angle: $direct\_angle = 90^o - direct'$

The white dots represent the antenna board. The x- and y-axis represent the outward normal orientation of the board.

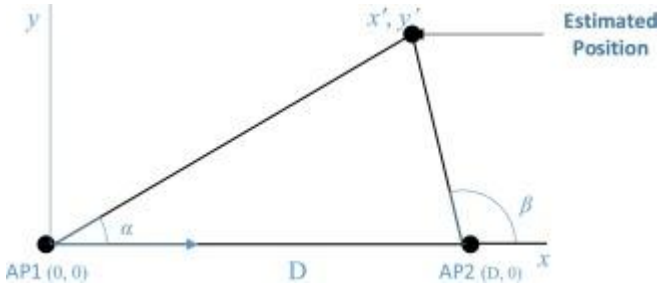


Hình 8. *Mô tả hệ trục tọa độ Anchor*

- 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 = distance * Sin(elevationRad) * Cos(azimuthRad)
  - y = distance * Sin(elevationRad) * Sin(azimuthRad)
  - z = distance * Cos(elevationRad)
- Đối với trường hợp có 2 Anchor và 1 Tag để định vị ta có thể thực hiện tính toán như sau:

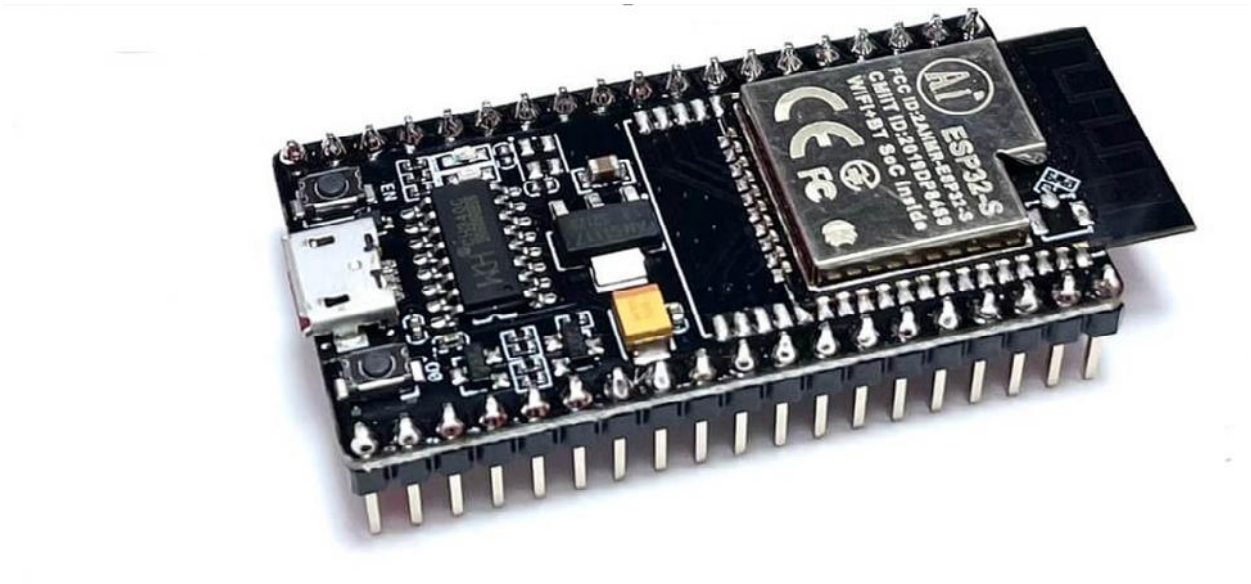


Hình 9. *Hệ tọa độ 2D của phương pháp AoA*

- $x' = \frac{D*\tan(\beta)}{\tan(\beta)-\tan(\alpha)}$
- $y' = \frac{D*\tan(\beta)*\tan(\alpha)}{\tan(\beta)-\tan(\alpha)}$

IV. Giới thiệu vi điều khiển Kit WiFi BLE ESP32 NodeMCU-32S

Kit WiFi BLE ESP32 NodeMCU-32S là một bo được phát triển trên nền Vi điều khiển trung tâm là ESP32 SoC với công nghệ Wifi, BLE và kiến trúc ARM mới nhất hiện nay. Đây là một nền tảng lý tưởng cho các ứng dụng IoT, nhà thông minh, và các hệ thống điều khiển tự động nhờ khả năng kết nối WiFi và Bluetooth (BLE) tích hợp. Bo mạch được thiết kế nhỏ gọn, tiết kiệm năng lượng và dễ dàng tích hợp vào nhiều dự án khác nhau.



Hình 10. *Kit Wifi BLE ESP32 NodeMCU-32S CH340 Ai-Thinker*

Thông số kỹ thuật chính:

- **Vi điều khiển trung tâm ESP32**:
  - Lõi kép 32-bit, xung nhịp lên đến 240 MHz.
  - Bộ nhớ Flash tích hợp 4MB.
  - Hỗ trợ kết nối WiFi 802.11 b/g/n và Bluetooth 4.2/BLE.
- **Cổng giao tiếp:**
  - **USB-to-Serial CH340**: Đảm bảo kết nối ổn định với máy tính qua cổng USB.
  - Hỗ trợ giao tiếp UART, SPI, I2C, PWM, và ADC/DAC.
- **Kích thước nhỏ gọn:**
  - Thiết kế tiện lợi, dễ dàng sử dụng trên các breadboard.
- **Điện áp hoạt động:**
  - Điện áp vào từ 5V (USB) hoặc 3.3V (chân cấp nguồn ngoài).
- **GPIO đa dụng:**
  - Hỗ trợ lên đến 34 chân GPIO, có thể lập trình cho nhiều mục đích khác nhau như đọc tín hiệu analog, xuất xung PWM, hoặc giao tiếp với cảm biến và thiết bị ngoại vi.

Tính năng nổi bật:

- **Kết nối WiFi và BLE**: Tích hợp cả WiFi và Bluetooth Low Energy, phù hợp cho các ứng dụng yêu cầu truyền dữ liệu không dây.
- **Chip CH340 tích hợp**: Giúp lập trình dễ dàng trên các hệ điều hành phổ biến như Windows, MacOS và Linux.
- **Hiệu suất cao, tiết kiệm năng lượng**: ESP32 có chế độ ngủ sâu (Deep Sleep), giúp kéo dài thời gian hoạt động cho các thiết bị chạy bằng pin.
- **Dễ dàng phát triển phần mềm**: Hỗ trợ Arduino IDE, PlatformIO, và ESP-IDF, giúp lập trình viên tiếp cận nhanh chóng.

Ứng dụng phổ biến:

- **Nhà thông minh (Smart Home):** Điều khiển đèn, quạt, và các thiết bị gia dụng.
- **IoT (Internet of Things):** Thu thập dữ liệu từ cảm biến và gửi lên đám mây.
- **Các dự án robot hoặc xe tự hành:** Kết nối không dây và điều khiển từ xa.
- **Hệ thống giám sát và cảnh báo từ xa:** Kết nối cảm biến với internet.

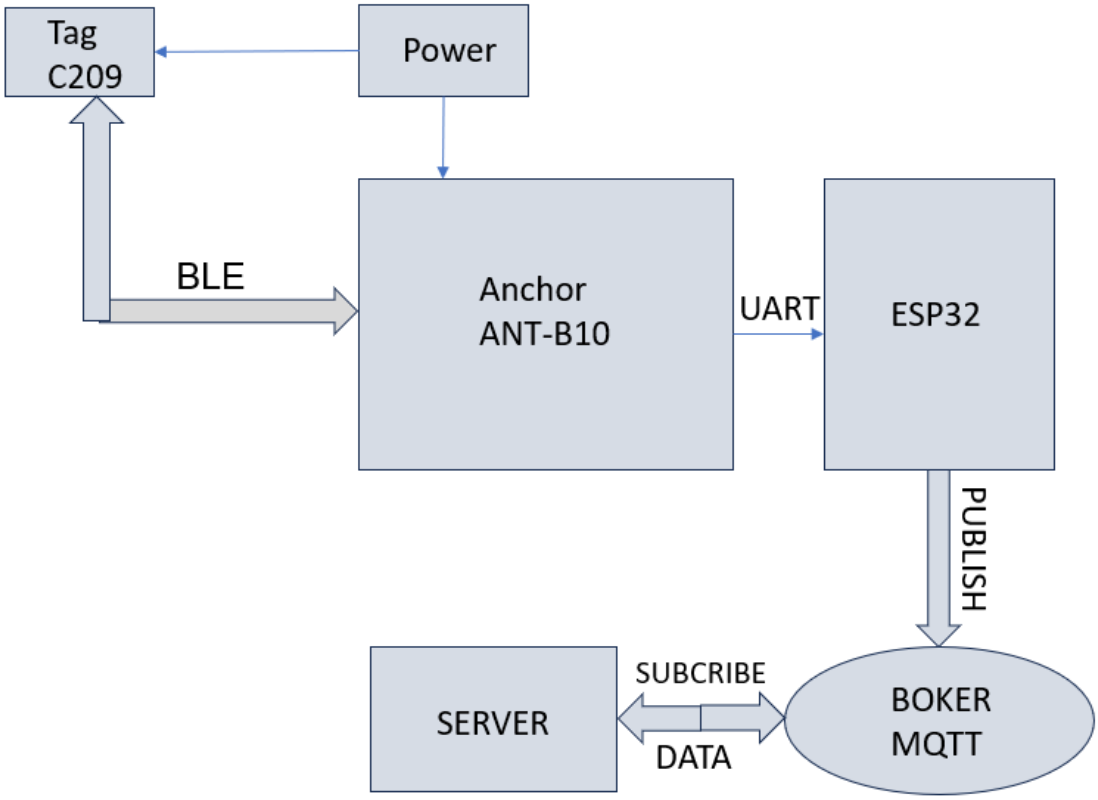
V. Thực hiện đề tài

1. Linh kiện

STT	Tên linh kiện	Số lượng
1	**Kit ANT- B10**	2
2	**C209 Tag**	1
3	**Kit Wifi BLE ESP32 NodeMCU-32S**	1
4	**Kit phát triển Wifi ESP8266 NodeMCU Lua V3**	1
5	**Nguồn điện 3.3V, dây nối,...**	

2. Mô hình phần cứng

Sơ đồ khối hệ thống:



Hình 11. *Sơ đồ khối hệ thống*

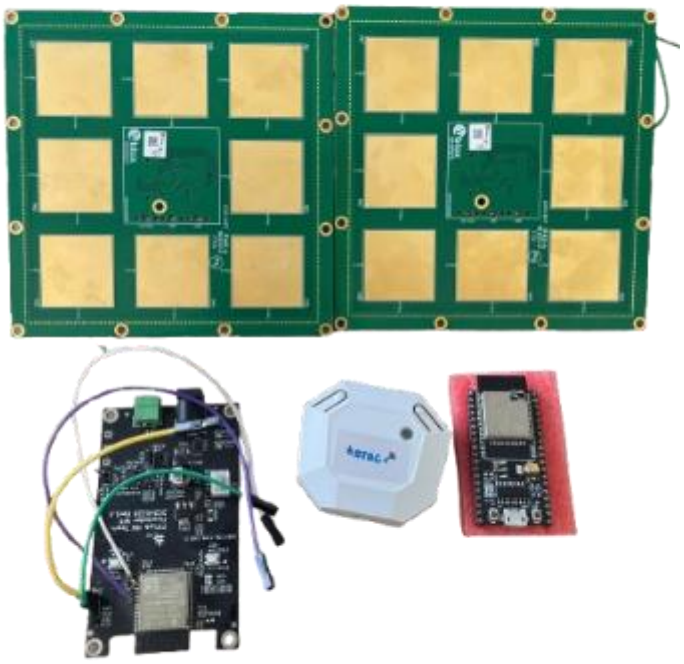
Nguyên lý hoạt động:

1. Anchor nhận tín hiệu từ Tag.

2. Anchor xử lý dữ liệu dựa trên chương trình đã lập trình, gửi dữ liệu đến ESP32 bằng giao thức UART.

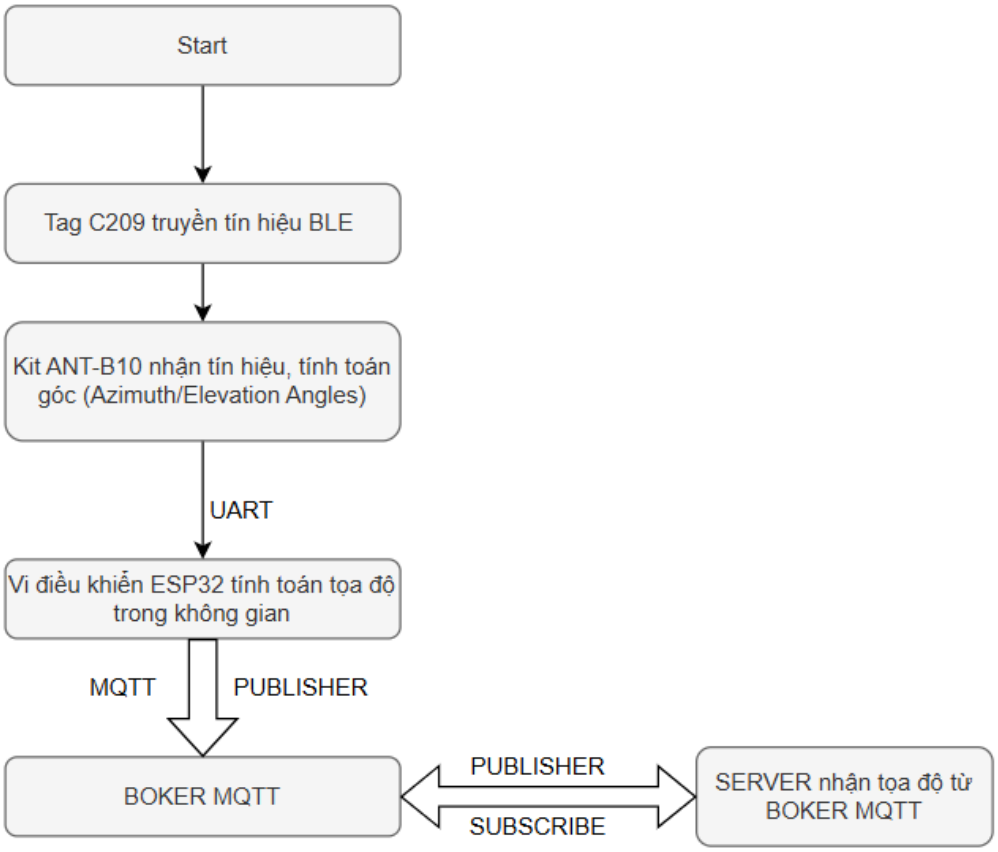
3. ESP32 gửi dữ liệu(Publish) nhận được lên BOKER bằng giao thức MQTT.

4. Server đăng kí(Subcribe) vào Topic đã khởi tạo cho Boker MQTT nhận dữ liệu, tính toán và hiển thị tọa độ, khoảng cách.

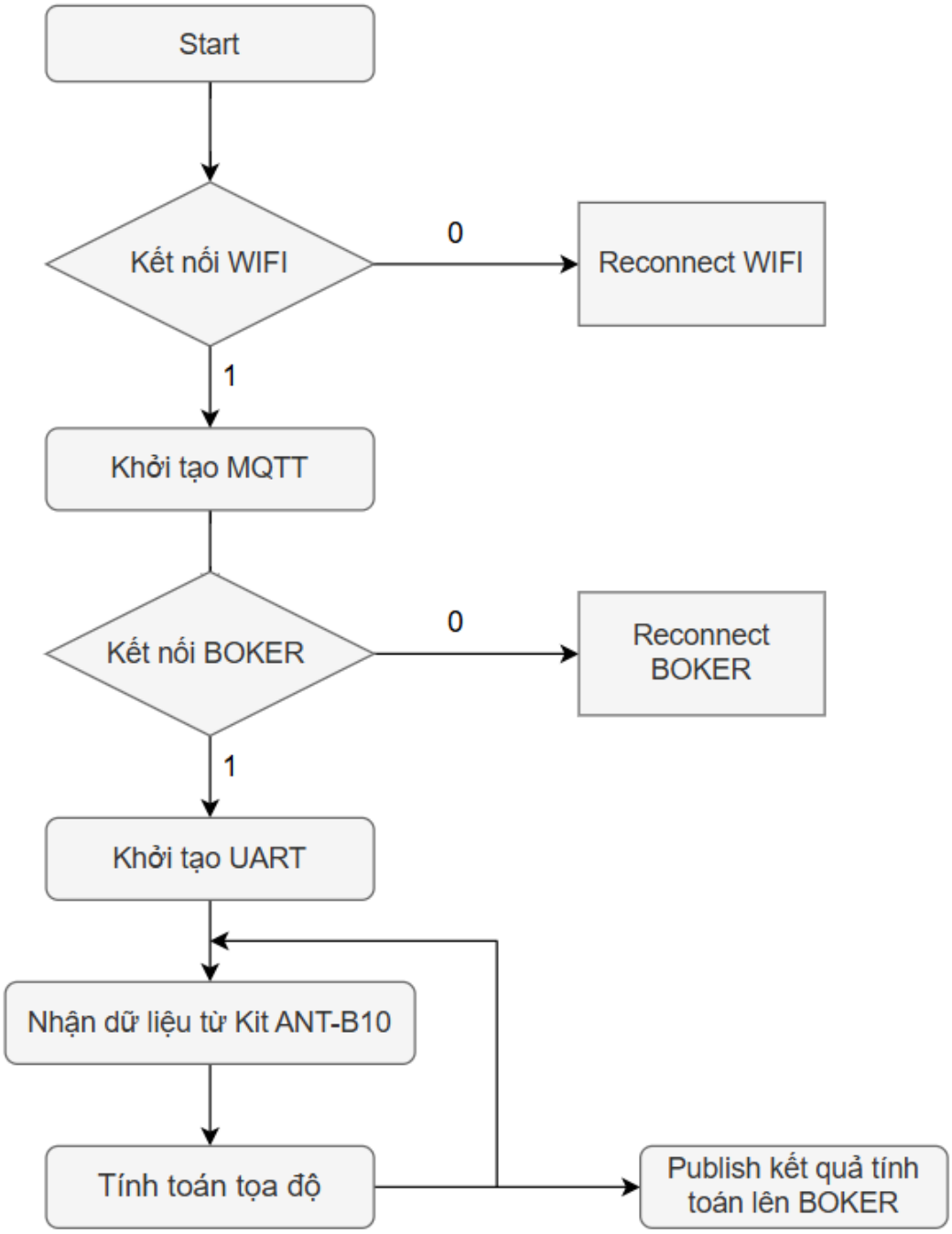


Hình 12. *Linh kiện phần cứng*

3. Lưu đồ giải thuật



***Hình 13**. Lưu đồ giải thuật cho hệ thống*

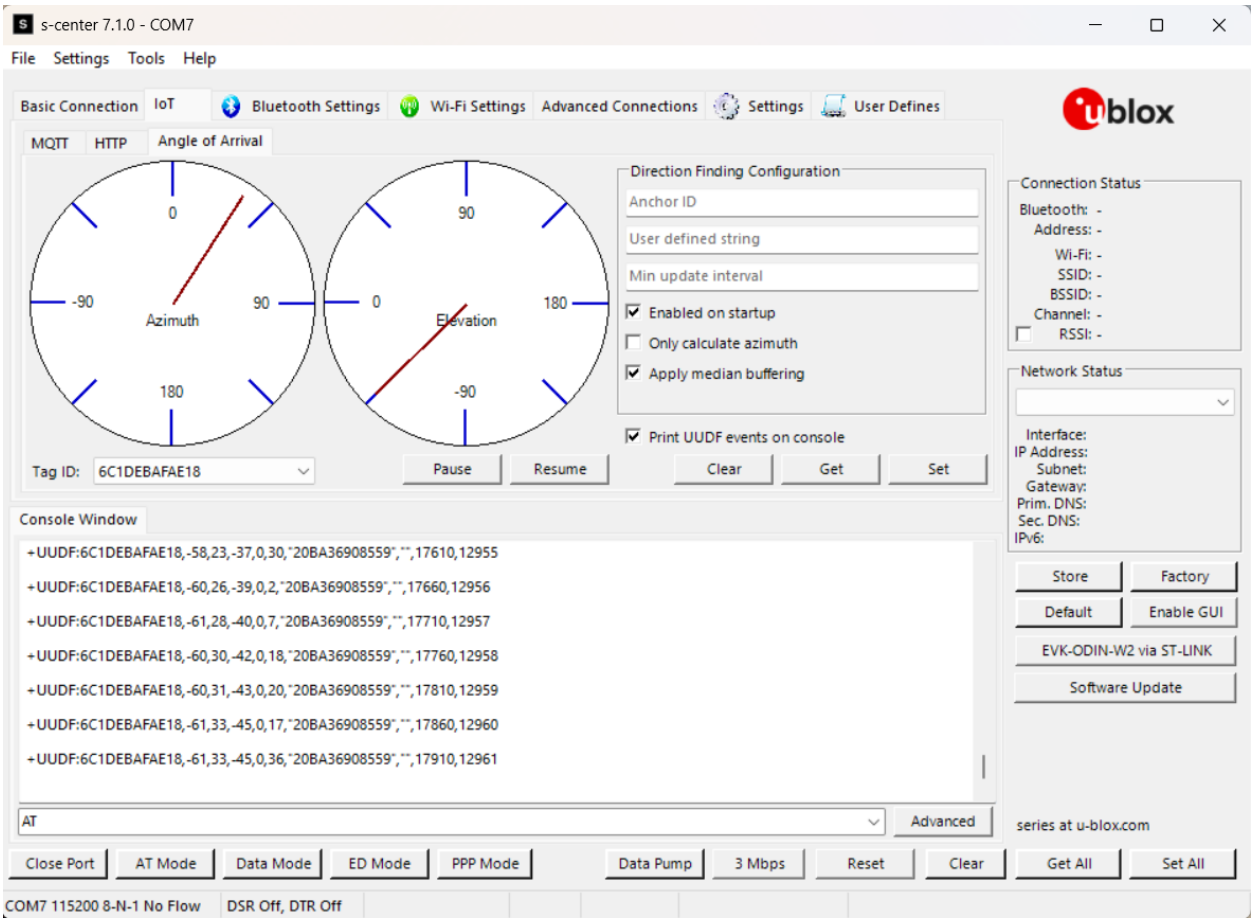


Hình 14. *Lưu đồ giải thuật phần mềm cho* ***ESP32***

4. Kết quả lập trình phần mềm

4.1. Kết quả thu được từ u-located

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



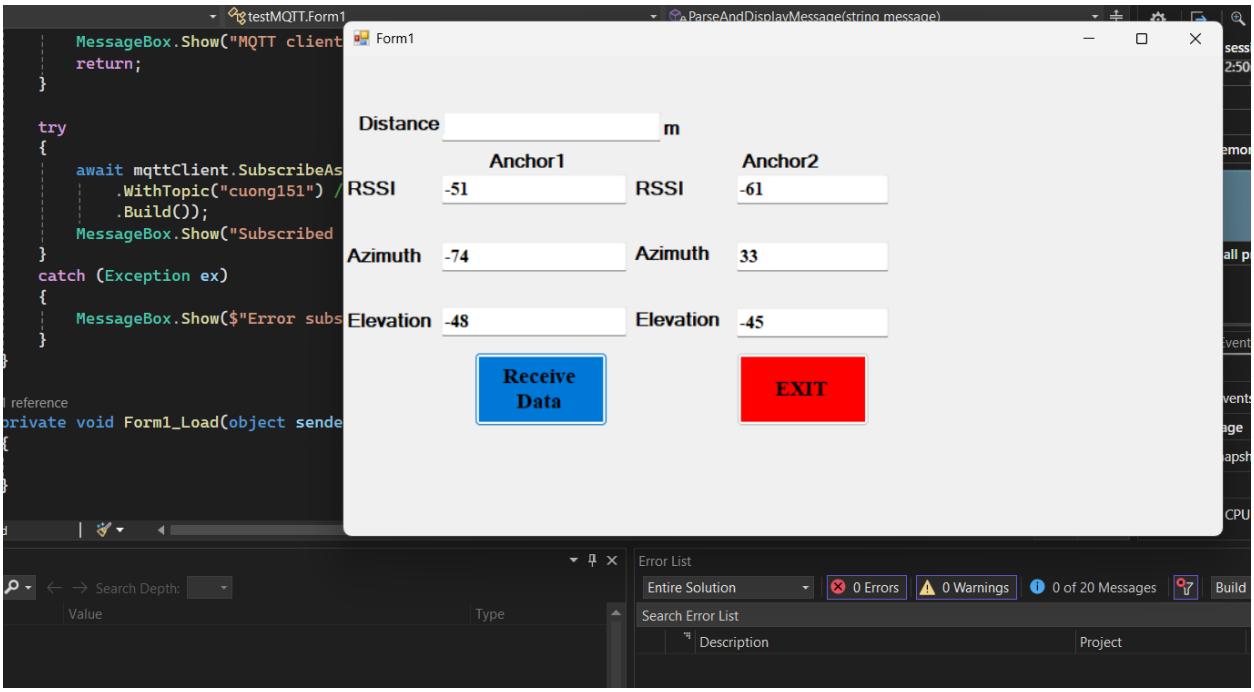
Hình 15. *Kết quả thực tế thu được từ phần mềm u-located*

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

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

4.2. Giao diện C# thu thập kết quả:

Đây là giao diện C# dùng để thu thập data được ESP32 publish lên MQTT Boker:



Hình 16. *Giao diện C# thu thập kết quả từ Boker*

5. Kết quả thực hiện

Dựa vào tín hiệu RSSI và các góc AoA thu được từ Anchor ta tiến hành tính khoảng cách và tọa độ dựa vào các công thức sau:

- Khoảng cách: $D = 10^{\frac{(TX_{Power}+RSSI+BASIC_{LOSS})}{2*PATH\_LOSS}}$

  Với: $TX_{Power} = 4(\text{dBm})$

  $BASIC_{LOSS} = 0.$

  PATH_LOSS = 2.0

- Tọa độ:

  x = distance * Sin(elevationRad) * Cos(azimuthRad)

  y = distance * Sin(elevationRad) * Sin(azimuthRad)

  z = distance * Cos(elevationRad)

Khoảng cách đo	Số lần đo	RSSI	Azimuth Angel	Elevation Angel
2m	1	-63	-10	-3
	2	-66	-8	-6
	3	-67	-9	-11
	4	-64	-8	-9
	5	-63	-11	-6
3m	1	-77	-26	-4
	2	-72	-24	-3
	3	-74	-22	-4
	4	-79	-23	-6
	5	-74	-23	-5

Khoảng cách đo	Số lần đo	Kết quả khoảng cách	% Sai số
2m	1	0.001m	99.95%
	2	0.0008m	99.996%
	3	0.0002m	99.99%
	4	0.001m	99.95%
	5	0.0007m	99.995%
3m	1	0.00032m	100%
	2	0.0002m	100%
	3	0.00032m	100%
	4	0.0004m	100%
	5	0.0002m	100%

$$\%\text{Sai số} = \frac{|Kết\ quả - khoảng\ cách\ đo|}{khoảng\ cách\ đo} \text{x} 100$$

- Đánh giá kết quả: Với kết quả trên ta nhận thấy sai số gần như là 100% bởi vì chỉ dùng 1 anchor và tính khoảng cách dựa trên RSSI. Trong môi trường tự nhiên RSSI sẽ bị hao hụt và gây ra sai số rất lớn do vậy ta tiến hành lựa chọn thông số $BASIC_{LOSS}$ phù hợp hơn.

- Chọn $BASIC_{LOSS}$ = 65(dBm).

Khoảng cách đo	Số lần đo	Kết quả khoảng cách	% Sai số
2m	1	1.995m	0.25%
	2	1.41m	29.5%
	3	1.26m	37%
	4	1.78m	11%
	5	1.995m	0.25%
3m	1	0.4m	86.66%
	2	0.71m	76.33%
	3	0.6m	80%
	4	0.32m	89.33%
	5	0.6m	80%

VI. Kết luận

1. Ứng dụng

Phương pháp định vị trong nhà dựa trên góc đến (AoA) mang lại nhiều tiềm năng ứng dụng trong thực tế. Trong lĩnh vực bán lẻ, AoA có thể hỗ trợ theo dõi vị trí của khách hàng để cung cấp dịch vụ cá nhân hóa, tối ưu hóa bố trí cửa hàng hoặc quản lý kho hàng thông minh. Trong chăm sóc sức khỏe, công nghệ này giúp định vị nhanh chóng thiết bị y tế hoặc theo dõi bệnh nhân trong các cơ sở lớn, đặc biệt là trong các tình huống khẩn cấp. Ngoài ra, AoA còn có thể được ứng dụng trong hệ thống dẫn đường trong nhà, như sân bay, trung tâm thương mại, hoặc bảo tàng, nhằm hướng dẫn người dùng đến các địa điểm mong muốn một cách chính xác và tiện lợi.

Trong công nghiệp và logistics, AoA được ứng dụng để giám sát vị trí chính xác của hàng hóa và phương tiện trong các kho bãi, giúp tối ưu hóa quy trình vận hành. Đặc biệt, trong các lĩnh vực yêu cầu độ chính xác cao như robot tự hành và thiết bị IoT, AoA đóng vai trò là nền tảng định vị để cải thiện hiệu suất và độ tin cậy của hệ thống. Với những ứng dụng đa dạng và tiềm năng này, AoA không chỉ mang lại giải pháp công nghệ vượt trội mà còn mở ra nhiều cơ hội đổi mới trong tương lai.

2. Hướng phát triển

Mặc dù phương pháp định vị trong nhà dựa trên góc đến (AoA) đã chứng minh tiềm năng lớn, vẫn còn nhiều khía cạnh cần được nghiên cứu và phát triển thêm. Một trong những hướng quan trọng là cải tiến thuật toán xử lý tín hiệu để nâng cao độ chính xác và độ bền vững của hệ thống trong các môi trường phức tạp, chẳng hạn như những nơi có nhiều nhiễu tín hiệu hoặc vật cản.

Ngoài ra, việc tích hợp phương pháp AoA với các công nghệ định vị khác như Time of Arrival (ToA), Received Signal Strength (RSS), hoặc sử dụng công nghệ trí tuệ nhân tạo (AI) để phân tích và tối ưu hóa dữ liệu tín hiệu có thể mở ra cơ hội tạo ra các hệ thống định vị lai (hybrid systems) mạnh mẽ hơn.

Hơn nữa, việc ứng dụng AoA trên các nền tảng phần cứng tiết kiệm năng lượng và thiết kế cảm biến tối ưu là một hướng phát triển quan trọng nhằm giảm chi phí triển khai và mở rộng khả năng ứng dụng trong các thiết bị IoT. Đồng thời, nghiên cứu về khả năng áp dụng AoA trong các môi trường phi truyền thống, như dưới nước hoặc các không gian hạn chế khác, cũng là một hướng đi thú vị và đầy tiềm năng.

Cuối cùng, việc xây dựng các tiêu chuẩn và giao thức chung để hỗ trợ triển khai AoA trong các hệ thống thương mại hóa cũng là một nhiệm vụ quan trọng nhằm đảm bảo tính tương thích, bảo mật và hiệu quả khi ứng dụng vào thực tế. Những hướng đi này không chỉ giúp hoàn thiện phương pháp AoA mà còn góp phần thúc đẩy sự phát triển của công nghệ định vị trong nhà trong tương lai.

TÀI LIỆU THAM KHẢO

[1] u-blox AG, "XPLR-AOA-3: Evaluation kit for ANT-B10 antenna boards - User Guide," Version R05, May 2023. [Online]. Available: https://www.u-blox.com/en/product/xplr-aoa-3-kit.

[2] u-blox AG, "u-locateEmbed: Short range standalone modules AT commands manual," Version R04, Sep 2024. [Online]. Available: https://www.u-blox.com/en/product/u-locateEmbed.

[3] u-blox AG, "ANT-B10: Direction Finding antenna board - System integration manual," Version R05, Jan 2024. [Online]. Available: https://www.u-blox.com/en/product/u-locateEmbed.

[4] ANT-B10 data sheet, UBX-22008373.

[5] u-blox tag software example, 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] "Angle of Arrival." *ScienceDirect*, n.d., https://www.sciencedirect.com/topics/engineering/angle-of-arrival.

PHỤ LỤC

Chương trình phần mềm cho ESP32

```
#include <WiFi.h>
#include <PubSubClient.h>
#define RX_PIN 16
#define TX_PIN 17
#define BAUD_RATE 115200
String dataBuffer = ""; // Biến lưu dữ liệu tạm thời
HardwareSerial MySerial(1);

// Thông tin WiFi
const char* WIFI_SSID = "PIFLab_M5";
const char* WIFI_PASSWORD = "khonghoisaocopass";

// Thông tin MQTT Broker
const char* MQTT_SERVER = "broker.hivemq.com";  // Broker
const int MQTT_PORT = 1883;
const char* MQTT_TOPIC_PUB = "cuong151";
const char* MQTT_TOPIC_SUB = "receive";

// Khai báo client
WiFiClient espClient;
PubSubClient mqttClient(espClient);

// Biến thời gian
unsigned long previousWiFiCheckTime = 0;
```

```
unsigned long previousPublishTime = 0;

// 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());
}

// Callback nhận tin nhắn
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("]: ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
```

```
  }
  Serial.println();
}

// Kết nối MQTT
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);
    }
  }
}

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

```
  mqttClient.setCallback(callback);
  mqttClient.setKeepAlive(15);

  // Cấu hình UART1 (TX = GPIO17, RX = GPIO16)
  MySerial.begin(115200, SERIAL_8N1, RX_PIN, TX_PIN); // Baudrate, 8 data bits, no parity, 1 stop bit
  Serial.println("UART1 initialized.");
}

bool isValidData(const String& data) {
  // Kiểm tra dữ liệu có bắt đầu bằng "+UUDF:" không
  if (!data.startsWith("+UUDF:")) {
    return false;
  }

  // Kiểm tra các trường cơ bản bằng cách đếm dấu phẩy
  int commaCount = 0;
  for (char c : data) {
    if (c == ',') {
      commaCount++;
    }
  }
  // Giả định một gói dữ liệu hợp lệ phải có ít nhất 9 dấu phẩy
  return commaCount >= 9;
}
```

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

  if (!mqttClient.connected()) {
    reconnectMQTT();
  }
  mqttClient.loop();

  // Đọc dữ liệu từ MySerial
  while (MySerial.available()) {
    char c = MySerial.read();
    if (c == '\n') { // Khi gặp ký tự kết thúc '\n'
      Serial.println("Received complete data: " + dataBuffer);
      if (isValidData(dataBuffer)) {
        Serial.println("Data is valid: " + dataBuffer);
    unsigned long currentMillis = millis();
    if (currentMillis - previousPublishTime > 2000) {
    previousPublishTime = currentMillis;
    if (dataBuffer.length() > 0 ) { // Chỉ gửi nếu có dữ liệu đầy đủ và hợp lệ
      mqttClient.publish(MQTT_TOPIC_PUB, dataBuffer.c_str()); // Gửi chuỗi data
qua MQTT
      Serial.println("Sent valid data via MQTT: " + dataBuffer);
      dataBuffer = ""; // Xóa buffer sau khi gửi
    } else {
```

```
      Serial.println("No valid data to send via MQTT.\n");
    }
     } else {
       Serial.println("Invalid data format, discarding: " + dataBuffer);
     }
  }
     dataBuffer = ""; // Xóa buffer sau khi xử lý
    } else {
     dataBuffer += c; // Ghép ký tự vào buffer
    }
   delay(10);
  }
}
```

Chương trình giao diện C#

```
using System;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MQTTnet;
using MQTTnet.Client;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace testMQTT
{
   public partial class Form1 : Form
   {
      private string rssiValue;       // Biến lưu RSSI
      private string azimuthValue;   // Biến lưu Azimuth
      private string elevationValue; // Biến lưu Elevation
      private string rssiValue1;       // Biến lưu RSSI
      private string azimuthValue1;   // Biến lưu Azimuth
      private string elevationValue1; // Biến lưu Elevation
      private IMqttClient mqttClient;

      public Form1()
      {
         InitializeComponent();
```

```
        InitializeMQTT();
    }

    private async void InitializeMQTT()
    {
        var factory = new MqttFactory();
        mqttClient = factory.CreateMqttClient();

        var options = new MqttClientOptionsBuilder()
            .WithClientId("MQTTClient")
            .WithTcpServer("broker.hivemq.com", 1883) // Thay broker
            .WithCleanSession()
            .Build();



        mqttClient.DisconnectedAsync += e =>
        {
            MessageBox.Show("Disconnected from MQTT broker.");
            return Task.CompletedTask;
        };

        mqttClient.ApplicationMessageReceivedAsync += e =>
        {
            string message =
Encoding.UTF8.GetString(e.ApplicationMessage.PayloadSegment.ToArray());
```

```
            ParseAndDisplayMessage(message);
            return Task.CompletedTask;
        };

        await mqttClient.ConnectAsync(options);
    }

    private void ParseAndDisplayMessage(string message)
    {
        try
        {
            // Tách các dòng thông điệp
            var messages = message.Split('\n'); // Tách chuỗi thành từng dòng nếu có nhiều thông điệp

            foreach (var msg in messages)
            {
                var parts = msg.Split(',');

                if (parts.Length > 6)
                {
                    // Xử lý riêng cho chuỗi chứa ID "20BA369085F8"
                    if (parts[6].Contains("20BA369085F8"))
                    {
                        // Trích xuất giá trị cho ID "20BA369085F8"
                        double rssi = double.Parse(parts[1].Trim());
```

```
            double azimuth = double.Parse(parts[2].Trim());
            double elevation = double.Parse(parts[3].Trim());

            // Tính khoảng cách

            // Hiển thị kết quả lên TextBox
            Invoke(new Action(() =>
            {
                textBox2.Text = rssi.ToString();
                textBox3.Text = azimuth.ToString();
                textBox4.Text = elevation.ToString();
            }));
        }

        // Xử lý riêng cho chuỗi chứa ID "20BA36908559"
        else if (parts[6].Contains("20BA36908559"))
        {
            // Trích xuất giá trị cho ID "20BA36908559"
            double rssi1 = double.Parse(parts[1].Trim());
            double azimuth1 = double.Parse(parts[2].Trim());
            double elevation1 = double.Parse(parts[3].Trim());
            // Hiển thị kết quả lên TextBox
            Invoke(new Action(() =>
            {
                textBox5.Text = rssi1.ToString();
```

```
                    textBox6.Text = azimuth1.ToString();
                    textBox7.Text = elevation1.ToString();
                }));
            }


        }
    }
    catch (Exception ex)
    {
        // Xử lý lỗi nếu có
        MessageBox.Show($"Error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }


}
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}


private async void button1_Click(object sender, EventArgs e)
{
    if (mqttClient == null || !mqttClient.IsConnected)
    {
```

```
            MessageBox.Show("MQTT client is not connected. Please wait or check
the connection.");
            return;
        }

        try
        {
            await mqttClient.SubscribeAsync(new MqttTopicFilterBuilder()
                .WithTopic("cuong151") // Thay topic
                .Build());
            MessageBox.Show("Subscribed to MQTT topic and waiting for data.");
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Error subscribing to topic: {ex.Message}");
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {
```

```
}

private void textBox2_TextChanged(object sender, EventArgs e)
{

}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void label1_Click(object sender, EventArgs e)
{

}

private void textBox1_TextChanged_1(object sender, EventArgs e)
{

}

private void textBox5_TextChanged(object sender, EventArgs e)
{
  }}}
```