

Mạng giải mã PLC - Tập trung hoàn toàn!

www.plcjiemi.com

Vị trí hiện tại: [Trang chủ](#) >> [Chia sẻ kinh nghiệm](#) >> Phân tích giao thức PPI của Siemens

Phân tích giao thức PPI của Siemens

Xin chào mọi người: Do nghiên cứu điên cuồng về việc giải mã giao thức Siemens PPI một thời gian trước, tôi đã vô tình nghiên cứu ra giao thức Siemens S7-200 PPI thực tế hơn, và hôm nay tôi sẽ dành tặng cho mọi người. Chúng tôi thường sử dụng nó để liên lạc giữa máy tính chủ, thiết bị hiện trường và S7-200CPU, nhưng Siemens chưa công bố định dạng của giao thức PPI. Nếu người dùng muốn sử dụng giao thức PPI để giám sát, họ phải mua các sản phẩm giám sát hoặc phần mềm cấu hình của họ từ các nhà sản xuất bên thứ ba. Bạn có cần biết các công ty cấu hình trong nước như Kingview, Zijingqiao, Power Control và các công ty cấu hình khác đã chi bao nhiêu để có được thỏa thuận PPI sâu? Trên thực tế, hành vi sản phẩm và độc quyền giá cực cao của các sản phẩm điều khiển công nghiệp Siemens đã gây ra sự tẩy chay và phản đối từ những người trong ngành và cả nước ta.

Điều này mang lại những khó khăn nhất định cho quá trình tự phát triển của người dùng, đặc biệt nếu họ muốn tự phát triển bằng VB, VC và các ngôn ngữ khác, không có cách nào để kết nối với PLC, hoặc bạn phải trả cho họ rất nhiều tiền. Tôi đã tìm ra định dạng thông điệp chính của giao thức PPI thông qua việc giám sát và phân tích dữ liệu của một phần mềm giám sát công nổi tiếp.

Trên thực tế, có nhiều cách giao tiếp giữa các PLC Siemens S7-200 hoặc giữa PLC và PC: Freeport, PPI, MPI, và Profibus. Khi lập trình ở chế độ công tự do, chương trình truyền dữ liệu phải được viết trong máy tính cấp trên và PLC. Khi sử dụng giao thức PPI để truyền thông, PLC có thể được lập trình mà không cần lập trình, có thể đọc và ghi tất cả các vùng dữ liệu, rất nhanh chóng và tiện lợi. Đây là lý do tại sao chúng ta phải nghiên cứu và tìm ra nguồn gốc của giao thức PPI!

Bây giờ chúng ta hãy nói về phương pháp phân tích!

Siemens 'STEP 7 MicroWIN là một công cụ phát triển cho PLC dòng S7-200. Nó sử dụng cổng COM trên PC để kết nối với cổng lập trình PLC thông qua cáp lập trình PC / PPI. Điều này cho thấy PC thực sự có thể giao tiếp với CPU S7-200 thông qua cổng nối tiếp. Chỉ là chúng ta không biết về giao thức truyền thông. Bằng cách chặn việc gửi và nhận dữ liệu trên cổng nối tiếp của PC, và so sánh các lệnh do phần mềm Bước 7 đưa ra, có thể phân tích các thông điệp và phương thức giao tiếp của các lệnh liên quan; sau đó, gửi trực tiếp các thông báo tới PLC thông qua cổng nối tiếp để xác minh các thông báo lệnh này có đúng hay không. Với ý nghĩ này, chúng tôi sử dụng các bước sau để lấy các gói này.

Đầu tiên bạn tải phần mềm giám sát sê-ri tiếng Anh ở trên, những cư dân mạng không giỏi tiếng Anh có thể sử dụng gói tiếng Trung chúng tôi đã bản địa hóa cho bạn và thay thế file gốc, bạn phải sử dụng phần mềm này vì trước đây mình đã sử dụng rất nhiều phần mềm giám sát. Trong trường hợp dữ liệu nhiều thì xảy ra hiện tượng crash, dẫn đến mất dữ liệu rất dễ đưa ra phân tích sai cho chúng ta. Tiếp theo, trước tiên bạn mở phần mềm, tạo một cổng mới, chọn COM1, sau đó kết nối cáp lập trình PC / PPI với COM1, để các thông báo do Step7 Micro / Win gửi đến PLC có thể hiển thị hoàn toàn trên phần mềm giám sát. trước mặt bạn. Chúng tôi thiết lập các thông số cổng nối tiếp theo hướng dẫn sử dụng hệ thống S7-200: 9600, 8, E chặn l, 1 bit dừng. Sau đó thiết lập phần mềm Step7 để nó có thể giao tiếp bình thường với CPU S7-200. Gửi một lệnh rõ ràng từ phần mềm Step7 và phần mềm giám sát có thể hiển thị thông báo này (được hiển thị dưới dạng hệ thập lục phân, chỉ một số phiên bản có thể được nhìn thấy trong mã ASCII và những số khác là vô nghĩa).

Chiến lược bề khóa của chúng tôi là phân tích giao thức truyền thông PPI vốn có bên trong PLC thông qua phương pháp giám sát phần mềm, sau đó máy tính chủ sử dụng lập trình VB, tuân theo giao thức truyền thông PPI, đọc và ghi dữ liệu PLC và thực hiện các tác vụ vận hành của người-máy. So với giao thức truyền thông miễn phí thông thường, phương thức truyền thông này bỏ qua phần viết chương trình truyền thông PLC, và chỉ cần ghi tài nguyên chương trình truyền thông của máy tính phía trên. Lớp vật lý của cổng lập trình của S7-200 là cấu trúc RS-485. SIEMENS cung cấp phần mềm MicroWin, sử dụng giao thức PPI (Point to Point). Để chuyển đổi cổng nối tiếp 232 sang 485, bạn có thể sử dụng cáp PPI tự chế tạo bởi trang web của chúng tôi, hiệu ứng tốt hơn Oh! [Vui lòng nhấp để tải xuống!](#) Hoặc tự mình làm điều đó, mặc đẹp!

Bạn không thể chỉ nói và thực hành! Bây giờ chúng ta hãy nói về cách PLC Siemens giao tiếp.

PC giao tiếp với PLC ở chế độ chủ-tớ. PC gửi lệnh đọc và ghi ở định dạng sau, PLC nhận được phản hồi chính xác (trả về dữ liệu phản hồi E5H hoặc F9H, xem phân tích bên dưới) và máy tính chủ nhận được phản hồi này và gửi lệnh xác nhận (10 02 00 5C 5E 16), PLC trả dữ liệu tương ứng về máy tính phía trên. Nói chung, nếu máy tính chủ muốn kết nối với PLC thì trước tiên nó phải gửi dữ liệu phân trang sau 10 02 00 49 4B 16 Các bạn ơi! Tất cả chúng ta đều là những động vật tiên tiến có máu, có thịt, có suy nghĩ và có cảm hứng. Làm sao bạn nhớ được quá nhiều số máy nhàm chán, vô vị, phức tạp và khó hiểu? Dù sao thì tôi cũng không nhớ được! (^ _ ^ Bất đầu tây nào) Lúc này, bạn có thể nhắm mắt lại và yên tĩnh, yên tĩnh, yên tĩnh trở lại. Hãy liên tưởng đến chế độ gọi máy bộ đàm dã chiến thời chiến, thì lệnh phân trang ban đầu (10 02 00 49 4B 16) này có thể hiểu là: "Đồng Hai HAI (02), tôi là Tùng Tùng (00), nghe xin hãy trả lời, nghe Vui lòng trả lời! hết! " .

Bây giờ chúng ta hãy phân tích ngắn gọn ý nghĩa cụ thể của lệnh này: 10 Biểu tượng bắt đầu, có nghĩa là bắt đầu nói bằng một tiếng ho. 02 là số trạm địa chỉ của PLC máy tính phía dưới sẽ được máy tính phía trên liên lạc, là người cần tìm. 00 là số trạm của chính máy tính chủ. Lệnh phân trang 49, ý nghĩa của lệnh gọi tìm. 16 terminator, nghĩa là hết, hết, hết. Trong số đó, 4B là mã kiểm tra, được thiết kế để tránh lỗi truyền dữ liệu. Nó được lấy theo cách này: hai chữ số cuối của tổng 02 + 00 + 49 là mã kiểm tra, được gọi là kiểm tra hoặc kiểm tra tổng Còn được gọi là kiểm tra phần dư, vì nó lấy phần còn lại sau khi chia cho 100. Khi máy tính tính toán theo hệ thập lục phân, số có được theo công

thức (02 + 00 + 49) mod 100 là mã kiểm tra. Bạn tính xem nó có bằng 4B không! Điều này đúng cho tất cả các kiểm tra giao thức PPI khác. Nếu PLC của trạm số 02 nhận được tín hiệu phân trang, nó sẽ trả lời: 10 00 02 00 02 16 có nghĩa là: "đã nhận được báo cáo về Dongdong (00), Dongliang (02) đã nhận được, xin hướng dẫn, hết!" có thể hiểu được! Bạn có một lời giải thích tốt hơn? Bây giờ bạn đã tìm thấy người bạn cần tìm, PC tiếp theo là chỉ huy! Bạn có thể phát hành đơn đặt hàng. Lúc này máy tính chủ đưa ra một lệnh sẽ được giải thích cụ thể bên dưới, sau khi lệnh được đưa ra, nếu PLC nhận đúng lệnh sẽ trả về ký tự E5, có nghĩa là: "Đã hiểu!". Thực ra PLC chỉ nói là hiểu, đã hiểu lệnh của PC máy tính thượng, còn không thực hiện lệnh thì làm sao thực hiện được lệnh? Nó chỉ được thực thi sau khi PC của máy tính phía trên đưa ra lệnh xác nhận. Lúc này, máy tính phía trên sẽ đưa ra lệnh xác nhận (10 02 00 5C 5E 16), trong đó 5C là lệnh thực hiện, có nghĩa là: "Hãy thực hiện ngay lập tức, hết!". Sau đó PLC thực hiện công việc của nó! Hóa ra PLC không dễ, chẳng trách nó được gọi là máy tính hạ đẳng! Đó là ý của người tiếp theo!

Bạn đang nói về nhiều hỗn loạn? Mục đích là để phân loại mối quan hệ giữa cấp trên và cấp dưới, mối quan hệ giữa chủ và cấp dưới, và thứ tự các chỉ thị, và sử dụng một phương pháp ghi nhớ tốt để nhớ mã máy nhàm chán.

Dưới đây chúng tôi liệt kê và phân tích hướng dẫn đọc mật khẩu PLC: 68 1B 1B 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 08 00 00 03 00 05 E0 D2 16

Đọc phân tích lệnh: đọc từng dữ liệu một

1. Bắt đầu dấu phân cách (68H)
2. Độ dài dữ liệu thông báo
3. Độ dài dữ liệu lặp lại
4. Bắt đầu phân tách (68H)
5. Địa chỉ đích từ xa, tham chiếu đến giá trị của địa chỉ. là địa chỉ PLC
6. Địa chỉ cục bộ, tham chiếu đến con trỏ của địa chỉ này, là địa chỉ riêng của máy tính chủ
7. Mã chức năng, 5CH là bộ kích hoạt chu kỳ thay thế, 6CH là bộ kích hoạt chu kỳ thông tin đầu tiên và 7CH là bộ kích hoạt chu kỳ thay thế .
- 8-17 Điểm truy cập dịch vụ đích
- 18-22 Điểm truy cập dịch vụ nguồn Phân tích 18-bit: 01: Sắp xếp bit 02: Sắp xếp byte 04: Sắp xếp từ 06: Sắp xếp từ kép
- 23-31. Đơn vị dữ liệu
32. Mã kiểm tra
33. Dấu phân cách cuối (16H)

Độ dài dữ liệu và độ dài dữ liệu lặp lại của thông báo là độ dài dữ liệu từ DA đến DU và mã kiểm tra là tổng kiểm tra dữ liệu từ DA đến DU và chỉ giá trị byte cuối cùng được lấy cho việc này Kiểm tra. Phương pháp tính toán của mã xác minh giống như được mô tả ở trên. Trong biến dữ liệu đọc và ghi PLC, mã chức năng đọc dữ liệu là 6CH và mã chức năng ghi dữ liệu là 7CH. Đó đọc một dữ liệu tại một thời điểm, các lệnh đọc đều là 33 byte. 1-22 byte đầu tiên giống nhau, đối với

1	2	3	4	5	6	7	số 8	9	10	11	12	13	14	15	16	17	18	19	20	hai mươi một	hai mươi hai
người bắt đầu	chiều dài		người bắt đầu	Từ xa	địa phương	mã chức năng			Số liên lạc				chiều dài tham số				04 đọc 05 ghi	Sự sắp xếp			
68	1B	1B	68	02	00	6C	32	01	00	00	00	88	00	0E	00	00	04	01	12	0A	10

Lệnh đọc mật khẩu PLC: 68 1B 1B 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 08 00 00 03 00 05 E0 D2 16

hai mươi ba	hai mươi bốn	25	26	27	28	29	30	31	32	33
độ dài đọc		số lượng dữ liệu		loại bộ nhớ		Bù lại		kiểm tra mã		Kẻ hủy diệt
02	00	08	00	00	03	00	05	E0	D2	16

Vì là lệnh đọc dữ liệu PLC từ PC nên SA = 00, DA = 02, nếu có nhiều trạm thì DA phải đổi thành số trạm tương ứng. Độ dài từ DA đến DU trong lệnh đọc là 1B hoặc 27 byte. Bắt đầu từ 23 byte, nó thay đổi tùy theo loại và vị trí của dữ liệu đọc. Bảng trên là Byte23-33 các lệnh bộ nhớ khác nhau để đọc.

byte	hai mươi ba	hai mươi bốn	25	26	27	28	29	30	31	32	33
Hàm số	độ dài đọc		số lượng dữ liệu		loại bộ nhớ		Bù lại			kiểm tra mã	Kẻ hủy diệt
Đọc Q0.1	01	00	01	00	00	82	00	00	00	64	16
Đọc M0.0	01	00	01	00	00	83	00	00	00	65	16
Đọc M0.1	01	00	01	00	00	83	00	00	01	66	16
Đọc SMB34	02	00	01	00	00	05	00	00	01	F9	16
Đọc VB100	02	00	01	00	01	84	00	03	20	8B	16
Đọc VW100	04	00	01	00	01	84	00	03	20	8D	16
Đọc VD100	06	00	01	00	01	84	00	03	20	8F	16
Đọc I0.5	01	00	01	00	00	81	00	00	05	68	16
Đọc I0.7	01	00	01	00	00	81	00	00	07	6A	16

Byte23-33 của lệnh read trong bảng trên, ta có thể rút ra kết quả sau từ bảng:
Byte 23 read data length
01: 1 Bit 02: 1 Byte
04: 1 Word 06: Double Word
Byte 25 data number, here is 01, xem hướng dẫn bên dưới khi đọc nhiều dữ liệu cùng một lúc.
Loại bộ nhớ Byte 27, bộ nhớ 01: V 00: loại
bộ nhớ Byte 28 khác
04: S 05: SM 06: AI 07: AQ 1E: C
81: I 82: Q 83: M 84: V 1F: T
Byte 29,30 , Con trỏ bù đắp bộ nhớ 31 (địa chỉ bộ nhớ * 8), chẳng hạn như: VB100, địa chỉ bộ nhớ là 100, con trỏ bù đắp là 800, được chuyển đổi sang hệ thập lục phân là 320H, thì ba byte của Byte 29-31 là: 00 03 20.
Byte 32 tổng kiểm tra, là từ (DA + SA + DSAP + SSAP + DU) Mod 256 như đã đề cập trước đó.
Đọc nhiều phần
dữ liệu cùng một lúc Đối với trường hợp đọc nhiều phần dữ liệu cùng một lúc, 21Byte đầu tiên tương tự như trên, nhưng độ dài LD, LDr và Byte 15 khác nhau:
Byte 15 byte chiếm dụng khối dữ liệu, cho biết số byte được khối dữ liệu chiếm giữ. Nó liên quan đến số lượng khối dữ liệu, chiều dài = 4 + số khối dữ liệu * 10, chẳng hạn như: 4 + 10 = 0E (H) cho một phần dữ liệu; 4 + 3 * khi đọc ba khối dữ liệu khác nhau M, V, Q đồng thời 10 = 22 (H).
Byte 23 luôn là 02 tức là Byte.
Byte 25 Số byte được đọc liên tiếp theo byte. Ví dụ: nếu 2 VD được đọc, Byte25 = 8
Byte 19 --- 30 được liệt kê theo định dạng ở trên là đọc một dữ liệu tại một thời điểm,
Byte 31 --- 42 Một loại dữ liệu khác cũng được đưa ra theo định dạng trên.
Và như vậy, đọc tối đa 222 byte dữ liệu cùng một lúc.

Viết phân tích lệnh:
Đề ghi dữ liệu kiểu Double Word tại một thời điểm, lệnh write là 40 byte và phần còn lại là 38 byte. Viết kiểu dữ liệu Double Word, 1-22 byte đầu tiên là:

1	2	3	4	5	6	7	số 8	9	10	11	12	13	14	15	16	17	18	19	20	hai mươi một	hai mươi hai
người bắt đầu	chiều dài		người bắt đầu	Từ xa	địa phương	mã chức năng															
68	hai mươi một	hai mươi một	68	02	00	7C	32	01	00	00	00	00	00	00	0E	00	00	04	01	12	0A 10

68 23 23 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10
Viết kiểu dữ liệu khác, các byte đầu tiên từ 0-21 là: (so với ở trên, chỉ có độ dài byte thay đổi)
68 21 21 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10

hai mươi ba	hai mươi bốn	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Độ dài dữ liệu		số lượng dữ liệu		Loại lưu trữ		Bù lại		Mẫu dữ liệu		bit dữ liệu		ghi giá trị		kiểm tra mã	Kẻ hủy diệt
01	00	01	00	00	82	00	00	00	00	03	00	01	01	79	16

Thay đổi từ 22 byte tùy thuộc vào giá trị và vị trí của dữ liệu được ghi. Bảng trên là Byte22-40 của một số lệnh ghi.
Byte 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
Ghi vị trí và độ dài giá trị Loại giá trị bit bù, mã kiểm tra, dấu kết thúc
M0.0 = 1 01 00 01 00 00 82 00 00 00 00 03 00 01 01 00 71 16
M0.0 = 0 01 00 01 00 00 83 00 00 00 00 03 00 01 00 00 70 16
M0.1 = 1 01 00 01 00 00 83 00 00 03 00 01 01 00 72 16
vb100 = 10 02 00 01 00 01 84 00 03 20 00 04 00 08 10 00 AE 16
vb100 = FF 02 00 01 00 VW11 84 00 03 20
000 04 00 FFFF 04 00 01 00 01 84 00 03 20 00 04 00 10 FF FF A6 16
VD100 = FFFFFFFF 06 00 01 00 01 84 00 03 20 00 04 00 20 FF FF FF FF B8 1

Byte22 của lệnh ghi — Cuối cùng, chúng ta có thể rút ra các kết quả sau sau khi phân tích:
Byte 23 - Byte 31 Độ dài, loại bộ nhớ và độ lệch bộ nhớ của dữ liệu ghi cũng giống như lệnh đọc. T, C, v.v. không thể được viết bằng lệnh write.
Byte 33 Nếu dữ liệu bit được ghi, byte này là 03, ngược lại là 04
Byte 35 Số bit của dữ liệu được ghi
01: 1 Bit 08: 1 Byte 10H: 1 Word 20H: 1 Double Word
Byte 36--41 Giá trị, mã kiểm tra, dấu chấm dứt
Nếu dữ liệu được viết bằng bit và byte, Byte35 là giá trị được ghi, Byte36 = 00, Byte37 = mã kiểm tra, Byte38 = 16H, kết thúc. Nếu những gì được viết là dữ liệu từ (byte kép), Byte35 và Byte36 là các giá trị được viết, Byte37 = mã kiểm tra, Byte38 = 16H, kết thúc. Nếu những gì được viết là dữ liệu từ kép (bốn byte), Byte35-38 là giá trị được viết, Byte39 = mã kiểm tra, Byte40 = 16H, kết thúc.

Sau khi đọc phần phân tích lệnh trên, bây giờ chúng ta sẽ đưa ra một số ví dụ về các giao thức PPI thường được sử dụng để phân tích:

PC phản trang: 10 02 00 49 4B 16

PLC return: 10 00 02 02 04 16

PC send: 10 02 00 5C 5E 16

PLC return : E5

Hãy xem lệnh đọc mật khẩu của Siemens S7-200PLC:

vui lòng sử dụng phần mềm công nối tiếp để gửi ở hệ thập lục phân, cổng được đặt thành 9600; e; 8; 1

gửi: 68 1B 1B 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 08 00 00 03 00 05 E0 D2 16 Ý nghĩa: 8 ký tự bắt đầu từ bit 05E0 của vùng lưu trữ hệ thống (vùng 03) được yêu cầu để truyền (nghĩa là , 8 giá trị mật khẩu).

Nếu giao tiếp đúng thì PLC sẽ trả về E5, nghĩa là: đã nhận

rồi thì máy tính chủ sẽ gửi lại lệnh thực thi xác nhận 10 02 00 5C 5E 16 nghĩa là: vui lòng thực hiện lệnh. (Nói đến đây thì dừng ở đây. Sau khi PLC trả về lệnh E5, PC máy tính phía trên cần gửi lệnh xác nhận trong thời gian rất ngắn. Nếu quá muộn, lệnh vừa rồi sẽ không có giá trị. có thể đo thời gian cụ thể. Dù sao cũng mất 1 hoặc 2 giây. Không có vấn đề gì. Đây cũng là lý do tại sao nhiều cư dân mạng hỏi tôi rằng giao tiếp không thành công.) Khi đó PLC sẽ thực sự thực hiện lệnh một cách ngoan ngoãn và trả về như sau ký tự: 68 1D 1D 68 00 02 08 32 03 00 00 00 00 02 00 0C 00 00 04 01 FF 04 00 40 9B 98 02 06 9D 9A 00 76 7D 16

Thời thì dừng ở đây, xem mật khẩu là gì nhé! Nếu bạn thực sự hiểu về giao thức PPI thì không khó để tìm ra mật khẩu, nhưng mật khẩu này được mã hóa hai lần, không phải là mật khẩu thật và cần được giải mã, về thuật toán mật khẩu, tôi không tiện tiết lộ ở đây , nhưng bạn phải thực hiện nhiều thí nghiệm hơn. Có thể thu được kết quả.

Hãy xem một lệnh để đọc số phiên bản PLC:

điều đầu tiên chúng ta cần xác định trong quá trình giải mã là số phiên bản PLC. Chỉ để xem đó là phiên bản cũ hay phiên bản 02, hoặc thực hiện một chương trình mã hóa và giải mã. Mã nguồn giao tiếp của anh ta như sau:

68 1B 1B 68 02 00 7C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 14 00 00 03 00 00 00 09 16

PLC trả về E5

sau khi gửi dữ liệu trên. Gửi lệnh xác nhận: 10 02 00 5C 5E 16

Tại thời điểm này, số phiên bản của plc được trả về. Xem bên dưới:

68 29 29 68 00 02 08 32 03 00 00 00 00 02 00 18 00 00 04 01 FF 04 00 A0 43 50 55 20 32 32 36 20 13 4E 20 20 20 30 31 D 20

Nhìn vào đoạn này: 43 50 55 20 32 32 36 20 43 4E 20 20 20 20 20 30 32 30 31 là mã ASCII của số phiên bản plc. Nếu hiển thị ở chế độ ASC, bạn sẽ thấy dữ liệu trên rõ ràng hơn: CPU SP 2 2 6 SP CN 0 2 0 1 (sp là khoảng trắng) 0201 là số phiên bản.

Cách khác là đọc lệnh mật khẩu TD200:

68 1B 1B 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 02 00 01 84 00 00 50 B9 16 (VW10)

gửi lệnh M0:

68 20 68 02 00 7C 32 01 00 00 00 00 00 0E 00 05 05 01 12 0A 10 01 00 01 00 00 83 00 00 00 00 03 00 01 01 80 16

Đọc dữ liệu vùng 222-bit 3 (vùng hệ thống):

68 1B 68 02 00 6C 32 01 00 00 00 00 00 0E 00 00 04 01 12 0A 10 02 00 DE 00 00 03 00 00 00 C3 16

Đọc lệnh Bit bảo vệ mật khẩu:

68 1B 1B 68 02 00 6C 32 01 00 00 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 03 00 05 D8 C3 16

Viết lại lệnh bit bảo vệ mật khẩu: (bạn có thể xác minh xem nó có khả thi không)




68 20 20 68 02 00 7C 32 01 00 00 00 00 0E 00 05 05 01 12 0A 10 08 00 01 00 00 03 00 05 D8 00 04 00 08 04 EF 16

68 20 20 68 02 00 7C 32 01 00 00 00 00 0E 00 05 05 01 12 0A 10 02 00 01 00 00 03 00 05 D8 00 03 00 08 04 E8 16

Xóa tất cả các lệnh: 0

2 6 8 21 C 32 07 00 00 00 24 00 08 00 0C 00 01 12 04 11 45 01 00 FF 09 00 08 16 19 06 0D 01 08 18 1E EE 16

Về thỏa thuận PPI này, tôi cũng có một số sai sót trong quá trình nghiên cứu sơ bộ, mong các cư dân mạng có những hiểu biết và khám phá tốt hơn để đăng trên diễn đàn của chúng tôi, hoặc để lại lời nhắn cho tôi trên QQ!

 ID WeChat: plcjiemi  Tel: 13969908936  E-Mail: plcjiemi@qq.com 