Software Design Document

Tài liệu Thiết kế Phần mềm

DataProcessing_HC12_ver1

Mã tài liệu:
Tổng số trang: 27 trang

BẢNG GHI NHẬN THAY ĐỔI

Phiên bản	Ngày	Vị trí thay đổi	Mô tả thay đổi	Người thay đổi	Người xem xét	Người phê duyệt
1						

MỤC LỤC

DANH MỤC HÌNH ẢNH

Lời cảm ơn

Em xin chân thành cảm ơn các anh Kiên, anh Hạnh, anh Tài và anh Trung đã tận tình hỗ trợ và hướng dẫn em trong quá trình thực hiện dự án nhỏ này. Nhờ sự giúp đỡ của các anh, em không chỉ hoàn thành được bản demo mà còn học hỏi thêm rất nhiều kiến thức thực tiễn về hệ thống nhúng và truyền thông không dây.

"There's nothing more rewarding than working with someone who is smarter than you are, so go find people who make you feel stupid in comparison." - Rusell Merrick.

Làm việc với những người giỏi hơn mình là cơ hội quý báu để trưởng thành - và em cảm thấy rất may mắn vì đã được học hỏi từ các anh trong suốt dự án này.

1. Giới thiệu

Báo cáo này về triển khai demo thành công một hệ thống Datalink truyền nhận bản tin không dây giữa hai vi điều khiển STM32, sử dụng mô-đun truyền thông không dây HC-12. Hệ thống gồm hai thành phần chính: khối Phát (Transmitter) và khối Thu (Receiver), kết nối với nhau qua giao tiếp UART.

Mục tiêu bước đầu của dự án là kiểm nghiệm tính thực tiễn và khả thi của hệ thống, trước khi mở rộng triển khai trên các thiết bị phần cứng mạnh hơn, phục vụ cho các ứng dụng truyền thông thực tế.

Tài liệu "Thiết kế phần mềm" mô tả chi tiết các thành phần, chức năng, cũng như luồng hoạt động của hệ thống đã được triển khai.

1.1 Mục đích tài liệu

Tài liệu này nhằm:

- Mô tả mối quan hệ chức năng giữa các thành phần chính như: **STM32**, **USB-UART**, **HC-12**.
- Trình bày chi tiết các chức năng được cài đặt trong **khối Phát** và **khối Thu**.
- Hỗ trợ cho việc **bảo trì, nâng cấp**, và **phát triển phiên bản tiếp theo** của phần mềm.

Đối tượng sử dụng tài liệu cụ thể như sau:

Người sử dụng	Mục đích
Người dùng	Hiểu rõ nguyên lý hoạt động và cấu trúc hệ thống.
Nhóm phát triển	Tham khảo để kiểm thử và cập nhật hệ thống.
Nhóm kiểm thử	Dùng để kiểm tra tính đúng đắn của các chức năng phần mềm.
SQA	Đảm bảo chất lượng phần mềm trong quá trình phát triển.

1.2. Phạm vi tài liệu

Tài liệu này được sử dụng trong phạm vi dự án DataProcessing_HC12_ver1.

1.3. Khái niệm, thuật ngữ, viết tắt

ID	Acronyms		Definition
1	STM32	STMicroelectronics 32-bit Microcontroller	
2	USB UART	Universal Serial Bus to Universal Asynchronous Receiver-Transmitter	Mạch chuyển đổi tín hiệu giữa cổng USB của máy tính và giao tiếp UART (TX/RX) của vi điều khiển

3	HC-12	433 MHz Wireless Serial Communication Module	Module truyền thông không dây tầm xa hoạt động ở tần số 433 MHz.
4	USB ST-link	STMicroelectronics In-Circuit Debugger and Programmer	Thiết bị lập trình và gỡ lỗi cho các vi điều khiển STM32
5	COM	Communication Port	Cổng nối tiếp (Serial Port)
6	IDE	Integrated Development Environment	Môi trường phát triển tích hợp.
7	Baudrate		Tốc độ truyền dữ liệu trên một kênh truyền thông nối tiếp (UART, RS-232, HC-12,..)
8	RCC	Reset and Clock Control	Module điều khiển hệ thống reset và xung nhịp
9	HSE	High-Speed External (Clock)	Xung nhịp ngoài (thạch anh)
10	DMA	Direct Memory Access	Truy cập bộ nhớ trực tiếp
11	PKCS7	Public-Key Cryptography Standards 7	Chuẩn đệm dữ liệu
12	AES256	Advanced Encryption Standard (256-bit key)	Thuật toán mã hóa khối sử dụng khóa 256 bit
13	Bit Manipulation		Xử lý bit
14	LDPC 1/2	Low Density Parity Check 1/2	Mã khối mật độ thấp 1/2
15	LSFR	Linear Feedback Shift Register	Mạch sinh chuỗi nhị phân giả ngẫu nhiên

Tài liệu tham khảo

STT	Reference	Version Number	Document Name/Source & Location reference	Mô tả	Remarks
1	KienNguyen		DataProcessing		

2. Mô tả tổng quan

2.1 Tổng thể hệ thống

Hệ thống được thiết kế nhằm thử nghiệm một liên kết truyền thông không dây (Datalink) giữa hai vi điều khiển STM32 sử dụng mô-đun HC-12. Cấu hình tổng thể bao gồm cả phần cứng và phần mềm như sau:

Phần cứng:

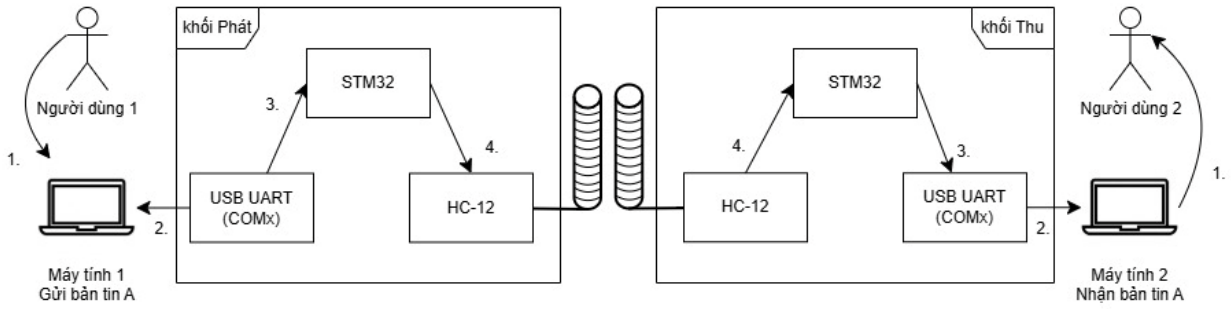
- **STM32F103C8T6:** 2 bộ (1 cho Transmitter, 1 cho Receiver).
- **USB ST-Link:** 2 bộ, dùng để nạp code cho STM32.
- **HC-12 (CP102):** 2 module truyền thông không dây.
- **USB UART (TTL to RS232 – loại 6 mode):** 2 bộ, đảm bảo việc truyền/nhận UART giữa máy tính (Hercules) và STM32.

Mỗi khối **Phát** và **Thu** bao gồm:

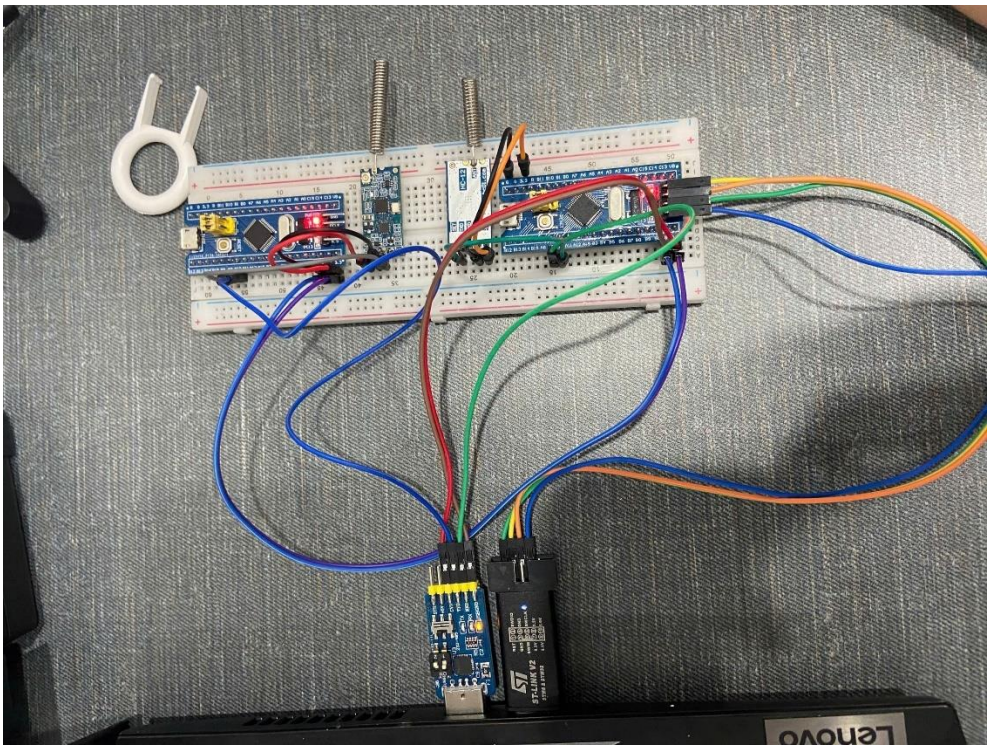
- 1 STM32.
- 1 HC-12.
- 1 USB UART.
- 1 USB ST-Link.

Phần mềm:

- Triển khai code cho STM32 sử dụng STM32 CubeMX, STM32 IDE, ST-link Utility.
- Hercules: Dùng để gửi bản tin, cấu hình cổng COM và điều khiển HC-12 thông qua AT command.



Hình 1: Ảnh demo dự án trên lý thuyết



Hình 2: Ảnh demo dự án thực tế

(tối giản - sử dụng 1 USB UART test trên 1 máy tính)

STT	Standard	Mandatory Specifications/ Parameters	Options considered	Options not in scope	Remarks
1	Giao tiếp	UART			UART phù hợp HC-12
2	Baudrate	115200			

2.2 Mô hình người dùng, ngữ cảnh hoạt động của hệ thống

Phần mềm sẽ có 2 người sử dụng:

- Người dùng 1: Khối Phát.
- Người dùng 2: Khối Thu.

Tác vụ người dùng:

Cấu hình trên Hercules:

- Cấu hình cổng COM: Chọn cổng COM tương ứng với thiết bị USB-UART được máy tính nhận diện. Cổng COM này có thể khác nhau tùy theo cấu hình trên từng máy.
- Cấu hình Baudrate: 115200 (Tạm thời mặc định).

- Cấu hình Data size: 8.
- Cấu hình Parity: None.
- Cấu hình Handshake: None.
- Cấu hình Mode: Free.

Cấu hình HC-12 (thông qua Hercules):

- Chân SET kết nối GND để đưa HC-12 vào chế độ AT Command. Reset các thông số của HC-12 về mặc định (Baudrate 9600, Stopbit 1, Parity, None,..).
- Dùng AT -> sẽ trả về Oke (Kiểm tra kết nối).
- Dùng AT + Bxxxx -> thay đổi Baudrate (VD AT+B115200).
- Dùng AT+Cxxx (với xxx là số kênh từ 000 đến 127) -> thay đổi Kênh (VD AT+C100).
- Dùng AT+Axxx (với xxx từ 000 đến 255) -> thay đổi địa chỉ (VD: AT+A200).
  *Lưu ý: 2 module HC-12 cần cùng kênh mới kết nối được, khối Phát và khối Thu cần có 2 địa chỉ không được trùng nhau.

Cấu hình STM32:

- Cấu hình Baudrate: 115200.
- Cấu hình Data size: 8bit.
- Cấu hình Parity: None.
- Cấu hình Stopbit: 1.
- Sử dụng Debug: Serial Wire.
- Bật RCC HSE: CrystalCeramic Resonator.
- Bật DMA

Các ngữ cảnh hoạt động của hệ thống:

- Người vận hành 1,2 cần cấu hình các tham số hoạt động của hệ thống.
- **Người vận hành 1** gửi 1 message A dưới dạng hex qua Hercules tới USB UART đến STM32. STM32 xử lý và truyền bản tin qua HC-12. (Khối Phát).
- Bên nhận nhận bản tin qua HC-12, truyền vào STM32. STM32 xử lý bản tin rồi gửi qua USB UART và hiển thị trên máy tính **Người vận hành 2** bằng Hercules. (Khối Thu).

3. Yêu cầu chức năng

Chức năng của DataProcessing_HC12_ver1 bao gồm:

Khối Phát:

1, STM32 lắng nghe bản tin thông qua Hercules: Người dùng 1 gửi bản tin thông qua Hercules.

2, STM32 nhận bản tin và xử lý (Encode): STM32 nhận được bản tin từ Người vận hành 1 thông qua USB UART.

3, Bản tin được mã hóa gửi đến HC-12: Bản tin sau khi được mã hóa gửi đến HC-12 (đã được cấu hình) chuẩn bị truyền qua vô tuyến.

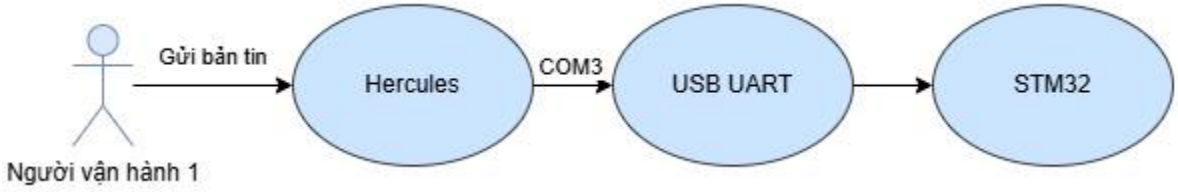
Khối Thu:

4, HC-12 nhận bản tin đã mã hóa: Bản tin đã mã hóa được truyền qua vô tuyến tới HC-12 của khối Thu.

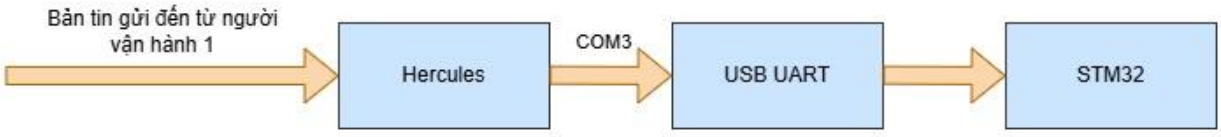
5, Decode bản tin mã hóa qua STM32: Bản tin đã mã hóa truyền đến STM32 của khối Thu, bản tin ở đây sẽ được giải mã (Decode).

6, Gửi bản tin hoàn chỉnh tới Hercules: Gửi bản tin đã giải mã từ STM32 của khối Thu tới Hercules thông qua USB UART, hiển thị bản tin cho Người dùng 2.

3.1 STM32 lắng nghe bản tin thông qua Hercules



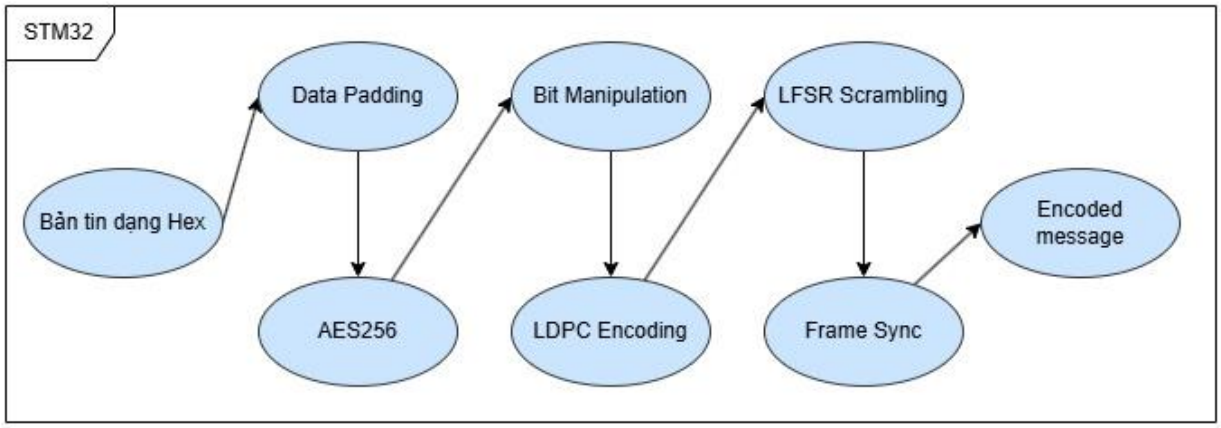
Hình 3a: Sơ đồ use case STM32 (khối Phát) lắng nghe bản tin thông qua Hercules.



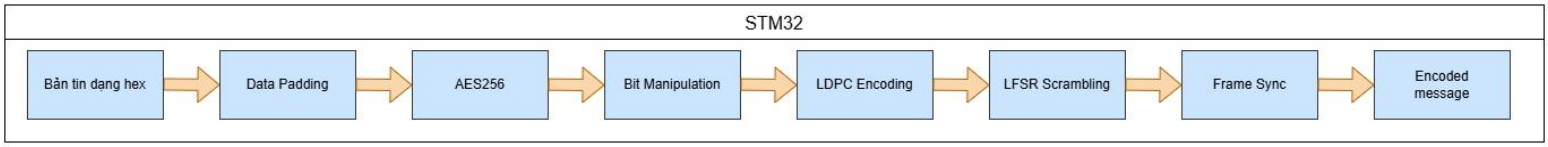
Hình 3b: Sơ đồ luồng STM32 (khối Phát) lắng nghe bản tin thông qua Hercules.

ID	
Mô tả	Người vận hành 1 gửi bản tin bằng Hercules (qua cổng COM3) tới USB UART (chế độ TTL-RS232). Sau đó bản tin được gửi đến STM32.
Đầu vào	Bản tin dạng hex.
Xử lý	1. Bản tin gửi đến từ người vận hành 1 qua Hercules và USB UART sẽ qua chân TX USB UART từ đó tới RX của STM32. 2. STM32 ở khối Phát sử dụng DMA sẵn sàng lắng nghe từ chân RX của STM32.
Đầu ra	Bản tin dạng Hex được lưu vào trong STM32.

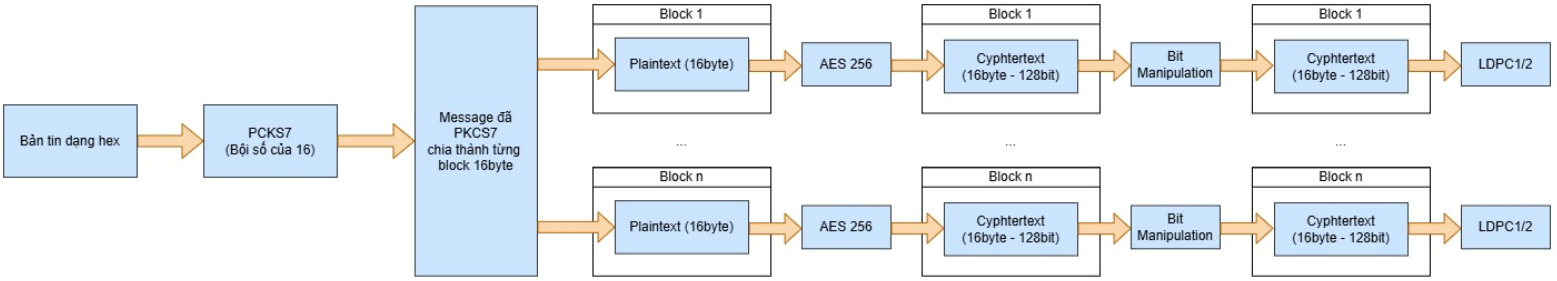
3.2 STM32 nhận bản tin và xử lý (Encode)



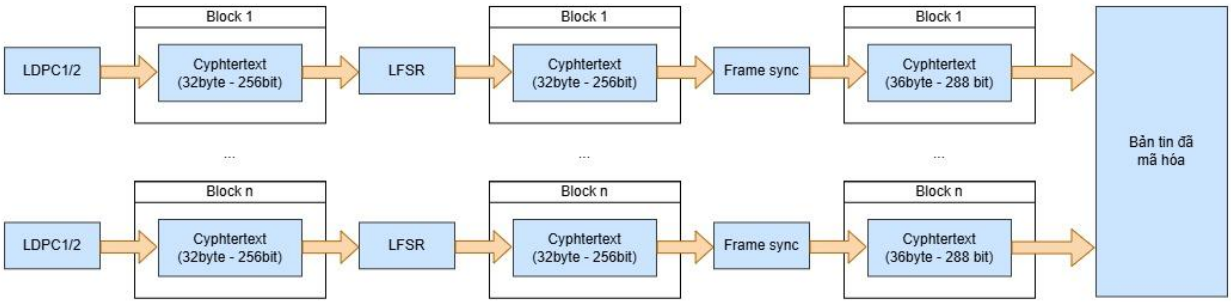
Hình 4a: Sơ đồ use case STM32 nhận bản tin và xử lý (Encode).



Hình 4b: Sơ đồ luồng STM32 nhận bản tin và xử lý (Encode).



Hình 5a: Sơ đồ input output quá trình mã hóa bản tin.



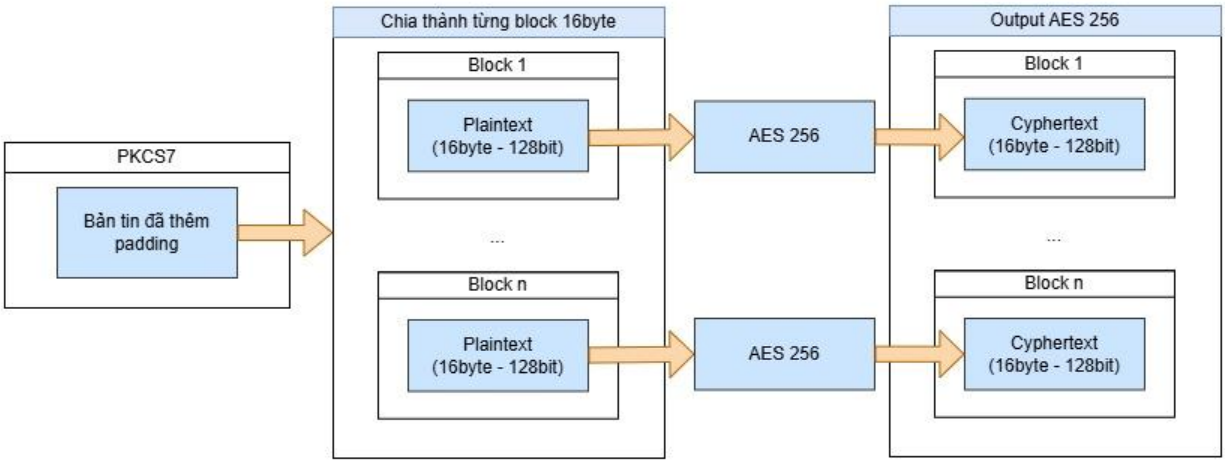
Hình 5b: Sơ đồ input output quá trình mã hóa bản tin.

ID	
Mô tả	Bản tin dạng Hex sau khi TX từ Hercules qua USB UART đến STM32 sẽ được mã hóa nhằm bảo mật thông tin bản tin.
Đầu vào	Bản tin dạng Hex đã được lưu trong STM32 (khối Phát)
Xử lý	Sử dụng các bước encode (mã hóa) sau đây: 1. Data Padding: Sử dụng thuật toán PKCS7. Đây là bước chuẩn bị dữ liệu cho quá trình mã hóa AES 256. 2. AES 256: Mã hóa dữ liệu theo chuẩn AES-256 ở chế độ ECB. 3. Bit Manipulation: Biến đổi các bit nhằm làm rối luồng dữ liệu. 4. LDPC Encoding: Mã hóa sửa lỗi chuyển tiếp (8×16). 5. LFSR Scrambling: Xáo trộn dữ liệu bằng bộ tạo dãy giả ngẫu nhiên LFSR. 6. Frame Sync: Thêm các dấu hiệu đồng bộ khung.
Đầu ra	Bản tin dạng Hex đã mã hóa.

3.2.1 Data Padding (PKCS7) Encode

ID	
Mô tả	Bản tin dạng Hex cần được thêm Padding nhằm dễ dàng phục vụ cho bước AES 256.
Đầu vào	Bản tin dạng Hex.
Xử lý	1.Bản tin dạng Hex cần là số chia hết cho 16. Để giúp thực hiện bước AES 256 đơn giản hơn. 2.Nếu độ dài bản tin chưa phải bội của 16, ta thêm các byte có giá trị bằng số byte cần thêm. Ví dụ 18byte -> cần thêm 14byte để thành 32byte (bội của 16) -> thêm 14 byte với mỗi byte giá trị là 0x0E (số 14 dạng Hex). 3.Nếu độ dài bản tin đã là bội của 16, PKCS7 vẫn sẽ thêm 1 block padding có giá trị là 0x10 (số 16 dạng Hex).
Đầu ra	Bản tin dạng Hex đã thêm padding PKCS7 với độ dài là bội số của 16.

3.2.2 AES256 Encode



Hình 6: Sơ đồ luồng AES 256 Encode.

ID	
Mô tả	Bản tin sau khi thêm padding sẽ được mã hóa bởi thuật toán AES256 (Mode ECB)
Đầu vào	Bản tin sau khi thêm padding (bội của 16).
Xử lý	1 Mã hóa AES-256 (Encrypt) Tính Round_key (240 byte – 60word) cũng được tách thành các block 16byte. (Đủ cho 15 round). Mã hóa từng block plantext (16byte) qua 15 round (từ Round_key 240 byte).
Đầu ra	Bản tin dạng Hex đã được mã hóa AES 256.

3.2.2.1 AES256 Key expansion

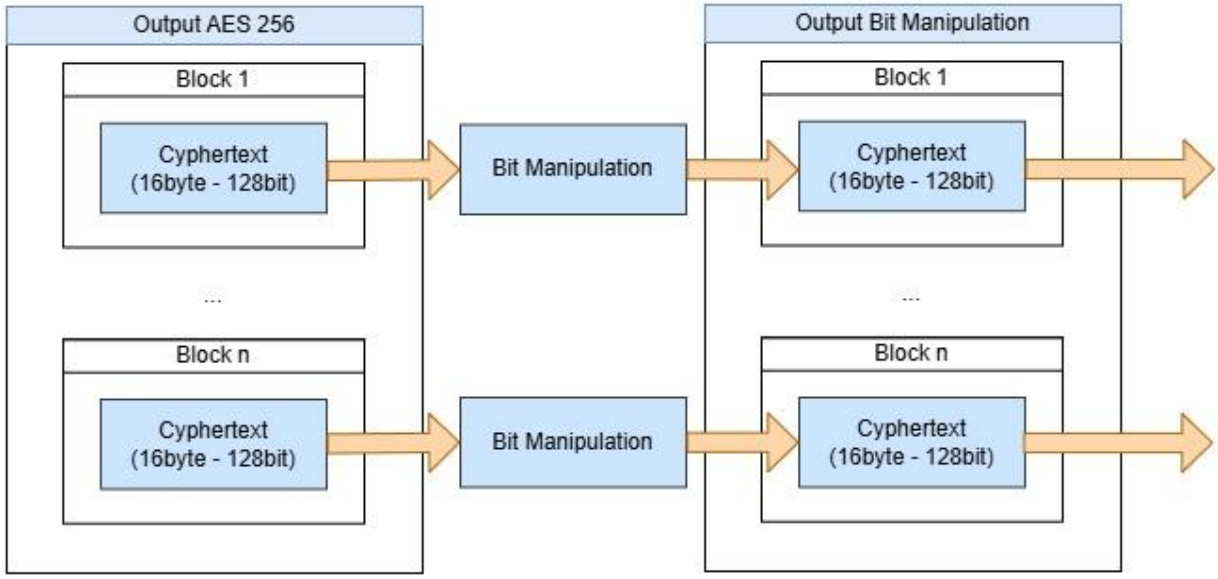
ID	
Mô tả	Bước đầu cho AES 256, tính toán Key expanstion.
Đầu vào	32 byte Key chính.
Xử lý	Tính khóa vòng (Key Expansion) Từ khóa chính 256 (bit) – 32byte -> Sẽ tạo ra 60 từ khóa con (với 32byte khóa chính là 8 từ khóa con) -> Cần tạo ra 52 từ khóa con (Tương đương 52 * 4 = 208 byte khóa) . - 4 byte (1 word) đầu tiên sẽ lấy từ 4byte cuối cùng của 32byte khóa chính và trải qua các bước: - Nếu word chia hết cho 8 thì: 1/ Rotate the word (left shift by one byte). 2/ Apply the S-box substitution. 3/ XOR the first byte with Rcon. - Nếu word chia 8 dư 4 thì: - 4 byte (1 word), từng byte sẽ lấy giá trị trong bảng Sbox. ➔ 4 byte sau khi được tạo mới, từng 4 byte sẽ XOR với word gần nhất.

Đầu ra	Tạo ra 240 byte Round_key (60 word).

3.2.2.2 AES256 Encrypt (15 rounds)

ID	
Mô tả	Các bước Encrypt (15 round) của AES 256.
Đầu vào	Bản tin đã thêm padding PKCS7 – là bội của 16.
Xử lý	- Tách từng block 16byte từ bản tin rồi mã hóa nó 15 vòng. 1/ Bản tin plaintext XOR với 16byte đầu tiên từ Round_key 2/ Bản tin sẽ trải qua 14 vòng mã hóa dùng: 1/ Áp dụng phép thay thế S-box. 2/ Dịch các hàng (Shift rows). 3/ Trộn các cột (Mix columns). 3/ Tới vòng cuối cùng bản tin sau khi được mã hóa 14 lần sẽ XOR với 16byte cuối từ Round_key.
Đầu ra	Block 16byte (128bit).

3.2.3 Bit Manipulation Encode

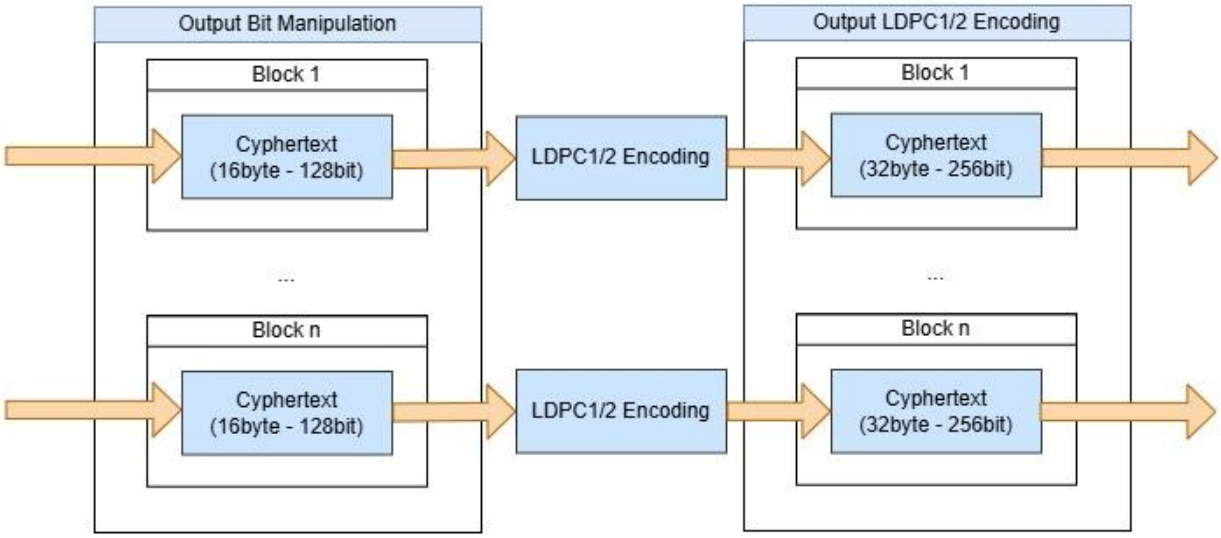


Hình 7: Sơ đồ luồng Bit Manipulation Encode.

ID	
Mô tả	Là bước thao tác bit trong mã hóa khi xử lý dữ liệu ở cấp độ bit. Giúp tăng hiệu suất, tạo sự hỗn loạn mạnh khi nó được nằm ngay sau AES256.
Đầu vào	Block 16byte (128bit) đã Bit Manipulation.
Xử lý	1. Trong 128 bit, đảo (invert) các bit 0 thành 1, 1 thành 0.

	2.Sau khi đảo, cứ 1 cặp bit sẽ hoán đổi vị trí cho nhau tới khi hết 128bit
Đầu ra	Block 128bit đã Bit Manipulation

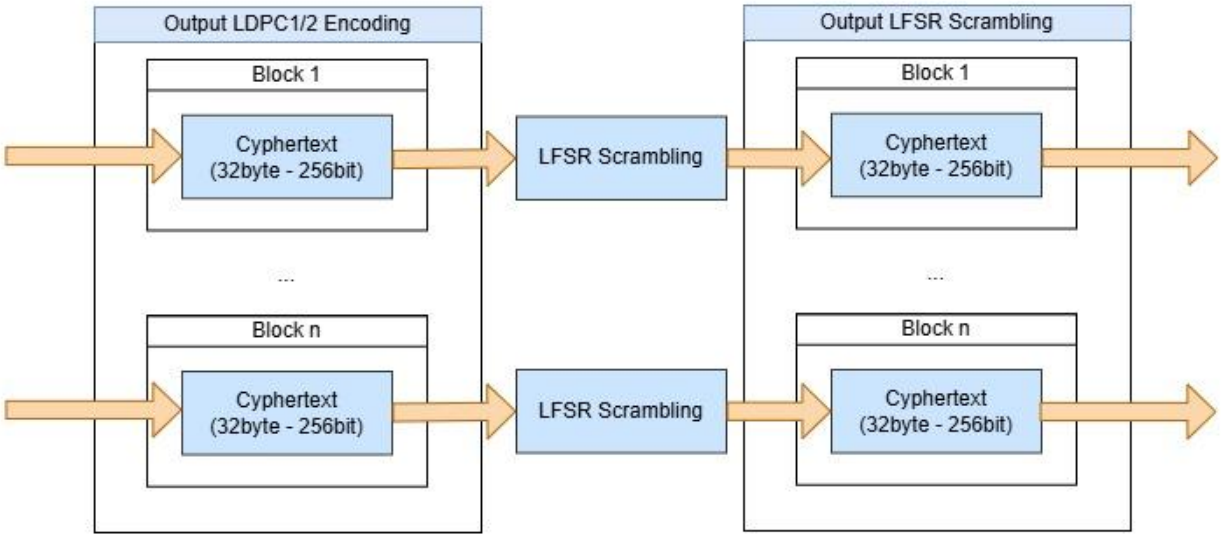
3.2.4 LDPC1_2 Encoding



Hình 8: Sơ đồ luồng LDPC1_2 Encoding.

ID	
Mô tả	LDPC1/2 là bước mã hóa giúp phát hiện và tự động sửa lỗi bit trong quá trình truyền dữ liệu.
Đầu vào	Block 128bit sau khi Bit Manipulation.
Xử lý	1.Thiết lập ma trận kiểm tra H cố định dùng để phát hiện lỗi dựa trên Syndrome. 2.Từng 8 bit sẽ được nhân với ma trận H để tạo thành 1 array u (1x16). Với 8bit đầu giữ nguyên, 8 bit sau sẽ tính parity để sau giúp cho việc tìm bit sai trong quá trình truyền không dây.
Đầu ra	Block 32 byte (256 bit) đã LDPC1/2.

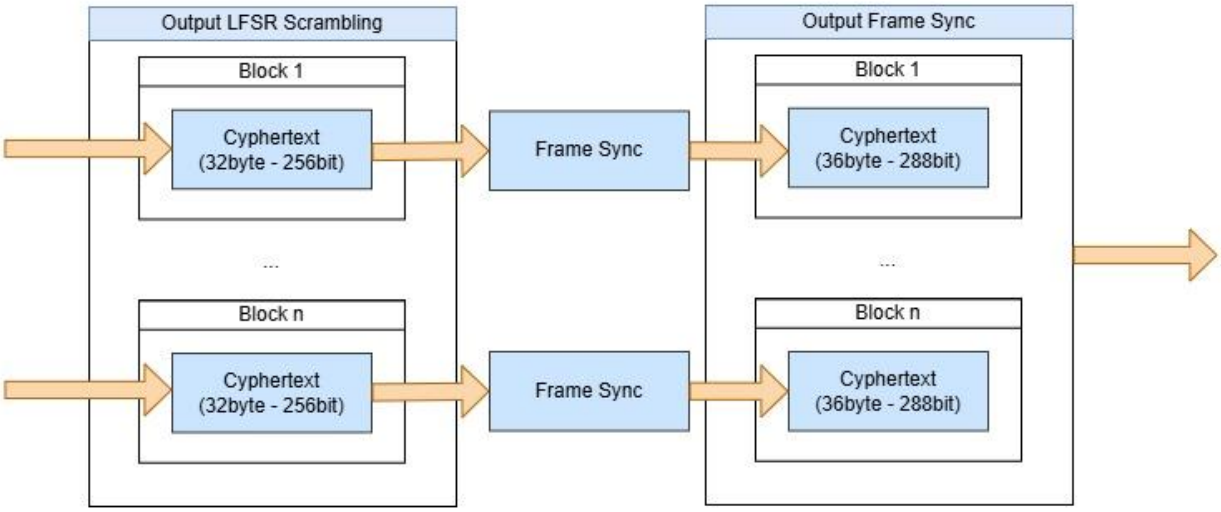
3.2.4 LFSR Scrambling Encode



Hình 9: Sơ đồ luồng LFSR Scrambling.

ID	
Mô tả	LFSR Scrambling là bước mã hóa kênh
Đầu vào	Block 256 bit từ bước LDPC1/2.
Xử lý	Sử dụng thanh ghi dịch tuyến tính (LFSR) 15-bit để tạo dãy giả ngẫu nhiên, sau đó XOR từng bit của block đầu vào với dãy này để làm nhiễu dữ liệu.
Đầu ra	Block 32 byte (256 bit) đã LFSR Scrambling.

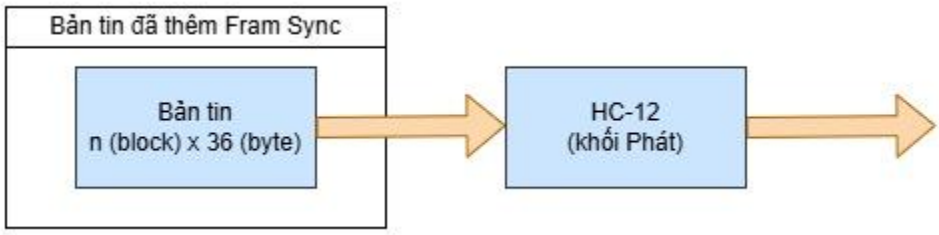
3.2.5 Sync Frame Encode



Hình 10: Sơ đồ luồng Sync Frame.

ID	
Mô tả	Sync Frame là bước chèn byte đồng bộ nhằm tạo cấu trúc khung rõ ràng cho dữ liệu sau LFSR Scrambling. Điều này giúp khối Thu dễ phát hiện vị trí bắt đầu của bản tin mã hóa để giải mã chính xác.
Đầu vào	Block 32 byte đã LFSR Scrambling.
Xử lý	Chèn 2 byte đồng bộ 0xEB 0x90 trước mỗi khối 16 byte dữ liệu. Đây là định dạng đồng bộ chuẩn khung PCM để hỗ trợ quá trình phát hiện và bóc tách dữ liệu ở phía khối Thu.
Đầu ra	Block 36 byte đã Fram Sync.

3.3 Bản tin được mã hóa gửi đến HC-12



Hình 11: Sơ đồ luồng Bản tin được mã hóa gửi đến HC-12.

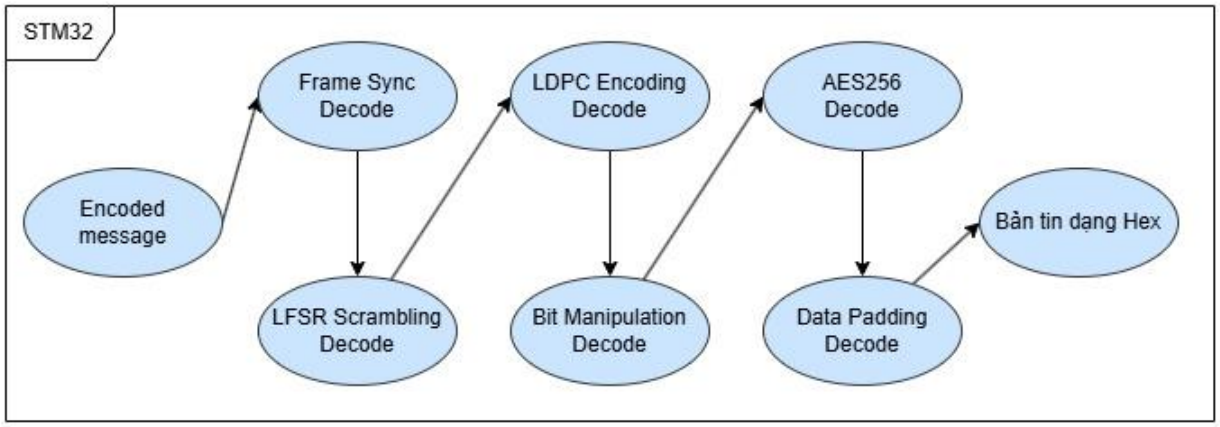
ID	
Mô tả	Sync Frame là bước chèn byte đồng bộ nhằm tạo cấu trúc khung rõ ràng cho dữ liệu sau LFSR Scrambling. Điều này giúp khối Thu dễ phát hiện vị trí bắt đầu của bản tin mã hóa để giải mã chính xác.
Đầu vào	Block 32 byte đã LFSR Scrambling.
Xử lý	Chèn 2 byte đồng bộ 0xEB 0x90 trước mỗi khối 16 byte dữ liệu. Đây là định dạng đồng bộ chuẩn khung PCM để hỗ trợ quá trình phát hiện và bóc tách dữ liệu ở phía khối Thu.
Đầu ra	Block 36 byte đã Fram Sync.

3.4 HC-12 nhận bản tin đã mã hóa

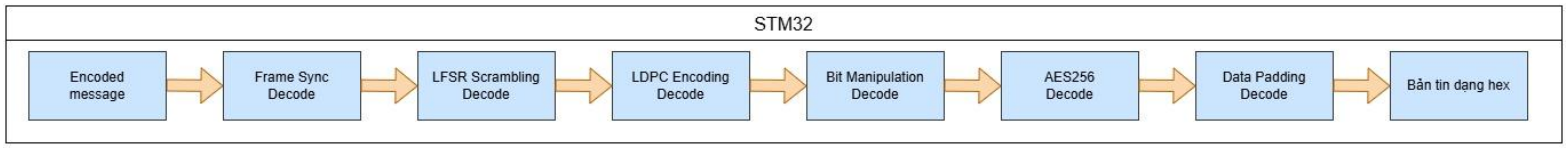
Hình 12: Sơ đồ luồng Bản tin được mã hóa gửi đến HC-12.

ID	
Mô tả	Bản tin đã mã hóa đi qua HC-12 của khối Phát -> qua vô tuyến -> tới HC-12 của khối Thu.
Đầu vào	Bản tin đã mã hóa.
Xử lý	Đi qua vô tuyến nhờ 2 ăng ten của HC-12. Với các config đã setup trên Hercules.
Đầu ra	Bản tin đã mã hóa.

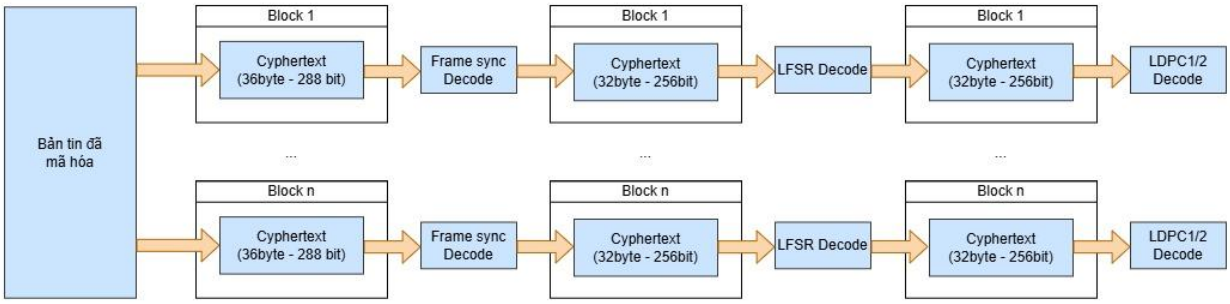
3.5 Decode bản tin mã hóa qua STM32



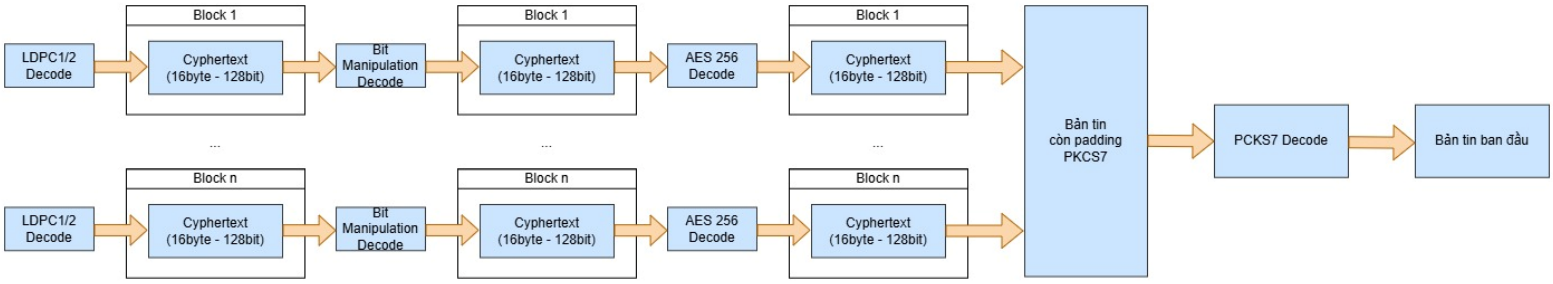
Hình 13a: Sơ đồ Use Case STM32 giải mã bản tin ở khối Thu.



Hình 13b: Sơ đồ luồng STM32 giải mã bản tin ở khối Thu.

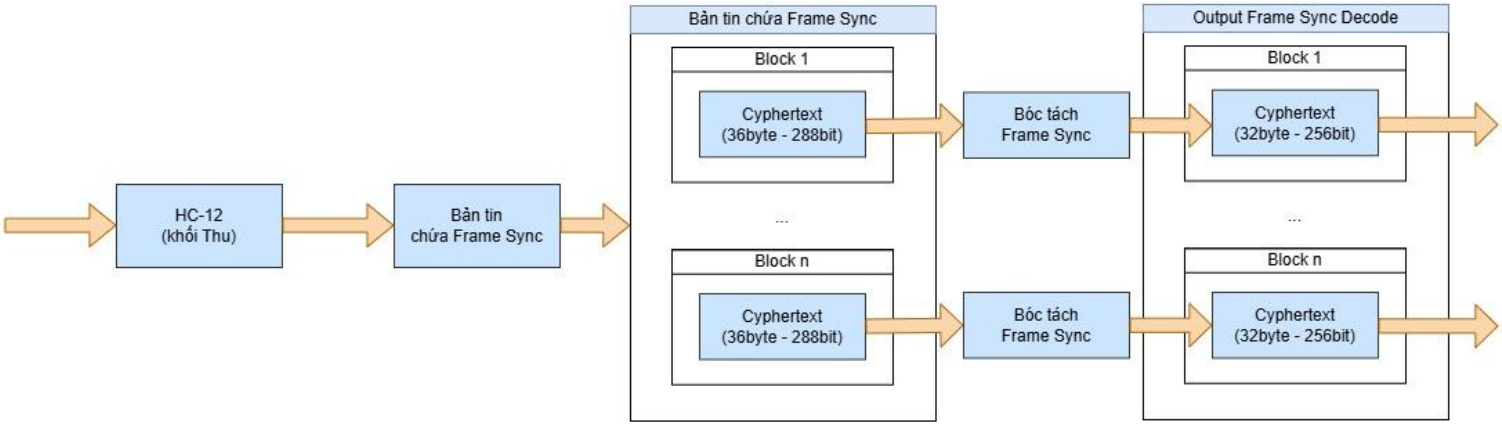


Hình 14a: Sơ đồ input output quá trình giải mã bản tin.



Hình 14b: Sơ đồ input output quá trình giải mã bản tin.

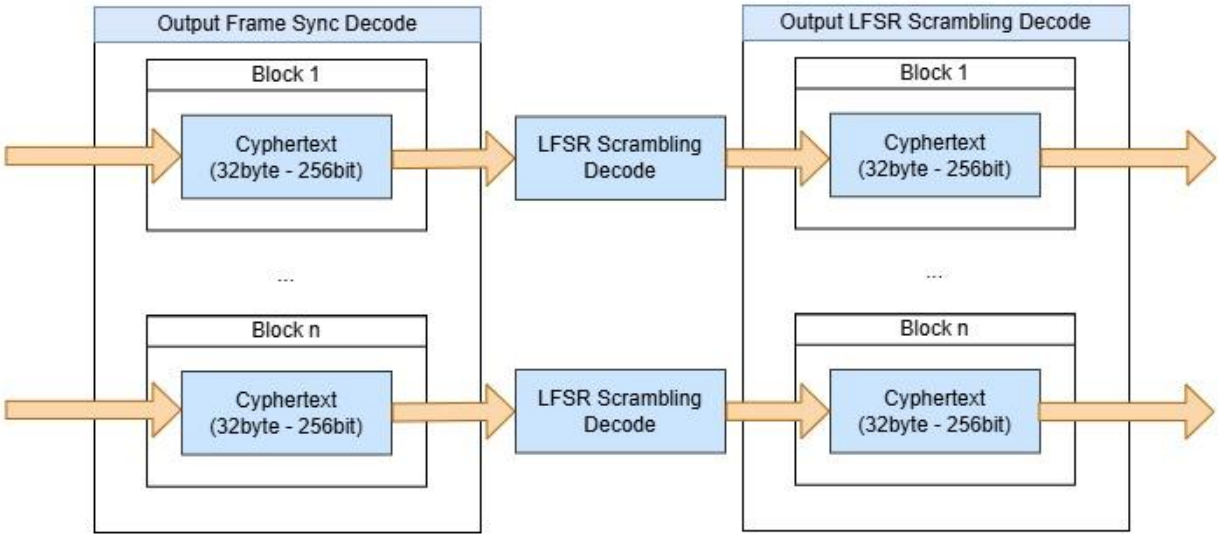
3.5.1 Frame Sync Decode



Hình 15: Sơ đồ luồng giải mã Frame Sync.

ID	
Mô tả	Bóc tách dữ liệu từ bản tin chứa Frame Sync.
Đầu vào	Bản tin sau mã hóa, chứa nhiều block 36 byte, trong đó mỗi block có 4 byte Frame Sync (cụ thể là 2 byte header 0xEB 0xEB chèn trước mỗi 16 byte dữ liệu).
Xử lý	1.Chia bản tin thành các block 36 byte. 2.Mỗi block chứa 32 byte dữ liệu và 4 byte header 0xEB 0x90 chèn trước mỗi 16 byte. 3.Loại bỏ 2 byte 0xEB 0xEB để khôi phục lại dữ liệu gốc. 4.Đầu ra: Các block 32 byte đã loại bỏ byte đồng bộ (Frame Sync).
Đầu ra	Các block 32 byte đã loại bỏ byte đồng bộ (Frame Sync).

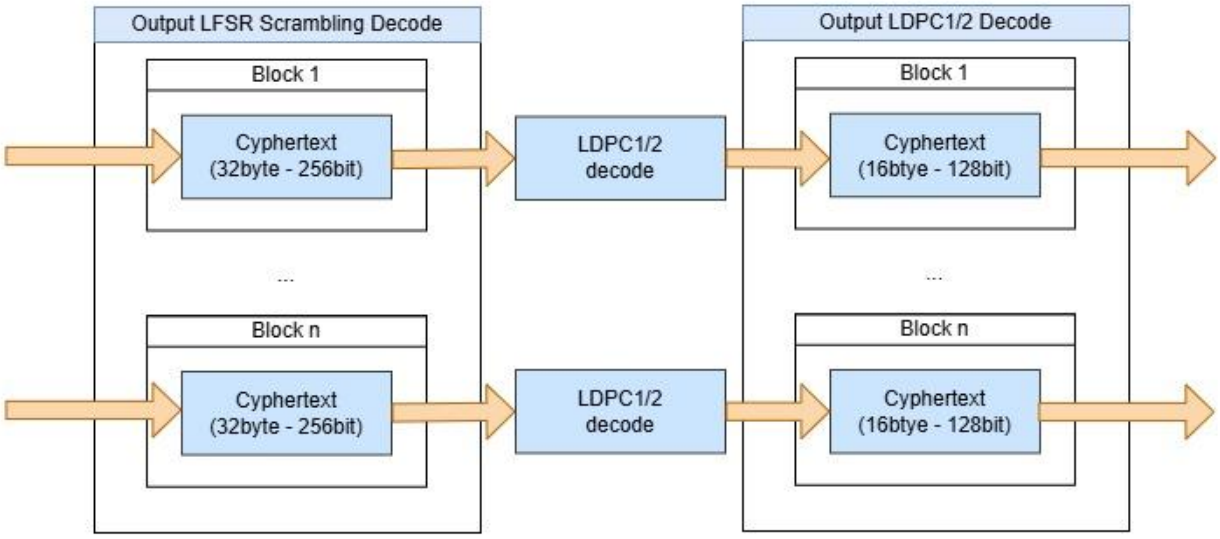
3.5.2 LFSR Scrambling Decode



Hình 16: Sơ đồ luồng giải mã LFSR Scrambling.

ID	
Mô tả	Bóc tách LFSR Scrambling mã hóa kênh.
Đầu vào	Block 32 byte đã được mã hóa bằng LFSR Scrambling.
Xử lý	1.Áp dụng thuật toán xáo trộn dựa trên thanh ghi dịch tuyến tính (LFSR). 2.Sử dụng chuỗi bit giả ngẫu nhiên (PRBS) tạo từ LFSR để thay đổi từng bit dữ liệu gốc. 3.Cập nhật thanh ghi dịch sau mỗi bit xử lý để đảm bảo tính liên tục của chuỗi PRBS.
Đầu ra	Block 32 byte dữ liệu gốc sau khi giải xáo trộn LFSR.

3.5.3 LDPC1_2 Decode

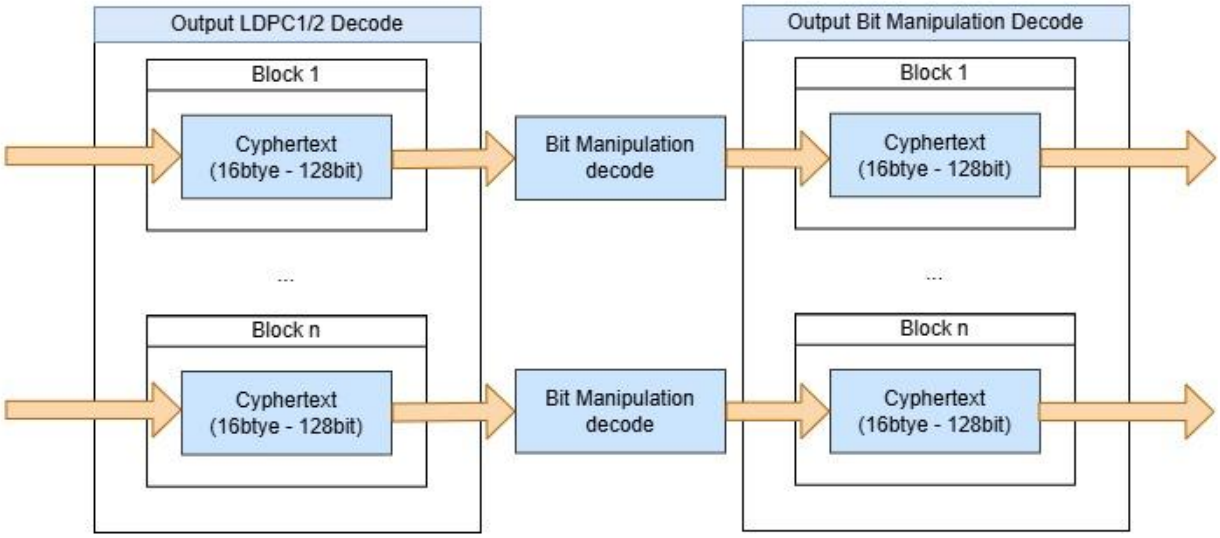


Hình 17: Sơ đồ luồng giải mã LDPC1_2.

ID	
Mô tả	Giải mã LDPC 1/2 – phát hiện và sửa lỗi bit xảy ra trong quá trình truyền qua kênh vô tuyến.
Đầu vào	Block 32 byte (256 bit) sau khi giải mã LFSR Scrambling.
Xử lý	1.Chia dữ liệu thành các đoạn 2 byte một (tương ứng với 1 byte data + 1 byte parity). 2.Với mỗi đoạn, tính **Syndrome** để kiểm tra lỗi: • Nếu Syndrome ≠ 0: tiến hành xác định vị trí bit lỗi và thực hiện lật bit tương ứng để sửa lỗi. • Lặp lại cho đến khi Syndrome = 0 (dữ liệu hợp lệ). 3.Khi tất cả các đoạn đều có Syndrome = 0, loại bỏ byte parity và giữ lại byte data.

Đầu ra	Block 16 byte (128 bit) dữ liệu đã sửa lỗi hoàn chỉnh (LDPC1/2 decode).

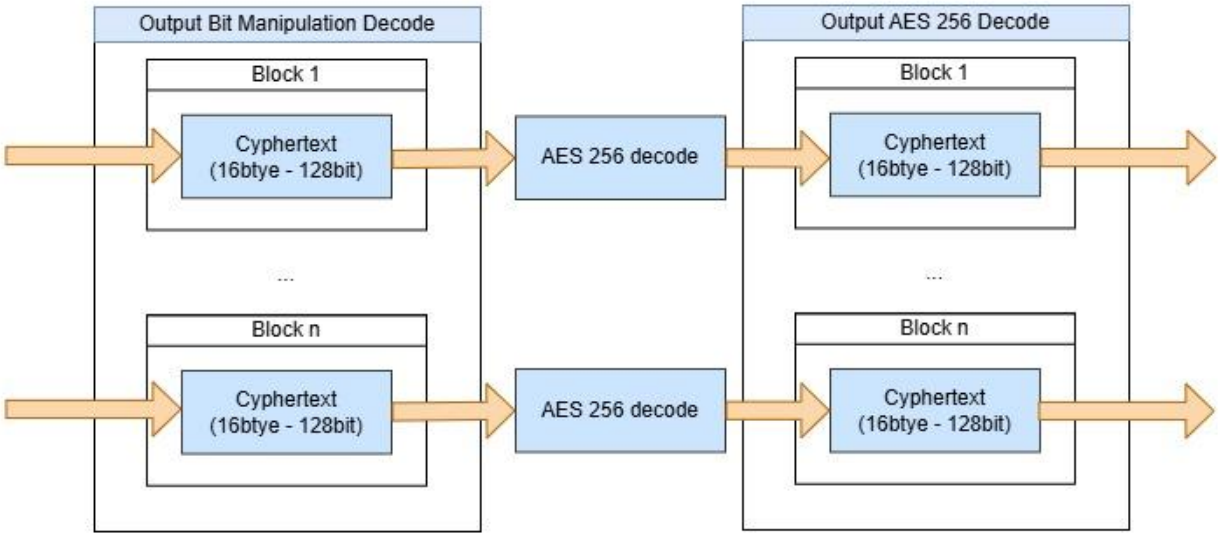
3.5.4 Bit Manipulation Decode



Hình 18: Sơ đồ luồng giải mã Bit Manipulation.

ID	
Mô tả	Giải mã thao tác bit (Bit Manipulation Decode) – khôi phục lại dữ liệu gốc từ block đã bị biến đổi bit.
Đầu vào	Block 16 byte (128 bit) đã bị thao tác bit.
Xử lý	1.Hoán đổi bit: Với toàn bộ 128 bit, thực hiện hoán đổi vị trí trong từng cặp bit liên tiếp. 2. Đảo bit (invert): Sau khi hoán đổi, đảo ngược từng bit.
Đầu ra	Block 16byte (128bit) sau khi Bit Manipulation Decode.

3.5.5 AES256 Decode



Hình 19: Sơ đồ luồng giải mã AES 256.

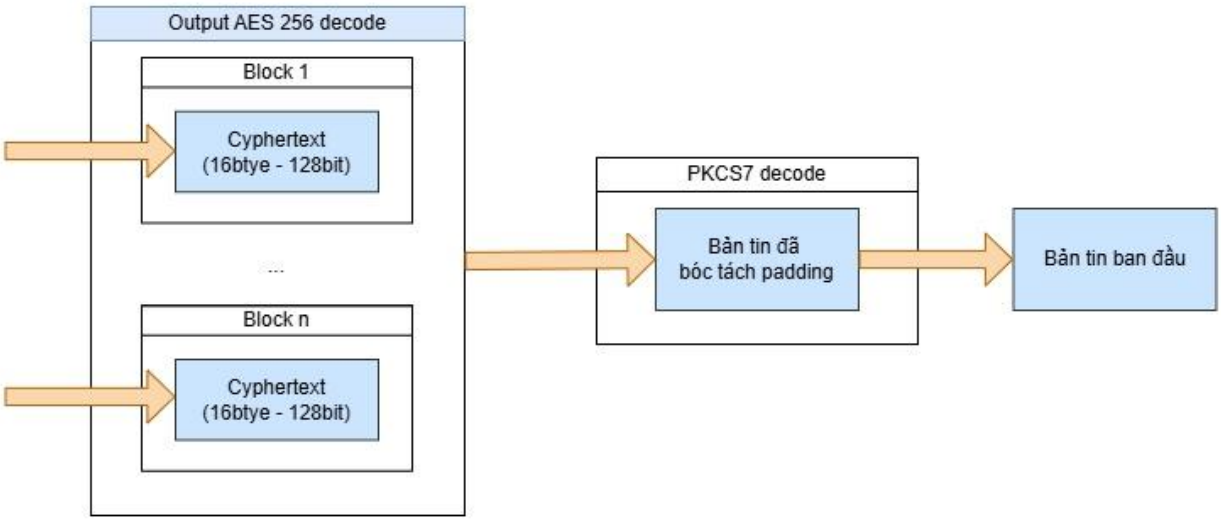
ID	
Mô tả	Giải mã bản tin mã hóa AES 256.
Đầu vào	Block 16 byte (128 bit) đã bị thao tác bit.
Xử lý	1.Tính Round_key (như Encode). 2.Giải mã từng block (15 vòng).
Đầu ra	Block 16byte (128bit) đã giải mã AES 256.

3.5.1 AES256 Decrypt (15 vòng)

ID	
Mô tả	Giải mã AES-256 với 15 vòng.
Đầu vào	Block 16byte (128bit) sau khi thao tác bit.
Xử lý	- Tách từng block 16byte từ bản tin rồi mã hóa nó 15 vòng. 1/ Bản tin plaintext XOR với 16byte cuối cùng từ Round_key. 2/ Bản tin sẽ trải qua 14 vòng mã hóa dùng: 1/ Dịch các hàng đảo (Inv Shift rows). 2/ Áp dụng phép thay thế Inv S-box. 3/ Xor bản tin với Round_key đảo. 4/ Đảo trộn các cột (Inv Mix Columns). 3/ Dịch các hàng đảo (Inv Shift rows). 4/ Áp dụng phép thay thế Inv S-box.

	5/ Tới vòng cuối cùng bản tin sau khi được mã hóa 14 lần sẽ XOR với 16byte đầu tiên từ Round_key.
Đầu ra	

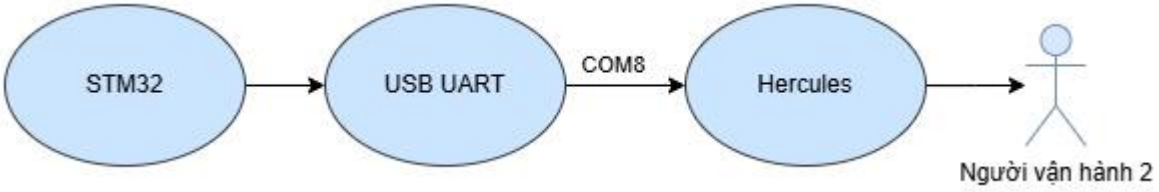
3.5.6 Data Padding (PKCS7) Decode



Hình 20: Sơ đồ luồng giải mã Data Padding PKCS7.

ID	
Mô tả	Bóc tách padding PKCS7 – loại bỏ phần đệm sau khi giải mã bản tin.
Đầu vào	Bản tin có độ dài là bội số của 16 byte, áp dụng padding theo chuẩn PKCS7.
Xử lý	1.Đọc byte cuối cùng của bản tin để xác định số byte padding. Gọi giá trị đó là P, 1 ≤ P ≤ 16. 2.Kiểm tra P byte cuối cùng có đều bằng giá trị P không (ví dụ: 05 05 05 05 05). 3.Nếu hợp lệ, loại bỏ P byte cuối cùng ra khỏi bản tin.
Đầu ra	Bản tin gốc không có padding.

3.6 Gửi bản tin hoàn chỉnh tới Hercules



Hình 21a: Sơ đồ use case STM32 (khối Thu) gửi bản tin đến Hercules.

Hình 21b: Sơ đồ luồng STM32 (khối Thu) gửi bản tin đến Hercules.

ID	
Mô tả	Giải mã bản tin mã hóa ra được bản tin gốc và gửi về Hercules qua USB UART.
Đầu vào	Bản tin đã giải mã.
Xử lý	Bản tin sau khi giải mã xong – STM32 sử dụng DMA gửi bản tin đến Hercules qua USB UART.
Đầu ra	Bản tin đến tay người vận hành 2, hiển thị trên Hercules

6. Tổng kết

Sau quá trình triển khai và kiểm thử, hệ thống Datalink truyền nhận bản tin không dây giữa hai vi điều khiển STM32 thông qua mô-đun HC-12 hiện tại đã hoạt động ổn định và đúng với mục tiêu ban đầu. Bản tin từ người vận hành 1 được mã hóa, xử lý qua chuỗi các bước như AES, Bit Manipulation, LDPC, LFSR, Frame Sync - sau đó truyền không dây và được giải mã thành công tại khối Thu đến tay người vận hành 2. Toàn bộ luồng Phát – Thu đều được theo dõi và kiểm chứng thông qua phần mềm Hercules.

Trong quá trình dự án còn tồn tại nhiều thiếu sót, cả về thiết kế thuật toán, tối ưu hiệu suất. Đây là một dự án thử nghiệm nhằm phục vụ chạy hệ thống dưới FPGA, tuy gặp nhiều khó khăn trong quá trình làm việc nhóm, tìm hiểu những thuật toán mới, môi trường (IDE), vi mạch mới nhưng đó lại chính cơ hội quý báu để nhóm rút kinh nghiệm, nâng cao kỹ năng lập trình hệ thống nhúng và xử lý dữ liệu thời gian thực. Trong tương lai, nhóm sẽ tiếp tục cải tiến hệ thống, hoàn thiện hơn về cả chức năng lẫn hiệu năng, hướng đến một hệ thống truyền thông không dây có tính ứng dụng cao hơn.