Có, việc phục hồi giao diện JTAG/SWD trên nền tảng được bảo vệ luôn là chủ đề nhạy cảm trong bảo mật nhúng .

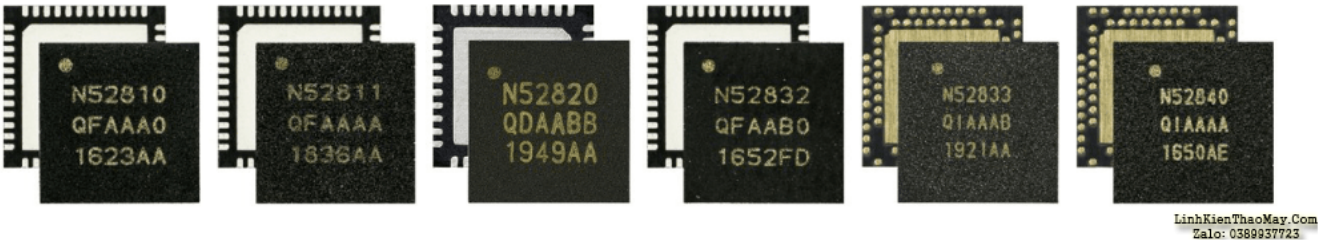
Cuộc điều tra bảo mật này trình bày một cách để vượt qua APPROTECT trên nRF52840 được bảo vệ, để kích hoạt lại Giao diện gỡ lỗi dây nối tiếp (SWD), cung cấp khả năng gỡ lỗi đầy đủ trên mục tiêu (truy cập R/W vào Flash/RAM/Đăng ký, Code Exec và lập trình lại). Tất cả các phiên bản nRF52 đều bị ảnh hưởng.

Do đặc điểm nội tại của nó, lỗ hổng này không thể được vá nếu không thiết kế lại Silicon, dẫn đến vô số thiết bị dễ bị tấn công tồn tại mãi mãi.

Nordic Semiconductor và LimitedResults không đồng ý về việc tiết lộ thông tin một cách có trách nhiệm.

Giới thiệu về dòng SoC dòng nRF52

Toàn bộ dòng nRF52 bao gồm sáu nền tảng nRF52 khác nhau, tất cả đều được xây dựng xung quanh CPU ARM Cortex-M4F.



Toàn bộ dòng SoC nRF52 của Bắc Âu

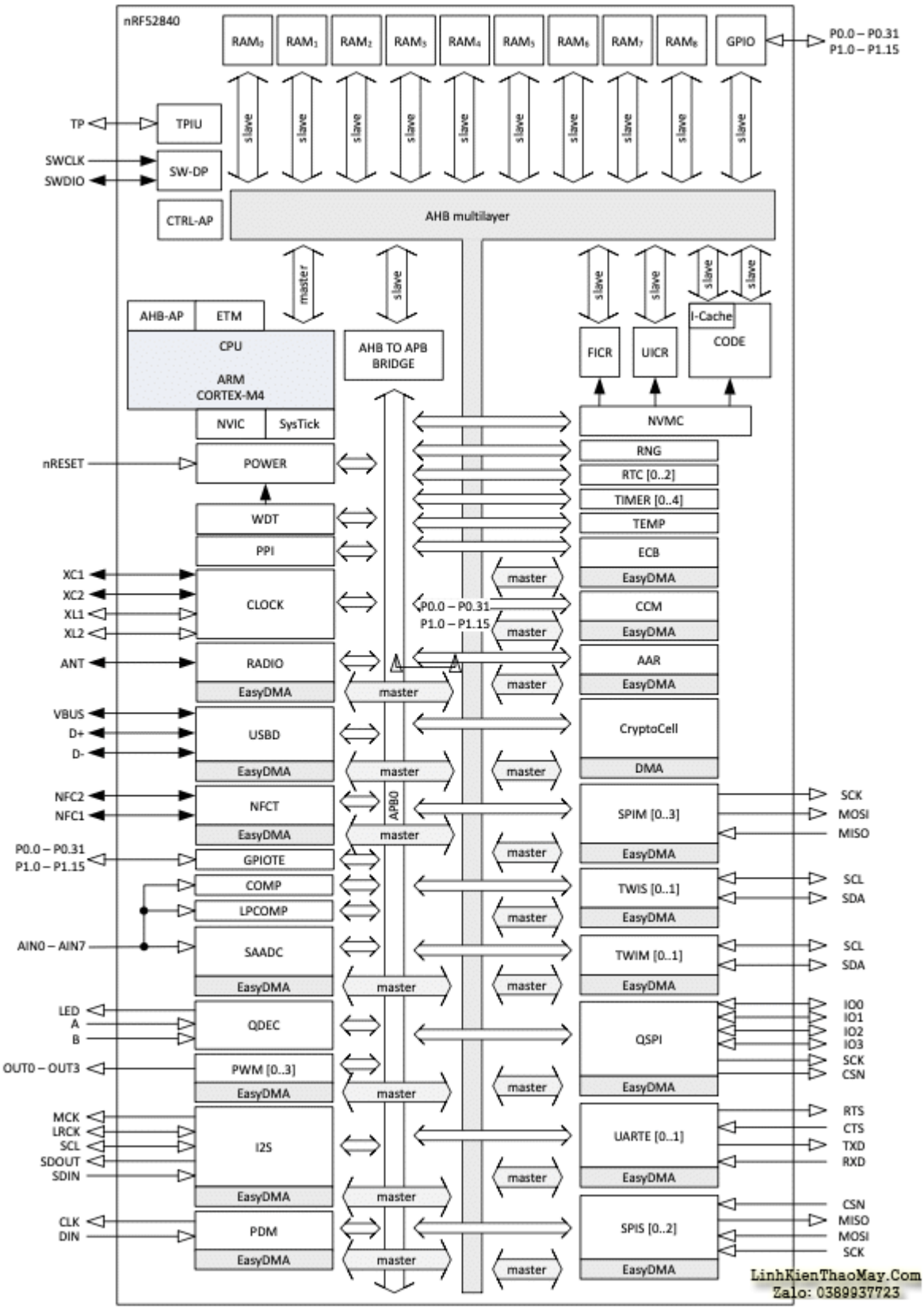
Một bảng so sánh giữa các nền tảng khác nhau có sẵn **ở đây** .

Hãy tập trung vào sản phẩm chủ lực của gia đình, nRF52840.

Mục tiêu: nRF52840

Hệ thống trên chip (SoC) nRF52840 là thành viên tiên tiến nhất trong dòng SoC dòng nRF52. Đây là SoC đa giao thức Bluetooth, Thread và Zigbee tiên tiến được xây dựng dựa trên CPU Cortex-M4F 64 MHz. Nó đi kèm với 1MB Flash và 256Kb RAM, cả hai đều được tích hợp đầy đủ.

Về mặt bảo mật, đơn vị mật mã ARM TrustZone CryptoCell được tích hợp dưới dạng công cụ mã hóa. Sơ đồ khối đưa ra ý tưởng về khả năng (và độ phức tạp) của nRF52840:



Sơ đồ khối của Nordic nRF52840

Bảo vệ đọc mã

Ngày nay, hầu hết các Nhà cung cấp bóng bán dẫn đều triển khai tính năng bảo mật có tên là **Bảo vệ đọc mã** bên trong MCU/SoC của họ, ngăn chặn kẻ tấn công có quyền truy cập vật lý để loại bỏ Mã/Dữ liệu được lưu trữ trong Bộ nhớ Flash tích hợp và cũng để sửa đổi nó.

Tính năng bảo mật này có thể có tên gọi khác nhau tùy theo nhà cung cấp chip (CRP, RDP, PCROP...)

Nói chung, khi kẻ tấn công có bản sao Phần sụn, hắn có thể bắt đầu quy trình kỹ thuật đảo ngược hoặc chỉ lấy một số dữ liệu nhạy cảm (chẳng hạn như khóa và mật khẩu).

Trước đây trên dòng nRF51 s

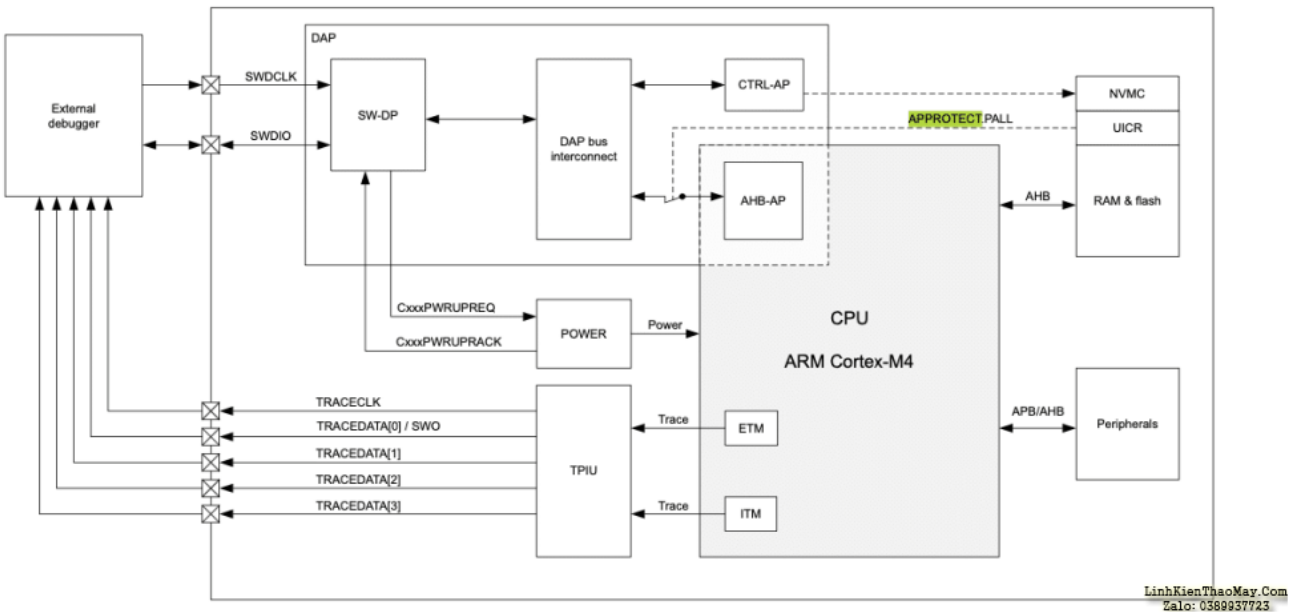
Trở lại năm 2015, **Bao gồm bảo mật** đã tìm thấy một lỗ hổng trong thiết kế tính năng bảo vệ đọc mã nRF51 (được gọi là **RBPCONF**), cho phép kẻ tấn công khôi phục toàn bộ Phần sụn bằng giao diện gỡ lỗi SWD.

Trên nRF51, khi tính năng bảo vệ RBPCONF được bật, việc sử dụng trình gỡ lỗi để truy cập trực tiếp vào Flash hoặc RAM sẽ không trả về gì ngoài số 0. Tuy nhiên, vẫn có thể kiểm soát việc thực thi mã cũng như đọc và ghi vào các thanh ghi (thậm chí vào Bộ đếm chương trình). Vì vậy, việc tìm kiếm một "tiện ích" trong bộ nhớ được bảo vệ, chẳng hạn như một lệnh tải từ đơn giản để đọc bộ nhớ từ một địa chỉ trong thanh ghi này sang thanh ghi khác là đủ để trích xuất toàn bộ Phần sụn. Do đó, kẻ tấn công có quyền truy cập vật lý vào thiết bị dựa trên nRF51 có thể hủy Flash, khai thác lỗ hổng thiết kế này.

Qua đó, tính năng bảo vệ đọc mã nRF51 (RBPCONF) được coi là cơ chế bảo mật bị hư và không bảo vệ được việc trích xuất Firmware.

Bảo vệ cổng truy cập ĐƯỢC PHÊ DUYỆT trên nRF52

Sau thất bại này, **NordicSemiconductor** đã quyết định thiết kế nRF52 với cơ chế bảo mật hạn chế hơn để bảo vệ chống lại việc đọc bộ nhớ. Tính năng bảo mật này được gọi là **Access Port Protection (APPROTECT)** , được đề cập rất ngắn gọn trong Datasheet nRF52:



Tổng quan về nRF52

Thông thường, trình gỡ lỗi bên ngoài đang truy cập CPU ARM thông qua Cổng truy cập gỡ

lỗi (DAP). Sau đó, có hai cổng khác nhau:

– **CTRL-AP.** Đây là Cổng gỡ lỗi chính và nó không phụ thuộc vào APPROTECT. Nó sẽ được sử dụng để khôi phục, ví dụ như trong trường hợp thiết bị bị brick.

– **AHB-AP.** Đây là Cổng gỡ lỗi thực sự để truy cập bộ nhớ và điều khiển CPU thông qua SWD.

NordicSemiconductor không cung cấp các thông tin nào về cơ chế PHÊ DUYỆT. mình khuyên bạn nên đọc **tài liệu tham khảo ARM của Cortex-M** .

UICR

Thanh ghi cấu hình thông tin người dùng (UICR) là các thanh ghi bộ nhớ không ổn định (NVM) để định cấu hình cài đặt dành riêng cho người dùng. Chúng được ánh xạ tại **0x10001000** .

PHÊ DUYỆT của UICR. **Cần phải viết 0xFFFFFF00 tại địa chỉ 0x10001208** để kích hoạt Bảo vệ cổng truy cập, như hiển thị bên dưới:

4.5.1.5 APPROTECT

Address offset: 0x208

Access port protection

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																												A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ID	Acce	Field	Value ID	Value								Description																							
A	RW	PALL										Enable or disable access port protection.																							
												See Debug and trace on page 50 for more information.																							
			Disabled	0xFF								Disable																							
			Enabled	0x00								Enable																							

APPROTECT Đăng ký cấu hình

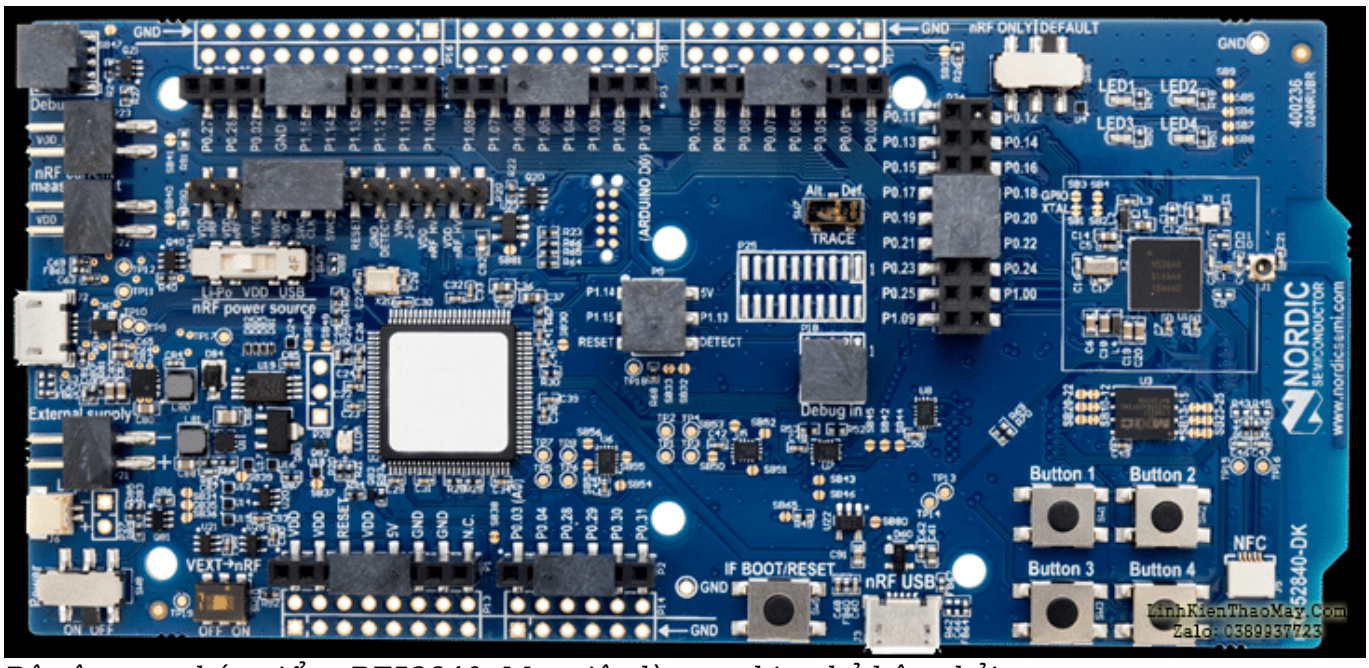
Sau đó, cần phải bật nguồn để áp dụng các cài đặt mới có trong UICR.

Theo Nordic, không thể tắt bảo mật APPROTECT nếu không xóa toàn bộ RAM và Memory Flash.

Thiết lập môi trường

Bộ công cụ phát triển nRF52840

Bộ công cụ phát triển **nRF52840-DK** được sử dụng cho các thử, dựa trên **nRF52840** .



Bộ công cụ phát triển nRF52840. Mục tiêu là con chip nhỏ bên phải.

Hướng dẫn sử dụng và sơ đồ có sẵn trên trang web của nhà sản xuất. NRF52840 là phiên bản QFN73, 7x7mm, nằm ở bên phải.

Bộ công cụ này có **trình gỡ lỗi Segger J-Link** được nhúng (con chip lớn nhất có nhãn dán màu trắng).

SDK là phiên bản mới nhất hiện có **nRF5_SDK_16.0.0_98a08e2** .

Nrfjprog v10.6.0 và JLink v6.64 được sử dụng làm công cụ dòng lệnh.

Để bắt đầu nhanh, mình chỉ biên dịch và flash một ví dụ từ SDK để kiểm tra xem môi trường của mình có ổn không. Mọi thứ diễn ra suôn sẻ, mình có thể tập trung vào tính năng bảo mật APPROTECT.

Kích hoạt APPROTECT

Phương pháp 1 (thông qua OpenOCD):

```
$ openocd -s /usr/local/share/openocd/scripts -f ./interface/jlink.cfg
-c "transport select swd" -f ./target/nrf52.cfg
#Telnet
$ telnet localhost 4444
> flash fillw 0x10001208 0xFFFFFF00 0x01
> reset
```

Phương pháp 2 (thông qua nrfjprog):

```
$ nrfjprog --memwr 0x10001208 -- val 0xFFFFFF00
```

Sau đó, mọi nỗ lực kết nối CPU sẽ không thành công, hiển thị " **Lỗi: Không thể tìm thấy**

MEM-AP để điều khiển lõi” :

```
xPack OpenOCD, 64-bit Open On-Chip Debugger 0.10.0+dev (2019-07-17-11:25)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
swd
Info : J-Link OB-SAM3U128-V2-NordicSemi compiled Jan 21 2020 17:30:48
Info : Hardware version: 1.00
Info : VTarget = 3.300 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x2ba01477
Error: Could not find MEM-AP to control the core
Info : Listening on port 3333 for gdb connections
Error: Target not examined yet
```


SWD bị vô hiệu hóa do kích hoạt APPROTECT.

Mọi nỗ lực đính kèm GDB đều hiển thị lỗi:

```
Error: attempted 'gdb' connection rejected
```

Vô hiệu hóa APPROTECT

Phương pháp 1 (thông qua OpenOCD):

```
# Read APPROTECTSTATUS
# (0x0 Access port protection enabled - 0x1 APP disabled)
> nrf52.dap apreg 1 0x0c 0x00000000
# Write ERASEALL register
> nrf52.dap apreg 1 0x04 0x01
> reset
```

Phương pháp 2 (thông qua nrfjprog):

```
$ nrfjprog -f NRF52 --recover
```

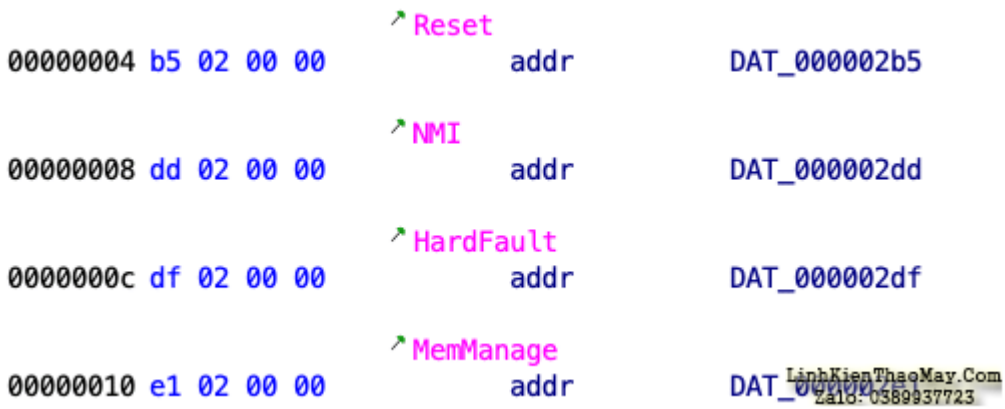
Tất cả Flash và RAM sẽ bị xóa trong quá trình này. Sau khi thiết lập lại, nRF52 có thể được lập trình lại.

Chiến lược tấn công

Quá trình khởi động

Quá trình khởi động tương đối đơn giản, do thực tế là nRF52840 không chứa mã bootROM (do đó không có các quy trình bộ tải khởi động được nhúng cũng như các quy trình IAP/ISP để thiết kế ngược...)

Điểm vào là 0x000002B4, nằm trong bộ nhớ Flash:



mình đã sử dụng Ghidra…thời buổi kinh tế khó khăn.

Cho đến nay, nó thực sự đơn giản. Tất cả các khối, như NVMC, bộ nhớ, Cổng truy cập gỡ lỗi… đều được khởi tạo hoàn toàn trong Phần cứng.

Cài lại

Một số nguồn có thể kích hoạt thiết lập lại. Tất cả các nguồn đặt lại và mục tiêu của chúng được tóm tắt trong bảng bên dưới:

Reset source	Reset target								
	CPU	Peripherals	GPIO	Debug[3]	SWJ-DP	RAM	WDT	Retained registers	RESETREAS
CPU lockup[6]	x	x	x						
Soft reset	x	x	x						
Wakeup from System OFF mode reset	x	x		x[7]		x[8]			
Watchdog reset[9]	x	x	x	x		x	x	x	
Pin reset	x	x	x	x		x	x	x	
Brownout reset	x	x	x	x	x	x	x	x	x
Power-on reset	x	x	x	x	x	x	x	x	

Đặt lại bảng

Dòng cuối cùng là thú vị. **Khởi động lại khi bật nguồn sẽ đặt lại toàn bộ SoC, bao gồm cả Cổng gỡ lỗi.**

Kế hoạch hạn chế

Mục tiêu là kích hoạt lại quyền truy cập vào Gỡ lỗi AHB-AP, bất chấp ĐƯỢC PHÊ DUYỆT.

Sau khi thiết lập lại bật nguồn, AHB-AP phải tự khởi tạo theo giá trị của APPROTECT, được lưu trữ trong UICR (phân vùng bộ nhớ Flash chuyên dụng).

Trong quá trình khởi động, Bộ điều khiển bộ nhớ (NVMC) phải giao tiếp với CPU (hoặc trực tiếp với Khối AHB-AP) để cung cấp giá trị APPROTECT được lưu trữ trong Flash, sau đó đặt trạng thái bảo vệ tương ứng.

Do không có BootROM, điều này đạt được bằng Phần cứng thuần túy và nó phải được thực hiện trong quá trình khởi động sớm, trước khi CPU bắt đầu tải từ Flash và thực thi Mã.

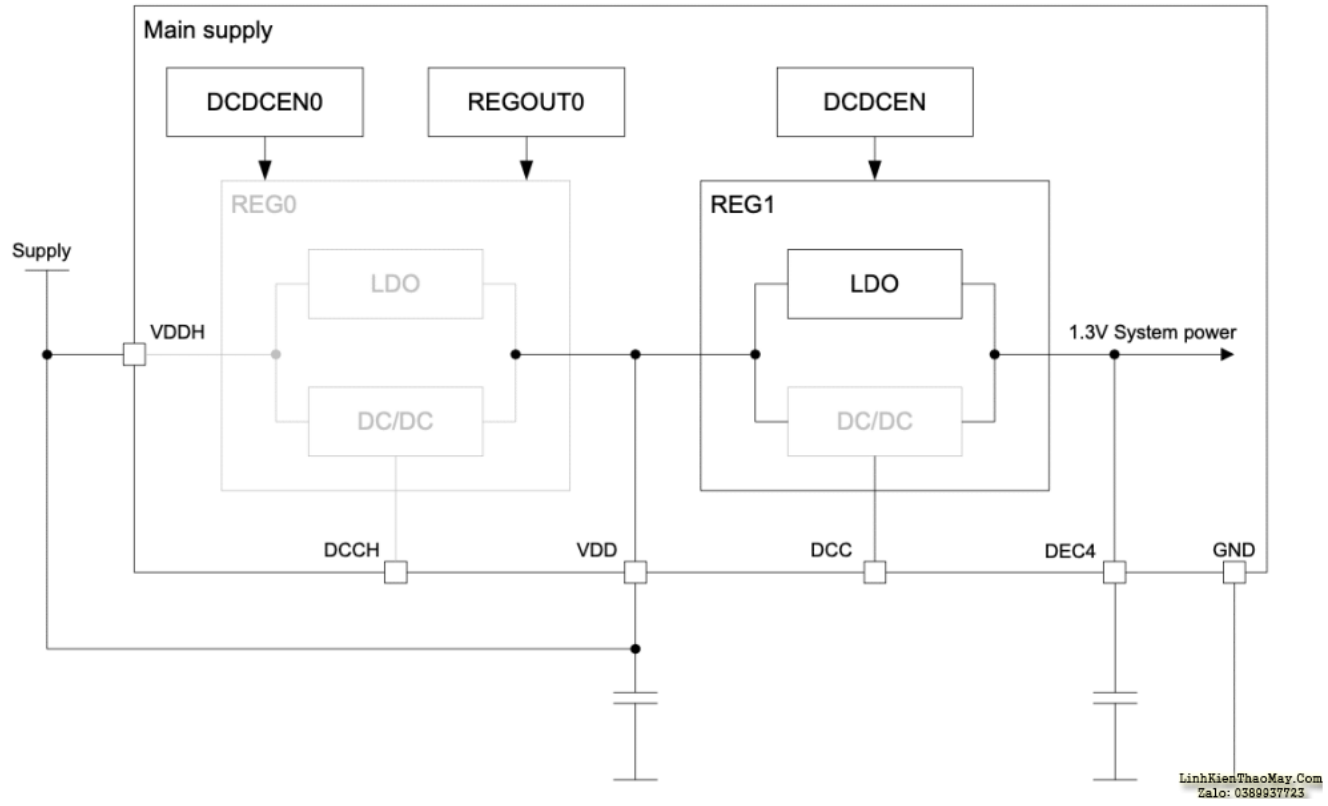
Mục tiêu của mình là xác định "quy trình CTNH thuần túy" này và sau đó cố gắng sửa đổi Khởi tạo gỡ lỗi AHB-AP. Nếu thành công, các tính năng Gỡ lỗi sẽ được kích hoạt lại cho đến lần Đặt lại bật nguồn tiếp theo.

Việc chèn lỗi là (một lần nữa) cách duy nhất để đi đến đây.

Chuẩn bị mục tiêu

Cấu hình nguồn

Hệ thống bao gồm hai giai đoạn điều chỉnh khác nhau, REG0 và REG1. Mỗi giai đoạn bộ điều chỉnh có hai tùy chọn khác nhau và có thể định cấu hình dựa trên Bộ điều chỉnh có độ sụt thấp (LDO) và Bộ điều chỉnh Buck (DC/DC).



Chế độ điện áp bình thường trên nRF52840, chỉ bật LDO

Chỉ có DEC4 được đề cập trên sơ đồ trên.

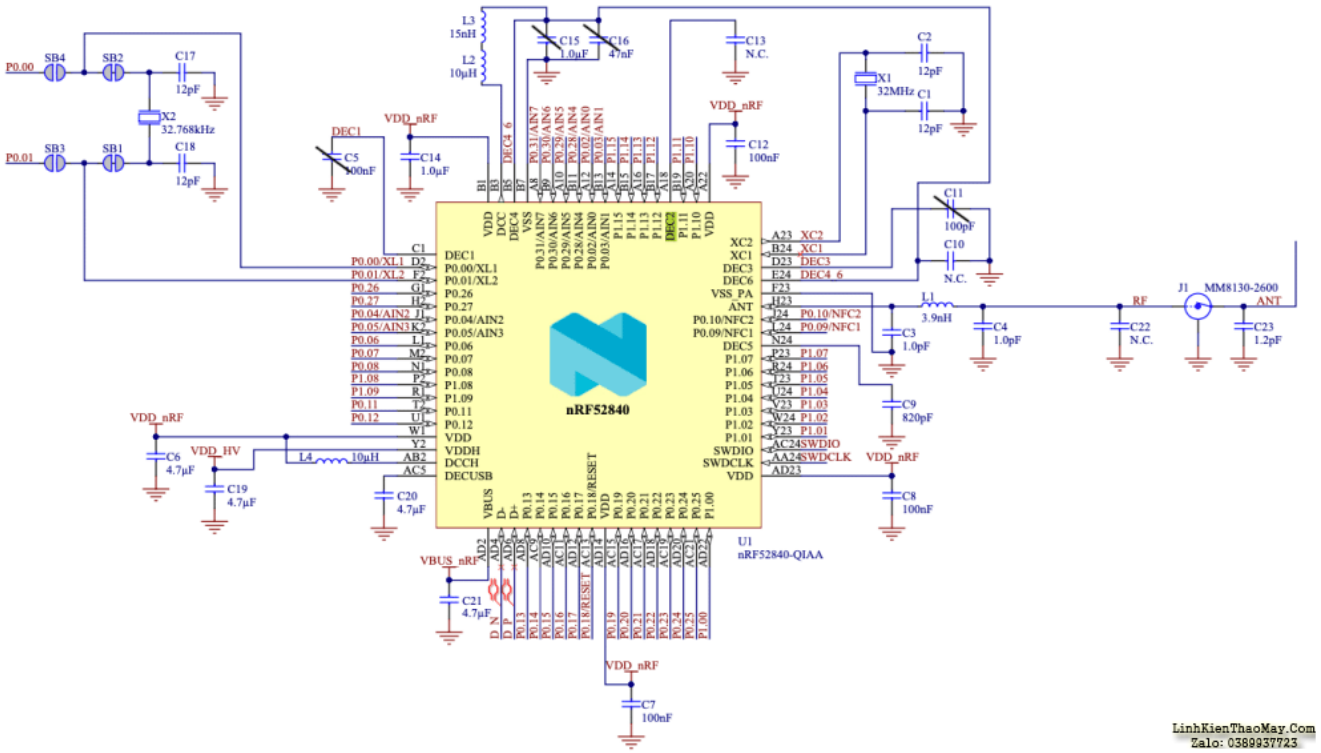
mình đặc biệt quan tâm đến các chân DEC (DEC1, DEC2 cho đến DEC6), nơi các tụ tách rời bên ngoài được kết nối. NordicSemiconductor không cung cấp các chi tiết nào về mạng điện nRF52.

Sau một hồi tìm hiểu nhanh, mình **quyết định tập trung vào DEC4 (1.2V-1.3V) và DEC1 (0.8V-0.9V):**

- **DEC4** là đường dây cấp điện sau giai đoạn REG1. Vì vậy, nó có thể cung cấp toàn bộ hệ thống kỹ thuật số (CPU & bộ nhớ)
- **DEC1** chắc chắn là đường nguồn của CPU.

Sửa đổi PCB

Để phân tích hoạt động bên trong chip, cần sửa đổi một chút PCB:



Sơ đồ PCB. Một số tụ điện đã được loại bỏ.

Các tụ tách tương ứng C5, C15, C16 và C11 được loại bỏ.

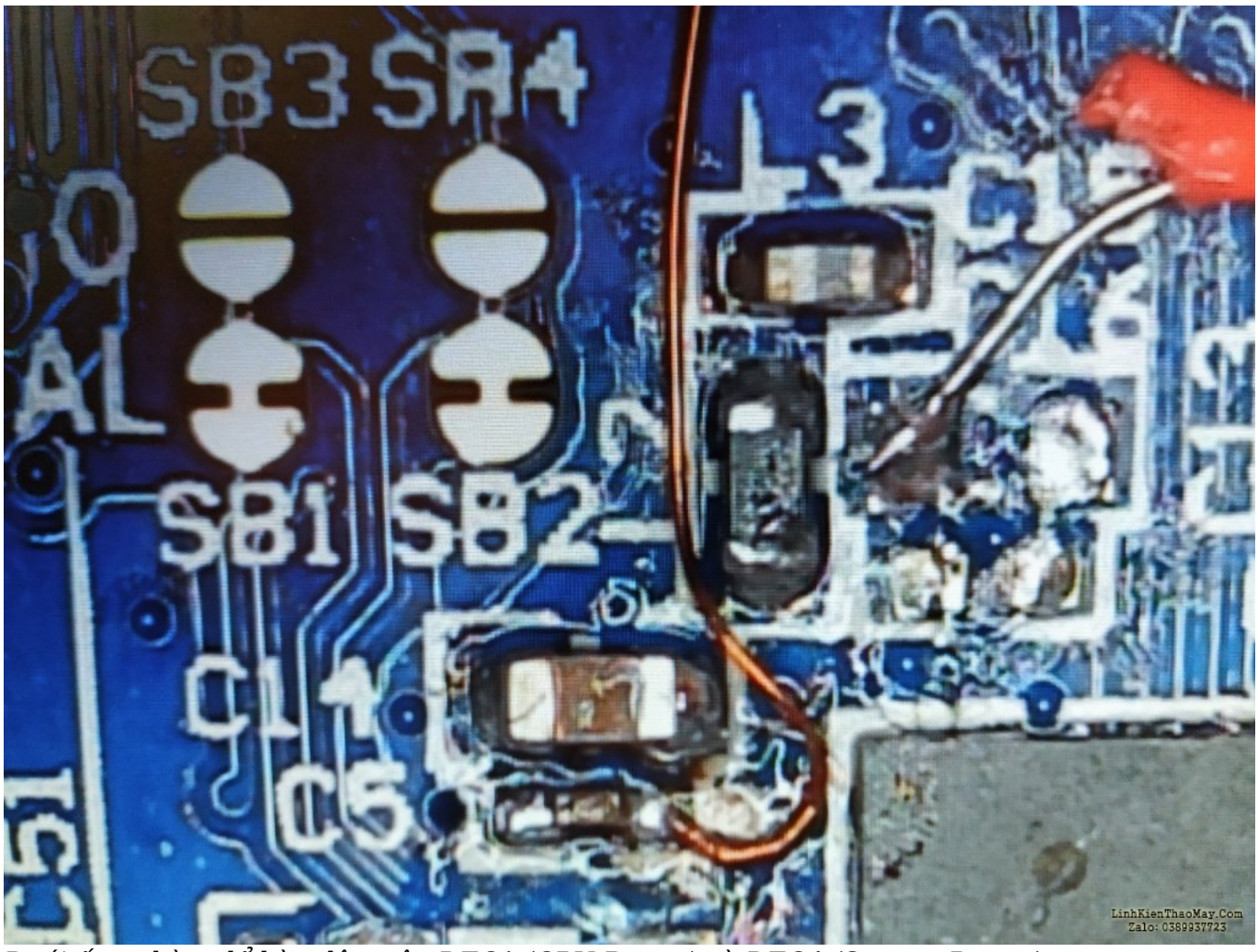
Hệ thống phun lỗi

Gliching điện áp chi phí thấp của mình là một hệ thống điện tử CTNH tự chế, được thiết kế để thực hiện việc chèn lỗi theo cách phù hợp. Tổng chi phí của bảng điện tử này chưa đến 5$, điều này chứng tỏ việc chèn lỗi là một kỹ thuật có chi phí rất thấp và có thể đạt được bởi một số ít tin tặc.

Đầu ra của hệ thống ổn định điện áp chi phí thấp của mình sẽ được kết nối với DEC1 , sử dụng đầu nối SMA đã được cài đặt trước đó.

hàn

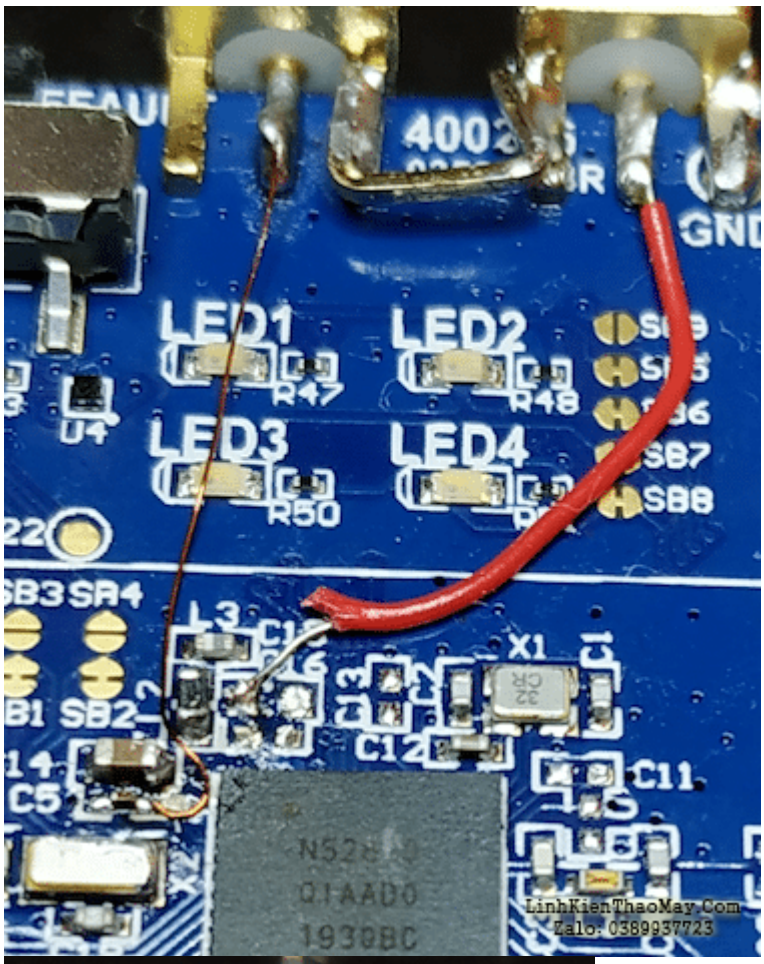
Một dây màu đỏ được hàn vào DEC4 và một dây mỏng được hàn vào DEC1:

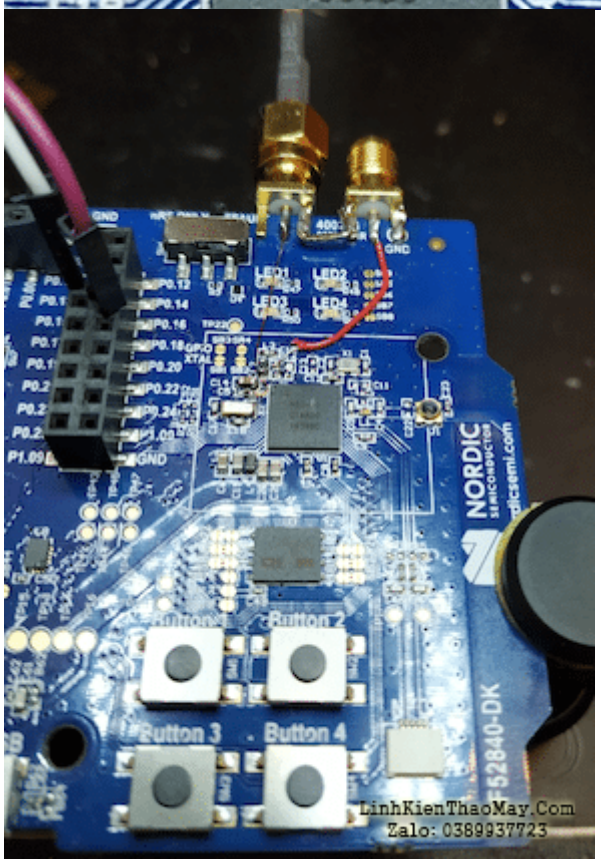


Dưới ống nhòm để hàn dây trên DEC1 (CPU Power) và DEC4 (System Power).

Lưu ý: Mình hàn lại tụ điện 100nF tại vị trí C5 để nguồn điện CPU "ổn định" hơn trong quá trình khởi động. Điều này không thực sự cần thiết nhưng nó có thể giúp ích trong trường hợp CPU gặp sự cố không mong muốn.

mình kết nối hai dây với đầu nối SMA và mình sử dụng điểm GND làm điểm chung:





Cáp micro-USB được sử dụng để cấp nguồn cho toàn bộ bo mạch. Cáp USB nối tiếp được kết nối để hiển thị thông báo UART từ ứng dụng (115200, 8N1).

Cấu hình phạm vi

Máy hiện sóng được kết nối với mục tiêu:

- **CHANNEL1 (CH1) = TxD (chân P0.06)**
- **CHANNEL2 (CH2) = Điện năng tiêu thụ của chip thông qua DEC1**
- **CHANNEL3 (CH3) = Lệnh xung cho debug**
- **CHANNEL4 (CH4) = Điện năng tiêu thụ của chip thông qua DEC4 (Scope Trigger)**

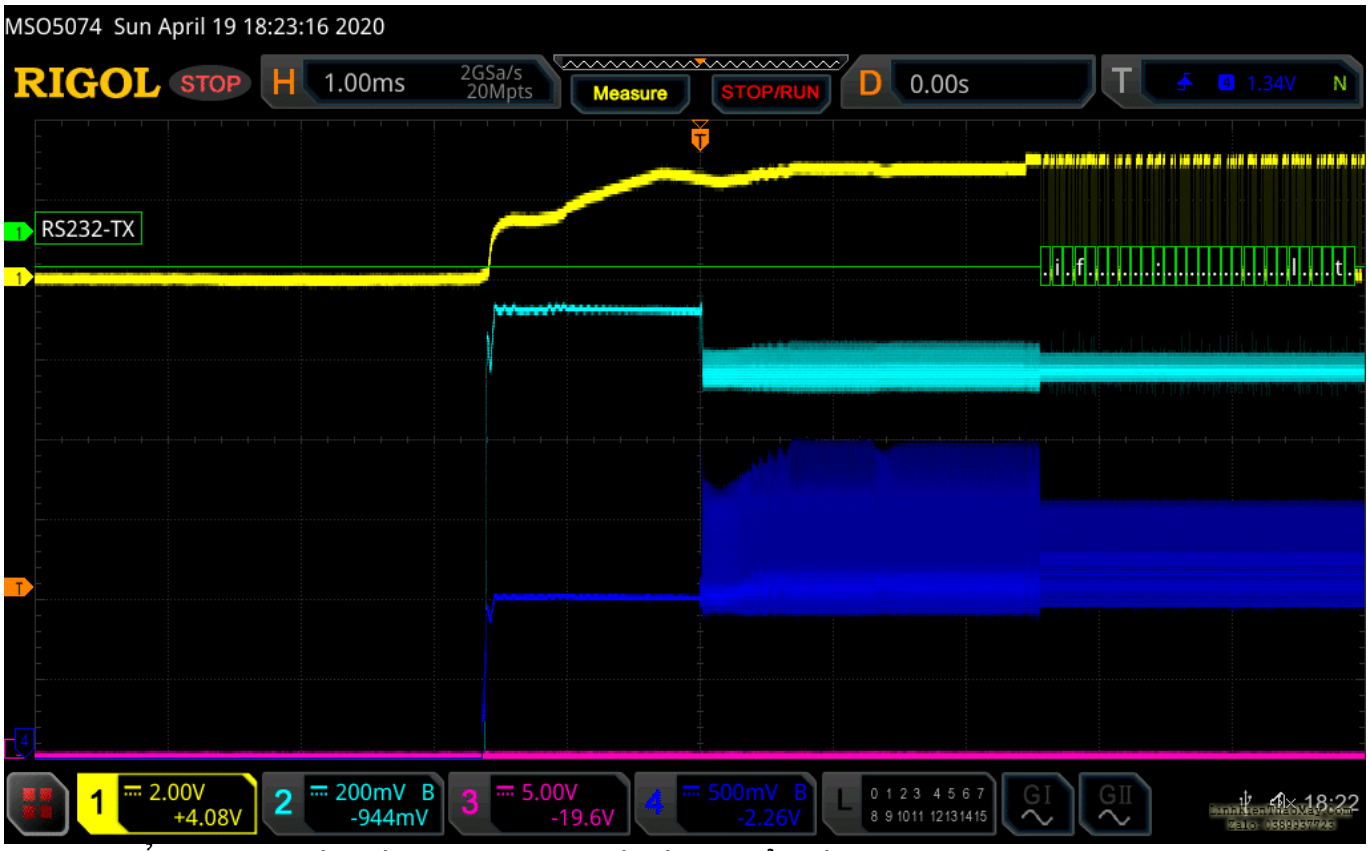
Kết quả

Đảo ngược hộp đen

Mục tiêu là xác định một mô hình cụ thể về mức tiêu thụ điện năng, điều này có thể xác thực giả định trước đó của mình.

Vì mục đích đó, mức tiêu thụ điện năng của CPU (CH2) trên DEC1 và mức tiêu thụ điện năng của hệ thống (CH4) trên DEC4 đều được theo dõi.

Đầu tiên, Hoạt động bộ nhớ Flash có thể phân biệt rõ ràng (tại dấu T kích hoạt) và tương ứng với các thao tác đọc liên tiếp từ Flash:



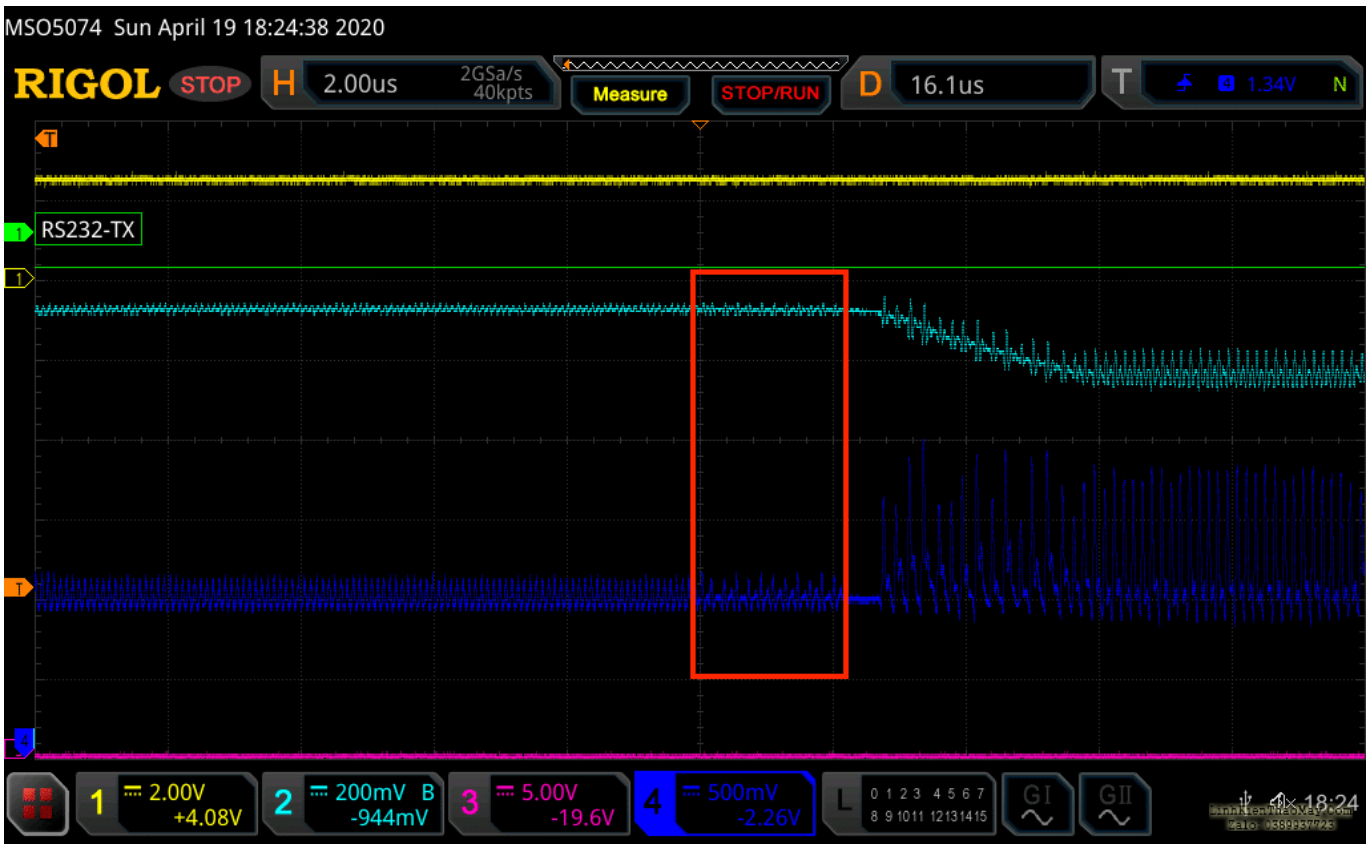
Phạm vi Ảnh chụp màn hình theo dõi quá trình khởi động nRF52840. CH1= UART, CH2 = Nguồn CPU, CH3 = Lệnh debug (không được kích hoạt) và CH4= Nguồn hệ thống.

Trên ảnh chụp màn hình tiếp theo, CPU Arm bắt đầu thực thi (lúc 19us sau dấu kích

hoạt). Có thể phân biệt trên CH2 (mức tiêu thụ điện năng của CPU):

CPU bắt đầu thực thi. CH1= UART, CH2 = Nguồn CPU, CH3 = Lệnh debug (không được kích hoạt) và CH4= Nguồn hệ thống.

Sau đó, mình quyết định tập trung phân tích để tìm ra hoạt động NVMC, bằng cách so sánh hai tín hiệu công suất DEC4 và DEC1:



Khởi tạo NVMC. CH1= UART, CH2 = Nguồn CPU, CH3 = Lệnh debug (không được kích hoạt) và CH4= Nguồn hệ thống.

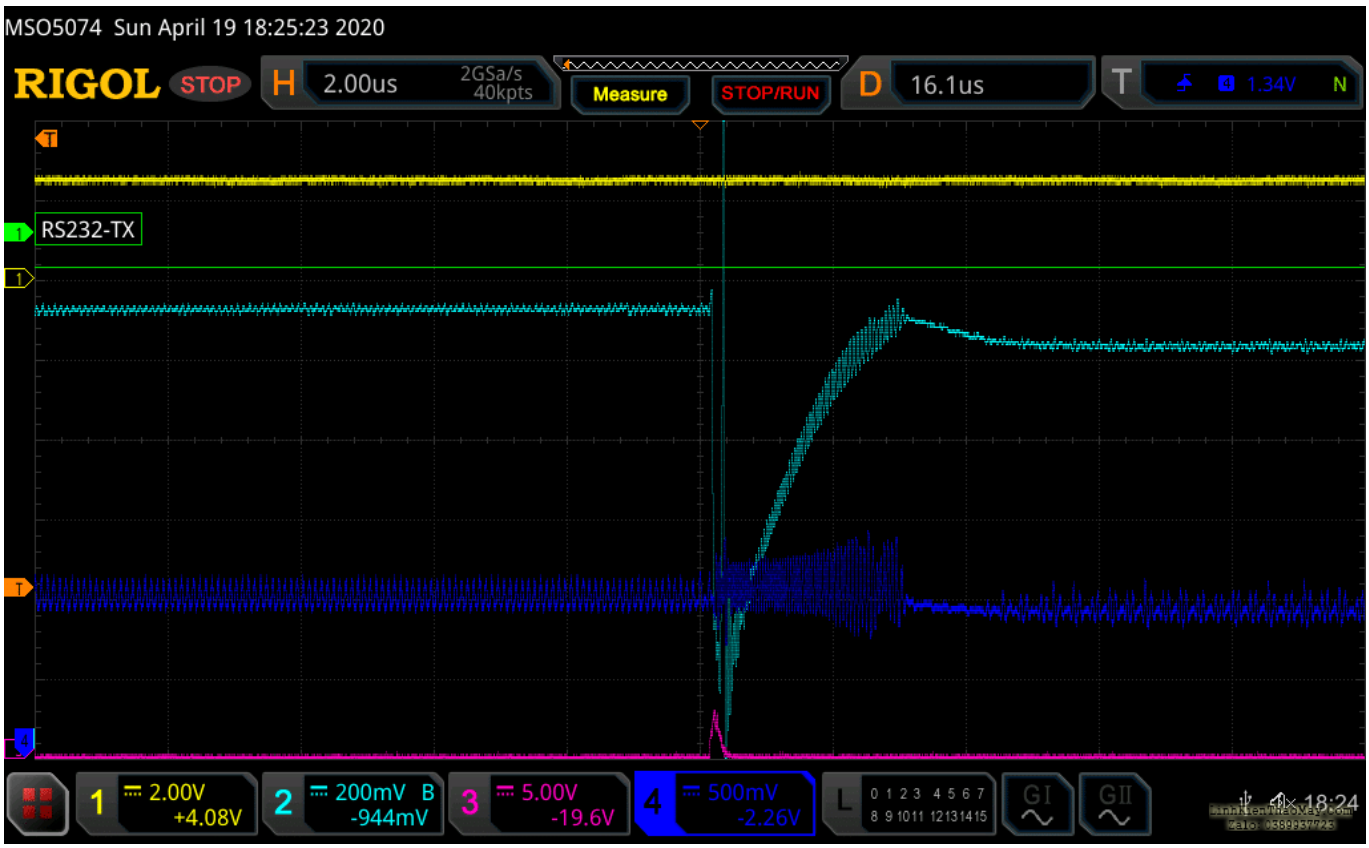
mình khá tự tin mẫu này (**khung màu đỏ**) tương ứng với bộ điều khiển bộ nhớ (NVMC) chuyển các giá trị UICR sang lõi (hoặc trực tiếp đến khối AHB-AP), để định cấu hình Cổng gỡ lỗi (AHB-AP, CTRL-AP) và đặt APPROTECT để cuối cùng tắt giao diện gỡ lỗi.

Bỏ qua APPROTECT trên nRF52840

Một tập lệnh python chịu trách nhiệm trang bị phạm vi, đặt các tham số debug khác nhau và đặt lại bảng.

Một chiến dịch debug được bắt đầu, tạo ra xung rất ngắn trên đường dây điện DEC1 để nhắm mục tiêu cụ thể vào mẫu đã xác định trước đó.

Ảnh chụp màn hình phạm vi tiếp theo cho thấy một lần thử debug thành công:



Khởi tạo NVMC. CH1= UART, CH2 = Nguồn CPU, CH3 = Lệnh long lanh (không được kích hoạt) và CH4= Nguồn hệ thống.

Lưu ý mức tiêu thụ điện năng của Hệ thống được sửa đổi như thế nào sau sự cố. Đây chính là kết quả mà bạn mong muốn đạt được.

Sau mỗi lần thử debug, tập lệnh sẽ phát lệnh OpenOcd sau đây để kết xuất toàn bộ bộ nhớ Flash (điều này sẽ chỉ xảy ra trong trường hợp kết nối thành công với giao diện Gỡ lỗi AHB-AP):

```
$ openocd -s /usr/local/share/openocd/scripts -f ./interface/jlink.cfg -c "transport select swd" -f ./target/nrf52.cfg -c "init;dump_image nrf52_dumped.bin 0x0 0x100000"
```

Cổng gỡ lỗi AHB-AP được mở khóa thành công nhờ hiệu ứng debug. Trình gỡ lỗi bên ngoài hiện có thể kết nối với CPU nRF52840 (cortex-M4F) để kết xuất bộ nhớ Flash:

```
 ----- APPROTECT BYPASS 0 -----
#### OpenOcd test ####
xPack OpenOCD, 64-bit Open On-Chip Debugger 0.10.0+dev (2019-07-17-11:25)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
swd
Info : J-Link OB-SAM3U128-V2-NordicSemi compiled Jan 21 2020 17:30:48
Info : Hardware version: 1.00
Info : VTarget = 3.300 V
Info : clock speed 1000 kHz
Info : SWD DPIDR 0x2ba01477
Info : nrf52.cpu: hardware has 6 breakpoints, 4 watchpoints
Info : nrf52.cpu: external reset detected
Info : Listening on port 3333 for gdb connections


!!! DUMPED !!!
```


Trình gỡ lỗi đã kết nối với nRF52840 mục tiêu mặc dù đã kích hoạt APPROTECT.

Sau đó, phiên Arm GDB sẽ được đính kèm. Ngay cả khi giá trị APPROTECT hiển thị nRF52840 được bảo vệ (0xFFFFFF00), hiện tại vẫn có thể gỡ lỗi mục tiêu, chẳng hạn bằng cách đọc Firmware ở 0x00000000:

```
(gdb) target remote :3333
Remote debugging using :3333
0x000061c4 in ?? ()
(gdb) x/1x 0x10001208
0x10001208:     0xffffff00
(gdb) monitor nrf52.dap apreg 1 0x0c
0x00000001

(gdb) x/10x 0x0
0x0:    0x20040000      0x000002b5      0x000002dd      0x000002df
0x10:   0x000002e1      0x000002e3      0x000002e5      0x00000000
0x20:   0x00000000      0x00000000
(gdb)
```


Phiên GDB được kết nối với nRF52840 mục tiêu mặc dù đã kích hoạt APPROTECT.

PoC: Khả năng gỡ lỗi đầy đủ

Chắc hẳn bạn cũng đoán được cũng có thể điều khiển hoàn toàn thiết bị (vì giao diện gỡ lỗi đã được kích hoạt lại).

Trong quá trình thử của mình, CPU bị kẹt trong vòng lặp vô tận ở 0x61C4 sau debug. Vì vậy, màn hình bên dưới hiển thị cách Bộ đếm chương trình (PC) được thiết lập để thực thi lệnh đầu tiên (ở 0x2b4), sau đó nhập mã dưới dạng Proof-of-Concept:

```
(gdb) target remote :3333
Remote debugging using :3333
0x000061c4 in ?? ()
(gdb) si
0x000061c4 in ?? ()
(gdb) si
0x000061c4 in ?? ()
(gdb) set $pc=0x2b4
(gdb) x/5i $pc
=> 0x2b4:       ldr     r1, [pc, #24]   ; (0x2d0)
   0x2b6:       ldr     r2, [pc, #28]   ; (0x2d4)
   0x2b8:       ldr     r3, [pc, #28]   ; (0x2d8)
   0x2ba:       subs    r3, r3, r2
   0x2bc:       ble.n   0x2c6
(gdb) si
0x000002b6 in ?? ()
(gdb) si
0x000002b8 in ?? ()
(gdb) si
0x000002ba in ?? ()
```


Thực thi mã trên nRF52840 được bảo vệ.

Bên dưới có quyền truy cập R/W vào các thanh ghi (các thao tác SRAM và Flash R/W cũng có sẵn, hãy tin mình):

```
(gdb) i r
r0             0x0      0
r1             0x745c   29788
r2             0x20000000       536870912
r3             0x200000b8       536871096
r4             0x0      0
r5             0x0      0
r6             0x0      0
r7             0x0      0
r8             0x0      0
r9             0x0      0
r10            0x20030000       537067520
r11            0x0      0
r12            0x0      0
sp             0x2003fff8       0x2003fff8
lr             0x610b   24843
pc             0x2ba    0x2ba
xPSR           0x61000000       1627389952
(gdb) set $r5=0x12345678
(gdb) i r
r0             0x0      0
r1             0x745c   29788
r2             0x20000000       536870912
r3             0x200000b8       536871096
r4             0x0      0
r5             0x12345678       305419896
r6             0x0      0
r7             0x0      0
r8             0x0      0
r9             0x0      0
r10            0x20030000       537067520
r11            0x0      0
```

Khả năng gỡ lỗi đầy đủ, giúp việc đảo ngược trong tương lai dễ dàng hơn

PoC này chứng minh khả năng kiểm soát Thực thi luồng và quyền truy cập R/W vào bộ nhớ và thanh ghi.

Truy cập FICR

Các thanh ghi cấu hình thông tin nhà máy (FICR) được lập trình sẵn tại nhà máy và người dùng không thể xóa được. Các thanh ghi này chứa thông tin cụ thể về chip (id thiết bị, khóa, địa chỉ thiết bị...) và được đặt tại 0x10000000:

```
(gdb) x/100x 0x10000000
0x10000000:     0x55aa55aa      0x55aa55aa      0xffffffff      0xffffffff
0x10000010:     0x00001000      0x00000100      0xfffffffe      0xffffffff
0x10000020:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000030:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000040:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000050:     0xffffffff      0xffffffff      0x00000000      0xffff015b
0x10000060:     0x0c0e2806      0xa2c6de33      0xffffffff      0xffffffff
0x10000070:     0x53534e50      0x2f0c3237      0xaa55aa14      0xffffff55
0x10000080:     0x7181a09e      0x99748040      0xdcc01efa      0xa2282d1c
0x10000090:     0x5990e0e5      0x2e3abc6d      0x98c26f54      0x562cc279
0x100000a0:     0xffffffff      0x4c169dee      0x1955b1a1      0xffffffff
0x100000b0:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x100000c0:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x100000d0:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x100000e0:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x100000f0:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000100:     0x00052840      0x41414430      0x00002004      0x00000100
0x10000110:     0x00000400      0xffffffff      0xffffffff      0xffffffff
0x10000120:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000130:     0x00000008      0x00000003      0x00000000      0xffffffff
0x10000140:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000150:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000160:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000170:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
0x10000180:     0xffffffff      0xffffffff      0xffffffff      0xffffffff
```

Đăng ký cấu hình thông tin nhà máy bị đổ

Thêm tính kiên trì vào Khai thác

Cuối cùng, mục tiêu là kích hoạt lại vĩnh viễn toàn bộ khả năng gỡ lỗi trên thiết bị đang được thử. Kịch bản như sau:

- Kết xuất bộ nhớ Flash ở 0x00000000
- Kết xuất UICR ở 0x10001000
- Xóa thiết bị *: nrfjprog -f NRF52 –recover*
- Khởi động lại nRF52 để ghi nội dung bộ nhớ Flash và UICR (tất nhiên với một bản vá thích hợp của giá trị APPROTECT được đặt thành 0xFFFFFFFF).

Sau đó, tính năng Gỡ lỗi qua giao diện SWD được bật và việc tấn công debug là không cần thiết.

Thưởng thức.

Cố lên LR…”hack dev-kit thật dễ dàng”

Có người có thể nói: *“Hack dev-kit bao giờ cũng dễ…”*

OK, bạn nói đúng và mình đồng ý

Để chứng minh việc áp dụng cách khai thác này dễ dàng như thế nào, cuộc tấn công đã được

tái hiện trên một sản phẩm thực, **Chuột Logitech Pro G của mình** .

RIP cô gái nhỏ của mình...

Đây sẽ là **phần 2 của cuộc điều tra** về SoC nRF52. mình sẽ chứng minh **lỗ hổng này hiện đang ảnh hưởng như thế nào đến TẤT CẢ các SoC nRF52, từ nRF52810 đến nRF52840.**

Phần kết luận

APPROTECT Bảo vệ cổng truy cập là tính năng bảo mật mới nhất được thiết kế bởi NordicSemiconductor để bảo vệ khỏi việc đọc chương trình cơ sở và kỹ thuật đảo ngược. Do đó, tính năng APPROTECT này hiện diện trong toàn bộ Dòng SoC nRF52.

Trong bài đăng trên blog này, một cuộc tấn công lỗi chi phí thấp đã được thực hiện thành công trên nRF52840. **Nó cho phép kẻ tấn công có quyền truy cập vật lý để vượt qua APPROTECT để kích hoạt lại giao diện gỡ lỗi SWD vĩnh viễn** (truy cập R/W vào bộ nhớ và thanh ghi, kiểm soát việc thực thi mã CPU, kết xuất bộ nhớ Flash, FICR và UICR...)

Do độ phức tạp thấp, cuộc tấn công vào nRF52840 này có thể được tái tạo một cách dễ dàng trên thực địa. mình chắc chắn rằng các hacker thành thạo có động cơ nào cũng có thể tái tạo cuộc tấn công này trong vòng chưa đầy một ngày và với thiết bị ít hơn 500 đô la.

Lỗ hổng nằm ở Silicon. Không có cách nào để vá mà không sửa đổi CTNH. Điều này đang tác động đến một số lượng lớn các sản phẩm dựa trên nền tảng nRF52 .

Hãy theo dõi Phần 2 .

Tiết lộ dòng thời gian

17/04/2020 : Liên hệ PSIRT Bắc Âu Kết quả có giới hạn.

22/04/2020 : Nordic PSIRT đề xuất mua báo cáo đầy đủ với mức giá thấp nhất.

22/04/2020 : LimitedResults nói “Không, xin lỗi” (và vẫn lịch sự).

28/04/2020 : LimitedResults đề xuất hợp tác với Nordic về việc tiết lộ thông tin có trách nhiệm dựa trên quy trình CVE.

29/04/2020 : Nordic PSIRT tích cực nhưng không đưa ra các đảm bảo nào.

29/04/2020 : LR yêu cầu bảo lãnh hợp lý.

06/08/2020: Sau một tháng im lặng và nhiều lần theo dõi mà không có các phản hồi nào từ Nordic PSIRT, Nordic PSIRT từ chối hợp tác với LimitedResults về việc tiết lộ thông tin có trách nhiệm.

06/10/2020: Đã đăng.

[đã chỉnh sửa] **06/12/2020** : Nordic đã gửi email cho tất cả khách của họ kèm theo thông báo sau đây **tại đây** .

Các bài viết tương tự:

1. 16994. Tivi LCD – TOSHIBA 47L5450
2. canon 2900 – hư ecu đèn nguồn nhấp nháy càn thay thế linh kiện ji nhỉ mấy anh
3. canon lbp 810 – nhấn núc in test tren may thi may chay hoài tói lúc gat reley nguon thi dung lai chu khong in
4. chào các thành viên mình mới làm thêm máy giặt tủ lạnh – mới nhận con máy giặt AW-E920Lv cọn chế độ giặt và cấp nước(ko vặt và xả)thì máy giặt xong tự tăt máy được,,còn nếu chọn giặt có vắt có xả máy giặt xong các quá trình thì ko tự tắt được chỉ hiện về 0 phút nhưng ko tắt(tắt là tắt nguồn)
5. dell – thao den nguon ra kiem tra thick co tu kick 12v,khi gan den vao thi co tieng keu rit rit,nhung den ko len
6. Project trò chơi điện tử trên ESP32 xuất lên màn hình CRT

7. Giải nén file dune_service_XXXX.dsf của Dune
8. j07m04000 – may khong len nguon
9. máy hàn que diện tử 2 pha – do khách cắm điện 1 pha nên em nó bốc khói
10. tcl ic tong av303 – co duong nam ngang khoảng 1 cm va nhao nháy phia tren man hinh
11. tim mua ic STU407D – Hoặc APM4052D hoặc APM4048D
12. Toshiba satellite L500 – main