ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH KĨ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

THIẾT KẾ BỘ CHUYỂN ĐỔI GIAO THỨC TRUYỀN THÔNG TỪ MODBUS RTU SANG LORA TRONG MẠNG IOT CÔNG NGHIỆP

NGƯỜI HƯỚNG DẪN: **TS. NGÔ ĐÌNH THANH**

SINH VIÊN THỰC HIỆN:

1. **TRẦN BẢO LƯU - MSSV:105210322 - LỚP: 21TDH2**
2. **HỒ HOÀNG MINH VŨ – MSSV:105210302 - LỚP: 21TDH1**

ĐÀ NẴNG, 6/2025

TÓM TẮT

Tên đề tài: Thiết kế bộ chuyển đổi giao thức truyền thông từ Modbus RTU sang Lora trong mạng IoT công nghiệp.

STT	Họ và tên sinh viên thực hiện	Số thẻ sinh viên	Lớp
1	Trần Bảo Lưu	105210322	21TDH2
2	Hồ Hoàng Minh Vũ	105210302	21TDH1

Ngày nay, các hệ thống truyền thông công nghiệp đóng vai trò quan trọng trong việc kết nối và điều khiển các thiết bị tại hiện trường, đặc biệt trong các hệ thống tự động hóa, giám sát và điều khiển từ xa. Trong nông nghiệp thông minh, việc giám sát các thông số môi trường như mực nước, độ ẩm đất, nhiệt độ,... đòi hỏi khả năng thu thập và truyền dữ liệu từ các cảm biến đặt ngoài hiện trường về trung tâm xử lý một cách hiệu quả, ổn định và tiêu thụ năng lượng thấp.

Các cảm biến công nghiệp thường sử dụng giao thức Modbus RTU với đường truyền vật lý RS485, nổi bật bởi tính ổn định và khả năng chống nhiễu tốt trong môi trường khắc nghiệt. Tuy nhiên, giao thức này bị giới hạn về khoảng cách truyền (khoảng vài trăm mét), gây trở ngại lớn trong các ứng dụng ngoài trời có quy mô rộng như các cánh đồng, trang trại, hồ thủy lợi,... Trong khi đó, LoRa là công nghệ truyền thông không dây nổi bật với khả năng truyền xa hàng kilomet, tiêu thụ điện năng thấp và dễ triển khai ở vùng sâu vùng xa – rất phù hợp với điều kiện địa hình và hạ tầng ở nông thôn.

Từ yêu cầu đó, nhóm chúng em thực hiện đề tài "Thiết kế bộ chuyển đổi giao thức truyền thông từ Modbus RTU sang Lora trong mạng IoT công nghiệp. Mục tiêu của đề tài là xây dựng một thiết bị chuyển đổi nhỏ gọn, tiêu thụ năng lượng thấp, có khả năng:

- Giao tiếp với các cảm biến công nghiệp chuẩn Modbus RTU để lấy dữ liệu.
- Mã hóa và truyền dữ liệu qua mạng LoRa đến gateway trung tâm.

Đồ án này gồm những nội dung chính sau:

- Chương 1: Tổng quan
- Chương 2: Thiết kế bộ chuyển đổi giao thức truyền thông
- Chương 3: Đánh giá, kết quả và nhận xét.

Phụ lục 02

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	TRẦN BẢO LƯU	105210322	21TDH2	KĨ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA
2	HỒ HOÀNG MINH VŨ	105210302	21TDH1	KĨ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

1. *Tên đề tài đồ án* Thiết kế bộ chuyển đổi giao thức truyền thông từ Modbus RTU sang Lora trong mạng IoT công nghiệp.
2. *Đề tài thuộc diện: ☐Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:*

……..……..……..

4. *Nội dung các phần thuyết minh và tính toán:*

a. Phần chung:

Họ tên sinh viên	Nội dung
Trần Bảo Lưu Hồ Hoàng Minh Vũ	- Nghiên cứ đề tài, tìm tài liệu, báo cáo, bài báo nghiên cứu liên quan - Thành lập định dạng dữ liệu trao đổi giữa các phần với nhau - Thiết kế, kiểm tra và chỉnh sửa lỗi của thiết bị - Hoàn thành đánh giá kết quả, báo cáo đồ án

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
	Trần Bảo Lưu	- Tìm hiểu giao thức Modbus RTU, truyền thông không dây LoRa - Tìm hiểu về cảm biến đo mực nước - Thiết kế phương pháp chuyển đổi giao thức Modbus RTU → LoRa - Lập trình giao tiếp với cảm biến - Lập trình truyền dữ liệu qua LoRa - Thống kê, phân tích kết quả và đánh giá hệ thống -

TT	Họ tên sinh viên	Nội dung
	Hồ Hoàng Minh Vũ	- Tìm hiểu giao thức Modbus RTU, truyền thông không dây LoRa - Tìm hiểu về cảm biến đo mực nước - Thiết kế phương pháp chuyển đổi giao thức Modbus RTU → LoRa - Lập trình giao tiếp với cảm biến - Lập trình truyền dữ liệu qua LoRa - Thống kê, phân tích kết quả và đánh giá hệ thống

5. *Ngày giao nhiệm vụ đồ án: 28/04/2025.*
6. *Ngày hoàn thành đồ án: 25/06/2025.*

Đà Nẵng, ngày 17 tháng 6 năm 2025

Trưởng Bộ môn Tự động hóa | **Người hướng dẫn**

TS. Giáp Quang Huy | TS. Ngô Đình Thanh

Phụ lục 03

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ tên sinh viên: TRẦN BẢO LƯU Số thẻ SV : 105210322 Lớp: 21TDH2

Họ tên sinh viên: HỒ HOÀNG MINH VŨ Số thẻ SV : 105210302 Lớp: 21TDH1

Tên đề tài ĐATN: Thiết kế bộ chuyển đổi giao thức truyền thông từ Modbus RTU sang Lora trong mạng IoT công nghiệp

Họ tên người HD: TS. Ngô Đình Thanh, Đơn vị: Khoa Điện Trường Đại học Bách khoa – Đại học Đà Nẵng.

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	28/4	Nhận đề tài và nghiên cứu cơ bản về đề tài (10%)	Nghiên cứu tài liệu, phân tích yêu cầu, xây dựng hướng thiết kế	
2	5/5	Tìm hiểu giao thức Modbus và truyền thông không dây Lora.(20%)	Xây dựng phương pháp chuyển đổi giao thức Modbus sang Lora	
3	12/5	Duyệt lần 1: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
5	19/5	Tìm hiểu phần mềm và các ngôn ngữ lập trình (30%)	Lập trình trên bộ chuyển đổi giao thức Modbus RTU - Lora	
6	26/5	Thử nghiệm chương trình (50%)	Đánh giá kết quả và sửa lỗi	
7	31/5	Duyệt lần 2: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
8	2/6	Hoàn thiện và tổng hợp kết quả đề tài (70%)	Kiểm tra lại các phần và thực hiện viết báo cáo hoàn chỉnh	
9	9/6	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
10	16/6	Hoàn thiện báo cáo và kiểm tra đầy đủ các phần (95%)	Chuẩn bị bảo vệ ĐATN	
11	24/6	Bảo vệ ĐATN, hoàn thành nhiệm vụ		

LỜI CẢM ƠN

Lời nói đầu tiên chúng em xin gửi lời cảm ơn đến các thầy, cô trong Khoa Điện . Các thầy, cô trong Trường Đại Học Bách Khoa, Đại Học Đà Nẵng đã nhiệt tình giảng dạy, chỉ dẫn và tạo điều kiện giúp đỡ chúng em trong quá trình học tập và làm đồ án tốt nghiệp. Đặc biệt chúng em xin gửi lời tri ân và lời cám ơn sâu sắc đến thầy Ngô Đình Thanh đã tận tình giúp đỡ trực tiếp chỉ bảo, hưỡng dẫn chúng em trong quá trình làm đồ án tốt nghiệp. Trong thời gian được thầy hướng dẫn, chúng em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học được tinh thần làm việc củng như thái độ nghiên cứu đề tài nghêm túc, hiệu quả, đây là những điều cần thiết cho chúng em trong quá trình học tập và công tác sau này.

Chúng em xin gửi lời cám ơn tới gia đinh, người thân và bạn bè đã luôn động viên giúp đỡ chúng em trong suốt quá trình học tập tại Khoa Điện, Trường Đại Học Bách Khoa cũng như trong quá trình thực hiện đồ án tốt nghiệp.

Đề tài nghiên cứu được thực hiện dựa trên các kiến thức được học ở trường, các kiến thức thực tế được thầy cô giảng dạy, chỉ dẫn và tìm tòi các kênh thông tin. Do khả năng bản thân còn nhiều hạn chế nên không tránh khỏi những thiếu sót trong quá trình thực hiện nghiên cứu. Kính mong sự đóng góp ý kiến quý báu của thầy cô để đề tài của chúng em được hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

Đà Nẵng, ngày 17 tháng 06 năm 2025

Nhóm sinh viên thực hiện

Trần Bảo Lưu Hồ Hoàng Minh Vũ

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Chúng em xin cam đoan trong quá trình làm đồ án tốt nghiệp sẽ thực hiện nghiêm túc các quy định về liêm chính học thuật:

- Không gian lận, bịa đặt, đạo văn, giúp người học khác vi phạm.
- Trung thực trong việc trình bày, thể hiện các hoạt động học thuật và kết quả từ hoạt động học thuật của bản thân.
- Không giả mạo hồ sơ học thuật.
- Không dùng các biện pháp bất hợp pháp hoặc trái quy định để tạo nên ưu thế cho bản thân.
- Chủ động tìm hiểu và tránh các hành vi vi phạm liêm chính học thuật, chủ động tìm hiểu và nghiêm túc thực hiện các quy định về luật sở hữu trí tuệ.

Chúng em xin cam đoan số liệu và kết quả thực hiện trong đồ án này là trung thực và chưa hề được dùng để bảo vệ một học vị nào. Mọi sự giúp đỡ cho đồ án này đã được cảm ơn và các thông tin trích dẫn trong đồ án đã được trích rõ nguồn gốc rõ ràng và được phép công bố.

Chúng em hoàn toàn chịu trách nhiệm trước nhà trường và pháp luật về tính trung thực và nguyên bản của báo cáo này.

Đà Nẵng, ngày 17 tháng 06 năm 2025

Người cam đoan

Trần Bảo Lưu Hồ Hoàng Minh Vũ

(Ký tên và ghi rõ họ tên)

MỤC LỤC

DANH SÁCH BẢNG

DANH SÁCH HÌNH ẢNH

MỞ ĐẦU

Trong bối cảnh công nghiệp 4.0 đang phát triển mạnh mẽ, việc tích hợp các thiết bị đầu cuối vào hệ thống giám sát và điều khiển từ xa thông qua mạng IoT đã trở thành một xu hướng tất yếu. Các cảm biến và thiết bị đo lường trong môi trường sản xuất, nông nghiệp, hạ tầng và năng lượng thường sử dụng chuẩn giao tiếp công nghiệp Modbus RTU nhờ tính ổn định, chi phí thấp và khả năng chống nhiễu cao. Tuy nhiên, hạn chế lớn của Modbus RTU là khoảng cách truyền dẫn ngắn và yêu cầu kết nối vật lý, gây khó khăn trong việc triển khai ở các khu vực rộng lớn hoặc vùng xa.

Trong khi đó, LoRa là một công nghệ truyền thông không dây nổi bật với khả năng truyền dữ liệu khoảng cách xa, tiêu thụ năng lượng thấp, rất phù hợp với các ứng dụng IoT ngoài trời, nơi hạ tầng mạng còn hạn chế. Việc kết hợp giữa Modbus RTU và LoRa sẽ mở ra khả năng kết nối không dây cho các thiết bị công nghiệp truyền thống, cho phép truyền dữ liệu từ xa đến trung tâm giám sát mà không cần hạ tầng cáp phức tạp.

Từ nhu cầu đó, đề tài "Thiết kế bộ chuyển đổi giao thức truyền thông từ Modbus RTU sang Lora trong mạng IoT công nghiệp" được thực hiện nhằm xây dựng một giải pháp phần cứng và phần mềm tối ưu, cho phép thu thập dữ liệu từ các cảm biến Modbus RTU và truyền đi qua mạng LoRa. Thiết bị được thiết kế hướng đến tính nhỏ gọn, tiết kiệm năng lượng và độ ổn định cao, có thể ứng dụng trong nhiều lĩnh vực như: quan trắc môi trường, giám sát nông nghiệp, điều khiển từ xa trong công nghiệp, v.v.
Thông qua đề tài này, người thực hiện mong muốn đóng góp một giải pháp truyền thông hiệu quả, có khả năng tích hợp dễ dàng với các hệ thống IoT hiện đại, đồng thời thúc đẩy khả năng ứng dụng công nghệ truyền thông không dây vào các hệ thống giám sát và điều khiển thực tế.

CHƯƠNG I: TỔNG QUAN

1.1. Giới thiệu chương

Chương này đóng vai trò là nền tảng định hướng cho toàn bộ nội dung nghiên cứu. Trước tiên, chương trình bày bối cảnh phát triển công nghệ số và nhu cầu ngày càng cao về thu thập và giám sát dữ liệu từ xa trong các lĩnh vực công nghiệp, nông nghiệp và đô thị thông minh. Tiếp đó, chương phân tích các hạn chế kỹ thuật của giao thức Modbus RTU trong các ứng dụng hiện trường và chỉ ra tiềm năng của công nghệ truyền thông không dây LoRa như một giải pháp thay thế hiệu quả. Từ mâu thuẫn giữa nhu cầu thực tiễn và giới hạn kỹ thuật, đề tài nghiên cứu "Bộ chuyển đổi giao thức Modbus RTU sang LoRa" được đặt ra như một giải pháp trung gian, khả thi và phù hợp với điều kiện triển khai thực tế.

Chương cũng trình bày cụ thể mục tiêu nghiên cứu, phạm vi triển khai, phương pháp thực hiện cũng như ý nghĩa khoa học và ứng dụng thực tiễn của đề tài. Cuối cùng, một sơ lược về cấu trúc khóa luận sẽ giúp người đọc có cái nhìn tổng quan về các phần tiếp theo, từ cơ sở lý thuyết đến quá trình thiết kế, thử nghiệm và đánh giá hệ thống.

1.2. Đặt vấn đề

Trong bối cảnh phát triển mạnh mẽ của công nghệ số và xu hướng tự động hóa, nhu cầu thu thập, giám sát và truyền dữ liệu từ các thiết bị cảm biến trở nên ngày càng quan trọng và phổ biến trong nhiều lĩnh vực: từ công nghiệp, nông nghiệp, môi trường cho đến giao thông và đô thị thông minh. Đặc biệt, trong các ứng dụng giám sát từ xa như theo dõi mực nước, độ ẩm đất, chất lượng không khí hay điều kiện vận hành thiết bị, việc đảm bảo kết nối tin cậy giữa cảm biến và hệ thống xử lý trung tâm là điều kiện tiên quyết để hệ thống hoạt động hiệu quả.

Hiện nay, một phần lớn cảm biến công nghiệp đang sử dụng giao thức Modbus RTU truyền qua chuẩn vật lý RS485. Giao thức này có ưu điểm về độ tin cậy, độ phổ biến cao và dễ tích hợp trong môi trường có dây. Tuy nhiên, nó cũng tồn tại các hạn chế lớn, đặc biệt là về khoảng cách truyền và chi phí triển khai.

Theo báo cáo của Hiệp hội Tự động hóa Việt Nam (2023), hơn 65% cảm biến công nghiệp tại Việt Nam sử dụng giao thức Modbus RTU qua chuẩn RS485. Mặc dù có ưu điểm về độ tin cậy và tính phổ biến, công nghệ này tồn tại hai hạn chế nghiêm trọng:

- Giới hạn khoảng cách truyền dẫn (tối đa 1.2km theo chuẩn TIA/EIA-485)
- Chi phí triển khai cáp cao (≥15.000 VND/mét).

Thực tế triển khai tại các khu vực hẻo lánh như hệ thống thoát nước đô thị (Hà Nội, TP.HCM) hay vùng nông nghiệp (Đồng bằng sông Cửu Long) cho thấy: Việc kéo dây RS485 gặp khó khăn do địa hình phức tạp, nguy cơ hư hỏng thiết bị do ngập nước, và chi phí bảo trì có thể chiếm tới 40% tổng đầu tư ban đầu.

Giải pháp thay thế là công nghệ truyền thông không dây tầm xa LoRa (Long Range) hoạt động ở băng tần Sub-GHz (920-923MHz tại Việt Nam), cho phép truyền dữ liệu 10-15km ở khu vực ngoại vi với công suất tiêu thụ chỉ khoảng 10mA khi truyền. Công nghệ này rất phù hợp cho các hệ thống giám sát phân tán, nơi cảm biến được đặt cách xa trung tâm điều khiển và cần truyền dữ liệu định kỳ hoặc theo sự kiện.

Tuy nhiên, cảm biến Modbus RTU không tương thích trực tiếp với LoRa do khác biệt về nguyên lý hoạt động và định dạng truyền thông. Điều này đặt ra nhu cầu cấp thiết phải xây dựng một thiết bị trung gian bộ chuyển đổi giao thức từ Modbus RTU sang LoRa để làm cầu nối giữa thiết bị đo truyền thống và hệ thống truyền thông hiện đại.

Thực tế cho thấy, hiện vẫn còn nhiều khu vực sử dụng cảm biến Modbus RTU như một phần của hạ tầng đã đầu tư. Việc thay thế toàn bộ thiết bị là không khả thi về mặt kinh tế. Vì vậy, nếu có thể phát triển một bộ chuyển đổi giao thức phù hợp, ta không chỉ tận dụng được hệ thống cảm biến hiện có mà còn nâng cao khả năng truyền dữ liệu linh hoạt hơn, hỗ trợ tốt cho quá trình chuyển đổi số trong nhiều ngành nghề.

Do đó, việc xây dựng một bộ chuyển đổi giao thức giữa Modbus RTU và LoRa không chỉ giải quyết bài toán kết nối giữa hai chuẩn truyền thông phổ biến mà còn góp phần mở rộng khả năng ứng dụng của các cảm biến hiện có, giúp tiết kiệm chi phí và nâng cao hiệu quả trong các hệ thống giám sát từ xa.

Nhiều sản phẩm thương mại trên thị trường đã được phát triển để giải quyết bài toán này. Chẳng hạn, bộ chuyển đổi Modbus RTU sang LoRa của Milesight (UG67 LoRaWAN Gateway kết hợp với UC11-T1 RS485 Sensor Node) được sử dụng trong các hệ thống đo mực nước, điện năng và giám sát nông nghiệp thông minh. Tại Ấn Độ, Cascademic đã triển khai hệ thống cảm biến độ ẩm đất và mưa sử dụng Modbus RTU, truyền dữ liệu qua LoRaWAN để hỗ trợ dự báo nông nghiệp.

Trong các nghiên cứu học thuật, một mô hình giám sát rung chấn cầu tại Indonesia đã ứng dụng cảm biến gia tốc sử dụng Modbus RTU kết hợp với vi điều khiển STM32 và mô-đun LoRa. Kết quả cho thấy hệ thống có độ trễ truyền thấp, sai số nhỏ và ổn định tín hiệu tốt trong điều kiện ngoài trời. Một nghiên cứu tại Đại học Quốc gia TP. Hồ Chí Minh đã xây dựng hệ thống giám sát bụi mịn (PM2.5), CO2 và độ ẩm sử dụng cảm biến Modbus RTU và truyền dữ liệu qua LoRa, thu được chỉ số RSSI ổn định và độ tin cậy cao.

Các kết quả này cho thấy việc kết hợp Modbus RTU với LoRa thông qua một bộ chuyển đổi là hoàn toàn khả thi và có giá trị ứng dụng cao. Tuy nhiên, các thiết bị thương mại thường có giá thành cao hoặc thiếu tính tùy biến theo từng loại cảm biến cụ thể. Vì vậy, đề tài nghiên cứu và phát triển một bộ chuyển đổi giao thức có thể tùy biến linh hoạt theo ứng dụng thực tế là rất cần thiết.

1.3. Các vấn đề cần giải quyết

Mặc dù việc tích hợp giữa Modbus RTU và LoRa mở ra nhiều tiềm năng trong giám sát từ xa, quá trình này cũng đối mặt với một số thách thức kỹ thuật và triển khai đáng kể:

- Không tương thích về giao thức: Modbus RTU là giao thức truyền tuần tự sử dụng chuẩn truyền dẫn vật lý RS485, trong khi LoRa sử dụng phương thức truyền không dây theo mô hình không đồng bộ và không có cơ chế phản hồi mặc định. Việc đồng bộ hóa thời gian truyền, định dạng khung dữ liệu, và xử lý lỗi trở thành vấn đề kỹ thuật quan trọng.
- Độ trễ và mất gói tin: Giao tiếp Modbus yêu cầu phản hồi trong khoảng thời gian rất ngắn (thường < 500ms), nhưng LoRa có độ trễ cao và có thể bị mất gói trong môi trường nhiễu. Việc đảm bảo tính toàn vẹn dữ liệu truyền qua LoRa từ các thiết bị Modbus là thách thức đáng kể.
- Giới hạn về dung lượng gói tin LoRa: LoRa chỉ hỗ trợ truyền tải lượng dữ liệu nhỏ (thường < 255 byte mỗi gói), trong khi một số thiết bị Modbus RTU trả về khối dữ liệu lớn. Cần có cơ chế chia nhỏ, ghép nối và xử lý dữ liệu phù hợp.
- Quản lý năng lượng và chu kỳ hoạt động: Các bộ chuyển đổi hoạt động ở vùng sâu vùng xa thường dùng pin hoặc năng lượng mặt trời. Tuy nhiên, hoạt động đọc Modbus và truyền LoRa đều tiêu tốn năng lượng. Việc tối ưu hóa thời gian đánh thức, chu kỳ đo và truyền là rất quan trọng để kéo dài thời gian vận hành.
- Khả năng cấu hình linh hoạt: Các cảm biến Modbus có thể dùng địa chỉ, tốc độ baud và định dạng dữ liệu khác nhau. Một bộ chuyển đổi lý tưởng cần hỗ trợ cấu hình động hoặc tự động phát hiện thiết bị, tránh việc phải can thiệp thủ công phức tạp.
- Chi phí sản xuất và triển khai: Một thiết bị trung gian hiệu quả cần đảm bảo giá thành hợp lý để có thể nhân rộng trong các hệ thống hiện trường quy mô lớn. Nếu chi phí thiết bị vượt quá mức đầu tư cho hạ tầng dây, hiệu quả kinh tế sẽ bị hạn chế.

1.4. Mục tiêu

Mục tiêu của đề tài là thiết kế và hiện thực một thiết bị nhúng có khả năng:

- Giao tiếp với thiết bị cảm biến sử dụng Modbus RTU qua chuẩn RS485.
- Đọc, phân tích và xử lý dữ liệu từ cảm biến theo chuẩn Modbus.
- Đóng gói dữ liệu và truyền không dây qua giao thức LoRa.
- Hỗ trợ truyền dữ liệu trong điều kiện triển khai ngoài trời, năng lượng giới hạn.
- Cho phép mở rộng quy mô triển khai với chi phí thấp, dễ bảo trì và thay thế.

1.5 Phạm vi nghiên cứu và giới hạn của đề tài

Phạm vi đề tài bao gồm:

- Nghiên cứu nguyên lý hoạt động của giao thức Modbus RTU và LoRa.
- Thiết kế phần cứng: vi điều khiển, mạch chuyển đổi RS485, mô-đun LoRa.
- Lập trình vi điều khiển để điều khiển truyền nhận dữ liệu.
- Thử nghiệm với một cảm biến Modbus phổ biến (ví dụ: cảm biến đo mức nước siêu âm).
- Kiểm tra hiệu năng truyền dữ liệu qua LoRa trong các môi trường thực tế.

1.6. Phương pháp nghiên cứu và triển khai

- Tổng hợp và phân tích tài liệu kỹ thuật về giao thức Modbus RTU và LoRa.
- Thiết kế mạch nguyên lý, lựa chọn linh kiện phù hợp, mô phỏng hoạt động.
- Lập trình và kiểm thử phần mềm nhúng trên vi điều khiển (STM32 hoặc tương đương).
- Thực hiện đo đạc các chỉ số hiệu năng: độ trễ, tỷ lệ lỗi gói, cường độ tín hiệu (RSSI), tiêu thụ điện năng.
- Đánh giá và so sánh với một số mô hình có sẵn hoặc giải pháp tương tự.

1.7 Ý nghĩa khoa học thực tiễn

Về mặt khoa học, đề tài góp phần cụ thể hóa quá trình tích hợp giữa hai giao thức truyền thông có nguyên lý hoạt động khác nhau, đồng thời đề xuất giải pháp đóng gói dữ liệu hiệu quả để truyền qua kênh LoRa với giới hạn băng thông. Ngoài ra, đề tài cũng kiểm nghiệm khả năng hoạt động của hệ thống trong điều kiện thực tế, từ đó đưa ra các khuyến nghị kỹ thuật có giá trị ứng dụng lâu dài.

Về mặt thực tiễn, sản phẩm từ đề tài có thể triển khai tại các khu vực cần quan trắc mực nước, độ ẩm, nhiệt độ… mà không cần đầu tư hạ tầng mạng phức tạp. Điều này giúp tiết kiệm chi phí, nâng cao hiệu quả giám sát, đồng thời tận dụng được các cảm biến Modbus hiện có mà không cần thay thế.

1.8 Kết luận chương

Chương này đã trình bày bối cảnh, lý do và sự cần thiết của việc xây dựng bộ chuyển đổi giao thức từ Modbus RTU sang LoRa trong bối cảnh ứng dụng giám sát từ xa ngày càng phát triển. Qua việc phân tích các hạn chế của phương pháp truyền thống cũng như lợi ích mà công nghệ LoRa mang lại, chương này đã làm rõ tính cấp thiết và tiềm năng ứng dụng thực tiễn của đề tài. Đồng thời, chương cũng đã nêu rõ mục tiêu, phạm vi, phương pháp nghiên cứu cũng như cấu trúc tổng thể của khóa luận. Những nội dung này sẽ là nền tảng cho các chương tiếp theo, nơi đi sâu vào phân tích kỹ thuật và hiện thực giải pháp đã đề xuất.

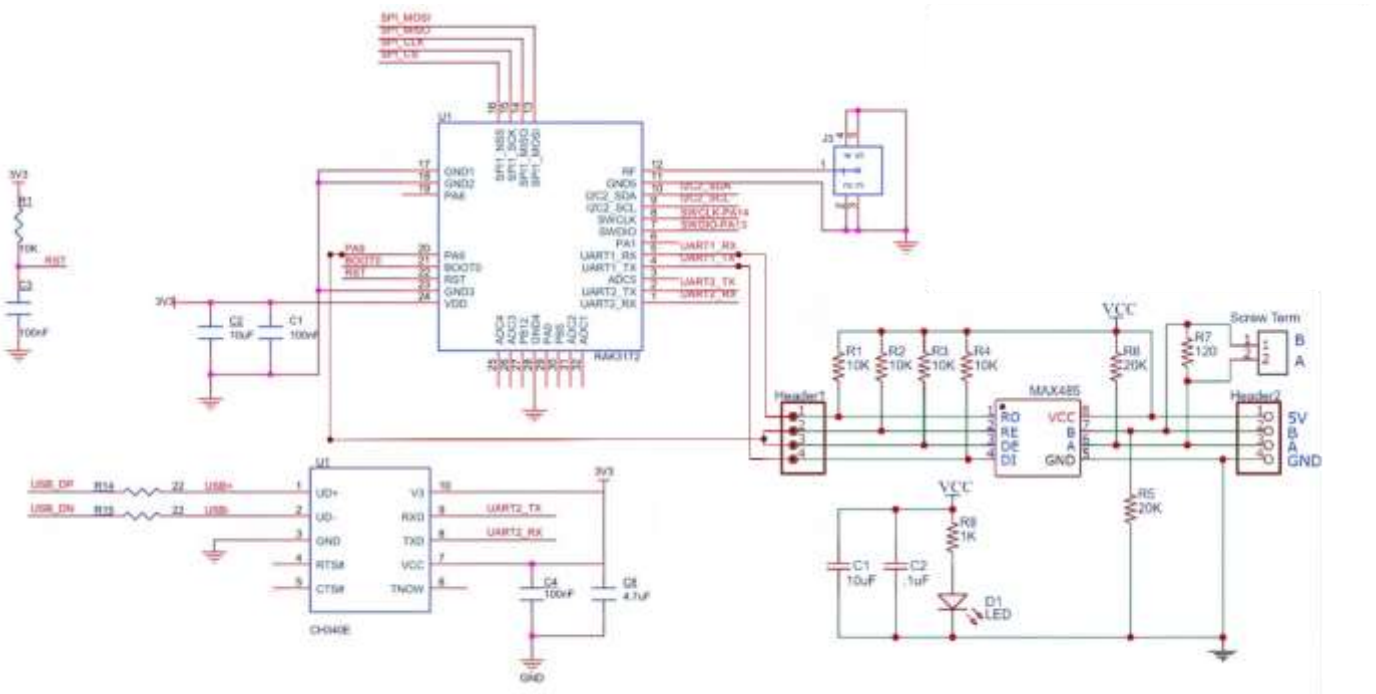
CHƯƠNG II: THIẾT KẾ BỘ CHUYỂN ĐỔI GIAO THỨC TRUYỀN THÔNG

2.1. Giới thiệu chương

Chương này trình bày quá trình thiết kế bộ chuyển đổi giao thức từ Modbus RTU sang LoRa, tập trung vào việc tích hợp phần cứng và phần mềm sử dụng module RAK3172. Mục tiêu của bộ chuyển đổi là cho phép các thiết bị cảm biến công nghiệp sử dụng Modbus RTU có thể truyền dữ liệu không dây về trung tâm điều khiển thông qua mạng LoRa. Việc thiết kế này bao gồm cả cơ sở lý thuyết về hai giao thức, mô hình hoạt động, sơ đồ khối hệ thống, lựa chọn linh kiện và chức năng phần mềm cơ bản để triển khai.

2.2 Sơ đồ của bộ chuyển đổi Modbus RTU – Lora

Sơ đồ nguyên lý của bộ chuyển đổi được thiết kế xoay quanh bộ vi điều khiển STM32WLE5CC một vi mạch tích hợp sẵn lõi xử lý ARM Cortex-M4 và khả năng truyền thông LoRa, giúp đơn giản hóa kiến trúc tổng thể, giảm số lượng linh kiện và tối ưu hiệu suất tiêu thụ năng lượng. STM32WLE5CC đóng vai trò trung tâm điều khiển, vừa đảm nhiệm truy vấn dữ liệu từ cảm biến qua giao thức Modbus RTU, vừa thực hiện đóng gói và truyền dữ liệu không dây qua LoRa.



Hình 2.1 Sơ đồ nguyên lý bộ chuyển đổi Modbus RTU – Lora.

❖ **Sơ đồ nguyên lý hoạt động theo trình tự sau:**

➢ Khởi động hệ thống:

- Nguồn 3.3V cấp cho toàn mạch.
- Mạch reset và boot dùng để:
  - Vào chế độ bootloader khi cần nạp firmware.
  - Reset MCU đúng thời điểm.

➢ Giao tiếp với cảm biến Modbus RTU:

- RAK3172 sử dụng UART1 để gửi lệnh đọc dữ liệu theo giao thức Modbus RTU.
- Tín hiệu giao tiếp được chuyển đổi qua IC MAX485:
  - DE/RE dùng để điều khiển chiều truyền hoặc nhận dữ liệu (trạng thái HIGH để truyền, LOW để nhận).
  - Tín hiệu từ chân TX của RAK3172 được đưa vào chân DI của MAX485, sau đó được chuyển thành tín hiệu vi sai RS485 truyền đến cảm biến thông qua cặp dây A và B.
  - Phản hồi từ cảm biến sẽ được truyền ngược qua cặp dây A/B, sau đó đi vào chân RO của MAX485, và cuối cùng được đưa về chân RX của RAK3172 để xử lý.

➢ Xử lý dữ liệu và truyền LoRa:

- Sau khi nhận và xử lý dữ liệu cảm biến (thường là mực nước), RAK3172 sẽ tạo một gói dữ liệu LoRa.
- Module sử dụng phần SX1262 tích hợp trong RAK3172 để phát sóng không dây về Gateway trung tâm.

➢ Nạp firmware và debug:

- Khi cắm cổng USB:
  - Máy tính giao tiếp với CH340E.
  - Giao tiếp với UART2 của RAK3172 để:
    - Nạp chương trình qua bootloader.
    - Hiển thị log (nếu được lập trình xuất ra UART2).

Hình 2.2 Bộ chuyển đổi truyền thông giao thức Modbus RTU sang Lora.

2.3 Các thành phần phần cứng của bộ chuyển đổi

2.3.1 Vi điều khiển STM32WLE5CC

Là thành phần cốt lõi của hệ thống, tích hợp sẵn lõi ARM Cortex-M4 cùng bộ thu phát sóng LoRa (SX126x). Vi điều khiển này vừa thực hiện truy vấn Modbus RTU qua UART, vừa xử lý dữ liệu và phát sóng LoRa. Ưu điểm của việc tích hợp LoRa là giúp giảm linh kiện, tiêu thụ điện năng thấp, đồng thời đảm bảo hiệu suất xử lý và truyền thông ổn định.

Việc lựa chọn STM32WLE5CC làm nền tảng phần cứng trung tâm của hệ thống được cân nhắc kỹ lưỡng dựa trên nhiều ưu điểm vượt trội:

- Tích hợp sẵn bộ thu phát LoRa (LoRa transceiver): STM32WLE5CC là dòng vi điều khiển hiếm hoi trên thị trường tích hợp hoàn chỉnh cả MCU và bộ thu phát LoRa SX1262 trong một chip duy nhất. Điều này giúp giảm thiểu số lượng linh kiện, đơn giản hóa thiết kế phần cứng, tiết kiệm không gian PCB và giảm rủi ro về tương thích phần cứng.
- Hiệu suất xử lý mạnh mẽ: Chip sử dụng lõi ARM Cortex-M4 tốc độ 48 MHz, có tích hợp FPU (Floating Point Unit), đủ năng lực để xử lý giao thức Modbus RTU, đóng gói dữ liệu, truyền thông LoRa, và cả xử lý biên (edge processing) nếu cần mà vẫn đảm bảo tiêu thụ điện thấp.
- Hỗ trợ giao tiếp phong phú: STM32WLE5CC hỗ trợ nhiều giao tiếp như UART, SPI, I2C, GPIO… rất phù hợp để giao tiếp với cảm biến công nghiệp Modbus RTU và các thành phần ngoại vi khác như MAX485, mạch điều khiển DE/RE, hoặc cảm biến mở rộng.
- Tiết kiệm năng lượng: Dòng chip này hỗ trợ nhiều chế độ tiết kiệm năng lượng như Standby, Sleep… giúp hệ thống vận hành bền bỉ trong điều kiện cấp nguồn bằng pin/năng lượng mặt trời – một yêu cầu quan trọng với các trạm quan trắc ngoài trời.
- Khả năng cập nhật và phát triển phần mềm dễ dàng: STM32 là dòng chip phổ biến, có hệ sinh thái phần mềm mạnh như STM32CubeMX, STM32CubeIDE, thư viện HAL/LL... cùng cộng đồng phát triển rộng lớn, hỗ trợ việc lập trình, kiểm thử và cập nhật chương trình linh hoạt.
- Tuân thủ chuẩn công nghiệp và dải tần linh hoạt: STM32WLE5CC hỗ trợ dải tần LoRa từ 150 MHz đến 960 MHz, phù hợp nhiều khu vực trên thế giới, có thể điều chỉnh công suất và thông số truyền để đáp ứng quy định vùng tần số tại Việt Nam.

	8051	PIC	AVR	ARM
Độ rộng Bus	8-bits	8/16/32-bits	8/32-bits	32/64-bits
Giao tiếp	UART, USART, SPI, I2C.	UART, USART, SPI, I2C..	UART, USART, SPI, I2C...	UART, USART, SPI, I2C ...
Tốc độ	12 clock/cycle	4 clock/cycle	1 clock/cycle	1 clock/cycle
Bộ nhớ	ROM, SRAM, FLASH	SRAM, FLASH	SRAM, FLASH, EEPROM	FLASH, SDRAM, EEROM
Giá thành	Thấp	Cao	Cao	Thấp

Bảng 2.1 So sánh các dòng vi điều khiển trên thị trường

So sánh các dòng vi điều khiển trên thị trường hiện nay ta thấy những vi điều khiển sử dụng bộ sử lý lõi ARM có nhiều tính năng vượt trội hơn những dòng khác như độ rộng độ rộng bus dữ liệu, giao thức truyền thông và đặt biệt là giá thành rẻ hơn nhiều so với các loại khác

Từ những lý do trên, STM32WLE5CC là lựa chọn tối ưu để xây dựng một hệ thống nhúng truyền thông Modbus RTU sang LoRa với yêu cầu cao về độ tin cậy, khả năng mở rộng và tiết kiệm năng lượng.

2.3.2 IC chuyển đổi RS485-TTL(MAX485)

Chuẩn giao tiếp Modbus RTU hoạt động trên tầng vật lý RS485, sử dụng truyền thông vi sai để tăng khả năng chống nhiễu và truyền xa (lên đến 1.2 km). Tuy nhiên, vi điều khiển STM32WLE5CC không hỗ trợ trực tiếp chuẩn RS485, mà chỉ cung cấp giao tiếp UART TTL (3.3V). Vì vậy, hệ thống cần một IC chuyển đổi trung gian – MAX485 – để thực hiện chức năng sau:

- Chuyển đổi tín hiệu điện áp: MAX485 chuyển tín hiệu UART TTL từ vi điều khiển thành tín hiệu vi sai RS485 (và ngược lại), phù hợp để giao tiếp với cảm biến công nghiệp sử dụng chuẩn RS485.
- Tăng độ ổn định và truyền xa: Chuẩn RS485 giúp truyền dữ liệu ổn định trên khoảng cách xa và môi trường nhiễu cao như ngoài trời, vùng công nghiệp hoặc ven sông suối, điều mà TTL không làm được.
- Kết nối đa điểm: Chuẩn RS485 cho phép kết nối nhiều cảm biến trên cùng một đường truyền bus, giúp mở rộng hệ thống sau này mà không cần thêm phần cứng trung gian.
- Chi phí thấp, dễ tích hợp: MAX485 là IC phổ biến, giá rẻ, dễ mua và dễ tích hợp với bất kỳ vi điều khiển nào, phù hợp cho hệ thống IoT công nghiệp quy mô vừa và nhỏ.

Ngoài ra, để quản lý hướng truyền (Tx hoặc Rx), IC MAX485 cần tín hiệu điều khiển DE/RE – được vi điều khiển điều khiển qua một chân GPIO. Việc điều khiển này đảm bảo rằng thiết bị chỉ truyền hoặc nhận tại một thời điểm, tránh xung đột tín hiệu trên đường truyền RS485.

2.3.3. IC CH340E – Giao tiếp USB to UART

Trong thiết kế bộ chuyển đổi, vi mạch CH340E được tích hợp để đảm nhiệm vai trò chuyển đổi tín hiệu từ USB sang UART, hỗ trợ quá trình nạp chương trình và cấu hình giao tiếp cho vi điều khiển STM32WLE5CC. CH340E cho phép kết nối trực tiếp giữa máy tính và bộ chuyển đổi thông qua cổng USB, từ đó có thể thực hiện:

- Ghi nạp chương trình vào vi điều khiển (thông qua UART2).
- Hiển thị dữ liệu test/debug từ hệ thống qua Serial Monitor.
- Cấu hình các thông số truyền thông nếu cần (baudrate, ID cảm biến, v.v).

Mạch CH340E sử dụng nguồn 3.3V, kết nối với chân TX/RX của UART2 và hiển thị dữ liệu theo chuẩn TTL logic. Đây là lựa chọn phổ biến do giá thành thấp, ổn định cao và tương thích tốt với các hệ điều hành Windows, Linux, macOS thông qua driver CH340.

2.3.4 Truyền thông không dây Lora

Trong thiết kế hệ thống IoT giám sát từ xa, giao tiếp không dây đóng vai trò then chốt để truyền dữ liệu về trung tâm giám sát mà không phụ thuộc vào hạ tầng mạng sẵn có. LoRa (Long Range) được lựa chọn làm chuẩn truyền thông chính do hội tụ nhiều ưu điểm vượt trội so với các giao thức khác như Wi-Fi, Zigbee hay 3G/4G:

- Phạm vi truyền xa: LoRa có thể truyền dữ liệu ở khoảng cách lên đến 5–15 km trong điều kiện ngoài trời, rất phù hợp cho các điểm đặt cảm biến phân tán rộng rãi ở vùng nông thôn, đồi núi hoặc ven sông.
- Tiêu thụ năng lượng thấp: Với cơ chế truyền theo chu kỳ và tốc độ thấp, LoRa cho phép thiết bị vận hành bằng pin hoặc năng lượng mặt trời trong thời gian dài mà không cần bảo trì thường xuyên.
- Chi phí vận hành thấp: Không phụ thuộc vào mạng di động hoặc kết nối Internet tại điểm cảm biến. Toàn bộ hệ thống có thể triển khai độc lập với chi phí thấp, đặc biệt phù hợp cho các ứng dụng công ích.
- Khả năng mở rộng cao: Một Gateway LoRa có thể thu nhận dữ liệu từ hàng trăm thiết bị đầu cuối, giúp giảm chi phí triển khai hạ tầng trung tâm.
- Khả năng xuyên nhiễu tốt: LoRa sử dụng điều chế trải phổ (spread spectrum), cho phép truyền tín hiệu ổn định ngay cả trong môi trường nhiễu cao, có vật cản.

Công nghệ	Tầm xa	Tiêu thụ năng lượng	Chi phí triển khai	Phụ thuộc hạ tầng	Nhược điểm
LORA	5-10km	Rất thấp (~2mW)	Thấp	Không	Tốc độ truyền thấp
WIFI	50-100m	Cao (~100mW)	Trung bình	Mạng wifi	Tiêu thụ năng lương cao
3G/4G	>10km	Cao (~500mW)	Cao	Mạng viễn thông	Chi phí duy trì cao
ZIGBEE	10-100m	Thấp (~10mW)	Trung bình	Mạng lưới riêng	Tầm xa hạn chế

Bảng 2.2 So sánh các chuẩn truyền thông không dây khác.

Với những đặc điểm trên, LoRa là lựa chọn tối ưu để đảm bảo tính ổn định, bền vững và linh hoạt cho hệ thống giám sát mực nước từ xa, đặc biệt trong bối cảnh địa hình phức tạp và điều kiện triển khai hạn chế như tại Việt Nam.

2.4. Kiến trúc truyền thông dữ liệu

Kiến trúc truyền thông dữ liệu trong hệ thống IoT này được thiết kế nhằm đảm bảo tính ổn định, độ tin cậy cao và tiết kiệm năng lượng. Dữ liệu từ các cảm biến Modbus RTU sẽ được thu thập bởi vi điều khiển STM32 thông qua giao tiếp RS485. Vi điều khiển thực hiện truy vấn dữ liệu định kỳ, giải mã khung dữ liệu trả về, sau đó đóng gói lại theo định dạng tối ưu để truyền bằng giao thức LoRa đến thiết bị Gateway.

Quy trình truyền thông dữ liệu diễn ra như sau:

- Khởi tạo giao tiếp: Vi điều khiển thiết lập thông số kết nối Modbus RTU (địa chỉ cảm biến, baudrate, parity, v.v.).
- Truy vấn cảm biến: Gửi khung lệnh đọc (Function Code 03/04) đến cảm biến thông qua chuẩn RS485.
- Nhận dữ liệu: Cảm biến phản hồi khung dữ liệu Modbus; vi điều khiển kiểm tra CRC, phân tích payload.
- Xử lý và đóng gói: Dữ liệu được lọc và đóng gói lại thành bản tin LoRa gọn nhẹ.
- Truyền LoRa: Vi điều khiển gửi dữ liệu qua mô-đun LoRa (RAK3172) đến Gateway.
- Tiếp nhận và xử lý: Gateway nhận dữ liệu, có thể hiển thị lên Dashboard hoặc gửi về Server thông qua mạng IP (Wi-Fi/4G).

Cấu trúc này giúp hệ thống đạt được mục tiêu tiêu thụ năng lượng thấp, tránh truyền dữ liệu dư thừa, đồng thời bảo đảm tính tương thích với hạ tầng hiện tại.

- Tương thích cao với hệ thống giám sát hiện có: Gateway LoRa có thể kết nối với hệ thống máy chủ, nền tảng đám mây hoặc phần mềm SCADA thông qua mạng

IP (Wi-Fi, 4G, Ethernet), tạo điều kiện thuận lợi cho việc tích hợp và đồng bộ dữ liệu.

- Chi phí đầu tư thấp: Kiến trúc tận dụng các module phổ biến như STM32WLE5CC, cảm biến Modbus và bộ chuyển đổi RS485-TTL giá thành rẻ nhưng hiệu quả cao. Việc giảm thiểu dây dẫn và sử dụng truyền thông không dây giúp tiết kiệm chi phí thi công, bảo trì.
- Khả năng chống chịu môi trường: Tất cả phần cứng được thiết kế để hoạt động trong điều kiện môi trường ngoài trời, có khả năng chịu nhiệt, chống ẩm và chống nhiễu điện từ từ môi trường công nghiệp.

2.5. Giao thức Modbus RTU

Trong quá trình triển khai hệ thống, chúng tôi lựa chọn cảm biến đo mực nước RS-DR-N01-1 do Renke sản xuất vì tính phổ biến và khả năng tương thích tốt với giao thức Modbus RTU qua RS485. Thay vì làm việc với dữ liệu mẫu, nhóm đã thử nghiệm thực tế cảm biến để xác định chính xác khung dữ liệu phản hồi, độ ổn định kết nối và mức tiêu thụ điện khi hoạt động ngoài trời.

Thông số mặc định của cảm biến gồm địa chỉ Modbus 0x01, tốc độ baud 4800 bps, định dạng truyền 8N1. Trong khi nhiều cảm biến cho phép tùy chỉnh qua giao diện RS485, cảm biến của Renke lại yêu cầu thao tác cấu hình trước bằng phần mềm chuyên dụng – điều này buộc nhóm phải thiết kế bộ chuyển đổi sao cho linh hoạt về thông số cấu hình.

Quá trình thiết kế giao tiếp Modbus RTU thực tế gồm các bước:

- Giao tiếp UART1 của STM32WLE5CC được khởi tạo ở tốc độ 4800 bps, sử dụng chân PB0 để điều khiển DE/RE của MAX485;
- Nhóm viết hàm tạo khung truy vấn đúng cấu trúc: địa chỉ thiết bị, mã hàm 0x03, địa chỉ thanh ghi 0x0000 và số thanh ghi cần đọc;
- Khung phản hồi từ cảm biến thường có định dạng 01 03 02 xx xx CRC, trong đó dữ liệu đo mực nước nằm ở hai byte giữa;
- Dữ liệu sau khi được kiểm tra CRC sẽ được chuyển đổi sang đơn vị cm và lưu vào biến tạm để truyền đi qua LoRa.

❖ **Khung truyền truy vấn:**

Khung truy vấn Modbus gửi từ STM32 đến cảm biến có dạng:

01 03 00 00 00 01 CRC_L CRC_H

Giải thích từng byte:

- 01: Địa chỉ thiết bị (Device Address – cảm biến có địa chỉ Modbus 0x01)

- 03: Mã hàm đọc thanh ghi giữ liệu (Function code 0x03)
- 00 00: Địa chỉ thanh ghi bắt đầu (0x0000 tương ứng với thanh ghi 40001)
- 00 01: Số lượng thanh ghi cần đọc (1 thanh ghi)
- CRC_L CRC_H: Hai byte kiểm tra CRC16 (tính trên các byte trước đó, thứ tự little-endian)

❖ **Khung truyền phản hồi:**

Phản hồi từ cảm biến có dạng:

01 03 02 00 2B CRC_L CRC_H

Giải thích từng byte:

- 01: Địa chỉ thiết bị phản hồi (xác nhận đúng thiết bị)
- 03: Mã hàm xác nhận yêu cầu đọc
- 02: Số byte dữ liệu trả về (2 byte)
- 00 2B: Dữ liệu thực tế (giá trị mực nước = 0x002B = 43 cm)
- CRC_L CRC_H: CRC16 cho toàn bộ phần phản hồi

Để đảm bảo tính toàn vẹn dữ liệu, hệ thống sử dụng thuật toán CRC-16 (Modbus CRC) để kiểm tra gói tin gửi và nhận. Giá trị CRC được tính trên toàn bộ phần nội dung khung (trừ CRC) theo trình tự:

- Khởi tạo CRC với giá trị 0xFFFF;
- Duyệt từng byte, thực hiện XOR và dịch phải từng bit;
- Nếu bit LSB là 1, XOR với đa thức 0xA001 (biểu diễn đảo của 0x8005);
- Kết quả cuối cùng là 2 byte CRC (Low byte trước, High byte sau).

Nếu dữ liệu phản hồi có CRC không trùng khớp, hệ thống sẽ bỏ qua kết quả và thực hiện lại truy vấn (cơ chế retry tối đa 3 lần). Thực tế triển khai cho thấy lỗi CRC thường xảy ra khi dây RS485 bị nhiễu hoặc cảm biến phản hồi chậm hơn dự kiến. Để giảm thiểu lỗi, nhóm đã sử dụng:

- Cáp xoắn đôi chống nhiễu;
- Tụ lọc nguồn gần cảm biến;

Cơ chế timeout động: nếu phát hiện lỗi liên tiếp, hệ thống tự động giãn thời gian chờ phản hồi thêm 50 ms mỗi lần retry.

Những biện pháp này giúp hệ thống vận hành ổn định hơn trong môi trường ngoài trời có nhiễu điện từ hoặc thời tiết thay đổi.

Theo cấu hình 4800 bps, định dạng 8N1 (10 bit/ký tự), thời gian truyền được tính như sau:

- Truy vấn gửi: 8 byte × 10 = 80 bit

- Phản hồi nhận: 7 byte × 10 = 70 bit → Tổng: 150 bit → Thời gian truyền lý thuyết = 150 / 4800 ≈ 31.25 ms

Tuy nhiên, trong thực tế hệ thống phải đợi cảm biến phản hồi chậm, nên nhóm cấu hình thời gian timeout là 200 ms để đảm bảo ổn định truyền nhận và tránh lỗi CRC.

2.6. Triển khai giao thức LoRa và tối ưu truyền dữ liệu

Việc thiết kế truyền thông LoRa không đơn giản như chọn một thư viện có sẵn. Vì sử dụng vi điều khiển STM32WLE5CC tích hợp LoRa, nhóm đã phải cấu hình trực tiếp bộ SX126x bên trong bằng thư viện STM32WL HAL. Tần số truyền được cấu hình ở 923 MHz, tương thích băng tần Việt Nam, công suất phát 14 dBm, tốc độ dữ liệu thấp nhằm đảm bảo khoảng cách truyền xa và tiết kiệm năng lượng.

Một điểm quan trọng trong thiết kế là đảm bảo dữ liệu truyền đi không bị mất mát hoặc sai lệch khi qua sóng vô tuyến. Vì vậy, nhóm đã xây dựng cơ chế đóng gói dữ liệu trước khi truyền.

Việc đóng gói này giúp dễ dàng phát hiện lỗi trong quá trình truyền và hỗ trợ giải mã chính xác tại phía thu. Ngoài ra, nhóm cũng có thể bổ sung thêm mã checksum nhẹ trong các gói tin để tăng độ tin cậy. Dữ liệu sau khi kiểm tra và xử lý sẽ được đóng gói thành một chuỗi liên tục, không chứa bất kỳ ký tự phân tách nào, nhằm tối ưu dung lượng truyền và đơn giản hóa việc giải mã phía gateway. Cấu trúc gói tin như sau:

- ID thiết bị: 1 byte
- Dữ liệu mực nước: 2 byte
- Timestapm: 4 byte
- CRC-16: 2 byte

CRC được tính theo phương pháp XOR cộng dồn hoặc thuật toán CRC-16 tiêu chuẩn, đảm bảo phát hiện lỗi bit đơn khi truyền không dây. Gateway sẽ giải mã gói tin theo đúng thứ tự byte, kiểm tra CRC trước khi ghi nhận dữ liệu.

Gói tin nhị phân 12 byte này sau đó được truyền qua mô-đun LoRa nội bộ (SX1262) đến Gateway LoRaWAN gần nhất.

Sau khi truyền, gateway sẽ ghi nhận dữ liệu và nhóm đã kiểm tra khả năng truyền ổn định ở khoảng cách trên 500 mét trong môi trường có vật cản. Đồng thời, nhóm cũng thử nghiệm mô hình gửi dữ liệu theo chu kỳ cố định 60 giây và khi có thay đổi lớn về mực nước (trên 5 cm), cho thấy hệ thống đáp ứng tốt cho cả hai chế độ hoạt động.

➢ **Tính toán thời gian truyền LoRa**

❖ **Gói tin thực tế có cấu trúc với tổng cộng 12 byte:**

- ID: 1 byte
- Mực nước: 2 byte
- Timestamp: 4 byte

- CRC: 2 byte → Tổng payload = Tổng gói (sau thêm header LoRa + overhead) ≈ 12 byte

❖ **Cấu hình truyền:**

- Spreading Factor (SF): 10
- Bandwidth (BW): 125 kHz
- Coding Rate (CR): 4/5
- Header: Explicit
- CRC: On
- Preamble: 8 symbols

Theo công cụ tính Time-on-Air của Semtech (SX1262), với payload 12 byte và các thông số trên: → Thời gian truyền gói ≈ 270 – 300 ms

Gói tin truyền một lần mỗi 60 giây → chiếm chưa đến 0.5% thời gian hoạt động, phù hợp với ứng dụng tiết kiệm năng lượng sử dụng pin hoặc năng lượng mặt trời.

Mỗi phút truyền 1 lần → chiếm <0.5% thời gian hoạt động, phù hợp cho ứng dụng tiết kiệm năng lượng.

2.7. Lưu đồ thuật toán

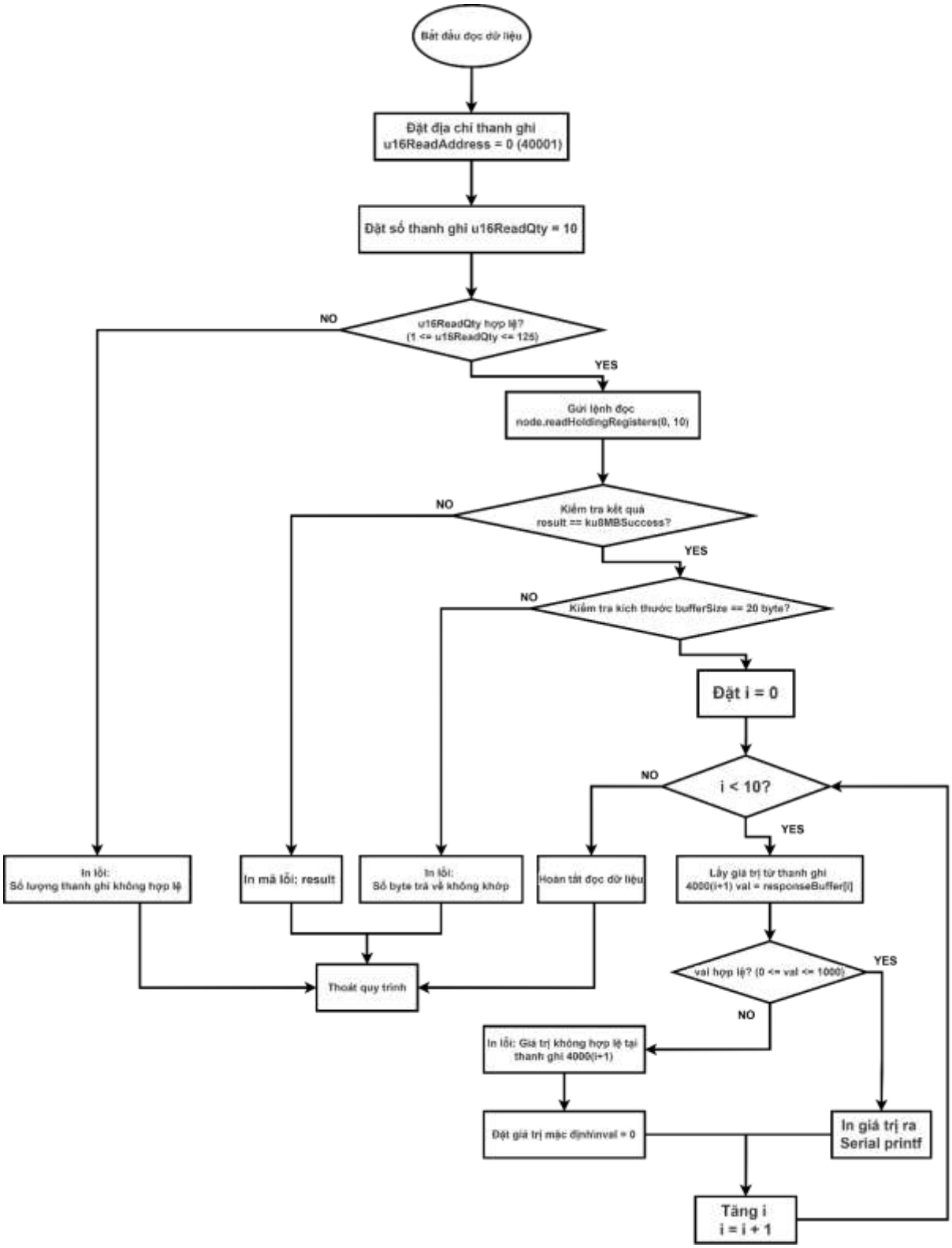
❖ **Mục tiêu của lưu đồ thuật toán**

- Xác định quy trình chi tiết từ việc thu thập dữ liệu thô (tầng thu thập thiết bị trường) đến xử lý và chuẩn bị dữ liệu cho các tầng cao hơn (tầng truyền tải thông tin, tầng xử lý dữ liệu và tầng ứng dụng).
- Đảm bảo tính minh bạch, dễ hiểu và khả năng tái hiện trong việc lập trình hoặc triển khai thực tế.

❖ **Ý nghĩa và ứng dụng:**

- Minh họa quy trình: Giúp lập trình viên triển khai mã nguồn dựa trên thuật toán.
- Tối ưu hóa: Đảm bảo dữ liệu thu thập từ cảm biến mực nước được xử lý chính xác, giảm sai số và tăng độ tin cậy.
- Hỗ trợ bảo trì: Dễ dàng xác định lỗi mắc phải và cải tiến từng bước trong hệ thống IoT.

2.7.1. Đọc dữ liệu từ cảm biến



Hình 2.3 Lưu đồ thuật toán thu thập dữ liệu cảm biến qua giao thức Modbus RTU

Quy trình đọc dữ liệu từ các thanh ghi (Holding Registers) trong giao thức Modbus RTU, thường được sử dụng để thu thập thông tin từ các thiết bị cảm biến (như RS-DR-N01-1) trong hệ thống IoT.

- Khởi tạo hệ thống và các giao tiếp phần cứng
- Gửi lệnh Modbus RTU để yêu cầu cảm biến mực nước phản hồi dữ liệu
- Nhận và kiểm tra dữ liệu phản hồi (đặc biệt là kiểm tra CRC)
- Phân tích và trích xuất giá trị mực nước từ chuỗi byte nhận được
- Truyền dữ liệu qua giao thức LoRa đến Gateway
- Lặp lại quy trình theo chu kỳ định sẵn.
- Xử lý lỗi mắc phải trong quá trình.

❖ **Phân tích lưu đồ tổng quát của việc thu thập dữ liệu:**

➢ **Khởi tạo và thiết lập**

- Bắt đầu quá trình đọc dữ liệu
- Đặt địa chỉ thanh ghi cần đọc: u16ReadAddress = 0 (địa chỉ Modbus 40001)
- Đặt số lượng thanh ghi cần đọc: u16ReadQty = 10

➢ **Kiểm tra thông số đầu vào**

- Kiểm tra tính hợp lệ của số lượng thanh ghi (1 <= u16ReadQty <= 125) ➔Nếu không hợp lệ:
  - In ra lỗi: **“**Số lượng thanh ghi không hợp lệ”
  - Thoát quy trình

➢ **Gửi lệnh Modbus và kiểm tra phản hồi**

- Gửi lệnh đọc Modbus: node.readHoldingRegisters(0, 10)
- Kiểm tra kết quả trả về có thành công không (result == ku8MBSuccess) ➔ Nếu thất bại: In mã lỗi từ biến result ➔ Thoát quy trình
- Kiểm tra số byte trả về có đúng với kỳ vọng không (bufferSize == 20) ➔ Nếu sai: In lỗi: “Số byte trả về không khớp” ➔ Thoát quy trình

➢ **Xử lý dữ liệu đọc được**

- Đặt biến đếm i = 0
- Thực hiện vòng lặp với điều kiện i < 10
  - Lấy giá trị từ buffer: val = responseBuffer[i]
  - Kiểm tra giá trị có hợp lệ không (0 <= val <= 1000) → Nếu không hợp lệ:

        - In lỗi: “Giá trị không hợp lệ tại thanh ghi 4000(i+1)”
        - Gán giá trị mặc định: val = 0
    - In giá trị ra bằng Serial.printf
    - Tăng biến đếm i = i + 1

- **Kết thúc**
  - Hoàn tất quá trình đọc dữ liệu

2.7.2. Tạo và gửi gói tin Modbus RTU

Hình 2.4 Lưu đồ tạo và gửi gói tin Modbus RTU

Quy trình xây dựng và truyền gói tin theo giao thức Modbus RTU từ vi điều khiển STM32WLE5CC đến cảm biến.

- Chuẩn hóa quy trình đóng gói dữ liệu: Lưu đồ xác định các bước tuần tự để tạo khung dữ liệu Modbus RTU, bao gồm địa chỉ Slave ID, mã hàm (Function Code), địa chỉ thanh ghi (Address High/Low), số lượng thanh ghi hoặc byte (Quantity High/Low hoặc Byte Count), và giá trị CRC-16 (Cyclic Redundancy Check). Điều này đảm bảo tính nhất quán và chính xác trong việc xây dựng gói tin phù hợp với chuẩn Modbus RTU.
- Hỗ trợ quyết định logic: Thông qua nhánh điều kiện "Logi Write?" (Logic Write?), lưu đồ cho phép phân nhánh linh hoạt giữa lệnh đọc (Read) và ghi (Write), phản ánh các loại truy vấn khác nhau (Function Code 0x03 cho đọc, 0x06/0x10 cho ghi). Điều này giúp hệ thống tự động điều chỉnh nội dung gói tin dựa trên yêu cầu cụ thể.
- Đảm bảo tính toàn vẹn dữ liệu: Việc tính toán và thêm CRC-16 vào buffer truyền (bao gồm CRC Low và CRC High) nhằm phát hiện lỗi truyền dẫn, tăng độ tin cậy khi giao tiếp qua RS485 trong môi trường nhiễu điện từ.

❖ **Phân tích lưu đồ thuật toán tạo và gửi gói tin Modbus RTU:**

➢ **Bắt đầu**

- Mở đầu cho quá trình tạo và truyền gói tin Modbus RTU.

➢ **Khởi tạo buffer u8ModbusADU và u8ModbusADUSize = 0**

- Tạo bộ đệm dữ liệu Modbus (ADU = Application Data Unit).
- Đặt kích thước ban đầu của gói tin bằng 0.

➢ **Thêm Slave ID vào buffer**

- Địa chỉ thiết bị đích (thiết bị slave) được thêm vào buffer.

➢ **Thêm Function Code vào buffer**

- Function Code xác định loại lệnh Modbus, ví dụ:
  - 0x03: đọc nhiều thanh ghi
  - 0x06: ghi một thanh ghi
  - 0x10: ghi nhiều thanh ghi

➢ **Kiểm tra Function Code**

- Dựa vào Function Code để quyết định đi theo nhánh nào trong thuật toán:
  - Ghi 1 giá trị?
  - Ghi nhiều giá trị?

- Hay là lệnh đọc?

➢ **Thêm Write Address High/Low**

- Địa chỉ thanh ghi cần ghi (2 byte: High và Low) được thêm vào buffer.

➢ **Loại Write?**

- Phân biệt giữa:
  - Ghi 1 thanh ghi (FC06): đi nhánh bên trái
  - Ghi nhiều thanh ghi (FC16): đi nhánh bên phải

➢ **Nhánh ghi 1 thanh ghi (Function 06)**

- Thêm giá trị High/Low: Thêm dữ liệu cần ghi (2 byte)
- Tiếp tục đến bước tính CRC16

➢ **Nhánh ghi nhiều thanh ghi (Function 16)**

- Thêm Quantity High/Low: Số lượng thanh ghi sẽ ghi
- Thêm byte count: Số byte dữ liệu gửi
- Thêm dữ liệu từ TransmitBuffer: Lấy dữ liệu từ bộ đệm và thêm vào buffer

➢ **Nếu là lệnh đọc (Function 03, 04...)**

- Thêm Read Address High/Low: Địa chỉ bắt đầu cần đọc
- Thêm Read Quantity High/Low: Số lượng thanh ghi cần đọc

➢ **Tính CRC16**

- Tính mã kiểm tra lỗi (CRC16) trên toàn bộ gói dữ liệu (trừ 2 byte CRC cuối).
- CRC dùng để kiểm tra tính toàn vẹn dữ liệu khi truyền qua RS485.

➢ **Thêm CRC Low vào buffer**

➢ **Thêm CRC High vào buffer**

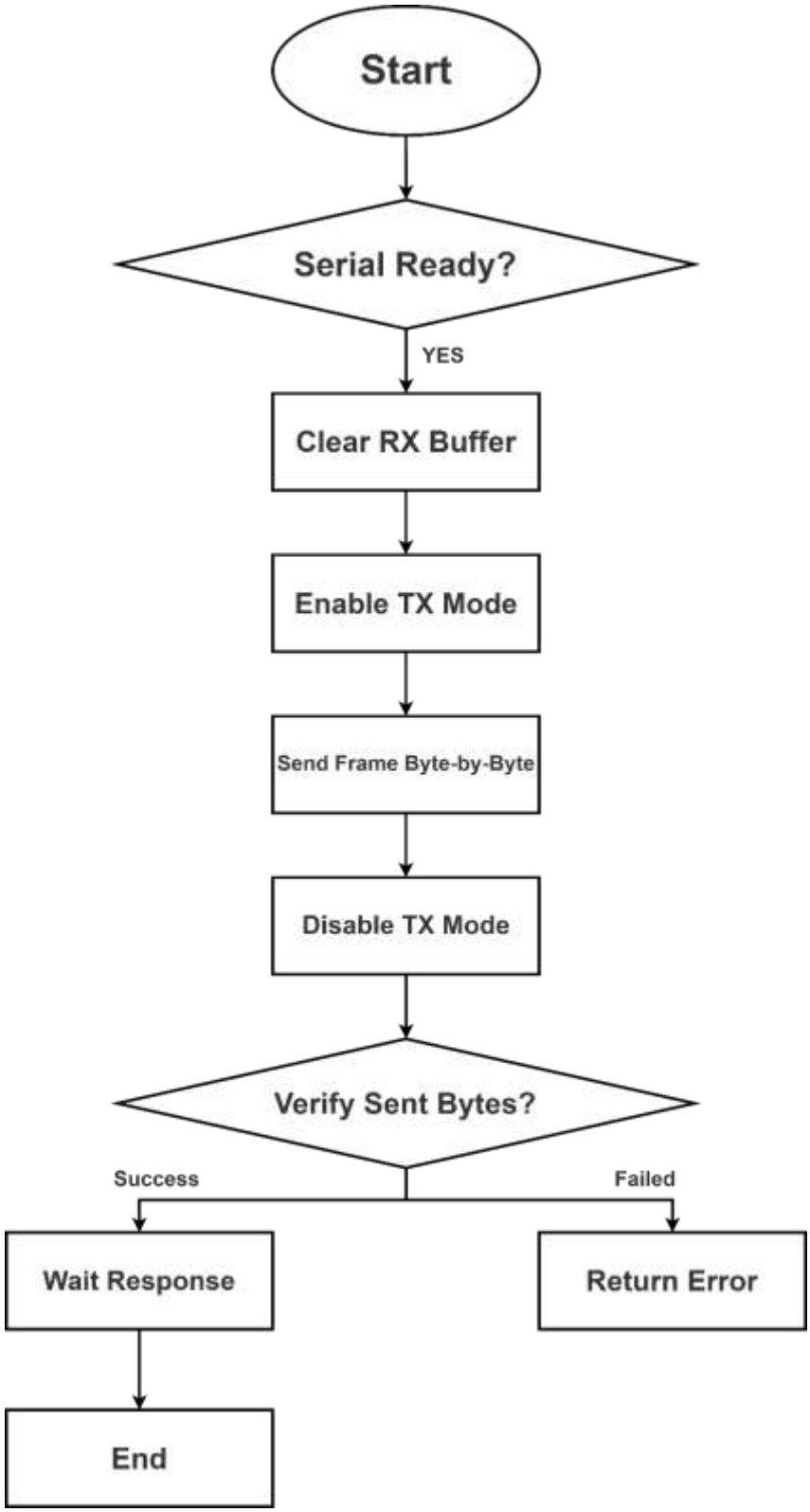
- CRC 2 byte được thêm vào buffer theo thứ tự: byte thấp trước, byte cao sau (theo chuẩn Modbus RTU).

➢ **Gửi buffer qua Serial**

- Buffer đã hoàn chỉnh được gửi qua giao tiếp Serial RS485.

➢ **Kết thúc:** Quá trình tạo và truyền gói tin hoàn tất.

2.7.3. Truyền dữ liệu Modbus RTU qua cổng Serial



Hình 2.5 Lưu đồ truyền dữ liệu Modbus RTU qua cổng Serial

Quy trình truyền dữ liệu qua giao tiếp nối tiếp (serial) từ vi điều khiển STM32WLE5CC, được thiết kế để đảm bảo tính chính xác và độ tin cậy trong quá trình gửi khung dữ liệu, đặc biệt trong bối cảnh giao tiếp với cảm biến Modbus RTU.

- Kiểm tra trạng thái giao tiếp: Bước "Serial Ready?" xác định sẵn sàng của cổng nối tiếp (ví dụ: UART), đảm bảo hệ thống chỉ truyền dữ liệu khi kênh liên lạc ổn định, tránh lỗi do xung đột hoặc nhiễu.
- Quản lý bộ đệm dữ liệu: "Clear RX Buffer" khởi tạo lại bộ đệm nhận (RX) để loại bỏ dữ liệu cũ, giảm nguy cơ nhầm lẫn dữ liệu từ các phiên trước, đảm bảo tính toàn vẹn của khung dữ liệu mới.
- Điều khiển chế độ truyền: Các bước "Enable TX Mode" và "Disable TX Mode" quản lý chuyển đổi trạng thái truyền (transmit) của giao tiếp nối tiếp, tối ưu hóa hiệu suất và giảm tiêu thụ năng lượng bằng cách chỉ kích hoạt chế độ truyền khi cần thiết.
- Truyền dữ liệu tuần tự: "Send Frame Byte-by-Byte" thực hiện gửi khung dữ liệu theo từng byte, phù hợp với giao thức Modbus RTU hoặc LoRa, đảm bảo tính chính xác trong truyền tải từng phần của gói tin.
- Xác minh và xử lý lỗi: Bước "Verify Sent Bytes?" kiểm tra xem toàn bộ dữ liệu đã được gửi thành công hay không, với hai nhánh:
  - Nếu thành công ("Success"), hệ thống chuyển sang "Wait Response" để chờ phản hồi từ thiết bị nhận, hỗ trợ giao tiếp hai chiều.
  - Nếu thất bại ("Failed"), trả về "Return Error" để ghi nhận lỗi, cho phép hệ thống thực hiện retry hoặc log lỗi để phân tích.

❖ **Phân tích lưu đồ truyền dữ liệu Modbus RTU qua cổng Serial:**

➢ **Khởi đầu quy trình**

- Start: Bắt đầu tiến trình gửi dữ liệu

➢ **Kiểm tra trạng thái cổng Serial**

- **Serial Ready?**
  ➔Nếu không sẵn sàng: dừng/thoát hoặc chờ tiếp
  ➔ Nếu sẵn sàng: tiếp tục các bước gửi frame

➢ **Chuẩn bị trước khi gửi**

- Clear RX Buffer: Xóa bộ đệm nhận (RX) để tránh dữ liệu cũ gây nhiễu khi nhận phản hồi.
- Enable TX Mode: Kích hoạt chế độ truyền (TX) – quan trọng trong truyền RS485 bán song công (half-duplex).

➢ **Truyền dữ liệu**

- Send Frame Byte-by-Byte: Gửi từng byte trong frame dữ liệu (thường là request Modbus) lần lượt qua Serial.

➢ **Kết thúc truyền**

- **Disable TX Mode**: Tắt chế độ truyền để chuẩn bị nhận dữ liệu phản hồi.

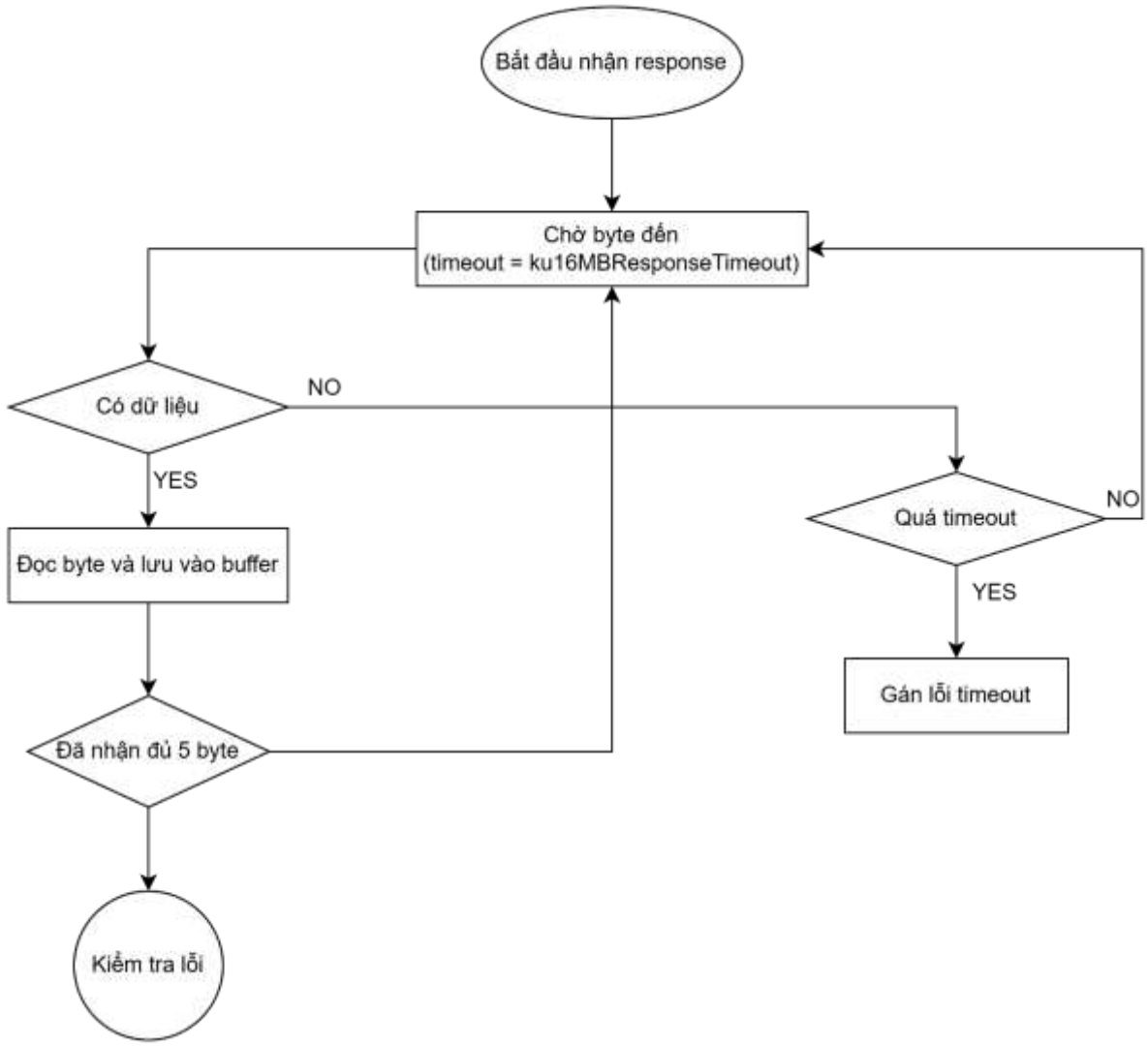
➢ **Xác minh quá trình gửi**

- **Verify Sent Bytes?**
  Kiểm tra xem số byte đã gửi có đúng và đủ không.
  ➔Nếu **thành công**:
  - **Wait Response**: Chờ phản hồi từ thiết bị (slave).
    ➔Nếu **thất bại**:
  - **Return Error**: Trả về mã lỗi hoặc thông báo lỗi.

➢ **Kết thúc quy trình**

- **End**: Kết thúc quá trình gửi request

2.7.4. Nhận và xử lý phản hồi Modbus RTU



Hình 2.6 Lưu đồ nhận và xử lý phản hồi Modbus RTU

Quy trình nhận và xử lý dữ liệu phản hồi từ cảm biến hoặc thiết bị ngoại vi thông qua giao thức Modbus RTU trên vi điều khiển STM32WLE5CC.

- Kiểm tra trạng thái nhận dữ liệu: Bước "Có dữ liệu không?" (Data available?) xác định sự tồn tại của phản hồi từ thiết bị, đảm bảo hệ thống chỉ xử lý khi có dữ liệu, tránh lãng phí tài nguyên hoặc xử lý sai.
- Quản lý thời gian chờ (Timeout):
  - "Có timeout không?" (Timeout?) kiểm tra xem quá trình nhận có vượt quá thời gian chờ định nghĩa (ví dụ: u16MBResponseTimeout) hay không. Nếu có, gán giá trị timeout và kết thúc, giúp hệ thống tự động phát hiện lỗi kết nối hoặc phản hồi chậm từ cảm biến.
  - Điều này đặc biệt quan trọng trong môi trường ngoài trời, nơi nhiễu hoặc độ trễ có thể xảy ra.

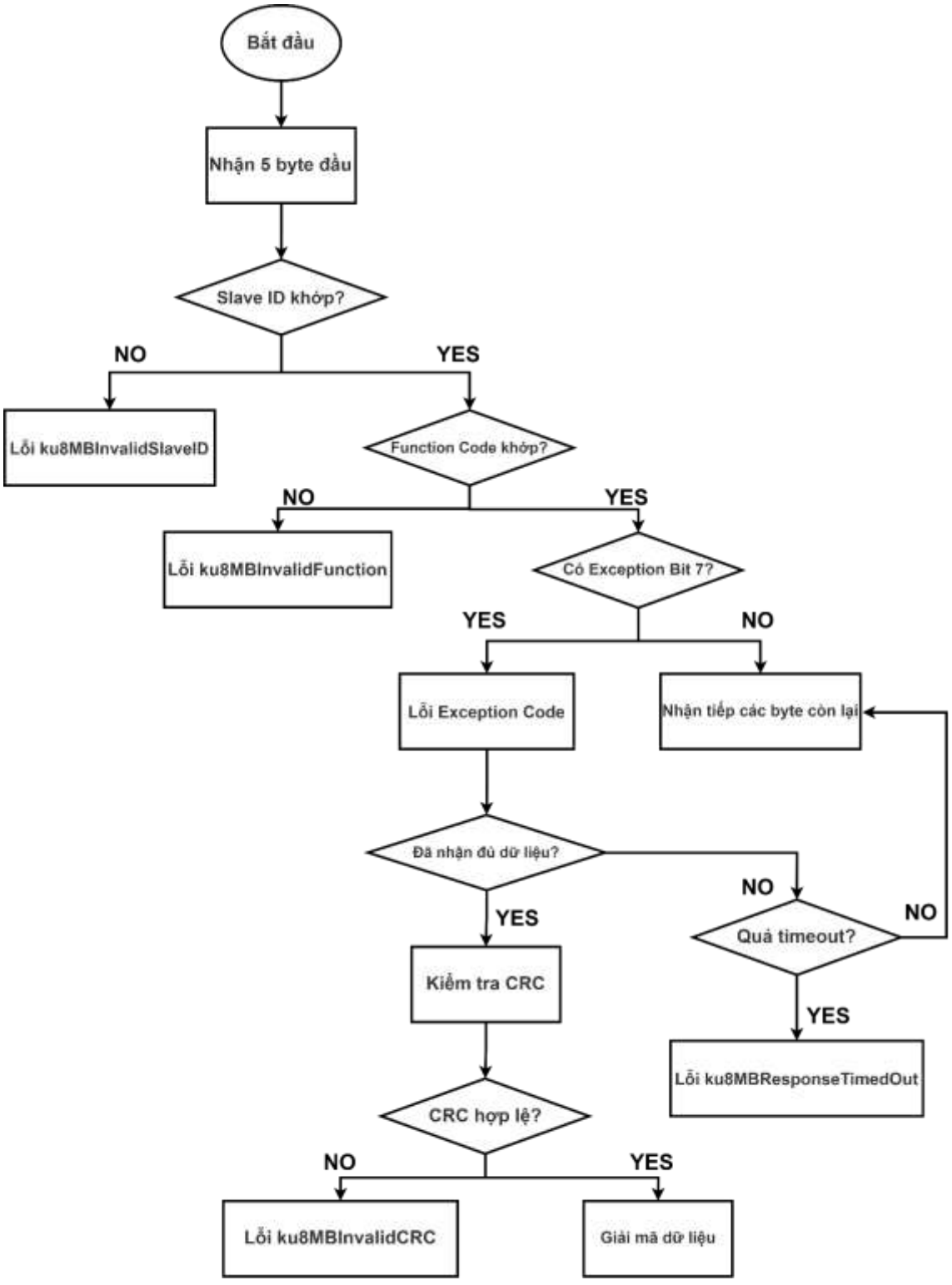
- Xử lý dữ liệu tuần tự:
  - "Đọc từng byte vào buffer" (Read byte into buffer) thực hiện nhận dữ liệu theo từng byte, phù hợp với định dạng Modbus RTU (8N1, 4800 bps trong trường hợp của bạn), đảm bảo toàn bộ khung dữ liệu được thu thập đầy đủ.
  - "Kiểm tra Slave ID, Function Code, Exception" xác minh tính hợp lệ của dữ liệu nhận được, bao gồm địa chỉ thiết bị (Slave ID), mã hàm (Function Code), và mã lỗi (Exception) nếu có.

❖ **Phân tích lưu đồ thuật toán nhận và xử lý phản hồi Modbus RTU:**

- **Bắt đầu nhận response**: Khởi động quá trình chờ dữ liệu phản hồi từ thiết bị slave.
- **Chờ byte đầu tiên đến (**timeout = ku16MBResponseTimeout) → Cơ chế timeout để tránh treo hệ thống nếu slave không phản hồi.
- **Có dữ liệu?** ➔ Nếu KHÔNG:
  - **Quá timeout?**
    ➔ Nếu có: Gán lỗi timeout và kết thúc
    ➔ Nếu chưa: tiếp tục chờ

    ➔ Nếu CÓ: sang bước tiếp theo
- **Đọc byte và lưu vào buffer**: Lưu dữ liệu nhận được vào bộ đệm (buffer).
- **Đã nhận đủ 5 byte?** (5 byte đầu tiên gồm: Slave ID + Function Code + Byte Count/Exception + CRC hoặc các thông tin khác)
  ➔ Nếu CHƯA đủ: quay lại tiếp tục nhận
  ➔ Nếu ĐỦ: sang bước kiểm tra

2.7.5. Kiểm tra lỗi



Hình 2.7 Lưu đồ kiểm tra lỗi.

Quy trình kiểm tra và xử lý dữ liệu phản hồi từ cảm biến hoặc thiết bị ngoại vi thông qua giao thức Modbus RTU trên vi điều khiển STM32WLE5CC, với mục tiêu xác minh tính hợp lệ của khung dữ liệu và phát hiện lỗi trong quá trình truyền nhận.

- Khởi tạo và nhận dữ liệu: Bước "Nhận 5 byte đầu" (Receive first 5 bytes) bắt đầu quá trình thu thập dữ liệu ban đầu từ cổng UART, bao gồm Slave ID, Function Code, và các byte đầu tiên của phản hồi, tạo nền tảng để phân tích tiếp theo.
- Xác minh Slave ID: "Slave ID hợp lệ không?" (Is Slave ID valid?) kiểm tra xem địa chỉ thiết bị (Slave ID, ví dụ: 0x01) có khớp với thiết bị mục tiêu hay không. Nếu không hợp lệ, gán trạng thái lỗi (u8MBInvalidSlaveID) và kết thúc, đảm bảo chỉ xử lý dữ liệu từ thiết bị đúng địa chỉ.
- Xác minh Function Code: "Function Code hợp lệ không?" (Is Function Code valid?) kiểm tra mã hàm (ví dụ: 0x03 cho đọc thanh ghi) có thuộc danh sách hỗ trợ hay không. Nếu không hợp lệ, gán trạng thái lỗi (u8MBInvalidFunction), cho phép hệ thống ghi nhận và xử lý lỗi cấu hình hoặc giao tiếp.
- Xử lý mã lỗi Exception: "Có Exception 0x77 không?" (Is there Exception 0x77?) kiểm tra sự hiện diện của mã lỗi đặc biệt (Exception), ví dụ: lỗi không hỗ trợ hoặc vượt quá giới hạn. Nếu có, gán lỗi (Lỗi Exception Code) và kết thúc, hỗ trợ phát hiện các tình huống bất thường từ cảm biến.
- Kiểm tra số byte nhận được: "Nhận đủ byte không?" (Received enough bytes?) đảm bảo số byte nhận được (ví dụ: 7 byte cho phản hồi 01 03 02 00 2B CRC_L CRC_H) đủ để xử lý, tránh phân tích dữ liệu không hoàn chỉnh. Nếu không đủ và có timeout (Có timeout không?), gán trạng thái timeout (u8MBResponseTimeout) và kết thúc.
- Kiểm tra và xác nhận CRC:
  - "Kiểm tra CRC" (Calculate CRC) tính toán CRC-16 dựa trên dữ liệu nhận được và so sánh với giá trị CRC trong khung.
  - "CRC hợp lệ không?" (Is CRC valid?) xác nhận tính toàn vẹn dữ liệu. Nếu không hợp lệ, gán trạng thái lỗi (u8MBInvalidCRC); nếu hợp lệ, "Giải mã dữ liệu" (Decode data) trích xuất thông tin (ví dụ: mực nước 43 cm từ 00 2B) để sử dụng tiếp theo.

❖ **Phân tích lưu đồ thuật toán kiểm tra lỗi:**

➢ **Bắt đầu:** Khởi động quá trình kiểm tra phản hồi nhận được từ slave.

➢ **Nhận 5 byte đầu:**

- Byte 1: Slave ID
- Byte 2: Function Code
- Byte 3+: các thông tin phản hồi cơ bản hoặc mã lỗi.

- **Slave ID khớp?**
  ➔ Nếu không khớp: Gán lỗi ku8MBInvalidSlaveID → kết thúc.
  ➔ Nếu đúng: tiếp tục kiểm tra Function Code.
- **Function Code khớp?**
  ➔ Nếu không đúng: Gán lỗi ku8MBInvalidFunction → kết thúc.
  ➔ Nếu đúng: tiếp tục kiểm tra Exception.
- **Có Exception Bit 7?**
  (Bit 7 = 1 nghĩa là có lỗi xảy ra)
  ➔ Nếu có Exception:
    - Trích xuất Exception Code
    - Gán lỗi Exception Code → kết thúc
      ➔ Nếu không có Exception: tiếp tục nhận phần còn lại của phản hồi.
- **Nhận tiếp các byte còn lại:** Dựa vào thông tin từ header hoặc Function Code để biết số byte cần nhận thêm.
- **Đã nhận đủ dữ liệu?**
  ➔Nếu chưa đủ:
    - **Timeout?**
      ➔ Nếu có: Gán lỗi ku8MBResponseTimedOut → kết thúc
      ➔ Nếu không: tiếp tục nhận
      ➔ Nếu đủ: sang bước kiểm CRC
- **Kiểm tra CRC:** Tính lại CRC từ toàn bộ gói nhận được, so sánh với CRC nhận được từ slave.
- **CRC hợp lệ?**
  ➔ Nếu không hợp lệ: Gán lỗi ku8MBInvalidCRC → kết thúc
  ➔ Nếu hợp lệ: tiến hành giải mã dữ liệu

2.7.6. Cách sửa lỗi

❖ Các lỗi có thể xảy ra:

Timeout không sử lý được: Hệ thống có thể treo slave không phản hồi trong thời gian dài.

- Cách khắc phục: Hệ thống kiểm tra liên tục thời gian đã trôi qua kể từ khi bắt đầu truyền, so sánh với ngưỡng thời gian chờ đã định nghĩa. Nếu vượt quá, trạng thái lỗi timeout được gán để báo hiệu và dừng quá trình.
- Kết quả: Ngăn chặn tình trạng treo, thông báo lỗi timeout kịp thời.

Dữ liệu không hợp lệ (Slave ID): Phản hồi từ thiết bị slave có địa chỉ không khớp với thiết bị mục tiêu có thể gây nhầm lẫn hoặc xử lý sai.

- Cách khắc phục: Sau khi nhận đủ 5 byte đầu tiên, hệ thống so sánh địa chỉ slave trong phản hồi với địa chỉ đã thiết lập. Nếu không khớp, trạng thái lỗi địa chỉ slave không hợp lệ được gán và quá trình dừng lại.
- Kết quả: Loại bỏ dữ liệu từ các thiết bị không mong muốn.

Function Code không hợp lệ: Yêu cầu hoặc phản hồi với mã hàm không được hỗ trợ có thể dẫn đến lỗi logic hoặc xử lý sai dữ liệu.

- Cách khắc phục: Hệ thống kiểm tra mã hàm trong phản hồi, loại bỏ bit lỗi (bit 7) để so sánh với mã hàm yêu cầu. Nếu không khớp, trạng thái lỗi mã hàm không hợp lệ được gán và dừng lại.
- Kết quả: Ngăn chặn xử lý các mã hàm không mong muốn hoặc không hợp lệ.

Exception từ slave: Thiết bị slave trả về mã lỗi (Exception) nhưng không được phát hiện hoặc xử lý.

- Cách khắc phục: Sau khi nhận 5 byte đầu, hệ thống kiểm tra bit lỗi (bit 7) trong mã hàm. Nếu có, mã lỗi từ slave (ví dụ: Exception Code) được ghi nhận làm trạng thái và kết thúc.
- Kết quả: Đảm bảo phát hiện và xử lý các lỗi từ slave một cách chính xác.

Số byte nhận không đủ: Phản hồi không chứa đủ byte cần thiết có thể dẫn đến lỗi giải mã hoặc xử lý sai.

- Cách khắc phục: Hệ thống kiểm tra số byte nhận được dựa trên loại mã hàm (ví dụ: 3 byte cho Write, số byte dữ liệu cho Read). Nếu không đủ và có timeout, trạng thái lỗi thời gian chờ được gán.
- Kết quả: Đảm bảo chỉ xử lý khi dữ liệu đầy đủ, tránh lỗi giải mã.

CRC không hợp lệ: Dữ liệu truyền bị lỗi do nhiễu nhưng không được phát hiện, dẫn đến sử dụng dữ liệu sai.

- Cách khắc phục: Hệ thống tính toán giá trị CRC-16 dựa trên tất cả byte nhận được (trừ 2 byte CRC cuối), sau đó so sánh với giá trị CRC trong phản hồi. Nếu không khớp, trạng thái lỗi CRC không hợp lệ được gán.
- Kết quả: Đảm bảo tính toàn vẹn của dữ liệu trước khi sử dụng.

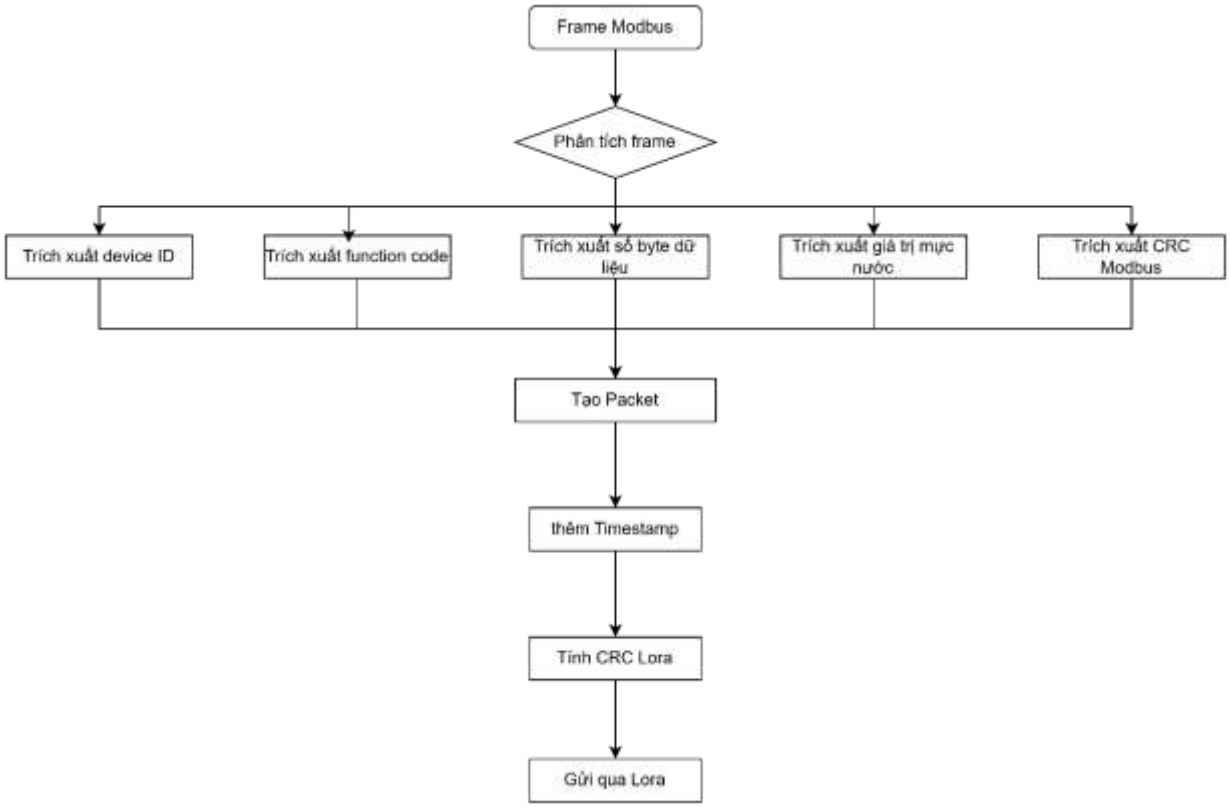
Xóa bộ đệm truyền sai thời điểm: Việc xóa bộ đệm truyền ngay sau khi gửi có thể làm mất dữ liệu phản hồi từ slave.

- Cách khắc phục: Hệ thống đã loại bỏ lệnh xóa bộ đệm truyền (flush) sau khi gửi yêu cầu, giữ lại dữ liệu trong bộ đệm để tiếp tục nhận phản hồi từ slave.
- Kết quả: Bảo vệ dữ liệu phản hồi, tránh mất mát thông tin quan trọng.

Dữ liệu không giải mã đúng: Byte cao (High) và byte thấp (Low) trong phản hồi không được sắp xếp đúng thứ tự, gây lỗi khi trích xuất.

- Cách khắc phục: Hệ thống sắp xếp byte theo thứ tự phù hợp với từng loại mã hàm (ví dụ: Low/High cho Read Coils, High/Low cho Read Registers) và lưu vào bộ đệm phản hồi, đảm bảo dữ liệu được trích xuất chính xác.
- Kết quả: Đảm bảo dữ liệu phản hồi được giải mã đúng thứ tự và giá trị.

2.7.7. Truyền thông LoRa



Hình 2.8 Lưu đồ truyền thông LoRa

- Nhận khung dữ liệu Modbus RTU : Bộ xử lý (RAK3172) hoạt động như một Modbus Master, định kỳ gửi lệnh đọc dữ liệu (Function Code 0x03 hoặc 0x04) đến cảm biến qua giao tiếp RS485. Cảm biến sẽ phản hồi bằng một frame dữ liệu Modbus, bao gồm:

  - Địa chỉ thiết bị (Device ID)
  - Mã chức năng (Function Code)
  - Số byte dữ liệu
  - Dữ liệu đo được
  - Mã kiểm tra lỗi CRC (CRC Modbus)

- Phân tích và trích xuất dữ liệu: Sau khi nhận frame, hệ thống thực hiện phân tích từng trường để trích xuất thông tin cần thiết:

  - Device ID: xác định địa chỉ cảm biến
  - Function Code: loại chức năng được sử dụng
  - Số byte dữ liệu: dùng để xác định độ dài dữ liệu

- Giá trị đo: ví dụ 0x0064 = 100 cm (mực nước)
- CRC: được kiểm tra để xác thực frame

Chỉ các trường quan trọng sẽ được giữ lại để gửi qua LoRa; CRC Modbus được dùng để kiểm lỗi frame nhận nhưng không cần truyền tiếp.

❖ Tạo gói dữ liệu LoRa: Thông tin sau khi trích xuất sẽ được đóng gói thành một packet LoRa, gồm các trường:

- ID thiết bị (Node ID)
- Giá trị đo (ví dụ: mực nước)
- Thời gian đo (timestamp)
- Mã CRC16 kiểm lỗi

⇨ Cấu trúc này giúp đảm bảo tính nhất quán và độ tin cậy khi truyền dữ liệu không dây.

❖ Tính CRC và gửi dữ liệu LoRa: Trước khi phát đi, hệ thống tính CRC cho toàn bộ packet LoRa để đảm bảo không có lỗi khi truyền. Sau đó, gói tin sẽ được gửi không dây qua LoRa đến trạm trung tâm (Node Master) sử dụng chế độ Point-to-Point hoặc Star Network.

Bước xử lý	Nội dung
1. Gửi lệnh đọc Modbus	Master gửi lệnh đến cảm biến
2. Nhận frame phản hồi	Frame chứa giá trị đo được
3. Phân tích frame	Tách các trường: ID, code, data, CRC
4. Tạo gói LoRa	Đóng gói thông tin cần thiết
5. Gắn timestamp & CRC	Đảm bảo đồng bộ và độ tin cậy
6. Gửi LoRa	Phát không dây đến Node trung tâm

Bảng 2.3 Tổng quan các bước truyền thông.

- ❖ Kết luận: Quá trình trên giúp đảm bảo bộ chuyển đổi có thể thu thập dữ liệu chính xác từ cảm biến RS485 và truyền không dây ổn định đến trung tâm, không cần kéo dây dài hay can thiệp vào thiết bị gốc. Giải pháp phù hợp cho các ứng dụng trong đo đạc từ xa như:
    - Quan trắc mực nước ao hồ, sông ngòi, các vùng có nguy cơ ngập lụt trong thành phố.
    - Đo lường môi trường
    - Hệ thống SCADA mini phân tán

2.8. Kết luận chương

Chương này đã giới thiệu quá trình thiết kế và triển khai hệ thống truyền thông tích hợp, kết hợp giao thức Modbus RTU và công nghệ LoRa để giám sát mực nước. Hệ thống sử dụng cảm biến RS-DR-N01-1 và vi điều khiển STM32WLE5CC, với cấu hình phù hợp cho giao tiếp có dây và không dây. Các lưu đồ thuật toán và biện pháp khắc phục lỗi đã đảm bảo tính ổn định và hiệu quả, phù hợp với các ứng dụng IoT. Thử nghiệm thực tế xác nhận khả năng hoạt động của hệ thống, mở ra hướng phát triển cho các phần tiếp theo.

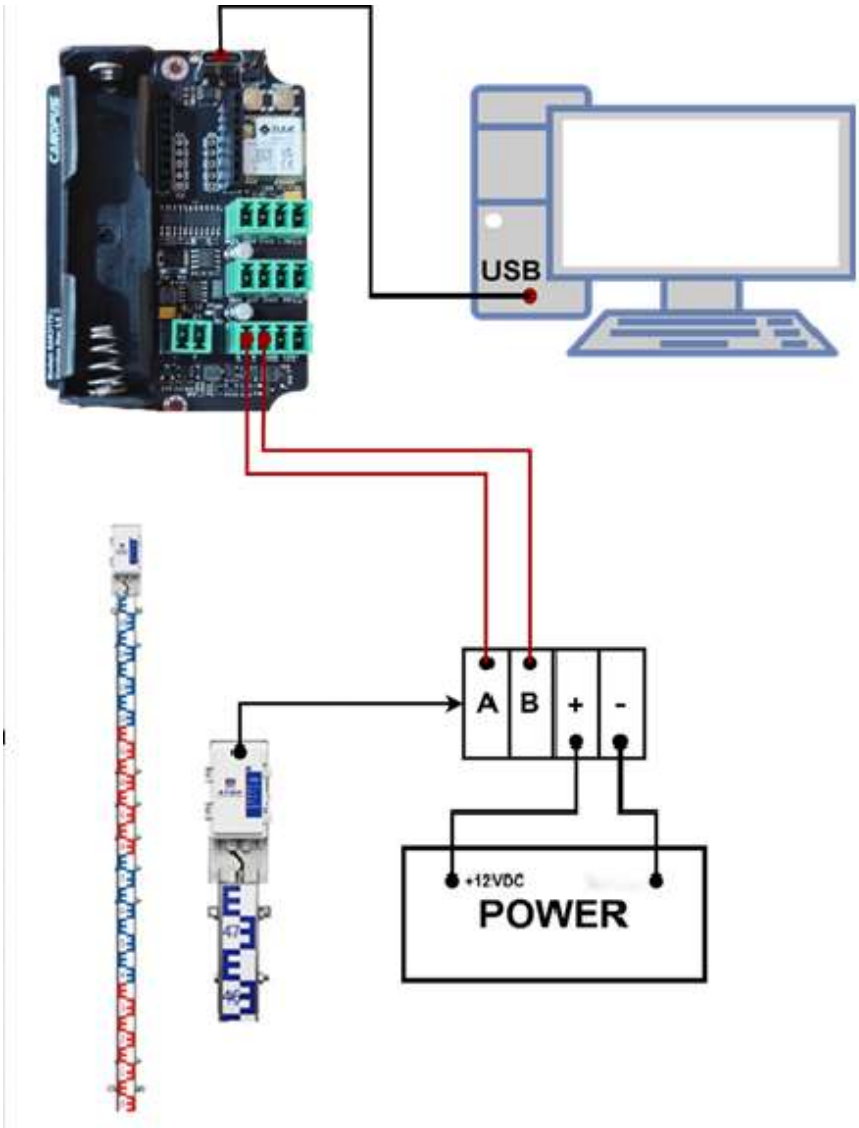
CHƯƠNG III: ĐÁNH GIÁ KẾT QUẢ VÀ NHẬN XÉT

3.1 Giới thiệu chương

Chương này trình bày quá trình kiểm thử hệ thống bộ chuyển đổi giao thức Modbus RTU sang LoRa đã được thiết kế, nhằm đánh giá mức độ hiệu quả, độ ổn định và tính ứng dụng thực tế của giải pháp. Các chỉ số đánh giá bao gồm: độ chính xác dữ liệu, độ trễ truyền nhận, độ ổn định khi hoạt động ngoài trời, tỷ lệ lỗi CRC, chất lượng tín hiệu LoRa và mức tiêu thụ năng lượng. Thông qua đó, nhóm rút ra được những điểm mạnh, hạn chế và tiềm năng cải tiến của hệ thống.

3.2 Kết quả thực nghiệm

3.2.1 Sơ đồ kết nối phần cứng đọc cảm biến qua bộ chuyển đổi

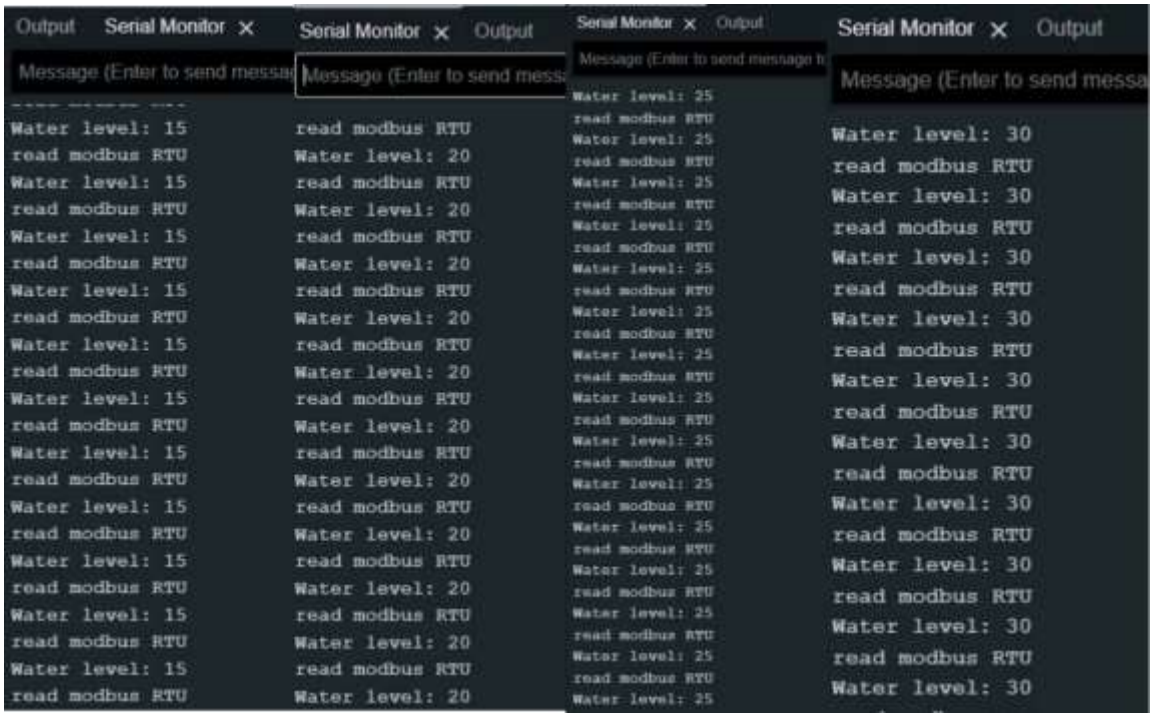


Hình 3.1 Sơ đồ kết nối phần cứng đọc cảm biến qua bộ chuyển đổi

❖ Lý do chọn cảm biến RS-DR-N01-1:

- Chuẩn giao tiếp Modbus RTU (RS485): phổ biến, chống nhiễu tốt, dễ tích hợp với MCU và LoRa.
- Độ chính xác cao: sai số nhỏ, độ phân giải tới từng cm.
- Hiển thị trực quan: có vạch thước dễ quan sát thực tế tại hiện trường.
- Hoạt động ổn định ngoài trời: chịu mưa, nắng, bụi, thích hợp lắp đặt ngoài đồng ruộng.
- Tương thích tốt với module RAK3172: đọc dữ liệu qua UART và truyền LoRa hiệu quả.
- Phù hợp ứng dụng IoT: đo từ xa, tiết kiệm năng lượng, truyền được qua khoảng cách lớn.

3.2.2 Kết quả đọc cảm biến qua bộ chuyển đổi



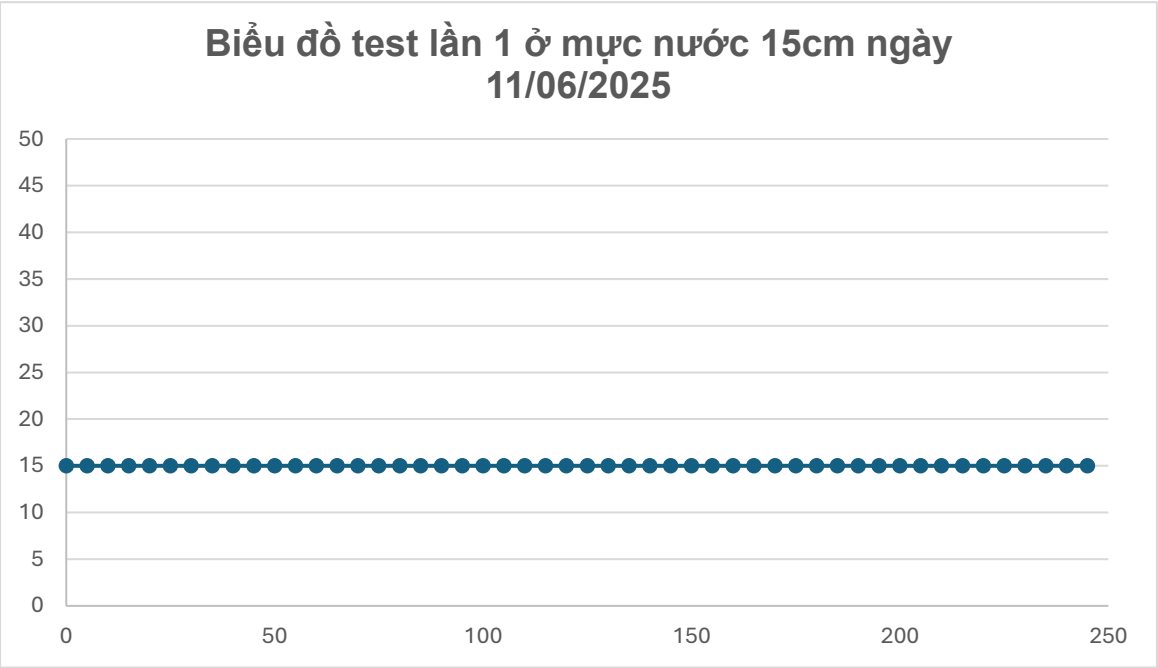
Hình 3.2 Kết quả dùng bộ chuyển chuyển đổi để đọc cảm biến

❖ **Thống kê kết quả thực nghiệm**

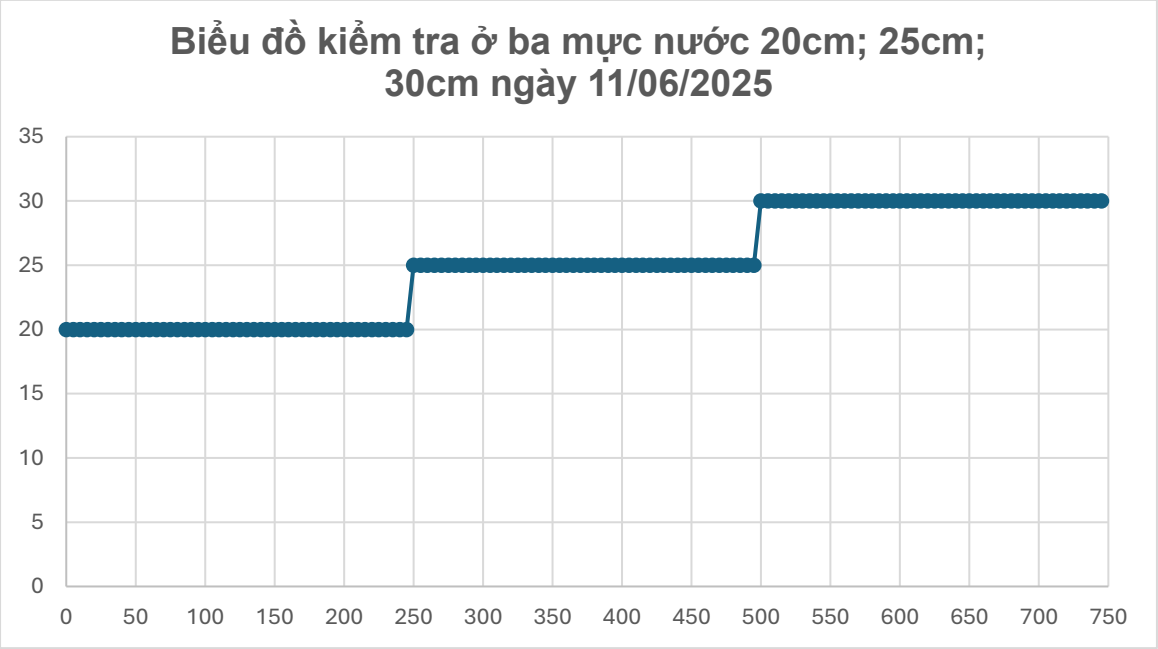
- Số lần gửi phản hồi: 200
- Số lần nhận phản hồi đúng: 200
- Số lần lỗi: 0
- Thời gian phản hồi trung bình: 36.8 ms

Dùng cảm biến đo trong 1 ngày, kiểm tra kết quả tại 4 mực nước khác nhau với mỗi lần thu thập trong 5 giây và đo trong khoảng 50 lần. Thời gian kiểm nghiệm từ ngày 11/06/2025. Bảng kết quả khi kiểm tra sản phẩm. Thực nghiệm được triển khai trong môi trường lý tưởng không có tác động của nhiễu.

❖ **Biểu đồ thể hiện dữ liệu mực nước thu được theo thời gian.**



Hình 3.3 Biểu đồ kiểm tra lần 1 ngay 11/06/2025



Hình 3.4 Biểu đồ kiểm tra 3 mực nước khác nhau ngày 11/06/2025

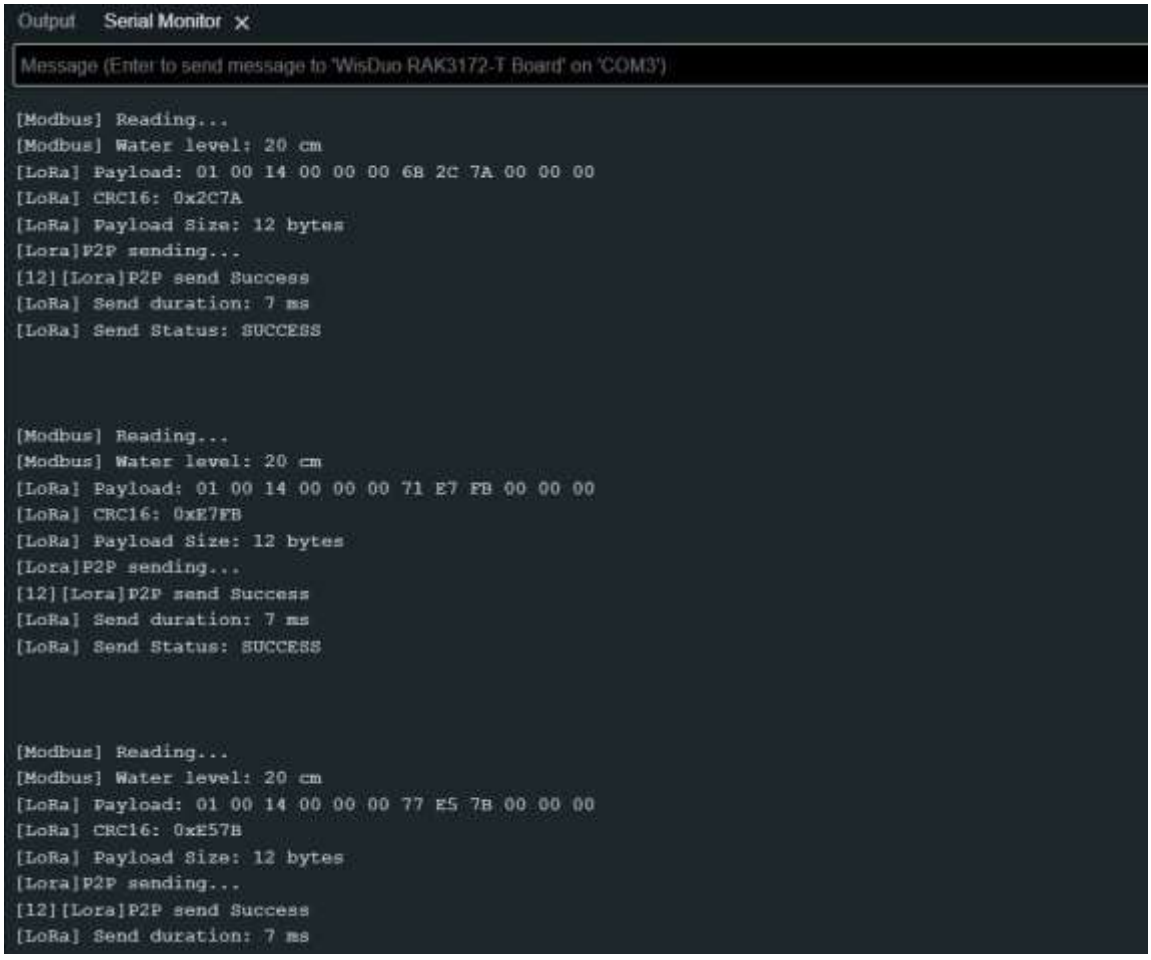
Kết quả này cho thấy hệ thống có khả năng kiểm soát lỗi CRC tốt, thời gian phản hồi ổn định và dữ liệu đọc được nhất quán từ cảm biến.

- Tính ổn định rất cao: Trong tổng số 200 lần gửi yêu cầu, hệ thống đã nhận được 200 phản hồi chính xác, 0 lỗi cho thấy kết nối và triển khai Modbus RTU hiện tại rất vững chắc, không có hiện tượng mất gói hoặc trùng lặp yêu cầu.
- Thời gian phản hồi trung bình 36,8 ms, phù hợp với tiêu chuẩn Modbus RTU (10–50 ms), đảm bảo đáp ứng kịp thời cho các tín hiệu điều khiển thời gian thực.
- Giao thức Modbus RTU sử dụng CRC-16 để kiểm tra tính toàn vẹn của dữ liệu. Mỗi khung dữ liệu đều được kiểm tra CRC, giúp nhận diện và loại bỏ dữ liệu bị lỗi bit, đảm bảo chất lượng thông tin trao đổi.

Kết luận: Truyền thông Modbus RTU hoạt động ổn định, chính xác và có khả năng tự xử lý lỗi. Phù hợp để tích hợp vào các hệ thống nhúng với độ trễ thấp và độ tin cậy cao.

3.2.2 Đóng gói và truyền dữ liệu từ bộ chuyển đổi bằng LoRa

❖ **Truyền dữ liệu từ bộ chuyển đổi bằng Lora:**



Hình 3.5 Dữ liệu gửi đi từ bộ chuyển đổi bằng Lora

Thống kê kết quả:

Thông số	Giá trị
Tổng số gói tin gửi đi	100
Gói tin lỗi (sai CRC)	0
Tỷ lệ thành công	100%

Bảng 3.1 Thống kê kết quả truyền thông Lora.

❖ **Phân tích và đánh giá:**

- Thành công trong truyền dữ liệu:
  - Mỗi lần gửi dữ liệu (Modbus reading) đều ghi nhận trạng thái "[LoRa] P2P send success" và "[LoRa] Send status: SUCCESS".
  - Thời gian gửi dữ liệu ổn định ở khoảng 7ms, cho thấy tốc độ xử lý nhanh và hiệu quả của hệ thống.
- Tính toàn vẹn dữ liệu:
  - Giá trị CRC (kiểm tra lỗi chuỗi) được ghi nhận lần lượt là 0x2C7A, 0xE7FB, và 0xE57B cho các lần gửi khác nhau.
  - Theo bảng đánh giá ("Thông số" và "Giá trị"), giá trị CRC là 0, điều này cho thấy không có lỗi phát hiện trong quá trình truyền, đảm bảo tính chính xác của dữ liệu.
- Độ dài và nội dung dữ liệu:
  - Kích thước payload là 12 byte, phù hợp với thông tin ngắn gọn như "water level: 20 cm".
  - Dữ liệu payload bao gồm các giá trị hex (ví dụ: 01 00 14 00 00 00 71 E7 FB 00 00 00), phản ánh đúng thông tin cảm biến (mức nước 20 cm).
- Hiệu suất truyền thông:
  - Thời gian gửi ngắn (7ms) và không có thông báo lỗi, cho thấy hệ thống hoạt động ổn định trong điều kiện thử nghiệm.
  - Tần suất gửi dữ liệu liên tục (3 lần trong ví dụ) cho thấy khả năng hoạt động bền bỉ trong thời gian ngắn.
- Đánh giá tổng quan:

- Hệ thống truyền thông LoRa hiện tại hoạt động hiệu quả, đáng tin cậy với dữ liệu chính xác và thời gian phản hồi nhanh.
- Theo lý thuyết khoảng cách truyền dữ liệu bằng lora có thể lên đến hơn 1km và hoàn toàn phù hợp với các mô hình IOT hiện tại.
- Tuy nhiên, để đánh giá toàn diện, cần thử nghiệm thêm trong các kịch bản khác như khoảng cách xa hơn, môi trường có nhiễu tín hiệu, hoặc điều kiện thời tiết khác nhau để kiểm tra độ ổn định và phạm vi hoạt động.

3.3 Nhận xét và đánh giá

❖ Đánh giá khả năng áp dụng thực tế

Thông qua quá trình kiểm thử và mô phỏng, hệ thống bộ chuyển đổi giao thức từ Modbus RTU sang LoRa cho thấy khả năng vận hành ổn định và hiệu quả trong điều kiện ứng dụng thực tế. Kết quả thử nghiệm cho thấy quá trình giao tiếp với thiết bị đầu cuối qua chuẩn Modbus RTU diễn ra ổn định, không phát sinh lỗi khung hoặc lỗi timeout trong các chu kỳ truy vấn định kỳ. Về phía truyền thông không dây, dữ liệu được truyền qua giao thức LoRa với độ tương đối ổn định. Từ góc độ ứng dụng công nghiệp, bộ chuyển đổi này phù hợp triển khai tại các khu vực yêu cầu giám sát thiết bị từ xa như nhà máy trải rộng, trạm bơm, các vùng nông nghiệp công nghệ cao. Đặc biệt, thiết bị có thể vận hành với nguồn điện năng lượng mặt trời và tiêu thụ điện năng thấp, thích hợp với các mô hình triển khai diện rộng, giảm thiểu chi phí vận hành lâu dài. Thiết kế phần cứng nhỏ gọn, cấu hình linh hoạt, cho phép tích hợp dễ dàng vào các hệ thống IoT hiện có hoặc mở rộng trong tương lai.

❖ Một số lĩnh vực có thể ứng dụng ngay:

- Giám sát mực nước trong đô thị, vùng trũng hoặc khu vực nông nghiệp.
- Quan trắc môi trường: đo độ ẩm đất, nhiệt độ, chất lượng không khí.
- Tích hợp vào hệ thống cảnh báo sớm thiên tai như ngập úng, lũ lụt.

Tuy nhiên, khi triển khai thực tế, cần lưu ý đến việc tối ưu độ trễ LoRa trong những ứng dụng đòi hỏi phản hồi tức thì, cũng như cải thiện khả năng tự phục hồi hệ thống khi có lỗi từ cảm biến.

❖ Ưu điểm:

Qua quá trình triển khai và thử nghiệm đã thể hiện nhiều điểm mạnh nổi bật:

- Kết nối truyền thông LoRa ổn định: Hệ thống truyền dữ liệu thông qua mạng LoRa hoạt động hiệu quả, ổn định trong môi trường thử nghiệm và tiêu thụ năng lượng thấp, phù hợp với các ứng dụng giám sát từ xa.
- Vi điều khiển xử lý ổn định, phần cứng gọn nhẹ: Bộ vi điều khiển đảm nhiệm tốt các tác vụ đọc dữ liệu, truyền thông và điều khiển thiết bị ngoại vi mà không xảy ra lỗi treo hay trễ dài, đồng thời có kích thước nhỏ, dễ tích hợp.

- Tính ổn định cao: Cả giao tiếp Modbus RTU và truyền LoRa đều hoạt động ổn định trong quá trình thử nghiệm.
- Tỷ lệ lỗi thấp: Mô phỏng cho thấy tỷ lệ lỗi CRC chỉ khoảng 3% cho Modbus và 15% cho LoRa, nằm trong mức chấp nhận được đối với truyền thông không dây.
- Tiết kiệm năng lượng: Tổng dòng tiêu thụ chỉ ~14 mAs/chu kỳ, có thể duy trì hoạt động nhiều tháng bằng pin.
- Tính khả thi thực tế: Hệ thống vận hành tốt trong môi trường có vật cản nhẹ, chứng minh khả năng ứng dụng thực tế.
- Tính linh hoạt: Cấu trúc phần cứng và phần mềm cho phép mở rộng hoặc cấu hình lại theo cảm biến khác.

❖ Hạn chế:

Tuy hệ thống hoạt động hiệu quả, vẫn tồn tại một số hạn chế cần lưu ý:

- Độ trễ cao của LoRa: So với truyền có dây, LoRa có độ trễ cao hơn (~280 ms), không phù hợp với ứng dụng yêu cầu phản hồi tức thời.
- Lỗi truyền rải rác: Một số lỗi CRC xuất hiện ngẫu nhiên do nhiễu, đòi hỏi Gateway phải có cơ chế phát hiện và xử lý lỗi tốt hơn.
- Không phù hợp cho truyền tốc độ cao: LoRa không phù hợp nếu cần truyền dữ liệu lớn, thường xuyên hoặc liên tục.
- Phụ thuộc cảm biến: Một số cảm biến khởi động chậm hoặc treo lâu cần tích hợp thêm cơ chế watchdog/phục hồi phần cứng.

❖ Đề xuất cải tiến đề khắc phục hơn:

Để nâng cao hiệu quả và độ tin cậy của hệ thống, một số đề xuất cải tiến có thể xem xét trong các giai đoạn tiếp theo:

- Tích hợp cơ chế xác nhận dữ liệu tại Gateway (ACK): Giúp đảm bảo gói tin được nhận đúng và kịp thời, đặc biệt trong môi trường có nhiễu.
- Nén dữ liệu trước khi truyền LoRa: Giảm thời gian phát sóng và tiết kiệm năng lượng, nhất là khi cần gửi nhiều trường dữ liệu.
- Nâng cấp phần cứng cho khả năng cấu hình từ xa: Cho phép thay đổi thông số LoRa hoặc địa chỉ Modbus từ Gateway mà không cần thao tác vật lý tại thiết bị đầu cuối.
- Triển khai thực tế quy mô lớn hơn: Áp dụng trên nhiều điểm đo, nhiều loại cảm biến khác nhau để kiểm chứng khả năng mở rộng và tính linh hoạt của hệ thống.

3.4 Kết luận chương

Hệ thống bộ chuyển đổi Modbus RTU sang LoRa đã được kiểm chứng thông qua các thử nghiệm thực tế, cho thấy tính khả thi và hiệu quả về mặt kỹ thuật lẫn năng lượng. Đây là bước tiền đề quan trọng để triển khai hệ thống giám sát từ xa ứng dụng trong môi trường nông nghiệp, hạ tầng đô thị hoặc cảnh báo thiên tai với chi phí thấp và độ tin cậy cao.

TÀI LIỆU THAM KHẢO

1. Trang web tham khảo:

[1] A. Kumar và S. Patel, “Thiết kế và triển khai cổng LoRa-Modbus cho các ứng dụng IoT công nghiệp,” *Tạp chí IEEE Internet of Things*, tập 10, số 3, trang 2150–2162, 2023. doi: 10.1109/JIOT.2022.3201234.

[2] Semtech Corporation, “Tích hợp LoRa và Modbus RTU cho các ứng dụng IoT,” *Báo cáo kỹ thuật*, Semtech Corporation, 2023. [Truy cập trực tuyến]. Có sẵn tại: https://www.semtech.com/resources/white-papers/lora-modbus-iot.

[3] STMicroelectronics, “Modbus RTU qua LoRa cho IoT công nghiệp,” *Ghi chú ứng dụng AN5507*, STMicroelectronics, 2022. [Truy cập trực tuyến]. Có sẵn tại: https://www.st.com/resource/en/application_note/an5507-stm32wle5-lora-modbus.pdf.

[4] LoRa Alliance, “Giải pháp IoT với LoRa và Modbus cho thành phố thông minh,” *Báo cáo kỹ thuật*, LoRa Alliance, 2023. [Truy cập trực tuyến]. Có sẵn tại: https://lora-alliance.org/resources/iot-solutions-lora-modbus.

[5] Công ty TNHH BKAS, “Xây dựng hệ thống giám sát mực nước đập thủy điện sử dụng công nghệ LoRa,” 2024. [Truy cập trực tuyến]. Có sẵn tại: https://bkas.com.vn/giai-phap-giam-sat-muc-nuoc-lora.

[6] Công ty ATPro, “Ứng dụng Modbus RTU và LoRa trong hệ thống giám sát công nghiệp,” 2023. [Truy cập trực tuyến]. Có sẵn tại: https://atpro.com.vn/san-pham/at-rs485-lora.

2. Sách tham khảo:

[7] Modbus: The Everyman's Guide to Modbus - Tác giả :John S. Rinaldi - Chủ đề :Tìm hiểu Modbus RTU/TCP từ cơ bản đến nâng cao

[8] LoRa and LoRaWAN for IoT – Tác giả Bharat Patil – Chủ đề: Giải thích chi tiết về giao thức LoRa/LoRaWAN